# فهرست مطالب


| عنوان | صفحه |
|---|---|
| ویرایش دوم | ۵ |
| فصل ۱ | ۵ |
| فصل ۲ | ۹ |
| فصل ۳ | ۱۵ |
| فصل ۴ | ۲۳ |
| فصل ۵ | ۳۰ |
| فصل ۶ | ۳۵ |
| فصل ۷ | ۴۳ |
| فصل ۸ | ۵۱ |
| فصل ۹ | ۵۹ |
| فصل ۱۰ | ۶۸ |
| فصل ۱۱ | ۷۸ |
| فصل ۱۲ | ۸۵ |
| ویرایش سوم | ۹۰ |
| فصل ۱ | ۹۰ |
| فصل ۲ | ۹۶ |
| فصل ۳ | ۱۰۳ |
| فصل ۴ | ۱۱۶ |
| فصل ۵ | ۱۲۴ |
| فصل ۶ | ۱۳۶ |
| فصل ۷ | ۱۴۲ |
| فصل ۸ | ۱۵۳ |
| فصل ۹ | ۱۶۵ |
| فصل ۱۰ | ۱۷۶ |
| فصل ۱۱ | ۱۸۳ |
| فصل ۱۲ | ۱۸۷ |
| ضمیمه ۱: سؤالات تستی امتحانی | ۱۹۷ |
| پاسخ سؤالات تستی امتحانی | ۲۲۲ |
| ضمیمه ۲: سؤالات تشریحی امتحانی | ۲۲۴ |
| پاسخ سؤالات تشریحی امتحانی | ۲۲۸ |
| ضمیمه ۳: سؤالات کنکور ارشد | ۲۳۶ |
| پاسخ سؤالات کنکور ارشد | ۲۴۳ |

# پیشـــگفتار

گرت ز دست برآید چو شمع تابان باش — بســوز خویشتن و جمع را فروزان باش
فضــای تیره‌ی کاشــانه‌ای منور ساز — چو آفتــاب درخشــنده، نورافشان باش
ببند تا بتوانــی کمر به خدمت خلق — بدین وسیله خود اندر پناه یــزدان باش

امروزه فناوری هوش‌مصنوعی جایگاهی شگرف در عرصه‌های مختلف تکنولوژی پیدا کرده است که علاوه بر جلب توجه افراد این رشته، دارای مشتاقان زیادی در بین علوم مختلف می‌باشد. مسلم است که پیشرفت در این علم، منوط به کسب دانش پایه خواهد بود که از بهترین منابع کسب دانش می‌توان به کتاب‌های مفید فارسی اشاره نمود. تاکنون متخصصان و اساتید با تجربه، در مورد هوش‌مصنوعی قلم بر دست گرفته و کتاب مرجع این رشته تالیف استوارت راسل و پیتر نورویگ را ترجمه و منتشر کرده‌اند، ولی دانشجویان همواره نشان داده‌اند که جهت فهم عمیق مطالب نیازمند حل تمرینات بیشتر و آزمودن خود با آنها می‌باشند. لذا بعنوان افرادی کوچک از جامعه بزرگ هوش‌مصنوعی بر آن شدیم تا کتابچه‌ای از تمارین هوش‌مصنوعی به همراه پاسخ آنها تدوین نماییم که مطالعه این کتاب را به کلیه دانشجویان گرایش‌های مختلف کامپیوتر، فناوری‌اطلاعات و علوم کامپیوتر پیشنهاد می‌کنیم. در تهیه این اثر سعی گشته است تا منبعی جامع در اختیار خواننده قرار گیرد، لذا در ابتدا تمامی تمرینات کتاب راسل اعم از ویرایش دوم و سوم را به زبانی ساده و قابل فهم ذکر نموده و سپس در انتهای کتاب، ضمیمه‌هایی متشکل از سئوالات تستی و تشریحی امتحانات پایان ترم و همچنین سئوالات کارشناسی ارشد آورده شده است تا بدین‌وسیله علاقه‌مندان بتوانند آموخته‌های خود را محک بزنند. لازم به یادآوری است که کتاب هوش‌مصنوعی(استوارت راسل- پیتر نورویگ) دارای ۲۷ فصل می‌باشد، اما از آنجایی که دوازده فصل اول آن به عنوان سرفصل مصوب شورای عالی برنامه‌ریزی دانشگاهی درنظر گرفته شده است، ما نیز به پاسخگویی دوازده فصل ابتدایی کتاب فوق بسنده کرده‌ایم.

در پایان بر خود لازم می‌دانیم که از این فرصت استفاده کرده و از پدر و مادر عزیزمان که همواره چراغ وجودشــان روشنــگر راه ما در سختی‌ها و مشکلات بوده است، سپاسگزاری نماییم، هر چند که نمی‌توانیم آن ســـــان که شایسته آنهاست حق خدمت را به جای آوریم. و همچنین بر همراهــی و همگامــی آقایان دکتر آرش حبیبی لشکری و مهندس سید امین اقوامی و رضا فتحی قدردانیم.

باشد که این اثر مورد استفاده اصحاب علم و فضل قرار گرفته و بتوانیم حرکتی ولو کوچک در چرخاندن چرخ علمی کشور انجام داده باشیم.

**شبان‌زاده- فرامرزی**

# فصل ۱ (ویرایش دوم)

◆ ***این تمرینات با هدف ایجاد انگیزه جهت تأمل بیشتر در مباحث، طراحی گشته‌اند که برخی از آنها می‌توانند به عنوان موضوعات پروژه انتخاب شوند. پیشنهاد می‌شود که در ابتدا به طور مقدماتی برای حل مسائل اقدام کرده و پس از تکمیل با رجوع مجدد به آنها، پاسخ‌های خود را بازبینی کرده و به اشکالات خود پی‌ببرید.***

**؟۱.۱ این لغات را به زبان خود تعریف کنید: الف) هوش ب) هوش‌مصنوعی ج) عامل یا کارگزار(Agent)** ***(این تمرین مشابه تمرین ۱.۱ در ویرایش سوم است)***

☑ حل: **الف)** طبق تعریف لغت‌نامه، می‌توان هوش را این گونه تعریف نمود: «ظرفیت کشف و اجرای دانش» یا «استعداد فکری در تفکر و استدلال» و یا «توانایی درک و استفاده از تجربیات». تمامی این پاسخ‌ها قابل‌قبول تلقی می‌شوند ولی اگر بخواهیم تعریفی سنجیده ارائه دهیم، می‌توان هوش را به صورت «توانایی استفاده از دانش به منظور اجرای بهتر در یک محیط» تعریف نمود. **ب)** هوش‌مصنوعی مطالعه برنامه‌های یک کارگزار تعریف می‌شود که بتواند در یک محیط مشخص و به ازای یک معماری کارگزار خاص، به خوبی عمل کرده و قابل اجرا باشد. به بیانی دیگر هر روبات یا نرم‌افزار که برای انجام یک کار، عملکردی هوشمندانه از خود نشان دهد و مسلم باشد که این هوش طبیعی و ذاتی نیست، گفته می‌شود که دارای هوش‌مصنوعی است. **ج)** عامل یا کارگزار در واکنش به ادراکاتی که از یک محیط دارد، عکس‌العمل نشان می‌دهد. این عامل می‌تواند یک روبات و یا نرم‌افزاری خودکار باشد که درون آن برنامه‌هایی نصب می‌شود تا بتواند به طور خودکار کاری را انجام دهد. هر کارگزار محیط پیرامون خود را با توجه به وسایل تعبیه شده در آن درک کرده و سپس اقداماتی در آن انجام می‌دهد. روبات جاروبرقی دو سنسور دارد: یکی برای تعیین اتاق (مانند GPS) و دیگری برای مشاهده آشغال بر روی زمین (مانند دوربین). پس از دریافت این ورودی‌ها و با توجه به برنامه تعبیه شده در آن، اقداماتی انجام می‌دهد: حرکت به اتاق بعدی (با استفاده از چرخ)، مکیدن آشغال (با استفاده از موتور ). از آنجا که این کار به نظر هوشمند آمده و به طور خودکار انجام می‌شود، برنامه مربوطه در حیطه هوش‌مصنوعی قرار می‌گیرد. به طور خلاصه کارگزارها برای عهده گرفتن قسمتی از وظایف انسانی طراحی می‌شوند مثلا به جای آنکه یک خدمتکار همواره اتاق‌های خانه را چک کند تا در صورت کثیفی جارو نماید، و یا به جای آنکه یک انسان پشت ماشین نشسته و مسافتی را بپیماید تا مسافران را به مقصد برساند، می‌توان از انسان در سایر زمینه‌های سطح بالاتر کمک گرفته و برای این وظایف از کارگزارهای ساخته دست بشر استفاده نمود.

**؟۲.۱ مقاله اصلی نظریه تورینگ در مورد هوش‌مصنوعی (Turing,1950) را مطالعه کنید. در این مقاله، نویسنده تعدادی از مخالفت‌های ممکن با تست هوشمندی و نظریه ارائه شده خود را مطرح کرده و سپس آنها را تکذیب می‌کند. کدام یک از این مخالفت‌ها همچنان برقرارند؟ آیا تکذیب‌های وی معتبر است؟ آیا شما می‌توانید مخالفت‌های جدیدی به غیر از آنچه در زمان نوشتن مقاله ارائه شده است، بیابید؟ نویسنده در مقاله پیش‌بینی می‌کند که تا سال 2000، یک کامپیوتر 30 درصد شانس گذراندن 5 دقیقه از تست تورینگ را داشته باشد به شرط آنکه طرف مقابل ماهر نباشد. به نظر شما شانس کامپیوتر امروزه چقدر است؟ در 50 سال بعد چطور؟** ***(این تمرین مشابه تمرین ۲.۱ در ویرایش سوم است)***

☑ حل: می‌دانیم که در تست تورینگ دیواری داریم که در یک طرف انسان و در طرف دیگر یک روبات قرار داده شده است. اگر این دو با هم رابطه برقرار کرده و در طی این ارتباط انسان نفهمد که طرف مقابلش یک روبات است و یا یک انسان، آنگاه می‌گوییم تست تورینگ پیروز شده است. ولی در این تست، احتمال فریب‌دادن طرف مقابل بستگی زیادی به میزان ناشی‌بودن و یا مهارت طرف مقابل دارد. یک داوطلب جایزه لوبنر[1] در سال 2002 توانست در رقابتی مشابه با شرایط تست تورینگ، با فریب یک سئوال، برنده شود. اگرچه با مراجعه به دانسته‌های وی، تشخیص چگونگی تفکرش در رسیدن به پاسخ، بسیار دشوار ارزیابی شد. البته امروزه مثال‌های متعددی از عامل‌های هوشمند مانند روبات‌های چت(Chatbot) به طور آنلاین طراحی شده‌اند تا برای فریب انسان‌ها بکار روند به طوری که انسان در پشت سیستم نمی‌فهمد که طرف مقابل وی در چت یک انسان است و یا یک روبات، حتی با پرسیدن سئوالات متفاوت از او نیز به این ماجرا پی نمی‌برد. به عنوان مثال به حساب See Lenny Foner در روبات چت Julia به آدرس:

www.media.mit.edu/people/foner/Julia

مراجعه نمایید. امروزه شانس برنده‌شدن کامپیوترها در تست تورینگ به حدود 10 درصد کاهش یافته‌است زیرا تنوع مهارت‌های طرف مقابل نسبت به پیشرفت برنامه‌ها، افزایش قابل‌ملاحظه‌ای دارد. ولی در 50 سال آینده، انتظار داریم که شانس کامپیوترها در شکست انسان بیشتر شده و سرمایه‌گذاری بیشتری در صنعت سرگرمی (فیلم، بازی‌های ویدیویی، تجارت و ...) جهت ایجاد و بکارگیری بازیکنان مصنوعی شود تا بتوان بازیکنانی باورنکردنی ایجاد نمود.

---

[1] هر ساله جایزه لوبنر به برنامه‌ای اعطا می‌شود که بتواند در نسخه‌ای از تست تورینگ بهتر از دیگران عمل نماید.

**؟۳.۱ هر ساله جایزه لوبنر(Loebner prize) به برنامه‌ای اهدا می‌شود که بتواند به طور تقریبی نسخه‌ای از تست تورینگ را با موفقیت بگذراند. در مورد آخرین برنده جایزه لوبنر تحقیق کرده و گزارشی ارائه دهید. این برنده از چه تکنیکی استفاده کرده است؟ این تکنیک چگونه می‌تواند موجب ارتقای هوش مصنوعی شود؟**

☑ حل: جایزه لوبنر در سال 2002 به برنامه Kevin Copple با نام ELLA داده شد. این برنامه شامل مجموعه‌ای از قواعد الگو/ واکنش بود یعنی تعدادی الگوی مشخص برای کارگزار تعریف شده بود که در صورت رخداد آنها بداند چه کاری را باید انجام دهد و چه متنی را بخواند. سپس در زمان انجام تست تورینگ اگر یک انطباق با یک الگوی ذهنی‌اش را مشاهده می‌کرد، آنگاه واکنش و پاسخ مربوط به آن را می‌دانست که به سرعت انجام می‌داد. البته این برنامه شامل پایگاه داده بزرگی از متون و دیکشنری آنلاین Worldnet بود تا بتواند برای پاسخ به هر سئوال، جوابی آماده داشته باشد که در این کار توانست با استفاده از ابزارهای قاعده‌مند به نتیجه دست‌یابد و نه تئوری‌های هوش‌مصنوعی. زیرا برای انجام این کار به هیچ هوشی نیاز نبوده و تنها از حافظه کمکی استفاده شده بود. این برنامه در حال جمع‌آوری مدارکی است که تعداد و نوع قواعد مناسب برای تولید یک نوع از ارتباط را بداند.

**؟۴.۱ برخی از مسائل دارای ماهیتی لجوجانه هستند که حل آنها برای کامپیوتر دشوار است و در برخی مسائل دیگر تصمیم‌گیری ناممکن می‌باشد. آیا این بدان معناست که وجود هوش‌مصنوعی غیرممکن است؟**

☑ حل: خیر. این موضوع بدان معناست که سیستم‌های هوش‌مصنوعی نباید همیشه به فکر حل موثر و کامل مسائل باشند زیرا معمولا آنها می‌توانند فقط رفتارهای بهینه را تخمین بزنند. البته توجه داشته باشید که حتی انسان‌ها نیز نمی‌توانند مسائل NP-کامل که پیچیدگی زیادی دارند را حل کنند. با این حال هوش‌مصنوعی در حل برخی مسائل خاص با داشتن تعداد زیادی ساختار یا بر مبنای دانش پس‌زمینه‌ای که در آنها قرار داده می‌شود، می‌تواند عملکرد خوبی داشته باشد. سیستم‌های هوشمند باید سعی کنند این چنین باشند.

**؟۵.۱ فرض کنید در بخشی از یک برنامه آنولوژی تغییراتی اعمال کنیم که بتواند در یک آزمون هوش استاندارد، امتیاز 200 را کسب کند. آیا این برنامه، هوشمندتر از انسان عمل می‌کند؟ توضیح دهید. *(این تمرین مشابه تمرین ۴.۱ در ویرایش سوم است)***

☑ حل: خیر. امتیازات کسب‌شده در آزمون هوش، بستگی به عوامل مختلفی چون موفقیت در دانشگاه دارد، ولی به شرط آنکه فقط انسان‌های نرمال و ایده‌آل را مورد بررسی قرار دهیم. همچنین آزمون هوش همه‌منظوره نیست پس یک برنامه که به طور خاص فقط برای کسب امتیاز در آزمون‌های هوش و یا حتی فقط بخش آنالوژی طراحی شود، در سایر زمینه‌های آزمون‌هوش بسیار ضعیف خواهد بود. بنابراین نمی‌توان چنین برنامه‌ای را هوشمندتر از انسان در تمام مسائل دانست. به مقاله‌ای که تحت عنوان *The Mismeasure Of Man* توسط استفان‌جی‌گلد و نورتن در سال 1981 ارائه شد و نیز مقاله هاوارد گاردنر با عنوان *Multiple intelligence: the theory in practice* درسال 1993 مراجعه کنید.

**؟۶.۱ چطور خویشتن‌نگری(Introspection) (گزارشی از روند تفکر داخلی) می‌تواند غیردقیق باشد؟ آیا من می‌توانم در مورد چیزی که فکر می‌کنم، اشتباه کنم؟ بحث کنید. *(این تمرین مشابه تمرین ۶.۱ در ویرایش سوم است)***

☑ حل: همان طور که شما از تمام مراحل صورت گرفته برای ایجاد یک ضربان قلب خود آگاه نیستید، از بیشتر اتفاقاتی که در تفکرتان می‌گذرد نیز ناآگاهید. البته ممکن است از برخی از فرآیندهای فکری مطلع باشید ولی همچنان اکثریت آنها به طور مبهم و غیرشفاف هستند. به همین منظور رشته روانکاوی بر این ایده بنا شد تا یک نفر به طور حرفه‌ای آموزش دیده و بتواند دیگران را در تحلیل تفکرشان یاری نماید.

**؟۷.۱ با توجه به مبحث هوش‌مصنوعی، هم اکنون کدام یک از این مسائل توسط کامپیوتر قابل انجام است: الف) انجام بازی قابل قبولی از تنیس روی میز (پینگ‌پنگ) ب) رانندگی در مرکز شهر قاهره ج) خرید خواروبار مورد نیاز هفتگی در فروشگاه د) خرید خواروبار مورد نیاز هفتگی در وب هـ) انجام بازی قابل قبولی از بریج به طور رقابتی و) کشف و اثبات تئوری‌های جدید ریاضی ز) نوشتن یک داستان خنده‌دار بطور عمدی ح) ارائه مشورتی حقوقی درمورد مسئله‌ای خاص از قانون ط) ترجمه بی‌درنگ مکالمات انگلیسی به مکالمات سوئدی ی) انجام یک عمل جراحی پیچیده سعی کنید. برای مواردی که هم اکنون در جامعه موجود نیستند، مشکلات را یافته و سپس پیش‌بینی کنید که چه زمانی این مشکلات برطرف خواهند شد. *(این تمرین مشابه تمرین ۱۴.۱ در ویرایش سوم است)***

☑ حل: **الف)** پینگ‌پنگ: این مسئله قبلا توسط روبات اندرسون انجام شده و کارایی قابل‌قبولی نیز ارائه داده است (Anderson,1988). **ب)** رانندگی در مرکز شهر قاهره: خیر. رانندگی اتوماتیک مسئله‌ای است که عوامل مختلفی در آن دخیل‌اند به همین خاطر معمولا در این گونه سیستم‌ها فرض می‌شود که عوامل موثر، نسبتا ثابت باشند: مثلا جاده دارای خط وسط و شانه خاکی باشد، جلوی ماشین طبق مسیر پیش‌بینی شده حرکت کند، ماشین‌ها در همان طرف خود حرکت کنند و .... به علاوه ممکن است به منظور مدرنیزه کردن ترافیک، تعدادی از جاده‌های یک شهر بزرگ در طی روز دچار تغییراتی شوند. بنابراین مجموع این شرایط نشان می‌دهد که رانندگی در شهر قاهره بسیار غیرقابل پیش‌بینی خواهد بود و برای یک روبات میسر نمی‌باشد. **ج)** خرید در فروشگاه: خیر. هیچ روباتی نمی‌تواند همزمان وظایفی چون حرکت در محیط‌های شلوغ، دیدن و تشخیص اشیا در بین انواع مختلف و گرفتن شی مورد نظر بدون آسیب‌زدن به سایرین را انجام دهد. شاید هریک از این وظایف به تنهایی قابل‌انجام باشد ولی انجام مجموعه آنها به طور همزمان کاری بس دشوار است. **د)** خرید در وب: بله- اگر سایت فروش به گونه‌ای طراحی شده باشد که در طی زمان تغییرات اساسی نکند، روبات‌های نرم‌افزاری قادر به انجام وظایف مشخص می‌باشند. **هـ)**

انجام بازی بریج: بله. هم‌اکنون برنامه‌هایی مانند GIB در یک سطح قابل‌قبول کار می‌کنند. **و)** اثبات قضیه: بله . به عنوان مثال ثبات جبری ROBBIN که در فصل‌های بعدی بیان شده است. **ز)** داستان خنده‌دار: خیر. شاید برخی از شعرها و متون ادبی که توسط کامپیوتر تولید می‌شود بسیار جالب و خنده‌دار باشد ولی امری غیرعمدی است. البته در برخی موارد کامپیوتر یک شعر یبا را بازگو می‌کند که در حافظه آن قرارداده شده است. در این حالت برنامه فقط از روی حافظه متن را بازگو کرده و در حالت لی سرودن شعر و متن برای وی میسر نیست. **ح)** مشورت قانونی: بله. در تاریخچه هوش‌مصنوعی، نرم‌افزارهای استدلال خودکار نطقی قدمتی زیاد دارند. به عنوان مثال سیستمی خبره برای دولت UK مبتنی بر پرولوگ طراحی و استفاده گشت تا بتواند فراد عادی را در ریزه‌کاری‌های امنیت اجتماعی و قوانین ملی راهنمایی کند. گفته می‌شود که این سیستم امنیت ملی برای ولت UK توانست حدود 150 میلیون دلار در اولین سال عملکردش ذخیره کند. سپس این نرم‌افزار را در مناطق پیچیده نظیر وافق قوانین برای کدگذاری دانش وب گسترش دادند که مربوط به تراکنش‌های تجاری و موافقت‌نامه‌ها و کسب‌وکارها می‌شد. **ط)** ترجمه: بله. اینکار در حالت محدود تقریبا انجام شده است. به مباحث انجام شده توسط کی‌گاردن و نوروین (سال 1994) و السنر (سال 2000) که در زمینه ترجمه صحبت و برخی محدودیت‌های کنونی بحث می‌کند، مراجعه کنید. **ی)** جراحی: بله. وبات‌ها به طور گسترده جهت عملیات‌های جراحی استفاده می‌شوند البته آنها تحت دستورات پزشک اقدام می‌کنند.

**؟۸.۱ برخی نویسندگان ادعا می‌کنند که ادراکات و مهارت‌های حرکتی یک روبات، مهمترین بخش هوشمندی تلقی می‌شود و سایر قابلیت‌های سطح بالاتر، اهمیت زیادی ندارند (به سادگی می‌توان این تسهیلات و امکانات را به آنها افزود). می‌دانیم که بیشتر تکامل و بخش بزرگی از مغز به همین ادراکات و مهارت‌های حرکتی اختصاص دارد در حالیکه هوش‌مصنوعی در بسیاری از واقع توانسته است کارهایی چون بازی و استنتاج منطقی را ساده‌تر از ادراک و مهارت‌های حرکتی انجام دهد. با توجه به این وضیحات، آیا فکر می‌کنید تمرکز زیاد هوش‌مصنوعی بر قابلیت‌های شناختی سطح بالاتر (نظیر استدلال)، کاری اشتباه است؟**

☑ حل: مسلم است که ادراکات و مهارت‌های حرکتی یک روبات مبحثی پراهمیت در زمینه بینایی و روباتیک محسوب می‌شوند شاید بتوان از آنها به عنوان هسته هوش‌مصنوعی یاد کرد زیرا انجام چنین کارهایی نیاز به محاسبه و مهندسی دقیق دارد)، ولی کارگزار پس از هر درک از محیط، بایستی تصمیم‌گیری کرده و سپس واکنشی را انجام دهد. این موضوع به همان گونه که در نیای حقیقی برای انسان انجام می‌شود، در دنیای کوچک هوش‌مصنوعی مانند بازی شطرنج نیز باید اجرا شود. بنابراین محاسبه انجام واکنش نیز برای هوش‌مصنوعی با وجود بخش‌های حرکتی و ادراکی که بر روی کارگزار تعبیه شده است، همچنان دشوار ست. به عبارت دیگر، تمرکز بر روی جزئیات جهان موجب می‌شود تا از عوامل حیاتی و مورد نظر در محیط غافل شویم.

**؟۹.۱ به چه علت در سیستم‌های عقلانی، تکامل(Evolution) حاصل می‌شود؟ این سیستم‌ها برای رسیدن به چه اهدافی طراحی شده‌اند؟** ***(این تمرین مشابه تمرین ۹.۱ در ویرایش سوم است)***

☑ حل: تکامل به معنای تولید مجدد و در نتیجه بقای اندام‌هایی است که به قدرکافی موفق بوده‌اند و در این تکامل، اندام‌هایی مورد توجه بیشتر قرار می‌گیرند که با دوام سن بلوغ جنسی، معیار کارایی را بهینه کنند. از آنجا که عقلانیت فقط به معنای هینگی معیار کارایی است بنابراین در راستای تکامل قرار دارد.

**؟۱۰.۱ آیا اقدامات واکنشی (نظیر پس کشیدن دست از روی اجاق داغ)، عقلانی هستند؟ آیا آنها هوشمند تلقی می‌شوند؟** ***(این تمرین مشابه تمرین ۳.۱ در ویرایش سوم است)***

☑ حل: این کار عقلانی است زیرا هر واکنش دیگری که کندتر از این و با مشورت بیشتری انجام شود، موجب آسیب رساندن یشتر به دست شما می‌شود. حال اگر هوش را به معنای «اجرای دانش» یا «استفاده از تفکر و استدلال» تعریف کنیم، برای انجام چنین عکس‌العملی نیاز به هوشمندی نداریم.

**؟۱۱.۱ «مسلما کامپیوترها هوشمند نیستند زیرا فقط می‌توانند آن چیزی را انجام دهند که برنامه‌نویسان به آنها می‌گویند» آیا جمله آخر صحیح است؟ آیا جمله آخر، دلیلی برای جمله اول محسوب می‌شود؟** ***(این تمرین مشابه تمرین ۱۱.۱ در ویرایش سوم است)***

☑ حل: این موضوع بستگی به تعریف شما از «هوش» و «tell[2]» دارد. از یک جهت، کامپیوترها فقط می‌توانند دستوراتی را اجرا نند که برنامه‌نویسان به آنها گفته‌اند ولی از جهت دیگر، چیزی که برنامه‌نویس از کامپیوتر برای انجام می‌خواهد، معمولا خیلی

---

هر روبات درون خود مجموعه دانشی در مورد کاری که قرار است انجام دهد دارد. به عنوان مثال یک دکتر را درنظر بگیرید. این فرد قبل از حصیل در این رشته معلوماتی اندک در مورد پزشکی دارد مانند اینکه «فرد سرماخورده سردرد دارد». به این دانش اولیه، دانش پس‌زمینه فته می‌شود. این فرد با تحصیل در رشته پزشکی به مرور مجموعه دانش خود را اضافه می‌کند. هر اطلاعات جدید در قالب یک جمله به او ضافه می‌شود. این کار را با نماد tell نشان می‌دهیم. مثلا tell(قرص استامینوفن برای سرماخوردگی خوب است)، tell(مصرف بیش از چهار دد قرص در روز موجب سکته می‌شود) و ... . با مطالعه هر چه بیشتر مجموعه دانش این پزشک که به آن پایگاه‌دانش (KB) می‌گوییم کمیلتر شده و می‌تواند با استفاده از آن پایگاه، استنتاج نماید مثلا به بیماری که سردرد دارد، سه عدد قرص استامینوفن تجویز نماید. دکتر مکن است در بسیاری از موارد برایش سئوالی ایجاد شود مانند اینکه «سردرد نشانه کدام بیماری است؟»، برای پاسخ آن باید به مجموعه انش ذهنی خود مراجعه کند که آن را به طور نمادین با Ask(سردرد نشانه کدام بیماری است) نمایش می‌دهیم. پس برای افزودن یک جمله انش به مجموعه پایگاه دانش در روبات از نماد tell و برای سئوال پرسیدن از پایگاه از دستور Ask استفاده می‌کنیم. اگر دقت کنید پزشکی ه مطالعات بیشتر داشته باشد در شناسایی بیماری و تجویز نسخه ماهرتر است زیرا پایگاهی کامل‌تر دارد.

کمتر از آن چیزی است که کامپیوتر واقعا می‌تواند انجام دهد. احتمالا شما نیز این موضوع را در زمانی که یک برنامه معمولی یا یک برنامه یادگیری‌ماشین حرفه‌ای نوشته‌اید به خوبی درک نموده‌اید که قابلیت کامپیوتر همیشه بیشتر از برنامه‌ای است که برای آن نوشته می‌شود. بنابراین از یک سو، ساموئل با استفاده از tell از کامپیوتر می‌خواهد: «بیاموز که بازی چکر را بهتر از من بازی کنی و این شیوه را برای بازی ادامه بده» و از سوی دیگر می‌توان با استفاده از tell از وی بخواهیم: «این الگوریتم یادگیری را ادامه بده». و این گونه کامپیوتر با توجه به مجموعه دانستنی‌هایش (پایگاه دانش) می‌آموزد که چگونه بازی کند. اکنون در موقعیتی هستیم که نمی‌دانیم یادگیری بازی چکر نشانی از هوش است و یا خیر. (یا شاید شما فکر کنید که یادگیری بازی به شیوه درست، نیازمند هوش است ولی به این شیوه خیر). یا حتی ممکن است نظر شما بر آن باشد که هوشمندی درون برنامه‌نویس یا حتی کامپیوتر نهفته است.

**؟ ۱۲.۱ «مسلما حیوانات هوشمند نیستند زیرا فقط می‌توانند آن چیزی را انجام دهند که ژن‌هایشان به آنها می‌گویند» آیا جمله آخر صحیح است؟ آیا جمله آخر، دلیلی برای جمله اول محسوب می‌شود؟** ***(این تمرین مشابه تمرین ۱۲.۱ در ویرایش سوم است)***

☑ حل: پاسخ این تمرین به نوعی مشابه پاسخ تمرین قبل است. بنابراین همان طور که در مورد هوشمندی کامپیوتر تصمیم‌گیری نمودیم، می‌توان همان نتیجه را در مورد حیوانات نیز تعمیم داد با این تفاوت که دلیل ما برای هوشمندی یک چیز، در مکانیزم عملکرد آن نهفته است پس برنامه‌نویسی در ژن‌ها را متناظر با برنامه‌نویسی انجام شده توسط انسان در نظر بگیرید. لازم بذکر است که Searle این مکانیزم را برای آرگومان‌های اتاق چینی استفاده کرده است.

**؟ ۱۳.۱ «مسلما حیوانات، انسان‌ها و کامپیوترها نمی‌توانند هوشمند باشند زیرا فقط می‌توانند آن چیزی را انجام دهند که قوانین فیزیک به اتم‌های سازنده‌شان دستور می‌دهند». آیا جمله آخر صحیح است؟ آیا جمله آخر، دلیلی برای جمله اول محسوب می‌شود؟** ***(این تمرین مشابه تمرین ۱۳.۱ در ویرایش سوم است)***

☑ حل: پاسخ تمرین ۱۱.۱ را می‌توان برای این تمرین تعمیم داد.

# فصل ۲ (ویرایش دوم)

**؟ ۱.۲ این اصطلاحات را به زبان خود تعریف کنید: الف) کارگزار ب) تابع کارگزار(Agent function) ج) برنامه کارگزار( Agent rogran) د) عقلانیت(Rationality) هـ) خودمختاری(Autonomy) و) کارگزار واکنشی(Reflex agent) ز) کارگزار بتنی بر مدل(Model-based agent) ح) کارگزار مبتنی بر هدف(Goal-based agent) ط) کارگزار مبتنی بر سودمندی(Utility-based agent) ی) کارگزار یادگیرنده(Learning agent) *(این تمرین مشابه تمرین ۲.۵ در ویرایش سوم است)***

☑ حل: برای این اصطلاحات تعاریف ممکن زیادی وجود دارد که برخی از آنها در اینجا آورده شده است. **الف)** کارگزار: روبات یا رم‌افزاری است که با درک و مشاهده محیط پیرامون خود، عکس‌العملی را انجام می‌دهد. به عبارت دیگر همواره در حال مشاهده انجام عکس‌العمل می‌باشد. اساسا نکته کلیدی در تعریف هر کارگزار، در نحوه پیاده‌سازی تابع کارگزار آن نهفته است یعنی نکه آن روبات قرار است در محیط چه وظیفه‌ای را برعهده گیرد. (توجه: برخی نویسندگان به جای اصطلاح تابع از «برنامه» ستفاده نموده‌اند که منظور از آن قطعه کدهایی است که بر روی ماشین‌ها و شبکه‌های مختلف قابلیت اجرا داشته و عملکرد سان را بر عهده بگیرند مانند کارگزارهای موبایل). **ب)** تابع کارگزار: تابعی است که نشان می‌دهد کارگزار در قبال هرگونه دنباله شاهدات ممکن، چه کاری را باید انجام دهد. این تابع می‌تواند به صورت‌های مختلف مانند جدول و یا متن بیان شود. به عنوان ثال برای کارگزار جاروبرقی می‌توان جدولی تشکیل داد که هر سطر آن یک حالت محیط مانند «اتاق چپ کثیف است» را شان دهد که در مقابل آن سطر کاری که کارگزار باید انجام دهد را می‌نویسیم مثلا در این مورد کارگزار باید «تمیز کردن اتاق» ا انجام دهد. کل جدول دربرگیرنده تمام حالات دنیا است تا عملکرد کارگزار را به طور کامل تعیین کرده باشیم. **ج)** برنامه کارگزار: برنامه‌ای کامپیوتری است که بر مبنای معماری کارگزار نوشته شده است و تابع کارگزاری که تاکنون به صورت جدولی ا متنی بوده است را در کامپیوتر پیاده‌سازی می‌کند. در یک طراحی ساده می‌توان گفت این برنامه، برنامه‌ای است که یک شاهده جدید رخ داده را گرفته و واکنش مناسب برای آن را برمی‌گرداند. **د)** عقلانیت: یک ویژگی کارگزار است و به معنی آن ست که کارگزار عاقلانه رفتار کند یعنی با دریافت مشاهدات(Percepts) در طی زمان، واکنشی را برای انجام انتخاب کند که به رای انجام آن، میانگین سودمندی‌اش بیشینه شود. **هـ)** خودمختاری: یک ویژگی کارگزار است و به معنی آن است که رفتار کارگزار فقط توسط تجربیات شخصی‌اش تعیین می‌شود تا آنکه صرفا توسط برنامه‌های اولیه(Initial programming) که رنامه‌نویس درون آن قرار می‌دهد. **و)** کارگزار واکنشی: کارگزاری است که واکنش آن در محیط، تنها به مشاهدات فعلی‌اش از حیط بستگی دارد. **ز)** کارگزار مبتنی بر مدل: کارگزاری است که واکنش آن مستقیما از یک مدل درونی مشتق می‌گردد که ین مدل، توصیفی از موقعیت جهان فعلی است. این مدل در طی زمان بروزرسانی می‌شود. **ح)** کارگزار مبتنی بر هدف: این کارگزار، واکنشی را از بین موارد ممکن انتخاب می‌کند که آشکارا وی را به هدف مربوطه برساند. **ط)** کارگزار مبتنی بر سودمندی: این کارگزار، واکنشی را از بین موارد موجود انتخاب می‌کند که در موفقیت‌های آتی، میانگین سودمندی وی را یشینه کند. **ی)** کارگزار یادگیرنده: کارگزاری است که در طی زمان و بر مبنای تجربیات شخصی‌اش، چیزهایی یاد گرفته و فتار خود را بهبود می‌بخشد. هر یک از موارد فوق کارگزاری متفاوت را نشان می‌دهد. مانند انسان‌ها که می‌توانند در زندگی خود انجام کارهایشان یا براساس عقل، تجربه، هدف و یا سودمندی رفتار کنند. به عنوان نمونه یک پزشک در عمل جراحی یا یک ونده در طی مسابقه برای انتخاب حرکت بعدی خود ایده‌ای متفاوت دارند. البته ممکن است یک کارگزار در هر زمان و با توجه ه موقعیت، ترکیبی از این رفتارها را از خود نشان دهد. ولی مسلم است که انتخاب کارگزار مناسب برای هر محیط و شرایط در سیدن به نتیجه بسیار تاثیرگذار است زیرا در بسیاری موارد شاید عاقلانه رفتارکردن نیاز نباشد و در برخی دیگر برعکس.

**؟ ۲.۲ معیارهای کارایی(Performance measure) و تابع سودمندی دو معیار جهت اندازه‌گیری مناسب بودن عملکرد کارگزار می‌باشند. تفاوت بین این دو معیار را شرح دهید.**

☑ حل: معیار کارایی توسط یک فرد خارجی اندازه‌گیری می‌شود که بسته به میزان موفقیت کارگزار در عمل، ارزیابی انجام ی‌شود. در واقع تابعی است که با توجه به تاریخچه ارزیابی‌های خود از واکنش‌های کارگزار، یک عدد حقیقی را به عنوان معیار ارایی کارگزار اعلام می‌کند ولی تابع سودمندی توسط خود کارگزار مورد استفاده و اندازه‌گیری قرار می‌گیرد تا وی بتواند گونگی موقعیت‌های مطلوب یا گذشته را ارزیابی کند و در انتخاب‌های بعدی خود تاثیر دهد. در مباحث ما، معیار تابع سودمندی و معیار کارایی یکسان نیستند. لازم به ذکر است یک کارگزار ممکن است هیچ تابع سودمندی مشخصی نداشته باشد، لی همواره دارای معیار کارایی است.

**؟ ۳.۲ این تمرین تفاوت بین تابع کارگزار و برنامه کارگزار را مورد بررسی قرار می‌دهد. الف) آیا برای یک تابع کارگزار داده شده، می‌توان بیش از یک برنامه کارگزار پیاده‌سازی نمود؟یا یک مثال ذکر کنید و یا آنکه نشان دهید چرا ذکر مثال ممکن ست؟ ب) آیا تابع کارگزاری وجود دارد که نتواند توسط هیچ برنامه کارگزاری پیاده‌سازی شود؟ ج) آیا یک برنامه کارگزار به**

**ازای معماری ماشین مشخص، دقیقا یک تابع کارگزار را پیاده‌سازی می‌کند؟ د) یک معماری با $n$ بیت فضا داده شده است. ممکن است چند برنامه کارگزار متفاوت در آن وجود داشته باشد؟** ***(این تمرین مشابه تمرین ۲.۶ در ویرایش سوم است)***

☑ حل: اگرچه این سوالات بسیار ساده به نظر می‌رسند ولی حاوی نتایج بنیادی فراوانی هستند. در پاسخ‌های این بخش، فرض کرده‌ایم که طراحی کارگزار ساده بوده و در محیط‌های استاتیک عمل می‌کند که این فرضیه تا زمانی که کارگزار اقدام به کنکاشی عمدی نکند، هیچ مشکلی ایجاد نمی‌کند. البته نتایج برای محیط‌های پویا جذاب‌تر بوده و حاوی مسائلی متفاوت می‌باشد. **الف)** بله. به عنوان مثال یک برنامه کارگزار را گرفته و در بخش‌هایی از آن عبارت تهی (مانند Enter و فاصله) قرار دهید به طوری که هیچ تأثیری در خروجی نگذارد. بنابراین برنامه‌های متفاوتی با همان عملکرد بدست می‌آید (تابع عملکرد یکسان و برنامه متفاوت). به عنوان مثالی دیگر فرض کنید از تعداد دانشجو بخواهیم تا برنامه‌ای به زبان C++ برای محاسبه فاکتوریل یک عدد بنویسند، مسلم است که برنامه هر دانشجو با دیگری متفاوت خواهد بود. ممکن است فردی آن را با دستور For و دیگری با دستور while و یا do-while بنویسد. همچنین نام متغیرها و تعداد خطوط هر برنامه با دیگری متفاوت است. **ب)** بله. تابع کارگزاری را در نظر بگیرید که عملکرد تست تورینگ را انجام می‌دهد و پس از مشاهده محیط و در صورت توقف، کلمه True و در غیر این صورت کلمه False را چاپ می‌کند. این تابع توسط هیچ برنامه کارگزاری قابل پیاده‌سازی نیست زیرا اگر این طور بود تست تورینگ توسط هر روباتی انجام شده و روبات‌ها به سادگی می‌توانستند با فریب انسان‌ها بر آنها غالب شوند. (توجه: در محیط‌های دینامیک و برای ماشین‌هایی با سرعت کمتر از بی‌نهایت، پیاده‌سازی تابع کارگزار عقلانی ممکن نیست. به عنوان مثال پیاده‌سازی تابع کارگزاری که همواره در بازی‌هایی مانند شطرنج حرکت برنده را انجام دهد میسر نیست). **ج)** بله. رفتار یک کارگزار توسط معماری و برنامه آن مشخص می‌باشد. **د)** حداکثر $2^n$ برنامه کارگزار وجود دارد. اگرچه بسیاری از آنها هرگز اجرا نخواهند شد. (توجه: از آنجا که در کل $n$ بیت داریم پس هر برنامه حداکثر به $n$ بیت از فضا برای ذخیره‌سازی نیازمند است. بنابراین وضعیت داخلی آن می‌تواند از بین فقط $2^n$ تاریخچه گذشته مشخص و متمایز شود. از آنجا که تابع کارگزار، نشان دهنده واکنش بر مبنای تاریخچه مشاهدات است، ممکن است توابع کارگزار بسیاری وجود داشته باشند که به خاطر فقدان حافظه در ماشین، نتوانند پیاده‌سازی شوند).

**؟ ۴.۲ می‌خواهیم عقلانیت چند توابع کارگزار مختلف برای جاروبرقی را بررسی کنیم. الف) نشان دهید که تابع کارگزار ساده‌ای که برای جارو برقی در شکل ۳.۲ توصیف شد، طبق فرضیات گفته شده در بخش ۲.۲ عقلانی است. ب) یک تابع برای کارگزار عقلانی توصیف کنید که در هر حرکت، معیار کارایی تغییر داده شده را یک واحد کم کند. آیا برنامه این کارگزار، به حالت داخلی(Internal state) نیازدارد؟ ج) در مورد طراحی‌های ممکن برای کارگزار، در حالتی که «مربع‌های تمیز می‌توانند دوباره کثیف شوند» و یا «جغرافیای محیط ناشناخته باشد»، بحث کنید. آیا این موضوع باعث می‌شود که کارگزار در این مورد از تجربیات شخصی‌اش یاد بگیرد؟ اگر این گونه باشد، چه چیز را بایستی یاد بگیرد؟** ***(این تمرین مشابه تمرین ۲.۲ در ویرایش سوم است)***

☑ حل: دقت شود که در اینجا با توجه به فرضیات ساده‌سازی محیط، هیچ نگرانی در مورد حالت‌های غیرقطعی نداریم. **الف)** برای این منظور فقط کافی است نشان دهیم که به ازای تمام حالت‌های ممکن، (مانند انواع حالات کثیفی‌ها و موقعیت‌های مختلف شروع) این کارگزار می‌تواند خانه‌ها را حداقل با سرعت سایر کارگزارهای موجود، تمیز کند زیرا در این صورت کارگزار عقلانی نامیده می‌شود. این موضوع در صورت عدم وجود کثیفی، مسلما درست است. حال برای وضعیتی که محل شروع کثیف باشد ولی هیچ خانه دیگری کثیف نباشد کارگزار با انجام یک مرحله، محیط را پاکیزه می‌کند و هیچ کارگزاری نمی‌تواند بهتر از این عمل کند. همچنین در وضعیتی که محل شروع کثیف نباشد ولی خانه دیگر کثیف باشد، پس از گذشت دو مرحله محیط پاکیزه می‌گردد و هیچ کارگزاری نمی‌تواند بهتر از این عمل کند. و در نهایت اگر هر دو خانه کثیف باشند، پس از سه مرحله، دنیا پاکیزه می‌گردد و هیچ کارگزاری نمی‌تواند بهتر از این عمل کند. **ب)** کارگزار گفته شده در قسمت الف، مدام به جلو و عقب حرکت می‌کند. حتی اگر دنیا تمیز باشد ولی بهتر است در صورت تمیزی دنیا، کارگزار هیچ عملیاتی انجام ندهد. از آنجا که ادراکات کارگزار اطلاعاتی در مورد تمیز بودن سایر خانه‌ها نمی‌دهد، به نظر می‌رسد که کارگزار بایستی دارای حافظه باشد تا بتواند از تمیزی سایر خانه‌ها نیز مطلع باشد. ایجاد دقیق این ویژگی بسیار دشوار است ولی شاید کارگزار بتواند با گذاشتن اشیاء و نمادهایی برای خود این موضوع را متوجه شود. به عنوان مثال در صورت قرار گرفتن در خانه چپ تمیز، بفهمد که خانه سمت راست نیز تمیز است؟ برای این منظور یک کارگزار بایستی از خود محیط به عنوان نوعی حافظه خارجی(External memory) کمک بگیرد. این تکنیک استفاده از اشیاء به صورت مشابه در انسان‌ها نیز وجود دارد. مثلا برای یادآوری قرار ملاقات به جای استفاده از تقویم، دستمال گردن خود را گره می‌زنیم. فرض کنید کارگزار می‌تواند واکنش‌های (تمیزکردن،A) و (تمیزکردن،B) را انجام دهد. اگر از کارگزار بخواهیم پس از تمیز کردن هر خانه هیچ عملیاتی انجام ندهد، آنگاه در موردی که سایر خانه‌ها کثیف باشند، کارگزار شکست می‌خورد و اگر از وی بخواهیم همواره فعال بوده، باز محکوم به حرکت بیهوده بین خانه‌ها خواهد شد. در حالت کلی این مشکل از آنجا نشأت می‌گیرد که کارگزارهای واکنشی بایستی در موقعیت‌هایی که شبیه به هم به نظر می‌رسند، کار یکسانی انجام دهند، حتی اگر موقعیت‌ها در واقعیت متفاوت از هم باشند. در دنیای جارو برقی(Vacuum-cleaner world) این امر مشکل بزرگی محسوب می‌شود. زیرا وقتی در خانه جاری قرار داریم در نظر کارگزار مربع دیگر هم شبیه به خانه‌ای کثیف است و هم تمیز. لذا باید به آنجا برود تا به حقیقت واقعه پی ببرد. **ج)** اگر برای جاروبرقی، طول عمری طولانی در نظر بگیریم، واضح است که یادگیری یک نقشه(Map) جهت مراجعه به آن بسیار مفید خواهد بود. زیرا می‌تواند از

ودن کارگزار به دیوار جلوگیری کند. همچنین کارگزار می‌آموزد که یک مکان کثیف با عملیات جمع‌آوری همراه بوده و ی‌تواند یک استراتژی جهت سرکشی بهینه تدبیر نماید. جزئیات مختصری از متدهای اکتشافی برای ساخت یک نقشه کامل در صل ۴ آورده شده است.

**۵.۲ برای هریک از این محیط‌های وظیفه که کارگزاران در آن فعالیت می‌کنند، مشخصات PEAS را بیان کنید: الف)روبات وتبالیست ب) کارگزار خریدار اینترنتی کتاب ج) مریخ‌نورد خودگردان د)دستیار اثبات قضایای ریاضی *(این تمرین مشابه مرین ۴.۲ در ویرایش سوم است)***

حل:

| سنسورها | محرک‌ها(ACTUATORS) | محیط | معیارکارایی | نوع کارگزار |
|---|---|---|---|---|
| وربین، سنسورهای لمسی، شتاب‌سنج، سنسورهای جهت‌یابی، رمز گذار لولا/ چرخ | وسایل(مانند پا) برای حرکت و انتقال و لگدزدن | زمین بازی، توپ، تیم خودی، تیم طرف مقابل، بدن خود | برنده‌شدن در مسابقه، اهداف بر علیه تیم مقابل | الف) کارگزار روبات فوتبالیست |
| صفحات‌وب، تقاضاهای کاربر | دنبال کردن لینک‌ها، پرکردن و ارسال داده‌ها در گزینه‌های فرم، نمایش به کاربر | اینترنت | یافتن تقاضا (کتاب مورد علاقه)، کمینه کردن هزینه | ب) کارگزار خرید اینترنتی کتاب |
| وربین، سنسورهای لمسی، شتاب‌سنج، سنسورهای جهت‌یابی، رمزگذار لولا/ چرخ، دریافت کننده رادیویی | چرخ/ پا، وسیله جمع آوری نمونه، وسایل آنالیز، فرستنده رادیویی | وسیله نقلیه، زمین، مریخ | کاوش زمین و مخابره گزارش، جمع‌آوری و آنالیز نمونه‌ها | ج) کارگزار مریخ‌نورد خودگردان |
| | | | | د)کارگزار دستیار اثبات قضایای ریاضی |

شکل ۱.۲S انواع کارگزارها و توصیفات PEAS آنها

**۶.۲ برای هر کدام از انواع کارگزارهای گفته‌شده در تمرین ۵.۲ طبق ویژگی‌های گفته‌شده در بخش ۳.۲ محیط را توصیف وده و یک طراحی مناسب برای کارگزار انتخاب نمایید.**

حل: ویژگی‌های محیط در جدول زیر آورده شده است. بهتر است که هر کارگزار را به این صورت طراحی نماییم: **الف)** معمولا یک کارگزار واکنشی مبتنی بر مدل برای بیشتر جنبه‌ها کاربرد دارد ولی برای بازی‌های تاکتیکی یک کارگزار مبتنی بر ودمندی با قابلیت پیش‌بینی می‌تواند مناسب باشد. **ب)** یک کارگزار مبتنی بر هدف برای درخواست کتاب مناسب خواهد بود ته برای انجام وظایفی مانند «یافتن کتاب مورد علاقه من برای مطالعه» نیاز است تا کارگزار بین سفارشات مختلف کمی تدبیر سبک و سنگین(Trade off) کند.

| کارگزارها (AGENTS) | گسستگی (DISCRETE) | ایستایی (STATIC) | مرحله‌ای (EPISODIC) | قطعیت (DETERMINISTIC) | مشاهده پذیری (OBSERVABLE) | محیط وظیفه TASK ) (ENVIRONMENT |
|---|---|---|---|---|---|---|
| چندکارگزاری | پیوسته | پویا | ترتیبی | اتفاقی | نیمه | روبات فوتبالیست |
| تک‌کارگزاری | گسسته | استاتیک | ترتیبی | قطعی | نیمه | خریداینترنتی کتاب |
| تک‌کارگزاری | پیوسته | پویا | ترتیبی | اتفاقی | نیمه | مریخ نوردخودگردان |
| چندکارگزاری | گسسته | استاتیک | ترتیبی | قطعی | کاملا | دستیار ریاضی |

شکل ۲.۲S مشخصات محیط

) معمولا برای وظایفی چون هدایت‌های سطح پایین و اجتناب از سد راه، یک کارگزار واکنشی مبتنی بر مدل مناسب است ی برای برنامه‌ریزی مسیریابی، برنامه‌ریزی اکتشاف، آزمایش و.... ترکیبی از کارگزارهای مبتنی بر هدف و کارگزارهای مبتنی بر ودمندی مورد استفاده قرار می‌گیرد. **د)** برای اثبات برخی قضایا، یک کارگزار مبتنی بر هدف مورد نیاز است ولی برای برخی سایای اکتشافی مانند «اثبات تعدادی موضوع مفید درباره عملیات روی رشته‌ها» ممکن است یک کارگزار مبتنی بر سودمندی ورد نیاز باشد.

***در تمرینات بعدی این فصل، پیاده‌سازی محیط‌های جاروبرقی را مورد بررسی قرار می‌دهیم.***

**۷.۲ برای دنیای جارو برقی شکل ۲.۲ که مشخصات آن در بخش ۱.۲ بیان شد، یک شبیه‌ساز محیط( Environment simulato) برای اندازه‌گیری معیار کارایی پیاده‌سازی نمایید. پیاده‌سازی شما بایستی ماژولار باشد تا حسگرها، محرک‌ها و شخصات محیطی (سایز، شکل، مکان کثیفی و...) به سهولت قابل تغییر باشند. *(این تمرین مشابه تمرین ۸.۲ در ویرایش سوم ست)***

☑ حل: می‌توانید با اندکی جستجو، کدی برای پیاده‌سازی محیط جاروبرقی به زبان لیسپ بیابید. سپس با ایجاد تغییراتی در آن نظیر شکل اتاقها و موانع، به نتیجه دلخواه خود دست یابید.

**؟ ۸.۲ یک کارگزار واکنشی‌ساده برای محیط جاروبرقی در تمرین ۷.۲ پیاده‌سازی کنید. شبیه‌ساز محیط را به ازای این کارگزار و برای تمام پیکره‌بندی‌های ممکن، یعنی تمام حالات شروع ممکن از جهت کثیفی و مکان کارگزار اجرا کنید. سپس به ازای هر پیکره‌بندی امتیاز کارایی این کارگزار را ثبت کرده و در نهایت میانگین کل امتیاز وی را محاسبه کنید. *(این تمرین مشابه تمرین ۹.۲ در ویرایش سوم است)***

☑ حل: یک برنامه کارگزار واکنشی که تابع کارگزار عقلانی را پیاده‌سازی می‌کند و در این فصل تشریح شد در زیر آورده شده است:

```
(defun reflex-rational-vacuum-agent (percept)
  (destructuring-bind (location status) percept
    (cond ((eq status 'Dirty) 'Suck)
          ((eq location 'A) 'Right)
          (t 'Left))))
```

برای وضعیت‌های 1 و 3 و 5 و 7 در شکل ۲۰.۳، معیار کارایی به ترتیب عبارتست از: 1996، 1999، 1998، 2000.

**؟ ۹.۲ نسخه تغییریافته‌ای از محیط جارو برقی تمرین ۷.۲ را در نظر بگیرید که در آن کارگزار به ازای هر حرکت، یک جریمه می‌شود. الف) آیا یک کارگزار واکنشی ساده می‌تواند در این محیط به طور کاملا عقلانی عمل کند. توضیح دهید. ب) در مورد یک کارگزار واکنشی با حالت درونی چطور؟ این کارگزار را طراحی نمایید. ج) اگر مشاهدات کارگزار به گونه‌ای باشد که وضعیت کثیفی/ تمیزی خانه‌ها در محیط را درک نماید، پاسخ‌های شما در قسمت‌های الف و ب چه تغییری می‌کند؟ *(این تمرین مشابه تمرین ۱۰.۲ در ویرایش سوم است)***

☑ حل: **الف)** خیر، به پاسخ قسمت ب در تمرین ۴.۲ مراجعه شود. **ب)** به پاسخ قسمت ب در تمرین ۴.۲ مراجعه شود. **ج)** در این مورد، یک کارگزار واکنشی ساده می‌تواند به طور کاملا عقلانی عمل نماید. این کارگزار از جدولی با 8 ورودی تشکیل شده است که به ازای هر مشاهده، عملیات ایندکس[3] را انجام داده و واکنش مربوط به آن حالت، مشخص و استخراج می‌شود. پس از انجام عکس‌العمل کارگزار جهان به روز شده و وقوع مشاهده بعدی، به کارگزار می‌گوید که چه کاری را باید انجام دهد. در محیط‌های بزرگتر، ساخت چنین جدولی غیرممکن است ولی در عوض کارگزار می‌تواند از الگوریتم‌های جستجوی بهینه فصل‌های ۳ و ۴ استفاده نماید تا بتواند اولین گام از دنباله راه حل را اجرا نماید. دراینجا نیز هیچ حالت درونی نیاز نمی‌باشد ولی اگر حالت درونی داشته باشیم می‌توانیم دنباله راه‌حل را ذخیره کنیم تا در هر مشاهده جدید، به جای محاسبه مجدد از همان استفاده کنیم.

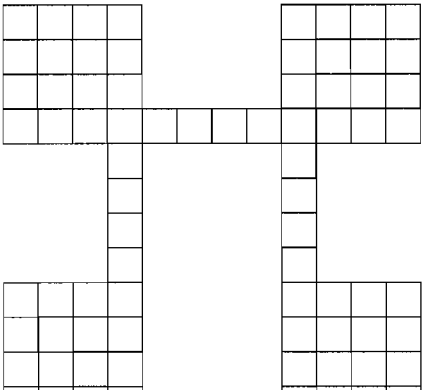
**؟ ۱۰.۲ نسخه تغییریافته‌ای از محیط جارو برقی تمرین ۷.۲ را در نظر بگیرید که در آن جغرافیای محیط (وسعت، مرز و موانع) ناشناخته باشند و همچنین در حالت شروع هیچ اطلاعی از وضعیت کثیفی نداشته باشیم. (کارگزار می‌تواند در جهات بالا، پایین، چپ و راست حرکت نماید.) الف) آیا یک کارگزار واکنشی ساده می‌تواند در این محیط به طور کاملا عقلانی عمل کند؟ توضیح دهید. ب) آیا یک کارگزار واکنشی ساده که دارای تابع کارگزاری تصادفی است، می‌تواند بهتر ازیک کارگزار واکنشی ساده عمل نماید؟ چنین کارگزاری را طراحی نموده و کارایی آنرا در محیط‌های مختلف اندازه‌گیری نمایید. ج) آیا می‌توانید محیطی طراحی کنید که کارگزار تصادفی طراحی شده توسط شما، در آن عملکرد بسیار ضعیفی داشته باشد؟ نتایج خود را بیان کنید. د) آیا یک کارگزار واکنشی با حالت درونی، بهتر از یک کارگزار واکنشی ساده عمل می‌کند؟ کارگزاری طراحی نموده و کارایی آنرا در محیط‌های مختلف اندازه‌گیری کنید. آیا می‌توانید برای چنین نوعی، کارگزار عقلانی طراحی کنید؟ *(این تمرین مشابه تمرین ۱۱.۲ در ویرایش سوم است)***

☑ حل: **الف)** به علت آنکه کارگزار هیچ اطلاعی از جغرافیا، درک مکانی و مشاهده کثیفی ندارد و همچنین نمی‌تواند اتفاقات رخ داده را به خاطر آورد، لذا هر زمان که کارگزار سعی در حرکت به یک جهت نماید و آن جهت ممکن نباشد، مدام به دیوار برخورد خواهد نمود ولی اگر به طور اتفاقی حرکت کند، این گونه نخواهد شد. **ب)** یک طراحی ممکن برای کارگزار آن است که کثیفی‌ها را تمیز کرده و در غیر این صورت به طور تصادفی حرکت نماید.

```
(defun randomized-reflex-vacuum-agent (percept)
  (destructuring-bind (location status)
    percept (cond ((eq status 'Dirty) 'Suck)
                  (t (random-element '(Left Right Up Down))))))
```

[3] عملیات ایندکس شیوه‌ای برای یافتن سطری خاص از جداول بزرگ است که خود دارای الگوریتم‌هایی می‌باشد که به ازای یک ورودی مشخص، سطر خروجی را برمی‌گرداند.

طراحی بسیار شبیه به عملکرد جاروبرقی Roomba™ است. (البته Roomba دارای حسگر ضربه نیز بود و تنها زمانی
کت تصادفی انجام می‌داد که به مانعی برخورد نماید.) در این صورت کارگزار در محیط‌های کوچک و خوب، به طور معقولی
کرد مناسب خواهد داشت ولی در محیط‌هایی شبیه Maze یا محیط‌هایی با رویدادهای مختلف، پوشش تمام خانه‌ها بسیار
بر خواهد شد. ج) مثالی از یک محیط را در شکل ۳.۲S مشاهده می‌کنید. البته دانشجویان علاقه‌مند می‌توانند در صورت
یل، زمان تمیز کردن را در محیط‌های مربعی و خطی با سایزهای متفاوت اندازه گرفته و سپس آن را با کارایی الگوریتم‌های
تجوی محلی که در فصل ۴ بیان شده است، مقایسه کنند.



شکل ۳.۲S محیطی که حرکت تصادفی در آن موجب می‌شود تا برای پوشش تمام خانه‌ها زمان زیادی صرف شود.

یک کارگزار واکنشی با حالت درونی، می‌تواند یک نقشه بسازد (برای کسب اطلاعات بیشتر به فصل ۴ مراجعه شود). یک
تجوی آنلاین اول عمق، می‌تواند در یک زمان خطی بر حسب اندازه محیط، به هر حالتی دستیابی پیدا کند، بنابراین کارگزار
واند خیلی بهتر از یک کارگزار واکنشی ساده عمل نماید. داشتن رفتاری عقلانی در محیط‌های ناشناخته، موضوعی پیچیده
ولی معمولا علاقه‌مندان زیادی برای تفکر در مورد آن وجود دارد. برای این موضوع نیاز است تا نگاهی به توزیع اولیه
مالاتی بر روی انواع محیط‌ها داشته باشیم که به آن حالت عقیده اولیه(Belief state) می‌گوییم. هر واکنشی که منجر به
جدیدی گردد و برای آپدیت این توزیع استفاده شود، باعث حرکت کارگزار به یک حالت عقیده جدید می‌شود. بدلیل
یدگی محیط، حالات عقیده در یک محیط ساده فرض می‌شوند. لذا مشکل اکتشاف بهینه، همان مسأله جستجو برای
راتژی بهینه در فضای حالت عقیده می‌باشد که با وجود لجاجت مسئله‌ای خوش‌تعریف است. به عنوان مثال برای اکتشاف
ه می‌توان به بازی کامپیوتری Minesweeper اشاره نمود. (تمرین ۱۱.۷ را ببینید) در محیط‌های کوچک این بازی،
شاف بهینه ممکن است عملی باشد ولی مرحله آپدیت حالات عقیده نیاز به شرح فراوان دارد.

**۱۱.۱ تمرین ۱۰.۲ را برای وضعیتی تکرار کنید که حسگر موقعیت را با یک حسگر ضربه جایگزین کنیم که این حسگر برخورد
گزار به مانع و یا عبور وی از مرزهای محیط را تشخیص می‌دهد. حال فرض کنید این حسگر ضربه از کار بیافتد. کارگزار چه
کند؟** ***(این تمرین مشابه تمرین ۱۳.۲ در ویرایش سوم است)***

حل: این مسئله در ابتدا بسیار شبیه به حالت قبلی خواهد بود و تفاوت اصلی آن در این است که به جای درک دقیق
یت مکانی در ساخت نقشه، کارگزار بایستی موقعیت خود را حدس بزند. (منظور همان گره‌ها در ساختمان داده معرف گراف
ی حالت هستند.) زمانی که یک ضربه تشخیص داده شود، کارگزار با فرض عدم تغییر این وضعیت، در نقشه خود یک دیوار
ت می‌کند. در محیط‌های خط‌کشی شده(Grid environment) کارگزار می‌تواند موقعیت $(x,y)$ خود را پیگیری نموده و
این قادر است رسیدن به یک حالت قدیمی را تشخیص دهد. ولی در حالت کلی تشخیص اینکه یک حالت جدید است یا
می، چندان ساده نیست.

**۱۲.۱ محیط‌های جاروبرقی بیان شده در تمرین‌های قبل، همگی قطعی بودند. اکنون به ازای هر یک از این نسخه‌های اتفاقی،
نه‌های کارگزار ممکن برای آنها را مورد بحث قرار دهید: الف) قانون مورفی(Murphy`s low): بیست وپنج درصد مواقع،
یات مکیدن منجر می‌شود که در صورت کثیفی کف اتاق تمیز نشود و یا یک کف اتاق تمیز، با خالی کردن آشغال‌ها کثیف
.اگر سنسور کثیفی در 10٪ مواقع پاسخ اشتباهی دهد، چه تأثیری روی برنامه کارگزار شما خواهد گذاشت؟ ب) بچه‌های
ک: در هر مرحله زمانی، هر خانه تمیز 10٪ شانس کثیف شدن دارد.آیا می‌توانید برای این مورد یک کارگزار عقلانی طراحی
ید؟** ***(این تمرین مشابه تمرین ۱۳.۲ در ویرایش سوم است)***

☑ حل: **الف)** برای یک کارگزار واکنشی این موضوع هیچ مشکل جدیدی ایجاد نمی‌کند زیرا این کارگزار مادامی که مکان فعلی کثیف باشد به عمل مکیدن آشغال‌ها ادامه می‌دهد. ولی برای کارگزاری که از دنباله مشاهدات خود یک نقشه می‌سازد، بایستی به جای عمل «مکیدن»، عمل «مکیدن تا زمان تمیز شدن» را جایگزین کنیم. اگر حسگر کثیفی در هر مرحله امکان اشتباه داشته باشد، آنگاه کارگزار بایستی قبل از آنکه تصمیم بگیرد عمل مکیدن یا حرکت به خانه بعد را انجام دهد، برای چندین مرحله منتظر بماند تا عملکردی قابل قبول‌تر داشته باشد. پس پر واضح است که بایستی شرایط را سبک و سنگین کنیم زیرا اگر زمان خیلی زیادی منتظر بمانیم، کثیفی‌ها نیز به مدت طولانی در کف اتاق باقی می‌ماند. (که کاری مستحق مجازات است) و از طرف دیگر واکنش فوری نیز دارای ریسک است زیرا موجب کثیف کردن یک اتاق تمیز یا نادیده گرفتن یک اتاق کثیف خواهد شد (به خاطر عملکرد اشتباه سنسور). بنابراین کار عاقلانه آنست که کارگزاری عقلانی مدام بین خانه‌ها حرکت و سرکشی کرده و کثیفی آنها را چک کند حتی اگر قبلا آن خانه بازدید شده باشد. (زیرا ممکن است حسگر در آن زمان جواب اشتباهی داده باشد.) البته نمی‌توان برای هر سرکشی میزان زمان توقف هر خانه را به طور دقیق تعیین نمود ولی با توجه به تجربه به طور ریاضی قابل‌حدس است. این مسئله بخشی از فرآیندهای تصمیم‌گیری مارکوف(Markov decision process) را شامل می‌شود برخی مسائل در حالت کلی دشوارند. ولی ممکن است در برخی موارد خاص از آنها بتوان به آنالیزهای دقیقی دست یافت. **ب)** در این مورد کارگزار بایستی به طور نامحدود بین خانه‌ها سرکشی کند. احتمال آنکه یک مربع که قبلا تمیز بوده، اکنون کثیف باشد در طی گذشت زمان افزایش می‌یابد بنابراین راهکار عقلانی آنست که به طور مکرر کوتاهترین سرکشی ممکن به تمام خانه‌ها را انجام دهیم (که به نوعی تکرار مکررات است زیرا ممکن است عواقبی چون سرکشی چند باره به یک خانه را در پی داشته باشد. البته به جغرافیای محیط نیز بستگی دارد). این مسئله نیز بخشی از فرآیندهای تصمیم‌گیری مارکوف محسوب می‌شود.

# فصل ۳ (ویرایش دوم)

**۱.۳ این اصطلاحات را به زبان خود تعریف کنید: الف) حالت ب) فضای حالت ج) درخت جستجو د) گره جستجو هـ) هدف و) واکنش ز) تابع پسین ح) ضریب انشعاب *(این تمرین مشابه تمرین ۱.۳ در ویرایش سوم است)***

حل: **الف)** یک حالت موقعیتی از محیط تعریف می‌شود که کارگزار می‌تواند در آن قرار بگیرد و خود دارای دو نوع می‌باشد: حالت‌های جهان (موقعیت‌های حقیقی در جهان واقعی) و حالت‌های نمایشی (توصیفی اجمالی از جهان واقع که توسط کارگزار در زمان انجام کارهای مورد توانش، مورد استفاده قرار می‌گیرد). البته هر حالت را می‌توان به صورت‌های مختلف ترسیمی، عددی و یا متنی بیان نمود. برای مثال نشان دادن موقعیت وزیرها در هر حالت از چیدمان $n$- وزیر را می‌توان به صورت شکلی از یک صفحه و مهره‌ها، رشته‌ای 8 عددی نشان‌دهنده سطر هر وزیر و یا یک پاراگراف متنی که در آن موقعیت هر وزیر تشریح شده است، بیان نمود. هر یک از این حالات، موقعیت فعلی محیط را برای کارگزار تشریح می‌کند. **ب)** فضای حالت گرافی است که گره‌های آن همان حالت‌ها بوده و لینک‌های موجود در گراف، واکنش‌هایی هستند که عامل انتقال از یک حالت به حالت دیگر هستند. با استفاده از این گراف یک کارگزار می‌فهمد که هر واکنش چه تاثیری بر روی محیط داشته و به چه وضعیتی در محیط منجر می‌شود. **ج)** درخت جستجو درختی است (درخت یک گراف بدون دور است) که در آن گرهٔ ریشه، همان حالت اولیه در محیط است و مجموعه فرزندان هر گره در این درخت، همان مجموعه حالت‌هایی هستند که با انجام یک واکنش در محیط قابل دسترسی‌اند. **د)** به هر گره در درخت جستجو، گرهٔ جستجو گفته می‌شود. **هـ)** هدف، حالتی از محیط است که کارگزار برای رسیدن به آن تلاش می‌کند. بنابراین معمولا تعدادی از گره‌های درخت جستجو با رنگی دیگر یا شکلی متفاوت علامت‌گذاری شوند که حالات پایانی را نشان دهند. **و)** واکنش عبارتست از هرکاری که کارگزار می‌تواند انجام دهد و اقدام کند که منجر به انتقال کارگزار از یک حالت به حالت دیگر می‌شود. (یال‌های درخت جستجو) **ز)** یک تابع پسین را می‌توان این گونه تعریف نمود: یک حالت محیط را می‌گیرد و سپس مجموعه‌ای از زوج مرتب‌های (حالت، واکنش) تولید می‌کند که نشان می‌دهد به ازای هر حالت از محیط، کارگزار چه واکنشی را بایستی انجام دهد. (تابع کارگزار) **ح)** ضریب انشعاب در یک درخت جستجو تعریف می‌شود و به معنای تعداد واکنش‌های ممکنی است که کارگزار در هر حالت می‌تواند انتخاب کرده و انجام دهد. [۴]

**۲.۳ توضیح دهید چرا تدوین مسأله بایستی پس از تدوین هدف صورت پذیرد؟ *(این تمرین مشابه تمرین ۱.۳ در ویرایش سوم است)***

حل: در عملیات تدوین هدف، تعیین می‌کنیم کارگزار علاقه‌مند به رسیدن به کدام حالت از جهان می‌باشد، هدف او در بیستم چیست، از کدام حالت‌ها بایستی صرف نظر کرده و از کدام یک دوری بجوید. ولی در عملیات تدوین مسأله، تعیین می‌کنیم که کدام بخش‌های محیط مهم بوده و بایستی لحاظ شوند و کدام جنبه‌ها را نادیده بگیریم[۵]. اگر ما عملیات تدوین مسأله را در ابتدا و قبل از تدوین هدف انجام دهیم، دقیقا نمی‌دانیم که چه بخش‌هایی برای هر حالت باید در نظر گرفته شود و چه بخش‌هایی حذف شود و این امر موجب شود تا در زمان یافتن راه‌حل مدام بین عملیات تدوین هدف، تدوین مسأله و حل مسأله سرگردان باشیم.

**۳.۳ فرض کنید تابع LEGAL-ACTION(~) نشان‌دهندهٔ مجموعه واکنش‌های مجاز در حالت s، و تابع RESULT(*a*,s) نشان‌دهندهٔ نتیجهٔ حاصل از اجرای واکنش مجاز a در حالت s باشد. اکنون تابع پسینی با عنوان SUCCESSOR-FN بر حسب LEGAL–ACTION و RESULT تعریف کنید (و بر عکس).**

حل: این برنامه در زبان پایتون عبارتست از:

```
#### successor_fn defined in terms of result and legal_actions def successor_fn(s):
```

---

در مساله مسافرت از شهر آراد تا شهر بخارست بر روی نقشه رومانی، منظور ازحالت، بودن در هر شهر است، فضای حالت کل ساختار نقشه و جاده‌های ارتباطی آنهاست که به صورت گراف ترسیم می‌شود، درخت جستجو درختی است که گره ریشه آن شهر آراد بوده و بسته به تعداد جاده متصل به آن چندین یال از آن خارج می‌شود و به سطح بعدی درخت می‌رسیم. مجددا از این سطح جاده‌های متصل به هر یک را رسم کرده و ادامه می‌دهیم. هر جا که شهر بخارست مشاهده شد آن را با دایره‌ای مشخص می‌کنیم تا هدف تعیین گردد، به هر یک از این شهرهای موجود در درخت، گره جستجو و به شهر بخارست، گره هدف می‌گوییم. واکنش همان اقدامی است که کارگزار انجام می‌دهد یعنی پیمودن جاده که یال‌های درخت جستجو می‌باشند و تابع پسین تابعی است که به طور ریاضی یا شهودی بتواند حالت بعدی یک حالت را به ازای واکنش‌های ممکن تعیین کند که در اینجا می‌توان جدولی به این منظور تشکیل داد. ضریب انشعاب تعداد یال خروجی از هر گره درخت است یعنی چند جاده به هر شهر متصل است.

به عبارت دیگر در تدوین مساله 8- وزیر، چگونگی نمایش قرارگیری تمام وزیرها در صفحه شطرنج در حالت‌های مختلف می‌تواند به شکل‌های گوناگونی انجام شود مثلا به طور جدولی، رشته‌ای از اعداد و یا نمادین که این تنها بخش تعریف حالت در تدوین مساله است و تعریف حالت اولیه، تابع پسین، آزمون هدف و هزینه نیز بایستی صورت پذیرد ولی تدوین هدف یعنی با توجه به این ساختار تعریف شده، شکل هدف که کارگزار باید بدنبال آن باشد چگونه است. مشخص است که قبل از تعیین شکل هدف باید شکل کلی حالات، تعیین شده باشد.

```
    return [(a, result(a, s)) for a in legal_actions(s)]
#### legal_actions and result defined in terms of successor_fn def legal_actions(s):
return [a for (a, s) in successor_fn(s)]
def result(a, s):
    for (a1, s1) in successor_fn(s):
        if a== a1:
            return s1
```

**؟ ۴.۳ نشان دهید که چیدمان‌های مختلف پازل 8- تایی، می‌توانند به دو مجموعه مجزا تقسیم شوند به طوری که نتوان هیچ حالتی از مجموعهٔ اول را با تعدادی حرکت، به حالتی از مجموعهٔ دوم تبدیل نمود. (راهنمایی: به Berlekamp et *al.* (سال 1982) مراجعه کنید). قطعه کدی پیشنهاد دهید که بگوید هر حالت داده شده در این پازل (هر چیدمان ممکن) در کدام یک از این دو مجموعه قرار دارد؟ توضیح دهید چرا این روش برای تولید حالات تصادفی مناسب است؟ *(این تمرین مشابه تمرین ۴.۳ در ویرایش سوم است).***

☑ حل: در آدرس http://www.cut-the-knot.com/pythagoras/fifteen.shtml می‌توانید حل این موضوع را برای مسأله پازل 15- تایی مشاهده کنید که می‌توان بخش‌هایی از آن را برای پازل 8- تایی نیز استفاده نمود: تعریف: حالت هدف را به صورت «خانه‌های مرتب شده» تعریف می‌کنیم و جهت بررسی هر حالت بایستی از گوشهٔ سمت چپ سطر اول شروع به حرکت نموده و با حرکت از چپ به راست، خانه‌ها را بررسی کنیم تا به مرتب بودن آنها یقین حاصل نماییم. با رسیدن به انتهای هر سطر، به سمت چپ‌ترین خانه در سطر بعدی جهش می‌کنیم. این کار تا پایان خانه‌ها ادامه پیدا می‌کند. در طی این پیمایش، اگر دو خانه یافتیم که عدد خانهٔ دوم بیشتر از خانهٔ قبلی‌اش باشد، آنگاه آن دو خانه بایستی تعویض شوند تا به هدف برسیم. پیشنهاد: پازل تعریف شده در بند فوق را در نظر گرفته و فرض کنید N نشان‌دهندهٔ مجموع تعداد کل تعویض‌ها به علاوه شماره سطر خانهٔ خالی باشد. بنابراین باقیماندهٔ N بر 2 مستقل از هر حرکت خواهد بود. یعنی اگر برای یک چیدمان N فرد باشد، پس از انجام یک حرکت همچنان فرد باقی می‌ماند و اگر زوج باشد، پس از انجام یک حرکت زوج باقی می‌ماند. (افزایش و کاهش دو واحد به عددی فرد، عددی فرد تولید می‌کند و همین طور برای عدد زوج). با توجه به این موضوع، حالت هدف شکل ۴.۳ را در نظر بگیرید[6]. در این شکل، تمام خانه‌ها به ترتیب مرتب شده و به هیچ تعویضی نیاز نیست و خانهٔ خالی در سمت چپ سطر اول قرار دارد، که طبق پیشنهاد گفته شده، مقدار N برابر یک خواهد شد، بنابراین برای رسیدن به این حالت هدف که مقدار N آن فرد است، سیستم باید از حالتی فرد شروع به کار نماید که در ابتدا Nای فرد برگ داشته و به تدریج با کسر دو واحد در هر مرحله به این حالت هدف فرد دست یابد. شروع از حالتی زوج، نمی‌تواند کارگزار را به این حالت هدف فرد برساند. بنابراین دو نوع چیدمان پازل داریم، چیدمان‌هایی که دارای N فرد هستند و چیدمان‌هایی که دارای N زوج هستند. **اثبات:** می‌دانیم که لغزاندن یک خانه در جهت افقی، هیچ تغییری در تعداد کل تعویض‌ها و شماره سطر خانه خالی ایجاد نمی‌کند، بنابراین در این اثبات فقط لغزش عمودی یک خانه را در نظر می‌گیریم. فرض کنید خانهٔ A دقیقا بالای خانه خالی واقع شده باشد. اگر حرکتی انجام شود که فقط بر روی خانه‌های B و C و D اثرگذار باشد و این خانه‌های B و C و D مستقل از حرکت A باشند (همگی بزرگتر از A باشند)، آنگاه با لغزش عمودی این خانه، سه واحد به مجموع تعویضات اضافه می‌شود که سه عددی فرد است. اکنون وضعیت دیگری را در نظر بگیرید که یکی از این سه خانه، کوچکتر از A باشد بنابراین بایستی قبل از حرکت و لغزش ابتدا آن خانه کوچکتر را با A تعویض کنیم و سپس حرکت را انجام دهیم. اکنون مجموع تعویضات مورد نیاز برای این خانه‌ها 2 می‌باشد. (با کسر 1 تعویض از عدد 3 به عدد 2 رسیدیم). البته این دو فرضیه هر دو نتایج یکسانی را دربردارند. لذا می‌توان به این نتیجه رسید که تغییر در مجموع N، همواره زوج است و این همان چیزی است که قصد اثبات آن را داشتیم (اعمال تغییرات زوج بر روی هر عدد، تغییری در زوج و فرد بودن آن ایجاد نمی‌کند). بنابراین قبل از حل یک پازل، بایستی مقدار N را برای حالت شروع و حالت هدف مد نظر محاسبه کنیم و مطمئن شویم که از نظر زوج و فرد بودن با هم تطبیق داشته باشند وگرنه هیچ راه‌حلی برای آن مساله وجود نخواهد داشت.

**؟ ۵.۳ مسأله $n$- وزیر با راه‌حل تدوین افزایشی موثر که در این فصل بیان شد، را در نظر بگیرید. توضیح دهید که چرا حداقل اندازهٔ فضای حالت، $\sqrt[3]{n!}$ است. همچنین بزرگترین $n$ ممکن برای حل کامل این مسئله را بیابید. (راهنمایی: با قرار دادن هر**

---

[6] قسمتی از شکل ۴.۳ که حالت هدفی با N=1 را نشان می‌دهد.

| | | |
|---|---|---|
| | 1 | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

**وزیر در ستونی مجزا و شمردن حداکثر تعداد خانه‌هایی که در تهدید آن هستند، یک حد پایینی برای فاکتور انشعاب تعریف نمایید). *(این تمرین مشابه تمرین ۳.۵ در ویرایش سوم است)***

☑ حل: در این تدوین، هر وزیر را در ستونی مجزا قرار می‌دهیم و وزیر جدید بایستی در خانه‌ای قرار بگیرد که توسط هیچ وزیر دیگری تهدید نشود. به منظور ساده‌سازی، به جای مسأله 8- وزیر، مسأله 8- رخ را در صفحهٔ شطرنج در نظر بگیرید. اولین رخ می‌تواند در هر خانه از ستون اول قرار بگیرد. دومین رخ می‌تواند در هر خانه‌ای از ستون دوم بجز سطر اول قرار بگیرد و. . . . در حالت کلی اندازهٔ فضای حالت در این وضعیت $n!$ می‌گردد. لازم به ذکر است که در مسألهٔ $n$- وزیر، هر وزیر سه خانه از ستون‌های مابعد را تهدید می‌کند (یک خانه در همان سطری که این وزیر قرار دارد، یک خانه مربوط به حرکت مورب وزیر به بالا و خانه دیگر مربوط به حرکت مورب به سمت پایین می‌باشد. پس به ازای هر وزیر در ستون بعد، سه خانه تهدید شده داریم). بنابراین به علت وجود وزیر اول در ستون اول، ستون دوم دارای حداقل $(n-3)$ خانه امن باشد که بایستی در انتخاب جایگاه وزیر این ستون لحاظ شود. به همین ترتیب در ستون سوم، به علت آنکه دو وزیر قبل از آن قرار دارند، $(n-6)$ انتخاب امن داریم و الی آخر. بنابراین اندازهٔ فضای حالت عبارت است از $S \geq n(n-3)(n-6)\ldots$ پس داریم:

$$\begin{aligned} S^3 &\geq n \cdot n \cdot n \cdot (n-3) \cdot (n-3) \cdot (n-3) \cdot (n-6) \cdot (n-6) \cdot (n-6) \cdot \cdots \\ &\geq n \cdot (n-1) \cdot (n-2) \cdot (n-3) \cdot (n-4) \cdot (n-5) \cdot (n-6) \cdot (n-7) \cdot (n-8) \cdot \cdots \\ &= n! \end{aligned}$$

یا به طور خلاصه:

$$S \geq \sqrt[3]{n!}$$

**؟ ۶.۳ آیا همواره برای یک فضای حالت متناهی، درخت جستجو نیز متناهی است؟ برای یک فضای حالت متناهی که یک درخت نیز باشد چطور؟ آیا می‌توانید دقیقا توضیح دهید که چه انواعی از فضای حالت، همواره منجر به درخت جستجوی متناهی می‌گردند؟ (اقتباس شده از Bender,1996).**

☑ حل: خیر. این گونه نیست که یک فضای حالت متناهی همواره منجر به ایجاد یک درخت جستجوی متناهی گردد. فضای حالتی با دو حالت را در نظر بگیرید که هر دوی آنها به ازای یک واکنش به دیگری می‌رسند. درخت جستجو برای این فضای حالت، نامتناهی است، زیرا می‌توانیم به دفعات، بین آنها حرکت کنیم (مثلا نقشه بین دو شهر را در نظر بگیرید که با دو جاده مختلف به یکدیگر می‌رسند. در این صورت یک نقشه متناهی داریم ولی می‌توانیم درختی رسم کنیم که ریشه آن شهر اول باشد، که با یک جاده به شهر دیگر برود، دوباره در سطح بعدی درخت به شهر اول برسد و این کار تا بی‌نهایت ادامه داشته باشد). ولی اگر فضای حالت به جای گراف به شکل درختی متناهی باشد یا در حالت کلی به شکل یک DAG محدود( Directed Acyclic Graph) ، انگاه به علت عدم وجود حلقه، درخت جستجو، متناهی خواهد شد.

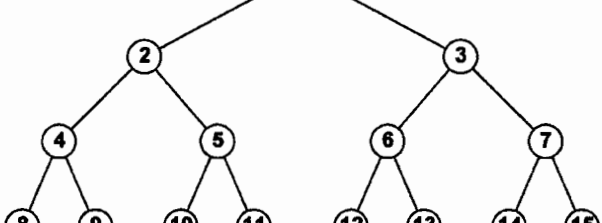
**؟ ۷.۳ برای هر یک از این موارد، حالت شروع، آزمون هدف، تابع پسین و تابع هزینه را ذکر کنید. همچنین تدوینی دقیق بیابید که قابل پیاده‌سازی باشد: الف) قصد رنگ‌آمیزی نقشه ای مسطح با 4 رنگ را دارید به طوری که هیچ دو منطقهٔ مجاوری، هم‌رنگ نباشند. ب) میمونی با قد 3 فوت در اتاقی به ارتفاع 8 فوت قرار دارد که تعدادی موز از سقف آن آویخته شده است و این میمون می‌خواهد این موزها را بگیرد. در این اتاق تعدادی جعبه به ارتفاع 3 فوت قرار دارد که می‌توان انها را حرکت داده و روی یکدیگر قرار داد و همچنین می‌توان از آنها بالا رفت. ج) برنامه‌ای دارید که با بررسی فایلی از رکوردهای ثبت شده، پیغام «رکورد ثبت شده غیر مجاز است» را در خروجی چاپ می‌کند. می‌دانیم که پردازش هر رکورد ثبت شده مستقل از سایر رکوردهاست و شما می‌خواهید غیرمجاز بودن یک رکورد را تشخیص دهید. د) شما سه کوزه با حجم‌های 12 گالن، 8 گالن، 3 گالن و نیز یک شیر آب در اختیار داشته و می‌توانید هر کوزه را از آب پر کرده و یا آب آن را بر روی زمین و یا کوزه‌های دیگر خالی کنید. شما می‌خواهید دقیقا یک گالن آب در یکی از کوزه‌ها داشته باشید. *(این تمرین مشابه تمرین ۳.۶ در ویرایش سوم است)***

☑ حل: **الف)** حالت شروع: هیچ منطقه‌ای رنگ‌آمیزی نشده است. آزمون هدف: تمام مناطق رنگ‌آمیزی شده باشند و هیچ دو منطقهٔ مجاوری همرنگ نباشد. تابع پسین: انتساب یک رنگ به یک منطقه. تابع هزینه: تعداد انتساب‌ها. **ب)** حالت شروع: حالتی که در متن ذکر شد. آزمون هدف: میمون، موز را گرفته باشد. تابع پسین: هل دادن یک جعبه از یک محل به محلی دیگر، راه رفتن از یک محل به محلی دیگر، گرفتن موز (اگر روی جعبه ایستاده باشد). تابع هزینه: تعداد واکنش‌های انجام شده. **ج)** حالت شروع: همهٔ رکوردهای ورودی در نظر گرفته شوند. آزمون هدف: چاپ پیغام «رکورد ثبت شده غیر مجاز است» به ازای یک رکورد مشخص. تابع پسین: بر روی نیمهٔ ابتدایی مجموعه رکوردها، عملیات اجرا را تکرار کن، بر روی نیمهٔ دوم مجموعه رکوردها، عملیات اجرا را تکرار کن. تابع هزینه: تعداد دفعات اجرا. توجه: این مسأله جزء دسته مسائل محتمل الوقوع قرا می‌گیرد زیرا برای انتخاب عملیات بعدی، بایستی اول بدانیم که در اجرای قبلی پیغام خطایی تولید می‌گردد یا خیر. **د)** حالت شروع: سه کوزه با مقادیر $[0,0,0]$. تابع پسین: به ازای $[x,y,z]$، تولید حالت‌های، $[12,y,z]$ و $[x,y,3]$ و $[x,8,z]$ با پر کردن هر کوزه، تولید حالت‌های $[x,0,z]$ و $[0,y,z]$ و $[x,y,0]$ با خالی کردن هر کوزه به ازای دو کوزهٔ $x,y$، آب درون $y$ را درون کوزهٔ $x$ بریزیم که موجب می‌شود کوزهٔ $x$ دارای حداقل $x+y$ یا $y$ آب باشد، کاهش حجم کوزهٔ اول توسط کوزهٔ $y$. تابع هزینه: تعداد دفعات انجام.

**؟ ۸.۳ فضای حالتی را در نظر بگیرید که در حالت شروع آن عدد 1 داریم و تابع پسین به ازای هر حالت $n$، دو حالت با اعداد $2n+1$ و $2n$ تولید می‌کند. الف) بخشی از فضای حالت برای حالت‌های 1 تا 15 را ترسیم کنید. ب) فرض کنید عدد 11 حالت**

**هدف باشد، ترتیب گره‌های ملاقات شده توسط روش‌های جستجوی اول سطح، جستجوی عمقی محدود شده با محدودیت 3، و جستجوی عمیق شونده تکراری را لیست کنید. ج) آیا برای این مسأله، جستجوی دوطرفه مناسب است؟ اگر این گونه است جزئیات چگونگی عملکرد آن را توضیح دهید. د) ضریب انشعاب در هر جهت از جستجوی دو طرفه چقدر است؟ هـ) آیا پاسخ ج یک تدوین مجدد از مسأله را نشان می‌دهد که به شما اجازه دهد راه‌حلی بدون هیچ جستجو از حالت 1 به هدف بیابید؟ *(این تمرین مشابه تمرین ۱۵.۳ در ویرایش سوم است)***

☑ حل: **الف)** به شکل ۱.۳s مراجعه شود.




شکل ۱.۳s فضای حالت

**ب)** اول‌سطح: 1 2 3 4 5 6 7 8 9 10 11 عمقی‌محدود: 1 2 4 8 9 5 10 11 عمیق‌شوندهٔ تکراری: 1, 1 2 3, 1 2 4 5 3 1, 1 2 4 8 9 5 10 11 6 7, 1 2 4 8 9 5 10 11 **ج)** جستجوی دوطرفه بسیار مفید است، زیرا تنها پسین حالت n در جهت معکوس، حالت $\lfloor (n/2) \rfloor$ است، که این موضوع به جستجو کمک می‌کند. **د)** در جهت پیش‌رو مقدار 2 دارد و در جهت معکوس مقدار 1. **هـ)** بله از حالت هدف شروع می‌کنیم و واکنش پسین معکوس را تا جایی ادامه می‌دهیم که به حالت 1 دست پیدا کنیم.

**؟ ۹.۳ مسأله‌ای تحت عنوان کشیش‌ها و آدم‌خوارها به این صورت تعریف می‌شود. سه کشیش و سه آدم‌خوار در یک طرف رودخانه قرار دارند که تنها یک قایق با ظرفیت یک یا دو نفر موجود است. می‌خواهیم تمام این افراد را به طرف دیگر رودخانه منتقل کنیم ولی در هیچ جا نباید تعداد کشیش‌ها از تعداد آدم‌خوارها کمتر باشد وگرنه خورده می‌شوند (چه در قایق و چه در هر طرف رودخانه). این مسأله در هوش‌مصنوعی بسیار مشهور است زیرا عنوان اولین مقاله‌ای بود که از دید تحلیلی و فلسفی به موضوع تدوین مسأله پرداخت (Amarel,1968) الف) این مسأله را به دقت تدوین کنید به طوری که فقط موارد مورد نیاز جهت اطمینان از یک راه‌حل معتبر را ذکر کنید. نموداری از فضای حالت کامل ترسیم کنید. ب) با استفاده از یک الگوریتم جستجو، این مسأله را به طور بهینه حل و پیاده‌سازی نمایید. آیا برای بررسی حالت‌های تکراری ایده‌ای مناسب وجود دارد؟ ج) با وجود سادگی فضای حالت، چرا مردم برای حل این معما با مشکل مواجه هستند؟ *(این تمرین مشابه تمرین ۹.۳ در ویرایش سوم است)***

☑ حل: **الف)** یکی از تدوین‌های ممکن برای حالت مساله چنین است که هر حالت را به صورت مجموعه‌ای 6 تایی از اعداد صحیح تعریف می‌کنیم که به ازای هر یک از کشیش‌ها و آدم‌خوارها یک عدد قرار داده باشیم (به عنوان مثال می‌توان برای هر فرد مقداری دودویی تعریف کرد به این صورت که اگر فرد در طرف اول رودخانه قرار دارد مقدار مربوط به آن یک باشد وگرنه صفر. مثلا مجموعه (1,1,0,1,0,1) نشان‌دهنده حالتی است که یک کشیش و یک آدمخوار در طرف دیگر رودخانه بوده و دو کشیش و دو آدمخوار در این طرف هستند. با فرض اینکه سه عدد اول مربوط به کشیش‌ها و سه عدد دوم مربوط به آدمخوارها باشد. طبق این تعریف در زمان شروع در حالت (1,1,1,1,1,1) و در هدف در حالت (0,0,0,0,0,0) هستیم.) همچنین می‌دانیم که قایق در ابتدا در این طرف رودخانه و سپس در طرف دیگر می‌باشد. حالت هدف حالتی است که در آن سه کشیش و سه آدم‌خوار در طرف دیگر رودخانه باشند و تابع هزینه به ازای هر عمل انجام شده محاسبه می‌شود. همچنین پسین یک حالت شامل تمام حالت‌هایی است که در آن 1 یا 2 نفر بر روی قایقی سوار شده و از یک سو به سوی دیگر حرکت می‌کنند. **ب)** فضای جستجو بسیار کوچک است. بنابراین هر الگوریتمی می‌تواند به طور بهینه در آن عمل کند. دقت داشته باشید که بایستی حرکت‌هایی که منجر به بازگشت به حالت‌های قبلی مشاهده شده می‌شوند را حذف کنید. **ج)** زیرا بسیاری از حرکت‌ها یا غیر مجازند (خورده شدن کشیش‌ها توسط آدم‌خوارها) و یا آنکه محیط را به یکی از وضعیتهای قبلی بازمی‌گرداند و دقیقا تشخیص آنها ممکن نیست و این به خاطر بزرگی فاکتور انشعاب در این مسأله است که هیچ راهی برای پیشروی در آن وجود ندارد.

**؟ ۱۰.۳ برای مسألهٔ پازل 8- تایی دو نوع تابع پسین پیاده سازی کنید: یکی از این توابع، همه پسین‌ها را همزمان تولید می‌کند برای این کار ساختمان داده پازل 8- تایی را کپی و ویرایش می‌کند. ولی تابع دیگر در هر دفعه که فراخوانی شود فقط یک پسین جدید را تولید کرده و برای این کار به طور مستقیم، حالت پدر را تغییر می‌دهد (البته در صورت عدم نیاز به تغییرات، هیچ**

**کاری انجام نمی‌دهد). اکنون نسخه‌هایی از جستجوی اول عمق عمیق شوندهٔ تکراری بنویسید که از این دو تابع پسین استفاده کنند. سپس کارایی آنها را مقایسه نمایید.**

☑ حل: دو تابع گفته شده برای مسأله پازل 8- تایی، تفاوت زیادی از جنبهٔ کارایی ندارند. معمولا هر حالت پازل 8-تایی را به صورت یک عدد صحیح 32 بیتی تعریف و پیاده‌سازی می‌کنیم. بنابراین چه تغییر در داده و چه کپی‌برداری از روی آن، هیچ تفاوتی در این حالت ندارند. ولی برای یک پازل $n \times n$ که عدد n مقدار بزرگی باشد، تغییر داده قبلی بسیار کاراتر از کپی مجدد از روی آن است. البته تابع پسینی که با تغییر دادهٔ قبلی کار می‌کند نیز معایبی دارد و آن این است که فقط با روش جستجوی اول‌سطح (یا اندکی تغییر در این الگوریتم مانند عمیق شوندهٔ تکراری) کار می‌کند.

**؟ ۱۱.۳ در بخش ۴.۳ در مورد الگوریتم جستجوی «طولانی‌کنندهٔ‌تکراری» بحث کردیم که نوعی تکراری از همان روش جستجوی هزینه یکنواخت است. به علاوه در هزینهٔ مسیر آن محدودیت افزایشی اضافه شد. اگر گره‌ای تولید شود که هزینه مسیر آن از محدودیت فعلی بیشتر باشد، فورا حذف می‌شود. در هر دفعه تکرار جدید این الگوریتم، محدودیت مورد نظر به اندازهٔ کمترین هزینهٔ مسیر گره‌هایی تعیین می‌شود که در تکرار قبلی حذف شده‌اند. الف) نشان دهید این الگوریتم برای حالت کلی هزینهٔ مسیر، بهینه است. ب) درختی یکنواخت با فاکتور انشعاب $b$، عمق راه‌حل $d$ و هزینه گام واحد را در نظر بگیرید. در این حالت الگوریتم طولانی‌کنندهٔ تکراری به چند تکرار نیازمند است؟ ج) اکنون فرض کنید هزینهٔ گام به طور پیوسته در بازهٔ $[\epsilon 0]$ بوده و حداقل هزینه، ثابت $\epsilon$ باشد، در بدترین حالت چه تعداد تکرار الگوریتم موردنیاز است؟ د) این الگوریتم را پیاده‌سازی کرده و سپس آن را در مسائل پازل 8- تایی و فروشندهٔ دوره‌گرد بکار ببرید. آنگاه کارایی حاصل از این روش را با کارایی جستجو با هزینهٔ‌یکنواخت مقایسه کرده و در مورد نتایج آن توضیح دهید.** ***(این تمرین مشابه تمرین ۱۷.۳ در ویرایش سوم است)***

☑ حل: **الف)** در این الگوریتم عملیات بسط گره‌ها به ترتیب افزایش هزینه مسیر صورت می‌گیرد بنابراین اولین هدف یافته شده همان هدف با کمترین هزینه خواهد بود. **ب)** این الگوریتم در این حالت مشابه روش عمیق‌شوندهٔ تکراری عمل خواهد کرد. یعنی به $d$ تکرار نیاز دارد که در طی آنها، $O(b^d)$ گره تولید می‌گردد. **ج)** $d / \epsilon$

**؟ ۱۲.۳ ثابت کنید که روش‌های جستجو با هزینه یکنواخت و جستجوی اول سطح با هزینهٔ ثابت، در زمانی که از الگوریتم جستجوی Graph-Search استفاده می‌کنند، بهینه خواهند بود. سپس یک فضای حالت با هزینه گام متغیر بیابید که در آن الگوریتم عمیق‌شوندهٔ تکراری با جستجوی Graph-Search می‌تواند یک راه‌حل نیمه بهینه پیدا کند.**

☑ حل: می‌دانیم که اگر از گرهٔ شروع تا یک گرهٔ خاص، دو مسیر موجود باشد، آنگاه کنار گذاشتن مسیر پرهزینه تر هیچ تأثیری در راه‌حل بهینه نمی‌گذارد. همچنین روش‌های جستجوی با هزینه یکنواخت و جستجوی اول سطح با هزینه گام ثابت، هر دو مسیرهایی هم هزینه را بسط می‌دهند. بنابراین اگر به حالتی رسیدیم که گرهٔ جاری قبلا بسط داده شده بود، مسلما مسیر جاری به آن گره پر هزینه‌تر از مسیر قبلی خواهد بود. پس بهتر است این مسیر کنار گذاشته شود تا به راه‌حل بهینه برسیم. در مورد روش عمیق‌شوندهٔ تکراری به این مثال ساده توجه کنید: فرض کنید دو مسیر به یک هدف داشته باشیم یکی از آنها با یک گام به هزینهٔ 3 به هدف می‌رسد و دیگری در طی دو گام که هر یک هزینهٔ 1 دارند به هدف می‌رسد. بنابراین توانستیم مثالی با هزینه گام متغیر ارائه دهیم که راه‌حلی نیمه‌بهینه را می‌یابد.

**؟ ۱۳.۳ فضای حالتی بیابید که در آن جستجوی عمیق‌شوندهٔ تکراری، بدتر از روش جستجوی اول‌عمق عمل نماید (به عنوان مثال با پیچیدگی $O(n^2)$ به جای $O(n)$).** ***(این تمرین مشابه تمرین ۱۸.۳ در ویرایش سوم است)***

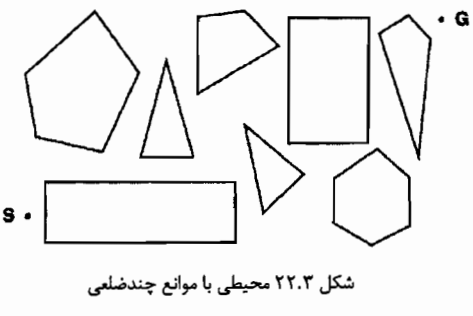
☑ حل: فضایی را در نظر بگیرید که در آن هر حالت یک پسین داشته و تنها یک هدف در عمق $n$ موجود باشد. جستجوی اول‌سطح می‌تواند این هدف را در طی $n$ گام بیابید ولی روش عمیق‌شوندهٔ تکراری بایستی تعداد $1+2+3+\ldots+n = O(n^2)$ گام را سپری کند تا به این هدف برسد.

**؟ ۱۴.۳ برنامه‌ای بنویسید که به عنوان ورودی، URL دو وب سایت را گرفته و سپس مسیری از لینک‌های یکی از این سایت‌ها به دیگری را پیدا می‌کند. چه استراتژی جستجویی مناسبی پیشنهاد می‌دهید؟ آیا روش جستجوی دوطرفه، ایده‌ای مناسب است؟ آیا می‌توان از یک موتور جستجو جهت دانستن صفحه قبلی یک لینک کمک گرفت؟** ***(این تمرین مشابه تمرین ۱۹.۳ در ویرایش سوم است)***

☑ حل: یک آدم معمولی (یا حتی کارگزار) تنها زمانی می‌تواند بفهمد صفحهٔ بعد یک لینک چیست که به آن مراجعه کند. پس بهتر است از روش‌هایی مانند اول‌سطح یا اول بهترین‌حریصانه استفاده نمود که بر حسب تعداد لغات رایج بین صفحهٔ ابتدایی و صفحهٔ هدف، عملیات جستجو را انجام دهند که شاید در رسیدن لینک‌ها به صفحه هدف کارساز باشد. در زمینهٔ استفاده از موتورهای جستجو، می‌دانیم که آنها گراف کاملی از وب را در اختیار دارند به طوری که می‌توانند یک کاربر را به تمام صفحات یا بخشی از صفحاتی که به یک صفحه لینک شده‌اند، هدایت کنند پس مشکل انسان‌ها را نداشته و استفاده از جستجوی دو طرفه در آنها ایدهٔ مناسبی است.

**؟ ۱۵.۳ مسألهٔ یافتن کوتاهترین مسیر بین دو نقطه در صفحه‌ای با موانع چند ضلعی محدب که در شکل ۲۲.۳ ترسیم شده است را در نظر بگیرید. این مسأله به نوعی حالت ایده‌آلی از مسأله یافتن مسیر در یک محیط شلوغ توسط روبات است. الف) فرض کنید فضای حالت را به صورت مجموعه تمام موقعیت‌های $(x,y)$ در صفحه در نظر بگیریم چه تعداد حالت در این فضا وجود دارد؟ چند مسیر برای رسیدن به هدف وجود دارد؟ ب) به طور مختصر توضیح دهید که چرا کوتاهترین مسیر از رأس یک چندضلعی**

رأس دیگری، بایستی شامل تعدادی قطعه خطوط مستقیمی باشد که به رئوس چند ضلعی‌ها ختم می‌شود. در این حالت یک سای حالت مطلوب را تعریف کنید. اندازهٔ این فضای حالت جدید چقدر است؟




شکل ۲۲.۳ محیطی با موانع چندضلعی

) توابع مورد نیاز برای پیاده‌سازی مسأله جستجویی را تعریف کنید که در آن یک تابع پسین وجود دارد که یک رأس را به نوان ورودی گرفته و مجموعه رئوسی که در یک خط مستقیم با آن رأس هستند را در خروجی بر می‌گرداند. (همسایگان همان ندضلعی که رأس ورودی بر رویش واقع شده است، را فراموش نکنید). برای تابع آرویتی از فاصلهٔ خط مستقیم کمک بگیرید. یک یا چند مورد از الگوریتم‌های گفته شده در این فصل را برای حل مسائلی این چنینی به کار ببرید و نظر خود را در مورد رایی آنها بیان کنید. *(این تمرین مشابه تمرین ۳.۷ در ویرایش سوم است)*

حل: **الف)** اگر تمام نقاط $(x,y)$ در صفحه به عنوان فضای حالت در نظر گرفته شوند، آنگاه تعداد حالات و تعداد مسیرها نهایت خواهد بود. **ب)** (در این مسأله نقطهٔ شروع و نقطهٔ هدف هر دو از رئوس می‌باشند.) می‌دانیم که کوتاهترین فاصلهٔ بین نقطه خط مستقیمی است که از آن دو می‌گذرد ولی در این مسأله پیمودن مسیری مستقیم به علت وجود موانعی بر سر راه سیر نیست بنابراین کوتاهترین مسیر ممکن در این وضعیت، از تعدادی قطعه خط تشکیل شده است که متوالیا به هم وصل ده و شکل کلی آنها اندکی با خط مستقیم اصلی تفاوت دارد ولی تا حد امکان به آن شبیه است. در این مسیر اولین قطعه خط محل رأس ابتدایی شروع شده و به نقطه‌ای بر روی یک مانع ختم می‌شود (هر مسیر دیگری که تماس بیشتری با مانع داشته ند، مسلما طولانی‌تر خواهد بود). از آن جا که موانع چندضلعی‌اند، نقاط تماس بایستی در رئوس موانع باشند، بنابراین کل سیر بایستی رأس به رأس ادامه یابد. اکنون با توجه به شرح فوق، می‌توان فضای حالت را به صورت مجموعه‌ای از مختصات وس موانع بیان نمود. برای شکل ۲۲.۳ تعداد رئوس موانع 35 می‌باشد. **ج)** تابع پسین حالات مجاز را تولید کرده و از روی کنش، نتیجه مربوطه را بدست می‌آورد. این موضوع هم شامل حالات درونی و هم حالات هدف میگردد. اگر در گره node قرار شته باشیم، برای رئوسی که در طی یک (واکنش، نتیجه) با هم برخورد دارند، در تابع پسین بنویسید:

یک گره جدید (گره شروع)←S

گره هدف←G

هزینه مسیر[node]+ تعداد هزینه‌های و←هزینه مسیر

| | اول سطح | هزینه‌یکنواخت | اول عمق | عمق محدود | عمیق‌شونده تکراری | طرفه |
|---|---|---|---|---|---|---|
| کامل بودن | بله [a] | بله [a,b] | خیر | خیر | بله [a] | [a,b] |
| پیچیدگی زمانی | $O(b^{d+1})$ | $O(b^{1+\lceil C^*/\varepsilon \rceil})$ | $O(b^m)$ | $O(b^l)$ | $O(b^d)$ | $O(b^d$ |
| پیچیدگی فضا | $O(b^{d+1})$ | $O(b^{1+\lceil C^*/\varepsilon \rceil})$ | $O(b^m)$ | $O(b^l)$ | $O(b^d)$ | $O(b^d$ |
| بهینگی | بله [c] | بله | خیر | خیر | بله [c] | c/2 |

۱۶.۳ مسأله هدایت روبات گفته شده در تمرین ۱۵.۳ را در این محیط در نظر بگیرید: • کارگزار می‌تواند مکان رئوسی که از قعیت جاری‌اش دیده می‌شود را مشاهده کند ولی نمی‌تواند هیچ درکی از مکان خود داشته باشد. البته می‌تواند موقعیت خود با توجه به نقشه پیموده شده محاسبه نماید. در این حالت فرض کنید هر مکان در این محیط، چشم‌اندازی متفاوت دارد. • هر حل شامل برداری است که در آن مشخصات یک خط مستقیم طبق این فرضیات بیان شده باشد: اگر در مسیر هیچ مانعی شد، آنگاه راه‌حل یک خط مستقیم بوده و واکنش سریعا انجام می‌شود. در غیر این صورت کارگزار در اولین نقطه ای از مسیر به مانع برخورد می‌کند متوقف می‌شود. همچنین اگر کارگزار به عنوان راه‌حل، برداری تهی را برگرداند یعنی در نقطهٔ هدف ار دارد (که مشخصات این نقطه را می‌دانیم)، در این صورت محیط بایستی کارگزار را به نقطه‌ای تصادفی (به جز نقاط درون انع چندضلعی) پرتاب کند. • معیارکارایی کارگزار به این طریق سنجیده می‌شود که به ازای هر واحدی از مسیر که طی شود، امتیاز و در صورت رسیدن به هدف 1000 امتیاز تعلق می‌گیرد. الف) این محیط و کارگزار حل مسأله آن را پیاده‌سازی کنید.

**کارگزار نیاز دارد تا پس از هر پرتاب، مجددا وضعیت جدید خود را تدوین و فرموله کند تا بتواند موقعیت فعلی‌اش را شناسایی نماید. ب) کارایی کارگزار خود را (با توجه به تفسیرات مناسبی که کارگزار در طی حرکت خود به دست می‌آورد)، ثبت کرده و سپس کارایی آن را در طی 100 مرحله، گزارش دهید. ج) محیط را به گونه‌ای تغییر دهید که کارگزار 30 درصد مواقع در نقطه‌ای که آن را اشتباها مقصد تصور می‌کند، متوقف می‌شود (که این نقطه به طور تصادفی از بین رئوس پیش‌روی کارگزار انتخاب می‌گردد و در صورتی که این مقصد خیالی، یافته نشود هیچ کاری انجام نمی‌شود). این مدل، تقریبا حرکت روبات واقعی را در محیط نشان می‌دهد که برخی نقاط را اشتباها مقصد تشخیص می‌دهد. اکنون کارگزار را به نحوی تغییر دهید که هر زمان که متوجه اشتباه خود شد، بتواند به شیوه‌ای به محل تصمیم اشتباه بازگشته و نقشهٔ خود را طبق همان نقشهٔ قبلی، اصلاح کند. به خاطر داشته باشید که در برخی مواقع، بازگشت مجدد به نقطه‌ای قبلی باز هم منجر به شکست می‌شود! مثالی از موفقیت کارگزار در طی دو حرکت متوالی و خطادار ذکر کنید که باز هم به هدف برسد. د) اکنون دو روش بازیابی متفاوت پس از هر کشف خطا را در نظر بگیرید: ۱- به نزدیک‌ترین رأس در مسیر اصلی مراجعه کنید. ۲- از مکان جدید، مجددا یک مسیر به سوی هدف برنامه‌ریزی کنید. اکنون کارایی این 3 روش مختلف رفع خطا را مقایسه کنید. آیا افزودن هزینهٔ جستجو بر این مقایسه تأثیرگذار است؟ هـ) اکنون فرض کنید در محیط مکان‌هایی با چشم‌انداز یکسان وجود دارد. (مثلا فرض کنید جهان شطرنجی شده و موانع مربعی‌اند). اکنون کارگزار با چه نوع مسأله‌ای مواجه است؟ راه‌حل آن به چه صورت است؟**

☑ حل: فرض کنید محیط قطعی و کاملا مشاهده‌پذیر بوده و کارگزار اثر تمام کارهایش را بر روی محیط می‌داند. بنابراین کارگزار دقیقا می‌داند که دنباله واکنش‌های او باعث می‌شود در چه حالتی از محیط قرار بگیرد و در هر لحظه در کدام حالت محیط است. هر فرض دیگری در مورد دنیا باعث می‌شود تا با مساله‌ای مجزا و متفاوت روبرو شویم. انواع مختلف مسائل: **۱.** مسائل بدون حسگر: اگر کارگزار هیچ سنسور و حسگری نداشته باشد، ممکن است در لحظه شروع در هر یک از حالات ممکن قرار داشته باشد. **۲.** مسائل دارای بازی: اگر محیط نیمه‌مشاهده‌پذیر باشد و یا آنکه اثر یک واکنش نامشخص باشد، آنگاه هر مشاهده کارگزار اطلاعات جدیدی را به وی می‌دهد. **۳.** مسائل شناسایی: اگر حالات و واکنشهای یک محیط ناشناخته باشند، کارگزار باید به گونه‌ای رفتار کند که آنها را کشف نماید. **ب)** کارایی کارگزار توسط حالات باورش تعیین می‌گردد. اگر s حالت فیزیکی داشته باشیم آنگاه $2^s$ حالات عقیده داریم. به ازای هر واحد مسافت پیموده شده از ۱۰۰ واحد، یک امتیاز به معیار کارایی کارگزار افزوده می‌گردد. **ج)** 30 درصد مواقع کارگزار در مقصدی کار خود را خاتمه می‌دهد که مقصد اشتباهی است. این کارگزار واکنش خود را بر مبنای مشاهده فعلی‌اش از محیط انتخاب می‌کند. فرض کنید این کارگزار در غیاب حسگر، آشغال‌ها را بر روی زمین مشاهده می‌کند و سپس یکی از واکنشهای «رفتن به راست»، «رفتن به چپ» و «تمیز کردن» را انتخاب کند. اگر کارگزار با خطا مواجه شود، حسگرهایش این موضوع را متوجه شده و نقشه‌ای تدوین می‌کنند تا کارگزار برگشته و کار صحیح را انجام دهد.

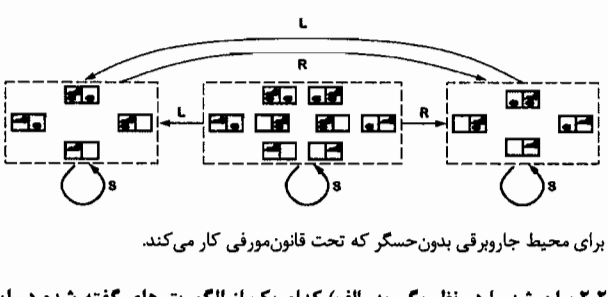
**؟ ۱۷.۳ در فصل ۳ گفته شد که نمی‌توان مسیری با هزینهٔ منفی داشته باشیم. حال در این تمرین این موضوع را با تعمق بیشتر بررسی می‌کنیم. الف) فرض کنید هر واکنش هزینهٔ منفی به اندازه‌ای دلخواه داشته باشد. توضیح دهید که چرا در این حالت یک الگوریتم بهینه باید کل فضای حالت را جستجو کند؟ ب) اگر تأکید کنیم که هزینهٔ مسیر بایستی بزرگتر و مساوی مقداری منفی و ثابت باشد، آیا کمکی می‌کند؟ هم درخت و هم گراف را در نظر بگیرید. ج) فرض کنید مجموعه‌ای از عملگرها تشکیل یک حلقه را بدهند که اجرای این مجموعه با ترتیبی خاص، هیچ تغییری در حالت سیستم نمی‌گذارد. اگر همه این عملگرها هزینهٔ منفی داشته باشند این محیط برای کارگزاری که قصد انجام رفتاری بهینه دارد چه چیزی را ایجاب می‌کند؟ د) می‌توان عملگرهایی با هزینه منفی زیاد را برای مسیریابی نیز تصور نمود. به عنوان مثال برخی پیچ‌های جاده به قدری مناظر زیبا دارند که حاضرید برای آن زمان و سوخت بیشتری را صرف کنید ولی تا ابد آنجا نمی‌مانید. با عبارات دقیق توضیح دهید که چرا انسان‌ها در هنگام جستجوی فضای حالت با یافتن حلقه‌ای مناسب (مانند پیچی زیبا در جاده) تا بی‌نهایت دور آن نمی‌چرخند؟ سپس توضیح دهید که چگونه فضای حالت و عملگر مسیریابی را تعریف کنیم تا کارگزار مصنوعی نیز از دورزدن بیهوده دور حلقه‌های مناسب اجتناب کند؟ هـ) آیا می‌توانید مسأله‌ای واقعی مثال بزنید که در آن هزینه مراحل موجب ایجاد حلقه شده باشد؟ *(این تمرین مشابه تمرین ۳ .۸ در ویرایش سوم است)***

☑ حل: **الف)** هر مسیر ممکن است در طی پیشروی خود، ناگهان با هزینهٔ منفی بزرگی روبرو شده و تمام هزینه‌های قبلی را از بین ببرد. پس حتی یک مسیر که در ابتدا بد و پرهزینه به نظر می‌رسد ممکن است در کل، مسیری بهینه باشد. بنابراین نمی‌توان در هر مرحله بین هزینه‌های پیش‌رو، یکی را انتخاب کرده و ادامه داد و حتما بایستی تمام مسیرهای ممکن جستجو شوند تا بتوان بهترین مسیر که هزینه کلی آن کمینه است را پیدا نمود. **ب)** فرض کنید بزرگترین هزینهٔ ممکن به ازای یک واکنش را برابر ثابت c در نظر بگیریم. حال اگر بیشترین عمق فضای حالت را بدانیم (مانند زمانی که فضای حالت یک درخت است) و آن را با d نمایش دهیم، آنگاه در هر مسیری با این عمق، حداکثر cd هزینه صرف خواهد شد و هر مسیری با هزینهٔ بیشتر از cd، بهینه نبوده و می‌تواند هرس شود. ولی اگر در فضای حالت حلقه داشته باشیم (گراف فضای حالت)، فرضیهٔ هزینهٔ ثابت هیچ کمکی به ما نمی‌کند، زیرا ممکن است برای چندین دفعه حلقه را دور بزنیم که در هر بار هزینه c به مجموع هزینه‌ها اضافه می‌شود. **ج)** کارگزار باید در هر بار این حلقه را دور بزند (با اینکه می‌تواند حلقه‌ای دیگر با هزینه‌ای مناسب‌تر پیدا کند). **د)** بایستی مقدار مناسب بودن یک حلقه در هر دفعه‌ای که بازدید شود، کاهش پیدا کند بنابراین یک چشم‌انداز عالی که جذابیت زیادی دارد، پس از 10 بار دیده شدن در یک ساعت خسته‌کننده بوده و دیگر هیچ جاذبه‌ای ندارد. برای پیاده‌سازی این

موضوع بایستی در فضای حالت یک حافظه گنجانده شود. در این صورت یک حالت فقط موقعیت جاری را نشان نمی‌دهد بلکه علاوه بر موقعیت جاری، دفعاتی که یک مکان ملاقات شده است را نیز در بردارد. جذابیت (پاداش) یک مکان جدید بر حسب تابعی نزولی به دست می‌آید که از تعداد دفعات دیده شدن آن مکان استفاده می‌کند. هـ) رفتاری تکراری و حلقوی مانند خوردن غذای حاضری ارزان و رفتن به سر کلاس.

**؟ ۱۸.۳ دنیای جاروبرقی را تحت شرایطی در نظر بگیرید که محیط دارای دو اتاق بوده، کارگزار هیچ سنسوری ندارد و قانون مورفی برقرار است. اکنون فضای حالت باوری ترسیم کنید که از هر یک از حالت‌های آغازین مانند {1,2,3,4,5,6,7,8} قابل دسترسی باشد و توضیح دهید که چرا این مسأله غیرقابل‌حل است. همچنین نشان دهید که اگر دنیا کاملا مشاهده‌پذیر باشد، آنگاه به ازای هر حالت شروع ممکن، به جای یک راه‌حل، دنباله ای از راه‌حل‌ها وجود دارد[7]. *(این تمرین تاحدودی مشابه تمرین ۱۰.۴ در ویرایش سوم است)***

☑ حل: فضای حالت باور را در شکل ۲.۳s مشاهده می‌کنید. ولی هیچ راه‌حلی برای آن ممکن نیست زیرا هیچ مسیری وجود ندارد که به حالت باوری ختم شود که تمام عناصر آن شرط هدف را داشته باشد (یک حالت باور در صورتی هدف تلقی می‌شود که تمام حالات فیزیکی درون آن، هدف باشند). اگر مسأله کاملا مشاهده‌پذیر باشد، کارگزار برای رسیدن به یک حالت هدف کافی است دنباله‌ای را اجرا کند. به عنوان مثال عملیات مکیدن آشغال فقط در یک اتاق کثیف قابل‌اجراست. این موضوع باعث می‌شود تا کارگزار رفتاری قطعی داشته باشد و به ازای هر حالت شروع، مسأله قابل‌حل باشد.




شکل ۲.۳s فضای حالت‌باور برای محیط جاروبرقی بدون‌حسگر که تحت قانون‌مورفی کار می‌کند.

**؟ ۱۹.۳ مسأله محیط جاروبرقی که در شکل ۲.۲ بیان شد را در نظر بگیرید. الف) کدام یک از الگوریتم‌های گفته شده در این فصل می‌تواند برای این مسأله مناسب باشد؟ آیا این الگوریتم حالت‌های تکراری را بررسی می‌کند؟ ب) با استفاده از الگوریتم انتخابی خود، دنباله بهینه‌ای از واکنش‌ها را بیابید، به طوری که محیط دارای ابعاد 3×3 بوده و در حالت شروع، سه مربع بالایی کثیف است و کارگزار در مرکز این محیط قرار دارد. ج) یک کارگزار جستجو برای محیط جاروبرقی را طراحی کنید و کارایی آن را در مجموعه‌ای از دنیاهای 3×3 که هر خانه با احتمال 0.2 کثیف باشد، ارزیابی کنید. هزینهٔ جستجو و هزینهٔ مسیر را با استفاده از یک ضریب معقول، در معیار کارایی خود دخیل کنید. د) بهترین کارگزار جستجوی خود را با یک کارگزار واکنشی تصادفی ساده در حالتی مقایسه کنید که کارگزار در صورت وجود آشغال در یک خانه آن را می‌مکد وگرنه به طور تصادفی حرکت می‌کند. هـ) اتفاقات حاصل از گسترش دنیا به ابعاد $n \times n$ را در نظر بگیرید. معیار کارایی کارگزار جستجو و کارگزار واکنشی چگونه بر حسب $n$ تغییر می‌کند؟ *(این تمرین مشابه تمرین ۳.۲۰ در ویرایش سوم است)***

☑ حل: واضح است که برای این موضوع بایستی از روش جستجوی گراف استفاده شود. این موضوع یک دنیای شطرنجی کلاسیک است که مسیرهای ثانویه بسیاری به هر حالت آن وجود دارد. دانشجویان شاید بتوانند برای این محیط دنباله راه‌حل بهینه را به سرعت بیابند ولی برای دنیای بزرگ $n \times n$، کاری دشوار و پرهزینه خواهد بود زیرا برای یک دنیا با ابعاد $n \times n$، فضای حالت دارای $n^2 \times 2^n$ حالت خواهد بود، با افزایش $n$ در این مسأله، زمان تکمیل برای یک کارگزار تصادفی رشدی کمتر از نمایی بر حسب $n$ خواهد داشت. همچنین تاکنون ضریب معقولی که برای هزینهٔ جستجو و هزینهٔ مسیر در کارگزار تصادفی استفاده شده است، موفقیت‌آمیز بوده است.

---

[7] حالت باور، معمولا در مسائلی تعریف می‌شود که کارگزار فاقد حسگر است و یا در محیط‌هایی قرار داشته باشد که کاملا مشاهده پذیر نیست. بنابراین کارگزار دقیقا نمی‌داند که در چه مکانی است ولی از نتیجه اعمال خود در محیط آگاه است. حالتی که واقعا کارگزار در آن قرار دارد (مثلا جاروبرقی که در خانه راست است و سمت چپ آن تمیز است) حالت فیزیکی نامیده شده و حالتی که کارگزار حدس می‌زند در آن شرایط قرار داشته باشد، حالت باور نامیده می‌شود (مثلا جاروبرقی که در خانه راست است و گمان میکند خانه سمت چپ یا تمیز است و یا کثیف). معمولا حالات باور درون خود مجموعه‌ای از حالات فیزیکی را در بردارند.

# فصل ۴ (ویرایش دوم)

**؟ ۱.۴ روش جستجوی A* را در مسألهٔ رسیدن به بخارست از شهر لوجیو (Lugio) اعمال کنید و تابع اکتشافی را همان «فاصله خط مستقیم» در نظر بگیرید. همچنین دنباله گره‌هایی که در این روش مورد استفاده قرار می‌گیرند و نیز مقادیر $g$ و $f$ و امتیاز هر گره را نشان دهید.** ***(این تمرین مشابه تمرین ۳.۲۳ در ویرایش سوم است)***

☑ حل: دنبالهٔ گره‌ها در صف به صورت زیر است:

L[0+244=244]
M[70+241=311], T[111+329=440]
L[140+244=384], D[145+242=387], T[111+329=440]
D[145+242=387], T[111+329=440], M[210+241=451], T[251+329=580]
C[265+160=425], T[111+329=440], M[210+241=451], M[220+241=461], T[251+329=580]
T[111+329=440], M[210+241=451], M[220+241=461], P[403+100=503], T[251+329=580], R[411+193=604], D[385+242=627]
M[210+241=451], M[220+241=461], L[222+244=466], P[403+100=503], T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627]
M[220+241=461], L[222+244=466], P[403+100=503], L[280+244=524], D[285+242=527], T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627]
L[222+244=466], P[403+100=503], L[280+244=524], D[285+242=527], L[290+244=534], D[295+242=537], T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627]
P[403+100=503], L[280+244=524], D[285+242=527], M[292+241=533], L[290+244=534], D[295+242=537], T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627], T[333+329=662]
B[504+0=504], L[280+244=524], D[285+242=527], M[292+241=533], L[290+244=534], D[295+242=537], T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627], T[333+329=662], R[500+193=693], C[541+160=701]

**؟ ۲.۴ الگوریتم اکتشاف مسیر (1997,pohl)، نوعی از جستجوی اول بهترین است که تابع‌ارزیابی آن به صورت:**

$$f(n)=(2-w)g(n)+wh(n)$$

**تعریف می‌گردد. به ازای چه مقادیری از $w$، این روش بهینگی را تضمین می‌کند؟ می‌توانید فرض کنید که $h$ قابل‌قبول باشد. به ازای $w=0$ و $w=1$ و $w=2$، این روش چه نوع جستجویی انجام می‌دهد؟** ***(این تمرین مشابه تمرین ۳.۲۵ در ویرایش سوم است)***

☑ حل: هرگاه $0\le w<2$ ، این روش کامل است. و به ازای $w=0$ داریم: $f(n)=2g(n)$. این عبارت نشان می‌دهد که در این حالت، این روش همان عملکرد جستجو با هزینهٔیکنواخت را دارد. فاکتور 2 هیچ تفاوتی در ترتیب گره‌ها ایجاد نمی‌کند و اگر $w=1$ باشد این روش همان جستجوی $A^*$ می‌شود و $w=2$ منجر می‌شود که $f(n)=2h(n)$ که جستجوی اول‌بهترین حریصانه است. همچنین داریم :

$$f(n) = (2-w)[g(n) + \frac{w}{2-w}h(n)]$$

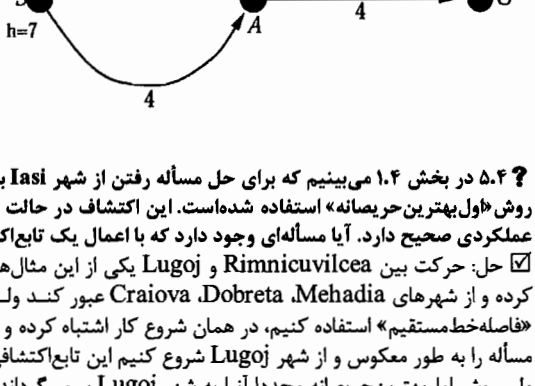
که همان عملکرد $A^*$ با تابع اکتشافی $h(n)$ ($w/(2-w)$) باشد. اگر $w\le 1$ باشد این تابع همواره کمتر از $h(n)$ است. بنابراین قابل‌قبول می‌باشد و طبق صورت سؤال می‌دانیم که خود مقدار $h(n)$ نیز قابل‌قبول است.

**؟ ۳.۴ هریک از این عبارات را اثبات کنید. الف) جستجوی اول‌سطح، نوعی خاص از جستجو با هزینهٔیکنواخت است. ب) جستجوهای اول سطح، اول عمق و جستجو با هزینه یکنواخت نوعی خاص از جستجوی اول‌بهترین در درخت هستند. ج) جستجو با هزینهٔیکنواخت، نوعی خاص از جستجوی $A^*$ است.** ***(این تمرین مشابه تمرین ۳.۲۱ در ویرایش سوم است)***

☑ حل: **الف)** اگر هزینه تمام مراحل برابر باشند $g(n) \propto depth(n)$ آنگاه جستجوی با هزینه‌یکنواخت عملکردی مشابه اول‌سطح خواهد داشت. **ب)** جستجوی اول سطح با در نظر گرفتن $f(n)=depth(n)$ همان جستجوی اول‌بهترین می‌باشد. جستجوی اول‌عمق با در نظر گرفتن $f(n)=-depth(n)$ همان جستجوی اول‌بهترین می‌باشد و در نهایت جستجو با هزینه یکنواخت با فرض $f(n)=g(n)$ همان اول‌بهترین است. **ج)** با در نظر گرفتن $h(n)=0$، جستجوی با هزینهٔیکنواخت همان $A^*$ می‌باشد.

**؟ ۴.۴ فضای حالتی مثالی بزنید که در آن روش $A^*$ با جستجوی Graph-Search بتواند یک راه‌حل نیمه‌بهینه بیابد به طوری که تابع اکتشافی $h(n)$ در آن، قابل‌قبول و ناسازگار باشد.** ***(این تمرین مشابه تمرین ۳.۲۴ در ویرایش سوم است)***

☑ حل: شکل ۱.۴s را ببینید.




شکل ۱.۴s گرافی با تابع اکتشافی ناسازگار که GRAPH-SEARCH نمی‌تواند در آن راه‌حل بهینه را بیابد. پسین S با درنظرگرفتن $f$=5، گره A بوده و با $f$=7 گره B خواهد شد. گره A در ابتدا گسترش می‌یابد بنابراین مسیر B کنارگذاشته می‌شود زیرا A در لیست قرار دارد.

**؟ ۵.۴ در بخش ۱.۴ می‌بینیم که برای حل مسأله رفتن از شهر Iasi به Fagaras، از تابع‌اکتشافی «فاصله‌خط‌مستقیم» و روش «اول‌بهترین‌حریصانه» استفاده شده‌است. این اکتشاف در حالت معکوس این مسأله یعنی رفتن از Fagaras به Iasi نیز عملکردی صحیح دارد. آیا مسأله‌ای وجود دارد که با اعمال یک تابع‌اکتشافی در دو جهت آن، عملکردی اشتباه مشاهده نماییم.**

☑ حل: حرکت بین Rimnicuvilcea و Lugoj یکی از این مثال‌هاست. کوتاهترین مسیر بایستی به طرف جنوب حرکت کرده و از شهرهای Mehadia، Dobreta، Craiova عبور کند ولی اگر از روش «اول‌بهترین‌حریصانه» و تابع‌اکتشافی «فاصله‌خط‌مستقیم» استفاده کنیم، در همان شروع کار اشتباه کرده و از شهر Rimnicuvilcea به شهر Sibiu می‌رود. اگر مسأله را به طور معکوس و از شهر Lugoj شروع کنیم این تابع‌اکتشافی ما را به طور صحیح به شهر Mehadia هدایت می‌کند. ولی روش اول‌بهترین‌حریصانه مجددا آنرا به شهر Lugoj بر می‌گرداند. بنابراین به طور تکراری بین این دو شهر سرگردان خواهیم ماند.

**؟ ۶.۴ یک تابع اکتشافی برای مسألهٔ پازل 8- تایی پیشنهاد دهید که گاهی بیشتر از حد تخمین می‌زند. نشان دهید که این تابع چگونه می‌تواند در یک مسأله خاص، راه‌حلی نیمه‌بهینه پیدا کند. (در صورت تمایل می‌توانید از کامپیوتر کمک بگیرید). اثبات کنید که اگر تابع $h$، هزینه را هیچگاه بیش از $c$ تخمین نزند، آنگاه روش A* با تابع اکتشافی می‌تواند راه‌حلی بیابد که در آن هزینهٔ راه‌حل بهینه بیشتر از $c$ نیست. *(این تمرین مشابه تمرین ۳.۲۸ در ویرایش سوم است)***

☑ حل: تابع اکتشافی $h=h_1+h_2$ (مجموع تعداد خانه‌هایی که در جای خود نیستند و فاصلهٔ منتهن)، در گاهی اوقات بیش از مقدار واقعی تخمین می‌زند. اکنون فرض کنید $h(n)\leq h^*(n)+c$ داده شده و همچنین می‌دانیم $G_2$ یک هدف نیمه‌بهینه با هزینه‌ای بیشتر از $c$ است. به عنوان مثال $g(G_2)>C^*+c$. اکنون هر گرهٔ $n$ که در مسیری به یک هدف بهینه وجود دارد را در نظر بگیرید. داریم:

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &\leq g(n)+h^*(n)+c \\ &\leq C^*+c \\ &\leq g(G_2) \end{aligned}$$

بنابراین $G_2$ هرگز قبل از بسط هدف بهینه، گسترش نخواهد یافت.

**؟ ۷.۴ ثابت کنید که اگر یک تابع اکتشافی، سازگار است، حتما باید قابل‌قبول باشد. یک تابع اکتشافی قابل‌قبول مثال بزنید که سازگار نباشد. *(این تمرین مشابه تمرین ۳.۲۹ در ویرایش سوم است)***

☑ حل: یک تابع اکتشافی سازگار است اگر به ازای هر گرهٔ $n$ که به ازای واکنش $a$ دارای پسین $n'$ است، داشته باشیم:

$$h(n)\leq c(n,a,n')+h(n')$$

برای یک اثبات ساده، تعداد $k$ گره از کوتاهترین مسیر به هدف که از گرهٔ $n$ شروع شود را در نظر بگیرید. اگر $k$=1 و $n'$ گرهٔ هدف باشد آنگاه:

$$h(n) \leq c(n,a,n')$$

به عنوان استنتاج فرض کنید که $n'$ روی کوتاهترین مسیر $k$ مرحله‌ای از هدف قرار دارد و $h(n')$ قابل‌قبول باشد آنگاه:

$$h(n)\leq c(n,a,n')+h(n')\leq c(n,a,n')+h^*(n')=h^*(n)$$

بنابراین تابع $h(n)$ در $k$+1 مرحله‌ای هدف، همچنان قابل‌قبول است.

**؟ ۸.۴ اگر برای مسألهٔ فروشندهٔ دوره‌گرد (TSP)، تابع اکتشافی را درخت پوشای مینیمم (MST) در نظر بگیریم، آنگاه با داشتن قسمتی از دوره، می‌توان هزینهٔ دورهٔ کامل را تخمین زده و مسأله حل شود. هزینهٔ MST برای مجموعه شهرها، عبارتست از حداقل مجموع لینک‌هایی که بتوانند تمام شهرها را به یکدیگر متصل کنند. الف) نشان دهید که این تابع اکتشافی می‌تواند از نسخهٔ سادهٔ TSP مشتق شود. ب) نشان دهید که تابع اکتشافی MST بر فاصله خط مستقیم حکمفرماست (از آن بیشتر است). ج) یک مولد مسأله برای نمونه‌ای از TSP بنویسید که در آن به جای شهرها، نقاطی تصادفی در مربعی واحد**

**داشته باشیم. د) یک الگوریتم کارا برای ساخت درخت‌پوشای مینیمم یافته و در کنار آن با استفاده از یک جستجوی قابل قبول نمونه‌ای از TSP را حل کنید.** ***(این تمرین مشابه تمرین ۳.۳۰ در ویرایش سوم است)***

☑ حل: این تمرین، تکرار قسمتی از مسألهٔ کلاسیکی است که توسط Held و Karp در سال ۱۹۷۰ انجام شد. **الف)** در مسألهٔ TSP بایستی کوتاهترین مسیر بین شهرها که تشکیل یک حلقهٔ بسته دهند را بیابیم و طبق تعریف درخت پوشای مینیمال، می‌بینیم که همان نسخهٔ ساده شدهٔ TSP است زیرا به دنبال گراف مینیمالی می‌گردد که حلقهٔ بسته نداشته باشد ولی بایستی کاملا متصل باشد. بنابراین MST می‌تواند یک اکتشاف قابل‌قبول باشد. (همواره مسیری کوتاهتر و یا مساوی با یک حلقهٔ بسته ارائه می‌دهد). **ب)** در این مسأله بایستی از شهر انتهایی مجددا به شهر شروع باز گردیم که تابع اکتشافی «فاصلهٔ خط مستقیم» در این موضوع ضعیف خواهد بود. اگر تعداد شهرها زیاد باشد، این تابع مقدار را کمتر از حد واقعی تخمین می‌زند ولی در صورت کم بودن تعداد شهرها شاید کارکرد آن خیلی بد نباشد. اکتشاف MST بر اکتشاف «فاصلهٔ خط مستقیم» حکمفرماست. یعنی همواره برای یک گره، مقداری بالاتر را ارائه می‌دهد. این موضوع بدیهی است زیرا در زمان ساخت درخت‌پوشای‌مینیمال، اگر گرهٔ هدف و گرهٔ جاری را داشته باشیم می‌توانیم به دو شیوه عمل کنیم. یا آن دو را با یک خط مستقیم به هم وصل کنیم. (همان اکتشاف فاصلهٔ خط مستقیم می‌شود) و یا توسط دو یا چند خط، این کار را انجام دهیم که این کار معمولا صورت می‌گیرد (طبق قاعدهٔ مثلث، این مقدار بیشتر از فاصلهٔ خط مستقیم خواهد بود). **ج)** در پیاده سازی خود بکوشید، تا هر نقطهٔ بازدید نشده را به نزدیکترین همسایه‌اش متصل کنید و این کار بر طبق الگوریتم درخت‌پوشای‌مینیمال (MST) صورت می‌گیرد. **د)** در (Cormen etal ,1990, P50S)، الگوریتمی با پیچیدگی زمانی $O(E \log^{E})$ را مشاهده می‌کنید که در آن منظور از E همان تعداد لبه‌های گراف است.[۸]

**؟ ۹.۴ در فصل ۳، نسخه‌ای از پازل 8- تایی ارائه شد که در آن یک مکعب می‌توانست از خانهٔ A به خانهٔ B برود، در صورتی که خانهٔ B، خالی باشد. راه‌حل دقیق این مسأله توسط اکتشاف Gaschnig (Gaschnig,1979) انجام شده است. توضیح دهید که چرا دقت این تابع اکتشافی کمتر از تابع « $h1$= تعداد مربع‌هایی که در خانهٔ اشتباهی هستند» می‌باشد؟ مواردی را نشان دهید که این تابع مقداری کمتر از $h1$ و $h2$ داشته باشد ($h2$= فاصلهٔ منهتن). چگونه می‌توان این تابع اکتشافی را به طور کارا محاسبه نمود.** ***(این تمرین مشابه تمرین ۳.۳۱ در ویرایش سوم است)***

☑ حل: برای مسأله پازل 8- تایی در حالتی که یک مربع بتواند از خانهٔ A به خانه B حرکت کند. دقیق‌ترین اکتشاف همان «$h1$ = تعداد مربع‌هایی که در خانهٔ اشتباهی هستند» می‌باشد ولی در نسخهٔ ساده‌تر این مسأله یعنی انتقال در صورت خالی بودن خانهٔ B، مقدار اکتشاف Gaschnig کمتر از مقدار اکتشاف $h1$ نبوده و همواره قابل‌قبول است. ( برای نسخهٔ سادهٔ این مسأله دقیق است). بنابراین اکتشاف Gaschnig دقیق‌تر خواهد بود. اگر در حالت هدف، مجاز باشیم تا دو مربع همسایه را پس‌وپیش کنیم آنگاه حالتی داریم که در آن اکتشاف «$h1$ = تعداد مربع‌های اشتباه و $h2$ = فاصله منهتن» مقدار 2 را بر می‌گردانند ولی اکتشاف Gaschnig مقدار 3 را در بردارد. به منظور محاسبهٔ مقدار اکتشاف Gaschnig این مراحل را تکرار کنید تا به حالت هدف برسید: فرض کنید B، مکان فعلی خالی باشد، اگر برای رسیدن به هدف بایستی در خانهٔ B، مربع X قرار گیرد. مربع X را به خانهٔ B منتقل کنید وگرنه یکی از مربع‌هایی که در مکان اشتباه هستند را به خانهٔ B منتقل کنید دانشجویان می‌توانند ثابت کنند که این روش برای این نسخه از مسأله راه‌حل بهینه را می‌یابد.

**؟ ۱۰.۴ برای مسأله پازل 8- تایی دو تابع اکتشافی پیشنهاد شد: فاصلهٔ منهتن و دیگری تعداد مربع‌هایی که در مکان اشتباه هستند. در مراجع، توابع اکتشافی دیگری جهت بهبود این امر پیشنهاد شده است. به عنوان مثال : Nilson(1971) و Mostow and Prieditis (1989) و Hansson *et al.* (1992) . با پیاده‌سازی این توابع اکتشافی، ادعای آنها را بررسی نموده و کارایی نتایج حاصل از الگوریتم‌های آنها را با هم مقایسه کنید.** ***(این تمرین مشابه تمرین ۳.۳۲ در ویرایش سوم است)***

☑ حل: دانشجویان به منظور مقایسهٔ نتایج آنها بایستی از جدولی حاوی زمان اجرا و تعداد گره تولیدی در هر یک استفاده کنند (توابع اکتشافی مختلف، هزینهٔ محاسباتی مختلفی دارند). از آن جا که زمان اجرا در مسألهٔ پازل 8- تایی بسیار کوتاه است می‌توانید آزمون خود را بر حسب پازل‌هایی 15- تایی یا 24- تایی قرار دهید. استفاده از پایگاه الگو، بدترین شیوهٔ تجربه و تحقیق است.

**؟ ۱۱.۴ نام الگوریتمی را ذکر کنید که از هر یک از این موارد خاص نتیجه می‌شود[۹]: الف) جستجوی پرتومحلی با $k$=1 ب) جستجوی پرتو محلی با یک حالت شروع که هیچ محدودیتی در نگهداری تعداد حالات ندارد. ج) شبه‌تاب‌کاری( Simulated**

---

[8] روش‌های ساخت درخت‌پوشای مینیمم را می‌توانید در درس ساختمان‌داده، به طور کامل مطالعه کنید. از آن جمله می‌توان به روش‌های پریم و سولین اشاره نمود. هدف این الگوریتم‌ها آن است که با داشتن گرافی متشکل از چندین رئوس و هزینه هر یال، بتوان درختی تشکیل دهیم که تمام رئوس در آن موجود باشند و مجموع هزینه یال‌های انتخابی، کمترین مقدار ممکن باشد. یکی از این روش‌ها به این صورت است که از رأسی دلخواه شروع به حرکت کرده و از بین یال‌های موجود، یال با کمترین هزینه را به شرطی انتخاب می‌کنیم که دور ایجاد نشود. به همین شیوه تا پایان رئوس ادامه می‌دهیم.

[9] جستجوی تپه‌نوردی که از دسته روشهای جستجوی‌محلی محسوب می‌شود به سان عملکرد یک انسان نابینا بر روی دامنه کوه است که موقعیت خود را نمی‌داند ولی قصد رسیدن به قله را دارد. این فرد در هر گام، با استفاده از پای خود زمین اطراف را لمس کرده و هر سمتی که به سوی بالا باشد را ادامه می‌دهد. اگر به موقعیتی برسد که همه نقاط اطرافش رو به پایین باشند، نتیجه میگیرد که به قله کوه رسیده است.

**annealing) با T=0 در همه زمان‌ها (و حذف آزمون پایانی) د) الگوریتم ژنتیک با اندازه جمعیت N=1 *(این تمرین مشابه تمرین ۴.۱ در ویرایش سوم است)***

☑ حل: **الف)** جستجوی پرتومحلی با $k$=1 همان جستجوی تپه‌نوردی است. (طبق توضیحات داده شده اگر به جای $k$ نقطه، فقط یک نقطه در نظر بگیریم، دقیقا روش تپه‌نوردی حاصل می‌شود) **ب)** جستجوی پرتومحلی با $k$=∞ تاثیر زیادی ندارد. ایده این روش به این صورت است که به علت بی‌نهایت بودن $k$، بایستی پسین تمام نقاط را نگهداری کنیم. لذا این جستجو، همان جستجوی اول‌سطح خواهدشد که در آن یک لایه از گره‌ها قبل از افزودن لایه بعدی، اضافه می‌گردند. با شروع از یک گره، این الگوریتم دقیقا همان جستجوی اول‌سطح را پی‌می‌گیرد. با این تفاوت که هر لایه به طور کامل در یک لحظه تولید می‌شود. **ج)** با حذف مرحله پایانی، این جستجو تبدیل به جستجوی «تپه‌نوردی با اولین انتخاب» می‌گردد. زیرا هر پسین روبه‌پایین با احتمال یک رد می‌شود.[10] **د)** اگر اندازه جمعیت 1 باشد، آنگاه در زمان انتخاب والدین بایستی هر دو والد را همان یک نفر انتخاب کنیم و پس از انجام عملیات پیوند(Crossover) بین آنها، فرزند ایجاد شده دقیقا کپی همان فرد اولیه خواهد بود و شانس ایجاد تغییر در مرحله جهش(Mutation) بسیار کم می‌گردد. بنابراین این الگوریتم عملکردی مانند حرکت تصادفی در یک فضای منحصر بفرد را دارد.

**؟ ۱۲.۴ در برخی مسائل برای مقایسه دو گره جهت جستجو، هیچ تابع‌ارزیاب[11] مناسبی وجود ندارد تا بتوان به هر گره مقداری عددی را انتساب داد ولی یک روش مقایسه مناسب وجود دارد: یعنی روشی هست که بدون انتساب مقادیری عددی به گره‌ها، می‌گوید که یک گره بهتر از گره دیگر است. نشان دهید که این شیوه برای انجام جستجوی اول‌بهترین کافی است. آیا راه‌کاری مشابه، برای جستجوی A* وجود دارد؟**

☑ حل: اگر فرض کنیم تابع‌مقایسه برای گره‌های مختلف، مقادیری متفاوت گزارش دهد یعنی به عنوان مثال بگوید این گره خیلی‌خوب، گره دیگر خوب و گره بعدی بد باشد و یا سایر مقادیری که بتوان آنها را از جهت ارزش مرتب نمود، آنگاه همواره می‌توان با استفاده از ارزش آن عناصر (ارزش کلامی و نه مقداری)، لیستی مرتب از گره‌ها در هر لحظه تشکیل داد و سپس گره‌ای که در بالاترین جایگاه این لیست مرتب قرار دارد را برای بسط انتخاب کنیم. برای این منظور می‌توان از صف‌اولویت استفاده کرد که ساختمان‌داده‌ای است که در عملیات مقایسه عملکردی مناسب داشته و کارایی را کاهش نمی‌دهد. البته عملیات مقایسه روی گره‌ها بسیار پرهزینه‌تر از عملیات مقایسه بر روی دو عدد است که همزمان تولید شده‌اند. روش A*، هزینه ارزیابی $f(n)$را متشکل از دو نوع هزینه می‌داند: هزینه‌ای که از ابتدا تا رسیدن به این گره صرف شده‌است ($g(n)$) و دیگری هزینه‌ای که از این گره تا هدف بایستی پرداخته شود ($h(n)$). اگر برای هر یک از این دو مورد، تابع‌مقایسه داشته باشیم آنگاه برای بسط یک گره، در صورتی یک گره را بر دیگری ترجیح می‌دهیم که هم تابع‌مقایسه مربوط به g و هم تابع‌مقایسه مربوط به $h$ برای آن گره ارزش بیشتری را گزارش دهند. متأسفانه چنین حالتی میسر نیست زیرا سبک‌وسنگین کردن(Trade off) بین $g(n)$ و $h(n)$ بدون مقادیر عددی امکانپذیر نیست. (اگر برای دو گره تابع مقایسه $g$ به‌ترتیب کلمات خوب و بد، همچنین تابع مقایسه h به‌ترتیب کلمات زیاد و کم را گزارش دهد نمیتوان گفت که ترکیب {خوب+زیاد} هزینه کمتری را شامل می‌شود و یا {بد+کم}).

**؟ ۱۳.۴ رابطه بین پیچیدگی زمانی LRTA* به پیچیدگی فضایی آن را شرح دهید.**

☑ حل: پیچیدگی فضایی LRTA* به چند مورد بستگی دارد: فضای موردنیاز برای $result[a,s]$، حاصلضرب تعداد گره‌های ملاقات‌شده ($n$) در تعداد واکنش‌ها در هر حالت ($m$) می‌باشد که پیچیدگی زمانی در پیاده‌سازی پایه $O(nm^2)$ خواهد بود. زیرا برای انجام هر واکنش بایستی H مقدار محاسبه شود که اینکار نیازمند مینیمم‌سازی بر روی کل واکنش‌ها می‌باشد. یک راه‌کار بهینه از این روش دارای پیچیدگی $O(nm)$ است. این عبارت نشان می‌دهد که زوج حالت/ واکنش تاجای امکان یکبار رخ می‌دهد تا آنکه هر جفت چندین بار رخ دهد. همان طور که در شکل ۲۲.۴ مشاهده می‌کنید.

**؟ ۱۴.۴ فرض کنید یک کارگزار در یک محیط پرپیچ‌وخم 3×3 که در شکل ۱۸.۴ می‌بینید قرار گرفته است. کارگزار می‌داند که موقعیت شروع آن در مختصات (1,1) و هدف در مختصات (3,3) بوده و مجاز است تا زمانیکه به دیوار برخورد نکرده است، به جهات بالا، پایین، چپ و راست حرکت کند. کارگزار از مکان دیوارها باخبر نیست، بنابراین در نظر کارگزار تعداد $2^{12}$=4096 حالت مختلف برای چیدمان‌های مختلف دیوارها که به آن پیکره‌بندی محیط گفته می‌شود، وجود دارد. هر حالت باور در نظر**

---

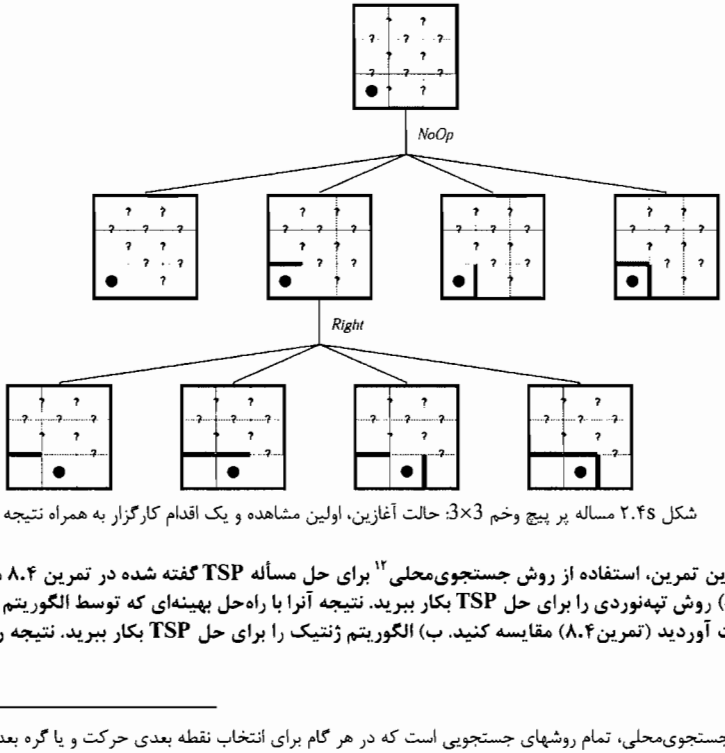
بنابراین اگر به جایی برسد که قله اصلی کوه نباشد ولی قله‌ای کوچک محسوب شود این فرد متوقف می‌شود. ولی در روش پرتومحلی، فرض می‌کنیم $k$ نفر بر روی نقاط مختلف کوه قرار دارند و همگی در هر گام، بهترین پسین خود را انتخاب می‌کنند و انتخاب بهتر را با یکدیگر به اشتراک می‌گذارند. (مشابه حرکت $k$ فرد نابینا که اطلاعات گام بعدی خود را با موبایل به هم اطلاع می‌دهند). بنابراین همگی به مرور در اطراف انتخاب برتر جمع می‌شوند. می‌توان همین ایده را به جای پیمایش بر روی یک کوه، بر روی نموداری ریاضی که نشان‌دهنده کارایی است درنظرگرفت و یا با اعمال کمی تغییرات در آن به جای یافتن ماکزیمم مطلق، مینیمم مطلق را بیابیم.

[10] جستجوی شبه‌تاب‌کاری در اوایل حرکت خود یعنی زمان‌های اولیه، حرکتی به شدت تصادفی دارد یعنی ممکن است گام بعدی در پیمایش کوه به سمت پایین انتخاب شود. این انتخاب رو به پایین همواره و در هر گام انجام می‌شود ولی با گذشت زمان، احتمال انتخاب حرکتی رو به پایین کمتر و کمتر می‌شود. بنابراین دیگر در قله‌ای محلی گیر نمی‌کنیم. ولی اگر شرط زمان از این الگوریتم برداشته شود، دیگر این حرکت رو به پایین و تغییرات احتمالی آن رخ نداده و در هر گام اولین پسینی که بهتر از موقعیت فعلی باشد انتخاب می‌گردد.

[11] منظور از تابع ارزیاب، کل هزینه‌ای است که به یک گره انتساب داده می‌شود که معمولا شامل مجموع هزینه‌های سپری شده تا آن گره به علاوه هزینه آن گره تا هدف می‌باشد. این تابع برای هر گره مقدار هزینه آن را نشان می‌دهد و معمولا با نماد $f(n)$نمایش داده می‌شود.

**ارگزار، زیرمجموعه‌ای از این پیکره‌بندی‌های ممکن را شامل می‌شود. الف) توضیح دهید که این مسأله‌جستجوی آنلاین چگونه ی تواند به یک مسأله‌جستجوی آفلاین تبدیل شود. که در آن حالت باور شروع، شامل تمام پیکره‌بندی‌های ممکن محیط باشد. ندازه حالت باور شروع چقدر است؟ اندازه فضای حالت باور چقدر است؟ ب) چند ادراک مختلف در حالت شروع ممکن است؟ ) چند شاخه نخست از نقشه محتمل این مسأله را توصیف کنید؟ اندازه نقشه کامل چقدر است؟ توجه داشته باشید که این شه محتمل، راه‌حلی برای هر پیکره‌بندی در تعریف فوق محسوب می‌شود. بنابراین یکی درمیان بودن جستجو و اجرا حتی در حیط‌های ناشناخته خیلی موردنیاز نیست. *(این تمرین مشابه تمرین ۱۲.۴ در ویرایش سوم است)***

**حل: الف)** با توجه به آنکه فرض شد تمام پیکره‌بندی‌های ممکن، در آغاز درون حالت باور کارگزار قرار داشته باشند پس الت باور اولیه شامل مجموعه‌ای از تمام 4096 پیکره‌بندی است (تمام چیدمان‌های مختلف محیط که کارگزار نمی‌داند کدام ک واقعیت دارد و در آغاز همه را در نظر می‌گیرد). بنابراین کل فضای حالت باور به تعداد زیرمجموعه‌های این مجموعه یعنی امل $2^{4096}$ حالت باور می‌باشد. (به ازای هر زیرمجموعه از پیکره‌بندی‌ها، یک حالت حساب می‌شود ولی بسیاری از آنها قابل ستیابی نیستند). پس از هر مشاهده و واکنش انجام‌شده، کارگزار می‌آموزد که بین خانه جاری و همسایگان آن آیا دیواری جود دارد یا خیر. بنابراین کارگزار می‌تواند هر حالت باور قابل‌دستیابی را با یکی از مقادیر (موجود، غیرموجود و ناشناخته) به ور مجزا توصیف کند. بنابراین حالت باور کاملا مشخص شده و تعداد $3^{12}$ حالت باور قابل‌دسترس وجود خواهد داشت. کارگزار ر هر لحظه می‌تواند چهار سوی خود را مشاهده کرده و وجود یا عدم وجود دیوارها در آن خانه را بررسی نماید و با توجه به شاهدات خود در هر خانه وضعیت کاملا برایش روشن می‌شود و دیگر نیازی نیست تا باورهای خیلی زیادی در مورد دیوارهای خانه حدس بزند پس تمام زیرمجموعه‌های ممکن برای وضعیت هر خانه برابر 16 ($2^4$) خواهد بود و سایر حالات باور که با شاهدات کنونی آن متفاوت باشد حذف می‌شوند زیرا در هر حالت باور 4 واکنش پیش‌رو دارد. **ب)** با فرض دانستن مکان یوارهای خارجی، در لحظه شروع دو دیوار داخلی مشاهده گشته و در نتیجه $2^2=4$ حالت ممکن وجود دارد. **ج)** واکنش اولیه جر به چهار حالت‌باور ممکن می‌شود که در شکل ۲.۴s نمایش داده شده‌است. در هر حالت باور، کارگزار یک واکنش انتخاب ی‌کند تا به 8 حالت باور برسد (در زمان ورود به مربع وسط). با تکرار گام‌های حرکت این کارگزار در رسیدن به پایان، می‌بینیم ه کارگزار کل محیط پرپیچ‌وخم را در حداکثر 18 مرحله سپری می‌کند. بنابراین نقشه کامل (به صورت یک درخت) بیشتر از 8 گره نخواهد داشت. به بیانی دیگر، فقط $3^{12}$ گره وجود دارد بنابراین می‌توان نقشه را به صورت جدولی از واکنش‌ها بیان کرد ه توسط حالت عقیده ایندکس شود.




شکل ۲.۴s مساله پر پیچ وخم 3×3: حالت آغازین، اولین مشاهده و یک اقدام کارگزار به همراه نتیجه آن.

**۱۵.۴ در این تمرین، استفاده از روش جستجوی‌محلی[۱۲] برای حل مسأله TSP گفته شده در تمرین ۸.۴ مورد تحقیق قرار یگیرد. الف) روش تپه‌نوردی را برای حل TSP بکار ببرید. نتیجه آنرا با راه‌حل بهینه‌ای که توسط الگوریتم A* با تابع اکتشافی MST بدست آوردید (تمرین۸.۴) مقایسه کنید. ب) الگوریتم ژنتیک را برای حل TSP بکار ببرید. نتیجه را با سایر روش‌ها**

[۱] منظور از جستجوی‌محلی، تمام روشهای جستجویی است که در هر گام برای انتخاب نقطه بعدی حرکت و یا گره بعدی برای بسط، فقط از لاعات همسایگان آن نقطه کمک گرفته شود مانند روش تپه‌نوردی.

**مقایسه کنید. می‌توانید جهت مشورت به Larranaga *et al.* (1999) و پیشنهادات آن مراجعه کنید. *(این تمرین مشابه تمرین ۳.۴ در ویرایش سوم است)***

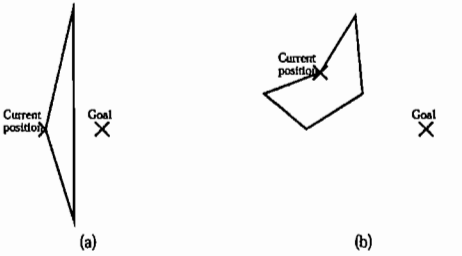
☑ حل: یک الگوریتم تپه‌نوردی ساده برای TSP به این شرح است: • تمام شهرها را توسط مسیری اختیاری به هم وصل کنید. • دو نقطه تصادفی روی مسیر انتخاب کنید. • مسیر را از این دو نقطه جدا کنید تا سه قطعه حاصل شود. • تمام 6 روش ممکن برای اتصال این سه قطعه به یکدیگر را امتحان نمایید و بهترین حالت اتصال را انتخاب کرده و مسیر را طبق آن مجدداً متصل کنید. • مراحل فوق را آنقدر تکرار کنید تا هیچ بهبودی حاصل نشود.

**؟ ۱۶.۴ نمونه‌های زیادی از مسأله 8- پازل و 8- وزیر تولید کرده و در صورت داشتن پاسخ، آنها را به روش‌های تپه‌نوردی (انواع بیشترین‌شیب و اولین‌انتخاب)، تپه‌نوردی‌باشروع‌مجدد‌تصادفی، و شبه‌تاب‌کاری حل کنید. هزینه جستجو و درصد مسائل حل شده را بیابید و سپس آنها را به همراه هزینه راه‌حل بهینه بر روی نموداری ترسیم کنید. در مورد نتایج خود توضیح دهید. *(این تمرین مشابه تمرین ۴.۴ در ویرایش سوم است)***

☑ حل: (راهنمایی: برای حل یک پازل 8- تایی به روش تپه‌نوردی بیشترین شیب، یک چیدمان تصادفی از اعداد 1 تا 8 را به عنوان حالت شروع در نظر بگیرید. سپس مقدار $h$ در آن حالت را محاسبه کنید. حال تمام حالت‌هایی که یک خانه تغییر می‌کند را امتحان کرده و در هر یک مقدار $h$ را مجدداً محاسبه نمایید. حالتی که کمترین مقدار $h$ را داشت، به عنوان حرکت بعدی در روش بیشترین شیب برگزینید. سپس آن حرکت را انجام داده و همین کار را برای آن حالت تکرار کنید تا به وضعیتی برسید که $h=0$ بوده و خانه‌ها مرتب شوند. در روش تپه‌نوردی اولین انتخاب، در هر مرحله نیازی نیست تا تمام حرکت‌های مجاز بعدی را امتحان کرده و بهترین را انتخاب کنید. بلکه حرکت‌های دیگر را امتحان کرده و اولین حرکتی که $h$ ای بهتر از h فعلی داشت را انتخاب کنید. در روش تپه‌نوردی با شروع مجدد تصادفی، چندین پازل را به طور همزمان در نظر گرفته و به طور موازی با روش تپه‌نوردی حل کنید. هدف یافته‌شده بین آنها را به اشتراک گذاشته و بهترین را انتخاب کنید زیرا ممکن است یکی از پازل‌ها در مینیمم محلی $h>0$ گیرکرده باشد. به‌همین‌ترتیب مراحل را برای مسأله 8- وزیر تکرار کنید با این تفاوت که در آنجا هر حالت بعدی با حرکت یکی از وزیرها حاصل می‌شود که بایستی در آن وضعیت مقدار $h$ را مجدداً محاسبه کرد.)

**؟ ۱۷.۴ در این تمرین تپه‌نوردی را در مسأله هدایت روبات بررسی می‌کنیم. به عنوان مثال از محیط شکل ۲۲.۳ استفاده کنید. الف) تمرین ۱۶.۳ را با تپه‌نوردی تکرار کنید. آیا کارگزار شما در یک مینیمم‌محلی[13] گیر می‌کند؟ آیا ممکن است این کارگزار در گوشه (محدب) موانع گیر کند؟ ب) یک محیط با چندضلعی‌های بدون گوشه بسازید که کارگزار در آن گیر کند. ج) الگوریتم تپه‌نوردی را به گونه‌ای تغییر دهید که جهت حرکت بعدی به جای انجام جستجویی با عمق یک، جستجویی با عمق $k$ انجام دهد. این روش بایستی بهترین مسیر $k$ مرحله‌ای را یافته و در طول مسیر آن یک گام بردارد سپس این فرایند را مجدداً تکرار کند. د) آیا مقدار $k$ ای وجود دارد که در این الگوریتم جدید، فرار از مینیمم محلی را تضمین کند؟ هـ) توضیح دهید که در این مورد، LRTA* چگونه در فرار از مینیمم‌محلی به کارگزار کمک می‌کند؟ *(این تمرین مشابه تمرین ۱۳.۴ ویرایش سوم است)***

☑ حل: تپه‌نوردی روشی است که در یافتن مسیر قابل‌قبول در زمانیکه مسیر بهینه وجود ندارد و هزینه محاسبه کم است، بسیار کارا است ولی در محیط‌های دوبعدی به تنهایی شکست می‌خورد. **الف)** این موضوع که در شکل ۳.۴s (a) بیان شده است، ممکن ولی بسیار بعید است. زیرا موانع بایستی یک شکل غیرمتعارف داشته و در مسیر هدف قرار داشته باشند. **ب)** در محیطی با موانع گوشه‌دار احتمال گیر کردن بسیار زیاد است. (قسمت (b) در شکل ۳.۴s را ببینید.) **ج)** دقت کنید که این روش همان جستجوی عمق محدود است که در آن یک گام در مسیر بهینه انتخاب می‌کند. حتی اگر یک راه‌حل نباشد. **د)** اگر $k$ را برابر حداکثر اضلاع چندضلعی در نظر بگیریم، آنگاه همواره فرار صورت می‌گیرد.




شکل ۳.۴s (a) گیرافتادن در یک مانع گوشه‌دار (b)گیرافتادن در یک مانع غیرگوشه‌دار

[13] منظور از مینیمم‌سراسری یا مطلق، نقطه‌ای است که از کل نقاط یک نمودار کمتر است و مینیمم‌محلی، نقطه‌ای است که از نقاط همسایه خود کمتر است ولی از مینیمم‌سراسری بیشتر است. در مسائل جستجوی‌محلی، برخی الگوریتم‌ها این مینیمم‌های‌محلی را با سراسری اشتباه گرفته و در همانجا متوقف می‌شوند. معمولا مینیمم‌سراسری دارای $h=0$ و مینیمم‌های محلی دارای $h$ کوچک ولی بیشتر از صفر هستند.

**۱۸.۴ کارایی روش‌های A* و RBFS را برا روی مجموعه‌ای از مسائل TSP و پازل‌های 8- تایی که به طور تصادفی تولید شده‌اند، مقایسه کنید. در مورد نتایج خود توضیح دهید. اگر یک مقدار تصادفی کوچک به مقدار اکتشافی افزوده شود چه تغییری بر روی کارایی RBFS خواهد داشت؟** ***(این تمرین مشابه ۳.۲۲ در ویرایش سوم است)***

حل: در مسألهٔ پازل 8- تایی، روش RBFS تعداد گره‌های بیشتری را بسط می‌دهد (به خاطر عدم تشخیص حالت‌های تکراری) ولی به ازای هر گره هزینه کمتری در بردارد. زیرا در این روش نیازی به نگهداری یک صف نداریم. البته در روش RBFS تعداد گره‌هایی که مجددا بسط می‌یابند، خیلی زیاد نیستند زیرا مسیر بهینه به ندرت تغییر می‌کند ولی زمانیکه مقدار اکتشافی کمی مناسب نباشد، این مزیت از بین رفته و کارایی RBFS به شدت کاهش می‌یابد. در مسألهٔ TSP، فضای حالت به شکل درخت است پس حالت‌های تکراری نداریم یا به بیانی دیگر مقادیر اکتشافی حقیقی بوده و به مقادیر حدسی نیازی نیست. براین RBFS محکوم به بسط مجدد گره‌هایی است که قبلا بسط یافته‌اند.

# فصل ۵ (ویرایش دوم)

**؟ ۱.۵ این لغات را به زبان خود تعریف کنید: مسئله ارضای‌محدودیت[14]، محدودیت، جستجوی بازگشت به عقب (پسگرد)، سازگاری کمان، پرش به عقب و تابع اکتشافی حداقل تناقاضات. *(این تمرین مشابه تمرین ۶.۶ در ویرایش سوم است)***

☑ حل: یک مسئله ارضای‌محدودیت، به طور خلاصه مسئله‌ای است متشکل از تعدادی متغیر، دامنه متغیر و تعدادی قانون محدودیت. هر متغیر دارای دامنه مقدار یعنی مجموعه مقادیر مجاز برای انتساب می‌باشد و هر محدودیت، قانونی ریاضی یا تعریفی بر روی متغیرهاست که برای انتساب مقدار به متغیرها شرط می‌گذارد. این تعریف ساده‌ای از مساله CSP است که می‌توان مسائل مختلف را با یافتن این سه مجموعه (متغیر، محدودیت، دامنه) تدوین نمود. به عنوان مثال اگر بخواهیم مساله رنگ‌آمیزی نقشه استرالیا را به عنوان CSP تدوین کنیم، بایستی این سه مجموعه را بیابیم. در اینجا مجموعه متغیرها {WA,NT,Q,SA,NSW,V,T}، مجموعه محدودیت‌ها {دو همسایه همرنگ نباشند} و دامنه هر متغیر {قرمز، سبز، آبی} می‌باشد. منظور از حل یک مساله CSP آن است که به هر متغیر مقداری را از دامنه‌اش انتساب دهیم و هیچ یک از قوانین محدودیت نقض نشود. مثلا یکی از راه‌حل‌های ممکن برای مساله فوق عبارتست از {WA=red,NT=green,Q=red,SA=blue,NSW=green,V=red,T=green} که هیچ محدودیتی را نقض نمی‌کند. یک محدودیت، یک قانون یا رابطه است که بر روی مقادیر ممکن متغیرها تعریف می‌شود و مقادیر مجاز آنها را محدود می‌کند. به عنوان مثال یک محدودیت برای متغیر $A$ می‌تواند آن باشد که در صورت $B=b$ بودن، $A=a$ مجاز نیست. البته می‌توان انواع مختلف محدودیت‌ها را تعریف نمود مثلا محدودیت $A>2$، یا $A^2+B=5$ یا ... . جستجوی بازگشت‌به‌عقب یا پسگرد نوعی جستجوی اول‌عمق است که در هر بار به یک متغیر مقداری از دامنه‌اش را انتساب داده و به سراغ متغیر بعدی می‌رویم مانند پیشروی در عمق درخت جستجو. اگر نوبت به متغیری رسید که با توجه به مقادیر سایر متغیرها و محدودیت‌های موجود، هیچ مقداری برایش مجاز نباشد، پس نتیجه می‌گیریم که در انتساب‌های قبلی اشتباه کرده‌ایم لذا یک مرحله به عقب بازگشته و مقدار انتسابی به آخرین متغیر را تغییر می‌دهیم. در صورت مقدار دادن به تمام متغیرها بدون نقض هیچ محدودیتی، جستجو پایان می‌یابد. یک کمان مستقیم از متغیر $A$ به متغیر $B$ در مسائل ارضای‌محدودیت، کمان سازگار نامیده می‌شود، در صورتی که به ازای هر مقدار موجود در دامنه $A$، مقداری مجاز برای $B$ وجود داشته باشد. پرش‌به‌عقب، یکی از روش‌های افزایش کارایی جستجوی پسگرد است که در آن هر زمان که به نقطه‌ای رسیدیم که نیاز به بازگشت به عقب بود، به جای آنکه فقط یک مرحله به عقب پرش کنیم و مقدار را اصلاح نماییم، به نقطه آغاز اشتباه که معمولا چند سطح قبلتر می‌باشد، می‌پریم. «حداقل‌تناقضات» یک تابع اکتشافی است که در روش‌های CSP با جستجوی محلی بکار می‌رود. منظور از این تابع آن است که در زمان انتساب مقدار به یک متغیر، مقداری را برای آن انتخاب کنیم که در صورت امکان با کمترین متغیرها تداخل و تناقض داشته باشد.

**؟ ۲.۵ مسئله رنگ‌آمیزی نقشه که در شکل ۱.۵ بیان شد، چند راه‌حل دارد؟ *(این تمرین مشابه تمرین ۶.۲ در ویرایش سوم است)***

☑ حل: برای رنگ‌آمیزی نقشه استرالیا با سه رنگ، تعداد 18 راه‌حل وجود دارد. با شهر $SA$ شروع می‌کنیم که می‌تواند هر یک از آن سه رنگ را داشته باشد. سپس در خلاف جهت عقربه‌های ساعت حرکت کرده و به سراغ شهر $WA$ که دو رنگ مجاز از سه رنگ دامنه‌اش را می‌تواند بگیرد می‌رویم. با رنگ‌آمیزی این دو شهر تکلیف رنگ سایر شهرها به طور دقیق مشخص می‌شود. پس تعداد راه‌حل‌ها به تعداد روش‌های مختلف رنگ‌آمیزی دو شهر $SA$ و $WA$ بستگی دارد که عبارتند از {(r,g),(r,b),(g,r),(g,b),(b,r),(b,g)}. از آنجا که شهر $T$ نیز مستقل از رنگ سایر شهرهاست، به 3 حالت مختلف رنگ می‌شود پس مجموعا به تعداد $6\times3=18$ روش رنگ‌آمیزی مختلف وجود دارد.

**؟ ۳.۵ توضیح دهید که چرا در مسئله CSP، «انتخاب متغیری که بیشترین محدودیت را ایجاد می‌کند» و «انتخاب مقداری که محدودکننده کمتری است»، توابع اکتشافی خوبی محسوب می‌شوند؟ *(این تمرین مشابه تمرین ۶.۹ در ویرایش سوم است)***

☑ حل: منظور از متغیری که بیش از همه محدودیت ایجاد می‌کند، متغیری است که نام آن در قوانین محدودیت بسیار استفاده شده است و به عبارتی دیگر مقدار آن متغیر با سایر متغیرها بسیار دخیل خواهد بود. به همین خاطر اگر آنها را در ابتدا مقدار ندهیم معمولا در زمان انتساب مقدار به آن، مقدار مجازی نمی یابیم و راه‌حل فعلی شکست می‌خورد بنابراین این تابع اکتشافی پیشنهاد می‌دهد که جهت افزایش کارایی و رسیدن به راه‌حل، بهتر است در زمان انتخاب متغیر، ابتدا به این متغیرها مقداری مجاز را انتساب داده و سپس به سراغ دیگران که معمولا محدودیتی کمتر دارند، برویم. (با این کار گویا قسمتی از فضای جستجو را هرس کرده‌ایم). روش «انتخاب مقداری که محدودیت کمتری ایجاد می‌کند» نیز اکتشافی خوب محسوب می‌شود زیرا برای

---

[14] Constraint satisfaction problem (CSP)

لوگیری از تناقض شانس بیشتری در انتساب‌های بعدی داریم. لازم بذکر است که این تابع کاری با انتخاب متغیر ندارد و فرض
ی‌کند متغیر مورد نظر به شیوه‌ای دلخواه انتخاب شده است، و تنها بر روی مقدار انتسابی به آن کار می‌کند.

**؟ ۴.۵ مسئله ساخت (ونه راه‌حل) جدول کلمات متقاطع را در نظر بگیرید که منظور از این جدول، قراردادن تعدادی کلمه در**
**ک شبکه مستطیلی شطرنجی می‌باشد**[15]**. دراین شبکه تعدادی از خانه‌ها بایستی خالی بوده و کلمات درآن قرار گیرند و تعدادی**
**یگر از خانه‌ها هاشور بخورند. فرض کنید لیستی از کلمات نظیر قسمتی از دیکشنری را داریم و می‌خواهیم با استفاده از روشی**
**شخص زیرمجموعه‌ای از این کلمات را درون خانه‌های خالی این جدول بگنجانیم. این مساله را به دو روش زیر تدوین کنید:**
**ف) به عنوان یک مسئله جستجوی عمومی، یک الگوریتم جستجوی مناسب انتخاب کرده و در صورت نیاز، یک تابع اکتشافی**
**ز تعیین نمایید. بهتر است درخانه‌های خالی در هر بار یک کلمه قرار دهیم یا در هر بار یک حرف؟ ب) به عنوان یک مساله**
**ضای‌محدودیت، متغیرهای این مساله باید حروف باشند یا لغات؟ به نظر شما کدام تدوین بهتر است؟ چرا؟** ***(این تمرین مشابه***
***مرین ۳.۶ در ویرایش سوم است)***

☒ حل: **الف)** جدول کلمات متقاطع را می‌توان به روش‌های زیادی حل نمود که یکی از ساده‌ترین روش‌ها، جستجوی اول‌عمق
ست. یعنی در هر بار یک لغت را در رشته‌ای از خانه‌های خالی قرار می‌دهیم و به سراغ لغت بعدی می‌رویم. اگر به جایی رسیدیم
ه نتوانیم لغت بعدی را قرار دهیم (به علت سازگار نبودن با تعداد خانه‌ها خالی یا حروف مشترک) بایستی به سراغ آخرین
نتساب رفته و آن را تغییر دهیم. اگر در تغییر آن نیز با مشکل روبرو بودیم باز به سراغ یک مرحله قبلتر رفته و آن را تغییر
ی‌دهیم. این رفتن به عمق و بازگشت به عقب طبق جستجویی اول‌عمق انجام می‌شود. همچنین بهتر است در هر بار یک لغت
ر شود تا یک حرف، زیرا تعداد مراحل موردنیاز کاهش می‌یابد. **ب)** از دید مساله ارضای محدودیت نیز همچنان راه‌حل‌های
یادی وجود دارد. برای تدوین مساله به CSP باید سه مجموعه متغیر، دامنه و محدودیت را بیابیم. می‌توان برای هر خانه
دول، یک متغیر با دامنه یک حرف تعریف نمود و باید تعداد زیادی محدودیت تعریف شود که شرط بگذارد این حروف تشکیل
ک کلمه را بدهند. در این راه‌کار بایستی قوانین محدودیت زیادی تعریف شود. به عنوان راه‌کاری دیگر، می‌توان مجموعه
انه‌های متوالی قرار گرفته در یک سطر یا ستون را به عنوان یک متغیر در نظر گرفته و دامنه متغیرها همان لغات دیکشنری با
ل سایز مربوطه باشند. از قوانین محدودیت می‌توان به این مورد اشاره نمود که در زمان تداخل دو کلمه بایستی حروف مشترک
جود داشته باشد. راه‌حل این تدوین نیاز به گام‌های کمتری دارد ولی دامنه متغیرها بزرگتر بوده (مخصوصا اگر دیکشنری بزرگی
شته باشیم) ولی در عوض محدودیت‌ها کمتر هستند. البته در هر صورت هر دو تدوین صحیح بوده و می‌توانند یک جدول را
سازد.

**؟ ۵.۵ برای هریک از این مسائل ارضای محدودیت، یک تدوین دقیق بیابید: الف) برنامه‌ریزی کف اتاق با خطوط مستقیم:**
**ی‌خواهیم در یک مستطیل بزرگ (مانند اتاق) تعدادی مستطیل کوچک‌تر (مانند موزاییک) قرار دهیم به طوری که هیچ‌یک از**
**ها با دیگری هم‌پوشانی نداشته باشد. ب) زمان‌بندی کلاس‌ها: تعداد مشخصی کلاس و استاد دریک دانشگاه، لیستی از**
**لاس‌های ارائه شده و لیستی از زمان‌های ممکن برای هرکلاس موجود است و می‌دانیم که هر استاد می‌تواند مجموعه‌ای از**
**لاس‌ها را تدریس کند.** ***(این تمرین مشابه تمرین ۴.۶ در ویرایش سوم است)***

☒ حل: **الف)** همان طور که گفته شد برای تدوین یک مساله به صورت CSP باید سه مجموعه متغیر، دامنه، و محدودیت را
یین کنیم. برای موزاییک کردن کف اتاق، می‌توان برای هر موزاییک یک متغیر در نظر گرفت که دامنه آن، مجموعه‌ای از
شته‌های چهارتایی اعداد جهت تعریف مکان موزاییک می‌باشد. که این 4 عدد نشان‌دهنده دو مختصات $(x,y)$ گوشه‌های اصلی
ستطیل یعنی نقطه بالا- چپ و نقطه پایین- راست می‌باشد. راه‌حل این مساله مجموعه‌ای از رشته‌های 4- تایی است که هر
شته، با ابعاد موزاییک‌ها همخوانی داشته و مجموعه مختصات‌ها کل اتاق را بپوشانند. محدودیت‌ها عبارتند از: هیچ دو رشته‌ای با
م همپوشانی نداشته باشند، به عنوان مثال اگر مقدار متغیر $R_1$ به صورت [0,0,5,8] باشد، آنگاه هیچ متغیر دیگری نباید
قداری بگیرد که با این موزاییک از نظر مختصاتی تداخل داشته باشد. **ب)** برای زمانبندی کلاس‌ها، می‌توان به ازای هرکلاس
وجود در دانشگاه، سه متغیر تعریف نمود؛ یکی برای زمان (با مقادیری نظیر MWF9:00 , TUTH8:00 , MWF 8:00)، یکی برای مکان کلاس (با مقادیری نظیر Wheeler 110, Evans330, ...) و یکی برای مدرس کلاس (با مقادیری نظیر Abelson, Bibel, Canny, ...). محدودیت این مساله آن است که در زمانی مشخص فقط یک کلاس می‌تواند در مکانی مشخص قرار گیرد و هر مدرس فقط در یک زمان می‌تواند یک کلاس را تدریس کند. البته می‌توان بسته به شرایط مساله محدودیت‌های بیشتری نیز تعریف کرد (به عنوان مثال یک مدرس نمی‌تواند دو کلاس پشت سر هم داشته باشد و... .)

**؟ ۶.۵ مساله حساب رمزی**[16] **شکل ۲.۵ را به شیوه‌های دستی، بازگشت به عقب، وارسی پیش‌رو و آروین‌های MRV و مقدار**
**کمترین‌محدودیت حل کنید.** ***(این تمرین مشابه تمرین ۵.۶ در ویرایش سوم است)***

---

[15] گینز برگ درسال 1990 چندین روش ساخت جدول کلمات‌متقاطع را بیان کرد و لیتمن (سال 1999) یافتن راه‌حل آنها که دشوارتر است انجام داد.

[16] منظور از حساب رمزی، مساله‌ای است که در آن تعدادی از حروف الفبا با رابطه‌ای ریاضی نظیر جمع یا تفریق در کنار هم قرار می‌گیرند. ید به هر حرف الفبا عددی از صفر تا نه تخصیص دهیم که اولا هیچ دو حرفی عدد یکسان نداشته باشند و دوما رابطه ریاضی برقرار باشد. ی مساله گفته شده در شکل ۲.۵ یک راه‌حل به صورت {T=9,W=2,O=8,F=1,U=5,R=6} است.

☑ حل: مراحل دقیق بسته به متغیر و انتخاب مقداری دارد که شما آغاز کرده و ادامه می‌دهید. دراینجا یک راه‌حل ممکن را مشاهده می‌کنید: **الف)** متغیر $X_3$ را انتخاب کنید. دامنه آن $\{0,1\}$ می‌باشد. **ب)** مقدار 1 را برای متغیر $X_3$ انتخاب کنید. (نمی‌توانستیم مقدار 0 را انتخاب کنیم زیرا طبق وارسی پیش‌رو، $F$ برابر صفر می‌شد که منجر به غیرصفر شدن ارقام مجموع می‌گشت). **ج)** متغیر $F$ را در نظر بگیرید، زیرا فقط یک مقدار در دامنه آن باقی مانده است. **د)** مقدار 1 را برای $F$ انتخاب کنید. **هـ)** اکنون متغیرهای $X_2$ و $X_1$ هر دو در دامنه خود نسبت به دیگران حداقل مقدار یعنی 2 تا را دارند. دراینجا $X_2$ را در نظر بگیرید. **و)** طبق وارسی پیش‌رو هر کدام از مقادیر مشکلی ایجاد نمی‌کند، دراینجا 0 را برای $X_2$ انتخاب کنید. **ز)** اکنون متغیر $X_1$ در دامنه خود، حداقل مقدار را دارد. **ح)** مجددا مقدار اختیاری 0 را برای $X_1$ انتخاب کنید. **ط)** متغیر $O$ بایستی یک عدد زوج و کمتر از 5 باشد. ($O$ زوج است زیرا $O=T+T$ که می‌دانیم مجموع هر عددی با خودش زوج می‌شود. همچنین $O$ باید کمتر از 5 باشد زیرا طبق $O+O=R+10X_1$ که $X_1$ صفراست، بایستی مجموع $O$ با خودش، تک رقمی باشد). این قانون باعث می‌شود این متغیر بسیار محدود شود. پس ابتدا آن را انتخاب کنید. **ی)** به طور دلخواه عدد 4 را برای متغیر $O$ در نظر بگیرید. **ک)** اکنون متغیر $R$ فقط یک مقدار مجاز در دامنه خود دارد پس آن را در نظر بگیرید. **ل)** مقدار 8 را برای $R$ انتخاب کنید. **م)** اکنون متغیر $T$ فقط یک مقدار مجاز در دامنه خود دارد. پس آن را در نظر بگیرید. **ن)** مقدار 7 را برای $T$ انتخاب کنید. **س)** $U$ متغیری است که درآن عدد زوج کمتر از 9 قرار دارد پس آن را در نظر بگیرید. **ع)** تنها مقداری برای $U$ که وارسی پیش‌رو تایید می‌کند عدد 6 است. **ف)** تنها متغیر باقی‌مانده $W$ است. پس آن را در نظر بگیرید. **ض)** تنها مقدار باقی‌مانده عدد 3 است پس برای $W$ انتخاب کنید. **گ)** تمام متغیرها مقدار گرفته و هیچ یک از آنها محدودیتی را نقض نمی‌کنند پس یک راه‌حل یافته شد. این مساله معمایی ساده است بنابراین تعجبی ندارد که در یافتن راه‌حل به هیچ بازگشت به عقبی نیاز نشد و با همان وارسی پیش‌رو به راه‌حل رسیدیم. ولی در صورت رسیدن به تناقض باید یک مرحله به عقب پسگرد کرده و مقدار آخرین متغیر را تغییر داد.

**؟ ۷.۵ شکل ۵.۵ الگوریتم‌های مختلف را روی مساله $n$- وزیر بررسی می‌کند. این الگوریتم‌ها را بر روی مساله رنگ‌آمیزی نقشه که به طور تصادفی و به این صورت تولید شده است، اعمال کنید: $n$ نقطه را درمربعی واحد پخش کنید، یک نقطه $x$را به صورت تصادفی انتخاب کنید، نقطه $x$را توسط خطی مستقیم به نزدیکترین نقطه با نام $y$متصل کنید به نحوی که این دو نقطه قبلا متصل نبوده باشند و این خط اتصالی از روی خطی دیگر عبور نکند، مرحله قبل را آنقدر تکرار کنید که دیگر هیچ اتصالی ممکن نباشد جدول کارایی به ازای تعداد رنگ‌های $d=4$ و $d=3$ و بزرگترین $n$ که شما قادر به مدیریت آن هستید را تشکیل دهید. درمورد نتایج بدست آمده توضیح دهید.** ***(این تمرین مشابه تمرین ۱۰.۶ در ویرایش سوم است)***

☑ حل: الگوریتم‌هایی که مساله CSP را پیاده سازی می‌کنند(به زبان‌های جاوا، لیسپ و پایتون) به شروع کار کمک می‌کنند. ولی بایستی کدهایی به آنها اضافه شود تا بتواند بر روی نتایج سرشماری گردد و حتی مکانیزم‌هایی افزوده شود که در صورت گذر از زمانی خاص یا تعداد مراحلی مشخص، مقدار شکست را برگرداند. البته این مقدار کد مورد نیاز کم می‌باشد و این تمرین بیشتر قصد اجرا و تحلیل آزمایشات را دارد.

**؟ ۸.۵ با استفاده از الگوریتم AC-3 نشان دهید که سازگاری کمان قادر است تا در مساله شکل ۱.۵ ناسازگاری حاصل از انتساب جزئی {WA=red , V= blue} را تشخیص دهد.** ***(این تمرین مشابه تمرین ۱۱.۶ در ویرایش سوم است)***

☑ حل: لازم بذکر است که منظور از انتساب کامل آن است که به تمام متغیرها مقداری را انتساب داده و هیچ تناقضی مشاهده نشود. و انتساب جزئی آن است که فقط به تعدادی از متغیرها مقدار داده و هنوز به پایان راه‌حل نرسیده باشیم. به منظور تشخیص ناسازگاری در یک انتساب جزئی، هرتکرار از حلقه **While** در الگوریتم $AC$=3 را دنبال می‌کنیم. (برای هر چیدمان مجاز از کمان‌ها) **الف)** حذف کمان $SA$-$WA$، برداشتن مقدار$R$ از دامنه متغیر $SA$. **ب)** حذف کمان $SA$-$V$ و برداشتن مقدار $B$ از متغیر $SA$، تنها مقدار $G$ باقی می‌ماند. **ج)** حذف کمان $NT$-$WA$، برداشتن مقدار $R$ از دامنه متغیر $NT$ **د)** حذف کمان $NT$-$SA$ و برداشتن مقدار $G$ از دامنه متغیر $NT$، تنها مقدار $B$ باقی می‌ماند. **هـ)** حذف کمان $NSW$-$SA$، برداشتن مقدار $G$ از دامنه متغیر $NSW$. **و)** حذف کمان $NSW$-$V$ و برداشتن مقدار $B$ از دامنه متغیر $NSW$، تنها مقدار $R$ باقی می‌ماند. **ز)** حذف کمان $Q$-$NT$، برداشتن مقدار $B$ از متغیر $Q$. **ح)** حذف کمان $Q$, $SA$ برداشتن مقدار $G$ از دامنه متغیر از متغیر $Q$. **ط)** حذف کمان $Q$-$NSW$، برداشتن مقدار $R$ از دامنه متغیر $Q$ هیچ مقداری در دامنه $Q$ باقی نمی‌ماند.

**؟ ۹.۵ در بدترین حالت، پیچیدگی AC-3 درمساله CSP با ساختار درختی چقدر است؟** ***(این تمرین مشابه تمرین ۱۲.۶ در ویرایش سوم است)***

☑ حل: در یک گراف با ساختار درختی، هیچ کمانی بیش از یک بار در نظر گرفته نمی‌شود، بنابراین الگوریتم $AC$-3 دارای پیچیدگی $O(ED)$ خواهد بود که درآن $E$ تعداد لبه‌ها و $D$ اندازه بزرگترین دامنه متغیر می‌باشد.

**؟ ۱۰.۵ الگوریتم AC-3 در زمان حذف مقداری از دامنه متغیر $X_i$ تمام کمان‌های $(X_i,X_1)$ را درون صفی قرار می‌دهد، حتی اگر هرمقدار از $X_1$ با چندین مقدار از دامنه $X_i$ سازگار باشد. فرض کنید به ازای هر کمان $(X_i,X_1)$ تعداد مقادیر باقیمانده از $X_i$ که با مقداری از $X_1$ سازگار هستند را نگهداری کنیم. توضیح دهید که چگونه این اعداد را به صورت کارا آپدیت کنیم و سپس نشان دهید که سازگاری کمان بایستی در زمان $O(n^2d^2)$ انجام شود.** ***(این تمرین مشابه تمرین ۱۳.۶ در ویرایش سوم است)***

☑ حل: ایده این کار بر مبنای پیش‌پردازش محدودیت‌ها استوار است یعنی به ازای هر مقدار از $X_i$ مقادیری از $X_i$ را بیابیم که یک کمان از $X_i$ به $X_i$ با آن انتساب‌های جزئی $X_i$ ارضاء شده و در آن صدق کند. زمان محاسبه این ساختمان‌داده بستگی به اندازه مساله دارد. بنابراین زمانی که یک مقدار از دامنه $X_i$ حذف شود از تعداد مقادیر مجاز برای هرکمان $(X_i,X_i)$ تحت آن مقدار، یک واحد کم می‌کنیم این راه‌کار بسیار شبیه به الگوریتم زنجیره‌ای پیش‌رو در فصل ۷ می‌باشد.

**؟ ۱۱.۵ نشان دهید که چگونه می‌توان با یک متغیر کمکی محدودیتی سه‌گانه مانند A+B=C را به سه محدودیت دوگانه، تبدیل نمود. می‌توانید فرض کنید که دامنه‌ها متناهی‌اند. (راهنمایی:یک متغیر جدید در نظر بگیرید که مقادیر آن ترکیبی از مقادیر دو متغیر دیگر باشد و همچنین محدودیتی نظیر« X اولین عنصر از ترکیب Y است» را نیز تعریف کنید. سپس نشان دهید که می‌توان همین کار را با محدودیت‌های دارای بیش از سه متغیر انجام داد. در نهایت نشان دهید که چگونه می‌توان محدودیت‌های یگانی را با تغییر دامنه سایر متغیرها، حذف نمود. این بدان معناست که هر مساله CPS را می‌توان به یک مساله CSP فقط با محدودیت‌های دوگانه، تبدیل نمود). *(این تمرین مشابه تمرین ۶.۶ در ویرایش سوم است)***

☑ حل: راهنمایی گفته شده در صورت مساله، راه‌حل را به طور کامل مشخص می‌کند. در محدودیتی سه‌گانه شامل سه متغیر $A$ و $B$ و $C$ نظیر $A+B=C$، در ابتدا یک متغیر جدید با نام $AB$ تعریف می‌کنیم، اگر دامنه متغیرهای $B$ و $A$ شامل مجموعه‌ای از اعداد $N$ باشد آنگاه دامنه $AB$ شامل مجموعه‌ای مرکب از $N$ از هر دو دامنه می‌باشد. به عنوان مثال $N\times N$ . اکنون سه محدودیت دوگانه داریم؛ یکی بین $A$ و $AB$ که می‌گوید هر مقدار از $A$ باید با اولین عنصر از زوج $AB$ برابر باشد؛ یکی بین $AB$ و $B$ که می‌گوید هر مقدار از $B$ باید با دومین عنصر از مقدار $AB$ برابر باشد؛ و محدودیت آخر که می‌گوید مجموع مقادیر بکار رفته در متغیر مرکب $AB$ بایستی با مقدار متغیر $C$ برابر باشد. برای سایر محدودیت‌های سه‌گانه به طور مشابه عمل می‌کنیم. همان طور که توانستیم محدودیت‌های سه‌گانه را به محدودیت‌های دوگانه تبدیل کنیم، می‌توانیم یک محدودیت 4- گانه شامل متغیرهای A و $B$ و $C$ و $D$ را نیز تبدیل به محدودیت‌های دوگانه کنیم. برای اینکار در ابتدا فقط متغیرهای $A$ و $B$ و $C$ را در نظر گرفته و طبق همان روش فوق، محدودیت‌های دوگانه را بدست می‌آوریم. سپس به محدودیت‌های سه‌گانه شامل $C$ و $AB$ و $D$ مراجعه کرده و با استفاده از تعریف متغیر جدید $CD$ آن را به محدودیتی دوگانه تبدیل می‌کنیم. با توجه به این توضیحات، می‌توانیم هر محدودیت $n$- گانه را در ابتدا به محدودیتی $N$-1 گانه تبدیل کنیم. می‌توانیم در همین مرحله محدودیت‌های دوگانه متوقف شده و به سراغ تبدیل به محدودیت‌های یگانی نرویم زیرا هر محدودیت یگانی را می‌توان با اعمال تغییراتی بر روی دامنه آن متغیر بدست آورده و آن محدودیت یگانه حذف می‌شود.

**؟ ۱۲.۵ فرض کنید کات‌ست چرخه‌ای در یک گراف، حداکثر $k$ گره داشته باشد. یک الگوریتم ساده برای یافتن یک کات‌ست چرخه‌ای مینیمال بیابید که زمان اجرای آن بیشتر از $O(n^k)$ در یک مساله CSP با $n$ متغیر نباشد. درمراجع مختلف جستجو کرده و متدهایی برای یافتن تقریبی کات‌ست چرخه‌ای مینیمال در زمانی با مرتبه چند جمله‌ای نسبت به سایز کات‌ست بیابید. یا وجود چنین الگوریتم‌هایی روش کات‌ست چرخه‌ای را عملی می‌کند؟ *(این تمرین مشابه تمرین ۱۷.۶ در ویرایش سوم است)***

☑ حل: یک الگوریتم ساده برای یافتن کات‌ست با حداکثر $k$ گره آن است که تمام زیرمجموعه‌های گره‌ها با اندازه $1,2,\ldots,k$ را شمرده و سپس به ازای هر زیرمجموعه، بررسی کنیم که آیا گره‌های باقی‌مانده تشکیل یک درخت را می‌دهند یا خیر. این الگوریتم دارای زمانی $\binom{\sum_{i=1}^{k} n}{nk}$ به مرتبه $O(n^k)$ می‌باشد.

بکر و گیگر (becker94approximation.html, 1994, http://citeseer.nj.nec.com/) الگوریتمی با نام MGA ارائه دادند. (نوعی تغییر یافته از الگوریتم حریصانه) که می‌تواند یک کات‌ست را در حداکثر دو برابر اندازه کات‌ست مینیمال بیابد و مرتبه زمانی آن $O(E+V\text{Log}V)$ می‌باشد که منظور از $E$ تعداد لبه‌ها و $V$ تعداد متغیرهاست. اینکه آیا این موضوع روش کات‌ست را عملی می‌کند یا خیر بیشتر از آنکه به الگوریتم یافتن مربوط شود به گرافی مربوط است که در آن بدنبال کات‌ست بیستیم. زیرا برای کات‌ست با اندازه $c$ قبل از حل CSP نیاز به فاکتوری نمایی ($d^c$) داریم. بنابراین هر گراف با کات‌ست بزرگ حلی دشوار خواهد داشت. حتی اگر بتوانیم بدون هیچ کوششی، کات‌ست بیابیم.

**؟ ۱۳.۵ این معمای منطقی را در نظر بگیرید: در 5 خانه با رنگ‌های مختلف، 5 نفر با ملیت‌های متفاوت زندگی می‌کنند و هر یک از آنها به نوع خاصی از سیگار، نوع خاصی از نوشیدنی، و نوع خاصی از حیوان علاقه‌مند هستند. با استفاده از این اطلاعات، پاسخ پرسش زیر را بیابید: «در کدام خانه حیوان گورخر نگهداری می‌شود و در کدام خانه آب نوشیده می‌شود؟» • مرد انگلیسی در خانه قرمز زندگی می‌کند. • مرد اسپانیایی یک سگ دارد. • مرد نروژی در اولین خانه سمت چپ زندگی می‌کند. • درخانه زرد رنگ سیگار kool مصرف می‌شود. • مردی که سیگار chester field مصرف می‌کند پس از خانه مردی زندگی می‌کند که روباه دارد. • مرد نروژی بعد از خانه آبی زندگی می‌کند. • مردی که سیگار Winston می‌کشد، صاحب حلزون است. • کسی که سیگار Lucky strike می‌کشد، آب پرتقال می‌نوشد. • مرد اوکراینی چای می‌نوشد. • مرد ژاپنی، سیگار parliaments مصرف می‌کند. • سیگار kool بعد ازخانه ای مصرف می‌شود که درآنجا اسب نگهداری می‌کنند. • درخانه سبز رنگ، قهوه نوشیده می‌شود. • خانه سبز رنگ دقیقا سمت راست خانه شیری است. (ازدید شما) • درخانه وسط، شیر نوشیده می‌شود. رمورد نمایش‌های مختلف این مساله به صورت یک CSP بحث کنید. چرا ممکن است یک نمایش بر دیگری ارجح باشد؟ *(این مرین مشابه تمرین ۷.۶ در ویرایش سوم است)***

☑ حل: این معما که به «معمای گورخر»شهرت دارد می‌تواند به صورت یک مساله CSP تفسیر شود پس باید سه مجموعه متغیر، دامنه و محدودیت را در آن بیابیم. در اینجا به ازای هر یک از رنگ‌ها، حیوانات، نوشیدنی‌ها و نام سیگارها یک متغیر در نظر می‌گیریم که مجموعا ۲۵ متغیر تشکیل می‌شود. حال نوبت به تعیین دامنه متغیرها می‌شود. هرمتغیر دارای یکی از اعداد 1 تا 5 که نشان‌دهنده شماره خانه است می‌باشد. این شیوه نمایش مناسب است زیرا می‌توان تمام محدودیت‌های تعریف شده در مساله را به سادگی با این متغیرها تعریف نمود. (کدی به زبان پایتون این مساله را پیاده‌سازی کرده است و می‌توان در سایر زبان‌ها مجددا آن را پیاده‌سازی نمود). دلیل انتخاب این شیوه نمایش CPS نسبت به سایرین، کارایی این روش در یافتن راه‌حل می‌باشد در برخی اجراها، جستجوی محلی با حداقل تناقضات می‌تواند راه‌حل این مساله را در طی چند ثانیه بیابد ولی سایر نمایش‌های ممکن پس از چند دقیقه این کار را انجام می‌دهند. یکی از روش‌های دیگر برای نمایش این مساله آن است که 5 متغیر برای هر خانه تعریف کنیم. یک متغیر برای رنگ با دامنه {قرمز، سبز، شیری، زرد، آبی} متغیری برای حیوان با دامنه {گورخر، حلزون، روباه، اسب و سگ}، متغیری جهت نوشیدنی با دامنه {آب پرتقال، چای، قهوه، آب} ، متغیر دیگر جهت نام سیگارها {parliaments, chesterfield, Winston, Lucky strike, kool} و متغیر دیگر مکان خانه‌ها تعریف کنیم.

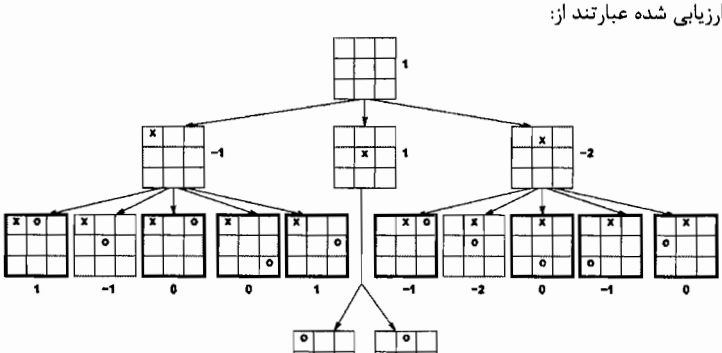
# فصل ۶ (ویرایش دوم)

**؟ ۱.۶ این مساله مفاهیم اصلی بازی را به کمک یک بازی نظیر دوز تمرین می‌کند (گذاشتن علامت‌های ضربدر و دایره در خانه‌ها). $X_n$ را تعداد سطرها، ستون‌ها یا قطرهایی تعریف می‌کنیم که دقیقا $n$ تا $X$ درآنها بکار رفته باشد و هیچ نماد $O$ای در آنها دیده نشود. همچنین $O_n$ را تعداد سطرها، ستون‌ها و قطرهایی می‌گیریم که فقط نماد $O$ داشته باشند. تابع سودمندی به هر موقعیتی که $X_3$=1 باشد مقدار 1+ و هر موقعیتی که $O_3$=1 باشد مقدار 1- را انتساب می‌دهد. سایرموقعیت‌های پایانی دارای سودمندی صفر هستند. برای موقعیت‌های غیرپایانی، از یک تابع‌ارزیاب خطی به این صورت استفاده می‌کنیم:**

$$Eval(s) = 3X_2(s)+X_1(s)-(3O_2(s)+ O_1(s))$$

**الف) تقریبا چند بازی دوز ممکن، وجود دارد؟ ب) درخت بازی را به طور کامل نمایش دهید. در ابتدا از صفحه‌ای خالی شروع کرده و سپس تا عمق دوم بازی را به طور متقارن در نظر بگیرید (به عنوان مثال یک $X$ و یک $O$ در صفحه موجود باشد). ج) ارزیاب تمام موقعیت‌ها در عمق 2 را بر روی درختی که ترسیم کردید، نمایش دهید. د) با استفاده از الگوریتم بیشینه-کمینه، مقادیر موقعیت‌ها را برای عمق 0 و 1 روی درخت مشخص کرده و با استفاده از آن مقادیر، بهترین حرکت آغازین را انتخاب کنید. هـ) گره‌هایی که در عمق 2 بوده و توسط هرس آلفا- بتا ارزیابی نمی‌شوند را با کشیدن یک دایره دور آنها مشخص کنید. با این فرض که گره‌ها برای هرس آلفا- بتا به‌ترتیب بهینه مرتب شده‌اند.** ***(این تمرین مشابه تمرین ۹.۵ در ویرایش سوم است)***

☑ حل: شکل ۱.۶s یک درخت بازی را نشان می‌دهد که مقدار تابع ارزیاب هر گره پایانی در زیر آن و مقدار پشتیبان هر گره غیرپایانی در سمت راست آن مشاهده می‌شود. بهترین حرکت آغازین برای $X$ را در مرکز می‌بینید. گره‌های پایانی که دوخطی شده‌اند، آنهایی هستند که با فرض مرتب‌سازی بهینه نیازی به ارزیابی ندارند. **ب)** درخت بازی را در شکل زیر مشاهده می‌کنید که برای برگهای درخت، اعداد سودمندی در زیر آنها درج گشته است و برای گره‌های غیربرگ اعداد بازگشت به عقب را در سمت راست آنها ثبت نموده‌ایم. اعداد نشان میدهند که بازیکن Max بهتر است در ابتدا نماد ضربدر خود را در مرکز صفحه قرار دهد. برگهایی که با خطوط پررنگ مشخص شده‌اند گره‌هایی هستند که با فرض ترتیب بهینه، نیازی به ارزیابی ندارند. موقعیتهای ممکن برای درخت ارزیابی شده عبارتند از:

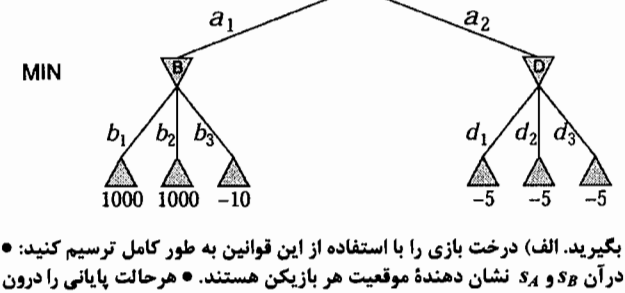


شکل ۱.۶s
قسمتی از درخت بازی دوز

**؟ ۲.۶ این ادعا را اثبات کنید: در هر درخت بازی، سودمندی حاصل از روش بیشینه-کمینه(MiniMax) برای بازیکن Max در صورتی که Min نیمه‌بهینه بازی کند، هرگز کمتر از حالتی نخواهد بود که Min بهینه بازی کند. آیا می‌توانید یک درخت بازی مثال بزنید که Max با یک استراتژی نیمه‌بهینه بتواند همچنان بهتر از Min با عملکرد نیمه‌بهینه بازی کند.** ***(این تمرین مشابه تمرین ۷.۵ در ویرایش سوم است)***

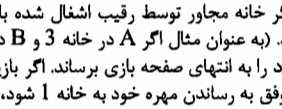
☑ حل: در ابتدا لازم به یادآوری است که در مسائل بازی معمولا فرض می‌شود که دو بازیکن با نام‌های Max و Min به صورت نوبت در میان بازی می‌کنند. این بازی می‌تواند هر نوع بازی مانند شطرنج، تخته‌نرد و. .. در نظر گرفته شود زیرا ما در اینجا می‌خواهیم کارگزاری طراحی کنیم که بتواند در انواع بازی‌ها به طور خودکار عمل کند مانند بازی شطرنج در کامپیوتر. در طی بازی، بازیکن Max باید سعی کند امتیاز بگیرد (امتیاز مثبت) و بازیکن Min باید از گرفتن امتیاز طرف مقابل جلوگیری کند بنابراین هر امتیازی که Max بگیرد منفی آن برای Min محسوب می‌شود (مجموع امتیازات دو طرف صفر خواهد بود). این گونه بازی‌ها تخاصمی نامیده می‌شود زیرا بازیکن Min با طرف مقابل خصومت دارد. معمولا دنباله حرکات مختلفی که در بازی میسر است و امتیاز هر حرکت را در قالب درختی با نام درخت بازی ترسیم می‌کنند که هر سطر آن نشان‌دهنده تمام حرکات مجاز برای یک بازیکن و امتیازات وی می‌باشد. منظور از بهینه بازی‌کردن Max آن است که در هر حرکت بهترین حرکت ممکن

برای کسب امتیاز را انجام دهد و منظور از بهینه بازی کردن Min آن است که در هر حرکت بهترین کاری که موجب بیشترین کسر امتیاز از Max شود را انجام دهد. در این سئوال فرض شده است که Max بازی خود را انجام می‌دهد ولی بازیکن Min خیلی با وی خصومت ندارد. می‌خواهیم اثبات کنیم که داشتن خصومت کامل و یا خصومت کمتر او، هیچ تاثیری در امتیاز بازیکن Max ندارد. برای این منظور یک گرهٔ Min که فرزندان آن گره‌های پایانی درخت هستند را در نظر بگیرید. اگر Min به صورت نیمه‌بهینه بازی کند یعنی حرکتی انجام دهد که باعث شود بازیکن Max بیشتر از حداقل مقدار ممکن امتیاز بگیرد، آنگاه مقدار آن گره بزرگتر و مساوی با مقداری خواهد بود که Min به صورت بهینه بازی کند. از این رو، مقدار گرهٔ Max که پدرگرهٔ Min می‌باشد، نیز طبق الگوریتم بیشینه-کمینه افزایش می‌یابد. این روال به همین ترتیب تا ریشه ادامه می‌یابد و در ریشه مقداری بیشتر و یا مساوی با بهینه بازی‌کردن Min خواهیم داشت. اگر بازی نیمه‌بهینهٔ Min قابل پیش‌بینی باشد، آنگاه می‌تواند بهتر از یک استراتژی بیشینه-کمینه بازی کند. به عنوان مثال اگر Min همواره به دنبال نوع خاصی از تله و نیرنگ باشد آنگاه تنظیم کردن تله، برنده‌شدن را تضمین می‌کند حتی اگر پاسخ اشتباه به Min بدهیم. این موضوع در شکل ۲.۶s نمایش داده شده است.




شکل ۲.۶s یک درخت بازی ساده که یک تله برای بازیکن MIN را نشان می‌دهد که درصورتی که این بازیکن حرکت $a_i$ را انتخاب کند، در آن می‌افتد. البته حرکت بیشینه-کمینه نیز $a_2$ با مقدار 5- می‌باشد.

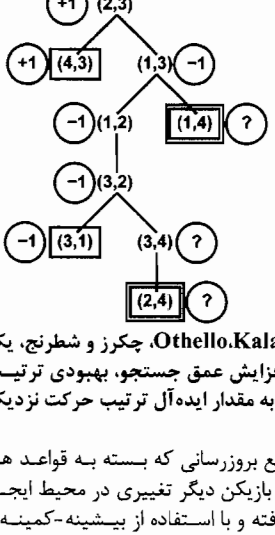
**؟ ۳.۶ بازی دو نفره در شکل ۱۴.۶ را در نظر بگیرید. الف) درخت بازی را با استفاده از این قوانین به طور کامل ترسیم کنید: • هر حالت را به صورت $(s_A,s_B)$ تعریف کنید که درآن $s_B$ و $s_A$ نشان دهندهٔ موقعیت هر بازیکن هستند. • هرحالت پایانی را درون یک مربع و مقادیر بازی را درون یک دایره قرار دهید. • حالات حلقه‌ای (حالاتی که در مسیر پیموده شده از ریشه قبلا استفاده شده‌اند) را درون یک مربع دوخطی قرار دهید. اگر مقدار آنها هنوز مشخص نشده است، با گذاشتن «?» درون دایره آنها را مشخص نمایید. ب) اکنون برای هرگره مقدار بیشینه-کمینه را بنویسید (هم چنین در دایره‌ها). توضیح دهید که چرا و چگونه از علامت «?» استفاده می‌کنید؟ ج) توضیح دهید که چرا الگوریتم استاندارد بیشینه-کمینه، در مورد این درخت بازی شکست می‌خورد و به طور خلاصه توضیح دهید که چگونه می‌توان با استفاده از قسمت ب، آن را اصلاح نمود. آیا الگوریتم تغییر یافته شما، می‌تواند تصمیم بهینه را برای تمام بازی‌های دارای حلقه بیابد؟ د) این بازی 4- مربعی را می‌توان به حالت کلی $n$- مربعی تعمیم داد. یعنی خانه‌های $n>2$ اثبات کنیدکه اگر $n$ زوج باشد بازیکن A می‌برد و اگر $n$ فرد باشد، می‌بازد. *(این تمرین مشابه تمرین ۸.۵ در ویرایش سوم است)***




شکل ۱۴.۶ موقعیت شروع یک بازی ساده. اول بازیکن A بازی می‌کند. دو بازیکن به طور نوبت چرخشی باز کرده و هر یک باید مهره خود را به فضای مجاور خالی در یکی از جهات منتقل کند. اگر خانه مجاور توسط رقیب اشغال شده باشد، طرف مقابل باید مهره خود را با پرش از روی رقیب به خانه بعدی ببرد البته اگر آنجا خالی باشد. (به عنوان مثال اگر A در خانه 3 و B در خانه 2 باشد، آنگاه A باید به خانه 1 برود). بازی زمانی خاتمه می‌یابد که یکی از بازیکنان مهره خود را به انتهای صفحه بازی برساند. اگر بازیکن A موفق به رساندن مهره خود به خانه 4 شود، بازیکن A، 1+ امتیاز می‌گیرد. و اگر بازیکن B موفق به رساندن مهره خود به خانه 1 شود، بازیکن A، 1- امتیاز می‌گیرد.

☑ حل: **الف)** درخت این بازی به همراه تمام نمادها و مقادیر بیشینه-کمینه در شکل ۳.۶s نمایش داده شده است. لازم بذکر است که در الگوریتم بیشینه-کمینه از برگ‌ها به سمت ریشه حرکت کرده و در هر بار برای هر گره پدر با توجه به سودمندی فرزندانش باید عددی را به آن اختصاص دهیم. اگر آن گره پدر نوبت بازی Min باشد، بازیکن Min همیشه سعی می‌کند تا کمترین امتیاز حاصل شود پس مسلم است که اگر در آن موقعیت بخواهد از بین فرزندان انتخاب کند ،کمترین سودمندی را بر می‌گزیند. پس کمترین سودمندی از بین فرزندان انتخاب شده و برای آن گره به عنوان مقدار بیشینه-کمینه نوشته می‌شود. اگر گره پدر در درخت بازی نوبت Max باشد از بین فرزندان بیشترین مقدار سودمندی را انتخاب خواهد نمود. پس آن را به عنوان مقدار بیشینه-کمینه در کنار گره یادداشت می‌کنیم. این کار به همین ترتیب تا ریشه ادامه می‌یابد و در کنار هر گره با توجه به نوبت بازی مقدارهای بیشینه-کمینه یادداشت می‌شود. سپس با حرکت از ریشه و با کمک مقادیر بیشینه-کمینه می‌توان به ترتیب بهترین انتخاب‌ها را در بازی انجام داد تا به هدف برسیم. **ب)** مقدار «?» با این فرض است که یک کارگزار از بین حالت برنده و حالت «?» همواره حالت برنده را انتخاب می‌کند. همچنین Min(-1,?)=-1 و Max(+1,?)=+1 . در صورتی که تمام حسین‌های یک حالت «?» باشند، آنگاه مقدار مورد نظر همان «?» خواهد شد. **ج)** روش استاندارد بیشینه-کمینه بر مبنای الگوریتم اول‌عمق است بنابراین ممکن است دریک حلقه بی‌نهایت گیر کند. از این رو جهت اصلاح آن می‌توان حالت فعلی را با

پشته حالات مقایسه نمود و در صورت تکراری‌بودن حالت، مقدار «?» را بازگرداند. انتشار «?» به همین شیوه صورت می‌گیرد. اگر چه از این علامت در اینجا استفاده کردیم ولی معمولا این کار انجام نمی‌شود زیرا مشخص نیست که نتیجهٔ مقایسهٔ «?» با یک موقعیت کنار گذاشته چیست و یا نمی‌دانیم که مقایسهٔ بین حالات برندهٔ متفاوت (مانند بازی تخته‌نرد) چگونه انجام شود. در نهایت، در بازی با گره‌های شانسی، چگونگی محاسبهٔ میانگین اعداد و «?» مشخص نیست. توجه داشته باشید که نباید با حالات تکراری به عنوان موقعیت‌های کنارگذاشته شده رفتار گردد. به عنوان مثال هر دو حالت (1,4) و (2,4) در درخت تکراری هستند ولی موقعیت‌های برنده محسوب می‌شوند. اگر درخت بازی بدون چرخه باشد، آنگاه الگوریتم بیشینه- کمینه می‌تواند این معادلات را با انتشار از برگ‌ها حل نماید. اگر درخت بازی دارای چرخه باشد، آنگاه بایستی روش‌های برنامه نویسی پویا مورد استفاده قرار گیرد. (تمرین ۱۷.۸ قسمتی از این مساله را بررسی می‌کند). این الگوریتم‌ها می‌توانند به طور دقیق تعیین کنند که کدام گره یک مقدار خوش تعریف دارد (مانند این مثال) و یا یک حلقه بی‌نهایت است که هر دو بازیکن تمایل به ماندن در آن چرخه را دارند (یا شاید انتخاب دیگری ندارند). دراین مورد، بایستی قواعدی در بازی تعیین کرد تا مقدار را مشخص کند (در غیراین صورت بازی هرگز خاتمه نمی‌یابد). در بازی شطرنج به عنوان مثال، حالتی که سه بار اتفاق بیافتد (بنابراین بایستی برای هر دو بازیکن منع شود) کنار گذاشته می‌شود. **د)** این سوال اندکی مهارت‌آمیز طراحی شده است. یک روش اثبات آن، استقرار بر روی اندازهٔ بازی است. بدیهی است که در حالت پایه بازی یعنی $n=3$ بازیکن $a$ می‌بازد و برای $n=4$ بازیکن A می‌برد. برای تمام $n>4$ حرکات آغازین یکسان است: A و B هر کدام به اندازهٔ یک واحد جلو می‌روند. اکنون می‌توان فرض کرد که قسمتی از بازی با اندازهٔ $n-2$ در مربع‌هایی با شماره‌های $[2,\ldots,n-1]$ شکل گرفته است با این تفاوت که در این زیربازی هر بازیکن در خانه‌های 2، $n-1$ یک حرکت دیگر نیز می‌تواند انتخاب کند. در ابتدای اثبات فرض کنید از این حرکات اضافه صرف‌نظر می‌کنیم. مشخص است که اگر A بخواهد با رسیدن به خانهٔ $n-2$ برنده محسوب شود، بایستی A قبل از رسیدن B به خانهٔ 2، به خانهٔ $n-1$ برسد (طبق تعریف برنده شدن) . بنابراین قبل از رسیدن B به خانهٔ یک، بازیکن A به خانهٔ $n$ می‌رسد بنابراین مقدار $n$ منجر به برنده شدن A گشت. طبق همین اثبات اگر $n-2$ برای B برنده محسوب شود آن گاه $n$ نیز برای B برنده خواهد بود. اکنون فرض کنید حرکات اضافه گفته شده نیز مجاز باشند، بازیکنی که قصد برنده شدن در فضای $[2,\ldots,n-1]$ دارد، هرگز به خانهٔ آغازین خود باز نمی‌گردد ولی اگر بازیکن قصد باخت داشته باشد این کار را انجام می‌دهد. این موضوع به خودی خود، باخت بازی را تخمین می‌کند. سایر بازیکنان به سادگی به جلو حرکت می‌کنند و یک زیربازی با اندازهٔ $n-2k$ مرحله به مرحله به خانهٔ آغازین بازنده نزدیک‌تر می‌شود.




شکل ۳.۶s درخت بازی برای بازی چهارخانه که در تمرین ۳.۶ بیان شد. حالات پایانی درون مربع‌های یک خطی و حالات حلقوی درون مربع‌های دوخطی ترسیم شده‌اند. برای هر حالت مقدار بیشینه-کمینه آن را درون دایره‌ای درکنار آن مشخص کرده‌ایم.

**؟ ۴.۶ تابع ارزیاب و مولد حرکت را برای یک یا چند بازی ذکر شده پیاده‌سازی کنید: Othello،Kalah، چکرز و شطرنج، یک کارگزار عمومی برای آلفا- بتا بیابید که از پیاده‌سازی شما استفاده کند. تاثیر حاصل از افزایش عمق جستجو، بهبودی ترتیب حرکت و ارتقاء تابع‌ارزیاب را با یکدیگر مقایسه کنید. ضریب‌انشعاب موثر شما تا چه حد به مقدار ایده‌آل ترتیب حرکت نزدیک است؟** ***(این تمرین اندکی مشابه تمرین ۱۱.۵ در ویرایش سوم است)***

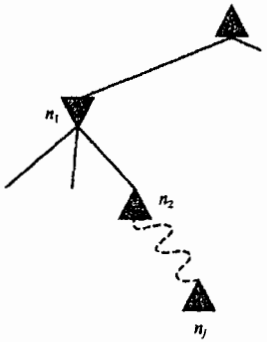
☑ حل: توجه داشته باشید که یک محیط بازی اساسا یک محیط عمومی است که یک تابع بروزرسانی که بسته به قواعد هر بازی تعریف می‌شود، به آن اضافه شده است. و تا زمانی که نوبت حرکت یک بازیکن است بازیکن دیگر تغییری در محیط ایجاد نمی‌کند. می‌توانید با کمی جستجو، کدی برای تعریف یک بازی ساده (سخت‌تر از دوز) یافته و با استفاده از بیشینه-کمینه و آلفا- بتا نسبت به حل نسخه‌های کوچک این بازی اقدام کنید (احتمالا تا 3×4). البته توجه داشته باشید که روش آلفا- بتا بسیار سریع‌تر از بیشینه‌کمینه اقدام می‌کند ولی هنوز نمی‌توان بدون داشتن یک تابع ارزیاب و برش افقی آن را به مقیاس‌های بزرگتر تعمیم داد. ایجاد یک تابع ارزیاب تمرینی جذاب خواهد بود. از دید طراحی ساختمان‌داده این موضوع نیز جذاب است که با استفاده از پیش محاسبه سطرها، ستون‌ها و قطرها، چگونه می‌توان مولد حرکت را سرعت بخشید. تعداد کمی از دانشجویان در بازی Kalah ماهر هستند بنابراین انتسابی منصفانه داریم ولی این بازی کسل‌کننده است. عمق 6 به نظر مناسب است و یک

تابع ارزیاب مناسب برای شکست بیشتر انسان‌ها کافی است. بازی اتلو جذاب بوده و سطح دشواری آن برای بیشتر دانشجویان مناسب است. بازی شطرنج و چکرز در بیشتر مواقع منصفانه صورت نمی‌گیرد چون ممکن است نیمی از دانشجویان کلاس ماهر بوده و نیمی دیگر مبتدی.

**؟ ۵.۶ اثباتی رسمی برای صحت عملکرد هرس آلفا- بتا ارائه دهید. برای این منظور موقعیت نشان داده شده در شکل ۱۵.۶ را در نظر بگیرید. اکنون سوال این است که آیا می‌توان گرهٔ $n_j$ که گره‌ای بیشینه بوده و فرزند گرهٔ $n_1$ است را هرس نمود. ایده اصلی این کار آن است که فقط در صورتی می‌توان هرس نمود که مقدار بیشینه-کمینهٔ آن مستقل از مقدار $n_j$ باشد. الف) مقدار $n_1$ که برحسب مقدار بیشینه-کمینه فرزندش حاصل می‌شود عبارت است از:**

$$n_1 = \min(n_2, n_{21}, \ldots, n_{2b_2})$$

**عبارتی مشابه برای $n_2$ و سپس عبارتی برای $n_1$ برحسب $n_j$ بیابید. ب) فرض کنید $l_i$ برای مقدار بیشینه (یا کمینه) گره‌های سمت چپ گرهٔ $n_i$ درعمق $i$ تعریف شود که مقدار بیشینه-کمینه آنها مشخص باشد. به طور مشابه فرض کنید $r_i$ برای مقدار کمینه (یا بیشینه) گره‌های بسط نیافته سمت راست گرهٔ $n_i$ در عمق $i$ تعریف شود. مجددا عبارت $n_1$ را بازنویسی کرده و از عبارات $l_i$ و $r_i$ استفاده کنید. ج) عبارت را مجددا فرموله کنید به طوری که نشان دهد برای تاثیرگذاری روی $n_1$، $n_j$ نباید از یک حد مشخص که از روی مقادیر $l_i$ به دست می‌آید، تجاوز کند. د) این مراحل را در حالتی که $n_j$ یک گرهٔ مینیمم باشد، تکرار کنید.** ***(این تمرین مشابه تمرین ۱۳.۵ در ویرایش سوم است)***




شکل ۱۵.۶ موقعیت مربوط به در نظر گرفتن هرس گره $n_j$

☑ حل: این تمرین به دشواری آنچه به نظر می‌رسد، نیست. عبارات زیر دقیقا منجر به تعاریف آلفا و بتا می‌شوند. نماد $n_i$ اشاره به مقدار گره‌ای در عمق $i$ دارد که در مسیری از ریشه تا برگ $n_j$ قرار دارد. گره‌های $n_{ib_i},\ldots,n_{i1}$ همگی همزاد (برادر)های گرهٔ $i$ هستند. **الف)** می‌توان تعریف کرد: $n_2 = \max(n_3, n_{31}, \ldots, n_{3b_3})$

داریم: $n_1 = \min(\max(n_3, n_{31}, \ldots, n_{3b_3}), n_{21}, \ldots, n_{2b_2})$

بنابراین زمانی که عبارتی شامل خود $n_j$ داشته باشیم $n_3$ می‌تواند جایگزین شود. **ب)** طبق اصطلاحات $r$ و $l$ داریم:

$$n_1 = \min(l_2, \max(l_3, n_3, r_3), r_2)$$

مجددا $n_3$ می‌تواند بسط یابد تا به $n_j$ برسد. داخلی‌ترین اصطلاح به صورت $\min(l_j,n_j,r_j)$ خواهد بود. **ج)** اگر $n_j$ یک گرهٔ بیشینه باشد، آنگاه حد پایینی بر روی مقدارش فقط در صورتی افزایش می‌یابد که پسین‌های آن ارزیابی شوند. واضح است که اگر این مقدار از $l_j$ تجاوز کند آنگاه هیچ تاثیری روی $n_1$ نخواهد داشت. در حالت گسترده‌تر اگر این مقدار از $\min(l_2,l_4,\ldots,l_j)$ تجاوز کند آنگاه دیگر هیچ تاثیری ندارد. بنابراین با نگهداری دنباله مقادیر آن می‌توان تصمیم گرفت که چه زمان $n_j$ را هرس نماییم. این مفهوم دقیقا همان عملکرد آلفا- بتا را نشان می‌دهد. **د)** حد مربوطه برای کمینه گره‌های $n_k$ عبارتست از:

$$\max(l_3,l_5,\ldots,l_k)$$

**؟ ۶.۶ الگوریتم‌های بیشینه-کمینه متوسط و آلفا- بتای‌ستاره‌دار که در Ballard(1983) بیان شد را جهت هرس درختان بازی با گره شانس پیاده‌سازی نمایید. سپس آنها را در بازی‌هایی چون تخته‌نرد امتحان کرده و کارایی هرس در روش آلفا- بتای ستاره‌دار را اندازه بگیرید.** ***(این تمرین مشابه تمرین ۱۷.۵ در ویرایش سوم است)***

☑ حل: پاسخ برعهده خواننده.

**؟ ۷.۶ ثابت کنید که توسط یک تبدیل خطی مثبت بر روی مقادیر برگ‌ها یعنی تبدیل یک مقدار $x$ به مقدار $ax+b$ که $a>0$ باشد در انتخاب حرکت از درخت بازی هیچ تاثیری نمی‌گذارد، حتی اگر این درخت شامل گره شانس نیز باشد.** ***(این تمرین مشابه تمرین ۱۸.۵ در ویرایش سوم است)***

☑ حل: استراتژی کلی آن است که با استفاده از استقراء روی عمق درخت، یک درخت بازی عمومی را به درختی تک‌لایه کاهش دهیم. مرحلهٔ استقراء را بایستی برای گره‌های Max، Min و شانس انجام داده و سپس نشان دهیم که تبدیل مربوطه بر روی این گره‌ها انجام می‌شود. فرض کنید که مقدار فرزندان یک گره $x_1,x_2,\ldots,x_n$ باشند و تبدیل را به صورت $ax+b$ که $a>0$ است تعریف کنیم. آنگاه داریم:

$$\begin{aligned}\min(ax_1+b, ax_2+b, \ldots, ax_n+b) &= a\min(x_1,x_2,\ldots,x_n)+b\\ \max(ax_1+b, ax_2+b, \ldots, ax_n+b) &= a\min(x_1,x_2,\ldots,x_n)+b\end{aligned}$$

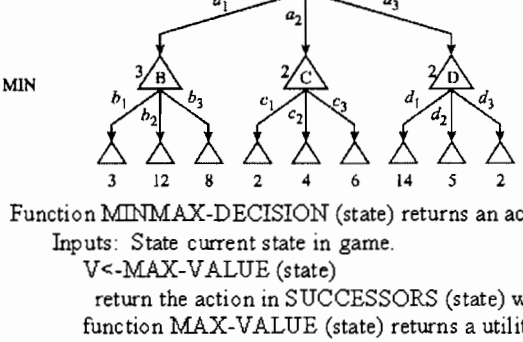
بنابراین کاهش درخت به درختی تک‌لایه که برگ‌های آن از مقادیر درخت اصلی برداشت شده باشند، تبدیلی خطی را نشان می‌دهد. بنابراین طبق $a>0 \Rightarrow ax+b>ay+b$ اگر $x>y$ بهترین انتخاب در ریشه با بهترین انتخاب در درخت اصلی یکسان خواهد بود.

**؟ ۸.۶ این مراحل را برای انتخاب حرکت در یک بازی دارای شانس در نظر بگیرید: • پرتاب‌های متوالی از پرتاب تاس (مثلا 50 بار) را تا رسیدن به عمقی خاص (مثلا 8) تولید کنید. • با دانستن اعداد روی هر تاس در هر مرتبه، درخت بازی قطعی می‌شود. به ازای هر دنباله از پرتاب‌های تاس، درخت بازی قطعی را به کمک آلفا- بتا حل کنید. • با استفاده از نتیجهٔ بدست آمده، مقدار هر حرکت را تخمین زده و بهترین آنها را انتخاب کنید. آیا این رویه به درستی کار می‌کند؟ چرا بله و چرا خیر؟** ***(این تمرین مشابه تمرین ۱۹.۵ در ویرایش سوم است)***

☑ حل: این رویه، نتایج نادرستی درپی‌دارد. به طور ریاضی با تشکیل دنباله پرتاب‌های تاس و تصمیم‌گیری بر مبنای آن، در واقع مقدار گره را بر اساس میانگین Min و Max می‌یابیم که در عمل این شیوه ممکن نخواهد بود. به طور شهودی این کار بدان معناست که انتخاب هر بازیکن در یک درخت قطعی معادل با داشتن دانشی کامل از آن چیزی است که تاس در آینده نشان خواهد داد. بنابراین نیازی نیست تا بازیکن حرکت خود را بر مبنای محیطی بدون دانش انتخاب کند زیرا آینده بازی را به طور کامل می‌داند. (به رابطهٔ بین مباحث بازی با کارت در بخش ۲.۶.۵ توجه کنید). البته این روش بازی در عمل نتایج منطقی خوبی داشته و می‌تواند جهت مقایسه با بازی‌هایی که مجموع صفر دارند، بکار رود.

**؟ ۹.۶ یک محیط بازی چندلایه و بی‌درنگ پیاده‌سازی و تشریح کنید که زمان قسمتی از حالت محیط بوده و بازیکنان زمان محدودی دراختیار دارند.** ***(این تمرین مشابه تمرین ۵.۵ در ویرایش سوم است)***

☑ حل: دو مرحله از یک بازی بی‌درنگ دو نفره را درنظر بگیرید. بازیکن max بدنبال کسب بیشترین امتیاز و بازیکن min بدنبال کسب کمترین امتیاز هستند.




۱. از موقعیت جاری در گره max شروع کنید.
۲. درخت بازی را با تعداد ثابتی از برنامه‌ها گسترش دهید.
۳. تابع ارزیاب را به گره‌های برگ اعمال نمایید.
۴. از پایین به بالا، مقادیر بازگشتی را محاسبه کنید.
۵. حرکتی را انتخاب کنید که در ریشه بیشترین مقدار بدست آید.

الگوریتم:

```
Function MINMAX-DECISION (state) returns an action.
  Inputs: State current state in game.
    V<-MAX-VALUE (state)
     return the action in SUCCESSORS (state) with value v.
    function MAX-VALUE (state) returns a utility value.
    If TERMINAL-TEST(State) then return UTILITY(State)
    V ← -∞
    for a, s in SUCCESSORS(State) do
     V← MAX(v, MIN-VALUE(s))
    return v
   function MIN-VALUE(State) returns a utility value.
    if TERMINAL-TEST(state) then returnUTILITY(state)
      V ← ∞
    for a, s in SUCCESSORS(state) do
      V←MIN(v,MAX-VALUES))
 Return s,
```

**؟ ۱۰.۶ توصیف حالات، مولد حرکت، آزمون پایانی، تابع سودمندی و تابع ارزیاب را برای یک یا چند مورد از بازی‌های ذکر شده توضیح داده و آنها را پیاده‌سازی نمایید: مونوپولی، اسکرابل، بازی بریج با قراردادی مشخص، بازی پوکر.** ***(این تمرین مشابه تمرین ۴.۵ در ویرایش سوم است)***

☑ حل: هرحالت نشان‌دهنده وضعیت فعلی بازی (مانند حالت دنیا در مسائل فصول قبل) می‌باشد که می‌توانند به روش‌های مختلف فرمولی و یا ترسیمی توصیف شود که پس از انجام حرکت هر نفر، به حالت دیگری از بازی می‌رسیم. به عنوان مثال حالت در برخی بازی‌ها عبارتست از: در بازی شطرنج چیدمان مهره‌ها در صفحه، در پوکر که نوعی ورق‌بازی می‌باشد انواع پنج کارت موجود در دست بازیکنان. حالت فیزیکی پایه در این بازی‌ها بسیار ساده است. یکی از نکات مهم که لازم به یاد آوری است آن است که در بازی‌های بریج و اسکرابل، حالت فیزیکی برای تمام بازیکنان در دسترس نیست زیرا هر بازیکن نمی‌تواند تمام کارت‌ها را ببیند و شبیه‌ساز محیط نمی‌تواند به طور مستقیم به ازای هر بازیکن به آن دسترسی داشته باشد. در بازی بریج، هر

بازیکن حدس‌ها و فرضیه‌هایی نسبت به حالت واقعی محیط داشته و بهترین فرضیه را نگهداری می‌کند. با توجه به آنکه این بازی به طور آنلاین در بسیاری از سایت‌ها وجود دارد به احتمال زیاد کد مربوط به پیاده‌سازی آن نیز به آسانی یافته شود. منظور از آزمون پایانی شرایطی است که باید توسط بازیکنان ایجاد شود تا بازی خاتمه یابد که در بازی‌های مختلف متفاوت است. به عنوان مثال در شطرنج مات‌کردن شاه طرف مقابل و در بازی پوکر بیشترین مبلغی که بازیکنان شرط می‌بندند مانند مناقصه. تابع سودمندی روش تعیین امتیاز به هر حالت پایانی بازی است. به عنوان مثال در پایان بازی پوکر که بازیکنان کارت‌های خود را رو می‌کنند اگر کسی پنج کارت همرنگ داشته باشد نسبت به کسی که پنج کارت همرنگ و مرتب داشته باشد امتیاز کمتری می‌گیرد. در این بازی شیوه امتیازدهی در قالب جدولی از انواع چیدمان کارت‌ها و امتیاز آنها تعیین می‌شود. تابع ارزیاب به هر حالت غیرپایانی بازی، عددی را انتساب می‌دهد که نشان‌دهنده تخمینی از متوسط سودمندی آن بوده و برای گره‌های پایانی این تابع همان سودمندی را نشان می‌دهد (مشابه تابع ارزیاب فصول قبل). این تابع به بازیکنان کمک می‌کند تا حالت فعلی بازی را ارزیابی کرده و بدانند که این حالت به نفع کدام بازیکن است. در بازی شطرنج می‌توان تابع ارزیابی به این صورت در نظر گرفت که با توجه به مهره‌ها تعیین می‌شود: هر سرباز یک امتیاز، هر اسب یا فیل دو امتیاز و هر وزیر نه امتیاز. همچنین متدهایی تعریف کنیم که امنیت شاه نیز مقدار داشته باشد، ترکیب خطی این موارد یک عدد می‌شود که حالت فعلی بازی را تعیین می‌کند. مشخص است که اگر بازیکن سفید در صفحه شطرنج وزیر، دو فیل و دو اسب داشته و رقیبش تنها 8 سرباز را در صفحه داشته باشد، این وضعیت فعلی به نفع سفید خواهد بود پس تابع ارزیاب باید عددی را نشان دهد که این موضوع را اثبات کند. در اینجا این تابع ارزیاب برای بازیکن سفید عدد 17 و برای بازیکن سیاه عدد 8 را نشان می‌دهد. این عدد موقعیت بازی را برای دو طرف روشن می‌کند.

**؟ ۱۱.۶ تاثیر شانس و اطلاعات ناقص را در هر یک از بازی‌های تمرین ۱۰.۶ بدقت بررسی کنید. الف) برای کدام بازی، مدل استاندارد بیشینه-کمینه میانگین مناسب است؟ الگوریتم آن به همراه تغییرات موردنیاز برای محیط بازی را پیاده‌سازی کرده و بر روی کارگزار بازی اجرا کنید. ب) برای کدام بازی روش گفته شده درتمرین ۸.۶ مناسب است؟ ج) توضیح دهید که چگونه می‌توان با این موضوع که در برخی بازی‌ها بازیکن هیچ اطلاعی از حالت فعلی بازی ندارد برخورد نمود.** ***(این تمرین مشابه تمرین ۲۲.۵ در ویرایش سوم است)***

☑ حل: از رخدادهای شانسی در یک بازی می‌توان به پرتاب تاس اشاره نمود که نمی‌توان نتیجه نهایی را از بین اطلاعاتی از پیش‌مرتب‌شده مشخص نمود (مانند کارت‌های بدست آمده در یک دست از ورق‌بازی). نکته مهم آن است که بازیکنان بایستی بین اطلاعات آشکار بازی برای دو طرف، و اطلاعاتی که برای هر بازیکن متفاوت است تفاوت قائل شوند. **الف)** روش بیشینه-کمینه میانگین فقط برای بازی‌هایی مانند تخته‌نرد و مونوپولی مناسب است. در بازی بریج و اسکرابل، هر بازیکن از کارت‌های خود آگاه است ولی از کارت‌های رقیب خیر. در اسکرابل کار عقلانی آن است که به طور تصادفی درمورد وضعیت رقیب استنتاج اندکی کرده و حرکت کنیم ولی در بازی بریج دانستن اطلاعات رقیب در یک بازی خوب سهم اساسی دارد و بایستی به طور کامل انجام شود. **ب)** خیر به خاطر دلیلی که اخیرا ذکر شد. **ج)** نکتهٔ کلیدی در مورد استنتاج وضعیت رقیب آن است که تعدادی باور برای رقیب در نظر گرفته و بسته به تاثیر اقدامات مختلف، باورها را بروزرسانی کرده و با استفاده از یک شیوهٔ نمایش، این حالات‌باور را نمایش دهیم. از آن جا که این حالات‌باور تمام حالات فیزیکی ممکن برای کارگزار عقلانی را دربرمی‌گیرند این کار مفید خواهد بود.

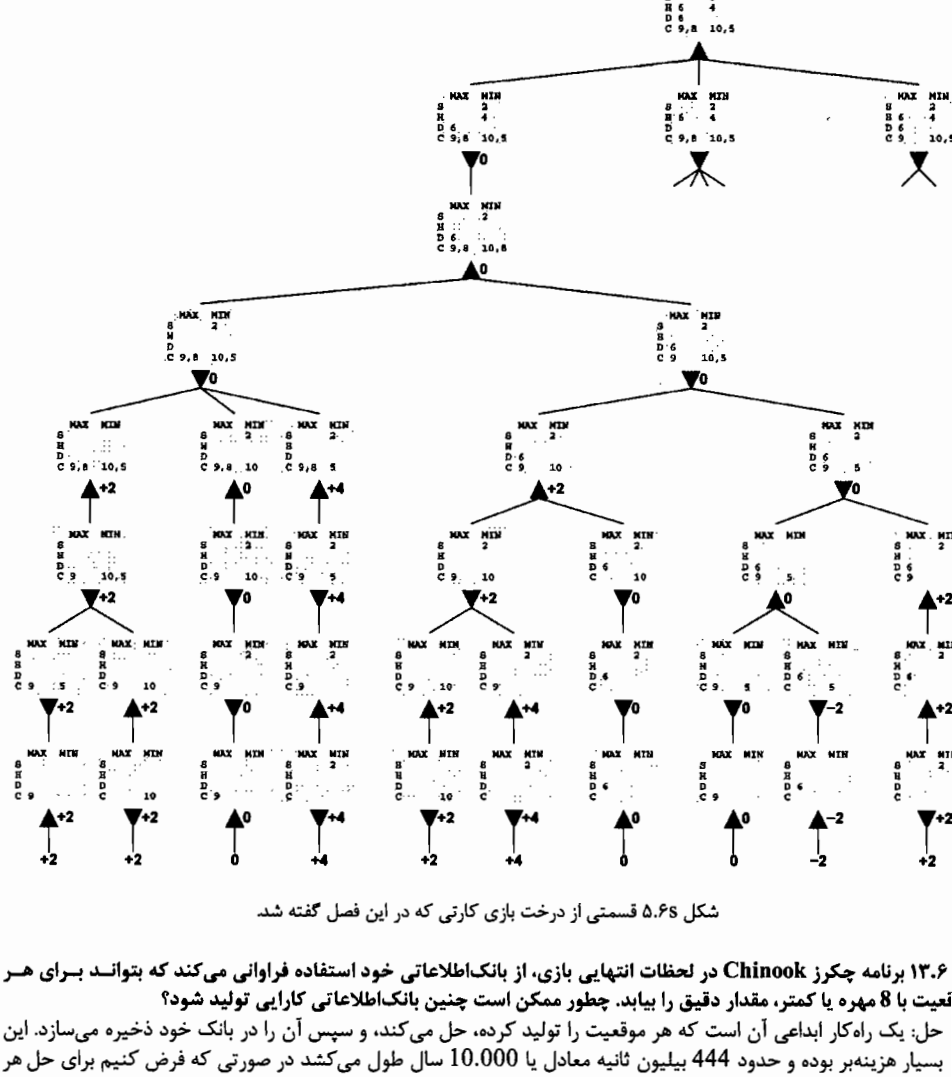
**؟ ۱۲.۶ در الگوریتم minimax فرض می‌شود که نوبت بازیکنان چرخشی باشد ولی در برخی ورق‌بازی‌ها مانند بازی bridge و whist برنده دور قبل آغاز کننده دور بعدی بازی خواهد بود. الف) الگوریتم را به طوری تغییر دهید که برای این بازی‌ها نیز کار کند، می‌توانید تابعی بنام WINNER($s$) تعریف کنید که برنده دور قبل را در صورت وجود معرفی می‌کند. ب) درخت بازی را برای اولین دست بازی که در بخش ۵.۶ آورده شده است ترسیم کنید.**

☑ حل: (در چاپ‌های اولیه این تمرین از تابع WINNER(trick) نام برده و در اینجا از WINNER($s$) که همان برنده دور قبل در صورت وجود می‌باشد). این تمرین در یک محیط قابل‌مشاهده بحث می‌شود. **الف)** تغییرات مقدار MAX را در شکل زیر می‌بینید، اگر بازیکن MAX دور قبلی را برده باشد، دوباره بایستی MAX بازی کند وگرنه نوبت بازیکن دیگر می‌رسد. بنابراین پسین یک گره MAX ترکیبی از گره‌های MAX , MIN خواهد بود که نتیجه بستگی به کارت بازی MAX دارد. همین تغییرات بایستی برای مقادیر MIN نیز اعمال شود.

```
function MAX-VALUE(state) returns a utility value
  if TERMINAL-TEST(state) then return UTILITY(state)
  v ← −∞
  for a, s in SUCCESSORS(state) do
    if WINNER(s) = MAX
        then v ← MAX(v, MAX-VALUE(s))
        else v ← MAX(v, MIN-VALUE(s))
  return v
```

شکل ۴.۶s قسمتی از الگوریتم تغییریافته بیشینه-کمینه که در آن برنده دور قبل، آغازکننده دور بعدی بازی خواهد بود.

ه) درخت بازی را در شکل زیر می‌بینید.



شکل ۵.۶۸ قسمتی از درخت بازی کارتی که در این فصل گفته شد.

**۱۳.۶ برنامه چکرز Chinook در لحظات انتهایی بازی، از بانک‌اطلاعاتی خود استفاده فراوانی می‌کند که بتواند برای هر موقعیت با 8 مهره یا کمتر، مقدار دقیق را بیابد. چطور ممکن است چنین بانک‌اطلاعاتی کارایی تولید شود؟**

حل: یک راه‌کار ابداعی آن است که هر موقعیت را تولید کرده، حل می‌کند، و سپس آن را در بانک خود ذخیره می‌سازد. این بسیار هزینه‌بر بوده و حدود 444 بیلیون ثانیه معادل یا 10.000 سال طول می‌کشد در صورتی که فرض کنیم برای حل هر قعیت به طور میانگین یک ثانیه زمان صرف شود (که بسیار بهینه فرض کردیم و در عمل اینگونه نیست). البته می‌توان برای هر موقعیت جدید، از بخشی از موقعیت‌های حل شده و شبیه به هم استفاده نمود. برای اطمینان از این موضوع، درابتدا عیت‌های انتهایی بازی را تولید کرده و حل می‌کنیم. سپس موقعیت‌های قبل آن را و به همین ترتیب ادامه می‌دهیم. دراین وه، مقدار دقیق تمام پسین‌ها را در زمان تولید هر حالت می‌دانیم. به این شیوه تحلیل پس‌رو می‌گویند.

**۱۴.۶ توضیح دهید که چگونه می‌توان روش استاندارد بازی‌کردن را به بازی‌هایی چون تنیس، بیلیارد و کریکت که در یک های حالت فیزیکی پیوسته انجام می‌شوند، اعمال نمود.** ***(این تمرین مشابه تمرین ۶.۵ در ویرایش سوم است)***

حل: یکی از واضح‌ترین تفاوت‌هایی که این بازی‌ها دارند، پیوستگی فضای اقدامات است. به عنوان مثال در بازی بیلیارد؛ ت چوب بیلیارد، زاویه، سرعت و نقطه تماس آن با توپ بیلیارد همگی عناصری پیوسته‌اند که در نتیجه بازی دخیل هستند. ه‌ترین راه‌حل آن است که این فضای پیوسته را گسسته نموده و سپس روش‌های بازی استاندارد را به آن اعمال نماییم یعنی ی هر عنصر محدوده مقادیر را تعیین نموده و سپس آن محدوده را به چندین زیرمحدوده تقسیم‌بندی نماییم و تعریف کنیم اگر آن عنصر مقداری در این محدوده داشت، عددی ثابت را به آن تخصیص دهیم. این راه‌کار شاید برای بازی تنیس موثر

باشد (بازی را فقط به عنوان شوت‌هایی دارای سرعت و جهت مدل نماییم) ولی برای بازی‌هایی نظیر بیلیارد و کراکت موثر نبوده و منجر به شکست می‌شود زیرا کوچکترین تغییر در زاویه، تاثیر بسیار زیادی در واکنش موردنیاز خواهد داشت. به عنوان مثال نمی‌توان تعریف کرد که اگر زاویه بین صفر تا 30 درجه بود فلان حرکت انجام شود. ولی در عوض با تحلیل بازی می‌توان مجموعه‌ای گسسته از هدف‌های محلی تشکیل داد مثلا در بازی بیلیارد «کسب چهار توپ» و در بازی کراکت «جمع‌کردن برای دست بعدی» جزء این هدف‌های محلی و گسسته محسوب می‌شوند. بنابراین یک روش بهینه‌سازی محلی می‌تواند هر هدف محلی را از روی یک مجموعه گسسته از انتخابات بیابد. معمولا این بازی‌ها تصادفی‌اند بنابراین مدل تخته‌نرد را به کار می‌بریم که در آن از نمونه‌برداری اتفاقات به جای مجموع تمام اتفاقات استفاده می‌کنیم. بازی‌های بیلیارد و کراکت به طور کامل در دسته بازی‌های turn-taking مدل می‌شوند ولی بازی تنیس اینگونه نیست. زمانی که یک بازیکن توپ را پرتاب می‌کند، بازیکن دیگر مدام درحال حرکت است تا بتواند مقصد توپ را پیش‌بینی کرده و آنرا به رقیب بازگرداند. این امر باعث می‌شود تا بازی تنیس مشابه بازی‌های «واکنش همزمان» باشد. در این مورد می‌توان یک استراتژی تصادفی جهت پرتاب به کار گرفت تا رقیب نتواند مقصد توپ را پیش‌بینی کند.

**؟ ۱۵.۶ توضیح دهید که در بازی‌های دو نفره که مجموع امتیاز غیرصفر است، هر بازیکن تابع سودمندی مجزا داشته ولی هر دو بازیکن از تابع سودمندی نفر دیگر باخبراست، الگوریتم‌های بیشینه-کمینه و آلفا- بتا چه تغییری می‌کنند؟ اگر هیچ محدودیتی بر روی دو سودمندی پایانی نداشته باشیم، آیا هر گره امکان هرس شدن توسط آلفا- بتا را داراست؟ اگر مقدار تابع سودمندی بازیکنان در هر حالت حداکثر به اندازهٔ ثابت $k$ متفاوت باشد چگونه می‌توان حداکثر مشارکت را داشت؟** ***(این تمرین مشابه تمرین ۱۲.۵ در ویرایش سوم است)***

☑ حل: الگوریتم بیشینه-کمینه برای بازی‌های مجموع غیرصفر در این فصل توصیف شد که در آنجا تابع ارزیاب، برداری از مقادیر تعریف شد که هر مقدار آن متعلق به یک بازیکن است و مرحلهٔ پشتیبان، برداری را انتخاب می‌کند که برای بازیکنی که نوبت حرکتش است بیشترین مقدار را داشته باشد. در مثالی که در پایان بخش ۲.۲.۵ آورده شده توضیح داده شد که هرس آلفا- بتا در حالت کلی در بازی‌هایی با مجموع غیرصفر ممکن نیست زیرا یک برگ بررسی نشده که توسط این الگوریتم کنارگذاشته می‌شود، ممکن است برای هر دو بازیکن بهینه باشد.

**؟ ۱۶.۶ فرض کنید یک برنامه شطرنج دارید که قادر به ارزیابی یک میلیون گره در هرثانیه است. یک نمایش فشرده از حالت بازی برای ذخیره‌شدن در یک جدول جابجایی ارائه دهید. چه تعداد ورودی در جدولی با حافظه 500 مگابایت می‌توان ذخیره نمود؟ آیا این مقدار حافظه برای سه دقیقه جستجو جهت یک حرکت، کافی است؟ در زمان هر ارزیابی، چه تعداد از خانه‌های جدول بررسی می‌شوند؟ اکنون فرض کنید این جدول را در یک دیسک ذخیره کرده‌ایم، در صورتی که سخت‌افزار دیسک استاندارد باشد، چه تعداد ارزیابی در زمان جستجوی دیسک انجام می‌شود؟** ***(این تمرین مشابه تمرین ۱۵.۵ در ویرایش سوم است)***

☑ حل: صفحه شطرنج 8×8 است که هر بازیکن 16 مهره و مجموع مهره‌های بازی 32 مهره می‌باشد. برای تعریف هر موقعیت از بازی بایستی موقعیت هریک از این 32 مهره را در 64 خانه معرفی کنیم. برای هر مهره بایستی مختصات$(x,y)$ را بیان کنیم. که هر یک از این x و yها از عدد 1 تا 8 مقدار می‌گیرند. از آنجا که اعداد 1 تا 8 را با سه بیت می‌توان نمایش داد. پس هر مختصات $(x,y)$ به 6 بیت نیازمند است. پس بیان موقعیت 32 مهره به فضای 32×6=192 بیت یا 24 بایت نیازمند است. بنابراین هر موقعیت به 24 بایت فضا جهت ذخیره‌سازی نیاز دارد پس در جدولی با 500 مگابایت، می‌توان 20 میلیون موقعیت را ذخیره نمود زیرا $500\text{MB}\times 2^{20}$ بایت می‌شود که تقسیم این مقدار بر 24 عدد 20 میلیون را نتیجه می‌دهد (با صرف نظر از اشاره‌گرهای لیست درجدول‌های درهم). با توجه به این موضوع در طی سه دقیقه می‌توان 240 میلیون حالت را ذخیره نمود ولی طبق صورت سئوال در سه دقیقه 180 میلیون گره ارزیابی می‌شوند. برنامه‌های مدرن نظیر (Heinz, 2000) کلید درهم‌سازی هر موقعیت را گرفته و آن را به ازای هر موقعیت جدید تغییر می‌دهند. فرض کنید این کار حدودا به اندازهٔ 20 عملگر پیچیدگی داشته باشد. بنابراین در ماشینی 2 گیگا هرتزی که برای یک ارزیابی حدود 2000 عملگر صرف می‌کند، می‌توان در هر ارزیابی، حدود 100 مراجعه انجام داد. با داشتن نموداری از یک میلی‌ثانیه از جستجوی دیسک، حتی می‌توانیم در هر مراجعه 1000 ارزیابی را انجام دهیم. واضح است که مقادیر موجود در جدول دیسک مقیم، مشکوک هستند حتی اگر جهت کاهش خواندن از دیسک، چندین مکان‌نما به مرجع اصلی ذکر کنیم.

# فصل ۷ (ویرایش دوم)

**؟ ۱.۷ دنیای ومپوز را طبق مشخصات محیط در فصل ۲، توصیف کنید؟**

☑ حل: دنیای ومپوز نیمه‌مشاهده‌پذیر، قطعی، ترتیبی (شما بایستی حالت یک مکان را جهت استفاده در مراجعات بعدی آن به خاطر بسپارید)، استاتیک، گسسته و تک‌کارگزاره است.(ومپوز که منتظر بلعیدن یک فرد سرگردان است، کارگزاری جداگانه محسوب نمی‌شود). بنابراین دنیایی ساده پیش‌رو داریم و تنها پیچیدگی اصلی در این دنیا، نیمه‌مشاهده‌پذیربودن آن است.

**؟ ۲.۷ فرض کنید کارگزار در موقعیت شکل ۴.۷(a) قرار دارد یعنی در خانهٔ [1,1] هیچ چیزی مشاهده نمی‌کند ولی در خانهٔ [2,1] وجود نسیم و در خانهٔ [1,2] وجود بو را احساس می‌کند. کارگزار در این موقعیت، دربارهٔ خانه‌های [1,3] و [2,2] و [3,1] نگران است زیرا ممکن است در هر کدام از آنها چاله و در یکی از آنها ومپوز قرار داشته باشد. مطابق مثال شکل ۵.۷ مجموعهٔ دنیاهای ممکن را تشکیل دهید. (بایستی 32 مورد را بیابید). سپس مدل‌هایی که طبق KB جمع‌آوری شده صحیح هستند و مدل‌هایی که عبارات زیر در آنها صحیح هستند را مشخص نمایید:**

**$\alpha_2$= هیچ چاله‌ای در خانهٔ [2,2] نیست. $\alpha_3$= یک ومپوز در خانهٔ [1,3] قرار دارد.**

**و نشان دهید که $KB \models \alpha_2$ و $KB \models \alpha_3$ می‌باشند.** ***(این تمرین مشابه تمرین ۱.۷ در ویرایش سوم است)***

☑ حل:

| Model | $KB$ | $\alpha_2$ | $\alpha_3$ |
|---|---|---|---|
| | | *true* | |
| $P_{1,3}$ | | *true* | |
| $P_{2,2}$ | | | |
| $P_{3,1}$ | | *true* | |
| $P_{1,3}$, $P_{2,2}$ | | | |
| $P_{2,2}$, $P_{3,1}$ | | | |
| $P_{3,1}$, $P_{1,3}$ | | *true* | |
| $P_{1,3}$, $P_{3,1}$, $P_{2,2}$ | | | |
| $W_{1,3}$ | | *true* | *true* |
| $W_{1,3}$, $P_{1,3}$ | | *true* | *true* |
| $W_{1,3}$, $P_{2,2}$ | | | *true* |
| $W_{1,3}$, $P_{3,1}$ | *true* | *true* | *true* |
| $W_{1,3}$, $P_{1,3}$, $P_{2,2}$ | | | *true* |
| $W_{1,3}$, $P_{2,2}$, $P_{3,1}$ | | | *true* |
| $W_{1,3}$, $P_{3,1}$, $P_{1,3}$ | | *true* | *true* |
| $W_{1,3}$, $P_{1,3}$, $P_{3,1}$, $P_{2,2}$ | | | *true* |
| $W_{3,1}$, | | *true* | |
| $W_{3,1}$, $P_{1,3}$ | | *true* | |
| $W_{3,1}$, $P_{2,2}$ | | | |
| $W_{3,1}$, $P_{3,1}$ | | *true* | |
| $W_{3,1}$, $P_{1,3}$, $P_{2,2}$ | | | |
| $W_{3,1}$, $P_{2,2}$, $P_{3,1}$ | | | |
| $W_{3,1}$, $P_{3,1}$, $P_{1,3}$ | | *true* | |
| $W_{3,1}$, $P_{1,3}$, $P_{3,1}$, $P_{2,2}$ | | | |
| $W_{2,2}$ | | *true* | |
| $W_{2,2}$, $P_{1,3}$ | | *true* | |
| $W_{2,2}$, $P_{2,2}$ | | | |
| $W_{2,2}$, $P_{3,1}$ | | *true* | |
| $W_{2,2}$, $P_{1,3}$, $P_{2,2}$ | | | |
| $W_{2,2}$, $P_{2,2}$, $P_{3,1}$ | | | |
| $W_{2,2}$, $P_{3,1}$, $P_{1,3}$ | | *true* | |
| $W_{2,2}$, $P_{1,3}$, $P_{3,1}$, $P_{2,2}$ | | | |

شکل ۱.۷s گزاره‌هایی که بعنوان true در خطوط مشخص شده لیست نشده‌اند، false هستند.

در این جدول، تنها ورودی‌های true ذکر شده است.

منظور از مدل در یک دنیا، تمام حالات ممکن از وضعیت‌هاست که یکی از آنها در واقعیت صحت دارد ولی ما به دلیل اطلاعات ناقص خود نمی‌توانیم مدل حقیقی که بر دنیا منطبق است را بیابیم. برای یافتن آن مدل حقیقی، به مشاهدات و استنتاج بیشتر نیازمندیم. با انجام مشاهدات بیشتر، دانش ما از دنیای ومپوز بیشتر خواهد شد و اصطلاحا گفته می‌شود که دانشی به پایگاه‌دانش افزوده شد. در ابتدای این تمرین در مورد سه خانه [1,3] و [2,2] و [3,1] هیچ اطلاعی نداریم و هر یک از این خانه‌ها دارای چاله، ومپوز، نسیم و یا بو هستند. بنابراین برای هر خانه: یا چاله وجود دارد یا ندارد (برای سه خانه، 8 حالت مختلف می‌شود)، یا ومپوز وجود دارد یا ندارد (برای سه خانه، 8 حالت مختلف می‌شود)، یا نسیم وجود دارد یا ندارد (برای سه خانه، 8 حالت مختلف می‌شود)، یا بو وجود دارد یا ندارد (برای سه خانه، 8 حالت مختلف می‌شود). باتوجه به مجموع موارد فوق بایستی در کل تعداد 32 مدل ممکن ترسیم شود. در کتاب پیشنهاد می‌دهد که تمام این مدل‌ها را به طور دیاگرامی ترسیم کرده و سپس دور مدل‌هایی که طبق دانش شما صحیح هستند خط بکشید. تعدادی از این مدل‌ها با توجه به پایگاه‌دانش یعنی اطلاعات ما از محیط، رد می‌شوند: به عنوان مثال می‌دانیم که در این محیط فقط یک ومپوز وجود دارد پس مدل‌هایی که دارای بیش از یک ومپوز هستند در محدوده پایگاه نیستند؛ همچنین با توجه به اینکه محتوای سه خانه از محیط را می‌دانیم، پایگاه دانش ما تکمیل‌تر است مثلا می‌دانیم که اگر در خانه‌ای بو باشد در مجاور آن حتما ومپوز وجود دارد پس مدل‌هایی که این قانون را نقض می‌کنند از محدوده خط KB کنار می‌روند. با ادامه همین روال مدل‌هایی که طبق دانش ما از محیط صحیح هستند را درون محدوده‌ای بنام KB قرار می‌دهیم که تنها 12 مدل می‌باشند (8 حالت مختلف برای نمایش چاله در سه خانه و چهار حالت مختلف برای نمایش ومپوز البته عدم وجود ومپوز نیز حساب شده است). در اینجا جهت صرفه‌جویی در فضای مصرفی، به جای ترسیم دیاگرام‌های دنیا می‌توانید از جدول استفاده کنید. اگر بخواهیم اثبات کنیم که می‌توان $\alpha_2$ را از این پایگاه‌دانش ایجاب کرد باید بتوان تمام مدل‌های KB را درون محدوده مدل‌های $\alpha_2$ قرار داد (یعنی محدوده KB زیرمجموعه‌ای از محدوده $\alpha_2$ باشد) که در این مثال محدوده مدل KB، زیرمجموعه هر کدام از $\alpha_2$ و $\alpha_3$ است. پس داریم: $KB \models \alpha_2$ و $KB \models \alpha_3$.

**؟ ۳.۷ مسأله تصمیم‌گیری در مورد صحت یک گزاره منطقی با توجه به مدلی مشخص را در نظر بگیرید: الف) یک الگوریتم بازگشتی به صورت ?PL-True(s,m) بنویسید که اگر و فقط اگر عبارت s در مدل m صحیح باشد، کلمهٔ true را برمی‌گرداند. (m برای هر نماد در عبارت s، مقداری حقیقی را برمی‌گرداند). این الگوریتم بایستی نسبت به اندازهٔ عبارت، در زمانی خطی اجرا شود. ب) سه جمله مثال بزنید که درست یا غلط بودن آنها را به طور قطعی می‌دانیم ولی مدل آنها جزئی است، یعنی فقط درستی برخی از نمادهای جملات مشخص است. ج) نشان دهید که در حالت کلی درستی یک جمله (در صورت وجود) در مدلی-جزئی نمی‌تواند به طور کارا تعیین شود. د) الگوریتم ?PL–True را تغییر دهید تا ضمن حفظ ساختار بازگشتی زمان اجرای خطی، بتواند گاهی مواقع در مورد درستی یک عبارت در مدلی جزئی قضاوت کند. سه جمله مثال بزنید که الگوریتم شما نمی‌تواند صحت آنها را در مدلی‌جزئی شناسایی کند. هـ) بررسی کنید که الگوریتم تغییریافته، چه تأثیری در کارایی TT–ENTAILS دارد؟** ***(این تمرین مشابه تمرین ۳.۷ در ویرایش سوم است)***

☑ حل: **الف)** قطعه کدی به زبان پاتیون برای PL–True و نسخه‌ای از ask به زبان Lisp برای این منظور وجود دارد. و این کدها به زبان جاوا برای تاریخ May 2003 وجود نداشته ولی ممکن است در آینده افزوده شود. **ب)** منظور از مدل‌جزئی آن است که مقدار تمام متغیرهای موجود در جمله مشخص نبوده و تنها مقدار بخشی از آنها را می‌دانیم. عباراتی نظیر True, $P \vee \neg P$, $P \wedge \neg P$ مقداری قطعی دارند یعنی یا قطعا درست و یا نادرست هستند حتی اگر مقدار متغیر $P$ را ندانیم (مدل جزئی). **ج)** دو جمله با $k$ متغیر را در مدلی‌جزئی یعنی شرایطی که مقدار درستی تمام متغیرها را نمی‌دانیم در نظر بگیرید. یک جمله به ازای تمام $2^k$ مقدار ممکن، صحیح است (مانند مثال‌های موجود در بخش قبل) ولی جملهٔ دیگر به ازای یک حالت از $2^k$ مقدار ممکن نادرست می‌باشد. بنابراین بایستی در حالت کلی تمام $2^k$ حالت ممکن را بررسی کنیم که زمانی نمایی صرف می‌شود. **د)** پیاده‌سازی PL–True در زبان پاتیون به گونه‌ای است که در صورتی مقدار True را برمی‌گرداند که تمام بخش‌های یک ترکیب عطفی صحیح باشند؛ و در صورتی که تمام بخش‌های یک ترکیب فصلی غلط باشند مقدار False را برمی‌گرداند. این اتفاق حتی در صورتی که فصل/عطف شامل متغیرهای بدون مقدار باشد نیز همین است. بنابراین در یک مدل‌جزئی که می‌دانیم $P$ صحیح است، آنگاه این برنامه برای $P \vee Q$ مقدار صحیح و برای $\neg P \wedge Q$ مقدار غلط را برمی‌گرداند زیرا یکی از بخش‌های هر ترکیب مقدار دارد. ولی این برنامه، مقدار درستی عباراتی چون $Q \vee \neg Q$ ، $Q \vee \text{True}$ ، $Q \wedge \neg Q$ را نمی‌داند زیرا متغیر $Q$ مقدار نداشته و این برنامه نمی‌تواند درصورت مقدار نداشتن هر دو بخش یک ترکیب آنرا ارزیابی کند. **هـ)** نسخهٔ tt–entails موجود از تابع PL–True تغییر یافته استفاده می‌کند که در صورتی که از آن استفاده نکنیم، عملکردی کندتر خواهد داشت.

**؟ ۴.۷ هر یک از این عبارات را اثبات کنید: الف) a صحیح است اگر و فقط اگر $a \models \text{True}$. ب) به ازای هر $a$، داریم $\text{False} \models a$. ج) $a \models \beta$ برقرار است اگر و فقط اگر عبارت $(a \Rightarrow \beta)$ برقرار باشد. د) $a \equiv \beta$ برقرار است اگر و فقط اگر عبارت $(a \Leftrightarrow \beta)$ برقرار باشد. هـ) $a \models \beta$ برقرار است اگر و فقط اگر عبارت $(a \wedge \neg \beta)$ برقرار باشد.** ***(این تمرین مشابه تمرین ۵.۷ در ویرایش سوم است)***

☑ حل: یادآوری می‌کنیم که منظور از عبارت $a \models \beta$ آن است که اگر در دنیای مساله، مدلی طبق جمله $a$ صحیح بود، آنگاه در جمله $\beta$ نیز صحیح خواهد بود. (به عنوان مثال برای دنیای جاروبرقی جملاتی نظیر «$a$- در خانهٔ سمت چپ آشغال نیست و جاروبرقی در خانه سمت راست قرار دارد» و «$\beta$- جاروبرقی در خانه سمت راست قرار دارد» دو جمله‌ای هستند که $a$ دارای دو مدل‌ممکن و $\beta$ دارای چهار مدل‌ممکن می‌باشد. منظور از مدل هر چیدمانی از وضعیت‌های دنیاست که با جمله مربوطه مطابقت

شته باشد. در مورد این مثال $a \models \beta$ برقرار است. می‌توانید با ترسیم این چهار مدل و خط‌کشیدن دور محدوده مدل‌های هر
ملة به این موضوع پی ببرید. اگر محدوده هر جمله درون محدوده جمله دیگر واقع شود، آنگاه آنرا ایجاب می‌کند.) بنابراین:
ب) اگر هر دنیای دلخواهی را در نظر بگیرید، جمله True فقط به ازای مدل واقعی دنیا صحیح است. به عنوان مثال در دنیای
اروبرقی، مدل‌هایی از دنیا در محدوده جمله True قرار می‌گیرند که حقیقت واقعی دنیا هستند یعنی موقعیت حقیقی آشغال
جاروبرقی (می‌توانید آن را برای خود در نظر بگیرید). حال فرض کنید برای جمله «$a$- در خانهٔ سمت‌چپ آشغال نیست و
اروبرقی در خانه سمت‌راست قرار دارد»، 2 مدل‌ممکن آنرا درون محدوده‌ای با نام $a$ قرار می‌دهیم. حال اگر جملهٔ $a$ صحیح
ند، یعنی با واقعیت دنیا تطبیق دارد پس مسلم است که می‌توان مدل‌واقعی دنیا را درون محدوده $a$ پیدا کرد؛ پس داریم
$a \models$ True. اگر در مورد درستی مدل‌های واقع در محدوده $a$ اطلاعی نداشته باشیم ولی بدانیم که $a \models$ True، پس نتیجه
می‌شود که مدل حقیقی دنیا زیرمجموعه‌ای از محدوده $a$ است پس حتما باید $a$ صحیح باشد. **ب)** False در هیچ مدلی برقرار
ست و طبق تعریف نماد ایجاب گفتیم که اگر مدلی در بخش سمت چپ وجود داشت، باید در سمت راست هم وجود داشته
ند تا بگوییم رابطه برقرار است. ولی در مورد عدم‌وجود چیزی گفته نشد. بنابراین $a$ در هر مدلی که False برقرار نباشد،
جود دارد. **ج)** عبارت $a \Rightarrow \beta$ در مدلی برقرار است که یا $\beta$ یا $\neg a$ برقرار باشد که با تعریف ایجاب یکسان است. **د)** اعمال اثبات
سمت ج در دو طرف. **هـ)** این مورد همان عبارت قسمت ج است زیرا جدول درستی $a \Rightarrow \beta$ دقیقا با جدول درستی $\neg a \vee \beta$
سان است.

**۵.۷ فرض کنید تنها 4 گزاره با نام‌های D,C,B,A داریم. چه تعداد مدل برای این عبارات وجود دارد: الف) $(A \wedge B) \vee (B \wedge C)$**
**ب) $A \vee B$ ج) $A \Leftrightarrow B \Leftrightarrow C$** ***(این تمرین مشابه تمرین ۷.۷ در ویرایش سوم است)***

حل: برای این منظور کافی است برای هر عبارت، جدول درستی تشکیل داده و تعداد سطرهای هر عبارت که نتیجه True
رد را بشماریم. یادآوری می‌کنیم که برای هر عبارت، متغیرهایی که ذکر نشده‌اند را نیز در تعداد دخالت دهید مثلا اگر عبارتی
ط شامل متغیرهای A و B می‌باشد آنگاه تعداد مدل‌های بدست آمده برای {A,B} را در عدد $2^2$ به منظور حالات مختلف
و D ضرب کنید. **الف)** 6 **ب)** 12 **ج)** 4

**۶.۷ در این فصل چهار رابط منطقی باینری برای اتصال عبارات تعریف شد. الف) آیا رابط دیگری نیز وجود دارد که مفید**
**شد؟ ب) تعداد رابطهای منطقی باینری چیست ؟ ج) چرا برخی رابطها خیلی مفید نیستند؟** ***(این تمرین مشابه تمرین ۸.۷ در***
***رایش سوم است)***

حل: منظور از رابط منطقی نمادی است که بتواند دو عبارت منطقی مانند $p$ و $q$ را به هم وصل کرده و طبق تعریف خود
رای یک جدول درستی باشد تا به ازای 4 حالت مختلف دو عبارت ([$p$=F,$q$=F]، [$p$=T,$q$=F]، [$p$=F,$q$=T]،
[$p$=T,$q$=T]) نشان دهد که نتیجه درست و یا نادرست است. از آنجا که برای دو عبارت نظیر $p$ و $q$ جدولی با 4 سطر داریم،
توانیم به ازای هر ترکیب مختلف T و F بودن نتیجه، یک رابط تعریف کنیم یعنی آن رابط باعث می‌شود تا آن ترکیب یکتا
ج دهد. با توجه به اینکه هر کدام از این 4 سطر می‌توانند درست یا نادرست باشند بنابراین $2^4$=16 حالت مختلف برای ستون
یجه خواهیم داشت. هر یک از این جداول درستی را، یک رابط می‌نامیم پس 16 رابط ممکن وجود خواهد داشت. شش مورد از
ن رابطها اهمیت ندارند زیرا یک یا هر دو ورودی را نادیده می‌گیرند مانند $\neg Q$، $\neg P$، $Q$، $P$، False و True. چهار مورد دیگر
رابطها را در این فصل مطالعه کردیم که عبارتند از ۸، ۷، $\Rightarrow$، $\Leftrightarrow$. از شش مورد مابقی نیز استلزام معکوس ($\Leftarrow$ به جای $\Rightarrow$)
ید می‌باشد. پنج مورد دیگر نقیض چهار عمل اصلی یعنی ۷، ۸، $\Rightarrow$، $\Leftrightarrow$، $\Leftarrow$ هستند.(به نقیض دو مورد اول به ترتیب
م‌های NAND و NOR داده‌اند).

**۷.۷ با استفاده از روش انتخابی خود در مورد اعتبار هر یک از هم‌ارزی‌های شکل ۱۱.۷ تحقیق کنید.** ***(این تمرین مشابه تمرین***
***۹.۷ در ویرایش سوم است)***

حل: در اینجا جهت صرفه‌جویی در پاسخ، فقط چهار جدول درستی اولیه را نشان می‌دهیم:

```
> (truth-table "P ^ Q <=> Q ^ P")
-----------------------------------------
 P  Q  P ^ Q  Q ^ P  (P ^ Q) <=> (Q ^ P)
-----------------------------------------
 F  F    F      F            \(true\)
 T  F    F      F             T
 F  T    F      F             T
 T  T    T      T             T
-----------------------------------------
NIL
```

```
> (truth-table "P | Q <=> Q | P")
-----------------------------------------
 P  Q  P | Q  Q | P  (P | Q) <=> (Q | P)
-----------------------------------------
 F  F    F      F            T
 T  F    T      T            T
 F  T    T      T            T
 T  T    T      T            T
-----------------------------------------
NIL
> (truth-table "P ^ (Q ^ R) <=> (P ^ Q) ^ R")
-------------------------------------------------------------------------------
 P  Q  R  Q ^ R  P ^ (Q ^ R)  P ^ Q ^ R  (P ^ (Q ^ R)) <=> (P ^ Q ^ R)
-------------------------------------------------------------------------------
 F  F  F    F          F           F                    T
 T  F  F    F          F           F                    T
 F  T  F    F          F           F                    T

 T  T  F    F          F           F                    T
 F  F  T    F          F           F                    T
 T  F  T    F          F           F                    T
 F  T  T    T          F           F                    T
 T  T  T    T          T           T                    T
-------------------------------------------------------------------------------
NIL

> (truth-table "P | (Q | R) <=> (P | Q) | R")
-------------------------------------------------------------------------------
 P  Q  R  Q | R  P | (Q | R)  P | Q | R  (P | (Q | R)) <=> (P | Q | R)
-------------------------------------------------------------------------------
 F  F  F    F          F           F                    T
 T  F  F    F          T           T                    T
 F  T  F    T          T           T                    T
 T  T  F    T          T           T                    T
 F  F  T    T          T           T                    T
 T  F  T    T          T           T                    T
 F  T  T    T          T           T                    T
 T  T  T    T          T           T                    T
-------------------------------------------------------------------------------
NIL
```

برای سایر عبارات، تنها نشان می‌دهیم که آنها طبق تابع Validity معتبر هستند:

```
> (validity "~~P <=> P")
VALID
> (validity "P => Q <=> ~Q => ~P")
VALID
> (validity "P => Q <=> ~P | Q")
VALID
> (validity "(P <=> Q) <=> (P => Q) ^ (Q => P)")
VALID
> (validity "~(P ^ Q) <=> ~P | ~Q")
VALID
> (validity "~(P | Q) <=> ~P ^ ~Q")
VALID
> (validity "P ^ (Q | R) <=> (P ^ Q) | (P ^ R)")
VALID
> (validity "P | (Q ^ R) <=> (P | Q) ^ (P | R)")
VALID
```

**۸.۷ کدامیک از عبارات زیر معتبر، ارضاپذیر و یا هیچ‌کدام هستند. نظر خود را با استفاده از جدول‌درستی و یـا قواعـد هـم ی شکل ۱۱.۷ اثبات کنید:**

**(Smoke⇒Fire) ⇒ (¬Smoke⇒¬Fire) ج) Smoke⇒Fire ب) Smoke⇒Smoke (**

**((Smoke ∧Heat) ⇒Fire)⇔ ((Smoke⇒Fire) ∨ (Heat ⇒Fire)) هـ) Smoke ∨ Fire ∨¬ Fire**

**Big ∨ Dumb ∨ (Big ⇒Dumb) ز) (Smoke ⇒Fire) ⇒ ((Smoke∧Heat) ⇒Fire)**

**(Big ∧ Dumb) ∨ ¬ Dumb** *(این تمرین مشابه تمرین ۱۰.۷ در ویرایش سوم است)*

حل: جهت تعیین اعتبار هر عبارت از تابع Validity استفاده می‌کنیم:

```
> (validity "Smoke => Smoke")
VALID
> (validity "Smoke => Fire")
SATISFIABLE
> (validity "(Smoke => Fire) => (~Smoke => ~Fire)")
SATISFIABLE
> (validity "Smoke | Fire | ~Fire")
VALID
> (validity "((Smoke ^ Heat) => Fire) <=> ((Smoke => Fire) | (Heat => Fire))
VALID
> (validity "(Smoke => Fire) => ((Smoke ^ Heat) => Fire)")
VALID
> (validity "Big | Dumb | (Big => Dumb)")
VALID
> (validity "(Big ^ Dumb) | ~Dumb")
SATISFIABLE
```

یاری از افراد در مورد قسمت هـ و ز، به خاطر وجود علت و معلول در استلزام دچار اشتباه می‌شوند. بنـابراین در قـسمت هـ ساس می‌شود که ترکیب دود و گرما منجر به آتش می‌شود و دلیلی وجود ندارد که یکی از آنها به تنهایی بتواند منجر به آتش د. به طور مشابه در قسمت ز، به نظر می‌رسد که هیچ رابطه‌ای بـین Dumb و Big وجـود نـدارد بنـابراین عبـارت بایـستی ناپذیر بوده ولی معتبر نیست. به هر حال این موارد اشتباه هستند زیرا استلزام نوعی علت و معلول نیست فقط نـوعی فـصل ـسوب مـی‌شـود. $P \Rightarrow Q$ همـان $\neg P \vee Q$ مـی‌باشـد. بنـابراین $Big \vee Dumb \vee (Big \Rightarrow Dumb)$ معـادل اسـت بـا $Big \vee Dumb \vee \neg Big \vee Dum$ که هم‌ارز است با $Big \vee \neg Big \vee Dumb$ و همواره صحیح است چه Big درست باشد و چـه رست، بنابراین عبارت معتبر است.

**۹.۷ (با اقتباس از Barwise وEtchemendy (سال 1993)) و فرضیهٔ زیر، آیا می‌توانید اثبات کنید که تک شاخ موجـودی طوره‌ای است؟ در مورد سحرآمیز بودن آن چطور؟ در مورد شاخ‌داری آن چطور؟ «اگر تک شاخ موجودی اسطوره‌ای اسـت، اه فناناپذیر خواهد بود ولی اگر اسطوره‌ای نباشد، آنگاه یک پستاندار فناپذیر خواهد بود. اگر یک تک شـاخ، پـستاندار یـا ناپذیر باشد، آنگاه شاخ‌دار خواهد بود و اگر تک شاخ یک موجود شاخ‌دار باشد حتما سحرآمیز است».** *(ایـن تمـرین مـشابه رین ۲.۷ در ویرایش سوم است)*

حل: طبق دو عبارت اول می‌یابیم که در صورتی که تک شاخ اسطوره‌ای باشد فناناپذیر است و در غیـر ایـن صـورت فناپـذیر ت. بنابراین بایستی یا پستاندار یا فناناپذیر باشد و بنابراین شاخ‌دار است. این بدان معناست که این موجود سحرآمیز نیز هست. هر حال نمی‌توان در مورد اسطوره‌ای بودن آن تصمیم گرفت. طبق کد استدلال گزاره‌ای داریم:

```
> (setf kb (make-prop-kb))
#S(PROP-KB SENTENCE (AND))
> (tell kb "Mythical => Immortal")
T
> (tell kb "~Mythical => ~Immortal ^ Mammal")
T
> (tell kb "Immortal | Mammal => Horned")
T
> (tell kb "Horned => Magical")
T
> (ask kb "Mythical")
NIL
> (ask kb "~Mythical")
```

```
> (ask kb "Magical")
T
> (ask kb "Horned")
T
```

**؟ ۱۰.۷ هر جمله در منطق گزاره‌ای معادل است با آنکه دنیایی داشته باشیم که مورد نادرست در آن وجود نداشته باشد. طبق این گفته اثبات کنید که می‌توان هر جمله را به صورت فرم CNF نوشت.** ***(این تمرین مشابه تمرین ۷.۱۱ در ویرایش سوم است)***

☑ حل: هر دنیا می‌تواند به عنوان ترکیب عطفی نمادها نظیر $(A \wedge C \wedge E)$ نوشته شود. به طوری که نماد نقیض در عطف نداشته باشیم و عبارتی نظیر $\neg(A \wedge C \wedge E)$ باید به صورت $(\neg A \vee \neg C \vee \neg E)$ بازنویسی شود. در این صورت یک جزء از فرم CNF تشکیل می‌شود و با ترکیب عطفی این اجزا یک عبارت کامل به فرم CNF شکل می‌گیرد و می‌توان تمام جملات ممکن برای دنیا را ذکر نمود.

**؟ ۱۱.۷ بازی کامپیوتری مین یاب[17] که بسیار مشهور است، ارتباط نزدیکی با دنیای ومپوز دارد. دنیای مین‌یاب شامل یک شبکه مستطیلی از N مربع با M مین مخفی درون آنهاست که هر مربع می‌تواند توسط کارگزار بررسی شود ولی در صورت وجود مین در آن خانه، کارگزار سریعا می‌میرد. در این بازی کارگزار پس از معاینهٔ هر خانه، عددی که نشان‌دهنده مجموع تعداد مین مجاور با آن خانه به طور افقی، عمودی و قطری است، را در آن خانه قرار می‌دهد. هدف این بازی آن است که تمام خانه‌های بدون مین را بدون مردن مشخص کنیم. الف) مقدار $X_{i,j}$ درست است اگر و فقط اگر مربع [i,j] دارای مین باشد. با استفاده از ترکیب منطقی گزاره‌های $X_{i,j}$، جمله‌ای بنویسید که نشان دهد دقیقا دو مین در مجاورت خانه [1,1] وجود دارد. ب) ادعای قسمت الف را تعمیم دهید و توضیح دهید که چگونه می‌توان یک جمله CNF ایجاد کرد که نشان دهد $k$ خانه از n همسایه شامل مین هستند. ج) به دقت توضیح دهید که یک کارگزار چگونه می‌تواند با استفاده از DPLL اثبات کند که یک خانه دارای مین است(یا نیست). با صرف نظر از اینکه تعداد کل مین‌ها در طی یافت‌ها باید M باشد. د) فرض کنید محدودیت تعداد کل مین‌ها به قسمت ب اضافه شود، تعداد بندها چه ارتباطی با M و N دارند؟ روشی ارائه دهید که با تغییر DPLL نیازی به نمایش صریح محدودیت تعداد کل نباشد. هـ) در صورت در نظر گرفتن محدودیت تعداد کل مین‌ها، آیا تمام نتایج حاصل از قسمت ج نامعتبر است؟ و) مثال‌هایی از پیکره‌بندی مقادیر هر کاوش که محدودیت تعدادکل داشته باشد ارائه دهید به طوری که محتوای هر خانهٔ بدون مین دارای اطلاعاتی از یک خانهٔ دورتر نیز باشد. (راهنمایی: یک صفحهٔ N×1 را در نظر بگیرید).** ***(این تمرین مشابه تمرین ۷.۲۳ در ویرایش سوم است)***

☑ حل: **الف)** این مورد یک ترکیب فصلی با 28 عملکرد فصل می‌باشد که هر کدام از آنها نشان‌دهندهٔ آن است که دو همسایه درست بوده و سایرین نادرست باشند. اولین ترکیب فصلی عبارتست از:

$$X_{2,2} \wedge X_{1,2} \wedge \neg X_{0,2} \wedge \neg X_{0,1} \wedge \neg X_{2,1} \wedge \neg X_{0,0} \wedge \neg X_{1,0} \wedge \neg X_{2,0}$$

هر کدام از 27 ترکیب فصلی دیگر، دو $X_{i,j}$ مختلف را جهت درست بودن انتخاب می‌کنند. **ب)** تعداد $\binom{n}{k}$ ترکیب فصلی وجود دارد که هر کدام نشان‌دهندهٔ آن است که $k$ تعداد از $n$ نماد درست بوده و سایرین غلط باشند. **ج)** به ازای هر سلول بررسی شده، عدد $n$ توسط بازی نتیجه می‌شود و یک جمله با $\binom{n}{8}$ فصل (به خاطر 8 همسایه هر خانه) ساخته می‌شود. تمام جملات را به هم وصل کنید. سپس از DPLL برای پاسخ به سؤال اینکه آیا به ازای جفت i,j دلخواه، این جمله شامل $X_{i,j}$ هست یا خیر، استفاده کنید. **د)** برای اعمال شرط «وجود تعداد کل مین‌ها به اندازهٔ M» باید یک ترکیب فصلی با $\binom{M}{N}$ عملکرد فصل که هر کدام به اندازهٔ N هستند، تشکیل دهیم. یادآوری می‌شود که: $\binom{M}{N} = M!/(M!-N!)$ است، بنابراین برای یک بازی مین‌یاب با 100 سلول و 20 مین، این مقدار بیش از $10^{39}$ خواهد بود و نمی‌تواند در کامپیوتر نشان داده شود. به هر حال می‌توان محدودیت تعداد کل مین‌ها را با الگوریتم DPLL اعمال نمود. پارامترهای *min* و *max* را به تابع DPLL اضافه می‌کنیم به طوری که آنها نشان‌دهندهٔ حداقل و حداکثر تعداد نمادهای انتساب نیافته در مدل باشند که بایستی در نهایت true شوند. برای حالتی که این محدودیت اعمال نشود بایستی برای این دو پارامتر به ترتیب مقادیر 0 , N اعمال شود. برای مسأله مین‌یاب، مقدار M هم برای *min* و هم برای *max* استفاده می‌شود. درون تابع DPLL اگر *min* کمتر از تعداد نمادهای باقی‌مانده باشد و یا *max* کمتر از صفر باشد، فورا تابع شکست خورده و مقدار false را برمی‌گرداند. برای هر فراخوانی بازگشتی از DPLL بایستی *max* و *min* با توجه به انتساب مقدار true به یک نماد، بروز شوند. **هـ)** با افزودن چنین قابلیتی به DPLL و رمزگذاری محدودیت تعداد کل مین‌ها هیچ نتیجه‌ای نامعتبر نیست. **و)** رشته‌ای از 1ها و خانه‌های بررسی‌نشده(که با علامت خط تیره مشخص شده است) را در نظر بگیرید:

|-|1|-|1|-|1|-|1|-|1|-|1|-|1|-|



اکنون دو مدل ممکن وجود دارد: مین‌ها در خانه‌هایی با تعداد زوج خط تیره قرار دارند یا در خانه‌هایی که تعداد خط تیره فرد است. با بررسی هر پایان، تعیین می‌شود که کدام سلول‌ها در پایان دور، خالی یا دارای مین هستند.

**؟ ۱۲.۷ این تمرین به بررسی رابطهٔ بین بند و جملات استلزام می‌پردازد. الف) نشان دهید که بند $(\neg P_1 \vee \ldots \vee \neg P_m \vee Q)$ به طور منطقی با جملهٔ استلزام $(P_1 \wedge \ldots \wedge P_m \Rightarrow Q)$ هم ارز است. ب) نشان دهید که هر بند (برخلاف تعداد الفاظ مثبت) می‌تواند به صورت $(P_1 \wedge \ldots \wedge P_m) \Rightarrow (Q_1 \vee \ldots \vee Q_n)$ نوشته شود که $Q$ها و $P$ها نمادهای گزاره‌ای می‌باشند. به پایگاه‌دانشی شامل این جملات، شکل نرمال استلزامی یا شکل کوالسکی می‌گویند. ج) قاعده تحلیل‌کامل را برای جملاتی به فرم نرمال استلزامی، بنویسید. *(این تمرین مشابه تمرین ۱۳.۷ در ویرایش سوم است)***

☑ حل: **الف)** عبارت $P \Rightarrow Q$ هم‌ارز است با $\neg P \vee Q$ که در آن علامت استلزام حذف شده است (شکل ۱۱.۷) و عبارت $\neg(P_1 \wedge \ldots \wedge P_m)$ طبق قاعده مورگان هم‌ارز است با $(\neg P_1 \vee \ldots \vee \neg P_m)$. بنابراین عبارت $(\neg P_1 \vee \ldots \vee \neg P_m \vee Q)$ هم‌ارز است با $(P_1 \wedge \ldots \wedge P_m) \Rightarrow Q$ **ب)** یک بند می‌تواند الفاظ مثبت و منفی داشته باشد. آنها را به صورت $(\neg P_1 \vee \ldots \vee P_m \vee Q_1 \vee \ldots \vee Q_n)$ مرتب کرده و با قرار دادن $Q = Q_1 \vee \ldots \vee Q_n$ داریم: $(\neg P_1 \vee \ldots \vee P_m \vee Q_1 \vee \ldots \vee Q_n)$ که هم‌ارز با $(P_1 \wedge \ldots \wedge P_m) \Rightarrow Q_1 \vee \ldots \vee Q_n$ است. **ج)** برای عبارات اتمیک $p_i$، $q_i$، $r_i$ و $s_i$ که در آنها $\text{UNIFY}(p_j, q_k) = \theta$ داریم:

$$\frac{p_1 \wedge \ldots p_j \ldots \wedge p_{n_1} \Rightarrow r_1 \vee \ldots r_{n_2} \qquad s_1 \wedge \ldots \wedge s_{n_3} \Rightarrow q_1 \vee \ldots q_k \ldots \vee q_{n_4}}{\text{SUBST}(\theta, (p_1 \wedge \ldots p_{j-1} \wedge p_{j+1} \wedge p_{n_1} \wedge s_1 \wedge \ldots s_{n_3} \Rightarrow r_1 \vee \ldots r_{n_2} \vee q_1 \vee \ldots q_{k-1} \vee q_{k+1} \vee \ldots \vee q_{n_4}))}$$

**؟ ۱۳.۷ در این تمرین بیشتر قسمت‌های کارگزار ومپوز مبتنی بر مدار را طراحی خواهیم نمود. الف) معادله‌ای مشابه معادلهٔ (۴.۷) برای گزاره Arrow بنویسید که در صورتی که کارگزار هنوز یک تیر داشته باشد، مقدار true را برگرداند. مدار مربوطه را ترسیم کنید. ب) با استفاده از معادلهٔ (۵.۷) به عنوان یک مدل، قسمت الف را برای گزاره *FacingRight* تکرار کنید. ج) نسخه‌ای از معادلات ۷.۷ و ۸.۷ را برای یافتن ومپوز ایجاد کرده و مدار آن را ترسیم کنید.**

☑ حل: **الف)**

$$Arrow^t \Leftrightarrow Arrow^{t-1} \wedge \neg Shoot^t$$

**ب)**

$$\begin{aligned} FacingRight^t \Leftrightarrow\ & (FacingRight^{t-1} \wedge \neg TurnRight^t \wedge \neg TurnLeft^t) \\ \vee & (FacingUp^{t-1} \wedge TurnRight^t \\ \vee & (FacingDown^{t-1} \wedge TurnLeft^t \end{aligned}$$

**ج)** این معادله، مشابه معادلات ۷.۷ و ۸.۷ خواهد بود با این تفاوت که به جای نماد P برای چاله، از $W$ برای ومپوز استفاده می‌کنیم و به جای $B$ برای نسیم از $S$ برای بو استفاده می‌کنیم:

$$\begin{aligned} K(\neg W_{4,4})^t &\Leftrightarrow K(\neg S_{3,4})^t \vee K(\neg S4,4)^t \\ K(W_{4,4})^t &\Leftrightarrow K(S_{3,4})^t \wedge K(\neg W_{2,4})^t \wedge K(\neg W_{3,3})^t \\ &\quad \vee (K(S_{4,3})^t \wedge K(\neg W_{4,2})^t \wedge K(\neg W_{3,3})^t \end{aligned}$$

**؟ ۱۴.۷ توضیح دهید که منظور از رفتار بهینه در دنیای ومپوز چیست. نشان دهید که تعریف ما از PL-Wumpus-Agent بهینه نیست و راه‌کاری جهت بهبود آن ارائه دهید.**

☑ حل: منظور از رفتار بهینه آن است که بتوان یک سودمندی متوسط به خوبی سایر برنامه‌های کارگزار بدست آورد. کارگزار PL-Wumpus-Agent زمانی که یک حرکت تصادفی انجام می‌دهد(حتی شاید در سایر موارد منطقی نیز غیربهینه باشد) به طور مشخص غیربهینه است. به عنوان مثال در مواردی که چندین خطر وجود دارد(بو و نسیم) ولی هیچ حرکت امنی وجود ندارد، این کارگزار به طور تصادفی انتخاب می‌کند. در یک تحلیل کاملتر بایستی نشان دهیم که چه زمان بهتر است این کار را کرده و چه زمان بهتر است به خانه اولیه برگشته و از دنیای ومپوز خارج شویم و از کسب طلا منصرف شویم. حتی زمانی که قرار است یک مکان غیرامن انتخاب شود، کارگزار درجهٔ امنیت آن را تشخیص نمی‌دهد. کارگزار فقط می‌داند که یک خانهٔ غیرامن شامل خطر در مدل‌های ممکن را برگزیند. این توضیحات نشان می‌دهد که استفاده از کارگزار منطقی برای این شرایط بسیار دشوار خواهد بود ولی کارگزار احتمالاتی می‌تواند از عهدهٔ این مسأله برآید.

**؟ ۱۵.۷ PL-Wumpus-Agent را به گونه‌ای گسترش دهید که تمامی واقعیت‌های مرتبط با پایگاه‌دانش را پیگیری کند.**

☑ حل: PL-Wumpus-Agent می‌تواند 6 متغیر حالت استاتیک را در پایگاه‌دانش پیگیری کند. مشکل زمانی اتفاق می‌افتد که این متغیرها تغییر کنند، زیرا نمی‌توانیم اطلاعات جدیدی درباره آنها بیافزاییم (مانند موقعیت چاله‌ها و نسیم‌ها) و تنها می‌توان اطلاعات موجود را تغییر داد. این موضوع با ماهیت منطق خیلی جور نبوده و برای حقیقت‌های جاویدان که همیشگی هستند طراحی شده است. بنابراین دو راه‌کار پیش رو داریم. یکی آنکه زمان را به هر گزاره به صورت بالانویس، اضافه کنیم

همان طور که در کارگزار مدار مشاهده شد). بنابراین می‌توانیم به عنوان مثال عمل $TELL(KB,A_{1,1}^3)$را برای وجود کارگزار در موقعیت [1,1] در زمان 3 بیان کنیم. آنگاه برای زمان 4 می‌توان بسیاری از بخش‌های گزارهٔ موجود را کپی کرده و سپس بخشی از آن را ویرایش یا اضافه نمود. روش دیگر آن است که برای هر گزاره یک زمان عمر تعیین کنیم که گزاره‌های خارج از زمان، از پایگاه‌دانش حذف شوند. بنابراین برای بازگویی حرکت کارگزار از خانهٔ 1,1 به 1,2 می‌توان ابتدا $RETRACT(KB,A_{1,1})$ و سپس $TELL(KB,A_{1,2})$ را اجرا کنیم. فصل ۱۰، مفاهیم و پیاده‌سازی RETRACT را توصیف می‌کند.

**؟ ۱۶.۷ چقدر طول می‌کشد تا عبارت $KB \models \alpha$ را با استفاده از DPLL اثبات کنید، به طوری که $\alpha$ یک لفظ موجود در KB باشد؟ توضیح دهید. *(این تمرین مشابه تمرین ۲۳.۷ در ویرایش سوم است)***

☑ حل: این زمان بستگی به تعداد نمادهای خالص به علاوه تعداد بندهای واحد دارد. طبق برهان خلف فرض می‌کنیم $KB \models \alpha$ نادرست باشد و سپس اثبات می‌کنیم که این فرض غلط است. می‌دانیم که $\neg(KB \models \alpha)$ معادل است با $KB \wedge \neg \alpha$ . طبق این جمله، الگوریتم در ابتدا تمام نمادهای خالص را حذف کرده سپس بر روی بندهای واحد کار می‌کند تا زمانی که به $\alpha$ برسد(که آن نیز بندی واحد است). در این زمان الگوریتم به سرعت تشخیص می‌دهد که هر انتخاب(درست یا نادرست) برای $\alpha$ منجر به شکست خواهد شد. این بدان معناست که فرض نقیض ابتدایی غلط بوده و مسأله اثبات شده است.

**؟ ۱۷.۷ عملکرد DPLL بر روی پایگاه‌دانش شکل ۱۵.۷ در راه اثبات Q را پیگیری نموده و عملکرد آن را با رفتار الگوریتم زنجیره‌ای پیش‌رو مقایسه کنید. *(این تمرین مشابه تمرین ۲۴.۷ در ویرایش سوم است)***

☑ حل: در اینجا از آوردن جزئیات پیگیری DPLL خودداری شده است ولی می‌توانید کد آنرا بیابید. ولی روال کار به طور ساده این گونه است: قاعده بند واحد در DPLL ما را مطمئن می‌سازد که تمام اتم‌های شناخته شده به سایر بندها منتشر شده‌اند.

# فصل ۸ (ویرایش دوم)

**؟ ۱.۸ یک پایگاه دانش منطقی، جهان را توسط مجموعه‌ای از جملات که هیچ ساختار روشنی ندارند، توصیف می‌کند. از طرف دیگر، در بازنمایی‌قیاسی، ساختار فیزیکی و ساختار آن چیزی که نمایش می‌دهد مرتبط می‌باشد. یک نقشه از جاده‌های کشورتان را به عنوان یک بازنمایی قیاسی از حقایق کشور در نظر بگیرید(این نقشه حقایق را توسط زبان نقشه بیان می‌کند). ساختار دوبعدی نقشه با ساختار دوبعدی سطح جاده متناظر است. الف) 5 نمونه از نمادهای موجود در زبان نقشه را بیان کنید. ب) یک جملهٔ صریح، جمله‌ای است که توسط سازندهٔ بازنمایی بیان شود. و یک جملهٔ غیرصریح، جمله‌ای است که با توجه به مشخصات بازنمایی‌قیاسی از جملات‌صریح نتیجه گرفته شود. سه نمونه از هر یک از جملات صریح و غیرصریح در زبان نقشه را بیان کنید. ج) سه نمونه از حقایقی در مورد ساختار فیزیکی کشور که قابل نمایش در زبان نقشه نیست، ارائه دهید. د) دو نمونه از حقایقی مثال بزنید که بیان آنها در زبان نقشه ساده‌تر از منطق مرتبهٔ اول باشد. هـ) دو نمونهٔ دیگر از بازنمایی‌قیاسی مفید مثال بزنید. مزایا و معایب هر یک از این زبان‌ها را بیان کنید. *(این تمرین مشابه تمرین ۱.۸ در ویرایش سوم است)***

☑ حل: این سؤال، راه‌حل‌های مختلفی دارد. تفاوت اصلی بین بازنمایی قیاسی و بازنمایی جمله‌ای در آن است که وقتی یک موضوع بدرستی رمزنگاری شود آنگاه بازنمایی قیاسی به طور خودکار نتایجی تولید می‌کند که به راحتی قابل درک است. **الف)** بسته به مقیاس و نوع نقشه، معمولا نمادهای زبان نقشه شامل این موارد می‌باشد: علامت‌های شهر و حومه، نمادهای جاده (انواع مختلف)، فانوس دریایی، بناهای تاریخی، جریان رودخانه، تقاطع آزاد راه‌ها و غیره ... **ب)** جملات صریح و غیرصریح: تفاوت آنها کمی نیاز به مهارت دارد. ولی ایده اصلی در آن است که زمانی که یک نماد در مکانی خاص از نقشه قرار می‌گیرد مسلم است که چیزی را به صراحت نشان می‌دهد(مثلا می‌گوید برج *Coit* در اینجا واقع است) ولی ساختار قیاسی برای بازنمایی نقشه از چندین جملهٔ غیرصریح برای بیان این موضوع استفاده می‌کند. جملات صریح: بنایی تاریخی بنام برج *Coit* در این مکان واقع است؛ خیابان *Lombard* حدودا از شرق به سمت غرب ادامه دارد؛ خلیج سان‌فرانسیسکو وجود داشته و به این شکل است. جملات غیرصریح: *Van Ness* بزرگتر از *Willard* شمالی است؛ اسکلهٔ *Fisherman* در شمال ناحیهٔ *Mission* قرار دارد؛ کوتاهترین مسیر حرکت بین برج *Coit* تا *Twin Peak* به فلان صورت است. **ج)** جملاتی که در زبان نقشه قابل بازنمایی نیستند: تپه *Telegraph* تقریبا مخروطی شکل بوه و 430 فوت ارتفاع دارد(با این فرض که نقشه نمادهای توپوگرافی را دربرنداشته باشد)؛ در سال 1980 هیچ پل ارتباطی بین سان‌فرانسیسکو تا کشور *Marin* وجود نداشت(نقشه نمی‌تواند تغییرات اطلاعات را نشان دهد)؛ فاصلهٔ *Walnut Greek* تا ایالت‌های شرقی یا غربی آن 680 می‌باشد(نقشه دارای اطلاعات مجزا نیست). **د)** جملاتی که در زبان نقشه ساده‌تر بیان می‌شوند: هر جمله‌ای که به سادگی در زبان نوشته شود برای این سؤال پاسخ خوبی نخواهد بود. هر عبارت زبانی که مربوط به ساختار فیزیکی سان‌فرانسیسکو شود (به عنوان مثال سان‌فرانسیسکو در انتهای یک شبه‌جزیره و مدخل خلیج واقع شده است) می‌تواند به سادگی در ساختار پیش‌بینی‌شدهٔ نقشه بگنجد زیرا نقشه برای این هدف طراحی شده است. حقایقی چون شکل خط‌ساحلی و یا مسیر یک جاده در زبان نقشه به بهترین نحو ممکن بیان می‌شوند. حتی می‌توان در مورد خط ساحلی روی نقشه جملات بسیار زیادی بیان نمود. یعنی به ازای هر نقطه جوهر بر روی نقشه جمله‌ای داشته باشیم به خصوص اگر نقشه به صورت دیجیتال ترسیم شده باشد. در این مورد مزیت نقشه آن است که علاوه بر سهولت استنتاج، برای محاسبات بصری انسان نیز مفید است. **هـ)** مثال‌هایی از سایر بازنمایی‌های قیاسی: • ضبط صدای **آنالوگ** بر روی نوار. مزایا: مدارهایی ساده می‌توانند صداها را ضبط و پخش مجدد کنند. معایب: ممکن است دچار خطا یا نویز شود، برای مجزا کردن صداهای مختلف یا حذف نویز به فرآیندی پیچیده نیازمندیم. • ساعت‌های باستانی. مزایا: به سرعت و راحت خوانده می‌شوند، جهت تعیین زمان در دسترس نیاز به محاسبات زیاد ندارد. معایب: خواندن زمان دقیق دشوار است، به سادگی می‌تواند واحدهای زمانی کوچک (مانند میلی ثانیه) را نشان دهد. • تمام انواع گراف‌ها، نمودار میله‌ای و نمودار دایره‌ای. مزایا: داده‌ها را بسیار فشرده می‌کند، تحلیل آنها ساده‌تر می‌شود، ارتباطات اطلاعاتی که بتوان به سادگی آنها را تفسیر کرد بهتر می‌شود. معایب: غیردقیق بوده و نمی‌تواند اطلاعات مجزا یا متناقض را نمایش دهد.

**؟ ۲.۸ پایگاه‌دانشی فقط شامل دو جمله $P(a)$ و $P(b)$ را در نظر بگیرید. آیا این پایگاه‌دانش $\forall x\ P(x)$را ایجاب می‌کند؟ پاسخ خود را بر اساس مدل‌ها توضیح دهید. *(این تمرین مشابه تمرین ۲.۸ در ویرایش سوم است)***

☑ حل: این پایگاه دانش نمی‌تواند $\forall x\ P(x)$را ایجاب کند. برای نشان دادن این موضوع باید مدلی ارائه دهیم که در آن خاصیت‌های $P(a)$ و $P(b)$ برقرار باشد ولی جمله $\forall x\ P(x)$ در آن غلط باشد. برای این منظور، مدلی با سه عنصر $a$، $b$ و $c$ در نظر بگیرید که فقط برای دو عنصر $a$ و $b$، خاصیت $P$ برقرار باشد. پس این جمله در حالت کلی برقرار نمی‌باشد.

**؟ ۳.۸ آیا جملهٔ $\exists x,y\ x=y$ معتبر است؟ توضیح دهید. *(این تمرین مشابه تمرین ۳.۸ در ویرایش سوم است)***

☑ حل: جملهٔ $\exists x,y\ x=y$ معتبر است. زیرا یک جمله در صورتی معتبر نامیده می‌شود که در تمام مدل‌ها و به ازای هرگونه مقداردهی صحیح باشد. جملاتی که دارای سور وجودی هستند در صورتی صحیح هستند که شرایطی یافت شود که به هر متغیر مقداری از دامنه آن تخصیص داده و آن جمله برقرار شود. بر طبق استاندارد مفاهیم FOL که در این فصل گفته شد، هر مدل

حداقل دارای یک دامنهٔ عنصر می‌باشد بنابراین می‌توان تفسیر کرد که به دو متغیر $x$ و $y$ اولین دامنهٔ عنصر را انتساب دهیم آنگاه $x=y$ صحیح خواهد شد.

**؟ ۴.۸ یک جملهٔ منطقی بنویسید که اگر در یک دنیا درست باشد، آن دنیا دقیقا دارای یک شی باشد.** ***(این تمرین مشابه تمرین ۴.۸ در ویرایش سوم است)***

☑ حل: عبارت $\forall x,y\ x=y$ قید می‌کند که دقیقا یک شی در دنیا موجود است. اگر فرض کنیم که در این دنیا دو شی موجود باشند یعنی $x$ و $y$ به اشیایی متفاوت منتسب شوند، آنگاه این جمله غلط خواهد شد.

**؟ ۵.۸ یک فرهنگ‌لغت نمادها شامل $c$ نماد ثابت، $P_k$ نماد مسند با درجهٔ $k$، $F_k$ نماد تابعی با درجهٔ $k$ که $1\le k\le A$ را در نظر بگیرید. اندازهٔ دامنه، ثابت و برابر $D$ می‌باشد. برای هر ترکیب مدل- تفسیر، هر نماد مسندی یا تابعی به ترتیب به یک رابطه یا تابع با همان درجه نگاشت می‌یابد. می‌توانید فرض کنید که توابع در مدل، ورودی‌های چندگانه دارند که در تابع هیچ مقداری نگرفته‌اند (به عنوان مثال این مقدار، شیء‌ای نامرئی است). فرمولی برای تعداد ترکیبات ممکن مدل- تفسیر در دامنهٔ $D$ بیابید. در مورد حذف ترکیب‌های تکراری نگران نباشید.** ***(این تمرین مشابه تمرین ۵.۸ در ویرایش سوم است)***

☑ حل: ما از ساده‌ترین روش شمارش که در آن کاری با ترکیبات تکراری ندارد استفاده می‌کنیم. از آنجا که تعداد نمادهای ثابت $c$ بوده و دامنه هریک دارای $D$ عضو است، پس برای نمادهای ثابت تعداد $D^c$ انتساب وجود دارد(به عنوان مثال فرض کنید نمادهای ثابت $a$ و $b$ و $c$ هر یک دارای دامنه true و false باشند. پس می‌توان به تعداد $2^3$ حالت مختلف به آنها مقدار داد). هر مسند با درجهٔ $k$ به یک رابطهٔ $k$- تایی یعنی رابطه‌ای با $k$ آرگومان نگاشته می‌شود که تعداد کل حالات ممکن $D^k$ خواهد بود. تعداد زیرمجموعه‌های $k$- عضوی آن، همان تعداد مورد نظر ما می‌باشد که برابر $2^{D^k}$ است. هر نماد تابعی با درجهٔ $k$ به یک تابع $k$- تایی نگاشته می‌شود و از آنجا که در دامنه ورودی‌های تابع می‌توان علاوه بر عناصر دامنه، عنصر نامرئی نیز داشته باشیم پس تعداد اعضای دامنه را $D+1$ در نظر می‌گیریم. پس تعداد $(D+1)^{D^k}$ تابع مختلف وجود خواهد داشت. بنابراین تعداد کل ترکیبات ممکن عبارتست از:

$$D^c \cdot \left(\sum_{k=1}^{A} 2^{D^k}\right) \cdot \left(\sum_{k=1}^{A} (D+1)^{D^k}\right)$$

به دو نکته توجه کنید: اول آنکه این تعداد متناهی است و دوم آنکه حداکثر درجهٔ $A$ پیچیده‌ترین پارامتری است که باید تعیین شود.

**؟ ۶.۸ این جملات را با استفاده از واژگان سازگار (طبق تعریف شما) به منطق مرتبهٔ اول بازنمایی کنید: الف) برخی دانشجویان درس فرانسه را در بهار 2001 اخذ کردند. ب) هر دانشجویی که درس فرانسه را اخذ کند، آن را می‌گذراند. ج) فقط یک دانشجو درس یونانی را در بهار 2001 اخذ کرد. د) بهترین نمرهٔ درس یونانی همیشه بالاتر از بهترین نمرهٔ درس فرانسه است. هـ) هرکسی که یک بیمه‌نامه می‌خرد، باهوش است. و) هیچ‌کسی بیمه‌نامهٔ گران نمی‌خرد. ز) کارگزاری وجود دارد که فقط به کسانی که بیمه نیستند، بیمه‌نامه می‌فروشد. ح) آرایشگری وجود دارد که تمام مردان شهر که خودشان را اصلاح نمی‌کنند، اصلاح می‌کند. ط) شخصی که متولد انگلستان باشد و یکی از والدینش یا شهروند انگستان و یا مقیم در انگلستان باشد، به واسطهٔ زادگاهش شهروند انگلستان محسوب می‌شود. ی) شخصی که خارج از انگلستان متولد شود و یکی از والدینش به واسطهٔ زادگاه شهروند انگلستانی باشد، آنگاه به واسطهٔ نژادش شهروند انگلستان محسوب می‌شود. ک) سیاستمداران می‌توانند همیشه، برخی مردم را فریب دهند و در برخی اوقات، همهٔ مردم را ولی نمی‌توانند همهٔ مردم را همیشه فریب دهند.** ***(این تمرین مشابه تمرین ۲۴.۸ در ویرایش سوم است)***

☑ حل: در این تمرین بهتر است در مورد جزئیات زمان افعال و سازگار بودن مقایسه‌ها و غیره نگران نباشیم. نکتهٔ اصلی آن است که مطمئن شویم کلمات ربط، کمیت‌سنج و استفاده از مسند، تابع، ثوابت و هم‌ارزی را می‌دانیم. فرهنگ واژگان زیر را در نظر بگیرید:

$Takes(x,c,s)$ : دانشجوی $x$ درس $c$ را در نیمسال $s$ اخذ می‌کند.

$Passes(x,c,s)$ : دانشجوی $x$ درس $c$ را در نیمسال $s$ می‌گذراند.

$Score(x,c,s)$ : نمره کسب‌شده توسط دانشجوی $x$ در درس $c$ در نیمسال $s$ را نشان می‌دهد.

$x>y$: $x$ بزرگتر از $y$ است.

$F$ و $G$: دروس فرانسه و یونانی را نشان می‌دهد (هر یک از آنها می‌تواند به عنوان هر درس در جملات تفسیر شود در آنصورت یک مسند بنام $Subject(c,f)$ برای نشان دادن اینکه درس $c$ در رشتهٔ $f$ است، استفاده می‌شود).

$Buys(x,y,z)$: $x$ $y$ را از $z$ می‌خرد.

$Sells(x,y,z)$: $x$ $y$ را به $z$ می‌فروشند.

$Shaves(x,y)$ : شخص $x$ شخص $y$ را اصلاح می‌کند.

$Born(x,c)$ : شخص $x$ در کشور $c$ متولد می‌شود.

$Parent(x,y)$ : $x$ یکی از والدین $y$ است.

$Citizen(x,y,r)$ : $x$ یک شهروند کشور $y$ به دلیل $r$ می‌باشد.

$Resident(x,c)$ : $x$ مقیم کشور $c$ است.

$Fools(x,y,t)$ : شخص $x$ شخص $y$ را در زمان $t$ می‌فریبد.
$Politician(x), Smart(x), Insured(x), Agent(x), Expensive(x), Barber(x), Man(x), Person(x), Student(x)$:
مسندهایی هستند که برای اعضای دسته‌های مختلف استفاده می‌شود.

**الف)** برخی دانشجویان درس فرانسه را در بهار 2001 اخذ کردند.

$$\exists x \ Student(x) \wedge Takes(x, F, Spring2001)$$

**ب)** هر دانشجویی که درس فرانسه را اخذ کند، آن را می‌گذراند.

$$\forall x, s \ Student(x) \wedge Takes(x, F, s) \Rightarrow Passes(x, F, s)$$

**ج)** فقط یک دانشجو درس یونانی را در بهار 2001 اخذ کرد.

$$\exists x \ Student(x) \wedge Takes(x, G, Spring2001) \wedge \forall y \ y \neq x \Rightarrow \neg Takes(y, G, Spring2001)$$

**د)** بهترین نمرهٔ درس یونانی همیشه بالاتر از بهترین نمرهٔ درس فرانسه است.

$$\forall s \ \exists x \ \forall y \ Score(x, G, s) > Score(y, F, s)$$

**هـ)** هرکسی که یک بیمه‌نامه می‌خرد، باهوش است.

$$\forall x \ Person(x) \wedge (\exists y, z \ Policy(y) \wedge Buys(x, y, z)) \Rightarrow Smart(x)$$

**و)** هیچ‌کسی بیمه‌نامه گران نمی‌خرد.

$$\forall x, y, z \ Person(x) \wedge Policy(y) \wedge Expensive(y) \Rightarrow \neg Buys(x, y, z)$$

**ز)** کارگزاری وجود دارد که فقط به کسانی بیمه‌نامه می‌فروشد که بیمه نباشند.

$$\exists x \ Agent(x) \wedge \forall y, z \ Policy(y) \wedge Sells(x, y, z) \Rightarrow (Person(z) \wedge \neg Insured(z))$$

**ح)** آرایشگری وجود دارد که تمام مردان شهر که خودشان را اصلاح نمی‌کنند، اصلاح می‌کند.

$$\exists x \ Barber(x) \wedge \forall y \ Man(y) \wedge \neg Shaves(y, y) \Rightarrow Shaves(x, y)$$

**ط)** شخصی که متولد انگلستان باشد و یکی از والدینش یا شهروند انگلستان و یا مقیم در انگلستان باشد، به واسطهٔ زادگاهش شهروند انگلستان محسوب می‌شود.

$$\forall x \ Person(x) \wedge Born(x, UK) \wedge (\forall y \ Parent(y, x) \Rightarrow ((\exists r \ Citizen(y, UK, r)) \vee Resident(y, UK))) \Rightarrow Citizen(x, UK, Birth)$$

**ی)** شخصی که خارج از انگلستان متولد شود و یکی از والدینش به واسطهٔ زادگاه شهروند انگلستانی باشد آنگاه به واسطهٔ نژادش شهروند انگلستان محسوب می‌شود.

$$\forall x \ Person(x) \wedge \neg Born(x, UK) \wedge (\exists y \ Parent(y, x) \wedge Citizen(y, UK, Birth)) \Rightarrow Citizen(x, UK, Descent).$$

**ک)** سیاستمداران می‌توانند همیشه برخی از مردم را فریب دهند و در برخی اوقات همهٔ مردم را. ولی نمی‌توانند همهٔ مردم را همیشه فریب دهند.

$$\forall x \ Politician(x) \Rightarrow (\exists y \ \forall t \ Person(y) \wedge Fools(x, y, t)) \wedge (\exists t \ \forall y \ Person(y) \Rightarrow Fools(x, y, t)) \wedge \neg(\forall t \ \forall y \ Person(y) \Rightarrow Fools(x, y, t))$$

**؟ ۷.۸ جملهٔ «تمام آلمانی‌ها به زبان یکسانی صحبت می‌کنند» را به جبر مسندی بیان کنید. از Speaks(x,l) به معنای آنکه شخص $x$ به زبان $l$ صحبت می‌کند، استفاده کنید.**

☑ حل: نکته کلیدی آن است که لغت «یکسان» به هر جفت از لهجه‌های آلمانی اشاره می‌کند. چندین معادل منطقی برای بیان این جمله وجود دارد. ساده‌ترین آنها به شکل بند هورن عبارتست از:

$$\forall x, y, l \ German(x) \wedge German(y) \wedge Speaks(x, l) \Rightarrow Speaks(y, l)$$

**؟ ۸.۸ با توجه به حقایق Male(Jim) و Spouse(Jim,Laura)، قانونی بنویسید که حقیقت Female(Laura) نتیجه شود.**

☑ حل: $\forall x, y \ Spouse(x, y) \wedge Male(x) \Rightarrow Female(y)$

البته این قانون ممکن است در برخی ایالات و کشورها صادق نباشد.

**؟ ۹.۸ مجموعه‌ای کلی از حقایق و قوانین بنویسید که ادعای «ولینگتون خبر مرگ ناپلئون را شنید» را بازنمایی کند و پاسخ صحیح سؤال «آیا ناپلئون خبر مرگ ولینگتون را شنید؟» را بیابید. *(این تمرین مشابه تمرین ۲۵.۸ در ویرایش سوم است)***

☑ حل: این تمرین بسیار جنبهٔ آموزشی دارد ولی خیلی بدیهی نیست. مسندهای اصلی عبارتند از: $Heard(x,e,t)$ (x رخداد $e$ را در زمان $t$ شنید)، $Occurred(e,t)$ (رخداد $e$ در زمان $t$ اتفاق افتاد)، $Alive(x,t)$ ($x$ در زمان $t$ زنده است):

$\exists t \ \ Heard(W, DeathOf(N), t)$
$\forall x, e, t \ \ Heard(x, e, t) \Rightarrow Alive(x, t)$
$\forall x, e, t_2 \ \ Heard(x, e, t_2) \Rightarrow \exists t_1 \ \ Occurred(e, t_1) \wedge t_1 < t_2$
$\forall t_1 \ \ Occurred(DeathOf(x), t_1) \Rightarrow \forall t_2 \ \ t_1 < t_2 \Rightarrow \neg Alive(x, t_2)$
$\forall t_1, t_2 \ \neg(t_2 < t_1) \Rightarrow ((t_1 < t_2) \vee (t_1 = t_2))$
$\forall t_1, t_2, t_3 \ \ (t_1 < t_2) \wedge ((t_2 < t_3) \vee (t_2 = t_3)) \Rightarrow (t_1 < t_3)$
$\forall t_1, t_2, t_3 \ \ ((t_1 < t_2) \vee (t_1 = t_2)) \wedge (t_2 < t_3) \Rightarrow (t_1 < t_3)$

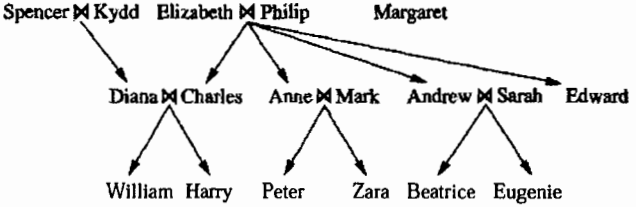
**؟ ۱۰.۸ حقایق گزاره‌ای دنیای وامپوز در بخش ۵.۷ را به منطق مرتبهٔاول بازنویسی کنید. این نسخه چه میزان فشرده‌تر خواهد بود؟**

☑ حل: می‌توان گفت برخی جملات نیازی به بازنویسی مجدد ندارند ولی یکی از آنها عبارتست از:

$$\forall s_1 \ \ Breezy(s_1) \Leftrightarrow \exists s_2 \ \ Adjacent(s_1, s_2) \wedge Pit(s_2)$$

این جمله می‌گوید در یک خانه نسیم است اگر و فقط اگر یک چاله در خانه‌های همسایه باشد. در حالت کلی، اندازهٔ مجموعهٔ قواعد، مستقل از اندازهٔ دنیای وامپوز است.

**؟ ۱۱.۸ اصولی بنویسید که مسندهای زیر را توصیف کند:**
**BrotherInLaw،Uncle،Aunt،Son،Daughter،Sister،Brother،GreatGrandparent،Grandchild،**
**FirstCousin،SisterInLaw. تعریف مناسبی برای mامین پسرخاله (دخترعمو یا پسردایی یا پسرعمه و....) بیابید که n بار جایگزین شده است و آن تعریف را به منطق مرتبهٔ اول بازگو کنید. سپس حقایقی در مورد شجره‌نامهٔ موجود در شکل ۵.۸ بنویسید. با استفاده از سیستم استدلال منطقی مناسب، تمام جملاتی که باید نوشته شود را به آن Tell نموده و سپس از آن Ask کنید که نوهٔ Elizabeth، برادرخوانده Diana و جد بزرگ Zara کیست؟** ***(این تمرین مشابه تمرین ۱۴.۸ در ویرایش سوم است)***




شکل ۵.۸ یک درخت خانواده رایج. نماد ⋈ نشان‌دهنده وصلت دو نفر و بردارها نشان‌دهنده فرزندان ماحصل از آن وصلت می‌باشد.

☑ حل: مطمئن شوید که در نوشتار خود از $\Leftrightarrow$ استفاده شده باشد. اگر شما از $\Rightarrow$ استفاده کنید، فقط قیودی را تحمیل کرده‌اید و تعریفی حقیقی ارائه نداده‌اید.

$GrandChild(c, a) \Leftrightarrow \exists b \ \ Child(c, b) \wedge Child(b, a)$
$GreatGrandParent(a, d) \Leftrightarrow \exists b, c \ \ Child(d, c) \wedge Child(c, b) \wedge Child(b, a)$
$Brother(x, y) \Leftrightarrow Male(x) \wedge Sibling(x, y)$
$Sister(x, y) \Leftrightarrow Female(x) \wedge Sibling(x, y)$
$Daughter(d, p) \Leftrightarrow Female(d) \wedge Child(d, p)$
$Son(s, p) \Leftrightarrow Male(s) \wedge Child(s, p)$
$AuntOrUncle(a, c) \Leftrightarrow \exists p \ \ Child(c, p) \wedge Sibling(a, p)$
$Aunt(a, c) \Leftrightarrow Female(a) \wedge AuntOrUncle(a, c)$
$Uncle(u, c) \Leftrightarrow Male(u) \wedge AuntOrUncle(a, c)$
$BrotherInLaw(b, x) \Leftrightarrow \exists m \ \ Spouse(x, m) \wedge Brother(b, m)$
$SisterInLaw(s, x) \Leftrightarrow \exists m \ \ Spouse(x, m) \wedge Sister(s, m)$
$FirstCousin(c, k) \Leftrightarrow \exists (\ p) AuntOrUncle(p, c) \wedge Parent(p, k)$

دومین پسرخاله، همان فرزند پدر و مادر اولین پسرخاله خواهد بود به همین ترتیب تعریف $n$ امین پسرخاله عبارتست از:

$$NthCousin(n,c,k) \Leftrightarrow \exists p,f \;\; Parent(p,c) \wedge NthCousin(n-1,f,p) \wedge Child(k,f)$$

حقایق موجود در شجره‌نامه بسیار ساده هستند:

هر بردار نشان‌دهندهٔ یک *Child* است.(به عنوان مثال *Child*(*William*,*Diana*) و *Child*(*William*,*Charles*)). هر نام نشان‌دهندهٔ یک گزارهٔ جنسی است (به عنوان مثال *Male*(*William*) یا *Female*(*Diana*)) هر خط دوتایی نشان‌دهندهٔ یک گزارهٔ ازدواج یا *Spouse* (به عنوان مثال *Spouse*(*Charles*,*Diana*)). پرس‌وجو از سیستم استدلال منطقی یکی از راه‌های اشکال‌زدایی تعاریف است.

**؟ ۱۲.۸ جمله‌ای بنویسید که در آن «+» را به عنوان یک تابع جابجایی‌پذیر توصیف کند. آیا جملهٔ شما از اصل پانو نتیجه می‌شود؟ اگر اینطور است توضیح دهید چرا و اگر نیست مدلی مثال بزنید که در آن اصل‌موضوعه درست باشد و جملهٔ شما غلط.**

☑ حل: $\forall x,y \; (x+y)=(y+x)$ این جمله از اصل‌موضوعه‌پانو نتیجه می‌شود (می‌توان اولین اصل برای «+» را به صورت $\forall m \; NatNum(m) \Rightarrow +(0,m)=m$ نوشت). تعریف + می‌گوید $x+y=S^x(y)=S^{x+y}(0)$ که منظور از $S^x$، همان تابع s است که x بار اعمال شود و به طور خلاصه آن را چنین بیان نمودیم. به طور مشابه $y+x=S^y(y)=S^{x+y}(0)$. از این رو اصول می‌گویند که $x+y$ و $y+x$ عبارات یکسانی هستند. می‌توان توسط روش استقرا آن را به طور رسمی اثبات کرد.

**؟ ۱۳.۸ تعریف زیر برای مسند عضویت مجموعه، $\in$ ، پیشنهاد شده است. توضیح دهید که چه مشکلی دارد:**

$$\forall x,y \;\; x \in \{x|s\}$$
$$\forall x,s \;\; x \in s \Rightarrow \forall \; y \;\; x \in \{y|s\}$$

***(این تمرین مشابه تمرین ۱۵.۸ در ویرایش سوم است)***

☑ حل: اگر چه این اصول برای اثبات زیرمجموعه‌بودن در زمانی که $x$ واقعا یک عضو از مجموعه باشد، کافی است ولی در مورد مواقعی که $x$ یک عضو نباشد هیچ چیزی نمی‌گوید. به عنوان مثال، نمی‌توان اثبات کرد که $x$ عضوی از مجموعهٔ تهی نیست. این اصول ممکن است برای یک سیستم منطقی نظیر پرولوگ مناسب باشد که از تناقضات به عنوان شکست استفاده می‌کند.

**؟ ۱۴.۸ با استفاده از اصول‌مجموعه به عنوان مثال، اصولی برای دامنهٔ فهرست شامل تمام ثابت‌ها، توابع و مسندهایی بنویسید که در این فصل بیان شد.** ***(این تمرین مشابه تمرین ۱۶.۸ در ویرایش سوم است)***

☑ حل: منظور از *List*? به معنای فهرست مناسب در اصطلاحات *Lisp* است. به عنوان مثال ساختاری که در سمت راست‌ترین رقم خود *Nil* داشته باشد.

$$List?(Nil)$$
$$\forall x,l \;\; List?(l) \Leftrightarrow List?(Cons(x,l))$$
$$\forall x,y \;\; First(Cons(x,y)) = x$$
$$\forall x,y \;\; Rest(Cons(x,y)) = y$$
$$\forall x \;\; Append(Nil,x) = x$$
$$\forall v,x,y,z \;\; List?(x) \Rightarrow (Append(x,y) = z \Leftrightarrow Append(Cons(v,x),y) = Cons(v,z))$$
$$\forall x \;\; \neg Find(x,Nil)$$
$$\forall x \;\; List?(z) \Rightarrow (Find(x,Cons(y,z)) \Leftrightarrow (x = y \vee Find(x,z))$$

**؟ ۱۵.۸ تعریف زیر برای خانه‌های مجاور در دنیای وومپوز، پیشنهاد شده است. توضیح دهید که چه مشکلی دارد:**

**$\forall x,y$ Adjacent ([x,y],[x+1,y])$\wedge$Adjacent ([x,y],[x,y+1])**

***(این تمرین مشابه تمرین ۱۷.۸ در ویرایش سوم است)***

☑ حل: این تعریف پیشنهادی چندین مشکل دارد. یکی از آنها این است که طبق این تعریف، عبارت:

Adjacent([1,1],[1,2])

این دو خانه را مجاور معرفی می‌کند در حالیکه Adjacent([1,2],[1,1]) طبق این تعریف برقرار نیست و مجاور نیستند در حالیکه در دنیای وومپوز مجاورند. لذا بایستی یک اصل برای تقارن عبارات اضافه شود. همچنین نمی‌توان با این تعریف، نادرستی عبارت Adjacent([1,1],[1,3]) را اثبات نمود. بنابراین بایستی آن را به صورت زیر نوشت:

$\forall \; s_1,s_2 \Leftrightarrow \ldots$

در نهایت، این تعریف تمام محدودیت‌های دنیا را شامل نمی‌شود و بایستی تعدادی شرایط به آن اضافه شود.

**؟ ۱۶.۸ با استفاده از نماد ثابت *Wumpus* و یک مسند دودویی *In*(*Wumpus*,*Location*) اصولی جهت استدلال در مورد مکان وومپوز بنویسید. به خاطر داشته باشید که فقط یک وومپوز در دنیا وجود دارد.** ***(این تمرین مشابه تمرین ۱۸.۸ در ویرایش سوم است)***

☑ حل: به این جملات نیاز داریم:

$$\forall s_1 \;\; Smelly(s_1) \Leftrightarrow \exists s_2 \;\; Adjacent(s_1,s_2) \wedge In(Wumpus,s_2)$$
$$\exists s_1 \;\; In(Wumpus,s_1) \wedge \forall s_2 \;\; (s_1 \neq s_2) \Rightarrow \neg In(Wumpus,s_2)$$

☑ حل: گیت‌های پایه باید توسط هم‌ارزهای منطقی تعریف شوند تا در زمان ترکیب آنها از صحت عملکرد این عناصر پایه مطمئن باشیم. اگر از یک سیستم برنامه‌سازی منطقی استفاده می‌کنید، به سادگی می‌توان این موارد را لیست نمود.
به عنوان مثال:

$$AND(0,0,0) \qquad AND(0,1,0) \qquad AND(1,0,0) \qquad AND(1,1,1)$$

برای یک جمع‌کنندهٔ یک بیتی داریم:

$$\begin{aligned}\forall i_1, i_2, i_3, o_1, o_2 \;\; & Add_1\, Circuit(i_1, i_2, i_3, o_1, o_2) \Leftrightarrow \\ \exists o_{x1}, o_{a1}, o_{x2} \;\; & XOR(i_1, i_2, o_{x1}) \wedge XOR(o_{x1}, i_3, o_1) \\ & \wedge AND(i_1, i_2, o_{a1}) \wedge AND(i_3, o_{x1}, o_{a2}) \\ & \wedge OR(o_{a2}, o_{a1}, o_2)\end{aligned}$$

سؤالی که جهت اعتبارسنجی مفید است عبارتست از:

$$\forall i_1, i_2, i_3, o_1, o_2 \;\; Add_1(i_1, i_2, i_3) = [, o_1, o_2] \;\Rightarrow\; Add_1\, Circuit(i_1, i_2, i_3, o_1, o_2)$$

نمی‌توان سؤال پرسید که چه ترمینال‌های خاصی در یک مدار موجود قرار گرفته‌اند زیرا ترمینال‌ها جسمیت ندارند (خود مدار واقعی را نداریم).

**؟ ۱۹.۸ یک فرم درخواست پاسپورت در کشورتان را بدست آورده و قواعد مربوط به واجدین شرایط آن را مطالعه نمایید. سپس طبق مراحل گفته شده در بخش ۴.۸، آن قواعد را به منطق مرتبهٔ اول تبدیل کنید.** ***(این تمرین مشابه تمرین ۲۷.۸ در ویرایش سوم است)***

☑ حل: پاسخ این سؤال در کشورهای مختلف متفاوت خواهد بود ولی دو قاعدهٔ اصلی برای پاسپورت‌های ایالت متحده را می‌توانید در پاسخ تمرین ۶.۸ بیابید.

# فصل ۹ (ویرایش دوم)

**۹.۱ با استفاده از اصول اولیه، اثبات کنید که نمونه‌سازی عمومی صحیح است و نمونه‌سازی وجودی، پایگاه‌دانشی تولید می‌کند که در استنتاج هم‌ارز است. *(این تمرین مشابه تمرین ۱.۹ در ویرایش سوم است)***

حل: در نمونه‌سازی عمومی، به جای متغیری که در کنار سور عمومی آمده است، تمام اصطلاحات موجود در محیط را جایگزین کرده و به ازای هریک جمله‌ای جدید تولید می‌شود. برای اثبات صحت اینکار، باید نشان دهیم که هر عبارت به شکل $\forall v\ a$ می‌تواند تمام نمونه‌سازی‌های عمومی ایجاد شده از آنرا ایجاب کند. عبارت $\forall v\ a$ درصورتی درست نامیده می‌شود که در تمام تفسیرهای ممکن صحیح باشد. به عنوان مثال $v$ را با یک اصطلاح پایه‌ای $g$ جایگزین می‌کنیم که یک تفسیر ممکن است. بنگاه اگر عبارت اصلی صحیح باشد، عبارت نمونه‌سازی شده نیز باید صحیح باشد.

**۹.۲ طبق عبارت $Likes(Jerry,Icecream)$ منطقی است که عبارت $\exists x\ Likes(x,IceCream)$ را نتیجه بگیریم. یک قاعدهٔ کلی به نام «معرفی وجودی» بنویسید که این نتیجه‌گیری را تایید کند. شرایطی که متغیرها و اصطلاحات این قاعده باید رعایت کنند را به دقت بیان کنید. *(این تمرین مشابه تمرین ۲.۹ در ویرایش سوم است)***

حل: برای هر عبارت $\alpha$ که شامل اصطلاح پایه‌ای $g$ بوده و برای هر متغیر $v$ که در $\alpha$ نباشد داریم:

$$\frac{\alpha}{\exists v\ \text{SUBST}_1(\{g/v\},\alpha)}$$

که $\text{SUBST}_1$ تابعی است که یک رخداد $g$ را با $v$ جایگزین می‌کند.

**۹.۳ فرض کنید یک پایگاه‌دانش تنها شامل یک عبارت به صورت $\exists x\ AsHighAs(x,Everest)$ می‌باشد. کدامیک از این موارد نتیجهٔ صحیح اعمال نمونه‌سازی وجودی می‌باشد: الف) $AsHighAs(Everest,Everest)$ ب) $AsHighAs(Kilimanjaro,Everest)\wedge AsHighAs(BenNevis,Everest)$ ج) (پس از اعمال دوبار) *(این تمرین مشابه تمرین ۳.۹ در ویرایش سوم است)***

حل: هردو مورد ب و ج معتبر می‌باشند زیرا در طی عملیات نمونه‌سازی وجودی باید نامی انتخاب شود که در هیچ جای دیگاه استفاده نشده باشد. مورد الف نامعتبر است زیرا از نماد $Everest$ که قبلا بکار رفته بود، استفاده کرده است. دقت کنید که قسمت ج نمی‌گوید دو کوه هم‌ارتفاع با $Everest$ داریم زیرا در هیچ جای آن نگفته است که نمادهای $BenNevis$ و $Kilimanjaro$ متفاوت هستند.

**۹.۴ برای هریک از جملات ساده زیر، در صورت وجود، عمومی‌ترین یکسان‌ساز را بیان کنید.**
**الف) $P(x,y,z)$ ب) $P(A,B,B),P(x,y,z)$ ج) $Q(y,G(A,B)),Q(G(x,x),y)$ د) $Older(Father(y),y),Older(Father(x),John)$ ه) $Knows(Father(y),y),Knows(x,x)$ *(این تمرین مشابه تمرین ۴.۹ در ویرایش سوم است)***

حل: منظور از یکسان‌سازی آن است که دو جمله را گرفته و برای آنها یک مجموعه جایگزینی بیابیم که با اعمال آن، هر دو جمله منطقا یکسان باشند. اگر دو جمله داشته باشیم که بتوان به روش‌های مختلف آنها را یکسان‌سازی نمود (چندین مجموعه جایگزینی ممکن باشد)، قانون عمومی‌ترین یکسان‌ساز، موردی را انتخاب می‌کند که کلی‌تر بوده و محدودیت کمتری برای متغیرها ایجاد نماید. **الف)** $\{x/A\ ,\ y/B\ ,z/B\}$ (ویا سایر جایگشت‌های مشابه). **ب)** یکسان‌سازی نمی‌شود (یک متغیر می‌تواند برای هر دو نماد $A$ و $B$ بکار رود). **ج)** $\{y/John\ ,\ x/John\}$ **د)** یکسان‌سازی نمی‌شود (زیرا «بررسی رخدادها» مانع از یکسان‌سازی $y$ با $Father(y)$ می‌شود. این روش بررسی می‌کند که یک متغیر با عبارتی شامل خود متغیر یکسان‌سازی نشود پس $y/Father(y)$ اشتباه است.)

**۹.۵ شبکه شمول شکل ۲.۹ را در نظر بگیرید: الف) برای عبارت $(Employs(Mother(John)\ ,\ Father(Richard))$ شبکه شمولی تشکیل دهید. ب) برای عبارت $Employs(IBM,y)$ که نشان‌دهندهٔ آن است که «همه برای $IBM$ کار می‌کنند»، شبکه شمولی تشکیل دهید. به خاطر داشته باشید که تمام پرس‌وجوهایی که با این عبارت یکسان‌سازی می‌شوند را به حساب آورید. ج) فرض کنید $STORE$ باعث می‌شود تا هرجمله به گره‌ای در شبکه شمول مرتبط شود. توضیح دهید که اگر برخی از این جملات حاوی متغیرهایی باشند، $FETCH$ چگونه باید کارکند به عنوان مثال از عبارات قسمت الف و ب و پرس‌وجوی $Employs(x,Father(x))$ استفاده کنید. *(این تمرین مشابه تمرین ۵.۹ در ویرایش سوم است)***

حل: برای ترسیم شبکه شمول قسمت الف به صورت شکل ۲.۹ نیاز به فضای زیادی داریم. لذا در اینجا به علت کمبود فضا آنرا به صورت گره‌های فرزند در زیر هم ترسیم می‌کنیم:

```
[1] Employs(x, y)
  [2] Employs(x, Father(z))
```

```
[3] Employs(x, Father(Richard))
  [4] Employs(Mother(w), Father(Richard))
    [5] Employs(Mother(John), Father(Richard))
[6] Employs(Mother(w), Father(z))
  [4] ...
  [7] Employs(Mother(John), Father(z))
    [5] ...
  [8] Employs(Mother(w), y)
    [9] Employs(Mother(John), y)
      [10] Employs(Mother(John), Father(z)
        [5] ...
    [6] ...
```

شبکه شمول یک عبارت، شامل تمام پرس‌وجوهایی است که می‌توانند با آن یکسان‌سازی شوند. در شبکه شمول، پایین‌ترین گره همان چیزی است که سئوال از ما خواسته است. یک سطح بالاتر کمی کلی تر بوده و تنها با یک جایگزینی در این سطح، به سطح بعد می‌رسیم. اینکار آنقدر تا مراحل بالا ادامه می‌یابد که به عمومی‌ترین حالت عبارت برسیم. یعنی بالاترین نسل مشترک هر دو گره در این شبکه، نتیجه اعمال عمومی‌ترین یکسان‌ساز است. **ب)** برای عبارت قسمت ب، دو گره $Employs(x,y)$ و $Employs(y,y)$ خواهیم داشت. **ج)** ابتدا بایستی عبارات $Employs(Mother(John)$ و $Father(Richard)$ و $Employs(IBM,y)$ واکشی شوند تا بتوانیم به سئوال $Employs(x,Father(x))$ پاسخ دهیم. که جواب آن $Richard$ $Father$ یعنی «پدر ریچارد» می‌گردد.

**؟ ۶.۹ فرض کنید بخشی از اطلاعات آماری مردم ایالت متحده شامل سن، شهرسکونت، تاریخ تولد و نام مادر هر شخص را درون پایگاه‌داده منطقی قرارداده‌ایم که شماره امنیت اجتماعی هرشخص یکتا بوده و جهت شناسایی او بکار می‌رود. بنابراین سن George به صورت $Age(443\text{-}65\text{-}1285\text{-}56)$ بیان می‌شود. که عدد درون پرانتز همان شماره امنیت اجتماعی جرج است.کدامیک از راه‌حل‌های $S_1$ تا $S_5$ برای پاسخ به سوالات $Q_1$ تا $Q_4$ راه‌حلی مناسب است؟ (با فرض استفاده از روش زنجیره ای پس گرد نرمال).**

**• $S_1$: شاخصی برای هر بسیط در هرموقعیت • $S_2$: شاخصی برای هر آرگومان اولیه • $S_3$: شاخصی برای هر مسند بسیط • $S_4$: شاخصی برای هر ترکیب مسند و آرگومان اولیه • $S_5$: شاخصی برای هر ترکیب مسند و آرگومان ثانویه و شاخصی برای هر آرگومان اولیه (غیراستاندارد) • $Q_1$: $Age(443\text{-}44\text{-}4321,x)$ • $Q_2$: $ResidesIn(x,Houston)$ • $Q_3$: $Mother(x,y)$ • $Q_4$: $Age(x,34) \wedge ResidesIn(x,TinyTownUSA)$ *(این تمرین مشابه تمرین ۱۸.۹ در ویرایش سوم است)***

☑ حل: در جدول زیر پیچیدگی زمانی در حالت متوسط برای هر ترکیب پرس‌وجو/راه‌حل بیان شده است. (منظور از «$n$;1» آن است که برای یافتن اولین راه‌حل به $O(1)$ و یافتن تمام راه‌حل‌ها به $O(n)$ نیازمندیم). فرضیه زیر را در نظر بگیرید: دسترسی به جدول، پیچیدگی $O(1)$ را نیاز دارد. $n$ فرد در بانک اطلاعاتی موجود است. $O(n)$ فرد با سنی مشخص داریم. هر فرد یک مادر دارد. $H$ فرد در $Houston$ و $T$ فرد در $TinyTown$ وجود دارند. $T$ خیلی کمتر از $n$ است در پرس وجوی $Q_4$ بخش دوم در ابتدا ارزیابی می‌شود.

| | Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|---|
| S1 | 1 | 1; $H$ | 1; $n$ | $T$; $T$ |
| S2 | 1 | $n$; $n$ | 1; $n$ | $n$; $n$ |
| S3 | $n$ | $n$; $n$ | 1; $n$ | $n^2$; $n^2$ |
| S4 | 1 | $n$; $n$ | 1; $n$ | $n$; $n$ |
| S5 | 1 | 1; $H$ | 1; $n$ | $T$; $T$ |

هرچیزی با $O(1)$ وحتی $O(t)$ می‌تواند کارا محسوب شود. توجه داشته باشید که $S_1$ و $S_5$ سایر روش‌ها برای این مجموعه پرس‌وجو را نشان می‌دهند. همچنین دقت کنید که شاخص‌گذاری مسندها هیچ نقشی در جدول ایفا نمی‌کند.(مگر درترکیب با یک آرگومان) زیرا فقط سه مسند داریم (که پیچدگی آن $O(1)$ است) که در فاکتور ثابت اندکی تغییر ایجاد می‌کند.

**؟ ۷.۹ به نظر می‌رسد که بتوان از مشکل تناقض متغیرها در طی عملیات زنجیره‌ای پیش‌رو برای استانداردسازی تمام جملات پایگاه جلوگیری کرد. نشان دهید که این شیوه برای برخی جملات کارساز نیست. (راهنمایی: جمله‌ای را در نظر بگیرید که یک بخش از آن با بخش دیگر یکسان‌سازی شود) *(این تمرین مشابه تمرین ۱۲.۹ در ویرایش سوم است)***

☑ حل: در صورتی این شیوه کارساز خواهد بود که در پایگاه دانش، قواعد بازگشتی نداشته باشیم. فرض کنید پایگاه دانشی شامل جملات زیر داشته باشیم:

$$Member(x,[x|r])$$
$$Member(x,r) \Rightarrow Member(x,[y|r])$$

تون پرس‌وجوی $Member(3,[1,2,3])$ را با روش زنجیره‌ای پس‌رو اعمال کنید. توسط جایگزین‌های $\theta=\{x/3,y/1,r/[2,3]$ پرس‌وجو را یکسان‌سازی می‌کنیم. سپس آن را به بخش سمت چپ اعمال می‌کنیم تا عبارت $Member(3,[2,3$ حاصل شود و سعی می‌کنیم با زنجیره‌ای پس‌رو به جایگزین $\theta$ بازگردیم. زمانی که سعی در اعمال مجدد یم با شکست مواجه می‌شویم زیرا $y$ نمی‌تواند هر دو مقدار 1 و 2 را درخود جای دهد. به عبارت دیگر اگر در جایی که قواعد گشتی برای یک راه‌حل نیاز است از استانداردسازی اجزا استفاده کنیم شکست در استانداردسازی یک بخش، موجب شکست آن مورد می‌شود.

**۸.۹ توضیح دهید که چگونه می‌توان مساله 3-SAT با هراندازهٔ دلخواه را به صورت یک بند متناهی مرتبهٔ اول و حداکثر 30 قعیت پایه بیان نمود. *(این تمرین مشابه تمرین ۸.۹ در ویرایش سوم است)***

] حل: یک مساله 3-SAT به فرم زیر را در نظر بگیرید:

$$(x_{1,1} \vee x_{2,1} \vee x_{3,1}) \wedge (x_{1,2} \vee x_{2,2} \vee x_{3,2}) \vee \ldots$$

ن مساله را به عنوان یک بند متناهی منفرد به شکل زیر بازنویسی می‌کنیم که حاوی چند بند پایه است:

$$A \wedge B \wedge C \wedge \ldots \Rightarrow Z,$$

ی‌توانیم طبق بند متناهی:

$$OneOf(x_{1,1}, x_{2,1}, x_{3,1}) \wedge OneOf(x_{1,2}, x_{2,2}, x_{3,2}) \wedge \ldots \Rightarrow Solved$$

بندهای پایه‌ای:

$$OneOf(True, x, y)$$
$$OneOf(x, True, y)$$
$$OneOf(x, y, True)$$

ن موضوع را بیان نماییم.

**۹.۹ برای جملات زیر بازنمایی منطقی بنویسید که قابل استفاده در قیاس‌استثنایی تعمیم‌یافته باشد. الف) اسب‌ها، گاوها و وک‌ها پستاندار هستند. ب) یک کره اسب، یک اسب است. ج) ریش آبی یک اسب است. د) ریش آبی والد چارلی است. هـ) ه بودن و والد بودن، روابطی معکوس هستند. و) هر پستاندار یک والد دارد. *(این تمرین مشابه تمرین ۶.۹ در ویرایش سوم است)***

] حل: در اینجا با استفاده از علم موجودات و هستی‌شناسی، مثال‌ها را بسیار ساده‌تر می‌کنیم:

ب)
$$Horse(x) \Rightarrow Mammal(x)$$
$$Cow(x) \Rightarrow Mammal(x)$$
$$Pig(x) \Rightarrow Mammal(x)$$

$$Offspring(x, y) \wedge Horse(y) \Rightarrow Horse(x)$$ (

$$Horse(Bluebeard)$$ (
$$Parent(Bluebeard, Charlie)$$

$$Offspring(x, y) \Rightarrow Parent(y, x)$$
$$Parent(x, y) \Rightarrow Offspring(y, x)$$

$$Offspring(x, y) \Leftrightarrow Parent(y, x)$$ (ب

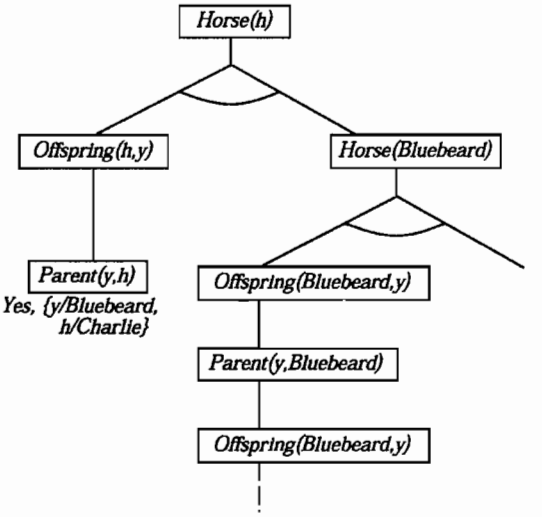
وجه داشته باشید که نمی‌توانیم قاعده‌ای به صورت $Offspring(x,y) \Leftrightarrow Parent(y,x)$ داشته باشیم زیرا به شکل قاعدهٔ س استثنایی تعمیم‌یافته نیست).

$Mammal(x) \Rightarrow Parent(G(x),x)$ (د $G$ در اینجا یک تابع اسکولم است).

**۱۰.۹ در این تمرین از جملاتی که شما در تمرین ۹.۹ نوشتید استفاده می‌کنیم تا با استفاده از الگوریتم زنجیره‌ای پس‌رو به سوال پاسخ دهیم. الف) درخت اثبات تولید شده توسط الگوریتم زنجیره‌ای پس‌رو برای سوال $\exists h \ Horse(h)$ را ترسیم ید که در آن، بندها طبق ترتیب داده شده منطبق باشند. ب) دربارهٔ این دامنه به چه چیزی باید توجه کنید. ج) طبق جملات ما، در حقیقت چه تعداد راه‌حل برای h وجود دارد؟ د) فکر می‌کنید بتوانید تمام آنها را بیابید؟ (راهنمایی: شاید نیاز به تفاده از مقالهٔ Smith (1986) داشته باشید). *(این تمرین مشابه تمرین ۱۳.۹ در ویرایش سوم است)***

] حل: این تمرین با حلقه‌ها در اثبات زنجیره‌ای پس‌رو سروکار دارد. یک حلقه، محدود به رخدادی است که یک زیرهدف با یگذاری در یکی از هدف‌های موجود در پشته بدست می‌آید. البته تمام حلقه‌ها به این شیوه به دست نمی‌آیند و برای حل این نه مسائل راه‌هایی داریم. **الف)** درخت اثبات را در شکل ۱.۹s مشاهده می‌کنید. شاخه‌های $Parent(y,Bluebeard)$ و

$Offspring(Bluebeard,y)$ به طور بی‌نهایت تکرار می‌شوند مابقی درخت هرگز به دست نمی‌آید. **ب)** بـه خـاطر قاعـدهٔ ب یعنی $Offspring(x,y) \wedge Horse(y) \Rightarrow Horse(x)$ یک حلقهٔ بی‌نهایت داریم. به خـاطر ترتیبـی کـه بـرای بنـدها در نظـر گرفتیم حلقه‌ای که در شکل مشخص است رخ می‌دهد. بهتر بود بند $Horse(Bluebeard)$ قبل از قاعدهٔ ب قرار می‌گرفت. به هرحال اگر بخواهیم برای اثبات تئوری از تمام راه‌حل‌ها استفاده کنیم، رخداد یک حلقه، ربطی به ترتیب قواعد ندارد. **ج)** باید به گونه‌ای باشد که اثبات نماید هر دوی *Charlie* و *Bluebeard* اسب هستند. **د)** Smith (1986) در مقالهٔ خود روش زیر را پیشنهاد می‌دهد: زمانی که یک هدف حلقوی رخ می‌دهد (که آن هدف یک نمونه جایگذاری در یک ابرهدف بالای پشته است) سعی کنید موقتا برای اثبات آن زیرهدف، آن را کنار بگذارید. با سایر شاخه‌های اثبات ابرهدف ادامه دهید و به راه‌حل‌هـا دسـت یابید. سپس این راه‌حل‌ها را به عنوان راه‌حل‌های همان زیرهدف‌های کنارگذاشته در نظر بگیرید و با شاخه‌های اثبات ادامه دهید تـا درصـورت وجـود بتوانیـد راه‌حـل‌هـای بیـشتری بیابیـد. طبـق اثبـات نمـایش داده شـده در شـکل، عبـارت $Offspring(Bluebeard,y)$ یک هدف تکراری است و باید کنار گذاشته شود. از آنجا که هیچ راه دیگری برای اثبـات وجـود آن نداریم. آن شاخه با شکست پایان می‌یابد. در این مورد روش Smith جهت یافتن هر دو راه‌حل مناسب است.




شکل ۱.۹s قسمتی از درخت اثبات برای یافتن اسبها

**؟ ۱۱.۹ یکی از معروف‌ترین معماهای کودکان عبارتست از: «من برادر و خواهری ندارم ولی پدرآن مرد پسر پدر من است». بـا استفاده از قواعد دامنهٔ خویشاوندی (فصل ۸) نشان دهید که «آن مرد چه کسی است؟». می‌توانید از روش‌های استنتاج گفتـه شده در این فصل نیز استفاده کنید. به نظر شما چرا این معما دشوار است؟** ***(این تمرین مشابه تمرین ۱۰.۹ در ویرایش سوم است)***

☑ حل: دشوارترین بخش این معما مربوط به جملهٔ «آن مرد چه کسی است» می‌شود. می‌خواهیم سوال «**چه** رابطه خویشاوندی بین آن مرد و شخص شناخته شده در معما وجود دارد؟ » را بپرسیم ولی اگر این رابطه خویشاوندی به صورت یک مسند ماننـد $Parent(x,y)$ در نظر بگیریم، آنگاه نمی‌توان رابطهٔ خویشاوندی را به صورت یک متغیر در منطق مرتبهٔ اول بیان نمود. بنابراین ناچاریم به روابط خویشاوندی جسمیت داده و آنها را به صورت شی در نظر بگیریم. از عبارت $Rel(r,x,y)$ برای بیان ایـن نکتـه استفاده می‌شود که رابطهٔ خویشاوندی $r$ بین فرد $y$ و $x$ وجود دارد. برای معرفی خود شخص از $Me$ و معرفی آن مـرد از $MrX$ استفاده می‌شود. همچنین از ثابت اسکولم $FM$ برای معرفی پدر شخص (پدر $Me$) و از $FX$ برای معرفـی پـدر آن مـرد (پـدر $MrX$) استفاده می‌کنیم. حقایق این مساله به شکل نرمال استلزامی عبارت است از:

(1) $Rel(Sibling, Me, x) \Rightarrow False$
(2) $Male(MrX)$
(3) $Rel(Father, FX, MrX)$
(4) $Rel(Father, FM, Me)$
(5) $Rel(Son, FX, FM)$

باید نشان دهیم که $Me$ تنها پسر پدرم می‌باشد و $MrX$ پدر $Me$ است که آن مذکر بوده و بنابراین آن مرد، پسر من می‌باشـد. این تعاریف طبق حوزهٔ روابط خویشاوندی عبارت است از:

(6) $Rel(Parent, x, y) \wedge Male(x) \Leftrightarrow Rel(Father, x, y)$
(7) $Rel(Son, x, y) \Leftrightarrow Rel(Parent, y, x) \wedge Male(x)$
(8) $Rel(Sibling, x, y) \Leftrightarrow x \neq y \wedge \exists p \ Rel(Parent, p, x) \wedge Rel(Parent, p, y)$
(9) $Rel(Father, x_1, y) \wedge Rel(Father, x_2, y) \Rightarrow x_1 = x_2$

و سوالی که ما داریم عبارت است از: $(Q)\ Rel(r, MrX, y)$

به پاسخ $\{r/Son, y/Me\}$ می‌رسیم. با ترجمه جملات 1 تا 9 و $Q$ به فرم INF و (نقیض $Q$ و تعریف $\neq$) داریم:

$(6a)\ Rel(Parent, x, y) \wedge Male(x) \Rightarrow Rel(Father, x, y)$
$(6b)\ Rel(Father, x, y) \Rightarrow Male(x)$
$(6c)\ Rel(Father, x, y) \Rightarrow Rel(Parent, x, y)$
$(7a)\ Rel(Son, x, y) \Rightarrow Rel(Parent, y, x)$
$(7b)\ Rel(Son, x, y) \Rightarrow Male(x))$
$(7c)\ Rel(Parent, y, x) \wedge Male(x) \Rightarrow Rel(Son, x, y)$
$(8a)\ Rel(Sibling, x, y) \Rightarrow x \neq y$
$(8b)\ Rel(Sibling, x, y) \Rightarrow Rel(Parent, P(x, y), x)$
$(8c)\ Rel(Sibling, x, y) \Rightarrow Rel(Parent, P(x, y), y)$
$(8d)\ Rel(Parent, P(x, y), x) \wedge Rel(Parent, P(x, y), y) \wedge x \neq y \Rightarrow Rel(Sibling, x, y)$
$(9)\ Rel(Father, x_1, y) \wedge Rel(Father, x_2, y) \Rightarrow x_1 = x_2$
$(N)\ True \Rightarrow x = y \vee x \neq y$
$(N')\ x = y \wedge x \neq y \Rightarrow False$
$(Q')\ Rel(r, MrX, y) \Rightarrow False$

توجه کنید که (۱) به شکل هورن نیست بنابراین باید از رزولوشن استفاده کنیم تا از دستیابی به راه‌حل مطمئن شویم. همچنین نشان می‌دهد که بایستی از مدولاسیون معکوس برای سروکار داشتن با تساوی استفاده کنیم. در ادامه لیستی از مراحل اثبات را می‌بینید که جواب هر مرحله درون پرانتز بیان شده است:

| | |
|---|---|
| $(10)\ Rel(Parent, FM, Me)$ | $(4, 6c)$ |
| $(11)\ Rel(Parent, FM, FX)$ | $(5, 7a)$ |
| $(12)\ Rel(Parent, FM, y) \wedge Me \neq y \Rightarrow Rel(Sibling, Me, y)$ | $(10, 8d)$ |
| $(13)\ Rel(Parent, FM, y) \wedge Me \neq y \Rightarrow False$ | $(12, 1)$ |
| $(14)\ Me \neq FX \Rightarrow False$ | $(13, 11)$ |
| $(15)\ Me = FX$ | $(14, N)$ |
| $(16)\ Rel(Father, Me, MrX)$ | $(15, 3, demodulation)$ |
| $(17)\ Rel(Parent, Me, MrX)$ | $(16, 6c)$ |
| $(18)\ Rel(Son, MrX, Me)$ | $(17, 2, 7c)$ |
| $(19)\ False\ \{r/Son, y/Me\}$ | $(18, Q')$ |

**؟ ۱۲.۹ اجرای الگوریتم زنجیره‌ای پس‌رو را در شکل ۶.۹ برای حل مساله جرم، دنبال کنید. دنباله مقادیر کسب‌شده توسط متغیرهای هدف را نشان داده و آنها را در یک درخت مرتب کنید.** ***(این تمرین مشابه تمرین ۱۴.۹ در ویرایش سوم است)***

☑ حل: یک درخت هدف را مشاهده می‌کنید:

```
goals = [Criminal(West)]
  goals = [American(West), Weapon(y), Sells(West, y, z), Hostile(z)]
  goals = [Weapon(y), Sells(West, y, z), Hostile(z)]
    goals = [Missle(y), Sells(West, y, z), Hostile(z)]
    goals = [Sells(West, M1, z), Hostile(z)]
      goals = [Missle(M1), Owns(Nono, M1), Hostile(Nono)]
      goals = [Owns(Nono, M1), Hostile(Nono)]
    goals = [Hostile(Nono)]
goals = []
```

**؟ ۱۳.۹ کد پرولوگ زیر یک مسند P را تعریف می‌کند:**

```
P(X, [X|Y])
P(X, [Y|Z]) :- P(X, Z)
```

**الف) برای سوالات P(2,[1,A,3]| و P(A,|1,2,3]) درخت اثبات و راه‌حل‌ها را نمایش دهید. ب) کدام عملگر لیست استاندارد، برای بازنمایی P استفاده می‌شود؟** ***(این تمرین مشابه تمرین ۱۵.۹ در ویرایش سوم است)***

☑ حل: **الف)** در اینجا، هر خط جلو آمده یک مرحله عمیق‌تر در درخت اثبات را نشان می‌دهد و دو خطی که یکسان جلو آمده‌اند، دو روش برای اثبات هدفی هستند که جلو نیامده است. $P_1$ و $P_2$ که در یک خط بکار رفته‌اند، به ترتیب بند اول و بند دوم از P را نشان می‌دهند:

```
P(A, [1,2,3])                    goal
  P(1, [1|2,3])                  P1 => solution, with A = 1
  P(1, [1|2,3])                  P2
    P(2, [2,3])                  P1 => solution, with A = 2
    P(2, [2,3])                  P2
      P(3, [3])                  P1 => solution, with A = 3
      P(3, [3])                  P2

P(2, [1, A, 3])                  goal
  P(2, [1|2, 3])                 P1
  P(2, [1|2, 3])                 P2
    P(2, [2|3])                  P1 => solution, with A = 2
    P(2, [2|3])                  P2
      P(2, [3])                  P1
      P(2, [3])                  P2
```

**ب)** بهتر است P را Member بنامیم زیرا در زمانیکه اولین آرگومان یک عنصر از لیستی باشد که آرگومان دوم است، نتیجه داده و موفق می‌شود.

**؟ ۱۴.۹ در این تمرین به مرتب‌سازی در پرولوگ می‌پردازیم. الف) بندهایی به زبان پرولوگ بنویسید که مسند sorted(L) را تعریف کند به طوری که اگر و فقط اگر لیست L به صورت صعودی مرتب باشد، true را برگرداند. ب) مسند perm(L,M) را به زبان پرولوگ تعریف کنید به طوری که اگر L یک جایگشت از M باشد، true را برگرداند. ج) sort(L,M) را با استفاده از perm و sorted تعریف کنید (M یک نقطهٔ مرتب شده از L است). د) Sort را برروی لیست‌هایی بزرگ و بزرگتر آنقدر را اجرا کنید که خسته شوید. پیچیدگی زمانی برنامهٔ شما چقدر است؟ هـ) الگوریتم مرتب‌سازی سریع‌تری مانند مرتب‌سازی درجی با مرتب‌سازی سریع را به زبان پرولوگ بنویسید.** ***(این تمرین مشابه تمرین ۱۶.۹ در ویرایش سوم است)***

☑ حل: نسخه‌های متفاوت sort می‌توانند تفاوت بین مفاهیم منطقی و تابعی در پرولوگ را نشان دهند:

**الف)**

```
sorted([]).
sorted([X]).
sorted([X,Y|L]) :- X<Y, sorted([Y|L]).
```

**ب)**

```
perm([],[]).
perm([X|L],M) :-
  delete(X,M,M1),
  perm(L,M1).

delete(X,[X|L],L).          %% deleting an X from [X|L] yields L
delete(X,[Y|L],[Y|M]) :- delete(X,L,M).

member(X,[X|L]).
member(X,[_|L]) :- member(X,L).
```

**ج)**

```
sort(L,M) :- perm(L,M), sorted(M).
```

این شکل از مرتب‌سازی، همان شکل مشخص و قابل اجرایی است که می‌توان تعریف کرد. حداقل تعریف مفیدی که از مرتب‌سازی می‌توان ارائه داد، آن است که برای آن که M مرتب شده‌ی L باشد، بایستی M همان عناصر L را داشته و همگی به ترتیب چیده شوند. **د)** متاسفانه نمی‌توان یک برنامه مشخص داشت که این کار را انجام دهد و باید به شیوه تولید/ بررسی این کار را انجام دهیم: perm در هر زمان یک جایگشت مناسب را تولید می‌کند و sorted آنها را بررسی می‌کند. در بدترین حالت (زمانی که فقط یک جایگشت مرتب موجود باشد و آن آخرین مورد تولیدی باشد)، دارای پیچیدگی $O(n!)$ برای تولید خواهد بود. از آنجا که هر perm دارای پیچیدگی $O(n^2)$ و هر sorted دارای پیچیدگی $O(n)$ می‌باشد، کل sort در بدترین حالت پیچیدگی $O(n!\ n^2)$ خواهد داشت. **هـ)** در این جا یک مرتب‌سازی درجی با پیچیدگی $O(n^2)$ را مشاهده می‌کنید:

```
isort([],[]).
isort([X|L],M) :- isort(L,M1), insert(X,M1,M).
```

```
insert(X,[],[X]).
insert(X,[Y|L],[X,Y|L]) :- X=<Y.
insert(X,[Y|L],[Y|M]) :- Y<X, insert(X,L,M).
```

**؟ ۱۵.۹ در این تمرین به کاربرد بازگشتی قواعد بازنویسی‌شده با استفاده از برنامه‌نویسی منطقی می‌پردازیم. یک قاعدهٔ بازنویسی‌شده (یا مدولاسیون معکوس در اصطلاحات OTTER) معادله‌ای با جهت مشخص می‌باشد. به عنوان مثال، قاعدهٔ بازنویسی‌شده $x+0 \rightarrow x$ پیشنهاد می‌دهد که هر عبارت منطبق با $x+0$ را با عبارت $x$ جایگزین کنید. کاربرد قواعد بازنویسی شده، مرکزی‌ترین بخش سیستم استنتاج می‌باشد. برای نمایش قواعد بازنویسی‌شده از مسند *rewrite(x,y)* استفاده می‌کنیم. به عنوان مثال، همان قاعدهٔ فوق را می‌توان به صورت *rewrite*$(x+0,x)$ بیان کرد. برخی اصطلاحات اصلی بوده و نمی‌توانند ساده‌تر شوند بنابراین *primitive*$(0)$ بیان می‌کند که 0 یک اصطلاح اصلی است. الف) یک تعریف از مسند *simplify(x,y)* بیان کنید که درصورتی که y نسخه ساده شده‌ای از $x$ باشد، true را برگرداند. (دیگر نتوان هیچ قاعدهٔ بازنویسی شده‌ای را به هر زیر بخش از $y$ اعمال نمود). ب) مجموع قواعدی برای ساده‌سازی عبارات حاوی عملگرهای محاسباتی بنویسید و سپس الگوریتم ساده‌سازی خود را به برخی عبارات به عنوان نمونه اعمال کنید. ج) مجموعه‌ای از قواعد بازنویسی‌شده برای مشتق‌گیری نمادین نوشته و سپس با استفاده از آنها و قواعد ساده‌سازی خود از عبارات محاسباتی و نمایی مشتق گرفته و آنها را ساده کنید. *(این تمرین مشابه تمرین ۱۷.۹ در ویرایش سوم است)***

☑ حل: این تمرین قدرت تطبیق الگو که در پرولوگ گنجانده شده است را نشان می‌دهد.

**الف)** که ساده‌سازی به نظر ساده می‌رسد ولی دانشجویان ممکن است در انتخاب بین ساده‌سازی و چرخیدن بی‌نهایت گیر کنند:

```
simplify(X,X) :- primitive(X).
simplify(X,Y) :- evaluable(X), Y is X.
simplify(Op(X)) :- simplify(X,X1), simplify_exp(Op(X1)).
simplify(Op(X,Y)) :- simplify(X,X1), simplify(Y,Y1), simplify_exp(Op(X1,Y1)).

simplify_exp(X,Y) :- rewrite(X,X1), simplify(X1,Y).
simplify_exp(X,X).

primitive(X) :- atom(X).
```

**ب)** چند قاعدهٔ بازنویسی‌شده از لیستی که Norving در سال 1992 ارائه داد را در زیر می‌بینید:

```
Rewrite(X+0,X).
Rewrite(0+X,X).
Rewrite(X+X,2*X).
Rewrite(X*X,X^2).
Rewrite(X^0,1).
Rewrite(0^X,0).
Rewrite(X*N,N*X) :- number(N).
Rewrite(ln(e^X),X).
Rewrite(X^Y*X^Z,X^(Y+Z)).
Rewrite(sin(X)^2+cos(X)^2,1).
```

**ج)** در زیرقواعدی جهت مشتق‌گیری مشاهده می‌کنید که d(X,Y) نشان‌دهندهٔ مشتق عبارت Y نسبت به متغیر X می‌باشد:

```
Rewrite(d(X,X),1).
Rewrite(d(U,X),0) :- atom(U), U /= X.
Rewrite(d(U+V,X),d(U,X)+d(V,X)).
Rewrite(d(U-V,X),d(U,X)-d(V,X)).
Rewrite(d(U*V,X),V*d(U,X)+U*d(V,X)).
Rewrite(d(U/V,X),(V*d(U,X)-U*d(V,X))/(V^2)).
Rewrite(d(U^N,X),N*U^(N-1)*d(U,X)) :- number(N).
Rewrite(d(log(U),X),d(U,X)/U).
Rewrite(d(sin(U),X),cos(U)*d(U,X)).
Rewrite(d(cos(U),X),-sin(U)*d(U,X)).
Rewrite(d(e^U,X),d(U,X)*e^U).
```

**؟ ۱۶.۹ در این تمرین پیاده‌سازی الگوریتم‌های جستجو در زبان پرولوگ را بررسی می‌کنیم. فرض کنید *successor*(X,Y) زمانی درست است که حالت Y یک پسین حالت X باشد و *goal*(X) زمانی درست است که x یک حالت هدف باشد. تعریفی برای *solve*(X,P) ارائه دهیم که درآن P یک مسیر (لیستی از حالات) با شروع از X بوده و به یک حالت هدف ختم شود و دنباله مراحل صحیح سپری شده توسط *succersor* مشخص می‌شود. روش جستجوی اول‌عمق، ساده‌ترین راه برای انجام این**

**کار خواهد بود. چگونه می‌توان به سادگی کنترل اکتشافی را به جستجو اضافه کنیم؟** ***(این تمرین مشابه تمرین ۱۸.۹ در ویرایش سوم است)***

☑ حل: اگر از کارکرد پرولوگ آگاه باشید پاسخ بسیار ساده خواهد بود:

```
solve(X,[X]) :- goal(X).
solve(X,[X|P]) :- successor(X,Y), solve(Y,P).
```

می‌توان این روش را به زبان ساده این گونه بیان نمود: «از حالت اولیه شروع به حرکت می‌کنیم. اگر همان یک حالت هدف بود آنگاه مسیر شامل فقط یک حالت اولیه، راه‌حل مساله می‌باشد. درغیر اینصورت، پسینی می‌یابیم که یک مسیر از آن پسین به هدف وجود داشته باشد. آنگاه یک راه‌حل عبارتست از شروع از حالت اولیه و ادامهٔ راه توسط آن مسیر». دقت داشته باشید که solve نمی‌تواند به تنهایی مسیر P که راه‌حل است را بیابد و فقط می‌تواند بگوید که یک مسیر داده شده راه‌حل هست یا خیر. اگر می‌خواهید تابع اکتشافی (یا حتی جستجوی اول سطح) را اضافه کنید، به یک صف مشخص نیاز دارید. الگوریتم این کار مشابه نسخه‌ای از کد به زبان لیسپ، پایتون و یا جاوا که در کتاب ذکر شده است می‌باشد.

**؟ ۱۷.۹ چگونه می‌توان از رزولوشن جهت تعیین اعتبار یک جمله استفاده کرد. جهت تعیین ارضاپذیرنبودن چطور؟** ***(این تمرین مشابه تمرین ۲۱.۹ در ویرایش سوم است)***

☑ حل: این تمرین به فهم دانشجویان از رزولوشن و توانایی آنها در تفکر به سطوح بالاتر روابط مجموعه‌ای می‌پردازد. یادآوری می‌کنیم که رزولوشن برای اثبات $KB \models \alpha$ اثبات می‌کند که $KB \wedge \neg\alpha$ نادرست است (مشابه برهان خلف). معمولا در سیستم رزولوشن سوال $ASK(KB,\alpha)$ استفاده می‌شود. اکنون می‌خواهیم نشان دهیم که جمله‌ای نظیر $\beta$ معتبر یا غیرارضاپذیر هست یا خیر. جملهٔ $\beta$ معتبر است اگر بتوانیم بدون هیچ اطلاعات اضافه‌ای آن را نمایش دهیم. این کار را با پرسیدن سوال $ASK(KB_0,\beta)$ انجام می‌دهیم که درآن $KB_0$ پایگاه دانش تهی می‌باشد. جملهٔ $\beta$ که غیرارضاپذیر است با خودش ناسازگار است. بنابراین اگر پایگاه دانش را مجددا تهی کرده و بپرسیم $ASK(KB_0,\neg\beta)$، آنگاه سیستم رزولوشن نقیض آن یعنی $\neg\neg\beta$ را در نظر می‌گیرد. اگر بتواند این کار را انجام دهد آنگاه $\neg\neg\beta$ ممکن شده و $\beta$ ناسازگار است.

**؟ ۱۸.۹ طبق جملهٔ «اسب‌ها حیوان هستند» می‌توان نتیجه گرفت که «سریک اسب، سریک حیوان است». نشان دهید که این استنتاج توسط مراحل زیر معتبر است. الف) مقدم و تالی را به زبان منطق مرتبهٔ اول ترجمه کنید. از سه مسند زیر استفاده کنید:** ***$HeadOf(h,x)$*** **(*** ***$h$*** **سر** ***$x$*** **است)،** ***$Horse(x)$*** **و** ***$Animal(x)$***

**ب) بخش تالی را نقیض کرده و سپس مقدم و تالی نقیض شده را به شکل نرمال عطفی (CNF) تبدیل کنید. ج) با استفاده از رزولوشن نشان دهید که تالی از مقدم نتیجه می‌شود.** ***(این تمرین مشابه تمرین ۲۳.۹ در ویرایش سوم است)***

☑ حل: این شکلی از استنتاج است که برای اثبات آن که «قیاس ارسطو نمی‌تواند درهمهٔ استنتاجات دقیق باشد» استفاده شد:

**الف)**

$$\forall x \;\; Horse(x) \Rightarrow Animal(x)$$
$$\forall x, h \;\; Horse(x) \wedge HeadOf(h,x) \Rightarrow \exists y \;\; Animal(y) \wedge HeadOf(h,y)$$

**ب)**

$$A.\ \neg Horse(x) \vee Animal(x)$$
$$B.\ Horse(G)$$
$$C.\ HeadOf(H,G)$$
$$D.\ \neg Animal(y) \vee \neg HeadOf(H,y)$$

(در اینجا $A$ از اولین جمله قسمت الف و سایرین از دومین جمله می‌آیند. $G$ و $H$ ثابت‌های اسکولم هستند.)

**ج)** $C$ و $D$ را حل کنید تا به $\neg Animal(G)$ دست یابید. سپس آن را با $A$ حل کنید تا به $\neg Horse(G)$ برسید. سپس آن را با B حل کنید تا به نتیجهٔ دلخواه برسید.

**؟ ۱۹.۹ در اینجا دو جمله به زبان منطق مرتبهٔ اول را مشاهده می‌کنید. (A)** $\forall x\, \exists y\, (x \geq y)$ **(B)** $\exists y\, \forall x\, (x \geq y)$ **الف) فرض کنید محدودهٔ متغیرها اعداد طبیعی** $0,1,2,\ldots,\infty$ **و بوده ومسند «≥» به معنای بزرگتر و مساوی بودن باشد. طبق این جملات (A)، (B) را به جملات صریح تبدیل کنید. ب) آیا طبق این تفسیر (A) صحیح است؟ ج) آیا طبق این تفسیر (B) صحیح است؟ د) آیا (A) به طور منطقی (B) را ایجاب می‌کند؟ هـ) آیا (B) به طور منطقی (A) را ایجاب می‌کند؟ و) با استفاده از رزولوشن اثبات کنید که (A) از (B) نتیجه می‌شود، حتی اگر فکر می‌کنید که (B) به طور منطقی (A) را ایجاب نمی‌کند، باز هم اثبات را ادامه دهید تا با شکست مواجه شوید. برای هر مرحله از رزولوشن، جایگزینی یکسان‌سازی را نشان دهید. اگر اثبات با شکست مواجه شده است به طور دقیق توضیح دهید که کجا، چرا و چگونه به شکست مواجه شده است. ز) اکنون اثبات کنید که (B) از (A) نتیجه می‌شود.** ***(این تمرین مشابه تمرین ۲۴.۹ در ویرایش سوم است)***

☑ حل: این تمرین فهم دانشجویان از مدل و ایجاب را نشان می‌دهد. **الف)** ترجمهٔ (A) به این صورت است: «برای هر عدد طبیعی، عدد طبیعی دیگری وجود دارد که کوچکتر و مساوی با آن باشد.» ترجمهٔ (B) به این صورت است: «یک عدد طبیعی خاص وجود دارد که کوچکتر و مساوی با هر عدد طبیعی دیگر است.» **ب)** بله. عبارت (A) تحت این تفسیر صحیح است. شما می‌توانید همواره خود عدد را به عنوان آن عدد دیگر در جمله در نظر بگیرید. **ج)** بله. عبارت (B) تحت این تفسیر صحیح است عدد صفر را به عنوان آن عدد طبیعی خاص در نظر بگیرید. **د)** خیر. عبارت (A) نمی‌تواند به طور منطقی (B) را ایجاب کند.

**هـ)** بله. عبارت (B) به طور منطقی (A) را ایجاب می‌کند. **و)** می‌خواهیم با استفاده از رزولوشن اثبات کنیم که عبارت (A)، (B) را ایجاب می‌کند. برای این منظور، پایگاه دانشی شامل عبارت (A) و نقیض (B) را در نظر می‌گیریم که آن را با (B-) نمایش می‌دهیم. در ابتدا باید (B-) و (A) را به صورت متعارف تبدیل کنیم. برای (B-) باید نماد ¬ را در پشت دو سور قرار دهیم و برای هر دو جمله نیاز به تعریف یک تابع اسکولم داریم:

(A) $x \geq F_1(x)$

(-B) $\neg F_2(y) \geq y$

اکنون سعی می‌کنیم هر دوی آنها را با هم حل کنیم ولی باید از یکسان‌سازی استفاده شود. به نظر می‌رسد که جایگزینی باید به صورت $\{x/F_2(y), y/F_1(F_2(y)\}$ باشد ولی این معادل است با $\{x/F_2(y), y/F_1(x)\}$ که با شکست مواجه می‌شود زیرا $y$ تنها در عبارات شامل $y$ کاربرد دارد. بنابراین رزولوشن شکست می‌خورد و هیچ مرحله دیگری از رزولوشن برای امتحان کردن باقی‌نمانده است، پس (B) از عبارت (A) نتیجه نمی‌شود. **ز)** برای اثبات آن که عبارت (B)، (A) را ایجاب می‌کند با پایگاه دانش شامل (B) و نقیض (A) شروع می‌کنیم که آن را با (A-) نمایش می‌دهیم.

(A) $\neg F_1 \geq y$

(-B) $x \geq F_2(x)$

با اعمال رزولوشن و جایگزینی $\{x/F_1, y/F_2(F_1)\}$ به $False$ می‌رسیم و اثبات می‌شود که (B)، عبارت (A) را ایجاب می‌کند.

**؟ ۲۰.۹ رزولوشن می‌تواند برای سوالاتی حاوی متغیر، اثبات‌های بی‌فایده تولید کند. بنابراین باید مکانیزمی معرفی شود تا پاسخ‌های متناهی استخراج کند. توضیح دهید که چرا این نتیجه در پایگاه دانش‌هایی شامل فقط بندهای متناهی، رخ نمی‌دهد.** ***(این تمرین مشابه تمرین ۲۵.۹ در ویرایش سوم است)***

☑ حل: یکی از این روش‌ها آن است که به رزولوشن اجازه دهیم از روی موارد استنتاج کند که دراینصورت می‌توان از روی درست بودن $B$ و $A$ مورد $C$ را اثبات نمود بدون آن که هر یک از آنها را بشناسیم. در پایگاه دانش‌هایی که فقط بند متناهی دارند، تنها با یک زنجیره واحد از استنتاجات روبرو هستیم که می‌توان زنجیره را دنبال نموده و به متغیرها نمونه داد.

**؟ ۲۱.۹ در این فصل گفته شد که رزولوشن نمی‌تواند برای تولید تمام نتایج منطقی حاصل از یک مجموعه جملات به کار رود. آیا الگوریتم دیگری وجود دارد که این کار را انجام دهد.** ***(این تمرین مشابه تمرین ۲۶.۹ در ویرایش سوم است)***

☑ حل: خیر. یکی از بخش‌های تعریف الگوریتم، پایان‌پذیربودن آن است. از آنجا که برای یک مجموعه جملات، می‌توانند بی‌نهایت نتیجه منطقی وجود داشته باشد، هیچ الگوریتمی قادر به تولید تمامی آنها نیست. یکی دیگر از دلایل موجود برای نبودن چنین الگوریتمی آن است که ایجاب برای FOL تصمیم‌ناپذیر است. اگر الگوریتمی وجود داشته باشد که بتواند تمام نتایج منطقی حاصل از مجموعه جملات $S$ را بیابد، آنگاه اگر بخواهیم بدانیم که آیا $B$ توسط $S$ ایجاب می‌شود یا خیر، یک راه آن است که بررسی کنیم B در مجموعه تولیدی هست یا خیر. ولی می‌دانیم که این کار غیرممکن است و تولید این مجموعه غیرممکن است.

# فصل ۱۰ (ویرایش دوم)

**؟ ۱.۱۰ عباراتی بنویسید که اثر واکنش *Shoot* در دنیای ومپوز را تعریف کند. اثر این واکنش را بر روی ومپوز توصیف کرده و بخاطر داشته باشید که برای شوت کردن بایستی از تیر کارگزار (پیکان) استفاده شود.**

☑ حل: شوت کردن کارگزار موجب مرگ آن شده و هیچ روشی برای زنده کردن مجدد آن وجود ندارد. بنابراین قاعده حالت پسین برای $Alive$ فقط دارای بند دوم می‌باشد:

$$\forall a, s \;\; Alive(Wumpus, Result(a, s)) \;\Leftrightarrow\; [Alive(x, y, s) \wedge \neg(a = Shoot \wedge Has(Agent, Arrow, s) \wedge Facing(Agent, Wumpus, s))]$$

که $Facing(a,b,s)$ نشان‌دهندهٔ موقعیت b,a و در جهت s می‌باشد. پس از شوت کردن تیر، مالکیت آن از دست می‌رود و هیچ راهی برای درست کردن مجدد آن نداریم:

$$\forall a, s \;\; Has(Agent, Arrow, Result(a, s)) \Leftrightarrow [Has(Agent, Arrow, s) \wedge (a \neq Shoot)]$$

**؟ ۲.۱۰ با استفاده از محاسبات موقعیت[18]، قواعدی بنویسید که به موقعیت $S_0$ زمان صفر را تخصیص داده و به سایر موقعیت‌هایی که با طی شدن $t$ واکنش از $S_0$ بدست می‌آیند، زمان $t$ را تخصیص دهد.**

☑ حل:

$$Time(S_0, 0)$$
$$Time(Result(seq, s), t) \;\Rightarrow\; Time(Result([a|seq], s), t + 1)$$

توجه داشته باشید که عملیات بازگشت به هیچ مورد پایه‌ای نیاز ندارد زیرا قاعدهٔ زیر را داریم: $Result([], s) = s$

**؟ ۳.۱۰ مسألهٔ مسیریابی روبات برای رفتن از یک شهر به شهر دیگر را در نظر بگیرید. واکنش پایه این روبات، *Go(x,y)* است که روبات را از شهر $x$ به شهر $y$ می‌برد. اگر بین آن دو شهر یک مسیر مستقیم موجود باشد، آنگاه عبارت *DirectRoute(x,y)* درست است اگر و فقط اگر یک مسیر مستقیم از $x$ به $y$ موجود باشد؛ می‌توانید فرض کنید که تمامی این حقایق در پایگاه‌دانش ثبت شده است (نقشه را در این فصل مشاهده کنید). روبات از شهر آراد شروع به حرکت کرده و باید به شهر بخارست برسد. الف) یک توصیفگر منطقی مناسب برای موقعیت شروع روبات بنویسید. ب) یک پرس‌وجوی منطقی مناسب بنویسید که پاسخ آن مسیرهای ممکن به هدف باشد. ج) عبارتی بنویسید که واکنش *Go* را توصیف کند. د) اکنون فرض کنید برای پیمودن مسیری مستقیم بین دو شهر، میزان سوخت مصرفی برابر با فاصلهٔ آن دو شهر باشد. در آغاز، روبات به اندازهٔ ظرفیت کامل باک خود سوخت دارد. بازنمایی خود را به طوری گسترش دهید که این موارد نیز در آن جای گیرد. البته توصیف شما از اقدامات روبات باید به گونه‌ای باشد که پرس‌وجوی قسمت قبل شما در برنامه‌ریزی محتمل همچنان دارای پاسخ باشد. هـ) موقعیت شروع را توصیف کرده و قاعده یا قواعد جدیدی بنویسید که واکنش *Go* را توصیف کنند. و) اکنون فرض کنید برخی از رئوس در نقشه، جایگاه بنزین بوده و روبات می‌تواند باک خود را در آنها پرنماید. بازنمایی خود را گسترش داده و تمام قواعد موردنیاز برای توصیف جایگاه‌های بنزین شامل واکنش *Fillup* را بنویسید.**

☑ حل: این تمرین از بازنمایی حالت اولیه شروع کرده و با افزودن اقدامات بیشتر، بازنمایی خود را گسترش می‌دهد. پیشنهاد می‌شود این استدلال را در یک اثبات‌گر قضیه پیاده‌سازی کنید، البته شاید استفاده از استدلال منطقی برای نقطهٔ شروع (یافتن یک مسیر در گراف) کاری بس دشوار باشد، ولی دقت داشته باشید که می‌توان یک موقعیت پیچیده را ساده نمود یعنی فقط بخش‌های پیچیده را توصیف کرده و هیچ چیزی به سیستم استدلال نیافزاییم:

**الف)** $At(Robot, Arad, S_0)$ **ب)** $\exists s \;\; At(Robot, Bucharest, s)$

**ج)** هم‌اکنون باید قاعدهٔ حالت پسین را به صورت مکانیکی در نظر بگیریم. $\forall a,x,y,s$:

$$At(Robot, y, Result(a, s)) \;\Leftrightarrow\; [(a = Go(x, y) \wedge DirectRoute(x, y) \wedge At(Robot, x, s)) \vee (At(Robot, y, s) \wedge \neg(\exists z \;\; a = Go(y, z) \wedge z \neq y))]$$

---

[18] Situation Calculus

**د)** برای نمایش مقدار سوخت روبات در یک جایگاه مشخص، از تابع $Fuel(Robot,s)$ استفاده می‌کنیم. فرض کنید $Distance(x,y)$ نشان‌دهندهٔ فاصلهٔ بین شهرهای $y$ و $x$ باشد که برحسب واحدهای مصرفی سوخت اندازه‌گیری می‌شود. همچنین عبارت $Full$ ثابتی است که ظرفیت کامل باک روبات را نشان دهد. **هـ)** موقعیت شروع را می‌توان اینگونه توصیف نمود:

$$At(Robot,Arad,S_0)\wedge Fuel(Robot,s)=Full$$

قاعدهٔ فوق را می‌توان اینگونه گسترش داد (لازم به ذکر است که برای حالتی که روبات از جایگاه بنزین عبور کند، هیچ چیزی در نظر نمی‌گیریم). داریم $\forall a,x,y,s$:

$$\begin{aligned}
At(Robot,y,Result(a,s)) \quad \Leftrightarrow \quad & [(a=Go(x,y)\\
& \wedge DirectRoute(x,y)\wedge At(Robot,x,s)\\
& \wedge Distance(x,y)<Fuel(Robot,s))\\
\vee \quad & (At(Robot,y,s)\\
& \wedge\neg(\exists z \;\; a=Go(y,z)\wedge z\neq y))]\\
Fuel(Robot,Result(a,s))=f \quad \Leftrightarrow \quad & [(a=Go(x,y)\\
& \wedge DirectRoute(x,y)\wedge At(Robot,x,s)\\
& \wedge Distance(x,y)<Fuel(Robot,s)\\
& \wedge f=Fuel(Robot,s)-Distance(x,y))\\
\vee \quad & (f=Fuel(Robot,s)\\
& \wedge\neg(\exists v,w \;\; a=Go(v,w)\wedge v\neq w))]
\end{aligned}$$

**و)** ساده‌ترین راه برای گسترش بازنمایی، افزودن مسند $GasStation(x)$ است که برای شهرهای دارای جایگاه بنزین درست می‌باشد. برای بازنمایی واکنش $FillUp$ باید بند دیگری را به قانون $Fuel$ فوق بیافزاییم. این بند می‌گوید در زمانی که $a=FillUp$ آنگاه $F=Full$.

**؟ ۱۰.۴ در مورد روش‌هایی تحقیق کنید که می‌توانند با گسترش محاسبات رخدادی[19]، رخدادهای همزمان را پوشش دهند. آیا می‌توان از ترکیب فراوان اصول اجتناب کرد؟** ***(این تمرین مشابه تمرین ۱۲.۱۳ در ویرایش سوم است)***

☑ حل: قسمت دشوار رخدادها و واکنش‌های همزمان، مربوط به چگونگی درست استفاده‌کردن از استنتاج‌های ممکن است. برای شروع بد نیست نظری به مقاله Shanahan که در سال 1999 نوشته است، نظری بیافکنید. بخش پنجم از این مقاله چگونگی مدیریت واکنش‌های همزمان را نشان می‌دهد که برای اینکار مسندهای $Cancels$ و $Canceled$ را اضافه کرده و چگونگی استنتاج واکنش‌ها با یکدیگر را به تفضیل بیان می‌کند.

**؟ ۱۰.۵ هفت عبارت زیر را با استفاده و گسترش از بازنمایی گفته شده در این فصل، بازنمایی کنید: الف) آب مایعی با درجهٔ بین 0 تا 100 است. ب) آب در 100 درجه می‌جوشد. ج) آب موجود در بطری آب John، یخ زده است. د) Perrier نوعی آب است. هـ) John در بطری آب خود Perrier دارد. و) تمام مایعات یک نقطهٔ انجماد دارند. ز) وزن یک لیتر آب، بیشتر از وزن یک لیتر الکل است. اکنون این تمرین را برای بازنمایی مبتنی بر مرئولوژیکی[20] تکرار کنید. به عنوان مثال،آب را شیای در نظر بگیرید که شامل تمام اجزای آب‌هایی است که در جهان وجود دارند.** ***(این تمرین مشابه تمرین ۱۲.۷ در ویرایش سوم است)***

☑ حل: در ابتدا بایستی مواد گفته شده در این تمرین را تعریف کنیم. بنابراین «آب» یک طبقه بوده و عناصر وجودی آن باعث می‌شوند تا به آن آب بگوئیم. یکی از مهمترین مشکلات ما با زبان نوشتاری رایج، کژتابی کلمات است. به عنوان مثال به نظر می‌رسد بخشی از کلمهٔ «آب» شامل مفهوم «یخ» هم شود (یخ همان آب منجمد است) ولی از جنبهٔ دیگر کلمهٔ «آب» و «یخ» مجزا هستند (این آب است و آن یخ). به نظر می‌رسد منظور جملات این تمرین همان مورد اول باشد. بنابراین ما پاسخ خود را بر این فرض بنا می‌داریم. همچنین به نظر می‌رسد که در این جملات منظور از «آب» همان $H_2O$ باشد. از دیگر مشکلات ما تغییر ماهیت اشیا در طی زمان است (انجماد و تبخیر). بنابراین اگر w مقداری آب باشد، نمی‌توانیم بگوییم $w\in Liquid$. زیرا $w$ ممکن است در یک زمان مایع و در زمان دیگر جامد باشد. به منظور ساده‌سازی، از بازنمایی محاسبات‌موقعیت مانند $T(w\in Liquid)$ استفاده می‌کنیم. در این صورت ممکن است پاسخ‌های صحیح زیادی برای هر یک از موارد پیدا کنید ولی نکتهٔ مهم آن است که اطلاعات باید به صورت «سازگار» بازنمایی شوند. به عنوان مثال اگر $Water$ به عنوان یک دسته از مواد استفاده شده باشد نباید $Liquid$ را به عنوان یک مسند در شی بکار ببرید. **الف)** می‌توان جملهٔ اول را به این صورت ترجمه نمود: « هر نوع آب در هر موقعیتی مایع است، اگر و فقط اگر دمای آب در موقعیتی بین 0 تا 100 درجهٔ سانتیگراد باشد». پس داریم:

---

[19] Event Calculus

[20] Mereological approach

$$\forall w,s\ \ w\in Water \Rightarrow (Centigrade(0) < Temperature(w,s) < Centigrade(100)) \Leftrightarrow T(w\in Liquid,s)$$

**ب)** می‌توان از ابزارهای ساخت استفاده نمود. در این فصل از *MeltingPoint* به عنوان یک مسند استفاده کردیم که به نمونه‌هایی مشخص از مواد اعمال شد. در اینجا *SBoilingPoint* را به عنوان نقطهٔ جوش تمام نمونه‌های یک ماده تعریف می‌کنیم. مفهوم کلی جوشیدن آن است که نمونه‌های ماده به اندازهٔ نقطهٔ جوش حرارت داشته باشند:

$$SBoilingPoint(c,bp) \Leftrightarrow \forall x,s\ \ x\in c \Rightarrow (\forall t\ \ T(Temperature(x,t),s) \wedge t > bp \Rightarrow T(x\in Gas,s))$$

بنابراین تنها نیاز است بگوییم:

$$SBoilingPoint(Water,Centigrade(100))$$

**ج)** از ثابت *Now* به منظور بازنمایی موقعیت فعلی یعنی موقعیتی که جمله در آن بیان می‌شود، استفاده می‌کنیم. البته دقت داشته باشید که ممکن است به سادگی در بیان عبارت «فقط برخی از آب‌های داخل بطری منجمد هستند» دچار اشتباه شوید:

$$\exists b\ \forall w\ \ w\in Water \wedge b\in WaterBottles \wedge Has(John,b,Now) \wedge Inside(w,b,Now) \Rightarrow (w\in Solid,Now)$$

**د)** $Perrier \subset Water$ Perrier نوعی آب است.

**هـ)** John در بطری آب خود Perrier دارد.

$$\exists b\ \forall w\ \ w\in Water \wedge b\in WaterBottles \wedge Has(John,b,Now) \wedge Inside(w,b,Now) \Rightarrow w\in Perrier$$

**و)** احتمالا منظور این عبارت آن است که «تمام موادی که در دمای اتاق مایع هستند، دارای نقطهٔ انجماد می‌باشند». اگر از *RTLiquidSubstance* برای بیان این دسته از مواد استفاده کنیم آنگاه داریم:

$$\forall c\ \ RTLiquidSubstance(c) \Rightarrow \exists t\ \ SFreezingPoint(c,t)$$

که در این عبارت *SFreezingPoint* تعریفی مشابه *SBoilingPoint* دارد. بد نیست بدانید که این عبارت در جهان حقیقی نادرست است: می‌توان دسته‌هایی مانند «blue liquid» اختراع کرد که یک نقطهٔ انجماد یکتا ندارند. می‌توان برای تمرین، مادهٔ خالصی تعریف نمود که تمام نمونه‌های آن دارای ترکیب شیمیایی یکسان باشند.

**ز)** وزن یک لیتر آب، بیشتر از وزن یک لیتر الکل است.

$$\forall w,a\ \ w\in Water \wedge a\in Alcohol \wedge Volume(w)=Liters(1) \wedge Volume(a)=Liters(1) \Rightarrow Mass(w) > Mass(a)$$

**؟ ۶.۱۰ برای این موارد تعریف‌هایی ارائه دهید: الف) *ExhaustivePartDecomposition* ب) *PartPartition* ج) *PartwiseDisjoint*. تعریف این موارد، مشابه تعریف *ExhaustiveDecomposition* Partition و Disjoint می‌باشد. آیا مواردی وجود دارد که در آن *PartPartition(s,BunchOf(s))* باشد؟ اگر وجود دارد آن را اثبات کنید وگرنه مثالی نقض ارائه کرده و شرایط لازم جهت برقراری آن را تعریف کنید. (این تمرین مشابه تمرین ۸.۱۲ در ویرایش سوم است)**

☑ حل: **الف)** *ExhaustivePartDecomposition* بین یک مجموعه اجزا و یک عنصر کامل قرار می‌گیرد و می‌گوید «هر چیزی که یکی از اجزای کامل باشد باید یکی از اجزای موجود در مجموعهٔ اجزا باشد»:

$$\forall s,w\ \ ExhaustivePartDecomposition(s,w) \Leftrightarrow (\forall p\ \ PartOf(p,w) \Rightarrow \exists p_2\ \ p_2\in s \wedge PartOf(p,p_2))$$

**ب)** *PartPartition* زمانی بین یک مجموعه اجزا و عنصر کامل قرار می‌گیرد که آن مجموعه مجزا و تجزیه‌ای جامع باشد:

$$\forall s,w\ \ PartPartition(s,w) \Leftrightarrow PartwiseDisjoint(s) \wedge ExhaustivePartDecomposition(s,w)$$

**ج)** یک مجموعه اجزا در صورتی *PartwiseDisjoint* است که اگر هر دو عنصر آن مجموعه را بردارید، دیگر هیچ چیزی نباشد که جزئی از آن دو عنصر باشد:

$$\forall s\ \ PartwiseDisjoint(s) \Leftrightarrow \forall p_1,p_2\ \ p_1\in s \wedge p_2\in s \wedge p_1\neq p_2 \Rightarrow \neg\exists p_3\ PartOf(p_3,p_1) \wedge PartOf(p_3,p_2)$$

رای هر $s$، عبارت $PartPartition(s,BunchOf(s))$ برقرار نیست. یک مجموعهٔ $s$ ممکن است شامل اشیایی باشد که از نظر یزیکی با هم همپوشانی داشته باشند، مانند «یک دست» و «انگشتان یک دست». آنگاه در این مثال، $BunchOf(s)$ معادل ست با دست ولی $s$ یک قسمتی از آن نیست. بنابراین نیاز است تا مطمئن شویم که عناصر $s$ مجزا بوده و همپوشانی نداشته اشند:

$$\forall s \;\; PartwiseDisjoint(s) \Rightarrow PartPartition(s, BunchOf(s))$$

**؟ ۷.۱۰ مجموعه جملاتی بنویسید که بتوان با داشتن قیمت هر پوند گوجه‌فرنگی، قیمت یک گوجه‌فرنگی مشخص را محاسبه کرد ( یا هر شی دیگر را). نظریهٔ خود را گسترش دهید تا بتوانیم قیمت یک کیسه گوجه را از روی آن محاسبه کنیم.**

☑ حل: برای هر نمونه $i$ از مادهٔ $s$ که هر پوند این ماده، قیمت $c$ و وزن $n$ داشته باشد، قیمت $i$ عبارت است از $n \times c$ یا به عبارت دیگر:

$$\forall i, s, n, c \;\; i \in s \wedge PricePer(s, Pounds(1)) = \$(c) \wedge Weight(i) = Pounds(n) \Rightarrow Price(i) = \$(n \times c)$$

گر $b$، مجموعه گوجه‌فرنگی‌های موجود در یک کیسه باشد، آنگاه $BunchOf(s)$ یک شی ترکیبی، حاوی تمام گوجه‌های وجود در کیسه می‌باشد. آنگاه داریم:

$$\forall i, s, n, c \;\; b \subset s \wedge PricePer(s, Pounds(1)) = \$(c) \wedge Weight(BunchOf(b)) = Pounds(n) \Rightarrow Price(BunchOf(b)) = \$(n \times c)$$

**؟ ۸.۱۰ یک راه کار ثانویه برای بازنمایی اندازه‌گیری آن است که تابع واحد(unit) را به شی انتزاعی طول اعمال کنیم. به عنوان ثال در این بازنمایی $Inches(Length(L_1))=1.5$ است. این راه‌کار در مقایسه با آنچه که در این فصل گفته شد، چگونه خواهد ود؟ نتایج بحث شامل تبدیل اصول، نامگذاری کمیت‌های انتزاعی(مانند 50 دلار) و مقایسه معیارهای انتزاعی در واحدهای ختلف (مثلا 50 اینچ بیشتر از 50 سانتی‌متر است) می‌باشد. *(این تمرین مشابه تمرین ۱۲.۹ در ویرایش سوم است)***

☑ حل: در راهکار ارائه شده در این فصل، یک اصل تبدیل مشابه این است:

$$\forall x \;\; Centimeters(2.54 \times x) = Inches(x)$$

«50 دلار» به صورت $\$(50)$ می‌باشد که نام یک کمیت انتزاعی و پولی است. برای هر تابع اندازه‌گیری نظیر $\$$، می‌توان از $>$ به صورت زیر استفاده کرد:

$$\forall x, y \;\; x > y \Rightarrow \$(x) > \$(y)$$

صل تبدیل برای دلار و سنت عبارت است از:

$$\forall x \;\; Cents(100 \times x) = \$(x)$$

ه سرعت می‌توان نتیجه گرفت که:

$$\$(50) > Cents(50)$$

در راه‌کار جدید که این تمرین پیشنهاد می‌دهد، باید اشیایی را معرفی کنیم که طول آنها تبدیل شده باشد:

$$\forall x \;\; Centimeters(Length(x)) = 2.54 \times Inches(Length(x))$$

هیچ روش مشخصی برای بیان مستقیم «50 دلار» و یا ارتباط آن با «50 سنت» وجود ندارد. مجددا باید اشیایی را تعریف کنیم که ارزش پولی آنها 50 دلار یا 50 سنت باشد:

$$\forall x, y \;\; \$(Value(x)) = 50 \wedge Cents(Value(y)) = 50 \Rightarrow \$(Value(x)) > \$(Value(y))$$

**؟ ۹.۱۰ یک بازنمایی برای تبدیل نرخ‌های پولی بنویسید که در آن نوسانات روزانه مجاز باشد. *(این تمرین مشابه تمرین ۱۴.۱۲ در ویرایش سوم است)***

☑ حل: در اینجا تابع $ExchangeRate$ را تعریف می‌کنیم که سه ورودی می‌پذیرد: نوع پول منبع، یک مدت زمانی و نوع پول مقصد. خروجی این تابع یک عدد است که نشان‌دهندهٔ نرخ تبدیل برای رسیدن از پول منبع به پول مقصد می‌باشد. به عنوان ثال:

$$ExchangeRate(USDollar, 17Feb1995, DanishKrone) = 5.8677$$

دان معنا که شما باید در هفدهم فوریه، 5.8677 کرون برای یک دلار بپردازید، این مقدار طبق نرخ تبدیلی که بانک فدرال در ساعت 10 صبح منتشر کرده است، محاسبه شده است. این عدد یک نقطهٔ میانه بین نرخ خرید و فروش در نظر گرفته شده ست. در تحلیل کاملتر بایستی علاوه بر نرخ خرید و نرخ فروش، دستمزد و حق ماموریت نیز برای هر تبدیل پولی در نظر گرفته شود. دقت داشته باشید که در عبارت فوق، بین یک پول مانند $USDoller$ و یک واحد اندازه‌گیری تفاوت وجود دارد. به عنوان ثال این مورد را در عبارت $USDollars(1.99)$ مشاهده می‌کنید.

**؟ ۱۰.۱۰ در این تمرین به بررسی رابطهٔ بین طبقه رخدادها و بازهٔ زمانی که آنها رخ می‌دهند، می‌پردازیم. الف) مسند $T(c,i)$ را برحسب اصطلاحات $During$ و $\in$ تعریف کنید. ب) توضیح دهید که چرا برای عطف طبقه رویدادها، نیاز به دو نهادگذاری مختلف نیست. ج) برای عبارات $T(OneOf(p,q),i)$ و $T(Either(p,q),i)$ یک تعریف رسمی ارائه دهید. د) توضیح دهید که چرا گمان می‌شود که برای نقیض رویدادها باید دو شکل داشته باشیم، مشابه با دو شکل برای عطف. آن دو شکل را Not و Never نامیده و برای آنها تعریفی رسمی ارائه دهید.**

☑ حل: **الف)** عبارت $T(c,i)$ بدان معناست که برخی رخدادها از نوع $c$ در بازهٔ زمانی $i$ رخ داده اند:

$i$:

$$\forall c,i \;\; T(c,i) \;\Leftrightarrow\; (\exists e \;\; e \in c \wedge During(i,e))$$

استفاده از $SubEvent$ به عنوان درخواست سئوال ساده نیست زیرا این بازه تمام رخدادهای درون آن را شامل می‌شود. **ب)** $Both(p,q)$ رخدادی است که می‌گوید در طی رخداد، هم $p$ وهم $q$ اتفاق افتاده‌اند. فقط یک راه برای این اتفاق وجود دارد: هم $p$ و هم $q$ باید در کل بازه برقرار باشند. راه دیگر آن است که:

$$\forall i,j \;\; During(j,i) \;\Rightarrow\; [T(p,j) \wedge T(q,j)]$$

که به طور منطقی معادل است با:

$$[\forall i,j \;\; During(j,i) \;\Rightarrow\; T(p,j)] \wedge [\forall i,j \;\; During(j,i) \;\Rightarrow\; T(q,j)]$$

که این معادله برای فصل‌ها شکست می‌خورد (به بخش بعدی مراجعه کنید). **ج)** عبارت $T(OneOf(p,q),i)$ بدان معناست که در طی بازه $i$، یک رخداد $p$ یا یک رخداد $q$ اتفاق می‌افتد.

$$\forall p,q,i \;\; T(OneOf(p,q),i) \;\Leftrightarrow$$
$$[(\forall j \;\; During(j,i) \;\Rightarrow\; T(p,j)) \vee (\forall j \;\; During(j,i) \;\Rightarrow\; T(q,j))]$$

به بیانی دیگر، $T(Either(p,q),i)$ در صورتی برقرار است که در هر نقطه‌ای از بازه $i$، یک $p$ یا یک $q$ در حال وقوع باشد:

$$\forall p,q,i \;\; T(OneOf(p,q),i) \;\Leftrightarrow\; (\forall j \;\; During(j,i) \;\Rightarrow\; (T(p,j) \vee T(q,j)))$$

**د)** $T(Never(p),i)$ بدان معناست که هرگز یک رخداد از نوع $p$ در هر زیر بازه‌ای از $i$ رخ نمی‌دهد و $T(Not(p),i)$ بدان معناست که هیچ تکرخدادی از نوع $p$ وجود ندارد که تمام بازهٔ $i$ را پوشش دهد. حتی ممکن است یک یا چند رخداد از نوع $p$ برای هر زیربازهٔ $i$ نیز موجود نباشد:

**vals of $i$:**

$$\forall p,i \;\; T(Never(p),i) \;\Leftrightarrow\; \neg\exists j \;\; During(j,i) \wedge T(p,j)$$
$$\forall p,i \;\; T(Not(p),i) \;\Leftrightarrow\; \neg T(p,i)$$

اساتید می‌توانند از دانشجویان بخواهند تا دو نوع نسخهٔ قانون مورگان را با استفاده از این دو نوع نقیض و فصل مربوطه به آنها، اثبات کنند.

**؟ ۱۱.۱۰ مسند $Fixed$ را به طوری تعریف کنید که $Fixed(Location(x))$ به معنای آن باشد که مکان شی $x$ در طی زمان ثابت است. *(این تمرین مشابه تمرین ۱۲.۱۵ در ویرایش سوم است)***

☑ حل: هر شی $x$ یک رویداد است و $Location(x)$ رویدادی است که برای هر زیربازه از زمان، به مکان x اشاره می‌کند. به عنوان مثال $Location(Peter)$ یک رویداد پیچیده است و شامل خانهٔ او از نیمه شب تا ساعت 9 امروز، اجزای مختلف مسیر جاده و دفتر کارش از ساعت 10 تا 1:30 می‌شود. برای گفتن آنکه یک رویداد ثابت است باید بگوییم که در هر دو لحظه از رویداد یکسان هستند:

$$\forall e \;\; Fixed(e) \;\Leftrightarrow$$
$$(\forall a,b \;\; a \in Moments \wedge b \in Moments \wedge Subevent(a,e) \wedge Subevent(b,e)$$
$$\Rightarrow\; SpatialExtent(a) = SpatialExtent(b))$$

**؟ ۱۲.۱۰ مسندهای $Before$، $After$، $During$ و $Overlop$ را با استفاده از مسند $Meet$ و توابع $Start$ و $End$ تعریف کنید ولی از تابع $Time$ و یا مسند $<$ استفاده نکنید.**

☑ حل: باید متغیرهای سراسری را حذف کنیم:

$$
\begin{array}{lcl}
Before(i,j) & \Leftrightarrow & \exists k \;\; Meet(i,k) \wedge Meet(k,j) \\
After(i,j) & \Leftrightarrow & Before(j,i) \\
During(i,j) & \Leftrightarrow & \exists k,m \;\; Meet(Start(j),k) \wedge Meet(k,i) \\
 & & \wedge Meet(i,m) \wedge Meet(m,End(j)) \\
Overlap(i,j) & \Leftrightarrow & \exists k \;\; During(k,i) \wedge During(k,j)
\end{array}
$$

**۱۳.۱۰ بخش ۵.۱۰ کتاب برای توصیف ارتباط بین صفحات وب از مسندهای Link و LinkText استفاده کرده است. اکنون با استفاده از مسندهای *Intag* و *Getpage* تعاریفی برای *Link* و *LinkText* ارائه دهید.**

حل:

$$
\begin{array}{l}
Link(url_1, url_2) \Leftrightarrow \\
\quad InTag(\text{“a”}, str, GetPage(utl_1)) \wedge In(\text{“}href = \text{“”} + url_2 + \text{“”“}, str) \\
LinkText(url_1, url_2, text) \Leftrightarrow \\
\quad InTag(\text{“a”}, str, GetPage(utl_1)) \wedge In(\text{“}href = \text{“”} + url_2 + \text{“”“} + text, str)
\end{array}
$$

**۱۴.۱۰ یکی از بخش‌های فرآیند خرید که در این فصل بحث نشد، بررسی سازگاری بین کالاهای خریداری شده است. به عنوان مثال، اگر مشتری یک کامپیوتر سفارش می‌دهد آیا قطعات آن هم‌خوانی دارند؟ اگر یک دوربین دیجیتال سفارش می‌دهد، آیا ابزان حافظه و باطری آن مناسب است؟ پایگاه دانشی بنویسید که تشخیص دهد یک مجموعه کالا سازگار هستند یا خیر و واند پیشنهاد دهد که در صورت ناسازگاری، کدام کالا باید جایگزین یا اضافه شود. اطمینان حاصل کنید که پایگاه دانش شما گونه‌ای باشد که حداقل با یک کالا نیز بدرستی کار کرده و بتوان آن را به سادگی برای اقلام بیشتر گسترش داد. *(این تمرین شابه تمرین ۱۲.۲۳ در ویرایش سوم است)***

حل: در اینجا طرح کلی از یک روش را بیان می‌کنیم ولی روش‌های دیگری نیز وجود دارد. هرشی که سفارش داده می‌شود مکن است برخی اجزای اضافی نیاز داشته باشد (مانند باطری) تا بتواند بدرستی کار کند و همچنین ممکن است بتوان اجزائی خواه را به آن افزود. در اینجا نیازمندی را به عنوان رابطهٔ بین یک شی و یک کلاس از شی نمایش می‌دهیم که با تعداد اشیایی ه به آن نیاز دارند، مقدار می‌گیرد:

$$\forall x \;\; x \in Coolpix995DigitalCamera \;\Rightarrow\; Requires(x, AABattery, 4)$$

همچنین نیاز به روشی داریم تا بدانیم یک شی سازگار هست یا خیر. یعنی نقش مورد نظر خود را ایفا می‌کند یا خیر. به عنوان ثال:

$$
\begin{array}{l}
\forall x,y \;\; x \in Coolpix995DigitalCamera \wedge y \in DuracellAABattery \\
\quad \Rightarrow Compatible(y, x, AABattery)
\end{array}
$$

ابراین به سادگی می‌توان مجموعه‌ای از اقلام سفارش داده شده را از جهت سازگاری (یعنی اشیای مورد نیاز هر شی)، بررسی ماییم.

**۱۵.۱۰ قواعدی بیافزایید که با استفاده از آنها بتوان تعریف مسند Name(S,C) را به طوری گسترش داد که رشته‌ای نظیر «Laptop Cpmputer» با نام طبقاتی از فروشگاههای مختلف منطبق شود. سعی کنید تعریفی عمومی ارائه دهید. به ده وشگاه آنلاین مراجعه کرده و با توجه به نام طبقاتی که آنها برای سه دسته مختلف گذاشته‌اند، تعریف خود را تست کنید.**

**نوان مثال، برای طبقهٔ لپ‌تاپ، نام‌هایی چون «Note Book»، «Laptop»، «Note Book Computer»، «Note Books» و «Note Books and Laptop» مشاهده می‌شود. برخی از این نام‌ها صریحا آن نام مربوطه را پوشش می‌دهند، ولی برخی دیگر استفاده از قواعد جمع، عطف و ... تولید شده‌اند. *(این تمرین مشابه تمرین ۱۰.۱۲ در ویرایش سوم است)***

حل: ترکیبات جمع توسط یک رابطهٔ *Plural* بین رشته‌ها تولید می‌شود:

$$Plural(\text{“}computer\text{”}, \text{“}computers\text{”})$$

علاوه می‌دانیم که ترکیب جمعی یک اسم، باز هم برای آن دسته یک اسم محسوب می‌شود:

$$\forall c, s_1, s_2 \;\; Name(s_1, c) \wedge (Plural(s_1, s_2) \vee Plural(s_2, s_1)) \;\Rightarrow\; Name(s_2, c)$$

کیب عطفی، در صورتی نام یک طبقه خواهد بود که یکی از عطف‌ها، نامی برای یک طبقه باشد:

$$\forall c, s, s_2 \;\; Conjunct(s_2, s) \wedge Name(s_2, c) \;\Rightarrow\; Name(s, c)$$

ه *Conjunct* از روی اصطلاحات الصاق رشته‌ای تعریف می‌شود. البته بهتر بود که به جای این کار، عبارت *RelevantCategoryNam* را مجددا تعریف می‌کردیم.

**۱۶.۱۰ یافتن یک راه‌حل کامل برای مسأله غیر دقیق انطباق در توصیف خریدار در فروشگاه کاری بس دشوار بوده و نیاز به جموعه کاملی از فرآیندهای پردازش زبان طبیعی و تکنیک‌های بازیابی اطلاعات دارد. یک کار ساده آن است که از کاربر**

**بخواهیم برای خصوصیات مختلف، مقدار مینیمم و ماکزیمم را مشخص کند. همچنین از خریدار می‌خواهیم از گرامر زیر برای توصیف محصولات استفاده کند:**

***Description→Category[Connector Modifier]****
***Connector→"with" | "and" | ","***
***Modfier→ Attribute | Attribute Op Value***
***Op→*** "=" | ">" | "<"

**در اینجا *Category* همان طبقه کالا، *Attribute* یک مشخصه از کالا نظیر *CPU* یا قیمت، و *Value* مقدار نهایی آن مشخصه (خصوصیت) می‌باشد.بنابراین پرس‌وجویی نظیر «کامپیوتری با حداقل یک *CPU* 2.5 گیگا هرتزی زیر 1000 دلار» به صورت «کامپیوتری با *CPU*>2.5GHz و price<1000$» تبدیل می‌شود. یک کارگزار فروشگاه پیاده‌سازی کنید که توصیفات این زبان را بپذیرد. *(این تمرین مشابه تمرین ۲۴.۱۲ در ویرایش سوم است)***

☑ حل: می‌توان یک رشته متنی را پیمایش کرده و اطلاعات معنایی خاصی را از آن استخراج کنیم. خروجی این فرآیند، تعریفی از اشیای مورد قبول برای کاربر در یک خرید خاص خواهد بود. این کار به روبات کمک می‌کند تا به فروشگاه مراجعه کرده و اقلام منطبق با نیازهای خود را بیابد. در اینجا از تعریف کامل این کار گزار صرف نظر کردیم ولی مراجعه به عملیات موجود در پروژه AIMA صفحات وب، برایتان مفید خواهد بود.

**؟ ۱۷.۱۰ در توصیفی که ما از خرید اینترنتی ارائه دادیم، تمام مراحل مهم خرید واقعی کالا را حذف کردیم. یک توصیف منطقی رسمی برای خرید با استفاده از محاسبات رویدادها ارائه دهید که در آن دنبالهٔ رویدادها از زمانی که کاربر یک سفارش را با کارت اعتباری ارسال می‌کند تا زمانی که صورتحساب و سرانجام کالا را تحویل می‌گیرد، گنجانده شده باشد. *(این تمرین مشابه تمرین ۲۵.۱۲ در ویرایش سوم است)***

☑ حل: در اینجا نسخهٔ ساده‌ای از پاسخ را می‌بینید زیرا اگر به جزئیات بپردازیم، پاسخ بسیار طولانی و شاید نامتناهی گردد. اصطلاح $Buy(b,x,s,p)$ را به عنوان طبقه رویدادی تعریف می‌کنیم که خریدار $b$، شی $x$ را از فروشنده $s$ با قیمت $p$ می‌خرد. می‌خواهیم بگوییم که $b$ پولی به $s$ داده و $s$ مالکیت $x$ را به $b$ می‌دهد:

$$T(Buy(b,x,s,p),i) \Leftrightarrow$$
$$T(Owns(s,x),Start(i)) \wedge$$
$$\exists m \; Money(m) \wedge p = Value(m) \wedge T(Owns(b,m),Start(i)) \wedge$$
$$T(Owns(b,x),End(i)) \wedge T(Owns(s,m),End(i))$$

**؟ ۱۸.۱۰ رویداد «معاملهٔ یک چیز یا چیز دیگر» را توصیف کنید. همچنین خرید را به عنوان نوعی معامله توصیف کنید که شیئی که در آن داد و ستد می‌شود، مقداری پول است. *(این تمرین مشابه تمرین ۱۶.۱۲ در ویرایش سوم است)***

☑ حل: فرض کنید $Trade(b,x,a,y)$ نشان دهندهٔ طبقه‌ای از رخدادها باشد که شخص $b$، شی $y$ خود را با شی $x$ از شخص $a$ معامله می‌کند:

$$T(Trade(b,x,a,y),i) \Leftrightarrow$$
$$T(Owns(b,y),Start(i)) \wedge T(Owns(a,x),Start(i)) \wedge$$
$$T(Owns(b,x),End(i)) \wedge T(Owns(a,y),End(i))$$

اکنون برای آنکه خرید را به عنوان نوعی معامله توصیف کنیم باید قیمت را از روی مجموعه‌ای حقیقی از پول‌ها مشخص کنیم:

$$T(Buy(b,x,a,p),i) \Leftrightarrow \exists m \; Money(m) \wedge Trade(b,x,a,m) \wedge Value(m) = p$$

**؟ ۱۹.۱۰ مفهوم مالکیت در دو تمرین قبل بسیار ابتدایی بحث شد. به عنوان مثال، در آغاز خریدار مالک اسکناس‌های دلار بود.این تصور در زمانی که مقداری از اسکناس‌ها در بانک باشد اشتباه خواهد شد زیرا شخص به مجموعه اسکناسی بیشتر از آنچه در اختیار دارد، مالکیت ندارد. با افزودن مفاهیمی چون قرض دادن، اجاره کردن، کرایه دادن، و امانت، این تصور دچار مشکلات بیشتری خواهد شد. دربارهٔ انواع مفاهیم قانونی و بدیهی مالکیت تحقیق کرده و طرحی ارائه دهید که بتوان این موارد را در آن به طور رسمی بازنمایی نمود. *(این تمرین مشابه تمرین ۱۷.۱۲ در ویرایش سوم است)***

☑ حل: برخی از نکات کلیدی عبارتند از: • مالکیت در هر زمانی اتفاق می‌افتد پس به روش‌های محاسبات موقعیت یا محاسبات بازه‌ای نیازمندیم. • شاید بتوان مالکیت و مالکیت حقوقی را به هم پیوند داد در آنصورت مالک، گروهی هم نوع تعریف می‌شود ولی در حالت ساده، مالک گروهی متشکل از یک نفر می‌باشد. • مالکیت شامل برخی کارها خواهد بود: استفاده کردن، فروش مجدد، بخشیدن و ..... بسیاری از این کارها به خودی خود در تعریف گنجانده نمی‌شوند ولی میزان مناسب بودن یک پاسخ، در گرو بازنمایی تعداد بیشتری از این موارد است. • مالکیت می‌تواند به معنای مالکیت اشیای مرتبط تفسیر شود. این ایده‌ی بانک‌ها و سوپرمارکت‌های آینده خواهد بود. زمانی که شما یک دلار در بانک سپرده می‌گذارید، شما مالک آن دلار خاص هستید و در زمان پس گرفتن، مالک دلار دیگری هستید که به شما داده می‌شود. (البته تصادفا ممکن است همان دلار خاص به شما برگردانده شود). مفاهیمی نظیر اجاره به همین روش تفسیر می‌شوند که نوعی بازنمایی هستند زیرا باید تبادلاتی این چنینی تعریف شوند. در این مورد $Withdraw(person,money,bank,time)$ یک شی بوده و مسند نیست.

**۲۰.۱۰ می‌خواهید سیستمی بسازید که در آن به دانشجویان کامپیوتر یک برنامهٔ درسی پیشنهاد دهد که در آن پیش‌نیازها یت شده باشد.(فراتر از بازهٔ زمانی مورد نظر) (از جدول نیازمندی‌های دروس دانشگاه خود استفاده نمایید). در ابتدا در مورد هنگ واژگان موردر نیاز برای بازنمایی تصمیم‌گیری نموده و سپس مساله را بازنمایی کنید. در مرحله بعد از سیستم خود والی بپرسید که باید یک برنامه معقول آموزشی را به عنوان نتیجه پاسخ دهد. سیستم شما باید به گونه‌ای باشد که برای هر شجو مناسب باشد یعنی از آنها بپرسد که قبلا چه دروسی را گذرانده‌اند و یا معادل کرده‌اند و برنامهٔ نباید دروس تکراری را ی دانشجویان برنامه‌ریزی کند. پیشنهاداتی ارائه دهید که این سیستم ارتقاء یابد. به عنوان مثال اطلاعات حساب و ظرفیت ی دانشجو و یا خوب/بد بودن استاد را به آن بیافزایید. برای هر کدام از این اطلاعات، توضیح دهید که چطور می‌توان آن را به ر منطقی بیان نمود. آیا سیستم شما با افزودن این اطلاعات می‌تواند بهترین برنامه درسی را برای یک دانشجو تولید کند؟**

حل: بیشتر دانشگاه‌ها بین دروس پیش نیاز، دروس اختیاری، دروس درون دانشکده و بیرون دانشکده تمایز قائل می‌شوند و ن هر یک از این موارد، تعداد دروس، تعداد واحد (تعداد واحد دروس مختلف، ممکن است متفاوت باشد) و نمرهٔ میانگین یف می‌شود. فرهنگ واژه‌ای که ما در اینجا به عنوان مثال انتخاب کردیم عبارت است از:

ل دروسی که jones به عنوان دانشجو باید در دانشگاه بگذراند عبارتست از:

Math1,CS1,CS2,CS3,CS21,CS33و CS34 و برخی دروس دیگر که در رشتهٔ او نیست:

$$Take(Jones,\\ \{Math1, EE1, Bio24, CS1, CS2, CS3, CS21, CS33, CS34|others\})$$

Jone جدول پیش‌نیازهای رشتهٔ علوم کامپیوتر را مشاهده می‌کند:

$$Major(Jones, CS)$$

روس Math1,CS1,CS2,CS3 پیش‌نیازهای رشتهٔ علوم کامپیوتر هستند:

$$Required(\{Math1, CS1, CS2, CS3\}, CS)$$
$$\forall s, d\ \ Required(s, d) \Leftrightarrow \\ (\forall p\ \exists others\ \ Major(p, d)\ \Rightarrow\ Take(p, Union(s, others)))$$

ک دانشجو برای دریافت مدرک علوم کامپیوتر باید حداقل ۱۸ واحد در دانشکده علوم واحد اخذ کند:

$$Department(CS1) = CS \wedge Department(Math1) = Math \wedge \ldots$$
$$Units(CS1) = 3 \wedge Units(CS2) = 4 \wedge \ldots$$
$$RequiredUnitsIn(18, CS, CS)$$
$$\forall u, d\ \ RequiredUnitsIn(u, d) \Leftrightarrow \\ (\forall p\ \exists s, others\ \ Major(p, d)\ \Rightarrow\ Take(p, Union(s, others)) \\ \wedge AllInDepartment(s, d) \wedge TotalUnits(s) \geq u$$
$$\forall s, d\ \ AllInDepartment(s, d)\ \Leftrightarrow\ (\forall c\ \ c \in s\ \Rightarrow\ Department(c) = d)$$
$$\forall c\ \ TotalUnits(\{\}) = 0$$
$$\forall c, s\ \ TotalUnits(\{c|s\}) = Units(c) + TotalUnits(s)$$

ه به سادگی می‌توانید نیازمندی‌های دیگری بیافزایید. در اینجا فقط یک نمونه را ذکر کردیم. در این جا جملهٔ « فراتر از بازهٔ ی مورد نظر» را به معنای آن تفسیر می‌کنیم که در مجموعهٔ دروس اخذ شده، زمان‌بندی ترمی نادیده گرفته شود. اگر خواهید این قابلیت را نیز اضافه کنید، باید اطلاعات بیشتری نظیر زمان آموزش درس، زمان بارگذاری درس در ترم و ی‌نیازهای هر درس را لحاظ نمایید. به عنوان مثال:

$$Taught(CS1, Fall)$$
$$Prerequisites(\{CS1, CS2\}, CS3)$$
$$TakeInSemester(Jones, Fall95, \{Math1, CS1, English1, History1\})$$
$$MaxCoursesPerSemester(5)$$

نظور یافتن بهترین برنامهٔ درسی، در ابتدا باید عبارت « بهترین» را برای یک دانشجو تعریف کنیم. یک تعریف ساده آن است گوییم همه چیز بر وفق مراد باشد یعنی استاد یک درس سخت، بسیار خوب باشد و در ساعات خستگی، درسی جذاب داشته د ولی در برخی موارد نمی‌توان تصمیم گرفت که کدام برنامه بهتر است. داشتن درسی با یک استاد خوب و داشتن درسی ه و جذاب و یا درسی کم شهریه! در واقع باید به هر کدام از این عناصر یک مقدار وزنی انتساب دهیم آنگاه برنامه‌ای بهترین ک که بیشترین وزن ممکن را کسب کند. بدون انجام این کار شما مجبور هستید هر دفعه یک برنامه یافته و اگر بدتر از قبلی آن را نادیده بگیرید و نمی‌توانید در یک مرحله بهترین برنامه را بیابید. در مورد پیچیدگی این مسأله باید بگوییم کـه اگـر از

اثبات‌گر تئوری عمومی استفاده کنیم پیچیدگی آن بیشتر از موردی می‌شود که تمام برنامه‌های ممکن را شمرده و سپس بهترین را انتخاب کنیم.

**؟ ۲۱.۱۰ شکل ۱.۱۰ مراحل فوقانی از سلسله مراتب هر چیزی را نشان می‌دهد. آن را گسترش دهید تا در حد امکان، طبقات حقیقی بیشتری را شامل شود. یک روش خوب آن است که تمام چیزهایی که در روزهای زندگیتان با آنها مواجه می‌شوید را در آن قرار دهید. این موضوع هم شامل اشیا و هم رویدادها می‌شود. از آغاز بیدار شدن خود شروع کرده و بترتیب تمام چیزهایی که می‌بینید، لمس می‌کنید، انجام می‌دهید و دربارهٔ آنها می‌اندیشید را ثبت کنید. به عنوان مثال، در یک قسمت تصادفی از زندگی داریم: موزیک، اخبار، شیر، پیاده‌روی، رانندگی، بنزین، ساختمان، Soda Hall، فرش، صحبت کردن، پروفسور فیتمن، ادویه مرغ، زبان، هفت دلار، خورشید، روزنامه روزانه، و ... . شما باید هم یک چارت سلسله مراتبی (در یک برگه بزرگ) و هم لیستی از اشیا و طبقات به همراه رابطه‌ای که بین اعضای هر طبقه وجود دارد تهیه کنید. هر شی باید در یک طبقه بوده و هر طبقه نیز در یک سلسله مراتب واقع باشد.** ***(این تمرین مشابه تمرین ۲.۱۲ در ویرایش سوم است)***

☑ حل: این تمرین و دو تمرین بعدی نسبتا پیچیده بوده و برای انتخاب به عنوان پروژه مناسب هستند. بهتر است خواننده برای امور روزمره زندگی، سلسله مراتب طبقات را یافته و سپس کار خود را با سلسله مراتب‌های آنلاین نظیر سایت یاهو مقایسه کنند.

**؟ ۲۲.۱۰ (با اقتباس از مثال Doug Lenat) شما باید با توجه به جمله زیر دانش را به شکل منطقی بیان کرده و سپس به پرسش‌هایی که در زیر آمده است پاسخ دهید. «دیروز جان به سوپر مارکت Safeway در شعبهٔ شمالی برکلی مراجعه کرده و دو پوند گوجه‌فرنگی و یک پوند گوشت چرخ‌کرده خریداری نمود.» در ابتدا مفهوم جملهٔ فوق را با تعدادی جمله بازنمایی کنید. جملات شما باید ساختار منطقی آسانی داشته باشند (به عنوان مثال، عباراتی که اشیای آن ویژگی‌های مشخص داشته باشند؛ و یا عباراتی که اشیای آن به هم مرتبط باشند؛ و یا عباراتی که تمام اشیای مشترک در یک ویژگی، در ویژگی دیگر هم صدق کنند). این موارد برای شروع به شما کمک می‌کنند: • به چه کلاس‌ها، اشیا و روابطی نیازمندید؟ والدین، هم نژادها و ....کدام هستند؟ (شما به رخدادها و ترتیب زمانی بین آنها نیاز خواهید داشت). • در چه مکانی می‌توان آنها را با یک سلسله مراتب عمومی‌تر تطبیق داد؟ • محدودیت ها و روابط متقابل بین آنها چیست؟ • به چه جزئیاتی از مفاهیم مختلف باید توجه داشت؟ پایگاه دانشی که شما ساخته‌اید باید قابلیت پاسخگویی به لیست سئوالاتی که ما مختصرا بیان می‌کنیم را داشته باشد. برخی از این سئوالات با عناصری که صریحا در داستان ذکر شده‌اند سرو کار دارند ولی بسیاری از آنها نیاز به یک شخص دارای دانش پیش‌زمینه هم دارند تا مفاهیم را از لابه‌لای خطوط استنتاج کند. شما باید با سئوالاتی چون: چه چیزهایی در یک سوپرمارکت وجود دارد؟ فردی که برای انتخاب یک چیز مراجعه می‌کند، چه چیزهایی سفارش می‌دهد؟ چه چیزهایی برای سفارش استفاده می‌شوند و ... مواجه شوید. بکوشید تا حد امکان بازنمایی خود را عمومی کنید. به یک مثال ساده توجه کنید: به عنوان مثال بازنمایی جملهٔ «مردم غذا را از Safeway می‌خرند» کم کاربرد است زیرا برای افرادی که از دیگر مغازه‌ها خرید می‌کنند بکار نمی‌رود و یا بازنمایی جملهٔ «جو اسپاگتی را از گوجه و گوشت‌چرخکرده درست می‌کند» نیز کم‌اهمیت است زیرا درست کردن اسپاگتی با هر چیز دیگر را شامل نمی‌شود. سعی کنید این نکته را در بازنمایی سئوالات رعایت کنید مثلا سئوال ج می‌پرسد «آیا جان مقداری گوشت خرید؟» نه اینکه «آیا جان یک پوند گوشت چرخکرده خرید»؟ زنجیره استدلال خود را گسترش دهید تا بتوانید به سئوالات پاسخ دهید. در طی فرآیند در مورد افزودن مفاهیم، ایجاد قوانین بیشتر و غیره دچار تردید نشوید. در صورت امکان از یک سیستم استنتاج منطقی برای نمایش کارایی پایگاه دانش خود کمک بگیرید. بسیاری از آنچه شما می‌نویسید ممکن است در دنیای حقیقی تخمینی باشد ولی نگران نباشید زیرا هدف ما فقط استخراج چیزی است که بتواند به این سئوالات پاسخ دهد و پاسخگویی دقیق و کامل به این سئوالات واقعا دشوار بوده و شاید فراتر از بازنمایی فعلی دانش باشد. ولی شما باید بتوانید یک مجموعه سازگار از قوانین برای این سئوالات محدود تشکیل دهید: الف) جان یک کودک است و یا یک بزرگسال؟ [بزرگسال] ب) آیا هم‌اکنون جان حداقل دو گوجه‌فرنگی دارد؟[بله] ج) آیا جان مقداری گوشت خرید؟ [بله] د) اگر ماری در همان زمانی که جان گوجه‌فرنگی خرید در حال خرید بوده باشد، آیا جان او را دیده است؟ [بله] هـ) آیا گوجه‌فرنگی‌ها در سوپرمارکت تولید می‌شوند؟ [خیر] و) جان قصد دارد با گوجه‌فرنگی‌ها چه کند؟ [آنها را بخورد] ز) آیا Safeway، بوگیر می‌فروشد؟ [بله] ح) آیا جان مقداری پول به سوپرمارکت آورده است؟ [بله] ط) آیا مقدار پول جان کمتر از مقدار پول او قبل از آمدن به سوپرمارکت است؟ [بله]** ***(این تمرین مشابه تمرین ۵.۱۲ در ویرایش سوم است)***

☑ حل: این مسأله به بازنمایی خیلی زیادی نیاز دارد لذا پیشنهاد می‌شود تیم‌های 2 الی 3 نفره تشکیل شده و با حداقل 2 هفته زمان، این کار را انجام دهند.

**؟ ۲۳.۱۰ پایگاه دانش تمرین قبل را به گونه‌ای تغییر دهید یا اضافه کنید که پاسخگوی سئوالات زیر باشد. نشان دهید که آنها می‌توانند به راستی با پایگاه دانش پاسخ داده شوند و در توضیحات خود، در مورد موارد قطعی، دلیل نیاز ما به آنها، اصلی و فرعی بودن آنها و غیره بحث کنید. الف) آیا در زمانی که جان در Safeway بود، افراد دیگری هم آنجا بودند؟ [بله، کارمند] ب) آیا جان گیاه‌خوار است؟ [خیر] ج) چه کسی صاحب بوگیر در Safeway است؟ [شرکت Safeway] د) آیا جان مقدار کمی گوشت چرخکرده دارد؟ [بله] هـ) آیا جایگاه بعدی Shell، بنزین دارد؟ [بله] و) آیا گوجه‌فرنگی‌ها در صندوق ماشین جان، جا می‌شوند؟ [بله]** ***(این تمرین مشابه تمرین ۶.۱۲ در ویرایش سوم است)***

☑ حل: پاسخ این تمرین در گرو پاسخ مناسب به تمرین قبل است زیرا با تغییر و کمی افزودنی به آن به این پاسخ دست خواهیم یافت. دانشجویان سعی کنند در بازنمایی خود تا حد امکان عمومی و درست بنویسند تا در این تمرینات نیز قابل استفاده باشد و تشخیص اشتباه و تغییرات ساده‌تر شود.

**۲۴.۱۰ می‌دانیم که اطلاعات وراثتی در شبکه‌های معنایی می‌تواند با عبارات استلزامی مناسب به طور منطقی بیان شود. در ن تمرین، کارایی استفاده از چنین عباراتی را برای وراثت بررسی می‌کنیم: الف) اطلاعات موجود در کاتالوگ‌های ماشین‌های یمی نظیر Kelly's Blue Book را در نظر بگیرید. به عنوان مثال، «ون‌های Dodge سال 1973 کمتر از 575 دلار ارزیدند». فرض کنید تمام این اطلاعات را (برای 11000 مدل) طبق آنچه که در این فصل گفته شد، با قواعد رمزنگاری ده‌ایم. اکنون سه قانون مشابه آنها بنویسید که برای ون Dodge سال 1973 باشد. اگر یک اثبات‌کنندهٔ قضیه زنجیره‌ای پس‌رو یر پرولوگ در اختیار داشته باشید، چگونه می‌توانید با استفاده از این قواعد، ارزش یک ماشین خاص را بیابید؟( مثلا برای شین JB که نوعی ون Dodge در سال 1973 است). ب) پیچیدگی زمانی روش زنجیره‌ای پس‌رو را با روش وراثتی که در یکه‌های معنایی استفاده می‌شود، برای این مسأله مقایسه کنید. ج) توضیح دهید که چگونه در یک سیستم مبتنی بر منطق، ش زنجیره‌ای پیش‌رو می‌تواند چنین مسائلی را به طور کارآمد حل کند؟ فرض کنید پایگاه دانش فقط شامل 11000 قانون در رد قیمت ماشین‌ها باشد. د) موقعیتی را تشریح کنید که در آن اگر از قیمت ماشینی خاص پرسش شود، نه روش زنجیره‌ای ش‌رو و نه زنجیره‌ای پس رو نتوانند به طور کارا پاسخگو باشند؟ هـ) آیا می‌توانید یک راه‌حل برای این نوع سئوالات ارائه دهید در تمام موارد در سیستم منطقی به طور کارا حل شوند؟ [ راهنمایی: یادآوری می‌شود که دو ماشین از یک دسته، قیمت سان دارند]. *(این تمرین مشابه تمرین ۲۱.۱۲ در ویرایش سوم است)***

ا حل: در بسیاری از کتاب‌های هوش‌مصنوعی و پرولوگ می‌بینیم که از استلزام برای پیاده‌سازی وراثت استفاده کرده‌اند. این ر به طور منطقی درست است ولی در عمل خیر. **الف)** در اینجا سه قانون پرولوگ را می‌بینید. البته در حقیقت باید به سمت ت این عبارت، بندهای بیشتری اضافه کنیم تا بتوان تمایز بین مدل‌ها، اختیارات و ... را بیان نمود.

```
worth(X,575) :- year(X,1973), make(X,dodge), style(X,van).
worth(X,27000) :- year(X,1994), make(X,lexus), style(X,sedan).
worth(X,5000) :- year(X,1987), make(X,toyota), style(X,sedan).
```

ی دانستن قیمت JB با فرض آنکه در پایگاه داده عبارات زیر موجود باشد:

style(jb,van) , make(jb,dodge),yaer(jb,1973)

اه اگر بخواهیم با روش زنجیره‌ای پس‌رو به جواب برسیم از هدف worth(jb,D) شروع کرده و در نهایت به قیمت D رسیم. **ب)** پیچیدگی زمانی این پرسش $O(n)$ است که $n$ همان 11000 ورودی کاتالوگ Blue Book است، یک شبکه نایی با وراثت اجازه می‌دهد که با پیگیری لینک‌ها از JB به 1973-dodge-van داشته باشیم و سپس از آنجا لینکی به wor که موجب می‌شود به قیمت مربوطه با زمان $O(1)$ برسیم. **ج)** با فرض آنکه پایگاه دانش شامل سه عبارتی است که قبلا تیم اگر بخواهیم با روش زنجیره‌ای پیش‌رو به جواب برسیم باید عبارت worth (jb,575) را به مجموعه حقایق بیافزاییم. س به سئوال worth(gb,D) می‌رسیم که با $O(1)$ به پاسخ دست می‌یابیم. این کار با ایندکس‌گذاری روی مسند و اولین گومان صورت می‌گیرد. به نظر می‌رسد استنتاج منطقی شبیه شبکه معنایی باشد با دو تفاوت: استنتاج منطقی از یک جدول ستجوی هش به جای پیگیری اشاره‌گرها استفاده می‌کند. همچنین استنتاج منطقی عبارات worth را برای هر ماشین ذخیره کند. لذا اگر تعداد ماشین‌ها زیاد باشد فضای زیادی مصرف می‌شود. (در اینگونه موارد ترجیح می‌دهیم به جای 11000 مدل، داد مدل کمتری را لحاظ کنیم). **د)** اگر ماشین‌های هر دسته مشخصات زیادی داشته باشند به عنوان مثال: برای هر قطعه سیون بتوان چندین جایگزین گذاشت آنگاه روش زنجیره‌ای پیش‌رو با متد استلزام نمی‌تواند راه‌کاری عملی در تعیین قیمت ماشین باشد. **هـ)** اگر قانونی از نوع زیر داشته باشیم:

```
worth(X,D) :- year-make-style(X,Yr,Mk,St),
              year-make-style(Y,Yr,Mk,St), worth(Y,D)
```

همراه سایر حقایق موجود در بانک اطلاعاتی در مورد سایر انواع ماشین‌های هم‌نوع با J B ،آنگاه سئوال worth (jb,D) در ن $O(1)$ ایندکس‌گذاری مناسب حل خواهدشد. صرف نظر از سایر حقایقی که در مورد آن ماشین می‌دانیم و تعداد انواع شین‌ها.

**۲۵.۱۰ ممکن است گمان کنید تشخیص نحو بین لینک‌های غیرمربعی و لینک‌های تک‌مربعی در شبکه‌معنایی نیاز نباشد زیرا ک‌های تک‌مربعی همواره به طبقات متصل اند. لذا یک الگوریتم وراثت به سادگی می‌تواند متوجه شود که یک لینک رمربعی اگر به یک طبقه متصل شود، قصد دارد تا به تمام اعضای آن طبقه اعمال شود. نشان دهید که این استدلال غلط، بسطه آمیز بوده و مثال‌هایی از خطاهایی که ممکن است بروز دهد را بیان کنید. *(این تمرین مشابه تمرین ۲۲.۱۲ در ویرایش وم است)***

حل: زمانی که طبقات به شی تبدیل می‌شوند، می‌توانند به عنوان اشیای فردی دارای مشخصاتی باشند (مانند *Cardinality* *Superset* که به عناصر داخلی آنها اعمال نمی‌شود. بدون تشخیص بین لینک‌های مربعی و غیرمربعی، رت *Cardinality*(*SingletonSets*,1) ممکن است بدان معنا تفسیر شود که هر مجموعه یکتا فقط یک عنصر دارد و یا ه فقط یک مجموعه یکتا وجود دارد.

# فصل ۱۱ (ویرایش دوم)

**؟ ۱.۱۱ تفاوت‌ها و شباهت‌های بین حل مسأله و برنامه‌ریزی را بیان کنید.** ***(این تمرین مشابه تمرین ۱.۱۰ در ویرایش سوم است)***

☑ حل: حل‌کننده مسأله و برنامه‌ریز، هر دو از نقطه‌ای شروع کرده و با استفاده از مجموعه عملگرها و اقدامات تعریف شده، قصد رسیدن به هدف را دارند که معمولا آنها را در محیط‌های قطعی، گسسته و مشاهده‌پذیر بکار می‌بریم. در برنامه‌ریزی، بازنمایی حالات، اهداف و برنامه را داریم که موجب می‌شود انواع الگوریتم‌ها بتوانند فضای جستجو را با روش‌های پس‌رو یا پیش‌رو جستجو کرده و همچنین از تولید خودکار تابع هیورستیک بهره ببرند.

**؟ ۲.۱۱ نمودار واکنش و حالت شروع که در شکل ۲.۱۱ ترسیم شده است، موجود است. اگر حالت به صورت زیر توصیف شود، تمام نمونه‌های کاربردی واقعی از $Fly(p,from,to)$ را بیابید.**

$$At(P_1,JFK) \wedge At(P_2,SFO) \wedge Plane(P_1) \wedge Plane(P_2) \wedge Airport(JFK) \wedge Airport(SFO)?$$

***(این تمرین مشابه تمرین ۲.۱۰ در ویرایش سوم است)***

☑ حل: منظور از کلمهٔ «نمونه کاربردی» آن است که نمونه تولیدی پیش‌شرط‌ها را ارضا کند و قابل اجرا باشد و منظور از «نمونه واقعی» آن است که متغیرهای آن توسط ثوابت جایگزین شده باشند. واکنش‌های کاربردی عبارتند از:

$$Fly(P_1, JFK, SFO)$$
$$Fly(P_1, JFK, JFK)$$
$$Fly(P_2, SFO, JFK)$$
$$Fly(P_2, SFO, SFO)$$

یک نکتهٔ ظریف در این مبحث قرار دارد و آن «پرواز به هیچ کجا» است یعنی از یک فرودگاه به خودش، که طبق تعریف $Fly$ مجاز بوده و قابل‌اجرا است (اگر چه بی فایده است).

**؟ ۳.۱۱ چگونگی تبدیل یک مجموعه طرح STRIPS به قوانین حالت پسین در محاسبات موقعیت را در نظر بگیرید. • طرح‌وارهٔ $Fly(p,from,to)$ را در نظر بگیرید. تعریفی منطقی برای مسند $FlyPrecond(p,from,to,s)$ بنویسید که در صورتی درست باشد که پیش شرط‌های $Fly(p,from,to)$ در موقعیت $s$ ارضا شوند. • سپس، فرض کنید $Fly(p,from,to)$ تنها طرح واکنش قابل دسترس برای کارگزار باشد. یک قانون حالت پسین برای $At(p,x,s)$ بنویسید که همان اطلاعات طرح واکنش را دربرگیرد. • اکنون فرض کنید یک روش دیگر برای حمل‌ونقل موجود باشد: $Teleport(p,from,to)$. این عبارت پیش‌شرط $\neg Warped(p)$ و تاثیر $Warped(p)$ را اضافه بر قبل دارد. توضیح دهید که پایگاه دانش محاسبات موقعیت چگونه باید تغییر کند. • در نهایت یک روال کلی و دقیق برای انجام حمل‌ونقل بین یک مجموعه طرح STRIPS به مجموعه دیگری از حالت پسین ارائه دهید.** ***(این تمرین مشابه تمرین ۱.۱۰ در ویرایش سوم است)***

☑ حل: برای یک طرح قاعده‌مند داریم:

$$FlyPrecond(p, f, to, s) \Leftrightarrow$$
$$At(p, f, s) \wedge Plane(p) \wedge Airport(f) \wedge Airport(to)$$
$$At(p, x, Result(a, s)) \Leftrightarrow$$
$$(At(p, x, s) \wedge (a \neq Fly(p, f, x) \vee \neg FlyPrecond(p, f, x, s)))$$
$$\vee At(p, f, s) \wedge a = Fly(p, f, x) \wedge FlyPrecond(p, f, x, s)$$

پس از افزودن $Warped$ داریم:

$$At(p, x, Result(a, s)) \Leftrightarrow$$
$$(At(p, x, s) \wedge (a \neq Fly(p, f, x) \wedge a \neq Teleport(p, f, x))$$
$$\vee a = Fly(p, f, x) \wedge \neg FlyPrecond(p, f, x, s)$$
$$\vee a = Teleport(p, f, x) \wedge \neg TeleportPrecond(p, f, x, s)$$
$$\vee At(p, f, s) \wedge a = Fly(p, f, x) \wedge FlyPrecond(p, f, x, s)$$
$$\vee At(p, f, s) \wedge a = Teleport(p, f, x) \wedge TeleportPrecond(p, f, x, s)$$

در حالت کلی برای هر واکنش، یک مسند $Precond$ ساخته و سپس برای هر fluent نظیر $At$، مسندی می‌سازیم که می‌گوید این fluent مقدار قبلی خود را حفظ می‌کند حتی اگر یک واکنش نامربوط رخ دهد و یا واکنشی که پیش‌شروط آن ارضا نشده است. این مسند زمانی مقدار جدید می‌پذیرد که پیش‌شروط واکنش ارضا شده باشد که این مقدار جدید بستگی به تاثیر واکنش مربوطه دارد.

**؟ ۴.۱۱ مسأله میمون و موزها بدین صورت است که یک میمون در یک آزمایشگاه با تعدادی موز که از سقفی بلند آویخته شده است قرار می‌گیرد. یک جعبه در آنجا قرار دارد که میمون با بالا رفتن از آن، می‌تواند به موزها دسترسی یابد. در ابتدا میمون در نقطه A، موزها در B و جعبه در C قرار دارد. قد میمون و جعبه هر دو کوتاه است ولی با رفتن میمون بر روی جعبه، مجموع قد آنها به قدری بلند می‌شود که به اندازهٔ موزها برسد. اقداماتی که میمون می‌تواند انجام دهد شامل حرکت از یک مکان به مکان دیگر(*Go*)، هل دادن یک شی از یک مکان به مکان دیگر (*Push*)، بالا رفتن (*ClimbUp*) و پایین آمدن از یک شی(*ClimbD*)، برداشتن(*Grasp*) و یا رها کردن (*Ungrasp*) یک شی، می‌باشد. اگر میمون و شی‌ای که می‌گیرد در یک ارتفاع قرار داشته باشند، میمون پس از گرفتن آن شی، آن را نگه می‌دارد. الف) توصیفی برای حالت شروع بنویسید. ب) تعاریفی به شکل STRIPS برای این شش واکنش بیان کنید. ج) فرض کنید میمون قصد فریب دانشمندان را دارد یعنی زمانی که آنها در حال چای خوردن و استراحت هستند بدون آنکه جعبه را در جای اصلی‌اش بگذارد، موزها را چنگ انداخته و برمی‌دارد، این موضوع را به عنوان یک هدف کلی (یعنی لزوما جعبه در محل C نباشد) به زبان محاسبات موقعیت بیان کنید. آیا این هدف می‌تواند توسط یک سیستم کلاسیک برنامه‌ریزی حل شود؟ د) احتمالا طرحی که شما از اقدام هل دادن دارید، نادرست است. زیرا اگر شیئی خیلی سنگین برداشته باشیم، با اعمال اقدام *Push* هیچ تأثیری در موقعیت آن حاصل نمی‌شود.آیا این موضوع مثالی از مسائل انشعابی و یا مسائل مشروط است؟ طرح خود را از اقدامات به گونه‌ای اصلاح کنید که اشیای سنگین را نیز در بربگیرد. *(این تمرین مشابه تمرین ۳.۱۰ در ویرایش سوم است)***

☑ حل: این مساله مشابه مساله Shakey (۳.۱۱) می‌باشد. بنابراین تنها کافیست این دو مساله را با هم تطبیق دهید تا به پاسخ برسید. **الف)** حالت شروع عبارتست از:

$$At(Monkey, A) \wedge At(Bananas, B) \wedge At(Box, C) \wedge$$
$$Height(Monkey, Low) \wedge Height(Box, Low) \wedge Height(Bananas, High) \wedge$$
$$Pushable(Box) \wedge Climbable(Box)$$

**ب)** اقدامات عبارتند از:

$$Action(\text{ACTION:}Go(x, y), \text{PRECOND:}At(Monkey, x),$$
$$\quad \text{EFFECT:}At(Monkey, y) \wedge \neg(At(Monkey, x)))$$
$$Action(\text{ACTION:}Push(b, x, y), \text{PRECOND:}At(Monkey, x) \wedge Pushable(b),$$
$$\quad \text{EFFECT:}At(b, y) \wedge At(Monkey, y) \wedge \neg At(b, x) \wedge \neg At(Monkey, x))$$
$$Action(\text{ACTION:}ClimbUp(b),$$
$$\quad \text{PRECOND:}At(Monkey, x) \wedge At(b, x) \wedge Climbable(b),$$
$$\quad \text{EFFECT:}On(Monkey, b) \wedge \neg Height(Monkey, High))$$
$$Action(\text{ACTION:}Grasp(b),$$
$$\quad \text{PRECOND:}Height(Monkey, h) \wedge Height(b, h)$$
$$\qquad \wedge At(Monkey, x) \wedge At(b, x),$$
$$\quad \text{EFFECT:}Have(Monkey, b))$$
$$Action(\text{ACTION:}ClimbDown(b),$$
$$\quad \text{PRECOND:}On(Monkey, b) \wedge Height(Monkey, High),$$
$$\quad \text{EFFECT:}\neg On(Monkey, b) \wedge \neg Height(Monkey, High)$$
$$\qquad \wedge Height(Monkey, Low)$$
$$Action(\text{ACTION:}UnGrasp(b), \text{PRECOND:}Have(Monkey, b),$$
$$\quad \text{EFFECT:}\neg Have(Monkey, b))$$

**ج)** طبق محاسبات موقعیت، هدف در یک حالت $s$ قرار دارد. نظیر:

$$Have(Monkey, Bananas, s) \wedge (\exists x \;\; At(Box, x, s_0) \wedge At(Box, x, s))$$

در STRIPS ناچاریم که فقط در مورد حالت هدف صحبت کرده و راهی برای بازنمایی حقایق مرتبط بین دو حالت درونی برنامه نداریم. (مانند هم‌ارزی‌های مکانی یک شی). بنابراین راهی برای بازنمایی این هدف نداریم. **د)** در واقع باید پیش شرط *Pushable* را به راه‌حل فوق بیافزاییم.

**؟ ۵.۱۱ توضیح دهید که چرا فرآیند تولید اجداد در جستجوی پسگرد، نیازی به افزودن الفاظی که تأثیر منفی بر روی واکنش‌ها ندارند، ندارد.**

☑ حل: برای یک حالت فقط الفاظ مثبت بازنمایی می‌شوند و عدم ذکر یک لفظ معادل با منفی بودن آن است.

**؟ ۶.۱۱ توضیح دهید که چرا در یک مساله STRIPS، حذف تأثیر منفی برای هر طرح واکنش، منجر به یک مسالهٔ ساده‌تر می‌گردد. *(این تمرین مشابه تمرین ۶.۱۰ در ویرایش سوم است)***

☑ حل: اهداف و پیش شروط، بایستی الفاظی مثبت باشند. بنابراین یک تأثیر منفی فقط رسیدن به هدف را دشوارتر می‌سازد (و یا رسیدن از پیش‌شرط به اقدامی که به هدف می‌رسد). لذا حذف تمام اثرات منفی فقط مساله را ساده‌تر می‌کند. البته اگر پیش‌شروط و اهداف مجاز به منفی باشند، آنگاه دیگر این موضوع درست نخواهد بود.

**؟۷.۱۱ تعریف جستجوی دوطرفه در فصل ۳ را بررسی می‌کنیم: الف) آیا جستجوی دوطرفه فضای حالت برای برنامه‌ریزی، ایده‌ای مناسب است؟ ب) جستجوی دوطرفه فضاهایی با برنامه‌ریزی جزئی چطور است ؟ ج) نسخه ای از برنامه ریزی نیمه مرتب ارائه دهید که در آن یک واکنش در صورتی به یک برنامه افزوده شود که پیش‌شرط آن توسط اثرات واکنش‌های موجود در برنامه، بدست آید. توضیح دهید که چطور با محدودیت‌های تناقض و ترتیب روبرو شویم؟ آیا این الگوریتم برای جستجوی فضای حالت پیش‌رو یکتاست؟ *(این تمرین مشابه تمرین ۱۲.۱۰ در ویرایش سوم است)***

☑ حل: **الف)** استفاده از جستجوی دوطرفه امکان پذیر است زیرا می‌توان در آن واکنش‌ها را معکوس نمود. با این حال بیشتر کارهایی که در این زمینه انجام شده است، نشان می‌دهد که جستجوی دوطرفه در حالت کلی کارا نیست زیرا جستجوهای پس‌رو و پیش‌رو تمایل به عدم وجود دیگری دارند. این موضوع در یک فضای حالت بزرگ مشخص خواهد شد. برخی برنامه‌ریزان نظیر PRODIGY (فنیک و بلیس،۱۹۹۸) از جستجوی دوطرفه استفاده کرده‌اند. **ب)** مجددا می‌گوییم که این موضوع ممکن می‌باشد ولی پرکاربرد و عمومی نیست. PRODIGY در حقیقت جزئی از یک برنامه‌ریز نیمه‌مرتب است: در جهت پیش‌رو می‌تواند یک برنامه کاملا مرتب ارائه دهد و در جهت پس‌رو یک برنامهٔ نیمه‌مرتب با ساختار درختی ارائه می‌دهد. **ج)** واکنش $A$ در صورتی افزوده می‌شود که تمام پیش‌شروط $A$ توسط سایر مراحل برنامه، حاصل شده باشند. زمانیکه $A$ افزوده شد، محدودیت‌های ترتیب و لینک‌های اتفاقی نیز افزوده می‌شوند تا مطمئن شویم که $A$ پس از تمام واکنش‌هایی که آن را فعال می‌کنند، ظاهر می‌شود. همین طور برای پیش‌شرطی که قبل از اجرای $A$ نباید مستقر شود. این الگوریتم جستجوی خود را به صورت پیش‌رو انجام می‌دهد ولی مشابه جستجوی فضای حالت پیش‌رو نیست زیرا در زمانیکه آنها تناقض ندارند، نیز می‌توانند واکنش‌ها را به طور موازی بسط دهد. به عنوان مثال اگر $A$ دارای سه پیش‌شرط باشد که توسط واکنش‌های بدون تناقض $D$ و $C$ و $B$ ارضا شوند، آنگاه راه‌حل برنامه می‌تواند توسط یک برنامه نیمه‌مرتب بازنمایی شود که در آن یک برنامه‌ریز فضای حالت باید تمام !3 جایگشت $D$ و $C$ و $B$ را در نظر بگیرد.

**؟۸.۱۱ برای گراف برنامه‌ریزی در مسالهٔ شکل ۲.۱۱، سطوح 0، 1 و 2 را تشکیل دهید. *(این تمرین مشابه تمرین ۹.۱۰ در ویرایش سوم است)***

☑ حل: ترسیم این سطوح کمی پیچیده و فراتر از فضای کنونی است لذا به بیان برخی نکات اساسی برای این کار بسنده می‌کنیم: ۱) هر دو اقدام *Load* و *Fly* در سطح $A_0$ ممکن هستند زیرا هواپیما با وجود خالی بودن هم می‌تواند پرواز کند. ۲) اثرات منفی در $S_1$ بکار می‌روند و در سایر اجزای مثبت بکار نمی‌رود.

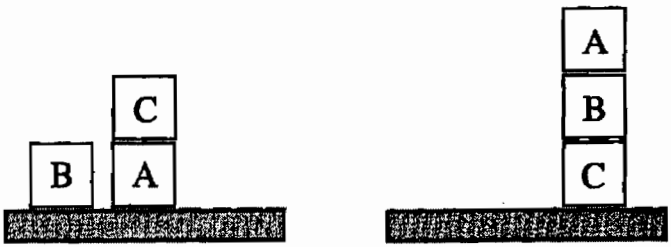
**؟۹.۱۱ اصول زیر را بری گراف برنامه‌ریزی اثبات کنید: الف) لفظی که در مرحلهٔ نهایی گراف بکار نمی‌رود، نمی‌تواند قابل دسترس باشد. ب) برای دسترسی به یک لفظ، هزینهٔ سطح آن لفظ در گراف سریال، بیشتر از هزینهٔ واقعی یک برنامهٔ بهینه نیست. *(این تمرین مشابه تمرین ۱۰.۱۰ در ویرایش سوم است)***

☑ حل: **الف)** الفاظ ماندگار هستند بنابراین اگر در مرحلهٔ نهایی بکار نروند آنگاه در هیچ جای دیگر نیز بکار نمی‌روند و در این صورت هرگز قابل‌دسترس نیستند. **ب)** در یک گراف برنامه‌ریزی سریال، در هر مرحلهٔ زمانی، فقط یک اقدام انجام می‌شود. هزینهٔ سطح (مرحله‌ای که آن لفظ اولین بار بکار رفته است) همان حداقل تعداد اقداماتی است که در یک برنامه صورت می‌گیرد که شاید در آن صورت به آن لفظ دست یابیم.

**؟۱۰.۱۱ ما برنامه‌ریزان جستجوی فضای حالت پیش‌رو و پس‌رو را با برنامه‌ریز نیمه‌مرتب مقایسه کرده و متوجه شدیم که مورد آخر یک جستجوگر فضای برنامه است. توضیح دهید که چگونه جستجوهای فضای حالت پیش‌رو و پس‌رو می‌توانند به عنوان مثال جستجوگرهای فضای برنامه نیز در نظر گرفته شوند و همچنین بگویید که عملگرهای پالایش برنامه چیستند؟ *(این تمرین مشابه تمرین ۱۳.۱۰ در ویرایش سوم است)***

☑ حل: یک برنامه‌ریز فضای حالت پیش‌رو، یک برنامه جزئی که همان دنباله خطی واکنش‌هاست را نگهداری می‌کند. عملگرپالایش برنامه باید یک واکنش قابل اجرا را به پایان دنباله بیافزاید. و بروزرسانی الفاظ مطابق با تأثیر واکنش انجام می‌شود. یک برنامه‌ریز فضای حالت پس‌رو، یک برنامهٔ جزئی که همان دنبالهٔ معکوس از واکنش‌هاست را نگهداری می‌کند. عملگر پالایش برنامه باید یک واکنش را به ابتدای دنباله به گونه‌ای بیافزاید که اندازهٔ تأثیر واکنش با اندازهٔ حالت آغازین دنباله سازگار باشد.

**؟۱۱.۱۱ شکل ۱۶.۱۱ مسالهٔ دنیای بلوک‌ها که مشهور به مساله «ساسمن بی‌قاعده» است، را نشان می‌دهد. این مساله از آن جهت بی‌قاعده نامیده می‌شود که هیچ یک از برنامه‌ریزان بدون جایگذاری در اوایل 1970 نتوانستند این مساله را حل کنند. تعریفی از این مساله به صورت نمادهای STRIPS ارائه داده و آن را یا با دست و یا با نرم‌افزاری برنامه‌ریز حل کنید. یک برنامه‌ریز بدون جایگذاری، برنامه‌ریزی است که اگر دو زیر هدف $G_1$ و $G_2$ را داشته باشد، می‌تواند برای $G_1$ یک برنامه متصل به برنامه $G_2$ تولید کند و یا برعکس. توضیح دهید که چرا یک برنامه‌ریز بدون جایگذاری نمی‌تواند این مساله را حل کند. *(این تمرین مشابه تمرین ۷.۱۰ در ویرایش سوم است)***




حالت شروع حالت هدف

شکل ۱۱.۱۶ مساله برنامه ریزی دنیای بلوکها «sussman anomaly»

☑ حل: حالت شروع عبارتست از:

$$On(B, Table) \wedge On(C, A) \wedge On(A, Table) \wedge Clear(B) \wedge Clear(C)$$

و هدف عبارتست از:

$$On(A, B) \wedge On(B, C)$$

در ابتدا توضیح می‌دهیم که چرا این مورد برای برنامه‌ریز بدون جایگذاری، یک بی‌قاعدگی محسوب می‌شود. دو زیرهدف وجود دارد: فرض کنید در ابتدا بر روی $On(A,B)$ کار می‌کنیم. می‌توانیم از $A$، $C$ را حذف کنیم سپس $A$ را به $B$ حرکت می‌دهیم. ولی با انجام این کار، دیگر هیچ راهی برای رسیدن به $On(B,C)$ نداریم. به طور مشابه اگر در ابتدا بر روی زیرهدف $On(B,C)$ کار کنیم باز هم پس از یک مرحله فورا به همین نتیجه می‌رسیم پس باید مجددا به همان حالت $A$ در $B$ برگردیم. اکنون نشان می‌دهیم که یک برنامه‌ریز بدون جایگذار نظیر POP چگونه کار می‌کند. از آنجا که $On(A,B)$ در حالت شروع صحیح نمی‌باشد، پس تنها یک راه داریم: برای برخی xها $Move(A,x,B)$. به طور مشابه، به مرحلهٔ $Move(B,x',C)$ برای برخی 'xها می‌رسیم. اکنون بیایید نظری به مرحلهٔ $Move(A,x,B)$ بیاندازیم. نیاز است تا پیش‌شرط آن یعنی $Clear(A)$ بدست آید. که این امر توسط $Move(b,A,y)$ و یا $MoveToTable(b,A)$ حاصل می‌شود. حال به مرحلهٔ بعدی نظری بیاندازیم. اگر $b$ را به $C$ بچسبانیم، آنگاه تمام پیش‌شرط‌های مرحلهٔ $MoveToTable(C,A)$ در حالت شروع، صحیح هستند. و می‌توانیم لینک‌هایی غیرمهم به آنها بیافزاییم. البته توجه داشته باشید که یک موضوع وجود دارد: مرحلهٔ Move$(B,x',C)$ از شرط $Clear(C)$ خبر می‌دهد که خود به مرحلهٔ $MoveToTable$ نیازمند است. این نکته را می‌توان با قرار دادن $Move(B,x',C)$ پس از مرحلهٔ $MoveToTable$ رفع نمود. در نهایت، دقت داشته باشید که تمام پیش‌شرط‌های $Move(B,x',C)$ در حالت شروع صحیح هستند. بنابراین یک برنامهٔ کامل داریم که تمام پیش‌شرط‌ها را ارضا می‌کند. لذا بایستی این سه مرحله به ترتیب طی شوند:

$MoveToTable(C, A)$
$Move(B, Table, C)$
$Move(A, Table, B)$

**؟ ۱۲.۱۱ مسأله پوشیدن کفش و جوراب که در بخش ۳.۱۱ تعریف شد را در نظر بگیرید. GRAPHPLAN را به این مسأله اعمال نموده و راه‌حل یافته شده را نشان دهید. اکنون اقداماتی برای پوشیدن یک کت و کلاه را نیز به آنها اضافه کنید. برنامه‌ای نیمه‌مرتب که یک راه‌حل هم باشد ارائه دهید. همچنین نشان دهید که 180 خطی‌سازی مختلف از این برنامه‌های نیمه‌مرتب وجود دارد. حداقل تعداد راه‌حل گراف برنامه‌ریزی مختلف برای بازنمایی تمام این 180 خطی‌سازی چیست؟**

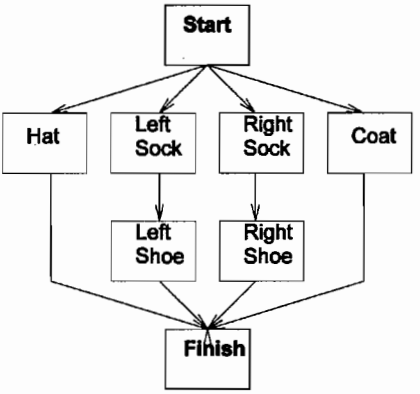
☑ حل: اقدامات موردنیاز عبارتند از:

$Action$(ACTION:$RightShoe$, PRECOND:$RightSockOn$, EFFECT:$RightShoeOn$)
$Action$(ACTION:$RightSock$, EFFECT:$RightSockOn$)
$Action$(ACTION:$LeftShoe$, PRECOND:$LeftSockOn$, EFFECT:$LeftShoeOn$)
$Action$(ACTION:$LeftSock$, EFFECT:$LeftSockOn$)

در یکی از راه‌حل‌هایی که توسط GRAPH PLAN یافته می‌شود، باید در گام اول $RightSock$ و $LeftSock$ اجرا شده و در گام دوم $RightShoe$ و $LeftShoe$. اکنون دو واکنش زیر را می‌افزاییم (هیچکدام از آنها پیش‌شرط ندارند):

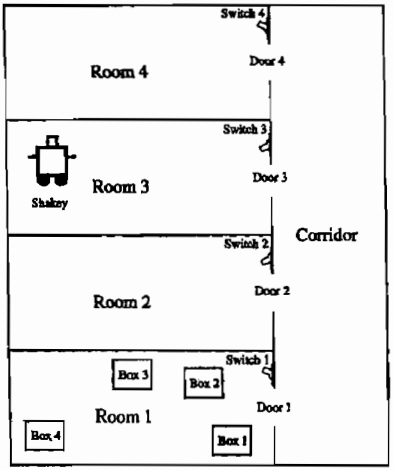
$Action$(ACTION:$Hat$, EFFECT:$HatOn$)
$Action$(ACTION:$Coat$, EFFECT:$CoatOn$)

برنامه نیمه‌مرتب را در شکل ۱.۱۱s می‌بینید. در کتاب خواندیم که 6 برنامه کاملا مرتب برای مساله جوراب/کفش وجود دارد و هر یک از این برنامه‌ها دارای 4 مرحله می‌باشند، پس 5 لینک داریم. در مرحلهٔ بعد *Hat* باید در یکی از این 5 موقعیت قرار گیرد که موجب می‌شود 6×5=30 برنامهٔ کاملا مرتب داشته باشیم که هر یک دارای 5 مرحله و 6 لینک هستند و در مرحلهٔ آخر *Coat* می‌تواند در هر یک از این 6 حالت قرار گیرد که منجر به ایجاد 180=30×6 برنامهٔ کاملا مرتب می‌شود.




شکل ۱۱.۱۱s برنامه‌ای جزئی شامل یک کت و کلاه.

**؟ ۱۳.۱۱ برنامه اصلی STRIPS برای کنترل روبات Shakey طراحی شد. شکل ۱۷.۱۱ نسخه‌ای از دنیای Shakey را نشان می‌دهد که شامل چهار درب بوده که با یک راهرو به هم متصل هستند. هر اتاق دارای یک درب و یک کلید لامپ می‌باشد. اقداماتی که در دنیای Shakey مسیر است شامل حرکت از یک مکان به مکان دیگر، هل دادن اشیای قابل حرکت (مانند جعبه)، بالا رفتن و پایین آمدن از روی اشیای سخت و محکم (مانند جعبه) و روشن و خاموش کردن لامپ با کلید آن می‌شود. روبات به تنهایی قادر به بالا رفتن از جعبه یا زدن کلید نبوده ولی برنامه‌ریز STRIPS که تواناتر از روبات بود می‌توانست برنامه‌ها را یافته و چاپ کند. شش اقدام Shakey عبارتند از: • *Go(x,y)*: یعنی Shakey در موقعیت $x$ بوده و به محل $y$ می‌رود که هم $x$ و هم $y$ در یک اتاق واقع هستند. طبق قرارداد، اگر دری بین دو اتاق وجود داشته باشد بایستی آنرا در هر دو اتاق مشاهده کنیم. • *Push(b,x,y)*: به معنای هل دادن جعبه $b$ از محل $x$ به محل $y$ در همان اتاق می‌باشد. در اینجا به مسند *Box* و ثابت‌هایی برای جعبه‌ها نیازمندیم. • *ClimbUp(b)*: به معنای بالا رفتن از جعبهٔ $b$ است. • *ClimbDown(b)*: به معنای پایین آمدن از جعبهٔ $b$ است، در اینجا نیز به مسند *On* و ثابت *Floor* نیازمندیم. • *TurnOn(s)*: به معنای روشن کردن لامپ $s$ با زدن کلید آن است. • *TurnOff(s)*: به معنای خاموش کردن لامپ $s$ با زدن کلید آن است.برای خاموش یا روشن کردن لامپ، روبات Sakey باید در همان موقعیت کلید، جعبه‌ای گذاشته و بر روی آن قرار بگیرد.**




شکل ۱۷.۱۱ دنیای Shakey.

این روبات می‌تواند درون یک اتاق مابین خطوط مشخص شده بر زمین حرکت کند، می‌تواند از در مابین دو اتاق گذر نماید، همچنین از اشیایی که قابل بالا رفتن هستند بالا رفته و اشیای قابل هل دادن را هل دهد. در ضمن قادر است دکمه چراغ را بزند.

**شش اقدام Shakey و حالت شروع شکل ۱۷.۱۱ را با نمادهای STRIPS توصیف کنید. برنامه‌ای طراحی کنید که روبات Shakey، جعبهٔ *Box2* را به اتاق *Room2* ببرد. *(این تمرین مشابه تمرین ۴.۱۰ در ویرایش سوم است)***

☑ حل: اقدامات این مساله بسیار شبیه به مساله میمون و موز است. اقدامات عبارتند از:

$Action(\text{ACTION}:Go(x,y), \text{PRECOND}:At(Shakey,x) \wedge In(x,r) \wedge In(y,r),$
$\text{EFFECT}:At(Shakey,y) \wedge \neg(At(Shakey,x)))$
$Action(\text{ACTION}:Push(b,x,y), \text{PRECOND}:At(Shakey,x) \wedge Pushable(b),$
$\text{EFFECT}:At(b,y) \wedge At(Shakey,y) \wedge \neg At(b,x) \wedge \neg At(Shakey,x))$
$Action(\text{ACTION}:ClimbUp(b), \text{PRECOND}:At(Shakey,x) \wedge At(b,x) \wedge Climbable(b),$
$\text{EFFECT}:On(Shakey,b) \wedge \neg On(Shakey,Floor))$
$Action(\text{ACTION}:ClimbDown(b), \text{PRECOND}:On(Shakey,b),$
$\text{EFFECT}:On(Shakey,Floor) \wedge \neg On(Shakey,b))$
$Action(\text{ACTION}:TurnOn(l), \text{PRECOND}:On(Shakey,b) \wedge At(Shakey,x) \wedge At(l,x),$
$\text{EFFECT}:TurnedOn(l))$
$Action(\text{ACTION}:TurnOff(l), \text{PRECOND}:On(Shakey,b) \wedge At(Shakey,x) \wedge At(l,x),$
$\text{EFFECT}:\neg TurnedOn(l))$

• حالت شروع عبارتست از:

$In(Switch_1,Room_1) \wedge In(Door_1,Room_1) \wedge In(Door_1,Corridor)$
$In(Switch_1,Room_2) \wedge In(Door_2,Room_2) \wedge In(Door_2,Corridor)$
$In(Switch_1,Room_3) \wedge In(Door_3,Room_3) \wedge In(Door_3,Corridor)$
$In(Switch_1,Room_4) \wedge In(Door_4,Room_4) \wedge In(Door_4,Corridor)$
$In(Shakey,Room_3) \wedge At(Shakey,X_S)$
$In(Box_1,Room_1) \wedge In(Box_2,Room_1) \wedge In(Box_3,Room_1) \wedge In(Box_4,Room_1)$
$Climbable(Box_1) \wedge Climbable(Box_2) \wedge Climbable(Box_3) \wedge Climbable(Box_4)$
$Pushable(Box_1) \wedge Pushable(Box_2) \wedge Pushable(Box_3) \wedge Pushable(Box_4)$
$At(Box_1,X_1) \wedge At(Box_2,X_2) \wedge At(Box_3,X_3) \wedge At(Box_4,X_4)$
$TurnwdOn(Switch_1) \wedge TurnedOn(Switch_4)$

• برنامهٔ دسترسی به هدف عبارتست از:

$Go(X_S,Door_3)$
$Go(Door_3,Door_1)$
$Go(Door_1,X_2)$
$Push(Box_2,X_2,Door_1)$
$Push(Box_2,Door_1,Door_2)$
$Push(Box_2,Door_2,Switch_2)$

**؟ ۱۴.۱۱ مشاهده کردیم که گراف‌های برنامه‌ریزی فقط می‌توانند واکنش‌های گزاره‌ای را پوشش دهند. اگر بخواهیم از گراف‌های برنامه‌ریزی برای یک مسأله با متغیرهایی در هدف، نظیر $At(P_1,x) \wedge At(P_2,x)$ که $x$ محدوده‌ای متناهی از موقعیت‌هاست، استفاده کنیم، چه می‌شود؟ چگونه می‌توانید یک مسأله را برای کار با گراف‌های برنامه‌ریزی رمزنگاری کنید؟ (راهنمایی: به واکنش Finish در برنامه‌ریزی POP مراجعه کنید. چه پیش‌شروطی باید داشته باشد؟)**

☑ حل: GRAPHPLAN یک الگوریتم گزاره‌ای است بنابراین همانطور که می‌توانیم FOL خاصی را با ترجمه به منطق گزاره‌ای، حل کنیم، در اینجا نیز می‌توانیم مسائل محاسبات موقعیت را با ترجمه به شکل گزاره‌ای، حل نماییم. و نکته آنجا است که چگونه این کار را به طور دقیق انجام دهیم. واکنش *Finish* در برنامه‌ریزی POP دارای همان پیش‌شروط حالت هدف است. می‌توانیم یک واکنش *Finish* برای GRAPHPLAN ایجاد کرده و به آن اثر *Done* را اعمال نماییم. در این مورد تعداد نمونه‌های واکنش *Finish* متناهی است و می‌توانیم با آنها استدلال نماییم.

**؟ ۱۵.۱۱ تاکنون فرض نمودیم که اقدامات فقط تحت موقعیت‌های خاص و مناسب اجرا می‌شوند. اکنون می‌خواهیم بدانیم چه قوانین حالت پسین گزاره‌ای نظیر معادله ۱.۱۱ باید در مورد اقدامات بیان شود، تا پیش‌شروط ارضا نشوند. الف) نشان دهید که این قانون پیش‌بینی می‌کند که در زمانیکه یک واکنش در حالتی اجرا شود که پیش‌شروط آن ارضا نشده باشند، هیچ اتفاقی نمی‌افتد. ب) یک برنامهٔ $p$ را به طوری در نظر بگیرید که شامل اقداماتی جهت رسیدن به یک هدف به همراه اقداماتی غیرمجاز باشد. آیا این مورد به صورت زیر است:**

$$initial\ state \wedge successor\text{-}state\ axioms \wedge p \models goal?$$

**ج) توسط قوانین حالت پسین مرتبهٔ اول در محاسبات موقعیت، آیا ممکن است اثبات شود که یک برنامه با اقدامات غیرمجاز به یک هدف دست یابد؟** *(این تمرین مشابه تمرین ۱۴.۱۰ در ویرایش سوم است)*

☑ حل: **الف)** نکته این تمرین در آن است که برنامه‌ریز باید در نظر گیرد که درصورت وقوع یک واکنش غیرممکن، چه اتفاقی رخ دهد (مانند پرواز هواپیما از مبدای که درآن قرار ندارد). به عنوان مثال، فرض کنید $P_1$ در $JFK$ باشد و بخواهیم با کمک آن از Bangalore به Brisbane پرواز کنیم (توسط ۱.۱۱). ازآنجا که $P_1$ در $JFK$ بوده و نمی‌تواند این پرواز را انجام دهد، پس همچنان در محل خود باقی می‌ماند. **ب)** بله. هواپیما به کار خود ادامه می‌دهد. زیرا این جریان قبل از وقوع واکنش غیرقابل‌اجرا تا حالت بعدی برقرار است. پس هواپیما می‌تواند از آن حالت کارخود را ادامه دهد. **ج)** این موضوع بستگی به چگونه نوشتن اصول و قوانین دارد. اصول ما به شکل $Action\ is\ possible \Rightarrow Rule$ هستند که هیچ اطلاعاتی در مورد غیرممکن بودن واکنش‌ها در اختیار ما نمی‌گذارد. بنابراین بایستی یا اصول را تدوین مجدد نماییم و یا مواردی اضافه کنیم که مشخص شود در صورت غیرممکن بودن واکنش، چه اتفاقی رخ دهد.

**؟ ۱۶.۱۱ مثال‌هایی از دامنهٔ هواپیمایی ارائه داده و توضیح دهید چگونه جداسازی نمادها موجب کاهش اندازهٔ قوانین پیش‌شرط و محرومیت واکنش‌ها در قوانین می‌شود. یک فرمول کلی برای اندازهٔ هر مجموعه قانون بر حسب تعداد مرحلهٔ زمانی، تعداد طرحواره واکنش، آریتی آنها و تعداد اشیا ارائه دهید.**

☑ حل: یک قانون پیش‌شرط به شکل زیر است:

$$Fly(P_1, JFK, SFO)^0 \Rightarrow At(P_1, JFK)^0$$

تعداد چنین قوانینی $O(T\times|P|\times|A|^2)$ است که $T$ همان تعداد مراحل زمانی، $|P|$ تعداد پرواز و $|A|$ تعداد فرودگاه‌هاست. در حالت کلی، اگر $n$ طرحواره واکنش با حداکثر $k$ آریتی و $|O|$ شی وجود داشته باشد آنگاه تعداد $O(n\times T\times|O|^k)$ قانون وجود خواهد داشت. توسط جداسازی نمادها، نیازی به توصیف هر پرواز خاص نداریم و تنها نیاز است بگوییم که برای یک پرواز به هر مکانی باید در آغاز در فرودگاه باشد. داریم:

$$Fly_1(P_1)^0 \wedge Fly_2(JFK)^0 \Rightarrow At(P_1, JFK)^0$$

در حالت کلی اگر $n$ طرحواره واکنش با حداکثر $k$ آریتی و $|O|$ شی داشته باشیم، و همینطور هر قانون پیش‌شرطی فقط به دو آرگومان نیازمند باشد آنگاه تعداد قوانین $O(n\times T\times|O|^2)$ خواهد شد که در حالت سریع به $O(|O|^{k-2})$ ختم می‌شود.
یک قانون محرومیت واکنش به شکل زیر است:

$$\neg(Fly(P_2, JFK, SFO)^0 \wedge Fly(P_2, JFK, LAX)^0)$$

طبق نمادگزاری که در قبل گفته شد، تعداد قوانین $Fly$ عبارتست از $O(T\times|P|\times|A|^3)$ که در حالت کلی به $O(n\times T\times|O|^{2k})$ منجر می‌گردد. با جداسازی نمادها، به هیچ قانونی برای $Fly$ دست نمی‌یابیم ولی در حالتی که متغیر دیگری باشد که به محرومیت مرتبط نباشد، آنگاه قانونی بدست می‌آید.

**؟ ۱۷.۱۱ در الگوریتم SATPLAN (شکل ۱۵.۱۱) هر فراخوانی الگوریتم ارضاپذیری به یک هدف $g^T$ منجر می‌شود که T دارای محدوده‌ای است که از 0 تا $T_{max}$ است. فرض کنید که به جای اینکار الگوریتم ارضاپذیری فقط یکبار و با هدف**

$$g^0 \vee g^1 \vee \ldots \vee g^{T_{max}}$$

**فراخوانی شود. الف) اگر طولی کمتر و یا مساوی با $T_{max}$ وجود داشته باشد، آیا این کار همچنان یک برنامه را تولید می‌کند؟ ب) آیا این راه‌کار، هر راه‌حل جدید جعلی را تولید می‌کند؟ ج) بحث کنید که چطور می‌توان یک الگوریتم ارضاپذیری را نظیر WALKSAT تغییر داد تا با داشتن یک هدف فصلی به این شکل، بتواند راه‌حلی کوتاه بیابد(در صورت وجود). *(این تمرین مشابه تمرین ۱۰ ۱۶ در ویرایش سوم است)***

☑ حل: **الف)** بله. این کار در هر زمان SATPLAN نرمال یک برنامه با طول حداقل $T_{max}$ بیابد، خواهد توانست یک برنامه بیابد. **ب)** خیر. این کار باعث نمی‌شود که SATPLAN یک راه‌حل نادرست برگرداند. ولی ممکن است در برنامه‌ریزی آن این اثر وارد شود. به عنوان مثال دست‌یابی و یا عدم دست‌یابی به هدف برای چندین مرتبه. **ج)** نمی‌توان راه‌حل ساده و روشن برای آنکه WALKSAT به راه‌حلی کوتاه برسد ارائه داد. زیرا طول یک برنامه را نمی‌دانیم- در واقع مسأله آنجاست که یک مساله برنامه‌ریزی، قسمتی از رمزنگاری است و نه قسمتی از WALKSAT. ولی با اعمال یک استراتژی تقریبی بر روی WALKSAT می‌توانیم به راه‌حلی کوتاهتر توسط روش زیر دست یابیم. برای هر اقدام متغیری گرفته و به هر متغیر اقدام به طور تصادفی مقدار False را برای شروع انتساب می‌دهیم. (به خصوص آنهایی که جنب آن هستند) و به طور تصادفی یک متغیر واکنش را نسبت به متغیر کناری‌اش معکوس می‌کنیم.

# فصل ۱۲ (ویرایش دوم)

**؟ ۱.۱۲ بازنمایی زمان و منبع در بخش ۱.۱۲ را به دقت بررسی کنید: الف) چرا بهتر است که *Duration(d)* را به عنوان اثر یک واکنش در نظر بگیریم تا آنکه یک فیلد مجزا در واکنش به شکل *Duration:d*؟ ب) چرا *Resource:m* به جای آنکه یک اثر باشد، یک فیلد مجزا در واکنش است؟**

☑ حل: **الف)** *Duration(d)* ویژگی‌های لازم برای اثر را داراست و اثر این واکنش، همان حرکت در جهت عقربه‌های ساعت به اندازهٔ d می‌باشد. امکان دارد که duration به نتیجهٔ حاصل از واکنش بستگی داشته باشد بنابراین اگر از اثرات شرطی یا فصلی استفاده شود، آنگاه باید روشی وجود داشته باشد تا durationها را به نتایج مرتبط کند. که اینکار به سادگی و با قراردادن durationها در عبارات نتیجه بدست می‌آید. **ب)** مدل STRIPS فرض می‌کند که واکنش‌ها بر حسب نقاط زمانی بوده و فقط توسط پیش‌شروط و اثراتشان مشخص می‌گردند. حتی اگر یک واکنش، منبعی را در اختیار گیرد که بر روی حالت نتیجه (خروجی) تأثیر نگذارد. بنابراین بایستی فرمول STRIPS را گسترش دهیم. اینکار را با RESOURCE انجام می‌دهیم. هر اثر با اثر دیگر متفاوت است و میزان تفاوت آنها به قدری زیاد است که باعث می‌شود نسبت به هر یک از آنها احساس جداگانه‌ای داشته باشیم و یک فیلد مجزا در واکنش محسوب شود.

**؟ ۲.۱۲ یک منبع مصرفی، به منبعی اتلاق می‌شود که توسط یک واکنش استفاده شده است (یا قسمتی از آن). به عنوان مثال: اتصال موتور به ماشین نیازمند پیچ است و پیچی که در یک جا استفاده شود نمی‌تواند برای سایر اتصالات بکار رود. الف) توضیح دهید که چطور می‌توانیم بازنمایی شکل ۳.۱۲ را تغییر دهیم تا در آغاز 100 عدد پیچ داشته باشیم که موتور $E_1$ به 40 عدد پیچ و موتور $E_2$ به 50 عدد پیچ نیاز داشته باشد. می‌توانید از نمادهای تابعی + و – در الفاظ تأثیر برای منابع استفاده کنید. ب) توضیح دهید که چگونه باید تعریف برخورد بین لینک‌های سببی و واکنش در برنامه‌ریزی نیمه‌مرتب تغییر داده شود تا منابع مصرفی را نیز شامل شود. ج) برخی واکنش‌ها (نظیر افزایش خدمات پشتیبانی کارخانه با پیچ و یا پرکردن سوخت ماشین) می‌توانند قابلیت دسترسی منابع را ارتقا دهند. اگر منبعی داشته باشیم که هیچ واکنشی آن را ارتقا ندهد به طور یکنواخت حرکتی غیرصعودی خواهد داشت، توضیح دهید که چگونه می‌توان از این ویژگی جهت هرس فضای جستجو استفاده نمود؟**

☑ حل: ایده‌آل اصلی آن است که سطح اولیه منبع جزء پیش‌شروط و هر تغییر در سطح منبع را جزء اثر هر واکنش ثبت کنیم. **الف)** فرض کنید *Screws(s)* نشان‌دهندهٔ وجود $s$ پیچ باشد. برای حالت شروع بایستی *Screws*(100) را به حالت شروع بیافزاییم و همچنین چهار آرگومان را به مسند *Engine* برای تعداد پیچ‌های موردنیاز اضافه کنیم. به عنوان مثال *Engine*($E_1$,$C_1$,30,40) و *Engine*($E_2$,$C_2$,60,50) . سپس عبارت *Screws*($s_0$) را به پیش‌شرط *AddEngine* و نیز $s$ را به عنوان آرگومان چهارم لفظ *Engine* اضافه می‌کنیم. در ادامه عبارت *Screwe*($s_0$-$s$) را به اثر *AddEngine* می‌افزاییم. **ب)** یک راه‌حل ساده آن است که بگوییم هر واکنشی که منبعی را مصرف کند احتمال دارد که با هر لینک سببی که از آن منبع محافظت می‌کند، برخورد داشته باشد. **ج)** برنامه‌ریز می‌تواند سیر نیازمندی‌های منابع برای واکنش‌هایی که اضافه می‌شوند را نگهداری کند و هر زمان که تعداد کل استفاده‌ها از مقدار اولیه بیشتر شد به عقب برگردد.

**؟ ۳.۱۲ مراحل *HireBuilder* و *GetPermit* در شکل ۷.۱۲ را تجزیه کرده و نشان دهید که یک زیربرنامهٔ تجزیه شده چگونه می‌تواند به برنامه کلی متصل شود.**

☑ حل: برای این سئوال می‌توان پاسخ‌های زیادی ارائه داد. نکته مهم آن است که بدانیم پیاده‌سازی صحیح یک واکنش از چه چیزهایی باید تشکیل شود. باید یک برنامه سازگار باشد که تمام پیش‌شروط و اثرات را شامل شود. لذا اولین چیزی که نیاز داریم تصمیم‌گیری در مورد پیش‌شروط و اثرات واکنش‌های سطح بالاست. برای *GetPermit* فرض کنید پیش‌شرط، «مالک بودن زمین» بوده و «داشتن مجوز برای آن قطعه از زمین» نیز اثر باشد. برای *HireBuilder* «داشتن قابلیت پرداخت» به عنوان پیش‌شرط و «داشتن یک قرارداد امضا شده» اثر تلقی می‌شود. یکی از تجزیه‌های ممکن برای *GetPermit* به صورت دنباله‌ای از سه مرحله است: *GetPermitForm*، *FillouForm*، *GetForrmApproved*. یک لینک سببی بین شرط *HaveForm* با دو مورد اول و یک لینک بین شرط *HaveCompletedForm* و دو مورد آخر وجود دارد. در نهایت مرحله GetFormApproved دارای اثر HavePermit است. این تجزیه معتبر است. برای *HireBuilder* دنباله‌ای شامل سه مرحله زیر را در نظر بگیرید:

*SignContract* , *ChooseBuilder*, *InterviewBuilder*

مرحله آخر شامل پیش‌شرط *AbleToPay* و اثر *HaveContractInHand* میباشد. همچنین لینک‌های سببی بین زیر مراحل وجود دارد ولی آنها هیچ تاثیری در صحت تجزیه نخواهد گذاشت.

**؟ ۴.۱۲ مثالی از دامنه ساخت خانه ارائه دهید که در آن دو زیر مرحله انتزاعی وجود داشته باشد که نتوان آنها را بدون به اشتراک‌گذاری مراحل در یک برنامه سازگار ادغام نمود. (راهنمایی: مکان‌هایی از خانه که به طور فیزیکی مجاور هستند، همان مکان‌هایی هستندکه دو زیربرنامه با هم در تبادل‌اند)**

☑ حل: مساله ساخت دو دیوار مجاور از خانه را در نظر بگیرید. اگر فرض کنیم کار هر زیربرنامه ساخت یک دیوار باشد، آنگاه بیشتر زیربرنامه‌های اینکار مستقل هستند ولی این زیربرنامه‌ها باید مرحله «ساخت گوشه دیوار» را بین خود به اشتراک بگذارند. اگر این مرحله به اشتراک گذاشته نشود آنگاه در انتها دو دیوار مجزا و غیرمتصل خواهیم داشت که بعلت عدم گذاشتن آجرها در لابه‌لای هم، مستحکم نخواهد بود. دقت داشته باشید که تجزیه یک برنامه به تعدادی زیربرنامه، باید به گونه‌ای باشد که تعداد مراحل اشتراک‌گذاری بین زیربرنامه‌ها کمینه شود. به عنوان مثال یک تجزیه در ساخت خانه می‌تواند به صورت زیربرنامه‌هایی نظیر «ساخت دیوار» و «ساخت سقف» تقسیم‌بندی شود ولی معماران واقعی اینکار را در عمل انجام نمی‌دهند و به جای آن مراحلی چون «ساخت دیوارهای اصلی»، «ساخت سقف‌های اصلی» و سپس مرحله «تکمیل» را سپری می‌کنند.

**؟ ۵.۱۲ برخی افراد می‌گویند یکی از مزایای برنامه‌ریزی HTN آن است که می‌تواند مسائلی چون «رفتن از لس‌آنجلس به نیویورک و برگشت» را حل کند که برای ملیت‌های غیر HTN دشوار است: زیرا حالات شروع و پایان هر دو یکسان و به صورت $(AT(LA))$ هستند. آیا می‌توانید روشی برای بازنمایی یافته و این مساله را بدون HTN حل کنید؟**

☑ حل: از دیدگاه HTN فضای تجزیه ممکن است راه‌حل‌های مجازی را تحمیل کند و برخی دنباله‌های ممکن از واکنش‌های اصلی را حذف نماید. به عنوان مثال تجزیه واکنش *LAToNY RoundTrip* قید می‌کند که کارگزار باید به نیویورک برود. اگر حالت شروع و حالت هدف را به صورت یکسان تعریف کنیم (یعنی بودن در شهر لس آنجلس)، آنگاه یک برنامه تهی یکی از راه‌حل‌ها خواهد بود. ولی بهتر است که حالت هدف را مجددا و به صورت دیگری تعریف کنیم. حالت هدف به صورت $(At(LA))$ نبوده و به صورت زیر است: $At(LA) \wedge Visited(NY)$. یعنی هدف مساله آن است که در لس‌آنجلس باشیم و نیویورک را نیز ملاقات کرده باشیم که رسیدن به این حالت مستلزم مسافرت است. مشاهده شد که با این تعریف، بین حالت شروع و هدف تمایز قائل شدیم. می‌توانیم $Visited(y)$ را به عنوان اثر $Fly(x,y)$ بیافزاییم. بنابراین راه‌حل، همان مسافرتی است که شامل نیویورک می‌شود و بایستی از توقف برنامه STRIPS در سایر مناطق سفر جلوگیری کرد که کاری دشوار است زیرا اهداف خنثی (نقیض) غیرمجاز هستند.

**؟ ۶.۱۲ نشان دهید که چطور می‌توان یک توصیف واکنش STRIPS استاندارد را به عنوان یک تجزیه HTN بازنویسی نمود. برای اینکار از عبارت $Achieve(p)$ برای نمایش واکنش بدست آوردن شرط $p$ استفاده کنید.**

☑ حل: فرض کنید برای $a$، یک توصیف واکنش STRIPS با پیش‌شرط $p$ و اثر $q$ داشته باشیم. واکنشی که باید تجزیه شود $Achieve(q)$ است که تجزیه آن شامل دو مرحله می‌شود: $Achieve(p)$ و $a$ . می‌توان راهی ساده و روشن جهت ترکیب اثرات و پیش‌شروط بکار برد.

**؟ ۷.۱۲ برخی عملگرهای موجود در زبان برنامه‌نویسی استاندارد، می‌توانند به عنوان واکنش‌هایی جهت تغییر حالت دنیا مدل شوند. به عنوان مثال عملگر انتساب در برنامه‌نویسی، محتوای یک محل از حافظه را تغییر داده و عملگر print(چاپ) حالت رشته خروجی را تغییر می‌دهد. یک برنامه‌نویسی شامل چنین عملگرهایی می‌تواند به عنوان برنامه‌ای در نظر گرفته شود که هدفش توسط آن برنامه‌نویسی مشخص می‌شود. بنابراین الگوریتم‌های برنامه‌نویسی می‌توانند برای ساخت برنامه‌هایی با هدف مشخص، استفاده شوند. الف) یک طرح‌واره واکنش برای عملگر انتساب (انتساب یک مقدار از یک متغیر به متغیر دیگر) بنویسید. یادآوری می‌شود که در این حالت باید روی مقدار اصلی، مقداری ریخته شود. ب) نشان دهید که تولید اشیاء چگونه می‌تواند توسط یک برنامه‌ریز بکار رود تا یک برنامه برای جابجایی یک مقدار بین دو متغیر با استفاده از یک متغیر کمکی تولید شود. (این تمرین مشابه ۷.۱۱ در ویرایش سوم این کتاب است)**

☑ حل: به یک واکنش *Assign* نیازمندیم که یک مقدار $d$ را از رجیستر منبع $sr$ به رجیستر مقصد $dr$ انتساب دهد. (می‌توانید به جای اصطلاح رجیستر از اصطلاح متغیر استفاده کنید ولی استفاده از رجیستر روشن می‌سازد که ما از محل‌های فیزیکی استفاده می‌کنیم):

$$Action(\text{ACTION}:Assign(dr, sr),$$
$$\text{PRECOND}:Register(dr) \wedge Register(sr) \wedge Value(dr, dv) \wedge Value(sr, sv),$$
$$\text{EFFECT}:Value(dr, sv) \wedge \neg Value(dr, dv))$$

اکنون فرض کنید از یک حالت شروع به صورت زیر آغاز می‌کنیم:

$$Register(R_1) \wedge Register(R_2) \wedge Value(R_1, V_1) \wedge Value(R_2, V_2)$$

و هدف ما عبارتست از:

$$Value(R_1, V_2) \wedge Value(R_2, V_1)$$

متاسفانه برای حل مساله به این صورت، هیچ راهی نداریم. باید شرط $Register(R_3)$ را به حالت شروع بیافزاییم و یا آنکه روشی ارائه دهیم که یک امکان ایجاد رجیستر جدید فراهم شود. برای حل این مشکل، واکنشی تعریف می‌کنیم که رجیستری جدید تولید نماید:

$$Action(\text{ACTION}:Allocate(r),$$
$$\text{EFFECT}:Register(r))$$

بنابراین دنباله مراحل زیر یک برنامهٔ معتبر را ارائه می‌دهد:

$Allocate(R_3)$
$Assign(R_3, R_1)$
$Assign(R_1, R_2)$
$Assign(R_2, R_1)$

**؟ ۸.۱۲ عبارت زیر را در نظر بگیرید: در چارچوبی که حالت‌های اولیه نامشخص مجاز است، «اثرات فصلی» فقط تسهیلاتی برای نمادگذاری داشته و قدرت اضافه‌ای در بازنمایی ندارند. «برای هر طرح واکنش a با اثر فصلی P∨Q زمانی می‌توانیم آن را با اثرات شرطی جایگزین کنیم که R:Q∧R:P¬ـ که می‌توانند به دو واکنش قاعده‌مند ساده شوند. گزاره R گزاره‌ای تصادفی است که در حالت اولیه ناشناخته بوده و هیچ واکنش ادراکی برای آن وجود ندارد». آیا عبارت فوق صحیح است ؟ دو مورد جداگانه را در نظر بگیرید یکی در صورتی که فقط یک نمونه از طرح واکنش a در برنامه باشد و دیگری آنکه بیش از یک نمونه در برنامه باشد.**

☑ حل: برای مورد اول یعنی حالتی که یک نمونه از طرح واکنش $a$ در برنامه موجود باشد، تدوین صحیح است. بدان صورت که راه‌حلی که برای تدوین فصلی اصلی داریم، همان راه‌حل برای تدوین جدید خواهد بود و برعکس. برای مورد دوم: یعنی حالتی که بیش از یک نمونه از طرح واکنش احتمال وقوع دارد، تدوین نادرست است. زیرا فرض می‌کند که خروجی نمونه‌ها توسط یک متغیر مخفی منفرد معین می‌شود. بنابراین اگر به عنوان مثال $P$، خروجی یک نمونه باشد باید خروجی نمونه دیگر نیز باشد. لذا ممکن است که تدوین مجدد در تدوین اصلی دچار شکست شود.

**؟ ۹.۱۲ چرا برنامه‌ریزی شرطی نمی‌تواند با غیرقطعیت نامحدود کار کند؟**

☑ حل: غیرقطعیت‌نامحدود بدان معناست که یا مجموعه اثرات ممکن برای هر واکنش ناشناخته باشد و یا آنکه شمارش آنها بسیار زیاد باشد. از این رو فضای دنباله واکنش، نیاز دارد تا تمام این امکانات را در نظر گیرد که بسیار زیاد خواهد شد.

**؟ ۱۰.۱۲ در دنیای بلوک‌ها دو واکنش STRIPS بنام‌های *Move* و *MoveToTable* را جهت برقرای مسند *Clear* تعریف می‌کنیم. نشان دهید که چگونه باید از اثرات شرطی استفاده کنیم تا بتوان هر دوی این موارد را توسط یک واکنش بازنمایی نمود. *(این تمرین مشابه تمرین ۱۱.۹ در ویرایش سوم است)***

☑ حل: طبق تعریف دومی که از $Clear$ در این فصل بیان گشت، (یک مربع خالی برای یک بلوک وجود داشته باشد)، تنها تغییر موردنیاز آن است که اگر یک جدول داریم مقصد، خالی باقی بماند:

$Action(Move(b, x, y),$
PRECOND:$On(b, x) \wedge Clear(b) \wedge Clear(y)$,
EFFECT:$On(b, y) \wedge Clear(x) \wedge \neg On(b, x) \wedge$ (**when** $y \neq Table$: $\neg Clear(y)$))

**؟ ۱۱.۱۲ اثر شرطی که برای واکنش Suck در دنیای جاروبرقی گفته شد آن است که خانه‌های محیط بسته به مکانی که روبات در آن قرار دارد، تمیز می‌شوند. آیا شما می‌توانید یک مجموعهٔ جدید از متغیرهای گزاره‌ای برای تعریف حالات دنیای جاروبرقی نظیر Suck که یک توصیف غیرشرطی دارد، بیابید؟ با استفاده از گزارهٔ خود توصیفاتی برای Suck، Left، Right ارائه داده و نشان دهید که این گزاره‌ها برای توصیفات تمام حالات ممکن در دنیا کافی هستند. *(این تمرین مشابه تمرین ۱۱.۱۰ در ویرایش سوم است)***

☑ حل: عبارت $CleanH$ درست است اگر و فقط اگر مکان فعلی روبات تمیز باشد و $CleanO$ درست است اگر و فقط اگر خانه دیگر محیط تمیز باشد. بنابراین $Suck$ عبارتست از:

$Action(Suck,$ PRECOND:, EFFECT:$CleanH)$

تاسفانه حرکت کردن بر روی این الفاظ جدید تأثیر می‌گذارد. برای $Left$ داریم:

$Action(Left,$ PRECOND:$AtR,$
EFFECT:$AtL \wedge \neg AtR \wedge$ **when** $CleanH$: $CleanO \wedge$ **when** $CleanO$: $CleanH$
$\wedge$ **when** $\neg CleanO$: $\neg CleanH \wedge$ **when** $\neg CleanH$: $\neg CleanO)$

برای $Right$ نیز همین است.

**؟ ۱۲.۱۲ توصیف کاملی از Suck برای جاروبرقی مورفی دوگانه ارائه دهید که در گاهی اوقات که به یک مربع مقصد تمیز می‌رسد آن را کثیف می‌کند وگاهی مواقع با انجام عمل Suck در یک مربع تمیز، آن را کثیف می‌کند.**

☑ حل: در اینجا از تعریف $Left$ که قبلا گفته شده استفاده می‌کنیم:

$Action(Suck,$ PRECOND:,
EFFECT:(**when** $AtL$: $CleanL \vee$ (**when** $CleanL$: $\neg CleanL$))
$\wedge$ (**when** $AtR$: $CleanR \vee$ (**when** $CleanR$: $\neg CleanR$)))

**؟ ۱۳.۱۲ فرشی کثیف که روی آن مانعی قرار نداشته باشد، را جاروبرقی بکشید. مسیر پیموده شده توسط تمیزکننده جاروبرقی را با دقت ممکن ترسیم کنید. این مسیر را با توجه به برنامه‌ریزی‌های گفته شده در این فصل توضیح دهید. *(این تمرین مشابه تمرین ۱۱.۱۱ در ویرایش سوم است)***

☑ حل: نکته مهم آن است که تمیزکننده جاروبرقی حرکت خود را در مکان‌های کثیف، چندین بار را تکرار می‌کند تا آن مکان کاملا تمیز شود. همچنین حرکت‌های به سمت جلو معمولا کوتاه و سریع بوده و جهت رسیدن به ابتدای منطقه صورت می‌گیرد. این موضوع بر حسب خروجی‌های فصلی عبارتست از: هر منطقه ممکن است کاملا تمیز باشد و یا خیر؛ حرکت برگشتی کارگزار را قادر به بررسی می‌کند؛ و تکرار کردن کارگزار را از تکمیل کار خود مطمئن سازد (البته اگر کثیفی دیرینه بوده و مربوط به نخ قالی شود آنگاه با تکرار جاروکشیدن نیز نمی‌توان آن را تمیز کرد). بنابراین یک برنامه حلقوی قوی با واکنش‌های حسی داریم.

**؟ ۱۴.۱۲ عبارات زیر معمولا در پشت قوطی‌های شامپو درج می‌شود و به هر کدام از آنها، اصطلاحات «مشروط»، «غیرمشروط» و یا «حاوی برنامه» گفته می‌شود. الف)«کف مالی کردن، شستشو و تکرار» ب) « شامپو را به مو و پوست سر مالیده، سپس برای چند دقیقه صبر کرده و آن را بشویید. در صورت نیاز اینکار را تکرار کنید» ج) «در صورت بروز مشکل با پزشک مشورت کنید»**

☑ حل: **الف)** این مورد یک «برنامه غیرمشروط» است که تحت اللفظی شامل یک حلقه بی‌نهایت می‌شود. اگر بدانیم که شامپو فقط بر روی موهای کثیف، کف می‌کند (اگر پیش‌شرط *Lather* عبارت $\neg Clean$ باشد) و هدف ما از استحمام تمیز شدن باشد (هدف به صورت *Clean* تعریف شود)، آنگاه با پیگیری این برنامه غیرمشروط می‌توانیم به جایی برسیم که تمیزی حاصل شود. **ب)** این مورد یک «برنامه شرطی» را نشان میدهد و این موضوع را می‌توان از عبارت «در صورت نیاز» متوجه شد. منظور از عبارت «درصورت نیاز» آن است که اگر موهایتان تمیز نشد($\neg Clean$)، آنگاه باید عملیات را تکرار کنید. **ج)** این مورد نیز یک «برنامه شرطی» است زیرا اگر پس از استفاده از شامپو با مشکل مواجه شدیم، آنگاه باید به پزشک مراجعه نماییم. البته دراینجا به طور واضح گفته نشده است که منظور از مشکلاتی که باید بررسی شوند، چیست.

**؟ ۱۵.۱۲ الگوریتم AND-OR-GRAPH-SEARCH در شکل ۱۰.۱۲ حالات تکراری را فقط در مسیر ریشه تا حالت جاری بررسی می‌کند. فرض کنید که این الگوریتم طوری تغییر یافته که هر حالت ملاقات شده را ذخیره کرده و می‌تواند مجددا لیست آنها را بررسی نماید. (به عنوان مثال (GRAPH-SEARCH)) در شکل ۳.۱۹ را ببینید. اطلاعاتی که باید ذخیره شوند را مشخص کرده و تعیین کنید که این الگوریتم در زمان یافتن یک حالت تکراری چگونه از این اطلاعات باید استفاده کند؟ (راهنمایی: حداقل بایستی بین حالاتی که یک زیربرنامه موفق قبلا تولید میکرده و حالاتی که توسط هیچ زیربرنامه‌ای یافته نمی‌شود تمایز قائل شوید.) توضیح دهید که چگونه از برچسب‌گذاری استفاده کنیم که از داشتن چندین کپی از زیربرنامه‌ها جلوگیری شود.**

☑ حل: پاسخ برعهده خواننده.

**؟ ۱۶.۱۲ به طور دقیق توضیح دهید چگونه الگوریتم AND-OR-GRAPHSEARCH را تغییر دهیم که در صورتی که هیچ برنامه غیرچرخشی در آن وجود نداشته باشد، بتواند یک برنامه چرخشی تولید کند. شما به سه مورد نیاز دارید: برچسب‌گذاری مراحل برنامه تا یک برنامه چرخشی بتواند به آخرین بخش از برنامه بازگشت کند؛ تغییر الگوریتم OR-SEARCH به طوری که پس از یافتن یک برنامه چرخشی همچنان برای یافتن برنامه غیرچرخشی جستجو کند؛ و تکمیل بازنمایی برنامه تا بتواند نشان دهد که یک برنامه چرخشی است. نشان دهید که الگوریتم شما در موارد زیر چگونه کار می‌کند. الف) دنیای جاروبرقی مورفی سه‌گانه. ب) نوعی دیگر دنیای جارو برقی مورفی دوگانه. در صوت تمایل می‌توانید از یک پیاده‌سازی رایانه‌ای جهت بررسی نتایج خود استفاده نمایید آیا می‌توانید برنامه‌ای برای قسمت ب با استفاده از نحو استاندارد حلقه‌ها بنویسید؟**

☑ حل: پاسخ برعهده خواننده.

**؟ ۱۷.۱۲ یک زیر روال بروزرسانی حالات باور برای محیط‌های نیمه‌مشاهده‌پذیر را به طور کامل مشخص کنید که در آن برای محاسبه بازنمایی جدید حالت باور (لیستی از گزاره‌های دانشی)، از بازنمایی فعلی حالت باور استفاده کرده و برای یک توصیف واکنش از اثرات شرطی بهره ببرد.**

☑ حل: درا بتدا باید تصمیم بگیریم که اگر پیش‌شروط ارضا شوند چه باید کنیم. سه مورد زیر وجود خواهد داشت: **الف)** اگر بدانیم که ارضاپذیر نیست، آنگاه حالت باور جدید دقیقا همان حالت قبلی خواهد بود (به عبارت دیگر اتفاقی رخ نمیدهد). **ب)** اگر بدانیم که ارضا می‌شود، آنگاه اثرات غیرشرطی (تمام گزاره‌های دانشی) افزوده شده و از حالت باور در شکل معمول STRIPS حذف می‌شوند. هر اثر شرطی که شرط آن شناخته شده باشد، به همان روش اعمال می‌شود. برای هر شرط ناشناخته یک حالت باور به همراه افزودنی‌ها و حذفیات مناسب می‌سازیم. **ج)** اگر وضعیت پیش‌شروط ناشناخته باشد، آنگاه هر حالت باور جدید عبارتست از ترکیب فصلی حالت باور قسمت الف بدون تغییر و یکی از حالات باور بدست آمده از قسمت ب. برای بازنمایی «لیست گزاره‌های دانشی» باید گزاره‌هایی را نگهداریم که در هر یک از حالات باوری که دورانداخته شده و یا بی‌تکلیف هستند مشخص می‌شود. این کار در حالات باور ضعیف‌تر جواب می‌دهد تا آنکه ترکیبات فصلی را نگهداری کنیم. به عبارت دیگر نگهداری ترکیبات فصلی در مراحل زیاد منجر به وجود فضای بازنمایی بسیار زیاد (نمایی) خواهد شد.

**؟ ۱۸.۱۲ طبق معادله ۲.۱۲ برای هر یک از واکنش‌های *Right* و *Suck* توصیف واکنش بنویسید. همچنین مطابق معادله ۳.۱۲ توصیفی نیز برای *CheckLocation* بیان کنید. سپس با استفاده از مجموعه دوم گزاره‌ها در تمرین ۱۱.۱۲ کار خود را تکرار کنید.**

☑ حل: برای واکنش *Right* به طور مشخص نسخه‌ای از معادله ۲.۱۲ بکار می‌آید:

$$Action(Right, \text{PRECOND}: AtL,$$
$$\text{EFFECT}: K(AtR) \wedge \neg K(AtL) \wedge \textbf{when } CleanL: \neg K(CleanL) \wedge$$
$$\textbf{when } CleanR: K(CleanR) \wedge \textbf{when } \neg CleanR: K(\neg CleanR))$$

برای *Suck* گاهی مواقع در صورت تمیزی اتاق، کثیفی رخ میدهد با استفاده از حسگر خودکار تشخیص کثیفی می‌توان همواره
ین موضوع را تشخیص داد. بنابراین یک اثر شرطی فصلی خواهیم داشت:



$$\begin{aligned}&Action(Suck, \text{PRECOND:},\\ &\quad \text{EFFECT:}\textbf{when } AtL \wedge \neg CleanL\text{: } K(CleanL)\\ &\qquad \wedge \textbf{when } AtL \wedge CleanL\text{: } K(CleanL) \vee \neg K(CleanL) \wedge\\ &\qquad \textbf{when } AtR \wedge \neg CleanR\text{: } K(CleanR)\\ &\qquad \wedge \textbf{when } AtR \wedge CleanR\text{: } K(CleanR) \vee \neg K(CleanR)\end{aligned}$$

**؟ ۱۹.۱۲ به لیست کارهایی است که کارگزار برنامه‌ریزمجدد نمی‌تواند انجام دهد (در این فصل) مراجعه کنید. الگوریتمی به طور
لی ارائه دهید که بتواند یک یا چند مورد از آن کارها را انجام دهد.**

☑ حل: کارگزار برنامه‌ریز پیوسته که در بخش ۶.۱۲ توصیف شد حداقل دارای این قابلیت است که می‌تواند در طی حرکت
خود، یک هدف جدید را بپذیرد. یک هدف جدید به سادگی به عنوان یک پیش‌شرط در مرحله *Finish* اضافه می‌شود و
رنامه‌ریز باید در صورت امکان راهی برای ارضای آن به کمک سایر اهداف باقیمانده پیدا کند. از انجا که ساختمان داده‌ای که
وسط کارگزار برنامه‌ریز پیوسته ایجاد می‌شود، با اجرای برنامه افزایش می‌یابد، هزینه برنامه‌ریزی مجدد، به نسبت کم و شکست،
سیار زیاد خواهد بود. در حالت کلی نمی‌توان هیچ سقف زمانی مشخص برای تخمین این کار ارائه داد. زیرا تغییر حتی یک
تغیر حالت منفرد ممکن است نیاز به ساخت مجدد تمام برنامه از نقطه شروع باشد.

**؟ ۲۰.۱۲ مساله زیر را در نظر بگیرید: یک بیمار به مطب دکتر مراجعه می‌کند که علائم مریضی وی یا «ناشی از کمبود آب بدن»
یا «بیماری D» (و نه هر دو) می‌باشد این پزشک می‌تواند فقط دو نسخه تجویز کند: یا «نوشیدن (*Drink*)» که به طور
یرشرطی مشکل کمبود آب را حل می‌کند و یا «درمان (*Medicate*)» که برای درمان بیماری D تجویز می‌شود. ولی مساله
نجاست که اگر بیماری دچار کمبود آب باشد و این درمان برای وی تجویز شود، آنگاه اثرات جانبی بر روی او خواهد
ذاشت. توصیف مساله را در PDDL نوشته و یک برنامه بدون‌حسگر ترسیم کنیدکه این مساله را حل کند. تمام دنیاهای ممکن
مرتبط را بشمارید.**

☑ حل: فرض کنید $T$ گزاره‌ای برای نشان دادن کمبود آب در بدن بیمار و $S$ گزاره‌ای برای نشان دادن اثرات جانبی دارو باشد
اریم:

$$\begin{aligned}&Action(Drink, \text{PRECOND:}, \text{EFFECT:}\neg T)\\ &Action(Medicate, \text{PRECOND:}, \text{EFFECT:}\neg D \wedge \textbf{when } T\text{: } S)\end{aligned}$$

حالت شروع عبارت است از:

$$\neg S \wedge (T \vee D) \wedge (\neg T \vee \neg D)$$

رنامه راه‌حل عبارت است از:

$$(Drink\ ,\ Medicate)$$

همچنین امکان دارد، دو دنیا داشته باشیم: یکی دنیایی که T برقرار باشد و دیگری دنیایی که $D$ برقرار باشد. در مورد اول
وشیدن موجب $\neg T$ شده و *Medicate* هیچ اثری نخواهد گذاشت و در مورد دوم نوشیدن هیچ اثری نداشته و *Medicate*
وجب $\neg D$ میگردد. در هر دو مورد حالت نهایی به صورت $\neg S \wedge \neg T \wedge \neg D$ است.

**؟ ۲۱.۱۲ به مساله درمان تمرین قبل یک واکنش *Test* را بیافزایید که اگر *Disease* درست باشد و یا
*Known*(*CultureGrowth*) برقرار باشد، آنگاه دارای اثر شرطی *CultureGrowth* خواهد بود (یعنی شرایطی رخ دهد که فرد
یمار باشد و بدانیم که کمبود آب هم دارد، آنگاه واکنش *Test* موجب کمبود آب بدن بیمار شود). یک برنامه شرطی ترسیم کنید
ه این مساله را حل کرده و استفاده از واکنش *Medicate* را به حداقل برساند.**

☑ حل: یک برنامه راه‌حل عبارتست از:

$$[Test, \textbf{if } CultureGroulth \textbf{ then } (Drink, Medicate)]]$$

# فصل ۱ (ویرایش سوم)

◆ ***این تمرینات با هدف ایجاد انگیزه جهت تامل بیشتر در مباحث، طراحی گشته‌اند که برخی از آنها می‌توانند به عنوان موضوعات پروژه انتخاب شوند. پیشنهاد می‌شود که در ابتدا به طور مقدماتی برای حل مسائل اقدام کرده و پس از تکمیل با رجوع مجدد به آنها، پاسخ‌های خود را بازبینی کرده و به اشکالات خود پی ببرید.***

**؟۱.۱ این لغات را به زبان خود تعریف کنید: الف) هوش ب) هوش‌مصنوعی ج) عامل یا کارگزار(Agent) د) عقلانیت(Rationality) هـ) استدلال منطقی *(این تمرین مشابه تمرین ۱.۱ در ویرایش دوم است)***

☑ حل: **الف)** طبق تعریف لغت‌نامه، می‌توان هوش را اینگونه تعریف نمود: «ظرفیت کشف و اجرای دانش» یا «استعداد فکری در تفکر و استدلال» یا «توانایی درک و استفاده از تجربیات». تمامی این پاسخ‌ها قابل‌قبول تلقی می‌شوند ولی اگر بخواهیم تعریفی سنجیده ارائه دهیم، می‌توان هوش را به صورت «توانایی استفاده از دانش به منظور اجرای بهتر در یک محیط» تعریف نمود. **ب)** هوش‌مصنوعی مطالعه برنامه‌های یک کارگزار تعریف می‌شود که بتواند در یک محیط مشخص و به ازای یک معماری کارگزار خاص، به خوبی عمل کرده و قابل‌اجرا باشد. به بیانی دیگر هر روبات یا نرم‌افزار که برای انجام یک کار، عملکردی هوشمندانه از خود نشان دهد و مسلم باشد که این هوش طبیعی و ذاتی نیست، گفته می‌شود که دارای هوش‌مصنوعی است.**ج)** عامل یا کارگزار در واکنش به ادراکاتی که از یک محیط دارد، عکس‌العمل نشان می‌دهد. این عامل می‌تواند یک روبات و یا نرم‌افزاری خودکار باشد که درون آن برنامه‌هایی نصب می‌شود تا بتواند به طور خودکار و هوشمند کاری را انجام دهد. هر کارگزار محیط پیرامون خود را با توجه به وسایل تعبیه شده در آن درک کرده و سپس اقداماتی در آن انجام می‌دهد. روبات جاروبرقی دو سنسور دارد: یکی برای تعیین اتاق (مانند GPS) و دیگری برای مشاهده آشغال بر روی زمین (مانند دوربین). پس از دریافت این ورودی‌ها و با توجه به برنامه تعبیه شده در آن، اقداماتی انجام می‌دهد: حرکت به اتاق بعدی (با استفاده از چرخ)، مکیدن آشغال (با استفاده از موتور ). از آنجا که این کار به نظر هوشمند آمده و به طور خودکار انجام می‌شود، برنامه مربوطه در حیطه هوش‌مصنوعی قرار می‌گیرد. به طور خلاصه کارگزارها برای عهده گرفتن قسمتی از وظایف انسانی طراحی می‌شوند مثلا به جای آنکه یک خدمتکار همواره اتاق‌های خانه را چک کند تا در صورت کثیفی جارو نماید، و یا بجای آنکه یک انسان پشت ماشین نشسته و مسافتی را بپیماید تا مسافران را به مقصد برساند، می‌توان از انسان در سایر زمینه‌های سطح بالاتر کمک گرفته و برای این وظایف از کارگزارهای ساخته دست بشر استفاده نمود. **د)** عقلانیت یک مشخصه سیستمی است که بتوانیم با توجه به چیزهایی که می‌دانیم، «چیز درست» را بیابیم. در بخش ۲.۲ می‌توانید توضیح کاملتری از این موضوع را بیابید. ولی هر دو تعریف، استدلال را به طور کامل تشریح می‌کنند. البته بخش ۲۷.۳ نیز به این موضوع اختصاص دارد. **هـ)** استدلال منطقی، فرآیندی است که در آن جملات جدید از روی جملات قدیمی مشتق می‌شوند به طوری که اگر جمله قدیمی درست باشد، لزوما جمله جدید نیز درست باشد. (لازم به ذکر است که این موضوع به زبان مشخص یا نحو خاصی اشاره نمی‌کند بلکه برای یک فرم خوش‌تعریف از حقیقت صادق است.)

**؟۲.۱ مقاله اصلی نظریه تورینگ در مورد هوش‌مصنوعی (Turing,1950) را مطالعه کنید. در این مقاله، نویسنده تعدادی از مخالفت‌های ممکن با تست هوشمندی و نظریه ارائه شده خود را مطرح کرده و سپس آنها را تکذیب می‌کند. کدام یک از این مخالفت‌ها همچنان برقرارند؟ آیا تکذیب‌های وی معتبر است؟ آیا شما می‌توانید مخالفت‌های جدیدی بغیر از آنچه در زمان نوشتن مقاله ارائه شده است، بیابید؟ نویسنده در مقاله پیش بینی می‌کند که تا سال 2000، یک کامپیوتر 30 درصد شانس گذراندن 5 دقیقه از تست تورینگ را داشته باشد به شرط آنکه طرف مقابل ماهر نباشد. به نظر شما شانس کامپیوتر امروزه چقدر است؟ در 50 سال بعد چطور؟ *(این تمرین مشابه تمرین ۲.۱ در ویرایش دوم است)***

☑ حل: می‌دانیم که در تست تورینگ دیواری داریم که در یک طرف انسان و در طرف دیگر یک روبات قرار داده شده است. اگر آن دو با هم رابطه برقرار کرده و در طی این ارتباط انسان نفهمد که طرف مقابلش یک روبات است و یا یک انسان، آنگاه می‌گوییم تست تورینگ پیروز شده است. ولی در این تست، احتمال فریب‌دادن طرف مقابل بستگی زیادی به میزان ناشی‌بودن و یا مهارت طرف مقابل دارد. یک داوطلب جایزه لوبنر[1] در سال 2002 توانست در رقابتی مشابه با شرایط تست تورینگ، با فریب یک سئوال، برنده شود. اگرچه با مراجعه به دانسته‌های وی، تشخیص چگونگی تفکرش در رسیدن به پاسخ، بسیار دشوار ارزیابی شد. البته امروزه مثال‌های متعددی از عامل‌های هوشمند مانند روبات‌های چت(Chatbot) به طور آنلاین طراحی شده‌اند تا برای فریب انسان‌ها بکار روند به طوری که انسان در پشت سیستم نمی‌فهمد که طرف مقابل وی در چت یک انسان است و یا یک

[1] هر ساله جایزه لوبنر به برنامه‌ای اعطا می‌شود که بتواند در نسخه‌ای از تست تورینگ بهتر از دیگران عمل نماید.

یات، حتی با پرسیدن سئوالات متفاوت از او نیز به این ماجرا پی نمی‌برد. به عنوان مثال به حساب See Lenny Foner در یات چت Julia به آدرس

www.media.mit.edu/people/foner/Julia

اجعه نمایید. امروزه شانس برنده‌شدن کامپیوترها در تست تورینگ به حدود 10 درصد کاهش یافته‌است زیرا تنوع مهارت‌های رف مقابل نسبت به پیشرفت برنامه‌ها، افزایش قابل‌ملاحظه‌ای دارد. ولی در 50 سال آینده، انتظار داریم که شانس کامپیوترها شکست انسان بیشتر شده و سرمایه‌گذاری بیشتری در صنعت سرگرمی (فیلم، بازی‌های ویدیویی، تجارت و ...) جهت ایجاد و ارگیری بازیکنان مصنوعی شود تا بتوان بازیکنانی باورنکردنی ایجاد نمود.

**۳.۱ آیا اقدامات واکنشی (نظیر پس کشیدن دست از روی اجاق داغ)، عقلانی هستند؟ آیا آنها هوشمند تلقی می‌شوند؟** ***(این رین مشابه تمرین ۱.۰ در ویرایش دوم است)***

حل: این کار عقلانی است زیرا هر واکنش دیگری که کندتر از این و با مشورت بیشتری انجام شود، موجب آسیب رساندن شتر به دست شما می‌شود. حال اگر هوش را به معنای «اجرای دانش» یا «استفاده از تفکر و استدلال» تعریف کنیم، برای ام چنین عکس‌العملی نیاز به هوشمندی نداریم.

**۴.۱ فرض کنید در بخشی از یک برنامه آنولوژی تغییراتی اعمال کنیم که بتواند در یک آزمون هوش استاندارد، امتیاز 200 را سب کند. آیا این برنامه، هوشمندتر از انسان عمل می‌کند؟ توضیح دهید.** ***(این تمرین مشابه تمرین ۱.۵ در ویرایش دوم است)***

حل: خیر. امتیازات کسب‌شده در آزمون هوش، بستگی به عوامل مختلفی چون موفقیت در دانشگاه دارد، ولی به شرط آنکه ط انسان‌های نرمال و ایده‌آل را مورد بررسی قرار دهیم. همچنین آزمون هوش همه‌منظوره نیست پس یک برنامه که به طور ص فقط برای کسب امتیاز در آزمون‌های هوش و یا حتی فقط بخش آنالوژی طراحی شود، در سایر زمینه‌های آزمون‌هوش یار ضعیف خواهد بود. بنابراین نمی‌توان چنین برنامه‌ای را هوشمندتر از انسان در تمام مسائل دانست. آنالوژی زیر را درنظر یرید: اگر یک شخص 100 متر را در 10 ثانیه بدود، ممکن است وی را یک ورزشکار خوب توصیف کنیم و انتظار داشته باشیم در سایر زمینه‌ها نیز موفق باشد مانند پیاده‌روی، مسابقه پرش، دو بامانع و پرتاب توپ. یک هواپیمای بوئینگ 747 را با آنکه 10 متر را در 0.4 ثانیه طی می‌کند، طبق این آنالوژی یک ورزشکار خوب نمی‌دانیم پس از او انتظار موفقیت در مسابقه دو با ع و پرتاب توپ را نداریم. پس با اینکه هواپیما امتیاز مربوطه را در یک زمینه کسب کرده است، در سایر زمینه‌ها موفق حسوب نمی‌شود (مانند روباتی که در صورت سئوال ذکر شده است). حتی آزمون هوش برای انسان‌ها نیز جای بحث دارد زیرا مولا فرضیاتی قبلی در مورد توانایی‌های فطری آنها درنظرگرفته می‌شود (علاوه بر اثرات آموزش) که درحالت کلی بر نتیجه گذار است. در سال 1981 مقاله‌ای تحت عنوان *The Mismeasure Of Man* توسط استفان‌جی‌گلد و نورتن ارائه گشت و ند سال بعد در سال 1993 هاوارد گاردنر مقاله *Multiple intelligence: the theory in practice* را ارائه داد که در رد آزمون‌های هوش و اینکه آنها چه چیزی را اندازه گرفته و چه جنبه‌هایی از هوش را در نظر می‌گیرند، بحث بیشتری می‌-د.

**۵.۱ ساختار عصبی حلزون دریایی بنام *Aplysia* به طور گسترده مورد مطالعه قرار می‌گیرد (اولین بار توسط نوبل لورات اریک دل استفاده گشت) زیرا دارای 20000 نورون است که بیشتر آنها بزرگ بوده و به راحتی قابل دستکاری هستند. فرض کنید به طور تقریبی سیکل زمانی برای نرون *Aplysia* و نرون انسانی یکسان باشد. اگر در هر ثانیه حافظه آپدیت شود، چه قدرت حاسباتی نیاز است؟ آن را با قدرت کامپیوترهای high-end که در شکل ۳.۱ ترسیم گشته است، مقایسه کنید.**

حل: با توجه به شکل، قدرت محاسباتی کامپیوتر 100 مرتبه بیشتر است.

**۶.۱ چطور خویشتن‌نگری(Introspection) (گزارشی از روند تفکر داخلی) می‌تواند غیردقیق باشد؟ آیا من می‌توانم در مورد یزی که فکر می‌کنم، اشتباه کنم؟ بحث کنید.** ***(این تمرین مشابه تمرین ۶.۱ در ویرایش دوم است)***

حل: همان طور که شما از تمام مراحل صورت گرفته برای ایجاد یک ضربان قلب خود آگاه نیستید، از بیشتر اتفاقاتی که در کرتان می‌گذرد نیز ناآگاهید. البته ممکن است از برخی از فرآیندهای فکری مطلع باشید ولی همچنان اکثریت آنها به طور هم و غیرشفاف هستند. به همین منظور رشته روانکاوی بر این ایده بنا شد تا یک نفر به طور حرفه‌ای آموزش دیده و بتواند گران را در تحلیل تفکرشان یاری نماید.

**۷.۱ کدام یک از این سیستم‌های کامپیوتری، نمونه‌ای از هوش‌مصنوعی تلقی می‌شوند: الف) اسکنر بارکد در سوپرمارکت. ب) تورهای جستجوی وب. ج) منوی مبتنی بر صدا در تلفن. د) الگوریتم مسیریابی اینترنت که به هر موقعیت در شبکه به طور نامیک پاسخ می‌دهد.**

حل: **الف)** با وجود اینکه اسکن‌کردن بارکد در دسته مسائل ماشین بینایی حواس قرار می‌گیرد ولی یک سیستم ش‌مصنوعی نیست زیرا مسئله خواندن بارکد فرمی محدود و مصنوعی از تفسیرهای بینایی است که برای سخت‌افزار مشخص، قت بالایی طراحی شده است. **ب)** برخی از بخش‌های موتورهای جستجو از روش‌های هوش‌مصنوعی بهره برده و برخی دیگر یر. یافتن یک صفحه وب مرتبط با تقاضای موردنظر کاربر، خود یکی از مسائل عرصه فهم زبان‌های طبیعی است. برخی از تورهای جستجو نظیر Ask.com که صفحات مرتبط را در یک دسته قرار می‌دهند، از تکنیک‌های خوشه‌بندی استفاده کنند ولی عملکرد برخی دیگر از موتورهای‌جستجو بر مبنای متدهای هوش‌مصنوعی است به عنوان نمونه بخش اصلاحگر

نوشتاری در موتورهای جستجو[2]، نوعی عملیات داده‌کاوی است که با توجه به اصلاحاتی که کاربران قبلی در طی جستجوهای خود بکار برده‌اند، عمل می‌کند. ولی مسئله فهرست کردن میلیون‌ها صفحه وب که بتوانند در طی چند ثانیه بازیافت شوند، یک مسئله مرتبط با طراحی پایگاه‌داده است و ربطی به هوش‌مصنوعی ندارد. ج) اگر مسئله را محدودشده در نظربگیریم، مثلا منوها فقط با لغات محدودی مانند «بله» یا «خیر» عمل کنند، این مسئله تحت کنترل طراح ساده شده است. ولی در حالت کلی، این مسئله اینگونه نیست و شامل یک فضای کنترل نشده با انواع صداها و تلفظ‌های مختلف می‌باشد. از آن جمله می‌توان به برنامه-هایی اشاره نمود که با صدا عمل می‌کنند و معمولا در کمپانی‌های تلفن مورداستفاده قرار می‌گیرند، که عملکرد آنها بر مبنای لغات زیاد و متنوع می‌باشد، پس قطعا یک برنامه هوش‌مصنوعی هستند. د) برای اینکه این عامل را در فضای سایبری هوشمند بدانیم به چند ویژگی آن توجه می‌کنیم: عملیات گمراه‌کننده است، اطلاعات موجود جزئی و ناقص است، تکنیک اکتشافی است(نه لزوما بهینه)، و موقعیت کاربر در جهان دینامیک است. تمام اینها، مشخصات یک اقدام هوشمند را نشان می‌دهند، پس این عملیات با آنچه که در تشخیص انسانی انجام می‌شود بسیار متفاوت خواهد بود.

**؟۸.۱ بسیاری از مدل‌های محاسباتی که تاکنون برای سیستم‌های تشخیصی ارائه شده‌اند، دارای عملیات پیچیده ریاضی هستند مانند انجام کانولوشن یک تصویر با یک تابع گاوسی که موجب یافتن مینیمم تابع آنتروپی می‌شود[3]. بیشتر انسان‌ها (و قطعا تمام حیوانات) هرگز این نوع ریاضیات را نیاموخته‌اند زیرا به دانشگاه نرفته‌اند، بنابراین نمی‌توانند عملیات کانولوشن یک تصویر را با تابع گاوسی در سرشان محاسبه کنند. پس چه اتفاقی می‌افتد که می‌گوییم سیستم بینایی می‌تواند این نوع بهبود تصویر را انجام دهد در حالیکه شخص حقیقی هیچ ایده‌ای در مورد چگونگی محاسبه ندارد؟**

☑ حل: به جرأت می‌توان گفت که مغز می‌تواند این عملیات پیچیده ریاضی را مانند سایر محاسبات انجام دهد ولی به شرط آنکه فرمول محاسبه کانولوشن را بداند. ولی تا قبل از دو قرن پیش انسان‌ها و حیوانات هیچ اطلاعی از محاسبه کانولوشن گاوسی نداشتند. اکنون سئوالی جالب مطرح می‌شود و آن این است که انسان در حقیقت چگونه این کار را انجام می‌دهد یعنی انسان می‌تواند ببیند ولی نمی‌تواند کانولوشن گاوسی را محاسبه کند! راه حل این پرسش در آن است که محاسبه کانولوشن گاوسی بخشی از عملیات دیدن وی محسوب می‌شود که به طور خودکار توسط مغز صورت می‌گیرد.

**؟۹.۱ به چه علت در سیستم‌های عقلانی، تکامل(Evolution) حاصل می‌شود؟ این سیستم‌ها برای رسیدن به چه اهدافی طراحی شده‌اند؟** ***(این تمرین مشابه تمرین ۱.۹ در ویرایش دوم است)***

☑ حل: تکامل به معنای تولید مجدد و در نتیجه بقای اندام‌هایی است که به قدرکافی موفق بوده‌اند و در این تکامل، اندام‌هایی مورد توجه بیشتر قرار می‌گیرند که با دوام سن بلوغ جنسی، معیار کارایی را بهینه کنند. از آنجا که عقلانیت فقط به معنای بهینگی معیار کارایی است بنابراین در راستای تکامل قرار دارد.

**؟۱۰.۱ هوش مصنوعی یک علم است یا مهندسی؟ یا هیچ کدام؟ و یا هر دو؟ توضیح دهید.**

☑ حل: این سئوال مربوط به ماهیت اصلی مسائل هوش مصنوعی و نیازمندی‌های آن برای حل مسائل می‌شود ولی می‌تواند به عنوان یک سئوال حوزه علم اجتماعی در مورد هوش مصنوعی نیز تعبیر شود. یک *علم*، زمینه‌ای از مطالعات تعریف می‌شود که می‌تواند با متدهای علمی، دانش‌تجربی را بدست آورد. و *مهندسی*، به معنای آن است که یک دانش تجربی ثابت را مدنظر گرفته و از آن استفاده کنیم تا مسائل موردعلاقه جامعه را حل نماییم. البته مهندسان از قسمت‌هایی از علم استفاده می‌کنند (مانند اندازه‌گیری مشخصاتی از مواد ساختمانی)، همین طور عالمان نیز از بخش‌هایی از مهندسی کمک می‌گیرند (مانند ساختن ابزار و وسایل جدید). همانطور که در بخش ۱.۱ گفته شد، قسمتی از هوش‌مصنوعی که به انسان مربوط می‌شود مشخصا یک *علم* تجربی است (علم تشخیص) زیرا این قسمت با استفاده از تجربیات روانشناسی، چگونگی عملکرد تشخیص انسانی را می‌یابد. ولی از جنبه عقلانی چگونه تفسیر می‌شود؟ اگر هوش‌مصنوعی را به صورت رابطه‌ای کلی بین محیط عملکرد، وسیله محاسباتی و برنامه در نظر بگیریم به طوری که هوش‌مصنوعی از آن وسیله محاسباتی استفاده کند تا بتواند در یک محیط عملکرد، بهترین کارایی را داشته باشد، آنگاه جنبه عقلانی هوش‌مصنوعی، حقیقتا ریاضیات مهندسی است که به هیچ دانش تجربی در مورد جهان حقیقی(و محیط وظیفه حقیقی) نیاز ندارد. به برنامه‌ای که در یک محیط مشخص، بخوبی کار می‌کند، قضیه(Theorem) می‌گویند. در عمل، ما به محیط‌های وظیفه‌ای علاقه‌مند هستیم که تخمینی از جهان حقیقی باشند پس در جنبه عقلانی هوش‌مصنوعی نیاز به فهم آن داریم که جهان حقیقی شبیه چیست. به عنوان مثال، در مطالعه عامل‌های عقلانی که با یکدیگر در تعامل‌اند، محیط‌هایی بیشتر موردنظر هستند که انسان را نیز شامل شوند پس بایستی بدانیم زبان انسانی شبیه چیست. به منظور درک این مطالعات، به سنسورهایی مانند دوربین مراجعه می‌کنیم که می‌توانند اطلاعات مفیدی را از جهان حقیقی استخراج کنند (البته در یک جهان بدون نور، دوربین قابل استفاده نیست). علاوه بر این، برای طراحی الگوریتم‌های بینایی، که برای استخراج اطلاعات از تصاویر دوربین مناسب باشند، نیاز به فهم جهانی داریم که این تصاویر در آن تولید شده‌اند. کسب فهم مورد نیاز، مربوط به مشخصات صحنه، نوع اشیا، علامتگذاری سطوح و غیره می‌شود که علمی متفاوت با علم فیزیک، شیمی و

---

[2] این بخش از موتورهای جستجو، غلط‌های املایی نوشته شده در متن کاربر را اصطلاح می‌کند.

[3] کانولوشن یک عملیات ریاضی و محاسباتی است که بر روی تک‌تک پیکسل‌های یک تصویر انجام می‌شود تا تصویر را بهبود بخشد.

ست‌شناسی است ولی همچنان یک *علم* است. به طور خلاصه، هوش‌مصنوعی یک مهندسی در کنار علم تجربی تعریف می‌شود
ا برخی جنبه‌های جهان حقیقی بر روی طراحی این سیستم‌های هوشمند اثرگذارند.

**۱۱.۱ «مسلما کامپیوترها هوشمند نیستند زیرا فقط می‌توانند آن چیزی را انجام دهند که برنامه‌نویسان به آنها می‌گویند» آیا جمله آخر صحیح است؟ آیا جمله آخر، دلیلی برای جمله اول محسوب می‌شود؟** ***(این تمرین مشابه تمرین ۱.۱ در ویرایش دوم است)***

حل: این موضوع بستگی به تعریف شما از «هوش» و «tell[4]» دارد. از یک جهت، کامپیوترها فقط می‌توانند دستوراتی را اجرا
ند که برنامه‌نویسان به آنها گفته‌اند ولی از جهت دیگر، چیزی که برنامه‌نویس از کامپیوتر برای انجام می‌خواهد، معمولا خیلی
تر از آن چیزی است که کامپیوتر واقعا می‌تواند انجام دهد. احتمالا شما نیز این موضوع را در زمانی که یک برنامه معمولی یا
برنامه یادگیری‌ماشین حرفه‌ای نوشته‌اید به خوبی درک نموده‌اید که قابلیت کامپیوتر همیشه بیشتر از برنامه‌ای است که
ی آن نوشته می‌شود. بنابراین از یک سو، ساموئل با استفاده از tell از کامپیوتر می‌خواهد: «بیاموز که بازی چکر را بهتر از من
ی کنی و این شیوه را برای بازی ادامه بده» و از سوی دیگر می‌توان با استفاده از tell از وی بخواهیم: «این الگوریتم یادگیری
دامه بده». و اینگونه کامپیوتر با توجه به مجموعه دانستنی‌هایش (پایگاه دانش) می‌آموزد که چگونه بازی کند. اکنون در
عیتی هستیم که نمی‌دانیم یادگیری بازی چکر نشانی از هوش است و یا خیر. (یا شاید شما فکر کنید که یادگیری بازی به
وه درست، نیازمند هوش است ولی به این شیوه خیر). یا حتی ممکن است نظر شما بر آن باشد که هوشمندی درون برنامه-
س یا حتی کامپیوتر نهفته است.

**۱۲.۱ «مسلما حیوانات هوشمند نیستند زیرا فقط می‌توانند آن چیزی را انجام دهند که ژن‌هایشان به آنها می‌گویند» آیا جمله
ر صحیح است؟ آیا جمله آخر، دلیلی برای جمله اول محسوب می‌شود؟** ***(این تمرین مشابه تمرین ۱.۲ در ویرایش دوم است)***

حل: پاسخ این تمرین به نوعی مشابه پاسخ تمرین قبل است. بنابراین همانطور که در مورد هوشمندی کامپیوتر تصمیم‌گیری
دیم، می‌توان همان نتیجه را در مورد حیوانات نیز تعمیم داد با این تفاوت که دلیل ما برای هوشمندی یک چیز، در مکانیزم
کرد آن نهفته است پس برنامه‌نویسی در ژن‌ها را متناظر با برنامه‌نویسی انجام شده توسط انسان در نظر بگیرید. لازم بذکر
ت که Searle این مکانیزم را برای آرگومان‌های اتاق چینی استفاده کرده است.

**۱۳.۱ «مسلما حیوانات، انسان‌ها و کامپیوترها نمی‌توانند هوشمند باشند زیرا فقط می‌توانند آن چیزی را انجام دهند که قوانین
یک به اتم‌های سازنده‌شان دستور می‌دهند». آیا جمله آخر صحیح است؟ آیا جمله آخر، دلیلی برای جمله اول محسوب
شود؟** ***(این تمرین مشابه تمرین ۱.۳ در ویرایش دوم است)***

حل: پاسخ تمرین ۱.۱ را می‌توان برای این تمرین تعمیم داد.

**۱۴.۱ با توجه به مبحث هوش‌مصنوعی، هم اکنون کدام یک از این مسائل توسط کامپیوتر قابل انجام است: الف) انجام بازی
گ پنگ در حد قابل‌قبول ب) رانندگی در مرکز شهر قاهره در مصر ج) رانندگی در ویکتوروایل-کالیفرنیا د) خرید خواروبار
دنیاز هفتگی در فروشگاه هـ) خرید خواروبار موردنیاز هفتگی در وب و) انجام بازی قابل قبولی از بریج به طور رقابتی ز)
ف و اثبات تئوری‌های جدید ریاضی ح) نوشتن یک داستان خنده‌دار بطور عمدی ط) ارائه مشورتی حقوقی درمورد مسئله‌ای
ن از قانون ی) ترجمه بی‌درنگ مکالمات انگلیسی به مکالمات سوئدی ک) انجام یک عمل جراحی پیچیده سعی کنید. برای
ردی که هم اکنون در جامعه موجود نیستند، مشکلات را یافته و سپس پیش‌بینی کنید که چه زمانی این مشکلات برطرف
هند شد.** ***(این تمرین مشابه تمرین ۱.۷ در ویرایش دوم است)***

حل: **الف)** پینگ‌پنگ: این مسئله قبلا توسط روبات اندرسون انجام شده و کارایی قابل‌قبولی نیز ارائه داده است (Anderson,198). **ب)** رانندگی در مرکز شهر قاهره- مصر: خیر. رانندگی اتوماتیک مسئله‌ای است که عوامل مختلفی در دخیل‌اند به همین خاطر معمولا در اینگونه سیستم‌ها فرض می‌شود که عوامل موثر، نسبتا ثابت باشند: مثلا جاده دارای خط
ط و شانه خاکی باشد، جلوی ماشین طبق مسیر پیش‌بینی شده حرکت کند، ماشین‌ها در همان طرف خود حرکت کنند و ....
علاوه ممکن است به منظور مدرنیزه کردن ترافیک، تعدادی از جاده‌های یک شهر بزرگ در طی روز دچار تغییراتی شوند.

---

ر روبات درون خود مجموعه دانشی در مورد کاری که قرار است انجام دهد دارد. به عنوان مثال یک دکتر را درنظر بگیرید. این فرد قبل از
سیل در این رشته معلوماتی اندک در مورد پزشکی دارد مانند اینکه «فرد سرماخورده سردرد دارد». به این دانش اولیه، دانش پس‌زمینه
ه می‌شود. این فرد با تحصیل در رشته پزشکی به مرور مجموعه دانش خود را اضافه می‌کند. هر اطلاعات جدید در قالب یک جمله به او
ه می‌شود. این کار را با نماد tell نشان می‌دهیم. مثلا tell(قرص استامینوفن برای سرماخوردگی خوب است)، tell(مصرف بیش از چهار
قرص در روز موجب سکته می‌شود) و ... . با مطالعه هر چه بیشتر مجموعه دانش این پزشک که به آن پایگاه‌دانش (KB) می‌گوییم
یلتر شده و میتواند با استفاده از آن پایگاه، استنتاج نماید مثلا به بیماری که سردرد دارد، سه عدد قرص استامینوفن تجویز نماید. دکتر
ن است در بسیاری از موارد برایش سئوالی ایجاد شود مانند اینکه «سردرد نشانه کدام بیماری است؟»، برای پاسخ آن باید به مجموعه
ل ذهنی خود مراجعه کند که آن را به طور نمادین با Ask(سردرد نشانه کدام بیماری است) نمایش می‌دهیم. پس برای افزودن یک جمله
ل به مجموعه پایگاه‌دانش در روبات از نماد tell و برای سئوال پرسیدن از پایگاه از دستور Ask استفاده می‌کنیم. اگر دقت کنید پزشکی
طالعات بیشتر داشته باشد در شناسایی بیماری و تجویز نسخه ماهرتر است زیرا پایگاهی کاملتر دارد.

بنابراین مجموع این شرایط نشان می‌دهد که رانندگی در شهر قاهره بسیار غیرقابل پیش‌بینی خواهد بود و برای یک روبات میسر نمی‌باشد. **ج)** رانندگی در ویکتوروایل-کالیفرنیا: همانطور که در مسئله شهری DARPA نشان داده شد، برخی کامیون‌ها برای جاده‌های انتقالی، تقاطع‌ها، ترافیک‌ها و پیاده‌روهای خوش‌رفتار مدیریت شده و آموزش دیده‌اند. بنابراین در شرایط خوب بصری قرار دارند. **د)** خرید در فروشگاه: خیر. هیچ روباتی نمی‌تواند همزمان وظایفی چون حرکت در محیط‌های شلوغ، دیدن و تشخیص اشیاء در بین انواع مختلف و گرفتن شی موردنظر بدون آسیب‌زدن به آنها را انجام دهد. شاید هریک از این وظایف به تنهایی قابل انجام باشد ولی انجام مجموعه آنها به طور همزمان کاری بس دشوار است. **هـ)** خرید در وب: بله- اگر سایت فروش به گونه‌ای طراحی شده باشد که در طی زمان تغییرات اساسی نکند، روبات‌های نرم‌افزاری قادر به انجام وظایف مشخص می‌باشند. **و)** انجام بازی بریج: بله. هم‌اکنون برنامه‌هایی مانند GIB در یک سطح قابل‌قبول کار می‌کنند. **ز)** اثبات قضیه: بله. به عنوان مثال اثبات جبری ROBBIN که در فصول بعدی بیان شده است. **ح)** داستان خنده‌دار: خیر. شاید برخی از شعرها و متون ادبی که توسط کامپیوتر تولید می‌شود بسیار جالب و خنده‌دار باشد ولی امری غیرعمدی است. البته در برخی موارد کامپیوتر یک شعر زیبا را بازگو می‌کند که در حافظه آن قرارداده شده است. در این حالت برنامه فقط از روی حافظه متن را بازگو کرده و در حالت کلی سرودن شعر و متن برای وی میسر نیست. **ط)** مشورت قانونی: بله. در تاریخچه هوش‌مصنوعی، نرم‌افزارهای استدلال خودکار منطقی قدمتی زیاد دارند. به عنوان مثال سیستمی خبره برای دولت UK مبتنی بر پرولوگ طراحی و استفاده گشت تا بتواند افراد عادی را در ریزه‌کاری‌های امنیت اجتماعی و قوانین ملی راهنمایی کند. گفته می‌شود که این سیستم امنیت ملی برای دولت UK توانست حدود 150 میلیون دلار در اولین سال عملکردش ذخیره کند. سپس این نرم‌افزار را در مناطق پیچیده نظیر توافق قوانین برای کدگذاری دانش وب گسترش دادند که مربوط به تراکنش‌های تجاری و موافقت‌نامه‌ها و کسب‌وکارها می‌شد. **ی)** ترجمه: بله. اینکار در حالت محدود تقریبا انجام شده است. به مباحث انجام شده توسط کی‌گاردن و نوروین (سال 1994) و والستر (سال 2000) که در زمینه ترجمه صحبت و برخی محدودیت‌های کنونی بحث می‌کند، مراجعه کنید. **ک)** جراحی: بله. روبات‌ها به طور گسترده جهت عملیات‌های جراحی استفاده می‌شوند البته آنها تحت دستورات پزشک اقدام می‌کنند.

**؟ ۱۵.۱ در بسیاری از مسائل هوش‌مصنوعی وظیفه‌ای استاندارد تعریف شده و از تعدادی خبره جهت ارائه بهترین راهکار دعوت می‌شود. از آن جمله می‌توان به مسئله اهدای DARPA در مسابقات روباتیک ماشینی، رقابت در مسابقات بین‌المللی، روبات بازیکن فوتبالیست، کنفرانس بازیابی اطلاعات TREC، ترجمه ماشینی و تشخیص صحبت اشاره‌نمود. پنج مورد از این موارد را انتخاب نموده و بررسی‌های صورت گرفته بر روی آنها در طی سالیان را توصیف کنید. ترفندهای هوش‌مصنوعی موجب چه پیشرفت‌هایی در آنها شده است؟ برای افزودن ایده‌های جدید چه میزان انرژی باید صرف شود؟**

☑ حل: مباحث عنوان شده برای این موارد منطبق بر واقعیت است ولی میزان توصیفات و نوشتار هر فرد از آنها متفاوت است. **الف)** مسئله اهدای **DARPA** در مسابقات روباتیک ماشینی: در سال 2004، شرکت‌کنندگان برای دریافت این جایزه بایستی مسافت 240 کیلومتر را در بیابان Mojave رانندگی می‌کردند که به خاطر خودمختاری رانندگان، وضعیتی پراسترس بوده و معمولا هیچ شرکت‌کننده‌ای نمی‌توانست مسابقه را به پایان برساند. بهترین تیم شرکت‌کننده که تنها توانست 12 کیلومتر از 240 کیلومتر را بپیماید، تیم CMU بود. بنابراین در سال 2005 تصمیم به ایجاد تغییراتی در مسابقه گرفته شد لذا هر دور تبدیل به 212 کیلومتر به همراه پیچ‌های کمتر و عریضتر شدن جاده‌ها شد. در این حالت 5 تیم موفق به پایان مسابقه شدند که Stanford اولین گروهی بود که به خط پایان رسید و دو شرکت‌کننده CMU نفرات بعدی بودند. این موضوع توانست برای روباتیک و مسائل مربوطه به آن راه ورودی مناسبی باشد. در سال 2007 مسئله شهری با نام مسابقه Urban به گونه‌ای طراحی شد که ماشین‌ها در موقعیتی مشابه شهر قرار گرفته و بایستی با رعایت قوانین ترافیکی و عدم‌برخورد با سایرین، به مسابقه ادامه دهند. در این مسئله، تیم CMU توانست گوی سبقت را از Stanford برباید. با انجام رقابت‌هایی اینچنینی می‌توان شرایط رقابت مناسب برای نزدیکی تئوری به عمل را یافت. همانطور که شکست در مسابقه 2004 توانست نیازمندی‌های صحیح رقابت را نشان دهد. از دیگر موارد مهم در موارد رقابتی آن است که بایستی در زمان مناسب که حداقل تئوری نشان می‌دهد انجام شود. به عنوان مثال می‌توان به رقابت Dickman (که در آن ماشین Vamp به طور خودکار برای مسافت 158 کیلومتر در سال 1995 رانندگی کرد ) و یا Pomerleau (که ماشین Navlab مسافت 5000 کیلومتر را در همان سال 1995 در آمریکا طی نمود که 98 درصد موارد تحت کنترل خودکار بوده البته به جز ترمز و شتاب‌سنج که در اختیار راننده بود.) اشاره نمود. **ب)** رقابت در بازی‌های بین‌المللی: در سال 1998، پنج بازیکن وارد رقابت شدند: STAN, SGP, IPP, HSP, Blackbox . در صفحه‌ای به آدرس ftp://ftp.cs.yale.edu/pub/mcdermott/aipscomp-results.html عنوان شده است که "همه این بازیکنان با توجه به استعدادشان در چند سال پیش بخوبی بازی کردند". معمولا بیشتر بازی‌ها دارای 30 یا 40 مرحله بودند ولی در گاهی اوقات تعداد مراحل به بیشتر از 100 نیز می‌رسید. در سال 2008، رقابت‌ها اندکی متفاوت گشت به طوری که بخش‌های بیشتری مدنظر قرار گرفت (انجام بازی در مقابل بهینگی، ترتیب مراحل در مقابل موقتی‌بودن، استاتیک‌بودن در مقابل یادگیری). در این مسابقه 25 بازیکن شرکت کردند که بخشی از آنها همان گروه‌های سال 1998 یا فرزندان آنها بودند و بخشی از تیم‌ها جدید بودند که یافتن راه‌حل در این مسابقه بسیار طولانی‌تر از مسابقه سال 1998 بود. به طور خلاصه این مسابقه توانایی‌های بازیکنان و مشارکت آنها را مورد بررسی و سنجش قرار می‌داد. در سال 1990 امکان ارائه یک مقاله بازی در حالت

رفا تئوری موجود بود ولی اکنون بایستی کارایی یک بازی اثبات شود که کاری بس دشوارتر است. پس بایستی به این
ی‌های رقابتی، اعتباراتی اختصاص یابد. به هرحال، برخی محققان بر روی انواع خاصی از مسائل که جنبه رقابتی دارند فعالیت
ی‌کنند ولی همچنان برای ورود به کاربردهای جهان حقیقی کافی و مناسب نیستند. **ج)** روبات فوتبالیست روبوکاپ: این رقابت
سیار متداول توانست در سال 2009، تعداد 407 تیم را از 43 کشور جهان جذب خود نماید (در سال 1997، تعداد 38 تیم از
1 کشور جهان شرکت داشتند). با پیشرفت تجهیزات روباتیکی، این روبات‌ها شباهت بیشتری به انسان‌ها یافته و هم قابلیت‌های
کتیکی آنها بهبود یافته‌است. اگر چه در حالت کنترل‌های توزیع شده، ابداعاتی صورت گرفته ولی در سال‌های اخیر بیشتر
یم‌های برنده از مهارت‌های فردی بیش از مهارت‌های تیمی استفاده‌کرده‌اند. این رقابت‌ها برای کسب مشارکت و جذابیت بیشتر،
مچنان در حال پیشرفت هستند تا بتوانند در سال 2050 یک تیم انسانی را شکست دهند. **د)** کنفرانس بازیابی اطلاعات
TREC: یکی از قدیمی‌ترین رقابت‌ها محسوب می‌شود که برای اولین بار در سال 1992 آغاز شد. در این رقابت‌ها هر ساله
عدادی محقق دور هم جمع شده و در مورد مقالاتی که در زمینه‌های گوناگون گردآوردی شده‌است، بحث و تبادل‌نظر می‌کنند
در مورد کیفیت آنها نظر می‌دهند. در سال‌های اولیه، این کنفرانس سعی در جمع‌آوری و ارزیابی متون نوشته شده‌داشت که با
ذشت زمان این دست‌نوشته‌ها، حجم زیادی را اشغال می‌کردند. ولی از سال 2000 کنفرانس TREC مجموعه نوشته‌های خود
در وب جهان گستر قرار داد تا برای تمام افراد در دسترس باشد و با این کار توانست به مرسولات بیشتری دست‌یابد. در این
الت جستجو در موتورهای جستجو بهتر از تحقیقات آکادمیک شد. **هـ)** ارزیابی ترجمه انجام شده توسط ماشین NIST: این
جموعه از ارزیابی‌ها (که معمولا از آنها به رقابت یاد نمی‌شود) از سال 2001 وجود داشته‌است. با گذشت زمان علاوه بر افزایش
یفیت ترجمه ماشینی، در تعداد زبان‌های تحت پوشش جهت ترجمه نیز پیشرفت‌هایی حاصل شده‌است. بیشتر این کاربردها در
دیل یک زبان از قواعد گرامری برمبنای علم آمار استفاده می‌کنند. ارزیاب NIST می‌تواند این تغییرات رخ داده را پیگیری
ند ولی قادر به اعمال تغییرات نیست. در حالت کلی می‌بینیم که این رقابت‌ها با گذشت زمان، میزان عملکردی بهتر داشته‌اند
ه سنجش عملکرد هر رقابت با توجه به معیاری است که از روی توانایی مورد بررسی تعریف می‌شود. در مورد ICAPS، برخی
حققان بازی نگران جذابیت‌های خیلی زیاد این رقابت بودند. در برخی موارد، مرحله مورد بررسی درون رقابت نهفته است. مانند
TREC، که منابع دردسترس در موتورهای جستجوی تجاری بیشتر از منابع جستجوی آکادمیک است. در این مورد، رقابت
TREC، مفید تلقی شده و به آموزش بسیاری از مردم مراجعه‌کننده به موتورهای جستجو کمک می‌کند. همچنین برای افزودن
ده‌های جدید هیچ انرژی صرف نمی‌شود.

# فصل ۲ (ویرایش سوم)

**؟ ۱.۲ فرض کنید معیار کارایی کارگزار در محیط، فقط با توجه به $T$ مرحله زمانی نخست در نظر گرفته شده و تمام اتفاقات پس از آن، نادیده گرفته شود. نشان دهید که یک واکنش عقلانی ممکن است علاوه بر حالت محیط به مرحله زمانی صورت گرفته در آن نیز بستگی داشته باشد.**

☑ حل: این سوال قصد بررسی فهم دانشجویان از اصطلاحات محیط، واکنش‌عقلانی و معیار کارایی را دارد. در محیط‌های ترتیبی که به ازای انجام به‌موقع‌کار پاداشی پرداخت می‌شود، می‌توان در هر مرحله از دنباله واکنش‌ها، به شیوه‌ای عمل نماییم که تا حد امکان پاداش بیشتری کسب کنیم. فرض کنید در هر حالت از این دنباله شما دو واکنش پیش‌رو دارید که می‌توانید یکی از آنها را انتخاب نمایید که این واکنش‌ها را با $a$ و $b$ نامگذاری می‌کنیم. همچنین دو فرضیه را در نظر بگیرید: کارگزار یا در زمان $T$ در حالت $s$ است و یا در زمان $T$-1. می‌دانیم که اگر در حالت $s$ بوده و واکنش $a$ را انتخاب کنیم، به حالت $s'$ رسیده و هیچ پاداشی نمی‌گیریم. ولی اگر درحالت $s$ واکنش $b$ را انتخاب کنیم، 1 امتیاز گرفته و دوباره به حالت $s$ برمی‌گردیم و می‌دانیم که گذر از حالت $s'$، 10 امتیاز پاداش را درپی‌دارد. با توجه به این موارد، در زمان $T$-1 که در حالت $s$ هستیم، کار عقلانی آن است که بعلت داشتن وقت واکنش $a$ را انجام دهیم. زیرا با عبور از حالت $s'$، به طور میانگین 10 امتیاز می‌گیریم. ولی اگر در زمان $T$ در حالت $s$ باشیم، کار عقلانی آن است که زمان را از دست نداده و واکنش $b$ که به طور میانگین 1 امتیاز دارد، را انجام دهیم. زیرا امکان اخذ 10 امتیاز پاداش در این زمان ممکن نیست. ممکن است دانشجویان برای پاسخ این سوال به دنبال مثال‌هایی از دنیای حقیقی باشند، لذا می‌توان به این موارد اشاره نمود: انجام سرمایه‌گذاری که پرداخت پول و سود فراوانی پس از پایان عمر شما داشته باشد و یا امتحانی که در آن برای سوالی با بارم بالا، زمان پاسخگویی خیلی کمی لحاظ شده باشد. بنابراین می‌توان فرض کرد که برای هر حالت محیط، عنصر زمان نیز اضافه شود که البته این فرضیه هیچ تأثیری در پاسخ نمی‌گذارد. (اکنون واکنش‌ها هم به بخش زمانی و هم به بخش غیرزمانی حالات بستگی دارند). همچنین این فرضیه باعث می‌شود که کارگزار هیچگاه در موقعیت یکسانی از محیط قرار نگیرد. (به علت تفاوت زمان‌ها)

**؟ ۲.۲ می‌خواهیم عقلانیت چند تابع کارگزار مختلف برای جاروبرقی را بررسی کنیم. الف) نشان دهید که تابع کارگزار ساده‌ای که برای جارو برقی در شکل ۳.۲ توصیف شد، طبق فرضیات گفته شده در بخش ۲.۲ عقلانی است. ب) یک تابع برای کارگزار عقلانی توصیف کنید که در هر حرکت، معیار کارایی تغییر داده شده را یک واحد کم کند. آیا برنامه این کارگزار، به حالت داخلی(Internal state) نیازدارد؟ ج) در مورد طراحی‌های ممکن برای کارگزار، در حالتی که «مربع‌های تمیز می‌توانند دوباره کثیف شوند» و یا «جغرافیای محیط ناشناخته باشد»، بحث کنید. آیا این موضوع باعث می‌شود که کارگزار در این مورد از تجربیات شخصی‌اش یاد بگیرد؟ اگر این گونه باشد، چه چیز را بایستی یاد بگیرد؟ *(این تمرین مشابه تمرین ۴.۲ در ویرایش دوم است)***

☑ حل: دقت شود که در اینجا با توجه به فرضیات ساده‌سازی محیط، هیچ نگرانی در مورد حالت‌های غیرقطعی نداریم. **الف)** برای این منظور فقط کافی است نشان دهیم که به ازای تمام حالت‌های ممکن، (مانند انواع حالات کثیفی‌ها و موقعیت‌های مختلف شروع) این کارگزار می‌تواند خانه‌ها را حداقل با سرعت سایر کارگزارهای موجود، تمیز کند زیرا در اینصورت کارگزار عقلانی نامیده می‌شود. این موضوع در صورت عدم وجود کثیفی، مسلما درست است. حال برای وضعیتی که محل شروع کثیف باشد ولی هیچ خانه دیگری کثیف نباشد کارگزار با انجام یک مرحله، محیط را پاکیزه می‌کند و هیچ کارگزاری نمی‌تواند بهتر از این عمل کند. همچنین در وضعیتی که محل شروع کثیف نباشد ولی خانه دیگر کثیف باشد، پس از گذشت دو مرحله محیط پاکیزه می‌گردد و هیچ کارگزاری نمی‌تواند بهتر از این عمل کند. و در نهایت اگر هر دو خانه کثیف باشند، پس از سه مرحله، دنیا پاکیزه می‌گردد و هیچ کارگزاری نمی‌تواند بهتر از این عمل کند. **ب)** کارگزار گفته‌شده در قسمت الف، مدام به جلو و عقب حرکت می‌کند. حتی اگر دنیا تمیز باشد ولی بهتر است در صورت تمیزی دنیا، کارگزار هیچ عملیاتی انجام ندهد. از آنجا که ادراکات کارگزار اطلاعاتی در مورد تمیز بودن سایر خانه‌ها نمی‌دهد، به نظر می‌رسد که کارگزار بایستی دارای حافظه باشد تا بتواند از تمیزی سایر خانه‌ها نیز مطلع باشد. ایجاد دقیق این ویژگی بسیار دشوار است ولی شاید کارگزار بتواند با گذاشتن اشیاء و نمادهایی برای خود این موضوع را متوجه شود. به عنوان مثال در صورت قرار گرفتن در خانه چپ تمیز، بفهمد که خانه سمت راست نیز تمیز است؟ برای این منظور یک کارگزار بایستی از خود محیط به عنوان نوعی حافظه خارجی( External memory) کمک بگیرد. این تکنیک استفاده از اشیاء به صورت مشابه در انسان‌ها نیز وجود دارد. مثلا برای یادآوری قرار ملاقات به جای استفاده از تقویم، دستمال گردن خود را گره می‌زنیم. فرض کنید کارگزار می‌تواند واکنش‌های (تمیزکردن،A) و (تمیزکردن،B) را انجام دهد. اگر از کارگزار بخواهیم پس از تمیز کردن هر خانه هیچ عملیاتی انجام ندهد، آنگاه در موردی که سایر خانه‌ها کثیف باشند، کارگزار شکست می‌خورد و اگر از وی بخواهیم همواره فعال بوده، باز محکوم به حرکت بیهوده بین خانه‌ها خواهد شد. در حالت کلی این مشکل از آنجا نشأت می‌گیرد که کارگزارهای واکنشی بایستی در موقعیت‌هایی که شبیه به هم به نظر می‌رسند، کار یکسانی انجام دهند، حتی اگر موقعیت‌ها در واقعیت متفاوت از هم باشند. در دنیای جارو

رقی(Vacuum-cleaner world) این امر مشکل بزرگی محسوب می‌شود. زیرا وقتی در خانه جاری قرار داریم در نظر کارگزار مربع دیگر هم شبیه به خانه‌ای کثیف است و هم تمیز. لذا باید به آنجا برود تا به حقیقت واقعه پی ببرد. **ج)** اگر برای جاروبرقی، طول عمری طولانی در نظر بگیریم، واضح است که یادگیری یک نقشه(Map) جهت مراجعه به آن بسیار مفید خواهد بود. زیرا می‌تواند از خوردن کارگزار به دیوار جلوگیری کند. همچنین کارگزار می‌آموزد که یک مکان کثیف با عملیات جمع‌آوری همراه بوده و می‌تواند یک استراتژی جهت سرکشی بهینه تدبیر نماید. جزئیات مختصری از متدهای اکتشافی برای ساخت یک نقشه کامل در فصل ۴ آورده شده است.

**؟ ۳.۲ کدام یک از جملات زیر صحیح و کدام یک غلط می‌باشد؟ در صورت صحت جملات، مثالی بیاورید وگرنه مثالی نقض ارائه دهید. الف) کارگزاری که فقط می‌تواند قسمتی از حالت محیط را درک کند، کاملا عقلانی نیست. ب) محیط وظیفه‌ای وجود دارد که در آن هیچ کارگزار واکنشی خالصی نمی‌تواند عاقلانه رفتار کند. ج) محیط وظیفه‌ای وجود دارد که در آن هر کارگزاری عاقلانه عمل می‌کند. د) ورودی یک برنامه کارگزار همان ورودی تابع کارگزار است. هـ) هر تابع کارگزار از ترکیب ماشین و برنامه پیاده‌سازی شده است. و) اگر یک عامل همواره واکنش خود را به طور تصادفی از بین واکنش‌های ممکن انتخاب کند، یک محیط وظیفه قطعی وجود دارد که این عامل در آن به صورت عقلانی عمل می‌کند. ز) ممکن است یک کارگزار در دو محیط وظیفه جداگانه، به طور کامل عقلانی عمل کند. ح) هر کارگزار در یک محیط غیرمشاهده‌پذیر، عقلانی است. ط) یک کارگزار عقلانی که بازیکن پوکر است، هرگز نمی‌بازد.**

☑ حل: **الف)** غلط است. زیرا عقلانیت کامل بدین معنی است که کارگزار بتواند با توجه به اطلاعات دریافتی از حسگرهایش، بهترین تصمیم را بگیرد. **ب)** صحیح است. زیرا یک کارگزار واکنشی خالص، مشاهدات قبلی خود را نادیده می‌گیرد. بنابراین در یک محیط نیمه‌مشاهده‌پذیر، نمی‌تواند تخمینی بهینه از حالت داشته باشد. به عنوان مثال، در شطرنج مکاتبه‌ای، هر بازیکن حرکت موردنظر خود را ارسال کرده و در صفحه مشاهده می‌نماید. در این حالت اگر حرکت سایر بازیکنان به طور یکجا در مشاهده فعلی از صفحه لحاظ شده باشد، آنگاه کارگزار واکنشی نمی‌تواند وضعیت بازی در صفحه شطرنج را پیگیری کند. لذا مجبور است متفاوت با بازی قبل خود، عمل کند. **ج)** صحیح است. به عنوان مثال محیطی با یک حالت را در نظر بگیرید که تمام واکنش‌ها، دارای پاداش یکسانی هستند. بنابراین انتخاب واکنش‌ها در میزان پاداش هیچ تأثیری نخواهد داشت. در حالت کلی‌تر، هر محیطی که در آن پاداش مستقل از واکنش باشد، این ویژگی وجود دارد. **د)** غلط است. زیرا تابع کارگزار ورودی خود را از کل دنباله مشاهدات می‌گیرد ولی برنامه کارگزار فقط مشاهده فعلی را به عنوان ورودی درنظر می‌گیرد. **هـ)** غلط است. به عنوان مثال می‌توان محیطی شامل ماشین تورینگ و نوارهای ورودی در نظر گرفت که کار عامل، حل مسائل غیرمداوم باشد. در اینجا تابع کارگزاری وجود دارد که جواب صحیح را بیابد، ولی نمی‌توان هیچ برنامه کارگزاری برای پیاده‌سازی آن بیان نمود. به عنوان مثالی دیگر برای مسائل لجوج با سایز دلخواه، می‌توان تابع کارگزاری یافت که در زمان ثابت آنها را حل کند. **و)** صحیح است. این جمله نوعی خاص از جمله قسمت ج است. یعنی اگر واکنش انتخابی شما تاثیری در نتیجه ندارد؛ انتخاب تصادفی نیز عقلانی است. **ز)** صحیح است. به عنوان مثال می‌توانیم بخشی از محیط که در یک راهکار بهینه هرگز قابل دسترسی نخواهد بود را به طور اختیاری تغییر دهیم. **ح)** غلط است. بعلت غیرمشاهده‌پذیربودن محیط، برخی از واکنش‌های ما احمقانه خواهد بود ولی ممکن است که در صورت داشتن مدلی از محیط، کارگزار این موضوع را بفهمد.) **ط)** غلط است زیرا با اینکه کارگزار بهترین بازی ممکن را ارائه می‌دهد ولی همواره این امکان وجود دارد که طرف مقابل دسته کارت بهتری داشته باشد، لذا منجر به باخت کارگزار می‌گردد. این اتفاق ممکن است در هر دست بعدی بازی نیز رخ دهد. پس بهتر است بگوییم که به طور میانگین، این کارگزار امکان بردن دارد.

**؟ ۴.۲ برای هریک از این موارد مشخصات PEAS از محیط وظیفه را با استفاده از اصطلاحاتی که در بخش ۲.۳.۲ ذکر شد، توصیف نمایید: الف) بازیکن فوتبالیست. ب) کاوشگر در زیر اقیانوس تیتان. ج) خریداری کتاب‌های هوش مصنوعی در اینترنت. د) بازیکن مسابقه تنیس. هـ) بازی کردن تنیس با دیوار. و) بالارفتن از ارتفاع. ز) جوابگوی آدم بد دهان. ح) پیشنهاد قیمت خرید در یک مزایده. *(این تمرین مشابه تمرین ۲.۵ در ویرایش دوم است)***

☑ حل: **الف)** نیمه مشاهده پذیر، اتفاقی، ترتیبی، دینامیک، پیوسته، چندکارگزاره **ب)** نیمه مشاهده پذیر، اتفاقی، ترتیبی، دینامیک، پیوسته، تک‌کارگزاره **ج)** نیمه مشاهده پذیر، قطعی، ترتیبی، استاتیک، گسسته، تک‌کارگزاره **د)** کاملا مشاهده پذیر، اتفاقی، مرحله ای (هر ضربه جداگانه است)، دینامیک، پیوسته، چندکارگزاره **هـ)** کاملا مشاهده پذیر، اتفاقی، مرحله ای، دینامیک، پیوسته، تک‌کارگزاره **و)** کاملا مشاهده پذیر، اتفاقی، ترتیبی، استاتیک، پیوسته، تک‌کارگزاره **ز)** کاملا مشاهده پذیر، قطعی، ترتیبی، استاتیک، پیوسته، تک‌کارگزاره **ح)** کاملا مشاهده پذیر، استراتژیک، ترتیبی، استاتیک، گسسته، چندکارگزاره

**؟ ۵.۲ این اصطلاحات را به زبان خود تعریف کنید: الف) کارگزار ب) تابع کارگزار(Agent function) ج) برنامه کارگزار(Agent rogram) د) عقلانیت(Rationality) هـ) خودمختاری(Autonomy) و) کارگزار واکنشی(Reflex agent) ز) کارگزار مبتنی بر مدل(Model-based agent) ح) کارگزار مبتنی بر هدف(Goal-based agent) ط) کارگزار مبتنی بر سودمندی(Utility-based agent) ی) کارگزار یادگیرنده(Learning agent) *(این تمرین مشابه تمرین ۲.۱ در ویرایش دوم است)***

☑ حل: برای این اصطلاحات تعاریف ممکن زیادی وجود دارد که برخی از آنها در اینجا آورده شده است. **الف)** کارگزار: روبات یا نرم‌افزاری است که با درک و مشاهده محیط پیرامون خود، عکس‌العملی را انجام می‌دهد. بعبارت دیگر همواره در حال مشاهده و

انجام عکس‌العمل می‌باشد. اساسا نکته کلیدی در تعریف هر کارگزار، در نحوه پیاده‌سازی تابع کارگزار آن نهفته است یعنی اینکه آن روبات قرار است در محیط چه وظیفه‌ای را برعهده گیرد. (توجه: برخی نویسندگان به جای اصطلاح تابع از «برنامه» استفاده نموده‌اند که منظور از آن قطعه کدهایی است که بر روی ماشین‌ها و شبکه‌های مختلف قابلیت اجرا داشته و عملکرد انسان را بر عهده بگیرند مانند کارگزارهای موبایل). **ب)** تابع کارگزار: تابعی است که نشان می‌دهد کارگزار در قبال هرگونه دنباله مشاهدات ممکن، چه کاری را باید انجام دهد. این تابع می‌تواند به صورت‌های مختلف مانند جدول و یا متن بیان شود. به عنوان مثال برای کارگزار جاروبرقی می‌توان جدولی تشکیل داد که هر سطر آن یک حالت محیط مانند «اتاق چپ کثیف است» را نشان دهد که در مقابل آن سطر کاری که کارگزار باید انجام دهد را می‌نویسیم مثلا در این مورد کارگزار باید «تمیز کردن اتاق» را انجام دهد. کل جدول دربرگیرنده تمام حالات دنیا است تا عملکرد کارگزار را به طور کامل تعیین کرده باشیم. **ج)** برنامه کارگزار: برنامه‌ای کامپیوتری است که بر مبنای معماری کارگزار نوشته شده است و تابع کارگزاری که تاکنون به صورت جدولی یا متنی بوده است را در کامپیوتر پیاده‌سازی می‌کند. در یک طراحی ساده می‌توان گفت این برنامه، برنامه‌ای است که یک مشاهده جدید رخ داده را گرفته و واکنش مناسب برای آن را برمی‌گرداند. **د)** عقلانیت: یک ویژگی کارگزار است و به معنی آن است که کارگزار عاقلانه رفتار کند یعنی با دریافت مشاهدات(Percepts) در طی زمان، واکنشی را برای انجام انتخاب کند که به ازای انجام آن، میانگین سودمندی‌اش بیشینه شود. **هـ)** خودمختاری: یک ویژگی کارگزار است و به معنی آن است که رفتار کارگزار فقط توسط تجربیات شخصی‌اش تعیین می‌شود تا آنکه صرفا توسط برنامه‌های اولیه(Initial programming) که برنامه‌نویس درون آن قرار می‌دهد. **و)** کارگزار واکنشی: کارگزاری است که واکنش آن در محیط، تنها به مشاهدات فعلی‌اش از محیط بستگی دارد. **ز)** کارگزار مبتنی بر مدل: کارگزاری است که واکنش آن مستقیما از یک مدل درونی مشتق می‌گردد که این مدل، توصیفی از موقعیت جهان فعلی است. این مدل در طی زمان بروزرسانی می‌شود. **ح)** کارگزار مبتنی بر هدف: این کارگزار، واکنشی را از بین موارد ممکن انتخاب می‌کند که آشکارا وی را به هدف مربوطه برساند. **ط)** کارگزار مبتنی بر سودمندی: این کارگزار، واکنشی را از بین موارد موجود انتخاب می‌کند که در موفقیت‌های آتی، میانگین سودمندی وی را بیشینه کند. **ی)** کارگزار یادگیرنده: کارگزاری است که در طی زمان و بر مبنای تجربیات شخصی‌اش، چیزهایی یاد گرفته و رفتار خود را بهبود می‌بخشد. هر یک از موارد فوق کارگزاری متفاوت را نشان می‌دهد. مانند انسان‌ها که می‌توانند در زندگی خود و انجام کارهایشان یا براساس عقل، تجربه، هدف و یا سودمندی رفتار کنند. به عنوان نمونه یک پزشک در عمل جراحی یا یک دونده در طی مسابقه برای انتخاب حرکت بعدی خود ایده‌ای متفاوت دارند. البته ممکن است یک کارگزار در هر زمان و با توجه به موقعیت، ترکیبی از این رفتارها را از خود نشان دهد. ولی مسلم است که انتخاب کارگزار مناسب برای هر محیط و شرایط در رسیدن به نتیجه بسیار تاثیرگذار است زیرا در بسیاری موارد شاید عاقلانه رفتارکردن نیاز نباشد و در برخی دیگر برعکس.

**؟ ۶.۲ این تمرین تفاوت بین تابع کارگزار و برنامه کارگزار را مورد بررسی قرار می‌دهد. الف) آیا برای یک تابع کارگزار داده شده، می‌توان بیش از یک برنامه کارگزار پیاده‌سازی نمود؟یا یک مثال ذکر کنید و یا آنکه نشان دهید چرا ذکر مثال ممکن نیست؟ ب) آیا تابع کارگزاری وجود دارد که نتواند توسط هیچ برنامه کارگزاری پیاده‌سازی شود؟ ج) آیا یک برنامه کارگزار به ازای معماری ماشین مشخص، دقیقا یک تابع کارگزار را پیاده‌سازی می‌کند؟ د) یک معماری با n بیت فضا داده شده است. ممکن است چند برنامه کارگزار متفاوت در آن وجود داشته باشد؟ هـ) فرض کنید یک برنامه کارگزار مشخص را بر روی ماشینی با دو برابر سرعت اجرا کنیم. آیا تغییری در تابع کارگزار رخ می‌دهد؟** ***(این تمرین مشابه تمرین ۳.۲ در ویرایش دوم است)***

☑ حل: اگرچه این سوالات بسیار ساده به نظر می‌رسند ولی حاوی نتایج بنیادی فراوانی هستند. در پاسخ‌های این بخش، فرض کرده‌ایم که طراحی کارگزار ساده بوده و در محیط‌های استاتیک عمل می‌کند که این فرضیه تا زمانی که کارگزار اقدام به کنکاشی عمدی نکند، هیچ مشکلی ایجاد نمی‌کند. البته نتایج برای محیط‌های پویا جذاب‌تر بوده و حاوی مسائلی متفاوت می‌باشد. **الف)** بله. به عنوان مثال یک برنامه کارگزار را گرفته و در بخش‌هایی از آن عبارت تهی (مانند Enter و فاصله) قرار دهید به طوری که هیچ تأثیری در خروجی نگذارد. بنابراین برنامه‌های متفاوتی با همان عملکرد بدست می‌آید (تابع عملکرد یکسان و برنامه متفاوت). به عنوان مثالی دیگر فرض کنید از تعداد دانشجو بخواهیم تا برنامه‌ای به زبان C++ برای محاسبه فاکتوریل یک عدد بنویسند، مسلم است که برنامه هر دانشجو با دیگری متفاوت خواهد بود. ممکن است فردی آن را با دستور For و دیگری با دستور while و یا do-while بنویسد. همچنین نام متغیرها و تعداد خطوط هر برنامه با دیگری متفاوت است. **ب)** بله. تابع کارگزاری را در نظر بگیرید که عملکرد تست تورینگ را انجام می‌دهد و پس از مشاهده محیط و در صورت توقف، کلمه True و در غیراینصورت کلمه False را چاپ می‌کند. این تابع توسط هیچ برنامه کارگزاری قابل پیاده‌سازی نیست زیرا اگر اینطور بود تست تورینگ توسط هر روباتی انجام شده و روبات‌ها به سادگی می‌توانستند با فریب انسان‌ها بر آنها غالب شوند. (توجه: در محیط‌های دینامیک و برای ماشین‌هایی با سرعت کمتر از بی‌نهایت، پیاده‌سازی تابع کارگزار عقلانی ممکن نیست. به عنوان مثال پیاده‌سازی تابع کارگزاری که همواره در بازی‌هایی مانند شطرنج حرکت برنده را انجام دهد میسر نیست). **ج)** بله. رفتار یک کارگزار توسط معماری و برنامه آن مشخص می‌باشد. **د)** حداکثر $2^n$ برنامه کارگزار وجود دارد. اگرچه بسیاری از آنها هرگز اجرا نخواهند شد. (توجه: از آنجا که در کل $n$ بیت داریم پس هر برنامه حداکثر به $n$ بیت از فضا برای ذخیره‌سازی نیازمند است. بنابراین وضعیت داخلی آن می‌تواند از بین فقط $2^n$ تاریخچه گذشته مشخص و متمایز شود. از آنجا که تابع کارگزار، نشان‌دهنده واکنش بر مبنای تاریخچه مشاهدات است، ممکن است توابع کارگزار بسیاری وجود داشته باشند که به

خاطر فقدان حافظه در ماشین، نتوانند پیاده‌سازی شوند). هـ) این موضوع به برنامه و محیط بستگی دارد. اگر محیط دینامیک باشد، ممکن است افزایش سرعت ماشین در انتخاب واکنش تاثیر بگذارد (شاید انتخاب بهتری انجام دهیم). ولی در صورت استاتیک بودن محیط و عدم تاثیر برنامه از گذر زمان، تابع کارگزار هیچ تغییری نمی‌کند.

**؟ ۷.۲ برنامه‌های کارگزاری برای عامل‌های مبتنی بر هدف و مبتنی بر سودمندی بنویسید.**

☑ حل: طراحی کارگزارهای مبتنی بر هدف و مبتنی بر سودمندی به ساختار محیط بستگی دارد. به عنوان مثال، از ساده‌ترین طراحی‌های ممکن که در فصل‌های ۳ و ۱۰ گفته شد، آن است که کارگزار در ابتدا تمام دنباله واکنش‌های آتی که به نفعش باشد را محاسبه نموده و سپس اقدام به انجام آنها کند. این راهکار در محیط‌های استاتیک و قطعی که قابل‌مشاهده نباشند، مناسب است. ولی برای محیط‌های استاتیک که کاملا مشاهده‌پذیر هستند، می‌توان کارگزار را به گونه‌ای طراحی نمود که در هر حالت، واکنشی محاسبه و انجام دهد که به نفعش باشد. حال اگر کارگزار در محیطی نیمه‌مشاهده‌پذیر قرار گیرد، می‌تواند یک نقشه مشروط محاسبه کند که در آن، دنباله واکنش‌ها بر حسب تابعی از ادراکات کارگزار مشخص شده باشد. به طور خلاصه، این نقشه مشروط می‌تواند به ازای هر اتفاق رخ داده، واکنش موردنظر برای انجام را پاسخ دهد. بنابراین به نوعی نمایش‌دهنده کل تابع کارگزار می‌باشد. از آنجا که معمولا فقط به فکر کسب سود بودن در محاسبه هر چیز، کاری لجوجانه و هزینه‌بر است، بهتر است به جای یک نقشه مشروط، فقط دنباله‌ای از واکنش‌ها را محاسبه کنیم که موجب دستیابی کارگزار به هدف شود و در کنار آن محیط را تحت کنترل قرار دهیم تا عوامل موفقیت این نقشه را چک کرده و در صورت مهیا نبودن آنها، شرایط را فراهم کنیم تا نقشه با موفقیت اجرا شود. همچنین بهتر است فقط قبل از اولین واکنش، نقطه آغاز نقشه را محاسبه نماییم و در مراحل بعد نقشه را ادامه دهیم. قطعه کد کارگزار ساده مبتنی بر هدف را در شکل ۱.۲S مشاهده می‌کنید. عبارت GOAL-ACHIEVED وضعیت فعلی را جهت هدف یا غیرهدف بودن بررسی می‌کند و در صورت غیرهدف بودن کار خاصی انجام نمی‌دهد.

```
function GOAL-BASED-AGENT(percept) returns an action
  persistent: state, the agent's current conception of the world state
              model, a description of how the next state depends on current state and action
              goal, a description of the desired goal state
              plan, a sequence of actions to take, initially empty
              action, the most recent action, initially none

  state ← UPDATE-STATE(state, action, percept, model)
  if GOAL-ACHIEVED(state,goal) then return a null action
  if plan is empty then
plan ← PLAN(state,goal,model)
  action ← FIRST(plan)
  plan ← REST(plan)
  return action
```

شکل ۱.۲s یک کارگزار مبتنی بر هدف

همچنین عبارت PLAN در این قطعه کد، دنباله واکنش‌ها به منظور دستیابی به هدف را محاسبه می‌کند. ممکن است در این بخش، فقط قسمت ابتدایی از نقشه کامل یافته شود و مابقی نقشه پس از اجرای همین قسمت ابتدایی بدست می‌آید. این کارگزار در راه رسیدن به هدف عمل می‌کند یعنی اگر در هر نقطه به هدف نرسیده باشد نقشه خود را دوباره بازبینی و اصلاح می‌کند تا دوباره به هدف برسد. در این سطح از تعریف، که وارد جزئیات نشده‌ایم، تفاوت زیادی بین کارگزار مبتنی بر هدف و مبتنی بر سودمندی مشاهده نمی‌شود. قطعه کد مربوط در شکل ۲.۲S آورده شده است.

```
function UTILITY-BASED-AGENT(percept) returns an action
  persistent: state, the agent's current conception of the world state
              model, a description of how the next state depends on current state and action
              utility − function, a description of the agent's utility function
              plan, a sequence of actions to take, initially empty
              action, the most recent action, initially none

state ← UPDATE-STATE(state, action, percept, model)
if plan is empty then
plan ← PLAN(state,utility − function,model)
action ← FIRST(plan)
plan ← REST(plan)
return action
```

شکل ۲.۲s یک کارگزار مبتنی بر سودمندی

◆ *در تمرینات بعدی، پیاده‌سازی محیط‌های دنیای جاروبرقی را مورد بررسی قرار می‌دهیم.*

**؟ ۸.۲ برای دنیای جاروبرقی شکل ۲.۲ که مشخصات آن در بخش ۱.۲ بیان شد، یک شبیه‌ساز محیط( Environment simulator) برای اندازه‌گیری معیار کارایی پیاده‌سازی نمایید. پیاده‌سازی شما بایستی ماژولار باشد تا حسگرها، محرک‌ها و مشخصات محیطی (سایز، شکل، مکان کثیفی و. . .) به سهولت قابل‌تغییر باشند.** ***(این تمرین مشابه تمرین ۷.۲ در ویرایش دوم است)***

☑ حل: می‌توانید با اندکی جستجو، کدی برای پیاده‌سازی محیط جاروبرقی به زبان لیسپ بیابید. سپس با ایجاد تغییراتی در آن نظیر شکل اتاقها و موانع، به نتیجه دلخواه خود دست یابید.

**؟ ۹.۲ یک کارگزار واکنشی‌ساده برای محیط جاروبرقی در تمرین ۸.۲ پیاده‌سازی کنید. شبیه‌ساز محیط را به ازای این کارگزار و برای تمام پیکره‌بندی‌های ممکن، یعنی تمام حالات شروع ممکن از جهت کثیفی و مکان کارگزار اجرا کنید. سپس به ازای هر پیکره‌بندی امتیاز کارایی این کارگزار را ثبت کرده و در نهایت میانگین کل امتیاز وی را محاسبه کنید.** ***(این تمرین مشابه تمرین ۸.۲ در ویرایش دوم است)***

☑ حل: یک برنامه کارگزار واکنشی که تابع کارگزار عقلانی را پیاده‌سازی می‌کند و در این فصل تشریح شد در زیر آورده شده است:

```
(defun reflex-rational-vacuum-agent (percept)
  (destructuring-bind (location status) percept
    (cond ((eq status 'Dirty) 'Suck)
          ((eq location 'A) 'Right)
          (t 'Left))))
```

برای وضعیت‌های 1 و 3 و 5 و7 در شکل ۲۰.۳، معیار کارایی به ترتیب عبارتست از: 1996، 1999، 1998، 2000

**؟ ۱۰.۲ نسخه تغییریافته‌ای از محیط جارو برقی تمرین ۸.۲ را در نظر بگیرید که در آن کارگزار به ازای هر حرکت، یک جریمه می‌شود. الف)آیا یک کارگزار واکنشی ساده می‌تواند در این محیط به طور کاملا عقلانی عمل کند. توضیح دهید. ب) درمورد یک کارگزار واکنشی با حالت درونی چطور؟ این کارگزار را طراحی نمایید. ج) اگر مشاهدات کارگزار به گونه‌ای باشد که وضعیت کثیفی/ تمیزی خانه‌ها در محیط را درک نماید، پاسخ‌های شما در قسمت‌های الف و ب چه تغییری می‌کند؟** ***(این تمرین مشابه تمرین ۹.۲ در ویرایش دوم است)***

☑ حل: **الف)** خیر، به پاسخ قسمت ب در تمرین ۴.۲ مراجعه شود. **ب)** به پاسخ قسمت ب در تمرین ۴.۲ مراجعه شود. **ج)** در این مورد، یک کارگزار واکنشی ساده می‌تواند به طور کاملا عقلانی عمل نماید. این کارگزار از جدولی با 8 ورودی تشکیل شده است که به ازای هر مشاهده، عملیات ایندکس[5] را انجام داده و واکنش مربوط به آن حالت، مشخص و استخراج می‌شود. پس از انجام عکس‌العمل کارگزار جهان به روز شده و وقوع مشاهده بعدی، به کارگزار می‌گوید که چه کاری را باید انجام دهد. در محیط‌های بزرگتر، ساخت چنین جدولی غیرممکن است ولی در عوض کارگزار می‌تواند از الگوریتم‌های جستجوی بهینه فصل‌های ۳ و ۴ استفاده نماید تا بتواند اولین گام از دنباله راه حل را اجرا نماید. دراینجا نیز هیچ حالت درونی نیاز نمی‌باشد ولی اگر حالت درونی داشته باشیم می‌توانیم دنباله راه‌حل را ذخیره کنیم تا در هر مشاهده جدید، به جای محاسبه مجدد از همان استفاده کنیم.

**؟ ۱۱.۲ نسخه تغییر یافته‌ای از محیط جارو برقی تمرین ۸.۲ را در نظر بگیرید که در آن جغرافیای محیط (وسعت، مرز و موانع) ناشناخته باشند و همچنین در حالت شروع هیچ اطلاعی از وضعیت کثیفی نداشته باشیم. (کارگزار می‌تواند در جهات بالا، پایین، چپ و راست حرکت نماید.) الف) آیا یک کارگزار واکنشی ساده می‌تواند در این محیط به طور کاملا عقلانی عمل کند؟ توضیح دهید. ب) آیا یک کارگزار واکنشی ساده که دارای تابع کارگزاری تصادفی است، می‌تواند بهتر ازیک کارگزار واکنشی ساده عمل نماید؟ چنین کارگزاری را طراحی نموده و کارایی آنرا در محیط‌های مختلف اندازه‌گیری نمایید. ج) آیا می‌توانید محیطی طراحی کنید که کارگزار تصادفی طراحی شده توسط شما، در آن عملکرد بسیار ضعیفی داشته باشد؟ نتایج خود را بیان کنید. د) آیا یک کارگزار واکنشی با حالت درونی، بهتر از یک کارگزار واکنشی ساده عمل می‌کند؟ کارگزاری طراحی نموده و کارایی آنرا در محیط‌های مختلف اندازه‌گیری کنید. آیا می‌توانید برای چنین نوعی، کارگزار عقلانی طراحی کنید؟** ***(این تمرین مشابه تمرین ۱۰.۲ در ویرایش دوم است)***

☑ حل: **الف)** بعلت آنکه کارگزار هیچ اطلاعی از جغرافیا، درک مکانی و مشاهده کثیفی ندارد و همچنین نمی‌تواند اتفاقات رخ داده را به خاطر آورد، لذا هر زمان که کارگزار سعی در حرکت به یک جهت نماید و آن جهت ممکن نباشد، مدام به دیوار برخورد

---

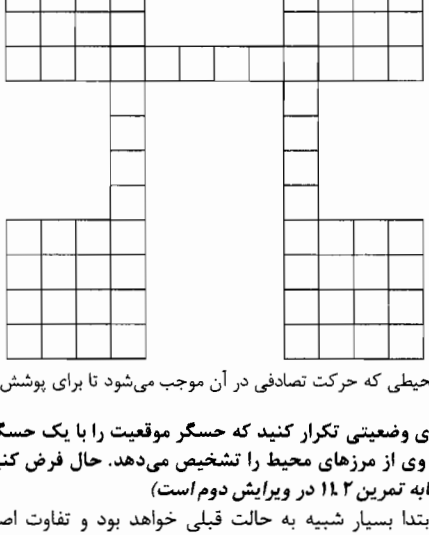
[5] عملیات ایندکس شیوه‌ای برای یافتن سطری خاص از جداول بزرگ است که خود دارای الگوریتم‌هایی می‌باشد که به ازای یک ورودی مشخص، سطر خروجی را برمی‌گرداند.

خواهد نمود ولی اگر به طور اتفاقی حرکت کند، اینگونه نخواهد شد. **ب)** یک طراحی ممکن برای کارگزار آن است که کثیفی‌ها را تمیز کرده و در غیراینصورت به طور تصادفی حرکت نماید.



```
(defun randomized-reflex-vacuum-agent (percept)
  (destructuring-bind (location status)
    percept (cond ((eq status 'Dirty) 'Suck)
                  (t (random-element '(Left Right Up Down))))))
```

این طراحی بسیار شبیه به عملکرد جاروبرقی Roomba™ است. (البته Roomba دارای حسگر ضربه نیز بود و تنها زمانی حرکت تصادفی انجام می‌داد که به مانعی برخورد نماید.) در اینصورت کارگزار در محیط‌های کوچک و خوب، به طور معقولی عملکرد مناسب خواهد داشت ولی در محیط‌هایی شبیه Maze یا محیط‌هایی با رویدادهای مختلف، پوشش تمام خانه‌ها بسیار زمانبر خواهد شد. **ج)** مثالی از یک محیط را در شکل ۳.۲S مشاهده می‌کنید. البته دانشجویان علاقه‌مند می‌توانند در صورت تمایل، زمان تمیزکردن را در محیط‌های مربعی و خطی با سایزهای متفاوت اندازه گرفته و سپس آنرا با کارایی الگوریتم‌های جستجوی محلی که در فصل ۴ بیان شده است، مقایسه کنند. **د)** یک کارگزار واکنشی با حالت درونی، می‌تواند یک نقشه بسازد (برای کسب اطلاعات بیشتر به فصل ۴ مراجعه شود). یک جستجوی آنلاین اول عمق، می‌تواند در یک زمان خطی بر حسب اندازه محیط، به هر حالتی دستیابی پیدا کند، بنابراین کارگزار می‌تواند خیلی بهتر از یک کارگزار واکنشی ساده عمل نماید. داشتن رفتاری عقلانی در محیط‌های ناشناخته، موضوعی پیچیده است ولی معمولا علاقه‌مندان زیادی برای تفکر در مورد آن وجود دارد. برای این موضوع نیاز است تا نگاهی به توزیع اولیه احتمالاتی بر روی انواع محیط‌ها داشته باشیم که به آن حالت عقیده اولیه(Belief state) می‌گوییم. هر واکنشی که منجر به درک جدیدی گردد و برای آپدیت این توزیع استفاده شود، باعث حرکت کارگزار به یک حالت عقیده جدید می‌شود. بدلیل پیچیدگی محیط، حالات عقیده در یک محیط ساده فرض می‌شوند. لذا مشکل اکتشاف بهینه، همان مسأله جستجو برای استراتژی بهینه در فضای حالت عقیده می‌باشد که با وجود لجاجت مسئله ای خوش‌تعریف است. و به عنوان مثال می‌توان به بازی رایانه‌ای Minesweeper اشاره‌نمود. (تمرین ۲۲.۷ را ببینید) در محیط‌های کوچک این بازی، اکتشاف بهینه ممکن است عملی باشد ولی مرحله آپدیت حالات عقیده نیاز به شرح فراوان دارد.



شکل ۳.۲s محیطی که حرکت تصادفی در آن موجب می‌شود تا برای پوشش تمام خانه‌ها زمان زیادی صرف شود.

**؟ ۱۲.۲ تمرین ۱۱.۲ را برای وضعیتی تکرار کنید که حسگر موقعیت را با یک حسگر ضربه جایگزین کنیم که این حسگر برخورد کارگزار به مانع و یا عبور وی از مرزهای محیط را تشخیص می‌دهد. حال فرض کنید این حسگر ضربه از کار بیافتد. کارگزار چه باید کند؟** ***(این تمرین مشابه تمرین ۲.۱۱ در ویرایش دوم است)***

☑ حل: این مسئله در ابتدا بسیار شبیه به حالت قبلی خواهد بود و تفاوت اصلی آن در این است که به جای درک دقیق موقعیت مکانی در ساخت نقشه، کارگزار بایستی موقعیت خود را حدس بزند. (منظور همان گره‌ها در ساختمان داده معرف گراف فضای حالت هستند.) زمانی که یک ضربه تشخیص داده شود، کارگزار با فرض عدم‌تغییر این وضعیت، در نقشه خود یک دیوار را ثبت می‌کند. در محیط‌های خط‌کشی شده(Grid environment) کارگزار می‌تواند موقعیت $(x,y)$ خود را پیگیری نموده و بنابراین قادر است رسیدن به یک حالت قدیمی را تشخیص دهد. ولی در حالت کلی تشخیص اینکه یک حالت جدید است یا قدیمی، چندان ساده نیست.

**؟ ۱۳.۲ محیط‌های جارو برقی بیان شده در تمرین‌های قبل، همگی قطعی بودند. اکنون به ازای هر یک از نسخه‌های اتفاقی زیر، برنامه‌های کارگزار ممکن برای آنها را مورد بحث قرار دهید. الف) قانون مورفی(Murphy`s low): بیست وپنج درصد مواقع، عملیات مکیدن منجر می‌شود که در صورت کثیفی کف اتاق تمیز نشود و یا یک کف اتاق تمیز، با خالی کردن آشغال‌ها کثیف شود. اگر سنسور کثیفی در 10٪ مواقع پاسخ اشتباهی دهد، چه تأثیری روی برنامه کارگزار شما خواهد گذاشت؟ ب)بچه‌های کوچک: در هر مرحله زمانی، هر خانه تمیز 10٪ شانس کثیف شدن دارد. آیا می‌توانید برای این مورد یک کارگزار عقلانی طراحی نمایید؟** ***(این تمرین مشابه تمرین ۲.۱۲ در ویرایش دوم است)***

☑ حل: **الف)** برای یک کارگزار واکنشی این موضوع هیچ مشکل جدیدی ایجاد نمی‌کند زیرا این کارگزار مادامی که مکان فعلی کثیف باشد به عمل مکیدن آشغال‌ها ادامه می‌دهد. ولی برای کارگزاری که از دنباله مشاهدات خود یک نقشه می‌سازد، بایستی بجای عمل «مکیدن»، عمل « مکیدن تا زمان تمیز شدن» را جایگزین کنیم. اگر حسگر کثیفی در **هر** مرحله امکان اشتباه داشته باشد، آنگاه کارگزار بایستی قبل از آنکه تصمیم بگیرد عمل مکیدن یا حرکت به خانه بعد را انجام دهد، برای چندین مرحله منتظر بماند تا عملکردی قابل قبول‌تر داشته باشد. پس پر واضح است که بایستی شرایط را سبک‌وسنگین کنیم زیرا اگر زمان خیلی زیادی منتظر بمانیم، کثیفی‌ها نیز به مدت طولانی در کف اتاق باقی می‌ماند. (که کاری مستحق مجازات است) و از طرف دیگر واکنش قوری نیز دارای ریسک است زیرا موجب کثیف کردن یک اتاق تمیز یا نادیده گرفتن یک اتاق کثیف خواهد شد (بخاطر عملکرد اشتباه سنسور). بنابراین کار عاقلانه آنست که کارگزاری عقلانی مدام بین خانه‌ها حرکت و سرکشی کرده و کثیفی آنها را چک کند حتی اگر قبلا آن خانه بازدید شده باشد. (زیرا ممکن است حسگر در آن زمان جواب اشتباهی داده باشد.) البته نمی‌توان برای هر سرکشی میزان زمان توقف هر خانه را به طور دقیق تعیین نمود ولی با توجه به تجربه به طور ریاضی قابل حدس است. این مسئله بخشی از فرآیندهای تصمیم‌گیری مارکوف(Markov decision process) را شامل می‌شود. برخی مسائل در حالت کلی دشوارند. ولی ممکن است در برخی موارد خاص از آنها بتوان به آنالیزهای دقیقی دست یافت. **ب)** در این مورد کارگزار بایستی به طور نامحدود بین خانه‌ها سرکشی کند. احتمال آنکه یک مربع که قبلا تمیز بوده، اکنون کثیف باشد در طی گذشت زمان افزایش می‌یابد. بنابراین راهکار عقلانی آنست که به طور مکرر کوتاهترین سرکشی ممکن به تمام خانه‌ها را انجام دهیم (که به نوعی تکرار مکررات است زیرا ممکن است عواقبی چون سرکشی چند باره به یک خانه را درپی‌داشته باشد. البته به جغرافیای محیط نیز بستگی دارد). این مسئله نیز بخشی از فرآیندهای تصمیم‌گیری مارکوف محسوب می‌شود.

# فصل ۳ (ویرایش سوم)

**؟ ۱.۳ توضیح دهید چرا تدوین مسأله بایستی پس از تدوین هدف صورت پذیرد؟** *(این تمرین مشابه تمرین ۲.۳ در ویرایش دوم است)*

☑ حل: در عملیات تدوین هدف، تعیین می‌کنیم کارگزار علاقه‌مند به رسیدن به کدام حالت از جهان می‌باشد، هدف او در سیستم چیست، از کدام حالت‌ها بایستی صرف نظر کرده و از کدام یک دوری بجوید. ولی در عملیات تدوین مسأله، تعیین می‌کنیم که کدام بخش‌های محیط مهم بوده و بایستی لحاظ شوند و کدام جنبه‌ها را نادیده بگیریم[۶]. اگر ما عملیات تدوین مسأله را در ابتدا و قبل از تدوین هدف انجام دهیم، دقیقا نمی‌دانیم که چه بخش‌هایی برای هر حالت باید در نظر گرفته شود و چه بخش‌هایی حذف شود و این امر موجب می‌شود تا در زمان یافتن راه‌حل مدام بین عملیات تدوین هدف، تدوین مسأله و حل‌مسأله سرگردان باشیم.

**؟ ۲.۳ فرض کنید هدف شما، هدایت روبات در یک محیط پرپیچ‌وخم(*maze*) باشد به‌طوری که روبات در لحظه شروع در مرکز این محیط و رو به سمت شمال قرار گرفته باشد. شما می‌توانید روبات را در یکی از جهات شمال، جنوب، شرق و غرب بچرخانید. همچنین می‌توانید آن را برای مسافتی مشخص به جلو حرکت داده و یا در صورت مواجهه با یک دیوار، آن را متوقف کنید. الف) این مسأله را تدوین و فرموله کنید. اندازه این فضای حالت چقدر است؟ ب) در مسیریابی یک محیط پرپیچ‌وخم، تنها جایی که نیاز به چرخش است، در مکان‌هایی است که دو یا چند راهرو تداخل دارند. اکنون با این دید، مجددا مسأله را تدوین کنید. اندازه این فضای حالت چقدر است؟ ج) زمانیکه کارگزار به یک نقطهٔ برگشت برسد یعنی تقاطعی که قبلا یکی از مسیرهای آن را انتخاب کرده و موفق نشده است و دوباره به آن رجوع کرده تا بتواند انتخاب دیگری داشته باشد، در این حالت می‌تواند در هر یک از چهار جهت اصلی بچرخد و این تنها واکنشی است که کارگزار می‌تواند انجام دهد. با توجه به این واکنش، مسأله را مجددا تدوین کنید؟ آیا کارگزار نیاز دارد تا دنباله چرخش‌های خود را نگهداری کند؟ د) در توصیف ابتدایی که از محیط داشتیم، با محدود کردن واکنش‌ها، مسأله را نسبت به دنیای حقیقی ساده‌تر کردیم، سه نمونه ساده‌سازی صورت گرفته را بیان کنید.**

☑ حل: **الف)** برای این محیط، یک سیستم مختصاتی تعریف می‌کنیم که مرکز محیط دارای مختصات (0,0) بوده و مختصات دو گوشهٔ اصلی محیط (-1,-1) و (1,1) باشد. حالت شروع: قرارگیری روبات در مختصات (0,0) و رو به شمال. آزمون هدف: اگر موقعیت جاری $(x,y)$ باشد، آن‌گاه در صورتی به هدف رسیده‌ایم که $|x|>1$ و یا $|y|>1$. تابع پسین: حرکت به جلو به اندازهٔ فاصلهٔ d، تغییر جهت روبات. تابع هزینه: کل مسافت پیموده شده. این فضای حالت به طور بی‌نهایت بزرگ است زیرا موقعیت کارگزار پیوسته است. **ب)** هر حالت می‌تواند شامل جهت کارگزار و تقاطعی که اکنون کارگزار در آن واقع است باشد. لذا با رسیدن کارگزار به پایان هر راهرو، به یک گره که آن را گره خروجی می‌نامیم، می‌رسد. البته مسلم است که برخی از گره‌ها در داخل محیط واقع شده‌اند و خروجی نیستند. حالت شروع: کارگزار در مرکز محیط و رو به شمال قرار دارد. آزمون هدف: رسیدن به یک گرهٔ خروجی. تابع پسین: حرکت به تقاطع بعدی که در مسیر جلویی کارگزار است، اگر تقاطع فقط یک گذر داشت به سمت آن می‌چرخد. تابع هزینه: کل مسافت پیموده شده. این محیط دارای $4n$ حالت است که $n$ همان تعداد تقاطع‌ها می‌باشد. **ج)** حالت شروع: قرارگیری کارگزار در مرکز محیط آزمون هدف: رسیدن به یک گرهٔ خروجی. تابع پسین: حرکت به سمت تقاطع بعدی در جهات شمال، جنوب، شرق و غرب. تابع هزینه: کل مسافت پیموده شده. در ضمن کارگزار نیازی به نگهداری دنبالهٔ چرخش‌های انجام دادهٔ خود ندارد زیرا هیچ کمکی در پیش‌بینی‌های بعدی کارگزار در جهات ندارد و همچنین در آزمون هدف نیز استفاده نمی‌شود. سیستم موتوری که قصد اجرای این مسأله را دارد نیازمند نگهداری جهت کنونی کارگزار می‌باشد تا بداند که کارگزار چه زمانی می‌چرخد. د) ساده‌سازی‌های صورت گرفته در حالات: (i) از قد روبات نسبت به زمین و میزان کجی آن صرف نظر شده است. (ii) روبات فقط می‌تواند در چهار جهت اصلی چرخش کند. (iii) از برخی حالت‌های محیط صرف‌نظر شده است: امکان وجود روباتی دیگر در این محیط یا نوع آب و هوا در Carribbean . ساده‌سازی‌های صورت گرفته در واکنش‌ها: (i) فرض می‌کنیم تمام موقعیت‌ها برای کارگزار به صورت امن قابل دسترس‌اند: مثلا روبات نمی‌تواند در جایی گیر کند یا حتی خراب شود. (ii) روبات می‌تواند به اندازهٔ مسافت مورد نظرش حرکت کند بدون آنکه نیاز به شارژ مجدد داشته باشد. (iii) در بخش حرکت نیز ساده‌سازی‌هایی صورت گرفته است: حرکت به جلو به اندازهٔ فاصله‌ای مشخص فرض شد، در حالیکه این موضوع نیاز به چندین موتور خود کنترل و همچنین تعدادی سنسور جهت جلوگیری از برخورد با دیوار دارد.

---

[۶] بعبارت دیگر در تدوین مساله 8- وزیر، چگونگی نمایش قرارگیری تمام وزیرها در صفحه شطرنج در حالتهای مختلف میتواند به روشهای گوناگونی انجام شود مثلا به طور جدولی، رشته‌ای از اعداد و یا نمادین که این تنها بخش تعریف حالت در تدوین مساله است وتعریف حالت اولیه، تابع پسین، آزمون هدف و هزینه نیز بایستی صورت پذیرد ولی تدوین هدف یعنی با توجه به این ساختار تعریف شده، شکل هدف که کارگزار باید بدنبال آن باشد چگونه است. مشخص است که قبل از تعیین شکل هدف باید شکل کلی حالات، تعیین شده باشد.

**؟ ۳.۳ فرض کنید دو دوست در دو شهر متفاوت بر روی نقشه زندگی می‌کنند مانند نقشه رومانی که در شکل ۲.۳ نمایش داده شده است. در هر دفعه، می‌توانیم یک دوست را به یکی از شهرهای همسایه اش در نقشه منتقل کنیم (به طور شبیه‌سازی شده) که مقدار زمان موردنیاز برای جابجایی از شهر $i$ به شهر همسایه $j$ به فاصلهٔ بین آن دو شهر بستگی دارد. در هر بار که یک دوست را حرکت می‌دهیم باید صبر کنیم تا دوست دیگر نیز منتقل شود (مانند انتظار در طی تماس تلفنی که معمولا یکی در میان انجام می‌شود). سپس دور بعدی مجددا به همین روال آغاز می‌شود. می‌خواهیم کاری کنیم که این دو دوست تا حدامکان سریعتر به یکدیگر برسند. الف) یک فرمول دقیق برای این مسأله جستجو بنویسید. ( تعریف برخی نمادها در اینجا به شما کمک می‌کند). ب) فرض کنید $d(i,j)$ نشان‌دهندهٔ فاصله خط مستقیمی است که بین شهرهای $j$ و $i$ ترسیم شود. کدام یک از این توابع اکتشافی قابل‌قبول‌اند؟ (i) $d(i,j)$ (ii) $2\times d(i,j)$ (iii) $d(i,j)/2$ ج) آیا یک نقشهٔ کاملا مستقل وجود دارد که نتوان هیچ راه‌حلی برای آن یافت؟ د) آیا نقشه‌ای وجود دارد که در تمام راه‌حل‌های آن، بایستی یکی از دوست‌ها از شهری عبور کند؟**

☑ حل: **الف)** فضای حالت: حالت‌ها عبارتند از تمام جفت شهرهای ممکن که هریک را با زوج مرتب $(i,j)$ نمایش می‌دهیم. (مؤلفه اول نشان‌دهندهٔ شهری است که دوست اول در آن واقع است و مؤلفه دوم نشان دهندهٔ شهر دوست دوم). دقت شود که نقشه، فضای حالت این مسأله نیست. تابع پسین: پسین حالت $(i,j)$، تمام زوج مرتب‌های $(x,y)$ است که روابط $Adjacent(x,i)$ و $Adjacent(y,j)$ برای آنها برقرار باشد. هدف: بودن در حالت $(i,i)$ برای هر مقدار $i$ (یعنی دو دوست در یک شهر باشند ولی نام شهر اهمیتی ندارد). تابع هزینه گام: هزینهٔ گام برای رفتن از حالت $(i,j)$ به حالت $(x,y)$ که عبارت از $max(d(i,x),d(j,y))$ است. **ب)** در بهترین حالت، سرگردانی هر دو دوست در حالت‌های مختلف برابر است بنابراین در هر مرحله، هزینه زمانی بین آن دو تقسیم شده و در کل کاهش می‌یابد. از این رو تابع سوم قابل‌قبول است. **ج)** بله. به عنوان مثال یک نقشه با دو شهر که توسط یک لینک به هم متصل هستند را در نظر بگیرید که دو دوست می‌توانند هر لحظه جای خود را به شهری دیگر تغییر دهند. در این حالت اگر هر دو دوست به تعداد دفعاتی فرد جابجا شوند، هرگز ملاقات صورت نمی‌گیرد (این موضوع بر روی یک گراف فضای‌حالت ترسیم شده، بهتر نمایان خواهد بود). حتی در صورت انجام تعدادی زوج از حرکت نیز ممکن است عدم ملاقات صورت پذیرد زیرا هر حرکت در این نقشه فاصلهٔ بین دو دوست را صفر یا 2 می‌کند. **د)** بله. نقشهٔ گفته شده در قسمت ج که ممکن است غیرقابل‌حل باشد را درنظربگیرید با این تفاوت که بر روی هر شهر، یک حلقه به خود آن شهر اضافه شده باشد. اگر یکی از دوست‌ها به تعداد دفعاتی فرد جابجا شوند، با انجام یک حلقهٔ بازگشتی موجب می‌شود تا فاصلهٔ آن دو دوست به 1 برسد بنابراین مسأله قابل‌حل می‌شود. ولی در صورت عدم‌وجود این حلقه‌ها، مسأله مانند آنچه که در قسمت ج گفته شد، غیرقابل‌حل است.

**؟ ۴.۳ نشان دهید که چیدمان‌های مختلف پازل 8- تایی، می‌توانند به دو مجموعه مجزا تقسیم شوند به طوری که نتوان هیچ حالتی از مجموعهٔ اول را با تعدادی حرکت، به حالتی از مجموعهٔ دوم تبدیل نمود. (راهنمایی: به Berlekamp et *al.* (سال 1982) مراجعه کنید). قطعه کدی پیشنهاد دهید که بگوید هر حالت داده شده در این پازل (هر چیدمان ممکن) در کدام یک از این دو مجموعه قرار دارد؟ توضیح دهید چرا این روش برای تولید حالات تصادفی مناسب است؟ *(این تمرین مشابه تمرین ۴.۳ در ویرایش دوم است)***

☑ حل: در آدرس http://www.cut-the-knot.com/pythagoras/fifteen.shtml می‌توانید حل این موضوع را برای مسأله پازل 15- تایی مشاهده کنید که می‌توان بخش‌هایی از آن را برای پازل 8- تایی نیز استفاده نمود: **تعریف:** حالت هدف را به صورت «خانه‌های مرتب شده» تعریف می‌کنیم و جهت بررسی هر حالت بایستی از گوشهٔ سمت چپ سطر اول شروع به حرکت نموده و با حرکت از چپ به راست، خانه‌ها را بررسی کنیم تا به مرتب‌بودن آنها یقین حاصل نماییم. با رسیدن به انتهای هر سطر، به سمت چپ‌ترین خانه در سطر بعدی جهش می‌کنیم. این کار تا پایان خانه‌ها ادامه پیدا می‌کند. در طی این پیمایش، اگر دو خانه یافتیم که عدد خانهٔ دوم بیشتر از خانهٔ قبلی‌اش باشد، آنگاه آن دو خانه بایستی تعویض شوند تا به هدف برسیم. **پیشنهاد:** پازل تعریف شده در بند فوق را در نظر گرفته و فرض کنید N نشان‌دهندهٔ مجموع تعداد کل تعویض‌ها به علاوه شماره سطر خانهٔ خالی باشد. بنابراین باقیماندهٔ N بر 2 مستقل از هر حرکت خواهد بود. یعنی اگر برای یک چیدمان N فرد باشد، پس از انجام یک حرکت همچنان فرد باقی می‌ماند و اگر زوج باشد، پس از انجام یک حرکت زوج باقی می‌ماند. (افزایش و کاهش دو واحد به عددی فرد، عددی فرد تولید می‌کند و همین طور برای عدد زوج). با توجه به این موضوع، حالت هدف شکل ۴.۳ را در نظر بگیرید[7]. در این شکل، تمام خانه‌ها به ترتیب مرتب شده و به هیچ تعویضی نیاز نیست و خانهٔ خالی در سمت چپ سطر اول قرار دارد، که طبق پیشنهاد گفته شده، مقدار N برابر یک خواهد شد، بنابراین برای رسیدن به این حالت هدف که مقدار N آن

---

[7] قسمتی از شکل ۴.۳ که حالت هدفی با N=1 را نشان می‌دهد.

| | | |
|---|---|---|
| | 1 | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

رد است، سیستم باید از حالتی فرد شروع به کار نماید که در ابتدا Nای فرد برگ داشته و به تدریج با کسر دو واحد در هر رحله به این حالت هدف فرد دست یابد. شروع از حالتی زوج، نمی‌تواند کارگزار را به این حالت هدف فرد برساند. بنابراین دو وع چیدمان پازل داریم، چیدمان‌هایی که دارای N فرد هستند و چیدمان‌هایی که دارای N زوج هستند. **اثبات:** می‌دانیم که غزاندن یک خانه در جهت افقی، هیچ تغییری در تعداد کل تعویض‌ها و شماره سطر خانه خالی ایجاد نمی‌کند، بنابراین در این ثبات فقط لغزش عمودی یک خانه را درنظر می‌گیریم. فرض کنید خانهٔ A دقیقا بالای خانه خالی واقع شده باشد. اگر حرکتی جام شود که فقط بر روی خانه‌های B و C و D اثرگذار باشد و این خانه‌های B و C و D مستقل از حرکت A باشند (همگی زرگتر از A باشند)، آنگاه با لغزش عمودی این خانه، سه واحد به مجموع تعویضات اضافه می‌شود که سه عددی فرد است. کنون وضعیت دیگری را در نظر بگیرید که یکی از این سه خانه، کوچکتر از A باشد بنابراین بایستی قبل از حرکت و لغزش تدا آن خانه کوچکتر را با A تعویض کنیم و سپس حرکت را انجام دهیم. اکنون مجموع تعویضات مورد نیاز برای این خانه‌ها 2 ی‌باشد. (با کسر 1 تعویض از عدد 3 به عدد 2 رسیدیم). البته این دو فرضیه هر دو نتایج یکسانی را دربردارند. لذا می‌توان به ین نتیجه رسید که تغییر در مجموع N، همواره زوج است و این همان چیزی است که قصد اثبات آن را داشتیم (اعمال تغییرات وج بر روی هر عدد، تغییری در زوج و فرد بودن آن ایجاد نمی‌کند). بنابراین قبل از حل یک پازل، بایستی مقدار N را برای حالت شروع و حالت هدف مدنظر محاسبه کنیم و مطمئن شویم که از نظر زوج و فرد بودن با هم تطبیق داشته باشند وگرنه یچ راه‌حلی برای آن مساله وجود نخواهد داشت.

**؟ ۵.۳ مسأله $n$- وزیر با راه‌حل تدوین‌افزایشی‌موثر که در این فصل بیان شد، را درنظر بگیرید. توضیح دهید که چرا حداقل ندازهٔ فضای حالت، $\sqrt[3]{n!}$ است. همچنین بزرگترین $n$ ممکن برای حل کامل این مسئله را بیابید. (راهنمایی: با قرار دادن هر زیر در ستونی مجزا و شمردن حداکثر تعداد خانه‌هایی که در تهدید آن هستند، یک حد پایینی برای فاکتور انشعاب تعریف مایید). *(این تمرین مشابه تمرین ۵.۳ در ویرایش دوم است)***

☑ حل: در این تدوین، هر وزیر را در ستونی مجزا قرار می‌دهیم و وزیر جدید بایستی در خانه‌ای قرار بگیرد که توسط هیچ وزیر یگری تهدید نشود. به منظور ساده‌سازی، به جای مسأله 8- وزیر، مسأله 8- رخ را در صفحهٔ شطرنج در نظر بگیرید. اولین رخ ی‌تواند در هر خانه از ستون اول قرار بگیرد. دومین رخ می‌تواند در هر خانه‌ای از ستون دوم بجز سطر اول قرار بگیرد و. . . . در حالت کلی اندازهٔ فضای حالت در این وضعیت $n!$ می‌گردد. لازم به ذکر است که در مسألهٔ $n$- وزیر، هر وزیر سه خانه از ستون‌های مابعد را تهدید می‌کند (یک خانه در همان سطری که این وزیر قرار دارد، یک خانه مربوط به حرکت مورب وزیر به الا و خانه دیگر مربوط به حرکت مورب به سمت پایین می‌باشد. پس به ازای هر وزیر در ستون بعد، سه خانه تهدیدشده داریم). نابراین به علت وجود وزیر اول در ستون اول، ستون دوم دارای حداقل$(n\text{-}3)$ خانه امن باشد که بایستی در انتخاب جایگاه وزیر ین ستون لحاظ شود. به همین ترتیب در ستون سوم، به علت آنکه دو وزیر قبل از آن قرار دارند، $(n\text{-}6)$ انتخاب امن داریم و الی خر. بنابراین اندازهٔ فضای حالت عبارت است از $S \geq n(n\text{-}3)(n\text{-}6)\ldots$ پس داریم:

$$\begin{aligned} S^3 &\geq n \cdot n \cdot n \cdot (n-3) \cdot (n-3) \cdot (n-3) \cdot (n-6) \cdot (n-6) \cdot (n-6) \cdot \cdots \\ &\geq n \cdot (n-1) \cdot (n-2) \cdot (n-3) \cdot (n-4) \cdot (n-5) \cdot (n-6) \cdot (n-7) \cdot (n-8) \cdot \cdots \\ &= n! \end{aligned}$$

$$S \geq \sqrt[3]{n!}$$

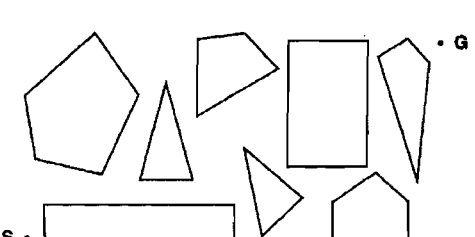
به طور خلاصه:

**؟ ۶.۳ برای هر یک از این موارد، حالت شروع، آزمون هدف، تابع پسین و تابع هزینه را ذکر کنید. همچنین تدوینی دقیق بیابید ه قابل پیاده‌سازی باشد. الف) قصد رنگ‌آمیزی نقشه ای مسطح با 4 رنگ را دارید به طوری که هیچ دو منطقهٔ مجاوری، هم‌رنگ باشند. ب) میمونی با قد 3 فوت در اتاقی به ارتفاع 8 فوت قرار دارد که تعدادی موز از سقف آن آویخته شده است و این میمون ی‌خواهد این موزها را بگیرد. در این اتاق تعدادی جعبه به ارتفاع 3 فوت قرار دارد که می‌توان آنها را حرکت داده و روی یکدیگر رار داد و همچنین می‌توان از آنها بالا رفت. ج) برنامه‌ای دارید که با بررسی فایلی از رکوردهای ثبت شده، پیغام «رکورد ثبت شده غیر مجاز است» را در خروجی چاپ می‌کند. می‌دانیم که پردازش هر رکورد ثبت شده مستقل از سایر رکوردهاست و شما ی‌خواهید غیرمجاز بودن یک رکورد را تشخیص دهید. د) شما سه کوزه با حجم‌های 12 گالن، 8 گالن، 3 گالن و نیز یک شیر آب ر اختیار داشته و می‌توانید هر کوزه را از آب پر کرده و یا آب آن را بر روی زمین و یا کوزه‌های دیگر خالی کنید. شما ی‌خواهید دقیقا یک گالن آب در یکی از کوزه‌ها داشته باشید. *(این تمرین مشابه تمرین ۷.۳ در ویرایش دوم است)***

☑ حل: الف) حالت شروع: هیچ منطقه‌ای رنگ‌آمیزی نشده است. آزمون هدف: تمام مناطق رنگ‌آمیزی شده باشند و هیچ دو نطقهٔ مجاوری همرنگ نباشد. تابع پسین: انتساب یک رنگ به یک منطقه. تابع هزینه: تعداد انتساب‌ها. ب) حالت شروع: حالتی ه در متن ذکر شد. آزمون هدف: میمون، موز را گرفته باشد. تابع پسین: هل دادن یک جعبه از یک محل به محلی دیگر، راه فتن از یک محل به محلی دیگر، گرفتن موز (اگر روی جعبه ایستاده باشد). تابع هزینه: تعداد واکنش‌های انجام شده. ج) حالت روع: همهٔ رکوردهای ورودی در نظر گرفته شوند. آزمون هدف: چاپ پیغام «رکورد ثبت شده غیر مجاز است» به ازای یک کورد مشخص. تابع پسین: بر روی نیمهٔ ابتدایی مجموعه رکوردها، عملیات اجرا را تکرار کن، بر روی نیمهٔ دوم مجموعه کوردها، عملیات اجرا را تکرار کن. تابع هزینه: تعداد دفعات اجرا. توجه: این مسأله جزء دسته مسائل محتمل الوقوع قرا می‌گیرد یرا برای انتخاب عملیات بعدی، بایستی اول بدانیم که در اجرای قبلی پیغام خطایی تولید می‌گردد یا خیر. د) حالت شروع: سه

کوزه با مقادیر [0,0,0]. تابع پسین: به ازای [x,y,z]، تولید حالت‌های، [12,y,z] و [x,y,3] و [x,8,z] با پر کردن هر کوزه، تولید حالت‌های [x,0,z] و [0,y,z] و [x,y,0] با خالی کردن هر کوزه به ازای دو کوزهٔ x,y. آب درون y را درون کوزهٔ x بریزیم که موجب می‌شود کوزهٔ x دارای حداقل x+y یا y آب باشد، کاهش حجم کوزهٔ اول توسط کوزهٔ y. تابع هزینه: تعداد دفعات انجام.

**؟ ۷.۳ مسألهٔ یافتن کوتاهترین مسیر بین دو نقطه در صفحه‌ای با موانع چند ضلعی محدب که در شکل ۳۱.۳ ترسیم شده است را در نظر بگیرید. این مسأله به نوعی حالت ایده‌آلی از مسأله یافتن مسیر در یک محیط شلوغ توسط روبات است. الف) فرض کنید فضای حالت را به صورت مجموعه تمام موقعیت‌های (x,y) در صفحه در نظر بگیریم چه تعداد حالت در این فضا وجود دارد؟ چند مسیر برای رسیدن به هدف وجود دارد؟ ب) به طور مختصر توضیح دهید که چرا کوتاهترین مسیر از رأس یک چندضلعی به رأس دیگری، بایستی شامل تعدادی قطعه خطوط مستقیمی باشد که به رئوس چند ضلعی‌ها ختم می‌شود. در این حالت یک فضای حالت مطلوب را تعریف کنید. اندازهٔ این فضای حالت جدید چقدر است؟**




شکل ۳۱.۳ محیطی با موانع چندضلعی که S و G به ترتیب حالات شروع و هدف میباشند.

**ج) توابع موردنیاز برای پیاده‌سازی مسأله جستجویی را تعریف کنید که در آن یک تابع پسین وجود دارد که یک رأس را به عنوان ورودی گرفته و مجموعه رئوسی که در یک خط مستقیم با آن رأس هستند را در خروجی بر می‌گرداند. (همسایگان همان چندضلعی که رأس ورودی بر رویش واقع شده است، را فراموش نکنید). برای تابع آروینی از فاصلهٔ خط مستقیم کمک بگیرید. د) یک یا چند مورد از الگوریتم‌های گفته شده در این فصل را برای حل مسائلی این چنینی به کار ببرید و نظر خود را در مورد کارایی آنها بیان کنید. *(این تمرین مشابه تمرین ۳ ۱۵ در ویرایش دوم است)***

☑ حل: **الف)** اگر تمام نقاط (x,y) در صفحه به عنوان فضای حالت در نظر گرفته شوند، آنگاه تعداد حالات و تعداد مسیرها بی‌نهایت خواهد بود. **ب)** (در این مسأله نقطهٔ شروع و نقطهٔ هدف هر دو از رئوس می‌باشند.) می‌دانیم که کوتاهترین فاصلهٔ بین دو نقطه خط مستقیمی است که از آن دو می‌گذرد ولی در این مسأله پیمودن مسیری مستقیم به علت وجود موانعی بر سر راه مسیر نیست بنابراین کوتاهترین مسیر ممکن در این وضعیت، از تعدادی قطعه خط تشکیل شده است که متوالیا به هم وصل شده و شکل کلی آنها اندکی با خط مستقیم اصلی تفاوت دارد ولی تا حد امکان به آن شبیه است. در این مسیر اولین قطعه خط از محل رأس ابتدایی شروع شده و به نقطه‌ای بر روی یک مانع ختم می‌شود (هر مسیر دیگری که تماس بیشتری با مانع داشته باشد، مسلما طولانی‌تر خواهد بود). از آن جا که موانع چندضلعی‌اند، نقاط تماس بایستی در رئوس موانع باشند، بنابراین کل مسیر بایستی رأس به رأس ادامه یابد. اکنون با توجه به شرح فوق، می‌توان فضای حالت را به صورت مجموعه‌ای از مختصات رئوس موانع بیان نمود. برای شکل ۳۱.۳ تعداد رئوس موانع 35 می‌باشد. **ج)** تابع پسین حالات مجاز را تولید کرده و از روی واکنش، نتیجه مربوطه را بدست می‌آورد. این موضوع هم شامل حالات درونی و هم حالات هدف میگردد. اگر در گره node قرار داشته باشیم، برای رئوسی که در طی یک (واکنش، نتیجه) با هم برخورد دارند، در تابع پسین بنویسید:

یک گره جدید (گره شروع)←S

گره هدف←G

هزینه مسیر[node]+ تعداد هزینه‌های و←هزینه مسیر

**د)**

| | اول سطح | هزینه‌یکنواخت | اول عمق | عمق محدود | عمیق‌شونده تکراری | دوطرفه |
|---|---|---|---|---|---|---|
| کامل بودن | بله [a] | بله [a,b] | خیر | خیر | بله [a] | بله [a,b] |
| پیچیدگی زمانی | $O(b^{d+1})$ | $O(b^{1+\lfloor c'/\varepsilon \rfloor})$ | $O(b^m)$ | $O(b^l)$ | $O(b^d)$ | $O(b^{d/2})$ |
| پیچیدگی فضا | $O(b^{d+1})$ | $O(b^{1+\lfloor c'/\varepsilon \rfloor})$ | $O(b^m)$ | $O(b^l)$ | $O(b^d)$ | $O(b^{d/2})$ |
| بهینگی | بله [c] | بله | خیر | خیر | بله [c] | بله $^{c/2}$ |

**۸.۳ در فصل ۳ گفته شد که نمی‌توان مسیری با هزینهٔ منفی داشته باشیم. حال در این تمرین این موضوع را با تعمق بیشتر بررسی می‌کنیم. الف) فرض کنید هر واکنش هزینهٔ منفی به اندازه‌ای دلخواه داشته باشد. توضیح دهید که چرا در این حالت یک الگوریتم بهینه باید کل فضای حالت را جستجو کند؟ ب) اگر تأکید کنیم که هزینهٔ مسیر بایستی بزرگتر و مساوی مقداری منفی ثابت باشد، آیا کمکی می‌کند؟ هم درخت و هم گراف را درنظر بگیرید. ج) فرض کنید مجموعه‌ای از عملگرها تشکیل یک حلقه بدهند که اجرای این مجموعه با ترتیبی خاص، هیچ تغییری در حالت سیستم نمی‌گذارد. اگر همه این عملگرها هزینهٔ منفی داشته باشند این محیط برای کارگزاری که قصد انجام رفتاری بهینه دارد چه چیزی را ایجاب می‌کند؟ د) می‌توان عملگرهایی با هزینه منفی زیاد را برای مسیریابی نیز تصور نمود. به عنوان مثال برخی پیچ‌های جاده به قدری مناظر زیبا دارند که حاضرید برای آن زمان و سوخت بیشتری را صرف کنید ولی تا ابد آنجا نمی‌مانید. با عبارات دقیق توضیح دهید که چرا انسان‌ها در هنگام جستجوی فضای حالت با یافتن حلقه‌ای مناسب (مانند پیچی زیبا در جاده) تا بی‌نهایت دور آن نمی‌چرخند؟ سپس توضیح دهید چگونه فضای حالت و عملگر مسیریابی را تعریف کنیم تا کارگزار مصنوعی نیز از دورزدن بیهوده دور حلقه‌های مناسب اجتناب کند؟ هـ) آیا می‌توانید مسأله‌ای واقعی مثال بزنید که در آن هزینه مراحل موجب ایجاد حلقه شده باشد؟** ***(این تمرین مشابه تمرین ۳.۱۷ در ویرایش دوم است)***

راه حل: **الف)** هر مسیر ممکن است در طی پیشروی خود، ناگهان با هزینهٔ منفی بزرگی روبرو شده و تمام هزینه‌های قبلی را از بین ببرد. پس حتی یک مسیر که در ابتدا بد و پرهزینه به نظر می‌رسد ممکن است در کل، مسیری بهینه باشد. بنابراین می‌توان در هر مرحله بین هزینه‌های پیش‌رو، یکی را انتخاب کرده و ادامه داد و حتما بایستی تمام مسیرهای ممکن جستجو شوند تا بتوان بهترین مسیر که هزینه کلی آن کمینه است را پیدا نمود. **ب)** فرض کنید بزرگترین هزینهٔ ممکن به ازای یک کنش را برابر ثابت $c$ در نظر بگیریم. حال اگر بیشترین عمق فضای حالت را بدانیم (مانند زمانی که فضای حالت یک درخت است) و آن را با $d$ نمایش دهیم، آنگاه در هر مسیری با این عمق، حداکثر $cd$ هزینه صرف خواهد شد و هر مسیری با هزینهٔ بیشتر از $cd$، بهینه نبوده و می‌تواند هرس شود. ولی اگر در فضای حالت حلقه داشته باشیم (گراف فضای حالت)، فرضیهٔ هزینهٔ مثبت هیچ کمکی به ما نمی‌کند، زیرا ممکن است برای چندین دفعه حلقه را دور بزنیم که در هر بار هزینه $c$ به مجموع هزینه‌ها اضافه می‌شود. **ج)** کارگزار باید در هر بار این حلقه را دور بزند (با اینکه می‌تواند حلقه‌ای دیگر با هزینه‌ای مناسب‌تر پیدا کند). بنابراین بایستی مقدار مناسب بودن یک حلقه در هر دفعه‌ای که بازدید شود، کاهش پیدا کند بنابراین یک چشم‌انداز عالی که جذابیت زیادی دارد، پس از 10 بار دیده شدن در یک ساعت خسته‌کننده بوده و دیگر هیچ جاذبه‌ای ندارد. برای پیاده‌سازی این موضوع بایستی در فضای حالت یک حافظه گنجانده شود. در این صورت یک حالت فقط موقعیت جاری را نشان نمی‌دهد بلکه علاوه بر موقعیت جاری، دفعاتی که یک مکان ملاقات شده است را نیز در بردارد. جذابیت (پاداش) یک مکان جدید بر حسب زمانی نزولی به دست می‌آید که از تعداد دفعات دیده شدن آن مکان استفاده می‌کند. **هـ)** رفتاری تکراری و حلقوی مانند خوردن غذای حاضری ارزان و رفتن به سر کلاس.

**۹.۳ مسأله‌ای تحت عنوان کشیش‌ها و آدم‌خوارها به این صورت تعریف می‌شود. سه کشیش و سه آدم‌خوار در یک طرف رودخانه قرار دارند که تنها یک قایق با ظرفیت یک یا دو نفر موجود است. می‌خواهیم تمام این افراد را به طرف دیگر رودخانه منتقل کنیم ولی در هیچ جا نباید تعداد کشیش‌ها از تعداد آدم‌خوارها کمتر باشد وگرنه خورده می‌شوند (چه در قایق و چه در هر طرف رودخانه). این مسأله در هوش‌مصنوعی بسیار مشهور است زیرا عنوان اولین مقاله‌ای بود که از دید تحلیلی و فلسفی به موضوع تدوین مسأله پرداخت (Amarel,1968) الف) این مسأله را به دقت تدوین کنید به طوری که فقط موارد موردنیاز جهت اطمینان از یک راه‌حل معتبر را ذکر کنید. نموداری از فضای حالت کامل ترسیم کنید. ب) با استفاده از یک الگوریتم جستجو، این مسأله را به طور بهینه حل و پیاده‌سازی نمایید. آیا برای بررسی حالت‌های تکراری ایده‌ای مناسب وجود دارد؟ ج) با وجود پیچیدگی فضای حالت، چرا مردم برای حل این معما با مشکل مواجه هستند؟** ***(این تمرین مشابه تمرین ۳.۹ در ویرایش دوم است)***

راه حل: **الف)** یکی از تدوین‌های ممکن برای حالت مساله چنین است که هر حالت را به صورت مجموعه‌ای 6 تایی از اعداد صحیح تعریف می‌کنیم که به ازای هر یک از کشیش‌ها و آدم‌خوارها یک عدد قرار داده باشیم (به عنوان مثال می‌توان برای هر حالت مقداری دودویی تعریف کرد به این صورت که اگر فرد در طرف اول رودخانه قرار دارد مقدار مربوط به آن یک باشد وگرنه صفر. مثلا مجموعه $(1,1,0,1,0,1)$ نشان‌دهنده حالتی است که یک کشیش و یک آدمخوار در طرف دیگر رودخانه بوده و دو کشیش و دو آدمخوار در این طرف هستند. با فرض اینکه سه عدد اول مربوط به کشیش‌ها و سه عدد دوم مربوط به آدمخوارها باشد. طبق این تعریف در زمان شروع در حالت $(1,1,1,1,1,1)$ و در هدف در حالت $(0,0,0,0,0,0)$ هستیم.) همچنین می‌دانیم که قایق در ابتدا در این طرف رودخانه و سپس در طرف دیگر می‌باشد. حالت هدف حالتی است که در آن سه کشیش و سه آدم‌خوار در طرف دیگر رودخانه باشند و تابع هزینه به ازای هر عمل انجام شده محاسبه می‌شود. همچنین پسین یک حالت عمل تمام حالت‌هایی است که در آن 1 یا 2 نفر بر روی قایقی سوار شده و از یک سو به سوی دیگر حرکت می‌کنند. **ب)** فضای جستجو بسیار کوچک است. بنابراین هر الگوریتمی می‌تواند به طور بهینه در آن عمل کند. دقت داشته باشید که بایستی حرکت‌هایی که منجر به بازگشت به حالت‌های قبلی مشاهده شده می‌شوند را حذف کنید. **ج)** زیرا بسیاری از حرکت‌ها یا غیر مجازند (خورده شدن کشیش‌ها توسط آدم‌خوارها) و یا آنکه محیط را به یکی از وضعیت‌های قبلی باز می‌گرداند و دقیقا تشخیص اینها ممکن نیست و این به خاطر بزرگی فاکتور انشعاب در این مسأله است که هیچ راهی برای پیشروی در آن وجود ندارد.

**؟ ۱۰.۳ این اصطلاحات را به زبان خود تعریف کنید: الف) حالت ب) فضای حالت ج) درخت جستجو د) گره جستجو هـ) هدف و) واکنش ز) تابع پسین ح) ضریب انشعاب *(این تمرین مشابه تمرین ۳.۱ در ویرایش دوم است)***

☑ حل: **الف)** یک حالت موقعیتی از محیط تعریف می‌شود که کارگزار می‌تواند در آن قرار بگیرد و خود دارای دو نوع می‌باشد: حالت‌های جهان (موقعیت‌های حقیقی در جهان واقعی) و حالت‌های نمایشی (توصیفی اجمالی از جهان واقع که توسط کارگزار در زمان انجام کارهای مورد توانش، مورد استفاده قرار می‌گیرد). البته هر حالت را می‌توان به صورت‌های مختلف ترسیمی، عددی و یا متنی بیان نمود. برای مثال نشان دادن موقعیت وزیرها در هر حالت از چیدمان $n$- وزیر را می‌توان به صورت شکلی از کل صفحه و مهره‌ها، رشته‌ای 8 عددی نشان‌دهنده سطر هر وزیر و یا یک پاراگراف متنی که در آن موقعیت هر وزیر تشریح شده است، بیان نمود. هر یک از این حالات، موقعیت فعلی محیط را برای کارگزار تشریح می‌کند. **ب)** فضای حالت گرافی است که گره‌های آن همان حالت‌ها بوده و لینک‌های موجود در گراف، واکنش‌هایی هستند که عامل انتقال از یک حالت به حالت دیگر هستند. با استفاده از این گراف یک کارگزار می‌فهمد که هر واکنش چه تاثیری بر روی محیط داشته و به چه وضعیتی در محیط منجر می‌شود. **ج)** درخت جستجو درختی است (درخت یک گراف بدون دور است) که در آن گرهٔ ریشه، همان حالت اولیه در محیط است و مجموعه فرزندان هر گره در این درخت، همان مجموعه حالت‌هایی هستند که با انجام یک واکنش در محیط قابل دسترسی‌اند. **د)** به هر گره در درخت جستجو، گرهٔ جستجو گفته می‌شود. **هـ)** هدف، حالتی از محیط است که کارگزار برای رسیدن به آن تلاش می‌کند. بنابراین معمولا تعدادی از گره‌های درخت جستجو با رنگی دیگر یا شکلی متفاوت علامتگذاری شوند که حالات پایانی را نشان دهند. **و)** واکنش عبارتست از هرکاری که کارگزار می‌تواند انجام دهد و اقدام کند که منجر به انتقال کارگزار از یک حالت به حالت دیگر می‌شود. (یال‌های درخت جستجو) **ز)** یک تابع پسین را می‌توان اینگونه تعریف نمود: یک حالت محیط را می‌گیرد و سپس مجموعه‌ای از زوج مرتب‌های (حالت، واکنش) تولید می‌کند که نشان می‌دهد به ازای هر حالت از محیط، کارگزار چه واکنشی را بایستی انجام دهد. (تابع کارگزار) **ح)** ضریب انشعاب در یک درخت جستجو تعریف می‌شود و به معنای تعداد واکنش‌های ممکنی است که کارگزار در هر حالت می‌تواند انتخاب کرده وانجام دهد. [8]

**؟ ۱۱.۳ تفاوت بین حالت‌دنیا، توصیف‌حالت و یک گرهٔ‌جستجو در چیست؟ این تمایز چه فایده‌ای دارد؟**

☑ حل: حالت دنیا عبارتست از وضعیتی که در جهان حقیقی برقرار است یا می‌تواند برقرار باشد. مثلا بودن در شهر آراد یک حالت دنیا، و بودن در شهر بخارست حالت دیگری از دنیا محسوب می‌شود که معمولا هر حالت دنیا درون خود شامل جزئیاتی بیشتر می‌باشد نظیر اینکه اکنون در کدام خیابان شهر هستیم، رادیو هم اکنون چه چیزی را پخش می‌کند و قیمت چای در چین چقدر است؟ ولی توصیف حالت، توصیفی از حالت دنیاست که توسط کارگزار و درون آن صورت می‌گیرد که نشان‌دهندهٔ برداشت کارگزار از آن حالت دنیا می‌باشد. به عنوان مثال $In(Arad)$, $In(Bucharest)$ هردو توصیفاتی از حالات بودن در شهر هستند که دیگر جزئیاتی چون کانال رادیو در آن لحاظ نمی‌شود چون هیچ کمکی به هدف کارگزار نمی‌کند. بنابراین توصیفات دنیا فقط جنبه‌هایی از حالت واقعی دنیا را توصیف می‌کنند و در صورت نیاز تخمین‌زده می‌شوند. لذا بایستی بین حالت دنیا و توصیف حالت تفاوت قائل شویم زیرا توصیف دنیا فقط خلاصه‌ای از حالت دنیاست که بسته به درک کارگزار و هدف آن ساخته می‌شود و به علت جایزالخطابودن کارگزار در درک محیط پیرامون، ممکن است به ازای یک حالت دنیا برداشت کارگزار اشتباه بوده و توصیف حالتی متفاوت با حالت واقعی دنیا شکل بگیرد. گره‌های جستجو در طی عملیات جستجو تولید می‌شوند که هریک از آنها نشان‌دهندهٔ حالتی هستند که بسته به الگوریتم مورداستفاده بدست می‌آیند. این گره‌ها علاوه بر توصیف حالات شامل اطلاعات دیگری نیز هستند مانند دنبالهٔ واکنش‌های موردنیاز جهت رسیدن به این حالت. این موضوع بسیار مفید خواهد بود زیرا ممکن است گره‌های جستجوی متفاوتی را تولید کنیم که همگی حالت یکسانی داشته باشند. پس بهتر است گرهٔ جستجو اطلاعاتی بیشتر از توصیف حالت در خود داشته باشد و با آن متفاوت باشد.

**؟ ۱۲.۳ هر واکنشی مانند *Go(Sibiu)* از مجموعه‌ای از ریز واکنش‌ها تشکیل شده است: روشن‌کردن خودرو، رهاسازی ترمز، راندن به جلو و.... با استفاده از ترکیب این ریز واکنش‌ها می‌توان تعداد مراحل راه‌حل و زمان جستجو برای آن را کاهش داد. فرض کنید تمام فراترکیب‌های ممکن بین واکنش مرکب در حجمی زیاد را یافته‌ایم، بنابراین می‌توان گفت که جواب هر مسأله**

---

[8] در مساله مسافرت از شهر آراد تا شهر بخارست بر روی نقشه رومانی، منظور ازحالت، بودن در هر شهر است، فضای حالت کل ساختار نقشه و جاده‌های ارتباطی آنهاست که به صورت گراف ترسیم می‌شود، درخت جستجو درختی است که گره ریشه آن شهر آراد بوده و بسته به تعداد جاده متصل به آن چندین یال از آن خارج می‌شود و به سطح بعدی درخت می‌رسیم. مجددا از این سطح جاده‌های متصل به هر یک را رسم کرده و ادامه می‌دهیم. هر جا که شهر بخارست مشاهده شد آن را با دایره‌ای مشخص می‌کنیم تا هدف تعیین گردد، به هر یک از این شهرهای موجود در درخت، گره جستجو و به شهر بخارست، گره هدف می‌گوییم. واکنش همان اقدامی است که کارگزار انجام می‌دهد یعنی پیمودن جاده که یالهای درخت جستجو می‌باشند و تابع پسین تابعی است که به طور ریاضی یا شهودی بتواند حالت بعدی یک حالت را به ازای واکنش‌های ممکن تعیین کند که در اینجا می‌توان جدولی به این منظور تشکیل داد. ضریب انشعاب تعداد یال خروجی از هر گره درخت است یعنی چند جاده به هر شهر متصل است.

**سط این واکنش مرکب حل شده است مانند *Go(Sibiu) Go(Rimnicu Vilcea) Go(Pltesti) Go(Bucharest)*. توضیح**
**ید که چگونه می‌توان در این اطلاعات جستجو کرد؟ آیا راه‌کاری عملی جهت افزایش سرعت حل این مسأله وجود دارد؟**

ا حل: در این مسأله فضای حالت، درختی با عمق یک است که تمام حالات، پسین حالت شروع هستند. لذا در این حالت فرقی
ن الگوریتم جستجوی اول‌عمق و اول‌سطح وجود ندارد. اگر طول دنباله محدود به گرهٔ ریشه نباشد لذا بی‌نهایت پسین خواهیم
شت پس فقط الگوریتم‌هایی می‌توانند به کار روند که در همان لحظهٔ تولید پسین، هدف بودن آن را بررسی می‌کنند. حال چه
اقی می‌افتد اگر واکنش‌های مرکب مرتب باشند؟ اگر هیچ ترتیب خاصی بین آنها وجود نداشته باشد، آنگاه یک جستجوی
ادفی ولی متقارن برای راه‌حل رخ می‌دهد و اگر طبق ترتیب دیکشنری مرتب شده باشند، یک جستجوی اول‌عمق پیاده‌سازی
شود و اگر بر حسب طول دنباله مرتب شده باشند، یک جستجوی اول‌سطح پیاده‌سازی می‌شود. عیب این جستجوی فضای
لت آن است که اگر در جایی بفهمیم که با شروع از واکنشی خاص مانند «جداکردن باتری» هرگز به راه‌حل نمی‌رسیم، هیچ
می وجود ندارد تا از تمام واکنش‌های مرکبی که از این ریزواکنش شروع می‌شوند، صرف‌نظر کنیم. که این موضوع برای
ش‌های جستجوی غیررسمی، مشکل محسوب می‌شود. نادیده‌گرفتن ساختار دنباله، راه‌کاری عملی جهت جستجو نیست.

**۱۳.۳ ثابت کنید که روش GRAPH-SEARCH دارای مشخصهٔ تفکیک گراف است که آن را در شکل ۹.۳ بیان نمودیم.**
**هنمایی: با نمایش این مشخصه در نقطهٔ شروع، آغاز به کار کرده و سپس نشان دهید که اگر این مشخصه قبل از تکرار یک**
**گوریتم وجود داشته باشد همواره برقرار است). یک الگوریتم جستجو توصیف کنید که این مشخصه را مختل می‌کند؟**

ا حل: تعریف مشخصهٔ تفکیک گراف به این صورت می‌باشد: «هر مسیر از حالت شروع به یک حالت بسط نیافته، باید از یک
لت موجود در صف بررسی عبور کند.»[۹]. در ابتدای جستجو، صف دارای حالت شروع می‌باشد، بنابراین بدیهی است که هر
سیر از حالت شروع به یک حالت بسط نیافته بایستی از یک گرهٔ موجود درصف (که در اینجا خود حالت شروع است) عبور کند.
نون فرض کنید این مشخصه در ابتدای یک تکرار دلخواه از الگوریتم GRAPH-SEARCH از شکل ۷.۳ برقرار باشد. در این
سعیت صف خالی نبوده و گرهٔ $n$ که برگی انتخابی است در آن قرار دارد. پس از پایان این تکرار، گره $n$ از صف حذف شده و
ین‌های آن در صف قرار می‌گیرند (اگر قبلا گرهٔ $n$ بسط نیافته و یا در صف نبوده باشد). حال مسیری از حالت شروع به یک
لت بسط‌نیافته را درنظربگیرید که در این مسیر حداقل یک گره از گره‌های موجود در صف وجود داشته باشد. به جز مواردی
این گره، $n$ باشد مشخصهٔ تفکیک به طور خودکار برقرار است. از این رو بایستی بر روی گذر مسیر از گرهٔ $n$ تمرکز کنیم.
بق تعریف، گرهٔ بعدی $n'$ که در مسیر عبوری از $n$ قرار دارد، بایستی یک پسین $n$ باشد که در صف بررسی نیست. به علاوه $n'$
ی‌تواند جزء گره‌های بسط‌یافته باشد پس فرض می‌کنیم که مسیری از $n'$ تا یک گرهٔ بسط‌نیافته موجود باشد که از صف
ی‌گذرد که این موضوع مشخصهٔ تفکیک را مختل می‌کند. طبق لم زیر می‌دانیم که هر گرهٔ بسط‌یافته توسط دنباله‌هایی از
ه‌های بسط‌یافته به حالت شروع متصل است (لم زیر را به همراه اثباتش مشاهده کنید که همواره برقرار است). از این رو $n'$
سو گره‌های بسط‌یافته نیست پس آن را به صف می‌افزاییم. پس مسیر دارای یک گره از گره‌های موجود در صف گشته است و
خصهٔ تفکیک برقرار است. این مشخصه توسط الگوریتمی مختل می‌شود که گره‌ها را از صف به مجموعهٔ بسط‌یافته منتقل
د، قبل از آنکه تمام پسین‌های آنها تولید شده باشند. بنابراین در افزودن برخی پسین‌ها به صف شکست می‌خورد. لازم به ذکر
ت که نیازی نیست تا تمام پسین‌های یک گره به طور هم‌زمان و قبل از گسترش سایر گره‌ها تولید شوند و تا زمان گسترش
صل می‌توانند در صف باقی بمانند. **لم:** هر گرهٔ بسط‌یافته، توسط مسیری از گره‌های بسط‌یافته به حالت شروع متصل است.
**ات:** این لم در آغاز صحیح است زیرا حالت شروع به خودش متصل است از آن جا که ما هرگز گره‌های بسط‌یافته را حذف
ی‌کنیم، فقط نیاز داریم تا یک گرهٔ جدید که به لیست گره‌های بسط‌یافته اضافه می‌شود را بررسی کنیم. فرض کنید $n$ یک
ه بسط‌یافتهٔ جدید باشد که قبلا در صف بررسی بوده است و همچنین همسایهٔ گرهٔ $n'$ است که آن نیز قبلا بسط یافته است
عنوان مثال می‌توان گفت $n'$ پدر $n$ است). طبق این لم، فرض می‌شود که $n'$ طبق مسیری از گره‌های بسط‌یافته به حالت
وع متصل است پس از آن جا که $n$ و $n'$ به هم متصل‌اند، می‌توان نتیجه گرفت که $n$ نیز با مسیری از گره‌های بسط‌یافته به
لت شروع متصل است.

**۱۴.۳ کدام یک از این جملات صحیح و کدام یک غلط هستند؟ پاسخ خود را توضیح دهید: الف) تعداد گره‌های بسط‌یافته در**
**ش جستجوی اول‌عمق، حداقل به اندازهٔ این گره‌ها در الگوریتم A* با تابع قابل‌قبول می‌باشد. ب) $h(n)=0$ یک تابع اکتشافی**
**ل‌قبول برای پازل 8- تایی است. ج) A* در مسائل روباتیک کاربرد ندارد، زیرا ادراکات، حالات و واکنش‌ها همگی پیوسته**
**ستند. د) روش جستجوی اول‌سطح کامل است، حتی اگر هزینهٔ مراحل صفر درنظرگرفته شود. هـ) فرض کنید در صفحهٔ**
**طرنج، یک رخ موجود است که می‌تواند هر تعداد خانه‌ای را در جهت افقی و یا عمودی بپیماید ولی قابلیت پرش از روی**
**ین را ندارد. در این حالت فاصلهٔ منهتن یک تابع اکتشافی قابل‌قبول برای مسألهٔ رسیدن رخ از خانهٔ A به B با کمترین تعداد**
**رکت می‌باشد.**

ا حل: **الف)** غلط. یک الگوریتم اول عمق ممکن است دقیقا d گره را برای دستیابی به هدف بسط دهد ولی A* به طور
سترده بر الگوریتم‌های جستجوی گراف احاطه دارد تا یافتن راه‌حل بهینه را تضمین کند. **ب)** صحیح. این تابع همواره قابل

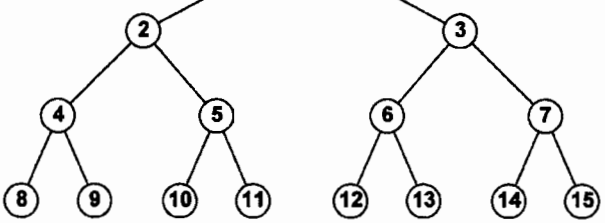
---

نظور از صف، لیستی است که گره‌هایی که بایستی بسط یابند در آن قرار می‌گیرند. هر گره که از ابتدای صف خارج شده و استفاده شود
اه فرزندان آن مجددا از انتها به صف بررسی اضافه می‌گردند. به این لیست صف بررسی می‌گوییم.

قبول است زیرا هزینه‌ها در آن غیرمنفی هستند. **ج)** صحیح. *A معمولا در روباتیک به کار می‌رود زیرا می‌توان فضارا گسترده نمود یا فقط بخش‌های اصلی آن را درنظرگرفت. **د)** صحیح. در روش اول‌سطح، عمق راه‌حل مهم است نه هزینهٔ آن. **هـ)** غلط. یک رخ با یک حرکت می‌تواند از یک طرف صفحهٔ شلوغ به انتها برود. در حالی که فاصله منهتن، عدد 8 را برای رسیدن آن به هدف تخمین می‌زند.

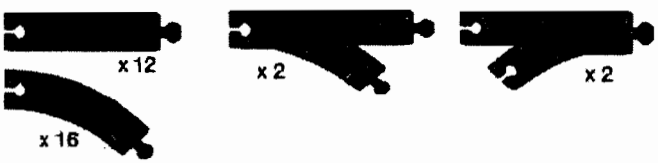
**؟ ۱۵.۳ فضای حالتی را در نظر بگیرید که در حالت شروع آن عدد 1 داریم و تابع پسین به ازای هر حالت $k$، دو حالت با اعداد $2k+1$ و $2k$ تولید می‌کند. الف) بخشی از فضای حالت برای حالت‌های 1 تا 15 را ترسیم کنید. ب) فرض کنید عدد 11 حالت هدف باشد، ترتیب گره‌های ملاقات شده توسط روش‌های جستجوی اول سطح، جستجوی عمقی محدود شده با محدودیت 3، و جستجوی عمیق شونده تکراری را لیست کنید. ج) آیا برای این مسأله، جستجوی دوطرفه مناسب است؟ ضریب انشعاب در هر جهت از جستجوی دو طرفه چقدر است؟ د) آیا پاسخ ج یک تدوین مجدد از مسأله را نشان می‌دهد که به شما اجازه دهد راه‌حلی بدون هیچ جستجو از حالت 1 به هدف بیابید؟ هـ) اقدامات رخداده از گره $k$ تا $2k$ در سمت چپ و $2k+1$ در سمت راست را بخوانید. آیا می‌توانید الگوریتمی بیابید که این مساله را بدون هیچ جستجویی حل کند؟** ***(این تمرین مشابه تمرین ۸.۳ در ویرایش دوم است)***

☑ حل: **الف)** به شکل ۱.۳s مراجعه شود. **ب)** اول‌سطح: 1 2 3 4 5 6 7 8 9 10 11 عمقی‌محدود: 1 2 4 8 9 5 10 11 عمیق‌شوندهٔ تکراری: 1, 1 2 3, 1 2 4 5 3 6 7, 1 2 4 8 9 5 10 11 **ج)** جستجوی دوطرفه بسیار مفید است، زیرا تنها پسین حالت $n$ در جهت معکوس، حالت $\lfloor (n/2) \rfloor$ است، که این موضوع به جستجو کمک می‌کند. در جهت پیش‌رو مقدار 2 دارد و در جهت معکوس مقدار 1. **د)** بله از حالت هدف شروع می‌کنیم و واکنش پسین معکوس را تا جایی ادامه می‌دهیم که به حالت 1 دست پیدا کنیم. **هـ)** راه‌حل با دودویی‌کردن هدف میسر است. بنابراین اعداد هدف را به صورت باینری بنویسید. از آنجا که با اعداد صحیح مثبت سر و کار داریم، پس بیت آغازین تمام آنها 1 خواهد بود. از پر ارزش‌ترین بیت تا کم ارزش‌ترین بیت حرکت می‌کنیم، اگر بیت صفر باشد به گره $2k$ و اگر بیت یک باشد به گره $2k+1$ می‌رویم (با صرف نظر از اولین بیت 1). به عنوان مثال فرض کنید هدف، عدد 11 باشد که باینری آن به صورت 1011 می‌شود. راه‌حل عبارتست از: چپ، راست، راست.




شکل ۱.۳s فضای حالت

**؟ ۱۶.۳ یک راه‌آهن چوبی از اجزایی مانند آنچه در شکل ۳۲.۳ می‌بینید تشکیل شده است. می‌خواهیم با این قطعات ریلی بسازیم که دو خط آهن همپوشانی نداشته و هیچ کدام بدون پایان رها نشوند که موجب سقوط قطار شوند. الف) فرض کنید که تمام قطعات بدون هیچ انقطاعی، برهم منطبق شوند. تدوینی دقیق از این مسأله به عنوان یک مسألهٔ جستجو ارائه دهید. ب) یک الگوریتم جستجوی غیررسمی و مناسب برای این مسأله بیابید و آن را توضیح دهید. ج) توضیح دهید که چرا حذف یکی از قطعات چنگالی، مسأله را غیرقابل‌حل می‌کند. د) حداکثر اندازه فضای حالتی که خود تدوین کردید، را بیابید. (راهنمایی: در مورد حداکثر فاکتور انشعاب و حداکثر عمق در طی عملیات ساخت بیاندیشید بدون آنکه در مورد هم‌پوشانی قطعات و عدم‌پایان مسیر نگران باشید. در ابتدا فرض کنید هر قطعه یکتاست).**




شکل ۳۲.۳ قطعات ریل برای مساله راه‌آهن چوبی که هر کدام از آنها را با عددی نشان دهنده تعداد موجود از آن قطعه مشخص کرده‌ایم. دقت کنید که قطعات منحنی شکل و چنگالی می‌توانند چرخانده شوند و در هر جهتی استفاده شوند. هر منحنی 45 درجه دارد.

☑ حل: **الف)** حالت شروع: یک قطعه که به‌طور دلخواه انتخاب شده است (یک قطعهٔ مستقیم). تابع پسین: به ازای هر اتصال نرگی باقی مانده، قطعه‌ای مناسب را از بین موارد باقی‌مانده انتخاب کنید[10]. (می‌توانید به جای اتمام یک مسیر، از قطعاتی استفاده کنید که مجددا در انتهای خود مادگی دارند تا قطعه‌ای بعدی با داشتن نرگی بتواند بر روی آن قرار بگیرد). برای قطعات منحنی، به چرخش آن در مسیر توجه کنید و برای قطعات چنگالی، به جهت توجه کرده و برای هر مادگی ایجاد شده ، اتصالات را قرار دهید. در تمام موارد سعی کنید از هم‌پوشانی و تداخل مسیرها جلوگیری کنید. (توجه: در مورد ادامه مسیرهایی که به مادگی ختم شده‌اند، نگران نباشید زیرا می‌توان آنها را با قطعات نرگی تکمیل نمود). آزمون هدف: تمام قطعات در یک ریل کاملا متصل، استفاده شده باشند که هیچ اتصال نرگی یا مسیری آزاد نبوده و با سایر مسیرها هم‌پوشانی نداشته باشد. هزینهٔ گام: به ازای هر قطعه محاسبه شود (در حقیقت اهمیت ندارد). **ب)** تمام راه‌حل‌ها در عمقی یکسان هستند پس جستجوی اول‌عمق پیشنهادی مناسب است (پیشنهاد دیگر الگوریتم جستجوباعمق‌محدود $n$-1 است ولی طبق مطالب گفته شده نیازی به بررسی محدودیت نیست، زیرا حالات در عمق $n$-1 دارای هیچ پسینی نیستند). فضا بسیار بزرگ است پس روش‌های هزینه‌یکنواخت و اول‌سطح شکست می‌خورند و الگوریتم عمیق‌شوندهٔ تکراری نمی‌تواند هیچ‌کار اضافه‌ای انجام دهد. بسیاری از حالات تکراری‌اند پس می‌توان از یک لیست بسته استفاده نمود. **ج)** در یک راه‌حل، هیچ مادگی یا نرگی نباید باز رها شود و هر اتصال نرگی در یک مادگی جای گیرد، پس تعداد اتصالات نر و ماده برابر است. حذف یک چنگال این خاصیت را نقض می‌کند. به دو روش می‌توان این موضوع را اثبات نمود: ۱- با اینکار تعداد پایان‌ها زوج می‌گردد. ۲- هر چنگال ریل را دو مسیری می‌کند و برای اتصال مجدد این دو مسیر به قطعهٔ چنگالی دیگر نیاز است. لذا اگر یکی از قطعات چنگالی حذف شود این کار میسر نمی‌گردد. این اثبات در حالت کلی نیز برقرار است یعنی می‌توان آن را به چنگال‌های سه‌تایی نیز اعمال نمود. (چنگالی با یک مادگی که سه مسیر نرگی ایجاد می‌کند یا چنگالی با یک نرگی که سه مسیر مادگی می‌سازد). عملیات پایان مسیر ممکن نیست، همانطور که برای چنگال/ اتصال‌مجدد بیان گشت. **د)** حداکثر تعداد نرهای رها شده 3 می‌باشد (در ابتدا یک نرگی داریم که پس از افزودن عنصر چنگالی، 3 می‌شود). فرض کنید هر قطعه یکتا باشد پس هر قطعه می‌تواند به یک نرگی اضافه شود پس برای هر نرگی تعداد (2.2.2)+(2.2)+(2.16)+12=56 انتخاب وجود دارد. کل عمق برابر 32 می‌باشد (زیرا 32 قطعه داریم). بنابراین باند بالایی $163^{32}$ / (!12.!16.!2.!2) می‌باشد. می‌توانید در تحلیل خود به صورتی عمل کنید که فاکتور انشعاب در طی پایین‌آمدن از درخت کاهش یابد ولی چندان جالب نیست.

**؟ ۱۷.۳ در بخش ۴.۳ در مورد الگوریتم جستجوی «طولانی‌کنندهٔ‌تکراری» بحث کردیم که نوعی تکراری از همان روش جستجوی هزینه‌یکنواخت است. به علاوه در هزینهٔ مسیر آن محدودیت افزایشی اضافه شد. اگر گره‌ای تولید شود که هزینه مسیر آن از محدودیت فعلی بیشتر باشد، فورا حذف می‌شود. در هر دفعه تکرار جدید این الگوریتم، محدودیت موردنظر به اندازهٔ کمترین هزینهٔ مسیر گره‌هایی تعیین می‌شود که در تکرار قبلی حذف شده‌اند. الف) نشان دهید این الگوریتم برای حالت کلی هزینهٔ مسیر، بهینه است. ب) درختی یکنواخت با فاکتور انشعاب $b$، عمق راه‌حل $d$ و هزینه گام واحد را در نظر بگیرید. در این حالت الگوریتم طولانی‌کنندهٔ‌تکراری به چند تکرار نیازمند است؟ ج) اکنون فرض کنید هزینهٔ گام به طور پیوسته در بازهٔ [0و1] بوده و حداقل هزینه، ثابت $\epsilon$ باشد، در بدترین حالت چه تعداد تکرار الگوریتم مورد نیاز است؟ د) این الگوریتم را پیاده‌سازی کرده و سپس آن را در مسائل پازل 8- تایی و فروشندهٔ دوره‌گرد بکار ببرید. آن‌گاه کارایی حاصل از این روش را با کارایی جستجو با هزینهٔ‌یکنواخت مقایسه کرده و در مورد نتایج آن توضیح دهید.** ***(این تمرین مشابه تمرین ۱.۳ در ویرایش دوم است)***

☑ حل: **الف)** در این الگوریتم عملیات بسط گره‌ها به ترتیب افزایش هزینه مسیر صورت می‌گیرد بنابراین اولین هدف یافته شده همان هدف با کمترین هزینه خواهد بود. **ب)** این الگوریتم در این حالت مشابه روش عمیق شوندهٔ تکراری عمل خواهد کرد. یعنی به $d$ تکرار نیاز دارد که در طی آنها، $O(b^d)$ گره تولید می‌گردد. **ج)** $d / \epsilon$

**؟ ۱۸.۳ فضای حالتی بیابید که در آن جستجوی عمیق شوندهٔ تکراری، بدتر از روش جستجوی اول‌عمق عمل نماید (به عنوان مثال با پیچیدگی $O(n^2)$ به جای $O(n)$).** ***(این تمرین مشابه تمرین ۱۳.۳ در ویرایش دوم است)***

☑ حل: فضایی را در نظر بگیرید که در آن هر حالت یک پسین داشته و تنها یک هدف در عمق $n$ موجود باشد. جستجوی اول‌سطح می‌تواند این هدف را در طی $n$ گام بیابید ولی روش عمیق شوندهٔ تکراری بایستی تعداد $1+2+3+\ldots+n=O(n^2)$ گام را سپری کند تا به این هدف برسد.

**؟ ۱۹.۳ برنامه‌ای بنویسید که به عنوان ورودی، URL دو وبسایت را گرفته و سپس مسیری از لینک‌های یکی از این سایت‌ها به دیگری را پیدا می‌کند. چه استراتژی جستجویی مناسبی پیشنهاد می‌دهید؟ آیا روش جستجوی دوطرفه، ایده‌ای مناسب است؟ آیا می‌توان از یک موتورجستجو جهت دانستن صفحه قبلی یک لینک کمک گرفت؟** ***(این تمرین مشابه تمرین ۱۴.۳ در ویرایش دوم است)***

---

[10] معمولا برای اتصال دو قطعه در یکی از آنها زائده و در دیگری حفره‌ای ایجاد می‌کنند تا آنها درون یکدیگر قرار گرفته و اتصال برقرار شود به این دو به ترتیب نرگی و مادگی در قطعات گفته می‌شود.

☑ حل: یک آدم معمولی (یا حتی کارگزار) تنها زمانی می‌تواند بفهمد صفحهٔ بعد یک لینک چیست که به آن مراجعه کند. پس بهتر است از روش‌هایی مانند اول‌سطح یا اول بهترین‌حریصانه استفاده نمود که بر حسب تعداد لغات رایج بین صفحهٔ ابتدایی و صفحهٔ هدف، عملیات جستجو را انجام دهند که شاید در رسیدن لینک‌ها به صفحه هدف کارساز باشد. در زمینهٔ استفاده از موتورهای جستجو، می‌دانیم که آنها گراف کاملی از وب را در اختیار دارند به طوری که می‌توانند یک کاربر را به تمام صفحات یا بخشی از صفحاتی که به یک صفحه لینک شده‌اند، هدایت کنند پس مشکل انسان‌ها را نداشته و استفاده از جستجوی دو طرفه در آنها ایدهٔ مناسبی است.

**؟ ۲۰.۳ مسأله محیط جاروبرقی که در شکل ۲.۲ بیان شد را در نظر بگیرید. الف) کدام یک از الگوریتم‌های گفته شده در این فصل می‌تواند برای این مسأله مناسب باشد؟ آیا این الگوریتم حالت‌های تکراری را بررسی می‌کند؟ ب) با استفاده از الگوریتم انتخابی خود، دنباله بهینه‌ای از واکنش‌ها را بیابید، به طوری که محیط دارای ابعاد 3×3 بوده و در حالت شروع، سه مربع بالایی کثیف است و کارگزار در مرکز این محیط قرار دارد. ج) یک کارگزار جستجو برای محیط جاروبرقی را طراحی کنید و کارایی آن را در مجموعه‌ای از دنیاهای 3×3 که هر خانه با احتمال 0.2 کثیف باشد، ارزیابی کنید. هزینهٔ جستجو و هزینهٔ مسیر را با استفاده از یک ضریب معقول، در معیار کارایی خود دخیل کنید. د) بهترین کارگزار جستجوی خود را با یک کارگزار واکنشی تصادفی ساده در حالتی مقایسه کنید که کارگزار در صورت وجود آشغال در یک خانه آن را می‌مکد وگرنه به طور تصادفی حرکت می‌کند. هـ) اتفاقات حاصل از گسترش دنیا به ابعاد $n\times n$ را در نظر بگیرید. معیار کارایی کارگزار جستجو و کارگزار واکنشی چگونه بر حسب $n$ تغییر می‌کند؟** ***(این تمرین مشابه تمرین ۳.۱۹ در ویرایش دوم است)***

☑ حل: واضح است که برای این موضوع بایستی از روش جستجوی گراف استفاده شود. این موضوع یک دنیای شطرنجی کلاسیک است که مسیرهای ثانویه بسیاری به هر حالت آن وجود دارد. دانشجویان شاید بتوانند برای این محیط دنباله راه‌حل بهینه را به سرعت بیابند ولی برای دنیای بزرگ $n\times n$، کاری دشوار و پرهزینه خواهد بود زیرا برای یک دنیا با ابعاد $n\times n$، فضای حالت دارای $n^2\times 2^n$ حالت خواهد بود، با افزایش $n$ در این مسأله، زمان تکمیل برای یک کارگزار تصادفی رشدی کمتر از نمایی بر حسب $n$ خواهد داشت. همچنین تاکنون ضریب معقولی که برای هزینهٔ جستجو و هزینهٔ مسیر در کارگزار تصادفی استفاده شده است، موفقیت‌آمیز بوده است.

**؟ ۲۱.۳ هریک از این عبارات را اثبات کنید. الف) جستجوی اول‌سطح، نوعی خاص از جستجو با هزینهٔ‌یکنواخت است. ب) جستجوهای اول عمق، نوعی خاص از جستجوی اول‌بهترین است. ج) جستجو با هزینهٔ‌یکنواخت، نوعی خاص از جستجوی A* است.** ***(این تمرین مشابه تمرین ۳.۴ در ویرایش دوم است)***

☑ حل: **الف)** اگر تمام هزینه‌های مراحل برابر باشند $g(n)\propto depth(n)$ آنگاه جستجوی با هزینه یکنواخت عملکردی مشابه اول سطح خواهد داشت. **ب)** جستجوی اول سطح با درنظر گرفتن $f(n)=depth(n)$ همان جستجوی اول بهترین می‌باشد. جستجوی اول عمق با در نظر گرفتن $f(n)=-depth(n)$ همان جستجوی اول بهترین می‌باشد و در نهایت جستجو با هزینه یکنواخت با فرض $f(n)=g(n)$ همان اول بهترین است. **ج)** با درنظرگرفتن $h(n)=0$، جستجوی با هزینه یکنواخت همان A* می‌باشد.

**؟ ۲۲.۳ کارایی روش‌های A* و RBFS را برا روی مجموعه‌ای از مسائل TSP و پازل‌های 8- تایی که به طور تصادفی تولید شده‌اند، مقایسه کنید. در مورد نتایج خود توضیح دهید. اگر یک مقدار تصادفی کوچک به مقدار اکتشافی افزوده شود چه تأثیری بر روی کارایی RBFS خواهد داشت؟** ***(این تمرین مشابه ۳.۱۸ در ویرایش دوم است)***

☑ حل: در مسألهٔ پازل 8- تایی، روش RBFS تعداد گره‌های بیشتری را بسط می‌دهد (به خاطر عدم تشخیص حالت‌های تکراری) ولی به ازای هر گره هزینه کمتری در بردارد. زیرا در این روش نیازی به نگهداری یک صف نداریم. البته در روش RBFS تعداد گره‌هایی که مجددا بسط می‌یابند، خیلی زیاد نیستند زیرا مسیر بهینه به ندرت تغییر می‌کند ولی زمانیکه مقدار اکتشافی کمی مناسب نباشد، این مزیت از بین رفته و کارایی RBFS به شدت کاهش می‌یابد. در مسألهٔ TSP، فضای حالت به شکل درخت است پس حالت‌های تکراری نداریم یا به بیانی دیگر مقادیر اکتشافی حقیقی بوده و به مقادیر حدسی نیازی نیست. بنابراین RBFS محکوم به بسط مجدد گره‌هایی است که قبلا بسط یافته‌اند.

**؟ ۲۳.۳ روش جستجوی A* را در مسألهٔ رسیدن به بخارست از شهر لوجیو (Lugio) اعمال کنید و تابع اکتشافی را همان فاصله خط مستقیم در نظر بگیرید. همچنین دنباله گره‌هایی که در این روش مورد استفاده قرار می‌گیرند و نیز مقادیر $g$ و $f$ و امتیاز هر گره را نشان دهید.** ***(این تمرین مشابه تمرین ۳.۱ در ویرایش دوم است)***

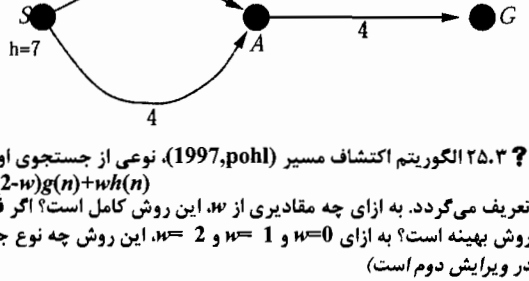
☑ حل: دنبالهٔ گره‌ها در صف به صورت زیر است:

L[0+244=244]
M[70+241=311], T[111+329=440]
L[140+244=384], D[145+242=387], T[111+329=440]
D[145+242=387], T[111+329=440], M[210+241=451], T[251+329=580]
C[265+160=425], T[111+329=440], M[210+241=451], M[220+241=461], T[251+329=580]
T[111+329=440], M[210+241=451], M[220+241=461], P[403+100=503], T[251+329=580], R[411+193=604], D[385+242=627]
M[210+241=451], M[220+241=461], L[222+244=466], P[403+100=503], T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627]
M[220+241=461], L[222+244=466], P[403+100=503], L[280+244=524], D[285+242=527], T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627]
L[222+244=466], P[403+100=503], L[280+244=524], D[285+242=527], L[290+244=534], D[295+242=537], T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627]
P[403+100=503], L[280+244=524], D[285+242=527], M[292+241=533], L[290+244=534], D[295+242=537], T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627], T[333+329=662]
B[504+0=504], L[280+244=524], D[285+242=527], M[292+241=533], L[290+244=534], D[295+242=537], T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627], T[333+329=662], R[500+193=693], C[541+160=701]

**؟ ۲۴.۳ فضای حالتی مثالی بزنید که در آن روش A* با جستجوی Graph-Search بتواند یک راه‌حل نیمه‌بهینه بیابد به طوری که تابع اکتشافی $h(n)$ در آن، قابل‌قبول و ناسازگار باشد.** ***(این تمرین مشابه تمرین ۴.۴ در ویرایش دوم است)***

☑ حل: شکل ۲.۳s را ببینید.

شکل ۲.۳s گرافی با تابع اکتشافی ناسازگار که GRAPH-SEARCH نمی‌تواند در آن راه‌حل بهینه را بیابد. پسین S با درنظرگرفتن $f=5$، گره A بوده و با $f=7$ گره B خواهد شد. گره A در ابتدا گسترش می‌یابد بنابراین مسیر B کنارگذاشته می‌شود زیرا A در لیست قرار دارد.




**؟ ۲۵.۳ الگوریتم اکتشاف مسیر (pohl,1997)، نوعی از جستجوی اول‌بهترین است که تابع‌ارزیابی آن به صورت:**
$$f(n)=(2-w)g(n)+wh(n)$$
**تعریف می‌گردد. به ازای چه مقادیری از $w$، این روش کامل است؟ اگر فرض کنید که $h$ قابل‌قبول باشد. به ازای چه مقادیری این روش بهینه است؟ به ازای $w=0$ و $w=1$ و $w=2$، این روش چه نوع جستجویی را انجام می‌دهد؟** ***(این تمرین مشابه تمرین ۴.۴ در ویرایش دوم است)***

☑ حل: هرگاه $0\le w<2$، این روش کامل است. و به ازای $w=0$، داریم: $f(n)=2g(n)$. این عبارت نشان می‌دهد که در این حالت، این روش همان عملکرد جستجو با هزینهٔیکنواخت را دارد. فاکتور 2 هیچ تفاوتی در ترتیب گره‌ها ایجاد نمی‌کند و اگر $w=1$ باشد این روش همان جستجوی A* می‌شود و $w=2$ منجر می‌شود که $f(n)=2h(n)$ که جستجوی اول‌بهترین حریصانه است. همچنین داریم:

$$f(n) = (2-w)[g(n) + \frac{w}{2-w}h(n)]$$

که همان عملکرد A* با تابع اکتشافی $h(n)$ ($w/(2-w)$) باشد. اگر $w\le 1$ باشد این تابع همواره کمتر از $h(n)$ است. بنابراین قابل‌قبول می‌باشد و طبق صورت سؤال می‌دانیم که خود مقدار $h(n)$ نیز قابل‌قبول است.

**؟ ۲۶.۳ نسخه‌ای نامحدود از صفحهٔ دوبعدی خط‌کشی‌شده در شکل ۹.۳ را در نظر بگیرید که در حالت شروع در مبدأ (0,0) بوده و حالت هدف در مختصات $(x,y)$ است. الف) در این فضای حالت، فاکتور انشعاب b چقدر است؟ ب) در عمق $k$ ($k>0$) چه تعداد حالت مجزا وجود دارد؟ ج) حداکثر تعداد گره‌های بسط‌یافته با روش جستجوی درختی اول‌سطح کدام است؟ د) حداکثر تعداد گره‌های بسط یافته با روش جستجوی گرافی اول‌سطح کدام است؟ هـ) تابع $h=|u-x|+|u-y|$ یک تابع قابل‌قبول برای حالتی در مختصات $(u,y)$ می‌باشد.توضیح دهید. و) چه تعداد گره در روش جستجوی گرافی A* و استفاده از $h$ بسط می‌یابند؟ ز) اگر برخی اتصالات حذف شوند، همچنان $h$ قابل‌قبول باقی می‌ماند. ح) اگر اتصالاتی را بین برخی حالات غیرمجاور بیافزاییم، $h$ همچنان قابل‌قبول باقی می‌ماند.**

☑ حل: **الف)** در این وضعیت، فاکتور انشعاب برابر 4 می‌باشد (تعداد همسایگان در هر مکان). **ب)** حالات موجود در عمق $k$، مربعی را تشکیل می‌دهند که 45 درجه چرخیده است. بنابراین تعدادی خطی از حالات که بر روی مرز این مربع هستند جواب این سؤال است که پاسخ آن $4k$ می‌باشد. **ج)** بدون بررسی حالات تکراری، روش اول‌سطح به طور نمایی گره‌ها را بسط می‌دهد

که اگر بخواهیم به طور دقیق تعداد آنها را بیان کنیم می‌توان گفت: $1-(4^{x+y+1}-1)/3)$ د) می‌توان گفت به اندازهٔ توان دوم، حالاتی است که درون مربعی در عمق $x+y$ وجود دارند که جواب آن برابر است با: $1-(x+y+1)(x+y)2$ هـ) صحیح. این فرمول همان فاصله منهتن می‌باشد. و) غلط. تمام گره‌های قرار گرفته در مستطیلی که گوشه‌های آن مختصات $(0,0)$ و $(x,y)$ دارند، کاندیدهایی برای مسیر بهینه هستند که تعداد آنها به صورت توان دو می‌باشد که در بدترین حالت ممکن است بسط یابند. ز) صحیح. زیرا برخی لینک‌ها مسیر انحرافی‌اند و به خاطر آنها بایستی مراحل بیشتری انجام شود. با حذف این لینک‌ها، $h$ همچنان تخمین کمتر است. ح) غلط. اتصالات غیرمحلی گفته‌شده می‌توانند مسیری کوتاهتر از فاصلهٔ منهتن ایجاد کند.

**؟ ۲۷.۳ در صفحه‌ای شطرنجی به ابعاد $n\times n$ تعداد $n$ کامیون قرار دارند که در خانه‌های $(1,1)$ تا $(n,1)$ می‌باشند. (می‌توان گفت در سطر آخر این صفحه حرکت می‌کنند). این کامیون‌ها قصد دارند به سطر بالایی منتقل شوند ولی با ترتیبی معکوس یعنی کامیون i که در ابتدا کار خود را در خانهٔ $(i,1)$ شروع کرده است بایستی در خانهٔ $(n-i+1,n)$ کارش را تمام کند. در هر مرحله از زمان، هر کدام از 11 کامیون می‌توانند به جهات بالا، پایین، چپ و راست حرکت کنند یا در جای خود باقی بمانند ولی اگر یک کامیون در جای خود باقی‌ماند، یکی از کامیون‌های مجاورش (نه بیشتر از یکی) می‌تواند از کنار آن عبور کند ولی به طور همزمان دو کامیون نمی‌توانند در یک خانه ثابت بمانند. الف) اندازه این فضای حالت را به صورت تابعی بر حسب $n$ بیابید. ب) فاکتور انشعاب را به صورت تابعی بر حسب $n$ محاسبه کنید. ج) فرض کنید کامیون $i$ در مختصات $(x_i,y_i)$ قرار داشته باشد. یک تابع اکتشافی $h_i$ قابل‌قبول برای تعداد حرکت‌های مورد نیاز این کامیون برای رسیدن به هدفش $(n-i+1,n)$ بیابید. فرض کنید هیچ کامیون دیگری بر روی صفحه نیست. د) برای مسأله رساندن همهٔ $n$ کامیون به مقاصدشان، کدام یک از این توابع اکتشافی، قابل قبول‌اند؟ توضیح دهید.** (i $\sum_{i=1}^{n} h_i$ (ii $\max\{h_1,\ldots,h_n\}$ (iii $\min\{h_1,\ldots,h_n\}$

☑ حل: **الف)** $n^{2n}$. زیرا $n$ کامیون داریم که می‌توانند در $n^2$ خانه قرار بگیرند. بنابراین با صرف‌نظر از محدودیت یک کامیون در هر خانه، تعداد حالات برابر با $(n^2)^n=n^{2n}$ است. **ب)** $5^n$ **ج)** فاصلهٔ منهتن: $|(n-i+1)-x_i|+|n-y_i|$ که برای مسأله‌ای با یک کامیون تخمینی دقیق به عمل می‌آورد. **د)** فقط مورد سوم که به صورت $\min\{h_1,\ldots,h_n\}$ است، قابل قبول می‌باشد. برای توضیح آن به این دو مورد توجه کنید. اول) فرض کنید کار $W$ در یک راه‌حل، کل فاصله‌ای باشد که تمام کامیون‌ها بایستی در مسیر خود بپیمایند که به ازای هر کامیون، طول مسیر بایستی اضافه شود. بنابراین داریم:

$$W_i \geq \sum_i h_i \gg n.min\{h_1,\ldots,h_n\}$$

دوم) کل کاری که در هر مرحله انجام می‌شود، کمتر و مساوی n است (توجه کنید که اگر کامیونی دو واحد حرکت کند، کامیونی دیگر حتما ثابت است که حرکتی صفر محسوب می‌شود. پس کل کار در هر مرحله به طور میانگین کمتر از $n$ است). بنابراین برای تکمیل تمام کارها حداقل به $n.min\{h_1,\ldots,h_n\}/n=min\{h_1,\ldots,h_n\}$ مرحله نیاز داریم.

**؟ ۲۸.۳ یک تابع اکتشافی برای مسألهٔ پازل 8- تایی پیشنهاد دهید که گاهی بیشتر از حد تخمین می‌زند. نشان دهید که این تابع چگونه می‌تواند در یک مسأله خاص، راه‌حلی نیمه‌بهینه پیدا کند. (در صورت تمایل می‌توانید از کامپیوتر کمک بگیرید). اثبات کنید که اگر تابع $h$، هزینه را هیچ‌گاه بیش از $c$ تخمین نزند، آنگاه روش A* با تابع اکتشافی می‌تواند راه‌حلی بیابد که در آن هزینهٔ راه‌حل بهینه بیشتر از $c$ نیست.** ***(این تمرین مشابه تمرین ۶.۴ در ویرایش دوم است)***

☑ حل: تابع اکتشافی $h=h_1+h_2$ (مجموع تعداد خانه‌هایی که در جای خود نیستند و فاصلهٔ منتهن)، در گاهی اوقات بیش از مقدار واقعی تخمین می‌زند. اکنون فرض کنید $h(n)\leq h^*(n)+c$ داده شده و همچنین می‌دانیم $G_2$ یک هدف نیمه‌بهینه با هزینه‌ای بیشتر از $c$ است. به عنوان مثال $g(G_2)>C^*+c$. اکنون هر گرهٔ $n$ که در مسیری به یک هدف بهینه وجود دارد را درنظربگیرید. داریم:

$$\begin{aligned} f(n) &= g(n)+h(n) \\ &\leq g(n)+h^*(n)+c \\ &\leq C^*+c \\ &\leq g(G_2) \end{aligned}$$

بنابراین $G_2$ هرگز قبل از بسط هدف بهینه، گسترش نخواهد یافت.

**؟ ۲۹.۳ ثابت کنید که اگر یک تابع اکتشافی، سازگار است، حتما باید قابل‌قبول باشد. یک تابع اکتشافی قابل‌قبول مثال بزنید که سازگار نباشد.** ***(این تمرین مشابه تمرین ۷.۴ در ویرایش دوم است)***

☑ حل: یک تابع اکتشافی سازگار است اگر به ازای هر گرهٔ $n$ که به ازای واکنش $a$ دارای پسین $n'$ است. داشته باشیم:

$$h(n)\leq c(n,a,n')+h(n')$$

برای یک اثبات ساده، تعداد $k$ گره از کوتاهترین مسیر به هدف که از گرهٔ n شروع شود را در نظر بگیرید. اگر $k=1$ و $n'$ گرهٔ هدف باشد آنگاه: $h(n)\leq c(n,a,n')$

به عنوان استنتاج فرض کنید که $n'$ روی کوتاهترین مسیر $k$ مرحله‌ای از هدف قرار دارد و $h(n')$ قابل‌قبول باشد آنگاه:

$$h(n)\leq c(n,a,n')+h(n')\leq c(n,a,n')+h^*(n')=h^*(n)$$

بنابراین تابع $h(n)$ در $k+1$ مرحله‌ای هدف، همچنان قابل‌قبول است.

**؟ ۳۰.۳ اگر برای مسألهٔ فروشندهٔ دوره‌گرد (TSP)، تابع اکتشافی را درخت پوشای مینیمم (MST) درنظر بگیریم، آنگاه با داشتن قسمتی از دوره، می‌توان هزینهٔ دورهٔ کامل را تخمین زده و مسأله حل شود. هزینهٔ MST برای مجموعه شهرها، عبارتست از حداقل مجموع لینک‌هایی که بتوانند تمام شهرها را به یکدیگر متصل کنند. الف) نشان دهید که این تابعِ اکتشافی می‌تواند از نسخهٔ سادهٔ TSP مشتق شود. ب) نشان دهید که تابع اکتشافی MST بر فاصله خط مستقیم حکمفرماست (از آن بیشتر است). ج) یک مولد مسأله برای نمونه‌ای از TSP بنویسید که در آن به جای شهرها، نقاطی تصادفی در مربعی واحد داشته باشیم. د) یک الگوریتم کارا برای ساخت درخت پوشای مینیمم یافته و در کنار آن با استفاده از یک جستجوی قابل قبول نمونه‌ای از TSP را حل کنید. *(این تمرین مشابه تمرین ۴.۸ در ویرایش دوم است)***

☑ حل: این تمرین، تکرار قسمتی از مسألهٔ کلاسیکی است که توسط Held و Karp در سال ۱۹۷۰ انجام شد. **الف)** در مسألهٔ TSP بایستی کوتاهترین مسیر بین شهرها که تشکیل یک حلقهٔ بسته دهند را بیابیم و طبق تعریف درخت پوشای مینیمال، می‌بینیم که همان نسخهٔ ساده شدهٔ TSP است زیرا به دنبال گراف مینیمالی می‌گردد که حلقهٔ بسته نداشته باشد ولی بایستی کاملا متصل باشد. بنابراین MST می‌تواند یک اکتشاف قابل‌قبول باشد. (همواره مسیری کوتاهتر و یا مساوی با یک حلقهٔ بسته ارائه می‌دهد). **ب)** در این مسأله بایستی از شهر انتهایی مجددا به شهر شروع باز گردیم که تابع اکتشافی «فاصلهٔ خط مستقیم» در این موضوع ضعیف خواهد بود. اگر تعداد شهرها زیاد باشد، این تابع مقدار را کمتر از حد واقعی تخمین می‌زند ولی در صورت کم بودن تعداد شهرها شاید کارکرد آن خیلی بد نباشد. اکتشاف MST بر اکتشاف «فاصلهٔ خط مستقیم» حکمفرماست. یعنی همواره برای یک گره، مقداری بالاتر را ارائه می‌دهد. این موضوع بدیهی است زیرا در زمان ساخت درخت پوشای مینیمال، اگر گرهٔ هدف و گرهٔ جاری را داشته باشیم می‌توانیم به دو شیوه عمل کنیم. یا آن دو را با یک خط مستقیم به هم وصل کنیم. همان اکتشاف فاصلهٔ خط مستقیم می‌شود) و یا توسط دو یا چند خط، این کار را انجام دهیم که این کار معمولا صورت می‌گیرد (طبق قاعدهٔ مثلث، این مقدار بیشتر از فاصلهٔ خط مستقیم خواهد بود). **ج)** در پیاده سازی خود بکوشید تا هر نقطهٔ بازدید نشده را به نزدیکترین همسایه‌اش متصل کنید و این کار بر طبق الگوریتم درخت‌پوشای‌مینیمال (MST) صورت می‌گیرد. **د)** در (Cormen etal ,1990, P50S)، الگوریتمی با پیچیدگی زمانی $O(E\ Log^{E})$ را مشاهده می‌کنید که در آن منظور از E همان تعداد لبه‌های گراف است.[۱۱]

**؟ ۳۱.۳ در فصل ۳، نسخه‌ای از پازل 8- تایی ارائه شد که در آن یک مکعب می‌توانست از خانهٔ A به خانهٔ B برود، در صورتی که خانهٔ B، خالی باشد. راه‌حل دقیق این مسأله توسط اکتشاف Gaschnig (Gaschnig,1979) انجام شده است. توضیح دهید که چرا دقت این تابع اکتشافی کمتر از تابع « $h1$=تعداد مربع‌هایی که در خانهٔ اشتباهی هستند» می‌باشد؟ مواردی را نشان دهید که این تابع مقداری کمتر از $h1$ و $h2$ داشته باشد ($h2$= فاصلهٔ منهتن). چگونه می‌توان این تابع اکتشافی را به طور کارا محاسبه نمود. *(این تمرین مشابه تمرین ۴.۹ در ویرایش دوم است)***

☑ حل: برای مسأله پازل 8- تایی در حالتی که یک مربع بتواند از خانهٔ A به خانه B حرکت کند. دقیق‌ترین اکتشاف همان $h1$ = تعداد مربع‌هایی که در خانهٔ اشتباهی هستند» می‌باشد ولی در نسخهٔ ساده‌تر این مسأله یعنی انتقال در صورت خالی بودن خانهٔ B، مقدار اکتشاف Gaschnig کمتر از مقدار اکتشاف $h1$ نبوده و همواره قابل‌قبول است. ( برای نسخهٔ سادهٔ این مسأله دقیق است). بنابراین اکتشاف Gaschnig دقیق‌تر خواهد بود. اگر در حالت هدف، مجاز باشیم تا دو مربع همسایه را جابجا پیش کنیم آنگاه حالتی داریم که در آن اکتشاف «$h1$ = تعداد مربع‌های اشتباه و $h2$ = تعداد فاصله منهتن» مقدار 2 را بر می‌گردانند ولی اکتشاف Gaschnig مقدار 3 را در بردارد. به منظور محاسبهٔ مقدار اکتشاف Gaschnig این مراحل را تکرار کنید تا به حالت هدف برسید: فرض کنید B، مکان فعلی خالی باشد، اگر برای رسیدن به هدف بایستی در خانهٔ B، مربع X قرار گیرد. مربع X را به خانهٔ B منتقل کنید وگرنه یکی از مربع‌هایی که در مکان اشتباه هستند را به خانهٔ B منتقل کنید دانشجویان می‌توانند ثابت کنند که این روش برای این نسخه از مسأله راه‌حل بهینه را می‌یابد.

**؟ ۳۲.۳ برای مسأله پازل 8- تایی دو تابع اکتشافی پیشنهاد شد: فاصلهٔ منهتن و دیگری تعداد مربع‌هایی که در مکان اشتباه هستند. در مراجع، توابع اکتشافی دیگری جهت بهبود این امر پیشنهاد شده است. به عنوان مثال : Nilson(1971) و Mostow and Prieditis (1989) و Hansson *et al.* (1992) با پیاده‌سازی این توابع اکتشافی، ادعای آنها را بررسی نموده و کارایی نتایج حاصل از الگوریتم‌های آنها را با هم مقایسه کنید. *(این تمرین مشابه تمرین ۴.۱۰ در ویرایش دوم است)***

☑ حل: دانشجویان به منظور مقایسهٔ نتایج آنها بایستی از جدولی حاوی زمان اجرا و تعداد گره تولیدی در هر یک استفاده کنند توابع اکتشافی مختلف، هزینهٔ محاسباتی مختلفی دارند). از آن جا که زمان اجرا در مسألهٔ پازل 8- تایی بسیار کوتاه است می‌توانید آزمون خود را بر حسب پازل‌هایی 15- تایی یا 24- تایی قرار دهید. استفاده از پایگاه الگو، بدترین شیوهٔ تجربه و تحقیق است.

---

[۱] روش‌های ساخت درخت‌پوشای مینیمم را می‌توانید در درس ساختمان‌داده، به طور کامل مطالعه کنید. از آن جمله می‌توان به روش‌های پریم و سولین اشاره نمود. هدف این الگوریتم‌ها آن است که با داشتن گرافی متشکل از چندین رئوس و هزینه هر یال، بتوان درختی تشکیل داد که تمام رئوس در آن موجود باشند و مجموع هزینه یال‌های انتخابی، کمترین مقدار ممکن باشد. یکی از این روش‌ها به این صورت است که از رأسی دلخواه شروع به حرکت کرده و از بین یال‌های موجود، یال با کمترین هزینه را به شرطی انتخاب می‌کنیم که دور ایجاد نشود. به همین شیوه تا پایان رئوس ادامه می‌دهیم.

# فصل ۴ (ویرایش سوم)

**؟ ۱.۴ نام الگوریتمی را ذکر کنید که از هر یک از این موارد خاص نتیجه می‌شود[12]: الف) جستجوی پرتومحلی با $k=1$ ب) جستجوی پرتومحلی با یک حالت شروع که هیچ محدودیتی در نگهداری تعداد حالات ندارد. ج) شبه‌تاب‌کاری( Simulated annealing) با $T=0$ در همه زمان‌ها د) شبه‌تاب‌کاری با $T=\infty$ در همه زمان‌ها. هـ) الگوریتم ژنتیک با اندازه جمعیت $N=1$ *(این تمرین مشابه تمرین ۴.۱۱ در ویرایش دوم است)***

☑ حل: **الف)** جستجوی پرتو محلی با $k=1$ همان جستجوی تپه‌نوردی است. (طبق توضیحات داده شده اگر به جای $k$ نقطه، فقط یک نقطه در نظر بگیریم، دقیقا روش تپه‌نوردی حاصل می‌شود) **ب)** جستجوی پرتومحلی با $k=\infty$ تاثیر زیادی ندارد. ایده این روش به اینصورت است که به علت بی‌نهایت بودن $k$ ، بایستی پسین تمام نقاط را نگهداری کنیم. لذا این جستجو، همان جستجوی اول‌سطح خواهدشد که در آن یک لایه از گره‌ها قبل از افزودن لایه بعدی، اضافه می‌گردند. با شروع از یک گره، این الگوریتم دقیقا همان جستجوی اول‌سطح را پی می‌گیرد. با این تفاوت که هر لایه به طور کامل در یک لحظه تولید می‌شود. **ج)** با حذف مرحله پایانی، این جستجو تبدیل به جستجوی «تپه‌نوردی با اولین انتخاب» می‌گردد. زیرا هر پسین روبه‌پایین با احتمال یک رد می‌شود.[13] **د)** شبه‌تاب‌کاری با $T=\infty$ در همه زمان‌ها بدان معناست که همواره یک حالت جدید را می‌پذیرد، پس همان جستجوی حرکت‌تصادفی است. **هـ)** اگر اندازه جمعیت 1 باشد، آنگاه در زمان انتخاب والدین بایستی هر دو والد را همان یک نفر انتخاب کنیم و پس از انجام عملیات پیوند(Crossover) بین آنها، فرزند ایجاد شده دقیقا کپی همان فرد اولیه خواهد بود و شانس ایجاد تغییر در مرحله جهش(Mutation) بسیار کم می‌گردد. بنابراین این الگوریتم عملکردی مانند حرکت‌تصادفی در یک فضای منحصر بفرد را دارد.

**؟ ۲.۴ تمرین ۱۶.۳ مسأله ساخت خط راه‌آهن را بیان کرد که در آن بایستی تمام قطعات به طور دقیق در کنار هم قرارگیرند و هیچ انقطاعی رخ ندهد. اکنون شکل حقیقی این مسأله را درنظربگیرید که در آن نیازی به کنارهم چیدن دقیق قطعات نیست و می‌توان هر قطعه را تا 10 درجه جهت چیدمان مناسب چرخاند. توضیح دهید که چگونه مسأله را تدوین کنیم تا بتواند توسط روش شبه‌تاب‌کاری حل شود؟**

☑ حل: این مسأله شباهت بسیاری به مسأله علمی «طراحی پروتئین» دارد. از جهت قطعات، فضایی گسسته داریم که بایستی قطعات را از آن انتخاب کرده و به ریل بیافزاییم ولی از جهت زوایای اتصال در مکان لینک دو قطعه، با فضایی پیوسته روبرو هستیم. بنابراین می‌توان یک حالت را به صورت مجموعه‌ای از قطعات چرخانده شده و متصل، زوایای اتصال در محدوده [-10,10] و همچنین مجموعه قطعات متصل‌نشده تعریف نمود. می‌دانیم که اتصالات و زوایای آنها به طور دقیق ساختار فیزیکی ریل را مشخص می‌کنند. البته بسته به نظر طراح عبور ریلی از روی دیگری می‌تواند مجاز یا غیرمجاز تعریف شود یا حتی به آن جریمه‌ای تعلق گیرد. تابع ارزیاب باید به گونه‌ای تعریف شود که حاوی اصطلاحاتی برای تعداد قطعات مورداستفاده، تعداد قطعات انتهایی و زاویه همپوشانی (در صورت مجازبودن همپوشانی) باشد. همچنین می‌توان اصطلاحاتی جهت «میزان علاقه و سود» و «قابلیت انتقال» برای هر ساختار ریل طراحی‌شده، درنظرگرفت تا یک طراحی را از این جهت بررسی کرده و امتیاز دهند. به عنوان مثال اگر یک قطار از یک قطعه شروع به حرکت کرده و در همان جهت و بدون هیچ چرخشی به قطعه نهایی برسد، ساختار ریل را بسیار مطلوب ارزیابی می‌کند و این دو پارامتر مقدار مناسبی خواهند داشت. بخش مهارت‌آمیز مسأله

---

[12] جستجوی تپه‌نوردی که از دسته روشهای جستجوی‌محلی محسوب می‌شود به سان عملکرد یک انسان نابینا بر روی دامنه کوه است که موقعیت خود را نمی‌داند ولی قصد رسیدن به قله را دارد. این فرد در هر گام، با استفاده از پای خود زمین اطراف را لمس کرده و هر سمتی که به سوی بالا باشد را ادامه می‌دهد. اگر به موقعیتی برسد که همه نقاط اطرافش رو به پایین باشند، نتیجه میگیرد که به قله کوه رسیده است. بنابراین اگر به جایی برسد که قله اصلی کوه نباشد ولی قله‌ای کوچک محسوب شود این فرد متوقف می‌شود. ولی در روش پرتومحلی، فرض می‌کنیم $k$ نفر بر روی نقاط مختلف کوه قرار دارند و همگی در هر گام، بهترین پسین خود را انتخاب می‌کنند و انتخاب بهتر را با یکدیگر به اشتراک می‌گذارند. (مشابه حرکت $k$ فرد نابینا که اطلاعات گام بعدی خود را با موبایل به هم اطلاع می‌دهند). بنابراین همگی به مرور در اطراف انتخاب برتر جمع می‌شوند. میتوان همین ایده را به جای پیمایش بر روی یک کوه، بر روی نموداری ریاضی که نشان‌دهنده کارایی است درنظرگرفت و یا با اعمال کمی تغییرات در آن به جای یافتن ماکزیمم مطلق، مینیمم مطلق را بیابیم.

[13] جستجوی شبه‌تاب‌کاری در اوایل حرکت خود یعنی زمان‌های اولیه، حرکتی به شدت تصادفی دارد یعنی ممکن است گام بعدی در پیمایش کوه به سمت پایین انتخاب شود. این انتخاب رو به پایین همواره و در هر گام انجام می‌شود ولی با گذشت زمان، احتمال انتخاب حرکتی رو به پایین کمتر و کمتر می‌شود. بنابراین دیگر در قله‌ای محلی گیر نمی‌کنیم. ولی اگر شرط زمان از این الگوریتم برداشته شود، دیگر این حرکت رو به پایین و تغییرات احتمالی آن رخ نداده و در هر گام اولین پسینی که بهتر از موقعیت فعلی باشد انتخاب می‌گردد.

همان مجموعه حرکات مجاز می‌باشد. واضح است که مایل نیستیم یک قطعه غیرمتصل را به قطعه‌ای لینک دهیم که یک نرگی باز دارد و در همان جهت مجاز قرار دارد. از دیگر موارد مشکل‌ساز، حرکتی است که یک نرگی و مادگی را در قطعاتی تقریبا متصل دوباره به یکدیگر متصل می‌کند و سپس جهت اتصال را تغییر می‌دهد. زیرا تغییر یک جهت ممکن است دیگران را نیز مجبور به تغییر کند و این تغییر مشخص نمی‌کند که آیا سایر قطعات در محدوده زاویه مجاز قرار دارند یا خیر. در حالت کلی یک راه‌حل مینیمال یکتا وجود ندارد که در آن تغییر یک زاویه به طور دنباله‌ای روی سایر زوایا اثر نگذارد و ممکن است برخی از تغییرات غیرممکن باشد.

**؟ ۳.۴ در این تمرین، استفاده از روش جستجوی‌محلی[14] برای حل مسأله TSP گفته شده در تمرین ۳۰۳ مورد تحقیق قرار می‌گیرد. الف) روش تپه‌نوردی را برای حل TSP بکار ببرید. نتیجه آنرا با راه‌حل بهینه‌ای که توسط الگوریتم *A با تابع اکتشافی MST بدست آوردید (تمرین ۳۰۳) مقایسه کنید. ب) الگوریتم ژنتیک را برای حل TSP بکار ببرید. نتیجه را با سایر روش‌ها مقایسه کنید. می‌توانید جهت مشورت به Larranaga *et al.* (1999) و پیشنهادات آن مراجعه کنید. *(این تمرین مشابه تمرین ۱۵.۴ در ویرایش دوم است)***

☑ حل: یک الگوریتم تپه‌نوردی ساده برای TSP به این شرح است: • تمام شهرها را توسط مسیری اختیاری به هم وصل کنید. • دو نقطه تصادفی روی مسیر انتخاب کنید. • مسیر را از این دو نقطه جدا کنید تا سه قطعه حاصل شود. • تمام 6 روش ممکن برای اتصال این سه قطعه به یکدیگر را امتحان نمایید و بهترین حالت اتصال را انتخاب کرده و مسیر را طبق آن مجددا متصل کنید. • مراحل فوق را آنقدر تکرار کنید تا هیچ بهبودی حاصل نشود.

**؟ ۴.۴ نمونه‌های زیادی از مسأله 8- پازل و 8- وزیر تولید کرده و در صورت داشتن پاسخ، آنها را به روش‌های تپه‌نوردی (انواع بیشترین‌شیب و اولین‌انتخاب)، تپه‌نوردی‌باشروع‌مجددتصادفی، و شبه‌تاب‌کاری حل کنید. هزینه جستجو و درصد مسائل حل شده را بیابید و سپس آنها را به همراه هزینه راه‌حل بهینه بر روی نموداری ترسیم کنید. در مورد نتایج خود توضیح دهید. *(این تمرین مشابه تمرین ۱۶.۴ در ویرایش دوم است)***

☑ حل: (راهنمایی: برای حل یک پازل 8- تایی به روش تپه‌نوردی‌بیشترین‌شیب، یک چیدمان تصادفی از اعداد 1 تا 8 را به عنوان حالت شروع در نظربگیرید. سپس مقدار $h$ در آن حالت را محاسبه کنید. حال تمام حالت‌هایی که یک خانه تغییر می‌کند را امتحان کرده و در هر یک مقدار $h$ را مجددا محاسبه نمایید. حالتی که کمترین مقدار $h$ را داشت، به عنوان حرکت بعدی در روش بیشترین‌شیب برگزینید. سپس آن حرکت را انجام داده و همین کار را برای آن حالت تکرار کنید تا به وضعیتی برسید که $h=0$ بوده و خانه‌ها مرتب شوند. در روش تپه‌نوردی اولین انتخاب، در هر مرحله نیازی نیست تا تمام حرکت‌های مجاز بعدی را امتحان کرده و بهترین را انتخاب کنید. بلکه حرکت‌های دیگر را امتحان کرده و اولین حرکتی که $h$ای بهتر از h فعلی داشت را انتخاب کنید. در روش تپه‌نوردی با شروع مجدد تصادفی، چندین پازل را به طور همزمان در نظر گرفته و به طور موازی با روش تپه‌نوردی حل کنید. هدف یافته‌شده بین آنها را به اشتراک گذاشته و بهترین را انتخاب کنید زیرا ممکن است یکی از پازل‌ها در مینیمم محلی $h>0$ گیرکرده باشد. به همین ترتیب مراحل را برای مسأله 8- وزیر تکرار کنید با این تفاوت که در آنجا هر حالت بعدی با حرکت یکی از وزیرها حاصل می‌شود که بایستی در آن وضعیت مقدار $h$ را مجددا محاسبه کرد.)

**؟ ۵.۴ الگوریتم AND-OR GRAPH-SEARCH که در شکل ۱۱.۴ بیان شد، حالات تکراری را فقط در مسیر ریشه تا هدف کنونی بررسی می‌کند. حال فرض کنید این الگوریتم بتواند هر حالت ملاقات‌شده را ذخیره کرده و مجددا لیست خود را بررسی نماید. (به عنوان مثال به BREADTH-FIRST-SEARCH در شکل ۱۱.۳ مراجعه کنید). اطلاعات موردنظر جهت ذخیره را تعیین کرده و بگویید این الگوریتم چگونه باید از این اطلاعات در زمان یافت یک حالت تکراری استفاده کند؟ (راهنمایی: بایستی بین حالاتی که منجر به یک زیرنقشه موفق می‌شوند و حالاتی که نمی‌توان هیچ نقشه‌ای از آنها یافت تمایز قائل شوید.) توضیح دهید که چگونه با استفاده از برچسب‌گذاری که در بخش ۳.۳.۴ گفته شد می‌توان از داشتن چندین کپی زیرنقشه جلوگیری نماییم.**

☑ حل: شکل ۱.۴s این الگوریتم تغییریافته را نشان می‌دهد. برای حالاتی که روش OR-SEARCH می‌تواند یک راه‌حل بیابد، راه‌حل یافته‌شده ثبت و ذخیره می‌شود. بنابراین اگر در ملاقات‌های بعدی به آن گره دوباره رجوع شود، فورا آن راه‌حل بازگردانده می‌شود. اگر روش OR-SEARCH نتواند یک راه‌حل بیابد، بایستی دقت کنید. از آنجا که یک حالت می‌تواند منجر به حل شود، در صورتی که روی مسیر درست قرار بگیرد، و نباید هیچ دوری داشته باشیم، می‌توانیم در مواردی که روش OR-SEARCH شکست می‌خورد، مقدار مسیر را ذخیره کنیم. حل اگر به حالتی برسیم که قبلا به عنوان شکست خورده ثبت شده است، روش OR-SEARCH مقدار شکست را نتیجه می‌دهد. برای اجتناب از یافتن قسمتی از راه‌حل به جای راه‌حل کامل، می‌توان تمام راه‌حل‌های جدید یافته‌شده را برچسب‌گذاری نمود و این برچسب‌ها را ذخیره کرد. بنابراین اگر این حالات مجددا ملاقات شدند، این برچسب‌ها بازگردانده می‌شوند. با استفاده از عملیات پس‌پردازش می‌توان برچسب‌های استفاده نشده را حذف نمود. همچنین می‌توان یک ساختار گرافی به جای درختی در خروجی داشته باشیم. برای جزئیات بیشتر به (Bertoli,2001) مراجعه کنید.

---

[14] منظور از جستجوی‌محلی، تمام روشهای جستجویی است که در هر گام برای انتخاب نقطه بعدی حرکت و یا گره بعدی برای بسط، فقط از اطلاعات همسایگان آن نقطه کمک گرفته شود مانند روش تپه‌نوردی.

```
function AND-OR-GRAPH-SEARCH(problem) returns a conditional plan, or failure
  OR-SEARCH(problem.INITIAL-STATE, problem, [])

function OR-SEARCH(state, problem, path) returns a conditional plan, or failure
  if problem.GOAL-TEST(state) then return the empty plan
  if state has previously been solved then return RECALL-SUCCESS(state)
  if state has previously failed for a subset of path then return failure
  if state is on path then
      RECORD-FAILURE(state, path)
      return failure
  for each action in problem.ACTIONS(state) do
      plan ← AND-SEARCH(RESULTS(state, action), problem, [state | path])
      if plan ≠ failure then
          RECORD-SUCCESS(state, [action | plan])
          return [action | plan]
  return failure

function AND-SEARCH(states, problem, path) returns a conditional plan, or failure
  for each s_i in states do
      plan_i ← OR-SEARCH(s_i, problem, path)
      if plan_i = failure then return failure
  return [if s_1 then plan_1 else if s_2 then plan_2 else ... if s_{n-1} then plan_{n-1} else plan_n]
```

شکل ۱.۴s جستجوی AND-OR به همراه بررسی حالات تکراری

```
function AND-OR-GRAPH-SEARCH(problem) returns a conditional plan, or failure
  OR-SEARCH(problem.INITIAL-STATE, problem, [])

function OR-SEARCH(state, problem, path) returns a conditional plan, or failure
  if problem.GOAL-TEST(state) then return the empty plan
  if state is on path then return loop
  cyclic − plan ← None
  for each action in problem.ACTIONS(state) do
      plan ← AND-SEARCH(RESULTS(state, action), problem, [state | path])
      if plan ≠ failure then
          if plan is acyclic then return [action | plan]
          cyclic − plan ← [action | plan]
  if cyclic − plan ≠ None then return cyclic − plan
  return failure

function AND-SEARCH(states, problem, path) returns a conditional plan, or failure
  loopy ← True
  for each s_i in states do
      plan_i ← OR-SEARCH(s_i, problem, path)
      if plan_i = failure then return failure
      if plan_i ≠ loop then loopy ← False
  if not loopy then
    return [if s_1 then plan_1 else if s_2 then plan_2 else ... if s_{n-1} then plan_{n-1} else plan_n]
  return failure
```

شکل ۲.۴s جستجوی AND-OR به همراه بررسی حالات تکراری

**۶.۴ توضیح دهید که چگونه می‌توان الگوریتم AND-OR-GRAPH-SEARCH را تغییر داد تا در صورت عدم‌وجود شه چرخه‌ای، یک نقشه چرخه‌ای بسازد. برای اینکار با سه موضوع سروکار دارید: برچسب‌گذاری مراحل نقشه تا در زمان واجهه با چرخه بتوان به اولین بخش نقشه بازگشت نمود، تغییر OR-SEARCH به گونه‌ای که پس از یافتن یک نقشه رخه‌ای باز هم به جستجو برای یافتن نقشه غیرچرخه‌ای ادامه دهد، و افزودن بخشی به بازنمایی نقشه که بتواند چرخه را نشان هد. نشان دهید که الگوریتم شما در الف) دنیای‌جاروبرقی‌دشوار و ب) دنیای‌جاروبرقی دشوار و سرگردان چگونه کار خواهد رد؟می‌توانید برای بررسی نتایج خود از پیاده‌سازی کامپیوتری استفاده نمایید.**

حل: صورت سئوال جزئیات لازم جهت تغییر در الگوریتم پایه را بخوبی توصیف می‌کند که شکل ۲.۴۸ تغییرات اعمال شده الگوریتم را نشان می‌دهد. زمانی که چرخه OR-SEARCH به حالتی بر روی *path* دوباره مراجعه نماید، کلمه token را می‌گرداند. و آن بدان معناست که آخرین بار از طریق این *path* دوباره به این حالت دسترسی داشته‌ایم. اگر *path* را در نقشه ز گردانده شده ذخیره نماییم آنگاه با جایگزینی آنها توسط برچسب، اطلاعات کافی برای پردازش‌های بعدی یا تغییر پیاده‌سازی واهیم داشت. نقشه باید به گونه‌ای باشد که با پیگیری سیر نقشه، چرخه‌دار بودن آن را متوجه شود (به عنوان مثال شامل *loop* بودن). بنابراین OR-SEARCH راه‌حل‌های بدون‌چرخه را ترجیح می‌دهد. AND-SEARCH درصورتی fail را می‌گرداند که تمام شاخه‌ها مستقیما منجر به یک حلقه شوند، در این مورد نقشه همیشه حاوی حلقه خواهد بود. تنها موردی که نیاز به بررسی دارد آن است که باید تمام شاخه‌ها در یک حلقه متناهی، گره‌های And باشند که فرزندان آنها مستقیما حلقه ایجاد نمایند.

**۷.۴ در بخش ۱.۴.۴ حالات‌باور جهت حل مسائل جستجوی‌بدون‌حسگر معرفی گشت. یک دنباله از واکنش‌ها می‌توانند یک مسأله بدون‌حسگر را حل کنند، اگر این دنباله بتواند هر حالت فیزیکی در حالت‌باور شروع *b* را به حالت هدف برساند. فرض نید کارگزار $h^*(s)$ را می‌داند که همان هزینه بهینه برای حل حالت فیزیکی *s* برای هر حالت *a* در *b* در یک مسأله کاملا شاهده‌پذیر است. یک تابع‌اکتشافی قابل‌قبول $h(b)$ برای این مسأله بیابید که از این اصطلاحات هزینه استفاده کند و سپس ابل‌قبول بودن آن را اثبات کنید. در مورد دقت این تابع‌اکتشافی در مسأله جاروبرقی شکل ۱۴.۴ نظر خود را بیان کنید.**

حل: همان طور که گفته شد دنباله‌ای از واکنش‌ها در صورتی راه‌حل یک مسأله حالت باور هستند که بتوانند هر حالت یزیکی اولیه را به یک حالت هدف برسانند. با کمی ساده‌سازی، فرض می‌کنیم فقط تعدادی از حالات فیزیکی اولیه به حالت دف می‌رسند. همچنین برای تعریف بهتر فرض کنید این دنباله در جهت پیداکردن راه‌حل برای حالت فیزیکی با بیشترین زینه راه‌حل تلاش می‌کند. اگر $h^*(s)$ نشان‌دهنده هزینه بهینه راه‌حل با شروع از حالت فیزیکی *s* باشد آنگاه:

$$h(S) = \max_{s \in S} h^*(s)$$

خمینی از اکتشاف این مسأله ساده‌شده می‌باشد. این اکتشاف فرض می‌کند هر راه‌حل که به دشوارترین حالت ختم شود، تمام حالات را حل خواهد کرد. در مسأله جاروبرقی بدون‌حسگر که در شکل ۱۴.۴ بیان شد، *h* به طور صحیح هزینه بهینه تمام حالات را تعیین می‌کند بجز سه حالت (که آنها توسط این عملیات رخ می‌دهند [suck]، [*suck* , *left*]، [*suck* , *right*] و یشه، در آنها *h* مقدار یک واحد کمتر از حقیقت برآورد می‌کند. این بدان معناست که روش A* این سه گره مرکزی را قبل از فتن به سوی هدف بسط می‌دهد.

**۸.۴ این تمرین رابطه زیرمجموعه- ابرمجموعه را بین حالات‌باور در یک محیط کاملامشاهده‌پذیر و نیمه‌مشاهده‌پذیر بررسی ی‌کند. الف) ثابت کنید اگر یک دنباله واکنش راه‌حلی برای حالت‌باور *b* باشد، آنگاه راه‌حلی برای هر زیرمجموعه‌ای از *b* نیز ست. آیا می‌توان در مورد ابرمجموعه *b* نیز نظر داد؟ ب) توضیح دهید که چگونه می‌توان جستجوی گراف در مسائل بدون سگر را تغییر داد تا از نتایج قسمت الف فایده بیشتری ببریم. ج) توضیح دهید که چگونه می‌توان جستجوی AND-OR را رای مسائل نیمه‌مشاهده‌پذیر تغییر داد، تا بیشتر از تغییرات انجام شده در قسمت ب باشد.**

حل: **الف)** هر دنباله واکنش برای حالت‌باور *b* در صورتی یک راه‌حل تلقی می‌شود که از هر حالت *s* که $s \in b$ شروع به رکت کرده و با انجام آن دنباله از واکنش‌ها، کارگزار را به حالت هدف برساند. هر دنباله واکنش که برای حالت‌باور *b* یک ه‌حل نباشد یعنی نتوان از آن به هدف رسید، برای هر ابرمجموعه آن نیز راه‌حل نمی‌باشد. این مخالف چیزی است که قصد بات آنرا داشتیم. زیرا در حالت کلی برای حالت‌های خارج از *b* و برای هر ابرمجموعه اختیاری نمی‌توانیم بگوییم که دنباله کنش‌ها ما را به هدف نمی‌رساند ولی چیزی که می‌توان گفت آن است که اگر یک دنباله واکنش بتواند یک حالت‌باور *b* و یک الت‌باور $b'$ را به هدف برساند، آنگاه می‌تواند اجتماع آن حالات‌باور یعنی $b \cup b'$ را نیز به هدف برساند. **ب)** در زمان بسط یک ره، اگر آن گره فرزند حالت‌باوری بود که آن حالت‌باور قبلا در یک ابرمجموعه‌ای از حالات‌باور بوده و بسط یافته باشد، آنگاه یازی نیست تا این گره را به صف بررسی اضافه نماییم. **ج)** اگر حالات‌باوری که قبلا حل شده‌اند رادر جایی ثبت و ذخیره کنیم، گاه فقط کافی است به ابتدای جستجوی AND-OR یک قسمت بررسی اضافه شود که در آن هر حالت‌باور را چک می‌شود ه آیا در زیرمجموعه‌ای از آن حالات‌باور حل شده قرار دارد یا خیر. اگر موجود بود فقط راه‌حل قبلی آن مورد را برمی‌گرداند.

**۹.۴ در این فصل فرض کردیم که هر واکنش که در هر حالت فیزیکی درون حالت باور مشخصی اجرا شود، هزینه یکسانی اشته باشد (این فرضیه موجب می‌گشت تا یک مسأله جستجوی حالت‌باور با هزینه گام خوش‌تعریف رخ دهد). حال چه اتفاقی ی‌افتد اگر این فرضیه برقرار نباشد؟ آیا در این صورت توجه به بهینگی همچنان مهم است یا نیاز به تغییراتی در آن داریم؟**

**همچنین فرض کنید تعریف هزینه اجرایی یک واکنش در هر حالت‌باور متفاوت باشد. به عنوان مثال، می‌توان مینیمم هزینه فیزیکی یا ماکزیمم آن و یا یک بازه بین مینیمم و ماکزیمم هزینه و یا تمام هزینه‌های ممکن برای آن واکنش را تعریف کرد. نشان دهید که به ازای هر یک از این موارد، A* (با تغییرات موردنیاز) می‌تواند راه‌حل بهینه را بیابد.**

☑ حل: یک مثال ساده را در نظر بگیرید. یک حالت‌باور اولیه $\{S_1,S_2\}$ و دو واکنش $a$ و $b$ داریم که هر کدام از این واکنش‌ها با شروع از هر حالت اولیه به هدف $G$ منتهی می‌شوند و داریم:

$$c(S_1,a,G)=3\,; \qquad c(S_2,a,G)=5\,;$$
$$c(S_1,b,G)=2\,; \qquad c(S_2,b,G)=6\,.$$

در این مورد، راه‌حل [a] دارای هزینه سه یا پنج است و راه‌حل [b] هزینه دو یا شش دارد. بنابراین در هر مورد مشاهده‌شده هیچ کدام بهینه نیستند. در برخی موارد، یک راه‌حل بهینه وجود خواهد داشت. یک مورد قطعی را درنظر بگیرید. در این مورد بایستی هزینه طرح نگاشت هر حالت فیزیکی اولیه به هزینه واقعی اجرای طرح اندیشیده شود. در مثال فوق، هزینه $[a]$ عبارت است از $\{S_1{:}3,S_2{:}5\}$ و هزینه $[b]$ عبارت است از $\{S_1{:}2,S_2{:}6\}$. می‌توان گفت طرح $p_1$ تسلط کمتری از $p_2$ دارد، اگر به ازای هر حالت اولیه، $p_1$ هزینه بیشتری از $p_2$ برآورد نکند. (همچنین $p_1$ بر $p_2$ تسلط دارد اگر برای آن حالت هزینه کمتری را برآورد کند). اگر یک طرح از تمام سایرین تسلط کمتری داشته باشد، آنگاه بهینه خواهد بود. لازم به ذکر است که این تعریف همان بهینگی معمولی در موردهای مشاهده‌پذیر است که هر حالت‌باور آن منحصربفرد باشد. طبق مثال‌های قبل، ممکن است مسأله‌ای در این مورد باشد که هیچ راه‌حلی نداشته باشد. شاید نسخه تغییریافته‌ای از A* بتواند هر راه‌حلی که توسط سایرین یافته نشده را بیابد. برای فهم چگونگی اعمال A* به همگی، دانستن ارتباط آن با اصول بهینگی Bellman(1957) کمک شایانی می‌کند: «یک سیاست بهینه دارای این ویژگی است که حالت‌اولیه و تصمیم‌گیری اولیه مشخص شده و سپس مابقی تصمیم‌گیری بایستی خود شامل یک سیاست بهینه باشد که از حالت بدست آمده از تصمیم‌گیری اولیه، ادامه می‌دهد.» مهم است که بدانیم این موضوع محدودیتی بر معیار کارآیی است. برای آسان کردن کارکرد الگوریتم ارائه شده است و یک تعریف عمومی از معنای بهینگی نیست. علی‌الخصوص، اگر هزینه یک طرح در فضای حالت‌باور به صورت مینیمم هزینه هر تشخیص فیزیکی تعریف شود، آنگاه قوانین Bellman نقض می‌گردد. برای تغییر مثال قبل، فرض کنید $a$ , $b$ از حالت $S_1$ به حالت $S_3$ و از حالت $S_2$ به حالت $S_4$ دسترسی دارند و سپس به هدف می‌رسند.

$$c(S_1,a,S_3)=6\,; \qquad c(S_2,a,S_4)=2\,;$$
$$c(S_1,b,S_3)=6\,; \qquad c(S_2,b,S_4)=1\,. c(S_3,a,G)=2\,; \qquad c(S_4,a,G)=2\,;$$
$$c(S_3,b,G)=1\,; \qquad c(S_4,b,G)=9\,.$$

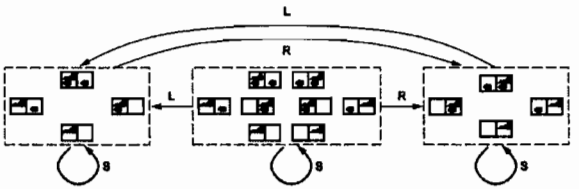
در حالت‌باور $\{S_3,S_4\}$، مینیمم هزینه $[a]$ همان $\min\{2,2\}=2$ و مینیمم هزینه $[b]$ همان $\min\{1,9\}=1$ می‌باشد. بنابراین طرح بهینه همان $[b]$ است. در حالت‌باور اولیه $\{S_1,S_2\}$ هزینه هر یک از چهار طرح ممکن به صورت زیر است.

$$[a,a]:\min\{8,4\}=4\,; [a,b]:\min\{7,11\}=7\,; [b,a]:\min\{8,3\}=3\,; [b,b]:\min\{7,10\}=7$$

بنابراین طرح بهینه در $\{S_1,S_2\}$ همان $[b,a]$ می‌باشد که نبایستی $b$ را در حالت $\{S_3,S_4\}$ انتخاب کرد حتی اگر در آن نقطه، طرح بهینه باشد. این رفتار دنباله مستقیم انتخابات مینیمم هزینه مسیر ممکن به عنوان معیار کارایی می‌باشد. این مثال تنها قسمت کوچکی از اتفاقات رخ داده در حالتی است که معیارکارایی غیرافزایشی باشد و جزئیات چگونگی تغییر روش A* برای تابع هزینه وابسته به مسیر در حالت کلی توسط Dechter and Pearl(1985) بیان شده است. که در آن بسیاری از جنبه‌های A* صرفنظر شد. به عنوان مثال هنوز می‌توانیم باندپایینی برای هزینه مسیر یک گره مشخص را بیابیم. برای یک حالت‌باور b، مینیمم مقدار $g(s)+h(s)$ برای هر حالت $s$ در $b$ همان باندپایینی مینیمم هزینه می‌باشد که از $b$ انجام شود.

**؟ ۱۰.۴ نسخه بدون حسگری از دنیای جاروبرقی سرگردان را در نظر بگیرید. فضای حالت باوری که از هر یک از حالات باور اولیه $\{1,2,3,4,5,6,7,8\}$ قابل دستیابی است ترسیم کرده و توضیح دهید که چرا این مسأله غیر قابل حل است؟ *(این تمرین تاحدودی مشابه تمرین ۳.۱۸ در ویرایش دوم است)***

☑ حل: فضای حالت باور را در شکل ۳.۴s مشاهده می‌کنید. هیچ راه‌حلی ممکن نیست زیرا هیچ مسیری وجود ندارد که به حالت باوری ختم شود که در آن تمام عناصر هدف باشند. اگر مسأله کاملا مشاهده‌پذیر باشد کارگزار می‌تواند با انجام دنباله‌ای از واکنش‌ها مانند مکیدن فقط در اتاق کثیف، به هدف برسد. در اینصورت بایستی رفتاری قطعی داشته و هر حالت کاملا قابل‌حل خواهد بود.




شکل ۳.۴s فضای حالت باور برای محیط جاروبرقی بدون‌حسگر که تحت قانون مورفی کار می‌کند.

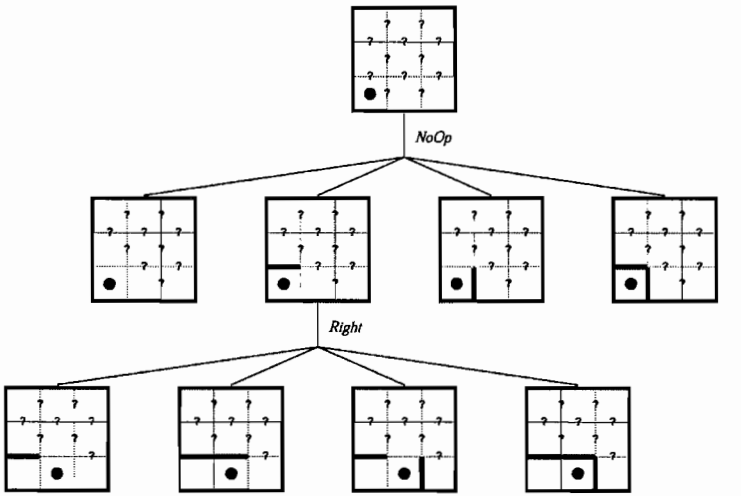
**۱۱.۴ می‌خواهیم به مسأله مسیریابی تمرین ۷.۳ در محیطی که در زیر توصیف شده است رجوع کنیم: • ادراکات کارگزار شامل لیستی از موقعیت‌ها است که نسبت به مکان کارگزار تا رئوس قابل‌مشاهده سنجیده می‌شود. ولی این ادراک هیچ اطلاعی موقعیت دقیق کارگزار نمی‌دهد و کارگزار بایستی مکان خود را از طریق نقشه بیاموزد. در این حالت فرض کنید که هر موقعیت ارای چشم‌اندازی متفاوت باشد. • هر واکنش برداری است که مسیری متشکل از خطوط را مشخص می‌کند. اگر مسیر مسدود باشد واکنش با موفقیت انجام می‌شود وگرنه کارگزار در اولین نقطه برخورد با مانع متوقف می‌شود. اگر کارگزار یک بردار حرکت یی برگردداند و در هدف قرار داشته باشد (که مختصات آنرا می‌دانیم)، بایستی کارگزار به یک مکان تصادفی پرتاب شود (غیر از قاط درون موانع). • معیار کارایی به این صورت محاسبه می‌شود که به ازای هر واحد از مسافت پیموده شده یک امتیاز و به ازای سیدن به هدف صد امتیاز تعلق بگیرد. الف) این محیط را به همراه کارگزاری برای آن طراحی کنید. پس از هر پرتاب کارگزار یستی مجدداً مسأله را تدوین کند که این کار شامل کشف موقعیت جاری نیز می‌باشد. ب) عملکرد کارگزار خود را گزارش هید (با تفسیرات مناسبی که کارگزار در طی حرکت خود تولید می‌کند) و کارایی آن را در طی 100 مرحله گزارش دهید. ج) حیط را به گونه‌ای تغییر دهید که کارگزار 30 درصد مواقع در نقطه‌ای که آن را اشتباها مقصد تصور می‌کند، متوقف می‌شود (که ین نقطه به طور تصادفی از بین رئوس پیش‌روی کارگزار انتخاب می‌گردد و در صورتی که این مقصد خیالی، یافته نشود هیچ اری انجام نمی‌شود). این مدل، تقریبا حرکت روبات واقعی را در محیط نشان می‌دهد که برخی نقاط را اشتباها مقصد تشخیص ی‌دهد. اکنون کارگزار را به نحوی تغییر دهید که هر زمان که متوجه اشتباه خود شد، بتواند به شیوه‌ای به محل تصمیم اشتباه زگشته و نقشهٔ خود را طبق همان نقشهٔ قبلی، اصلاح کند. به خاطر داشته باشید که در برخی مواقع، بازگشت مجدد به نقطه‌ای بلی باز هم منجر به شکست می‌شود! مثالی از موفقیت کارگزار در طی دو حرکت متوالی و خطادار ذکر کنید که باز هم به هدف رسد. د) اکنون دو روش بازیابی متفاوت پس از هر کشف خطا را در نظر بگیرید: ۱- به نزدیک‌ترین رأس در مسیر اصلی مراجعه نید. ۲- از مکان جدید، مجدداً یک مسیر به سوی هدف برنامه‌ریزی کنید. اکنون کارایی این 3 روش مختلف رفع خطا را مقایسه نید. آیا افزودن هزینهٔ جستجو بر این مقایسه تأثیرگذار است؟ هـ) اکنون فرض کنید در محیط مکان‌هایی با چشم‌انداز یکسان جود دارد. (مثلا فرض کنید جهان شطرنجی شده و موانع مربعی‌اند). اکنون کارگزار با چه نوع مسأله‌ای مواجه است؟ راه‌حل آن ه چه صورت است؟**

☑ حل: دانشجویان برای پاسخ به این سوال بایستی چندین طراحی ممکن را امتحان کنند. ابتدا اینکه رئوس اشیا چگونه شخص می‌شوند؟ در این مسأله، ادراکات شامل لیست موقعیت‌های رئوس هستند ولی به طور دقیق مشخص نمی‌شوند. یک تخاب خوب آن است که کارگزار یک عضو گردان برای خود داشته باشد (مانند یک سر که به درجات مختلف می‌چرخد). رئوس بل‌مشاهده در خلاف جهت عقربه‌های ساعت و با شروع از جهت اولیه سر کارگزار مرتب می‌شوند. هر رأس دارای یک زاویه (از صفر تا 360 درجه) و یک فاصله است. همچنین می‌خواهیم با عباراتی نقاط هر مانع را به طور دقیق مشخص کنیم بنابراین ادهای L، R یا I را به ترتیب برای نقاط روی لبه سمت چپ، لبه سمت راست، و نقاط داخلی مانع تعریف می‌کنیم. در این سمت نیاز است تا دانشجویان از برخی محاسبات‌هندسی استفاده کنند: برخورد کارگزار به مانع همان اشتراک مسیر و مجموعه طعه خطوط می‌باشد و همچنین برای فهم مشاهدات نیاز به محاسبات بینایی داریم. برای این موارد الگوریتم‌های کارایی وجود ارد که می‌توانند بر روی این مجموعه قطعه خطوط کار کنند ولی در مورد کارایی نگران نباشید زیرا یک الگوریتم جامع وجود ارد. اگر این کار را سخت می‌دانید از استاد مربوطه تقاضای یک شبیه‌ساز محیط کرده و شما فقط برنامه کارگزار را بنویسید. ای پاسخ قسمت ج، دانشجو بین زمان جستجو و زمان حرکت بایستی چندین بار سبک‌وسنگین کند و شاید کار بسیار دشواری شد که شبیه‌سازی آن بسیار به حقیقت نزدیک باشد. روش ساده‌تر آن است که به ازای هر محاسبه، جریمه‌ای لحاظ شود. برای سمت د، بایستی مجموعه‌ای از موقعیت‌های ممکن را نگهداری کند و هر زمان که کارگزار حرکت می‌کند بایستی تعدادی از آن وارد ممکن را با توجه به حرکتش از لیست حذف کند. کارگزار می‌تواند حرکت‌هایی را درنظر بگیرد که عدم‌قطعیت را کاهش هد به جای آنکه به هدف برسد.

**۱۲.۴ فرض کنید یک کارگزار در یک محیط پرپیچ‌وخم 3×3 که در شکل ۱۹.۴ می‌بینید قرار گرفته است. کارگزار می‌داند که وقعیت شروع آن در مختصات (1,1) و هدف در مختصات (3,3) بوده و مجاز است تا زمانیکه به دیوار برخورد نکرده است، به هات بالا، پایین، چپ و راست حرکت کند. کارگزار از مکان دیوارها باخبر نیست، بنابراین در نظر کارگزار تعداد $4096=2^{12}$ الت مختلف برای چیدمان‌های مختلف دیوارها که به آن پیکره‌بندی محیط گفته می‌شود، وجود دارد. هر حالت باور در نظر کارگزار، زیرمجموعه‌ای از این پیکره‌بندی‌های ممکن را شامل می‌شود. الف) توضیح دهید که این مسأله‌جستجوی‌آنلاین چگونه ی‌تواند به یک مسأله‌جستجوی‌آفلاین تبدیل شود. که در آن حالت باور شروع، شامل تمام پیکره‌بندی‌های ممکن محیط باشد. ندازه حالت باور شروع چقدر است؟ اندازه فضای حالت باور چقدر است؟ ب) چند ادراک مختلف در حالت شروع ممکن است؟ ) چند شاخه نخست از نقشه محتمل این مسأله را توصیف کنید؟ اندازه نقشه کامل چقدر است؟ توجه داشته باشید که این شه محتمل، راه‌حلی برای هر پیکره‌بندی در تعریف فوق محسوب می‌شود. بنابراین یکی درمیان بودن جستجو و اجرا حتی در حیط‌های ناشناخته خیلی موردنیاز نیست. *(این تمرین مشابه تمرین ۱۴.۴ در ویرایش دوم است)***

☑ حل: **الف)** با توجه به آنکه فرض شد تمام پیکره‌بندی‌های ممکن، در آغاز درون حالت باور کارگزار قرار داشته باشند پس با وجه به آنکه 12 مکان ممکن برای دیوارهای داخلی وجود دارد، پس حالت باور اولیه شامل مجموعه‌ای از تمام 4096 یکره‌بندی است (تمام چیدمان‌های مختلف محیط که کارگزار نمی‌داند کدام یک واقعیت دارد و در آغاز همه را در نظر ی‌گیرد). بنابراین کل فضای حالت باور به تعداد زیرمجموعه‌های این مجموعه یعنی شامل $2^{4096}$ حالت باور می‌باشد. (به ازای

هر زیرمجموعه از پیکره‌بندی‌ها، یک حالت حساب می‌شود ولی بسیاری از آنها قابل‌دستیابی نیستند). پس از هر مشاهده و واکنش انجام‌شده، کارگزار می‌آموزد که بین خانه جاری و همسایگان آن آیا دیواری وجود دارد یا خیر. بنابراین کارگزار می‌تواند هر حالت باور قابل‌دستیابی را با یکی از مقادیر (موجود، غیرموجود و ناشناخته) به طور مجزا توصیف کند. بنابراین حالت باور کاملا مشخص شده و تعداد $3^{12}$ حالت باور قابل‌دسترس وجود خواهد داشت. کارگزار در هر لحظه می‌تواند چهار سوی خود را مشاهده کرده و وجود یا عدم وجود دیوارها در آن خانه را بررسی نماید و با توجه به مشاهدات خود در هر خانه وضعیت کاملا برایش روشن می‌شود و دیگر نیازی نیست تا باورهای خیلی زیادی در مورد دیوارهای آن خانه حدس بزند پس تمام زیرمجموعه-های ممکن برای وضعیت هر خانه برابر 16 ($2^4$) خواهد بود و سایر حالات باور که با مشاهدات کنونی آن متفاوت باشد حذف می‌شوند زیرا در هر حالت باور 4 واکنش پیش‌رو دارد. **ب)** با فرض دانستن مکان دیوارهای خارجی، در لحظه شروع دو دیوار داخلی مشاهده گشته و در نتیجه $2^2=4$ حالت ممکن وجود دارد. **ج)** واکنش اولیه منجر به چهار حالت باور ممکن می‌شود که در شکل ۴.۴s نمایش داده شده‌است. در هر حالت باور، کارگزار یک واکنش انتخاب می‌کند تا به 8 حالت باور برسد (در زمان ورود به مربع وسط). با تکرار گام‌های حرکت این کارگزار در رسیدن به پایان، می‌بینیم که کارگزار کل محیط پرپیچ‌وخم را در حداکثر 18 مرحله سپری می‌کند. بنابراین نقشه کامل (به صورت یک درخت) بیشتر از $8^{18}$ گره نخواهد داشت. به بیانی دیگر، فقط $3^{12}$ گره وجود دارد بنابراین می‌توان نقشه را به صورت جدولی از واکنش‌ها بیان کرد که توسط حالت عقیده ایندکس شود.




شکل ۴.۴s مساله پر پیچ و خم 3×3. حالت آغازین، اولین مشاهده و یک اقدام کارگزار به همراه نتیجه آن.

**؟ ۱۳.۴ در این تمرین تپه‌نوردی را در مسأله هدایت روبات بررسی می‌کنیم. به عنوان مثال از محیط شکل ۳۱.۳ استفاده کنید. الف) تمرین ۱۱.۴ را با تپه‌نوردی تکرار کنید.آیا کارگزار شما در یک مینیمم محلی[15] گیر می‌کند؟ آیا ممکن است این کارگزار در گوشه (محدب) موانع گیر کند؟ ب) یک محیط با چندضلعی‌های بدون گوشه بسازید که کارگزار در آن گیر کند. ج) الگوریتم تپه-نوردی را به گونه‌ای تغییر دهید که جهت حرکت بعدی به جای انجام جستجویی با عمق یک، جستجویی با عمق $k$ انجام دهـد. این روش بایستی بهترین مسیر $k$ مرحله‌ای را یافته و در طول مسیر آن یک گام بردارد سپس این فرایند را مجددا تکرار کند. د) آیا مقدار $k$ای وجود دارد که در این الگوریتم جدید، فرار از مینیمم محلی را تضمین کند؟ هـ) توضیح دهید که در این مـورد، LRTA* چگونه در فرار از مینیمم محلی به کارگزار کمک می‌کند؟** ***(این تمرین مشابه تمرین ۱۷.۴ ویرایش دوم است)***

☑ حل: تپه‌نوردی روشی است که در یافتن مسیر قابل قبول در زمانی که مسیر بهینه وجود ندارد و هزینه محاسبه کـم اسـت، بسیار کارا است ولی در محیط‌های دوبعدی به تنهایی شکست می‌خورد. **الف)** این موضـوع کـه در شـکل ۵.۴s (a) بیـان شـده است، ممکن ولی بسیار بعید است. زیرا موانع بایستی یک شکل غیرمتعارف داشته و در مسیر هدف قـرار داشـته باشـند. **ب)** در محیطی با موانع گوشه‌دار احتمال گیر کردن بسیار زیاد است. (قسمت (b) در شکل ۵.۴s را ببینید.) **ج)** دقـت کنیـد کـه ایـن

[15] منظور از مینیمم‌سراسری یا مطلق، نقطه‌ای است که از کل نقاط یک نمودار کمتر است و مینیمم‌محلی، نقطه‌ای است که از نقاط همسایه خود کمتر است ولی از مینیمم‌سراسری بیشتر است. در مسائل جستجوی‌محلی، برخی الگوریتم‌ها این مینیمم‌های‌محلی را با سراسری اشتباه گرفته و در همانجا متوقف می‌شوند. معمولا مینیمم‌سراسری دارای $h$=0 و مینیمم‌های محلی دارای $h$ کوچک ولی بیشتر از صفر هستند.

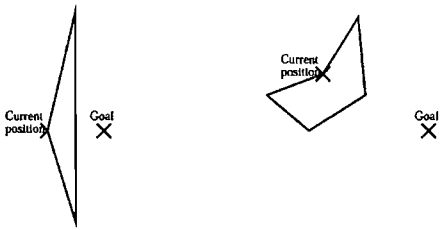
روش همان جستجوی عمق‌محدود است که در آن یک گام در مسیر بهینه انتخاب می‌کند. حتی اگر یک راه‌حل نباشد. د) اگر $k$ ا برابر حداکثر اضلاع چندضلعی در نظر بگیریم، آنگاه همواره فرار صورت می‌گیرد.

**؟ ۱۴.۴ روش DFS آنلاین مانند DFS برای فضاهای حالت قابل‌نقض که مسیرهای نامتناهی دارد، کامل نیست. به عنوان مثال رض کنید حالات همان نقاطی باشند که به تعداد بی‌نهایت در صفحه شطرنجی دوبعدی قرار دارند و واکنش‌ها بردارهایی با طول واحد به صورت (1,0)، (0,1)، (-1,0)، (0,-1)می‌باشند. نشان دهید که DFS آنلاین با شروع از مختصات (0,0) هرگز به (1,-1) دسترسی نمی‌یابد. فرض کنید کارگزار علاوه بر حالت فعلی خود می‌تواند تمام حالات پسین واکنش‌هایی که منجر به آنها می‌شوند را ببیند. الگوریتمی بنویسید که برای این فضای حالت که مسیر نامتناهی هم دارد کامل باشد.کدام حالت‌ها در طی رسیدن به (1,-1) ملاقات خواهند شد؟**

☑ حل: اگر بتوان حالت‌های پسین را دانست، آنگاه همواره می‌دانیم که چگونه به یک حالت قبلی بازگشت کنیم. این بدان معناست که می‌توانیم جستجوی عمیق شونده تکراری را برای حل این مسأله تغییر داده و بکار ببندیم. تنها تفاوت با روش پایه این است که بایستی عملیات بازگشت به عقب گنجانده شود.

Depth 1: $(0,0), (1,0), (0,0), (-1,0), (0,0)$

Depth 2: $(0,1), (0,0), (0,-1), (0,0), (1,0), (2,0), (1,0), (0,0), (1,0), (1,1), (1,0), (1,-1)$




شکل ۵.۴s (a) گیرافتادن در یک مانع گوشه‌دار (b)گیرافتادن در یک مانع غیرگوشه‌دار

# فصل ۵ (ویرایش سوم)

**؟ ۱.۵ فرض کنید شما یک پیشگو ، $OM(s)$ ،دارید که می‌تواند در هرحالت، حرکت رقیب را به طور صحیح پیش‌بینی کند. این تعریف بازی را برحسب یک جستجوی تک‌کارگزاره تدوین کنید. الگوریتمی برای یافتن حرکت بهینه بیابید.**

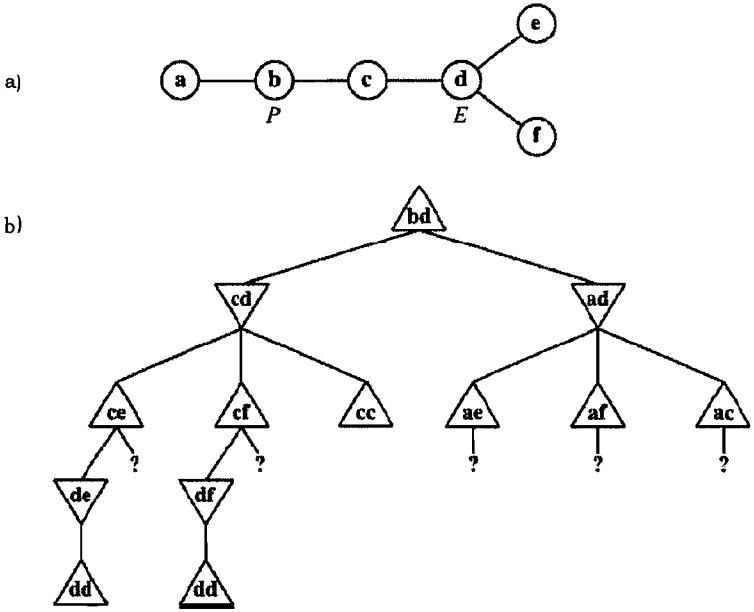
☑ حل: اقدامات رقیب را در قالب مدلی به نام $OM(s)$ معرفی می‌کنیم که شامل تمام اقدامات وی در بازی می‌شود. بنابراین اگر در حالت $s$ در بازی باشیم، اقدام رقیب را به صورت $OM(s)$ و حالتی که پیشگو پیش‌بینی کرده است را به صورت $P(s)$ نمایش می‌دهیم. البته ممکن است رقیب در زمان نوبت خود، چندین حرکت را قبل از نوبت ما، انجام دهد، بنابراین پیش‌بینی بایستی به طور بازگشتی انجام شود. اگر نوبت بازی ما باشد و در همین حالت $s$ بازی پایان یابد، آنگاه $P(s)=s$ وگرنه $P(s)=\text{RESULT}(s,OM(s))$ می‌باشد یعنی رقیب با توجه به حالت فعلی بازی ($s$) حرکتی را طبق مدل خود انجام می‌دهد. تدوین این مسئله به صورت یک مساله جستجو عبارتست از: حالت شروع: $P(s_0)$ که $s_0$ همان حالت اولیه بازی و $P(s_0)$ اولین حرکت رقیب است. اقدامات (واکنش‌ها): بسته به نوع بازی تعریف شده (مانند انداختن کارت، حرکت دادن یک مهره و یا پرتاب توپ) و با ACTION نمایش داده می‌شود. تابع پسین: $\text{RESULT}'(s,a) = P(\text{RESULT}(s,a))$ . آزمون هدف: هدف‌ها همان حالات پایانی هستند که در هر بازی طبق شرایطی مشخص تعیین می‌شود مانند مات‌کردن شاه در شطرنج. هزینه گام: هزینهٔ یک واکنش در صورتی صفر است که حالت به دست آمدهٔ $s'$ پایانی باشد. در غیر این صورت هزینه عبارت از $M$-$\text{UTILITY}(s')$ است که در آن $M=\max_s \text{UTILITY}(s)$ می‌باشد. توجه کنید که تمام هزینه‌ها غیرمنفی‌اند. لازم به ذکر است که فضای حالت در این مساله جستجو شامل حالاتی است که ما بازی می‌کنیم و موارد حالات پایانی و حالاتی که رقیب بازی می‌کند را درنظرنمی‌گیریم. البته می‌توانید این حالات را نیز درنظربگیرید ولی در آن صورت بایستی فقط یک اقدام انجام شود. برای این مساله می‌توانید **هر** کدام از الگوریتم‌های جستجوی فصل ۳ را بکار ببرید. به عنوان مثال اگر تمام بازی‌ها پایانی داشته باشند، جستجوی اول‌عمق می‌تواند برای حل این مساله بکار برده شود. همچنین اگر $OM(s)$ همواره بیشینه-کمینه حرکت را در حالت $s$ انجام دهد، می‌توان از الگوریتم بیشینه-کمینه در حالت پایه استفاده کرد.

**؟ ۲.۵ مساله حل دو پازل 8- تایی را درنظر بگیرید: الف) تدوین کاملی از این مساله به شیوه فصل سوم ارائه دهید. ب) چه مقدار از فضای حالت قابل دستیابی است. یک عبارت عددی دقیق ارائه دهید. ج) فرض کنید این مساله را به صورت زیر تغییر دهیم: دو بازیکن داریم که هر یک می‌توانند پازل خود را تغییر دهند که با انداختن یک سکه مشخص شود که نوبت کدام پازل است. در این بازی، برنده کسی است که بتواند پازل خود را زودتر تمام کند. در این مساله چه الگوریتمی برای انتخاب یک حرکت، می‌تواند استفاده شود؟ د) با یک اثبات غیررسمی نشان دهید که یک نفر در صورتی برنده می‌شود که هر دو بازیکن بهترین بازی خود را بدون هیچ نقصی ارائه دهند.**

☑ حل: **الف)** حالت شروع: دو پازل 8- تایی که در حالتی دلخواه قرار دارند. تابع پسین: انجام یک حرکت بر روی پازل حل نشده (می‌توانید اقدامی انجام دهید که بر روی هر دو پازل تاثیر بگذارد ولی بایستی تعریف کنید که اگر یک پازل حل شده و دیگری حل نشده باشد، چه اتفاقی رخ می‌دهد). آزمون هدف: هر دو پازل در حالت هدف باشند. هزینهٔ گام: یک واحد به ازای هر حرکت. **ب)** از آنجا که هر پازل 9 خانه دارد ، می‌توان این خانه‌ها را به $9!$ حالت مختلف چید. ولی از آنجا که نیمی از حالات غیر قابل دستیابی‌اند، هر پازل تنها دارای $9!/2$ حالت قابل دستیابی است. فضای حالت توام یعنی حالت‌های مختلف دو پازل دارای $(9!)^2/4$ حالت می‌باشد. **ج)** این مساله شبیه بازی تخته‌نرد است که به نظر می‌رسد بیشینه-کمینه بتواند در آن کارساز باشد. **د)** متن صورت سوال کامل نیست و بنابراین همان نسخهٔ قبلی مساله که در قسمت ج گفته شد را درنظر می‌گیریم. در اینصورت رقیب تمام سعی خود را می‌کند تا ما برنده نشویم و در نهایت با پرتاب سکه به دفعات زیاد، می‌توانیم یک پازل را بدون هیچ وقفه‌ای حل کنیم. برای بازی قسمت ج، حالتی را درنظر بگیرید که سکه بالا انداخته شده و شیر آمده است و پازل مربوطه تنها دو گام تا رسیدن به حالت هدف فاصله دارد. آیا شما می‌توانید در نوبت خود یک مرحله به هدف نزدیک‌تر شوید؟ اگر شما با انجام یک حرکت، پازل را یک گام به هدف نزدیکتر کنید آنگاه در صورتی که رقیب شیر بیاورد برنده خواهد شد بنابراین شما به برنده شدن رقیب خود کمک کرده‌اید. یا حتی اگر رقیب خط بیاورید، شما در نوبت خود مجددا خط بیاورید و دوباره رقیب سکه را پرتاب کرده و شیر بیاورد، باز هم رقیب برنده خواهد شد. در حالت کلی هر دنباله‌ای که در آن n بار خط آمده و سپس رقیب شیر بیاورد، شما بازنده شده‌اید. بنابراین احتمال برنده شدن شما حداقل $1/2+1/8+1/32+\ldots=2/3$ می‌باشد. بنابراین بنظر می‌رسد که بهتر است شما به جای نزدیک‌شدن به هدف، حرکتی انجام دهید که پازل را از هدف دور کند. (چون هیچ راهی برای ثابت نگه داشتن پازل در همان حالت ندارید و مجبورید حرکتی انجام دهید چه در جهت هدف و چه در خلاف آن). از دید رقیب به نظر می‌رسد در این شیوه بازی شما در طی مراحل مختلف، در حال خودکشی تدریجی بوده و خود را از صحنه رقابت کنار می‌کشید.

**؟ ۳.۵ تصور کنید در تمرین ۳.۳ به جای آن که هر دو دوست قصد رسیدن به هم را داشته باشند یکی از آنها قصد فرار از دست دیگری را دارد ولی دوست دیگر همچنان بدنبال اوست، این مساله شبیه به بازی «دزد و پلیس» می‌باشد. فرض می‌کنیم در این**

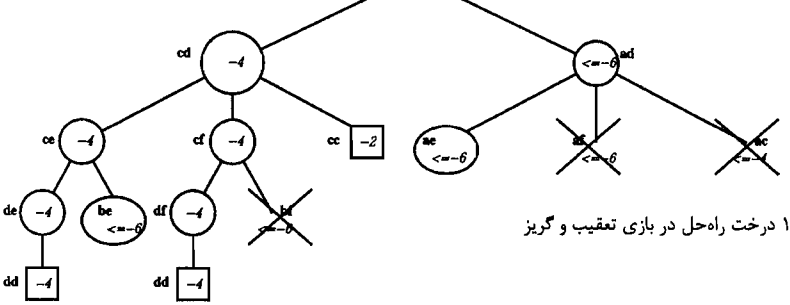
ی هر دو بازیکن می‌توانند حرکت کرده و بازی زمانی تمام می‌شود که دو بازیکن در یک گره به یکدیگر برسند (پلیس، دزد را برد). پاداش پلیس عبارت است از منفی کل زمان سپری شده و دزد درصورتی برنده است که دستگیر نشود. یک مثال از این ی را در شکل ۱۶.۵ می‌بینید.




ل ۱۶.۵ (a) نقشه‌ای که هزینه هر یال آن 1 میباشد. در ابتدا تعقیب‌کننده P در گره b و فرارکننده E در گره d قرار دارد. (b) قسمتی از خت بازی برای نقشه فوق. هر گره با موقعیت‌های P,E مشخص شده است. اول P حرکت می‌کند. شاخه‌هایی که با «؟» مشخص شده‌اند، وز بسط نیافته‌اند.

ب) درخت بازی را کپی کرده و مقدار هرگره پایانی را علامت‌گذاری کنید. ب) پس از هرگرهٔ داخلی، با استفاده از نامساوی‌هایی ن «14<» ویا «?» قواعدی قرار دهید که بتوان در مورد مقادیر عددی قسمت الف، استنتاج نمود. ج) در زیر هر علامت سوال، گره‌ای که توسط آن شاخه قابل‌دستیابی است را بنویسید. د) توضیح دهید که چگونه می‌توان یک مقدار گره در قسمت ج را کمک طول کوتاهترین مسیر نقشه محدود کرده و برای آن باند بیابیم. یادآوری می‌شود که هزینهٔ رسیدن به یک برگ با هزینهٔ ل آن برابر است. هـ) اکنون فرض کنید یک درخت مشخص که برگ آن طبق قسمت د محدود شده است، را از چپ به راست یابی کنیم. دور گره‌هایی با علامت سوال «?» که طبق باندهای قسمت د نیازی به بسط ندارند، دایره بکشید و روی گره‌هایی اصلا نیازی به بررسی ندارند خط بزنید. و) آیا می‌توانید در حالت کلی اثبات کنید که که در یک نقشه به شکل درخت، چه سی می‌برد؟

حل:




شکل ۱.۵s درخت راه‌حل در بازی تعقیب و گریز

**الف)** شکل ۱.۵s را ببینید. مقادیر فقط (منفی) تعداد مراحل سپری شده از ریشه می‌باشند. **ب)** شکل ۱.۵s را ببینید. توجه کنید که فرزند سمت چپ ریشه هم باند بالایی و هم باند پایینی دارد. **ج)** به شکل مراجعه کنید. **د)** طول کوتاهترین مسیر بین دو بازیکن، همان باند پایینی در کل زمان سپری شده می‌باشد. (در اینجا بازیکنان می‌توانند جابجا شوند بنابراین نیازی به تقسیم بر دو نمی‌باشد). بنابراین برای رسیدن به برگ‌هایی با علامت «؟» بایستی زمانی بزرگتر و مساوی با مجموع هزینه سپری شده از ریشه و طول کوتاهترین مسیر صرف شود. توجه کنید که این باند در زمانی بدست می‌آید که دزد خیلی بد بازی کند و مقدار صحیح هر گره در زمانی بدست می‌آید که هر دو بازیکن بهترین بازی خود را ارائه دهند. بنابراین می‌توانیم باند بهتری با فرض بازی بهتر ارائه دهیم. به عنوان مثال زمانی که دزد به جای آن که فقط به جلو بدود و پلیس دنبالش کند حرکات متوالی به سمت جلو و به سمت عقب انجام دهد، می‌توانیم باند بهتری از روی هزینه ارائه دهیم. **هـ)** به شکل مراجعه کنید (در اینجا از باندهای ساده استفاده کرده‌ایم). توجه داشته باشید که هرگاه بدانیم فرزند سمت راست مقداری کمتر از 6- دارد. نیازی نیست تا سایر پسین‌ها را درنظربگیریم. **و)** درصورتی که درخت بازی متناهی باشد، پلیس همواره برنده می‌شود. به منظور اثبات این ادعا فرض کنید گرهٔ جاری پلیس در ریشه درخت واقع شده باشد (درخت از آن گره شروع شده و سایر شاخه‌ها در زیر آن واقع هستند) این بدان معناست که پلیس در ابتدا حرکت خود را شروع می‌کند و سپس نوبت حرکت دزد خواهد بود. دزد نیز می‌خواهد در ریشه درخت باشد ولی در این مورد پلیس برنده شده است یا حداقل در یک زیردرخت برنده خواهد بود. در حالت کلی پلیس با توجه به حرکتی که دزد در نوبت خود کرده و یک شاخه را برمی‌گزیند، آن شاخه را دنبال می‌کند. این کار حداکثر $d$ بار تکرار می‌شود که $d$ همان حداکثر عمق زیر درخت می‌باشد یعنی هر بار دزد یک زیردرخت را انتخاب کرده و پلیس بدنبالش می‌آید. این تعقیب تا زمانی ادامه می‌یابد که پلیس یا بتواند دزد را بگیرد یا به یک گرهٔ برگ برسد. اگر پلیس بتواند دزد را بگیرد که برنده است و اگر به برگ برسد با توجه به اینکه برگ درخت هیچ زیردرختی برای فرار ندارد باز هم دزد را دستگیر کرده و برنده می‌شود. بنابراین در درخت بازی متناهی پلیس همواره برنده خواهد شد.

**؟ ۴.۵ توصیف حالات، مولدحرکت، آزمون‌پایانی، تابع‌سودمندی و تابع‌ارزیاب را برای یک یا چند مورد از بازی‌های ذکر شده توضیح داده و آنها را پیاده‌سازی نمایید: مونوپولی، اسکرابل، بازی بریج با قراردادی مشخص، و نوعی خاص از بازی پوکر.** ***(این تمرین مشابه تمرین ۱۰.۶ در ویرایش دوم است)***

☑ حل: هرحالت نشان‌دهنده وضعیت فعلی بازی (مانند حالت دنیا در مسائل فصول قبل) می‌باشد که می‌تواند به روش‌های مختلف فرمولی و یا ترسیمی توصیف شود که پس از انجام حرکت هر نفر، به حالت دیگری از بازی می‌رسیم. به عنوان مثال حالت در برخی بازی‌ها عبارتست از: در بازی شطرنج چیدمان مهره‌ها در صفحه، در پوکر که نوعی ورق‌بازی می‌باشد انواع پنج کارت موجود در دست بازیکنان. حالت فیزیکی پایه در این بازی‌ها بسیار ساده است. یکی از نکات مهم که لازم به یاد آوری است آن است که در بازی‌های بریج و اسکرابل، حالت فیزیکی برای تمام بازیکنان در دسترس نیست زیرا هر بازیکن نمی‌تواند تمام کارت‌ها را ببیند و شبیه‌ساز محیط نمی‌تواند به طور مستقیم به ازای هر بازیکن به آن دسترسی داشته باشد. در بازی بریج، هربازیکن حدس‌ها و فرضیه‌هایی نسبت به حالت واقعی محیط داشته و بهترین فرضیه را نگهداری می‌کند. با توجه به آنکه این بازی به طور آنلاین در بسیاری از سایت‌ها وجود دارد به احتمال زیاد کد مربوط به پیاده‌سازی آن نیز به آسانی یافته شود. منظور از آزمون‌پایانی شرایطی است که باید توسط بازیکنان ایجاد شود تا بازی خاتمه یابد که در بازی‌های مختلف متفاوت است. به عنوان مثال در شطرنج مات‌کردن شاه طرف مقابل و در بازی پوکر بیشترین مبلغی که بازیکنان شرط می‌بندند مانند مناقصه. تابع‌سودمندی روش تعیین امتیاز به هر حالت پایانی بازی است. به عنوان مثال در پایان بازی پوکر که بازیکنان کارت‌های خود را رو می‌کنند اگر کسی پنج کارت همرنگ داشته باشد نسبت به کسی که پنج کارت همرنگ و مرتب داشته باشد امتیاز کمتری می‌گیرد. در این بازی شیوه امتیازدهی در قالب جدولی از انواع چیدمان کارت‌ها و امتیاز آنها تعیین می‌شود. تابع‌ارزیاب به هر حالت غیرپایانی بازی، عددی را انتساب می‌دهد که نشان‌دهنده تخمینی از متوسط سودمندی آن بوده و برای گره‌های پایانی این تابع همان سودمندی را نشان می‌دهد (مشابه تابع ارزیاب فصول قبل). این تابع به بازیکنان کمک می‌کند تا حالت فعلی بازی را ارزیابی کرده و بدانند که این حالت به نفع کدام بازیکن است. در بازی شطرنج می‌توان تابع ارزیابی به این صورت درنظر گرفت که با توجه به مهره‌ها تعیین می‌شود: هر سرباز یک امتیاز، هر اسب یا فیل دو امتیاز و هر وزیر نه امتیاز. همچنین متدهایی تعریف کنیم که امنیت شاه نیز مقدار داشته باشد، ترکیب خطی این موارد یک عدد می‌شود که حالت فعلی بازی را تعیین می‌کند. مشخص است که اگر بازیکن سفید در صفحه شطرنج وزیر، دو فیل و دو اسب داشته و رقیبش تنها 8 سرباز را در صفحه داشته باشد، این وضعیت فعلی به نفع سفید خواهد بود پس تابع ارزیاب باید عددی را نشان دهد که این موضوع را اثبات کند. در اینجا این تابع ارزیاب برای بازیکن سفید عدد 17 و برای بازیکن سیاه عدد 8 را نشان می‌دهد. این عدد موقعیت بازی را برای دو طرف روشن می‌کند.

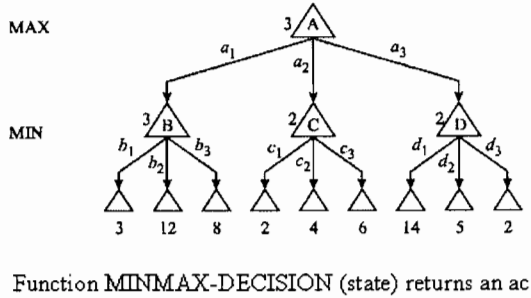
**؟ ۵.۵ محیطی از یک بازی چندنفره و بلادرنگ را توصیف و پیاده‌سازی کنید که در آن زمان بخشی از حالت محیط محسوب شود و بازیکنان برای انجام بازی فقط زمان محدودی در اختیار داشته باشند** ***(این تمرین مشابه تمرین ۹.۶ در ویرایش دوم است)***

☑ حل: دو مرحله از یک بازی بی‌درنگ دو نفره را درنظر بگیرید. بازیکن max بدنبال کسب بیشترین امتیاز و بازیکن min بدنبال کسب کمترین امتیاز هستند.

۱. از موقعیت جاری در گره max شروع کنید.

۲. درخت بازی را با تعداد ثابتی از برنامه‌ها گسترش دهید.

۳. تابع ارزیاب را به گره‌های برگ اعمال نمایید.
۴. از پایین به بالا، مقادیر بازگشتی را محاسبه کنید.
۵. حرکتی را انتخاب کنید که در ریشه بیشترین مقدار بدست آید.




الگوریتم:

```
Function MINMAX-DECISION (state) returns an action.
    Inputs:  State current state in game.
      V<-MAX-VALUE (state)
       return the action in SUCCESSORS (state) with value v.
      function MAX-VALUE (state) returns a utility value.
      If TERMINAL-TEST(State) then return UTILITY(State)
      V ← -∞
      for a, s in SUCCESSORS(State) do
       V← MAX(v, MIN-VALUE(s))
     return v
     function MIN-VALUE(State) returns a utility value.
      if TERMINAL-TEST(state) then returnUTILITY(state)
        V ← ∞
      for a, s in SUCCESSORS(state) do
        V←MIN(v,MAX-VALUES))
  Return s,
```

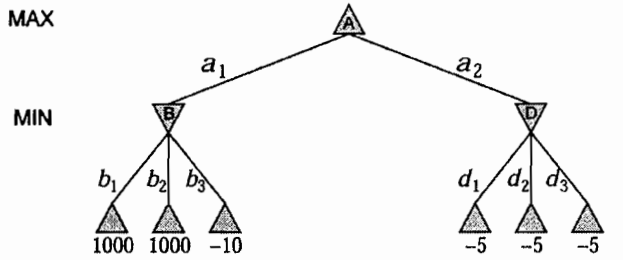
**؟ ۶.۵ توضیح دهید که چگونه می‌توان روش استاندارد بازی‌کردن را به بازی‌هایی چون تنیس، بیلیارد و کریکت که در یک فضای حالت فیزیکی پیوسته انجام می‌شوند، اعمال نمود.** ***(این تمرین مشابه تمرین ۴.۶ در ویرایش دوم است)***

☑ حل: یکی از واضح‌ترین تفاوت‌هایی که این بازی‌ها دارند، پیوستگی فضای اقدامات است. به عنوان مثال در بازی بیلیارد؛ جهت چوب بیلیارد، زاویه، سرعت و نقطه تماس آن با توپ بیلیارد همگی عناصری پیوسته‌اند که در نتیجه بازی دخیل هستند. ساده‌ترین راه‌حل آن است که این فضای پیوسته را گسسته نموده و سپس روش‌های بازی استاندارد را به آن اعمال نماییم یعنی برای هر عنصر محدوده مقادیر را تعیین نموده و سپس آن محدوده را به چندین زیرمحدوده تقسیم‌بندی نماییم و تعریف کنیم که اگر آن عنصر مقداری در این محدوده داشت، عددی ثابت را به آن تخصیص دهیم. این راه‌کار شاید برای بازی تنیس موثر باشد (بازی را فقط به عنوان شوت‌هایی دارای سرعت و جهت مدل نماییم) ولی برای بازی‌هایی نظیر بیلیارد و کراکت موثر نبوده و منجر به شکست می‌شود زیرا کوچکترین تغییر در زاویه، تاثیر بسیار زیادی در واکنش موردنیاز خواهد داشت. به عنوان مثال می‌توان تعریف کرد که اگر زاویه بین صفر تا 30 درجه بود فلان حرکت انجام شود. ولی در عوض با تحلیل بازی می‌توان مجموعه‌ای گسسته از هدف‌های محلی تشکیل داد مثلا در بازی بیلیارد «کسب چهار توپ» و در بازی کراکت «جمع‌کردن برای ست بعدی» جزء این هدف‌های محلی و گسسته محسوب می‌شوند. بنابراین یک روش بهینه‌سازی محلی می‌تواند هر هدف محلی را از روی یک مجموعه گسسته از انتخابات بیابد. معمولا این بازی‌ها تصادفی‌اند بنابراین مدل تخته‌نرد را به کار می‌بریم که در آن از نمونه‌برداری اتفاقات به جای مجموع تمام اتفاقات استفاده می‌کنیم. بازی‌های بیلیارد و کراکت به طور کامل در دسته بازی‌های turn-taking مدل می‌شوند ولی بازی تنیس اینگونه نیست. زمانی که یک بازیکن توپ را پرتاب می‌کند، بازیکن دیگر بلافاصله درحال حرکت است تا بتواند مقصد توپ را پیش‌بینی کرده و آنرا به رقیب بازگرداند. این امر باعث می‌شود تا بازی تنیس مشابه بازی‌های «واکنش همزمان» باشد. در این مورد می‌توان یک استراتژی تصادفی جهت پرتاب به کار گرفت تا رقیب نتواند مقصد توپ را پیش‌بینی کند.

**؟ ۷.۵ این ادعا را اثبات کنید: در هر درخت بازی، سودمندی حاصل از روش بیشینه-کمینه(MiniMax) برای بازیکن Max رصورتی که Min نیمه‌بهینه بازی کند، هرگز کمتر از حالتی نخواهد بود که Min بهینه بازی کند. آیا می‌توانید یک درخت بازی ثال بزنید که Max با یک استراتژی نیمه‌بهینه بتواند همچنان بهتر از Min با عملکرد نیمه‌بهینه بازی کند** ***(این تمرین مشابه تمرین ۲.۶ در ویرایش دوم است)***

☑ حل: در ابتدا لازم به یادآوری است که در مسائل بازی معمولا فرض می‌شود که دو بازیکن با نام‌های Max و Min به صورت وبت در میان بازی می‌کنند. این بازی می‌تواند هر نوع بازی مانند شطرنج، تخته‌نرد و. .. درنظرگرفته شود زیرا ما در اینجا ی‌خواهیم کارگزاری طراحی کنیم که بتواند در انواع بازی‌ها به طور خودکار عمل کند مانند بازی شطرنج در رایانه. در طی

بازی، بازیکن Max باید سعی کند امتیاز بگیرد (امتیاز مثبت) و بازیکن Min باید از گرفتن امتیاز طرف مقابل جلوگیری کند بنابراین هر امتیازی که Max بگیرد منفی آن برای Min محسوب می‌شود (مجموع امتیازات دو طرف صفر خواهد بود). اینگونه بازی‌ها تخاصمی نامیده می‌شود زیرا بازیکن Min با طرف مقابل خصومت دارد. معمولا دنباله حرکات مختلفی که در بازی میسر است و امتیاز هر حرکت را در قالب درختی با نام درخت‌بازی ترسیم می‌کنند که هر سطر آن نشان‌دهنده تمام حرکات مجاز برای یک بازیکن و امتیازات وی می‌باشد. منظور از بهینه بازی‌کردن Max آن است که در هر حرکت بهترین حرکت ممکن برای کسب امتیاز را انجام دهد و منظور از بهینه بازی کردن Min آن است که در هر حرکت بهترین کاری که موجب بیشترین کسر امتیاز از Max شود را انجام دهد. در این سئوال فرض شده است که Max بازی خود را انجام می‌دهد ولی بازیکن Min خیلی با وی خصومت ندارد. می‌خواهیم اثبات کنیم که داشتن خصومت کامل و یا خصومت کمتر او، هیچ تاثیری در امتیاز بازیکن Max ندارد. برای این منظور یک گرهٔ Min که فرزندان آن گره‌های پایانی درخت هستند را درنظر بگیرید. اگر Min به صورت نیمه‌بهینه بازی کند یعنی حرکتی انجام دهد که باعث شود بازیکن Max بیشتر از حداقل مقدار ممکن امتیاز بگیرد، آنگاه مقدار آن گره بزرگتر و مساوی با مقداری خواهد بود که Min به صورت بهینه بازی کند. از این رو، مقدار گرهٔ Max که پدرگرهٔ Min می‌باشد، نیز طبق الگوریتم بیشینه-کمینه افزایش می‌یابد. این روال به همین ترتیب تا ریشه ادامه می‌یابد و در ریشه مقداری بیشتر و یا مساوی با بهینه بازی‌کردن Min خواهیم داشت. اگر بازی نیمه‌بهینهٔ Min قابل پیش‌بینی باشد، آنگاه می‌تواند بهتر از یک استراتژی بیشینه-کمینه بازی کند. به عنوان مثال اگر Min همواره به دنبال نوع خاصی از تله و نیرنگ باشد آنگاه تنظیم کردن تله، برنده‌شدن را تضمین می‌کند حتی اگر پاسخ اشتباه به Min بدهیم. این موضوع در شکل ۲.۵s نمایش داده شده است.



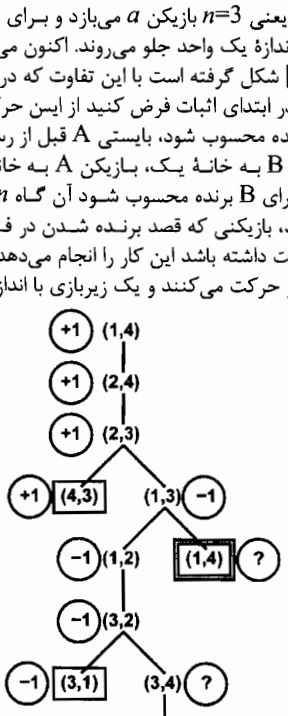

شکل ۲.۵s یک درخت بازی ساده که یک تله برای بازیکن MIN را نشان می‌دهد که درصورتی که این بازیکن حرکت $a_i$ را انتخاب کند، در آن می‌افتد. البته حرکت بیشینه-کمینه نیز $a_2$ با مقدار 5- می‌باشد.

**؟ ۸.۵ بازی دونفره در شکل ۱۷.۵ را درنظر بگیرید. الف) درخت بازی را با استفاده از این قوانین به طور کامل ترسیم کنید. • هرحالت را به صورت ($s_A$,$s_B$) تعریف کنید که درآن $s_B$ و $s_A$ نشان دهندهٔ موقعیت هر بازیکن هستند. • هرحالت پایانی را درون یک مربع و مقادیر بازی را درون یک دایره قرار دهید. • حالات حلقه‌ای (حالاتی که در مسیر پیموده شده از ریشه قبلا استفاده شده‌اند) را درون یک مربع دوخطی قرار دهید. اگر مقدار آنها هنوز مشخص نشده است، با گذاشتن «?» درون دایره آنها را مشخص نمایید. ب) اکنون برای هرگره مقدار بیشینه-کمینه را بنویسید (هم چنین در دایره‌ها). توضیح دهید که چرا و چگونه از علامت «?» استفاده می‌کنید؟ ج) توضیح دهید که چرا الگوریتم استاندارد بیشینه-کمینه، در مورد این درخت بازی شکست می‌خورد و به طور خلاصه توضیح دهید که چگونه می‌توان با استفاده از قسمت ب، آن را اصلاح نمود. آیا الگوریتم تغییر یافته شما، می‌تواند تصمیم بهینه را برای تمام بازی‌های دارای حلقه بیابد؟ د) این بازی 4- مربعی را می‌توان به حالت کلی $n$- مربعی تعمیم داد. یعنی خانه‌های $n>2$ اثبات کنیدکه اگر $n$ زوج باشد بازیکن $a$ می‌برد و اگر $n$ فرد باشد، می‌بازد. *(این تمرین مشابه تمرین ۳.۶ در ویرایش دوم است)***

شکل ۱۷.۵ موقعیت شروع یک بازی ساده. اول بازیکن A بازی می‌کند. دو بازیکن به طور نوبت چرخشی باز کرده و هر یک باید مهره خود را به فضای مجاور خالی، در یکی از جهات منتقل کند. اگر خانه مجاور توسط رقیب اشغال شده باشد، طرف مقابل باید مهره خود را با پرش از روی رقیب به خانه بعدی ببرد البته اگر آنجا خالی باشد. (به عنوان مثال اگر A در خانه 3 و B در خانه 2 باشد، آنگاه A باید به خانه 1 برود). بازی زمانی خاتمه می‌یابد که یکی از بازیکنان مهره خود را به انتهای صفحه بازی برساند. اگر بازیکن A موفق به رساندن مهره خود به خانه 4 شود، بازیکن A، 1+ امتیاز می‌گیرد. و اگر بازیکن B موفق به رساندن مهره خود به خانه 1 شود، بازیکن A، 1- امتیاز می‌گیرد.

☑ حل: **الف)** درخت این بازی به همراه تمام نمادها و مقادیر بیشینه-کمینه در شکل ۳.۵s نمایش داده شده است. لازم بذکر است که در الگوریتم بیشینه-کمینه از برگ‌ها به سمت ریشه حرکت کرده و در هر بار برای هر گره پدر با توجه به سودمندی فرزندانش باید عددی را به آن اختصاص دهیم. اگر آن گره پدر نوبت بازی Min باشد، بازیکن Min همیشه سعی می‌کند تا کمترین امتیاز حاصل شود پس مسلم است که اگر در آن موقعیت بخواهد از بین فرزندان انتخاب کند ،کمترین سودمندی را بر می‌گزیند. پس کمترین سودمندی از بین فرزندان انتخاب شده و برای آن گره به عنوان مقدار بیشینه-کمینه نوشته می‌شود. اگر

گره پدر در درخت بازی نوبت Max باشد از بین فرزندان بیشترین مقدار سودمندی را انتخاب خواهد نمود. پس آن را به عنوان مقدار بیشینه-کمینه در کنار گره یادداشت می‌کنیم. این کار به همین ترتیب تا ریشه ادامه می‌یابد و در کنار هر گره با توجه به نوبت بازی مقدارهای بیشینه-کمینه یادداشت می‌شود. سپس با حرکت از ریشه و با کمک مقادیر بیشینه-کمینه می‌توان به ترتیب بهترین انتخاب‌ها را در بازی انجام داد تا به هدف برسیم. **ب)** مقدار «?» با این فرض است که یک کارگزار از بین حالت برنده و حالت «?» همواره حالت برنده را انتخاب می‌کند. همچنین Min(-1,?)=-1 و Max(+1,?)=+1. درصورتیکه تمام پسین‌های یک حالت «?» باشند، آنگاه مقدار موردنظر همان «?» خواهد شد. **ج)** روش استاندارد بیشینه-کمینه بر مبنای الگوریتم اول‌عمق است بنابراین ممکن است دریک حلقه بی‌نهایت گیر کند. از این رو جهت اصلاح آن می‌توان حالت فعلی را با پشته حالات مقایسه نمود و در صورت تکراری‌بودن حالت، مقدار «?» را بازگرداند. انتشار «?» به همین شیوه صورت می‌گیرد. اگر چه از این علامت در اینجا استفاده کردیم ولی معمولا این کار انجام نمی‌شود زیرا مشخص نیست که نتیجهٔ مقایسهٔ «?» با یک موقعیت کنار گذاشته چیست و یا نمی‌دانیم که مقایسهٔ بین حالات برندهٔ متفاوت (مانند بازی تخته‌نرد) چگونه انجام شود. درنهایت، در بازی با گره‌های شانسی، چگونگی محاسبهٔ میانگین اعداد و «?» مشخص نیست. توجه داشته باشید که نباید با حالات تکراری به عنوان موقعیت‌های کنارگذاشته شده رفتار گردد. به عنوان مثال هر دو حالت (1,4) و (2,4) در درخت تکراری هستند ولی موقعیت‌های برنده محسوب می‌شوند. اگر درخت بازی بدون چرخه باشد، آنگاه الگوریتم بیشینه- کمینه می‌تواند این معادلات را با انتشار از برگ‌ها حل نماید. اگر درخت بازی دارای چرخه باشد، آنگاه بایستی روش‌های برنامه نویسی پویا که مورد استفاده قرار گیرد. (تمرین ۱۷.۷ قسمتی از این مساله را بررسی می‌کند). این الگوریتم‌ها می‌توانند به طور دقیق تعیین کنند که کدام گره یک مقدار خوش‌تعریف دارد (مانند این مثال) و یا یک حلقه بی‌نهایت است که هر دو بازیکن تمایل به ماندن در آن چرخه را دارند (یا شاید انتخاب دیگری ندارند). دراین مورد، بایستی قواعدی در بازی تعیین کرد تا مقدار را مشخص کند (در غیراین صورت بازی هرگز خاتمه نمی‌یابد). دربازی شطرنج به عنوان مثال، حالتی که سه بار اتفاق بیافتد (بنابراین بایستی برای هر دو بازیکن منع شود) کنار گذاشته می‌شود. **د)** این سوال اندکی مهارت‌آمیز طراحی شده است. یک روش اثبات آن، استقرار بر روی اندازهٔ بازی است. بدیهی است که در حالت پایه بازی یعنی $n$=3 بازیکن $a$ می‌بازد و برای $n$=4 بازیکن A می‌برد. برای تمام $n$>4 حرکات آغازین یکسان است: A و B هر کدام به اندازهٔ یک واحد جلو می‌روند. اکنون می‌توان فرض کرد که قسمتی از بازی با اندازهٔ $n$-2 در مربع‌هایی با شماره‌های [2,...,$n$-1] شکل گرفته است با این تفاوت که در این زیربازی هر بازیکن در خانه‌های 2, $n$-1 یک حرکت دیگر نیز می‌تواند انتخاب کند. در ابتدای اثبات فرض کنید از این حرکات اضافه صرف‌نظر می‌کنیم. مشخص است که اگر A بخواهد با رسیدن به خانهٔ $n$-2 برنده محسوب شود، بایستی A قبل از رسیدن B به خانهٔ 2، به خانهٔ $n$-1 برسد (طبق تعریف برنده شدن) . بنابراین قبل از رسیدن B به خانهٔ یک، بازیکن A به خانهٔ n می‌رسد بنابراین مقدار $n$ منجر به برنده‌شدن A گشت. طبق همین اثبات اگر $n$-2 برای B برنده محسوب شود آن گاه $n$ نیز برای B برنده خواهد بود. اکنون فرض کنید حرکات اضافه گفته شده نیز مجاز باشند، بازیکنی که قصد برنده شدن در فضای [2,...,$n$-1] دارد، هرگز به خانهٔ آغازین خود باز نمی‌گردد ولی اگر بازیکن قصد باخت داشته باشد این کار را انجام می‌دهد. این موضوع به‌خودی‌خود، باخت بازی را تخمین می‌کند. سایر بازیکنان به سادگی به جلو حرکت می‌کنند و یک زیربازی با اندازهٔ $n$-2$k$ مرحله به مرحله به خانهٔ آغازین بازنده نزدیک‌تر می‌شود.




شکل ۳.۵s درخت بازی برای بازی چهارخانه که در تمرین ۸.۵ بیان شد. حالات پایانی، درون مربع‌های یک خطی، و حالات حلقوی درون مربع‌های دوخطی ترسیم شده‌اند. برای هر حالت مقدار بیشینه-کمینه آن را درون دایره‌ای در کنار آن مشخص کرده‌ایم.

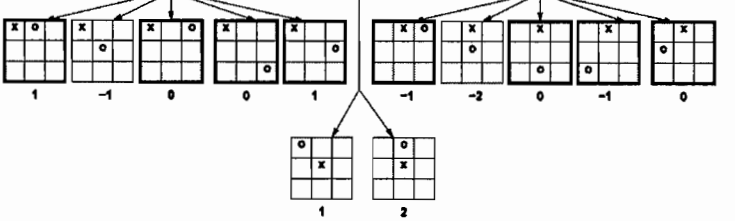
**؟ ۹.۵ این مساله مفاهیم اصلی بازی را به کمک یک بازی نظیر دوز تمرین می‌کند (گذاشتن علامت‌های ضربدر و دایره در خانه‌ها). $X_n$ را تعداد سطرها، ستون‌ها یا قطرهایی تعریف می‌کنیم که دقیقا $n$ تا $X$ درآنها بکار رفته باشد و هیچ نماد $O$ای در آنها دیده نشود. همچنین $O_n$ را تعداد سطرها، ستون‌ها و قطرهایی می‌گیریم که فقط نماد $O$ داشته باشند. تابع سودمندی به هر**

موقعیتی که $X_3$=1 باشد مقدار 1+ و هر موقعیتی که $O_3$=1 باشد مقدار 1- را انتساب می‌دهد. سایرموقعیت‌های پایانی دارای سودمندی صفر هستند. برای موقعیت‌های غیرپایانی، از یک تابع‌ارزیاب خطی به این صورت استفاده می‌کنیم:

$$Eval(s) = 3x_2(s)+x_1(s)-(3O_2(s)+O_1(s))$$

الف) تقریبا چند بازی دوز ممکن، وجود دارد؟ ب) درخت بازی را به طور کامل نمایش دهید. در ابتدا از صفحه‌ای خالی شروع کرده و سپس تا عمق دوم بازی را به طور متقارن در نظربگیرید (به عنوان مثال یک $X$ و یک $O$ در صفحه موجود باشد). ج) ارزیاب تمام موقعیت‌ها در عمق 2 را بر روی درختی که ترسیم کردید، نمایش دهید. د) با استفاده از الگوریتم بیشینه-کمینه، مقادیر موقعیت‌ها را برای عمق 0 و 1 روی درخت مشخص کرده و با استفاده از آن مقادیر، بهترین حرکت آغازین را انتخاب کنید. هـ) گره‌هایی که در عمق 2 بوده و توسط هرس آلفا- بتا ارزیابی نمی‌شوند را با کشیدن یک دایره دور آنها مشخص کنید. با این فرض که گره‌ها برای هرس آلفا- بتا به‌ترتیب بهینه مرتب شده‌اند. *(این تمرین مشابه تمرین ۱.۶ در ویرایش دوم است)*

☑ حل: طبق آنچه که در سئوال گفته شد، در صورتی در پایان بازی امتیاز 1+ کسب می‌شود که $X_3$=1 باشد یعنی یک سطر، یک ستون و یا یک قطر پیدا شود که در آن سه $X$ وجود داشته باشد (مانند دوز معمولی). می‌توان بازیکنی که مهره $X$ می‌گذارد را همان Max و بازیکن دیگر را Min نامید. تابع ارزیاب تعریف شده در هر بار به صفحه بازی عددی را انتساب می‌دهد. اگر آن عدد مثبت باشد یعنی آن حالت از بازی تا به نفع بازیکن Max (یا همان $X$) است و اگر عددی منفی شود به نفع بازیکن دیگر. می‌توانید این عدد این تابع را برای حالت‌های مختلف بازی بدست آورده و به همین نتیجه برسید (فرض کنید $X$ ها در مختصات [1,3] و [2,3] بوده و $O$ ها در مختصات [1,2] و [2,1] که عدد تابع ارزیاب 2+ خواهد شد). **الف)** تعداد 9 خانه داریم که هر دور بازی بدان معناست که بازیکنان این خانه‌ها را به ترتیب مختلف انتخاب کرده و مهره‌های خود را در آن قرار دهند. پس سئوال این است که به چند روش می‌توان 9 خانه را در کنار هم قرار داد و صفحه را پر نمود که جواب، حداکثر !9 می‌باشد. (البته قبل از پرشدن صفحه، چندین بار حالت‌های برد و باخت رخ می‌دهد.) برای قسمت‌های ب و هـ درخت بازی شکل ۱.۶s را مشاهده کنید که در زیر هر گرهٔ پایانی مقدار تابع ارزیاب و در سمت راست گره‌های غیرپایانی مقدار پشتیبان را ملاحظه می‌کنید. مقداری که نشان‌دهندهٔ بهترین حرکت آغازین است برای نمادهای $X$ در مرکز نوشته شده است. گره‌های پایانی که پررنگ شده اند، گره‌هایی هستند که با فرض مرتب بودن بهینه گره‌ها، نیازی به ارزیابی ندارند. **ب)** درخت بازی را در شکل زیر مشاهده می‌کنید که برای برگهای درخت، اعداد سودمندی در زیر آنها درج گشته است و برای گره‌های غیربرگ اعداد بازگشت به عقب را در سمت راست آنها ثبت نموده‌ایم. اعداد نشان میدهند که بازیکن Max بهتر است در ابتدا نماد ضربدر خود را در مرکز صفحه قرار دهد. برگهایی که با خطوط پررنگ مشخص شده‌اند گره‌هایی هستند که با فرض ترتیب بهینه، نیازی به ارزیابی ندارند. موقعیتهای ممکن برای درخت ارزیابی شده عبارتند از:



۴.۵s
قسمتی از درخت بازی دوز

**؟ ۱۰.۵ خانواده‌ای کلی از بازی دوز را طبق این تعریف در نظر بگیرید: هر بازی با مجموعه‌ای $S$ تایی از خانه‌ها و یک مجموعهٔ $W$تایی از موقعیت‌های برنده مشخص می‌شود و موقعیت برنده به صورت زیر مجموعه‌ای از $S$ تعریف می‌شود. به عنوان مثال در بازی استاندارد دوز، $S$ مجموعه‌ای 9 خانه‌ای و حالت‌های برنده می‌توانند هریک از 8 زیرمجموعهٔ زیر باشند: سه سطر، سه ستون و دو قطر. ازجنبهٔ دیگر، این بازی شبیه دوز استاندارد است. بازیکنی که هر خانه را در موقعیت برنده، علامت‌گذاری کند، برندهٔ بازی است. اگر تمام خانه‌ها علامت‌گذاری شده باشند و هیچ بازیکنی برنده نشود حالت تساوی رخ داده است. الف) فرض کنید $N=|S|$ که $S$ همان تعداد خانه‌ها باشد. یک باند بالایی برای تعداد گره‌ها در درخت بازی کامل در دوز کلی برحسب تابعی از $N$ بیابید. ب) یک باند پایینی برای اندازهٔ درخت بازی در بدترین حالت که $W=\{\}$ است بیابید. ج) یک تابع ارزیاب ممکن برای بازی دوز کلی بیابید که بتواند برای هر نمونه ای از این بازی بکاررود. این تابع می‌تواند برحسب $W$ و $S$ باشد. د) فرض کنید که می‌توانیم یک صفحهٔ جدید تولید کرده و برنده بودن یک حالت را با ماشینی با 100N دستورالعمل و پردازنده 2GHZ بررسی کنیم و هیچ محدودیتی از نظر حافظه نداریم. با استفاده از تخمین قسمت الف چه اندازه‌ای از درخت بازی می‌تواند به طور کامل توسط هرس آلفا- بتا در یک ثانیه از CPU حل شود. در یک دقیقه چقدر؟ دریک ساعت چقدر؟**

☑ حل: **الف)** در دوز با N خانه، حداکثر تعداد گره‌های پایانی در درخت، $N!$ می‌باشد که در واقع به ازای هر ترتیب ممکن از $N$ خانه یک عدد حساب شده است و یک باند پایینی برای تعداد کل گره‌ها (حداقل تعداد) عبارت از $\sum_{i=1}^{N} i!$ است، که این مقدار خیلی بیشتر از $N!$ به نظر نمی‌رسد ولی تابعی با رشد نمایی است. البته این باند کمی اغراق‌آمیز است زیرا برخی بازی‌ها در زمانی که یک موقعیت برنده رخ دهد، به زودی خاتمه می‌یابند. در محاسبه این تعداد، پس و پیش‌های داخلی محسوب نشده‌اند. یک باند بالایی برای تعداد حالات مجزا در بازی $3^N$ خواهد بود که در آن هر خانه یا خالی است و یا توسط نماد یکی از بازیکنان پرشده است. به یاد داشته باشید که با نگاه به صفحهٔ بازی می‌توان به طور قطعی تعیین نمود که هم اکنون نوبت کدام بازیکن است. **ب)** دراین مورد هیچ بازی‌ای زود هنگام خاتمه نمی‌یابد و به ازای هر کنار گذاشتن، $N!$ بازی مختلف وجود خواهد داشت. بنابراین با صرف نظر از حالات تکراری، دقیقا تعداد $\sum_{i=1}^{N} i!$ گره خواهیم داشت. پس از پایان بازی، تمام خانه‌ها بین دو بازیکن تقسیم شده‌اند ( $\lceil N/2 \rceil$ خانه برای بازیکن اول و $\lfloor N/2 \rfloor$ برای بازیکن دوم). بنابراین یک باند پایینی برای تعداد حالات مجزا عبارت $\binom{N}{\lceil N/2 \rceil}$ می‌باشد که نشان‌دهندهٔ تعداد حالات پایانی مجزا است. **ج)** فرض کنید به ازای حالت $s$ عبارت $X(s)$ نشان‌دهندهٔ موقعیت‌های برنده‌ای است که هیچ نماد $O$ نداشته و $O(s)$ نشان‌دهندهٔ موقعیت‌های برنده‌ای است که هیچ نماد $X$ ندارند. بنابراین یک تابع ارزیاب را می‌توان به صورت: $Eval(s) = X(s) - O(S)$ تعریف نمود. توجه داشته باشید که موقعیت‌های برنده که هیچ نمادی ندارند، در تابع ارزیاب لحاظ نمی‌شوند. البته می‌توان تابع ارزیاب دیگری توصیف نمود که به هر موقعیت مقداری را انتساب دهد که آن مقدار بسته به میزان نزدیکی آن حالت تا حالت برنده محاسبه شده باشد. **د)** با استفاده از باند بالایی محاسبه شده در قسمت الف یعنی $N!$ تعداد $100N!$ دستورالعمل میسر است. در پردازنده‌ای با 2GHz می‌توان دو بیلیون دستورالعمل را در یک ثانیه پردازش نمود. بنابراین برای Nهای بزرگ که حداکثر این تعداد دستورالعمل را شامل شوند، راه‌حل ممکن است. برای یک ثانیه، بایستی $N$=9 باشد، برای یک دقیقه $N$=11 و برای یک ساعت $N$=12.

**؟ ۱۱.۵ یک برنامه بازی در حالت کلی بنویسید که قابلیت بازی‌کردن در انواع گوناگون بازی را داشته باشد: الف) تابع مولد حرکت و تابع ارزیاب را برای یک یا چند بازی زیر پیاده‌سازی نمایید: Othello،Kalah، چکرز و شطرنج. ب) یک کارگزار عمومی آلفا- بتا بسازید. ج) تاثیرات افزایش عمق جستجو، بهبود ترتیب حرکت و بهبود تابع‌ارزیاب را با یکدیگر مقایسه کنید. ضریب انشعاب موثر تا چه حد به حالت ایده‌ال ترتیب حرکت نزدیک خواهد شد؟ د) یک الگوریتم جستجو به اختیار خود انتخاب کرده مانند جستجوی B* (Berliner ,1979)، جستجوی conspiracy number (MoAllester,1988) و یا MGSS* (Russell و Wefald، 1989) و سپس کارایی آن را با روش جستجوی A* مقایسه کنید. *(این تمرین اندکی مشابه تمرین ۴.۶ در ویرایش دوم است)***

☑ حل: توجه داشته باشید که یک محیط بازی اساسا یک محیط عمومی است که یک تابع بروزرسانی که بسته به قواعد هر بازی تعریف می‌شود، به آن اضافه شده است. و تا زمانی که نوبت حرکت یک بازیکن است بازیکن دیگر تغییری در محیط ایجاد نمی‌کند. می‌توانید با کمی جستجو، کدی برای تعریف یک بازی ساده (سخت‌تر از دوز) یافته و با استفاده از بیشینه-کمینه و آلفا- بتا نسبت به حل نسخه‌های کوچک این بازی اقدام کنید (احتمالا تا 3×4). البته توجه داشته باشید که روش آلفا- بتا بسیار سریع‌تر از بیشینه‌کمینه اقدام می‌کند ولی هنوز نمی‌توان بدون داشتن یک تابع ارزیاب و برش افقی آن را به مقیاس‌های بزرگتر تعمیم داد. ایجاد یک تابع‌ارزیاب تمرینی جذاب خواهد بود. از دید طراحی ساختمان‌داده این موضوع نیز جذاب است که با استفاده از پیش محاسبه سطرها، ستون‌ها و قطرها، چگونه می‌توان مولد حرکت را سرعت بخشید. تعداد کمی از دانشجویان در بازی Kalah ماهر هستند بنابراین انتسابی منصفانه داریم ولی این بازی کسل‌کننده است. عمق 6 به نظر مناسب است و یک تابع ارزیاب مناسب برای شکست بیشتر انسان‌ها کافی است. بازی اتلو جذاب بوده و سطح دشواری آن برای بیشتر دانشجویان مناسب است. بازی شطرنج و چکرز در بیشتر مواقع منصفانه صورت نمی‌گیرد چون ممکن است نیمی از دانشجویان کلاس ماهر بوده و نیمی دیگر مبتدی.

**؟ ۱۲.۵ توضیح دهید که در بازی‌های دو نفره که مجموع امتیاز غیرصفر است، هر بازیکن تابع سودمندی مجزا داشته ولی هر دو بازیکن از تابع سودمندی نفر دیگر باخبراست، الگوریتم‌های بیشینه-کمینه و آلفا- بتا چه تغییری می‌کنند؟ اگر هیچ محدودیتی بر روی دو سودمندی پایانی نداشته باشیم، آیا هر گره امکان هرس شدن توسط آلفا- بتا را داراست؟ اگر مقدار تابع سودمندی بازیکنان در هر حالت حداکثر به اندازهٔ ثابت $k$ متفاوت باشد چگونه می‌توان حداکثر مشارکت را داشت؟ *(این تمرین مشابه تمرین ۱۵.۶ در ویرایش دوم است)***

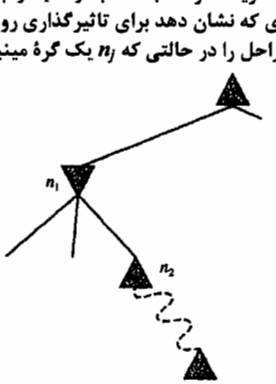
☑ حل: الگوریتم بیشینه-کمینه برای بازی‌های مجموع غیرصفر در این فصل توصیف شد که در آنجا تابع ارزیاب، برداری از مقادیر تعریف شد که هر مقدار آن متعلق به یک بازیکن است و مرحلهٔ پشتیبان، برداری را انتخاب می‌کند که برای بازیکنی که نوبت حرکتش است بیشترین مقدار را داشته باشد. در مثالی که در پایان بخش ۲.۲.۵ آورده شده توضیح داده شد که هرس

آلفا- بتا در حالت کلی در بازی‌هایی با مجموع غیرصفر ممکن نیست زیرا یک برگ بررسی نشده که توسط این الگوریتم کنارگذاشته می‌شود، ممکن است برای هر دو بازیکن بهینه باشد.

**؟ ۱۳.۵ اثباتی رسمی برای صحت عملکرد هرس آلفا- بتا ارائه دهید. برای این منظور موقعیت نشان داده شده در شکل ۱۸.۵ را درنظر بگیرید. اکنون سوال این است که آیا می‌توان گرهٔ $n_j$ که گره‌ای بیشینه بوده و فرزند گرهٔ $n_1$ است را هرس نمود. ایده اصلی این کار آن است که فقط در صورتی می‌توان هرس نمود که مقدار بیشینه -کمینهٔ آن مستقل از مقدار $n_j$ باشد. الف) مقدار $n_1$ که برحسب مقدار بیشینه -کمینه فرزندش حاصل می‌شود عبارت است از:**

$$n_1 = \min(n_2, n_{21}, \ldots, n_{2b_2})$$

**عبارتی مشابه برای $n_2$ و سپس عبارتی برای $n_1$ برحسب $n_j$ بیابید. ب) فرض کنید $l_i$ برای مقدار بیشینه (یا کمینه) گره‌های سمت چپ گرهٔ $n_i$ درعمق $i$ تعریف شود که مقدار بیشینه-کمینه آنها مشخص باشد. به طور مشابه فرض کنید $r_i$ برای مقدار کمینه (یا بیشینه) گره‌های بسط نیافته سمت راست گرهٔ $n_i$ در عمق $i$ تعریف شود. مجددا عبارت $n_1$ را بازنویسی کرده و از عبارات $l_i$ و $r_i$ استفاده کنید. ج) عبارت را مجددا فرموله کنید به طوری که نشان دهد برای تاثیرگذاری روی $n_1$، $n_j$ نباید از یک حد مشخص که از روی مقادیر $l_i$ به دست می‌آید، تجاوز کند. د) این مراحل را در حالتی که $n_j$ یک گرهٔ مینیمم باشد، تکرار کنید.** ***(این تمرین مشابه تمرین ۵.۶ در ویرایش دوم است)***




شکل ۱۸.۵ موقعیت مربوط به درنظرگرفتن هرس گره $n_j$

☑ حل: این تمرین به دشواری آنچه به نظر می‌رسد، نیست. عبارات زیر دقیقا منجر به تعاریف آلفا و بتا می‌شوند. نماد $n_i$ اشاره به مقدار گره‌ای در عمق $i$ دارد که در مسیری از ریشه تا برگ $n_j$ قرار دارد. گره‌های $n_{ib_i},\ldots,n_{i1}$ همگی همزاد (برادر)های گرهٔ $i$ هستند. **الف)** می‌توان تعریف کرد: $n_2 = \max(n_3, n_{31}, \ldots, n_{3b_3})$

داریم: $n_1 = \min(\max(n_3, n_{31}, \ldots, n_{3b_3}), n_{21}, \ldots, n_{2b_2})$

بنابراین زمانی که عبارتی شامل خود $n_j$ داشته باشیم $n_3$ می‌تواند جایگزین شود. **ب)** طبق اصطلاحات $r$ و $l$ داریم:

$$n_1 = \min(l_2, \max(l_3, n_3, r_3), r_2)$$

مجددا $n_3$ می‌تواند بسط یابد تا به $n_j$ برسد. داخلی‌ترین اصطلاح به صورت $\min(l_j, n_j, r_j)$ خواهد بود. **ج)** اگر $n_j$ یک گرهٔ بیشینه باشد، آنگاه حد پایینی بر روی مقدارش فقط در صورتی افزایش می‌یابد که پسین‌های آن ارزیابی شوند. واضح است که اگر این مقدار از $l_j$ تجاوز کند آنگاه هیچ تاثیری روی $n_1$ نخواهد داشت. در حالت گسترده‌تر اگر این مقدار از $\min(l_2, l_4, \ldots, l_j)$ تجاوز کند آنگاه دیگر هیچ تاثیری ندارد. بنابراین با نگهداری دنباله مقادیر آن می‌توان تصمیم گرفت که چه زمان $n_j$ را هرس نماییم. این مفهوم دقیقا همان عملکرد آلفا- بتا را نشان می‌دهد. **د)** حد مربوطه برای کمینه گره‌های $n_k$ عبارتست از:

$$\max(l_3, l_5, \ldots, l_k)$$

**؟ ۱۴.۵ اثبات کنید که هرس آلفا- بتا با فرض ترتیب بهینه حرکت‌ها دارای زمان $O(2^{m/2})$ است. که $m$ نشان‌دهندهٔ حداکثر عمق درخت بازی است.**

☑ حل: نتیجه در بخش ۶ از Knuth (1975) آورده شده است. عبارت دقیق (نتیجهٔ یک از تئوری یک) آن است که این الگوریتم تعداد:

$$b^{\lfloor m/2 \rfloor} + b^{\lceil m/2 \rceil} - 1$$

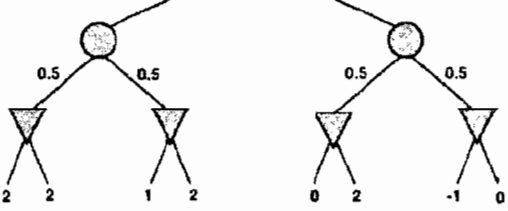
گره را در سطح $m$ بررسی می‌کند. این گره‌ها دقیقا همان گره‌هایی هستند که بازیکن Min در صورت انجام حرکت بهینه و یا بازیکن Max در صورت انجام حرکات بهینه به آنها دسترسی می‌یابد. این اثبات با استقرار روی $m$ انجام گشت.

**۱۵.۵ فرض کنید یک برنامه شطرنج دارید که قادر به ارزیابی ده میلیون گره در هرثانیه است. یک نمایش فشرده از حالت بازی برای ذخیره‌شدن در یک جدول جابجایی ارائه دهید. چه تعداد ورودی در جدولی با حافظه 2 گیگابایت می‌توان ذخیره نمود؟ آیا این مقدار حافظه برای سه دقیقه جستجو جهت یک حرکت، کافی است؟ در زمان هر ارزیابی، چه تعداد از خانه‌های جدول بررسی می‌شوند؟ اکنون فرض کنید این جدول را در یک دیسک ذخیره کرده‌ایم، در صورتی که سخت‌افزار دیسک استاندارد باشد، چه تعداد ارزیابی در زمان جستجوی دیسک انجام می‌شود؟** ***(این تمرین مشابه تمرین ۱۶.۶ در ویرایش دوم است)***

☑ حل: صفحه شطرنج 8×8 است که هر بازیکن 16 مهره و مجموع مهره‌های بازی 32 مهره می‌باشد. برای تعریف هر موقعیت از بازی بایستی موقعیت هریک از این 32 مهره را در 64 خانه معرفی کنیم. برای هر مهره بایستی مختصات(x,y) را بیان کنیم.

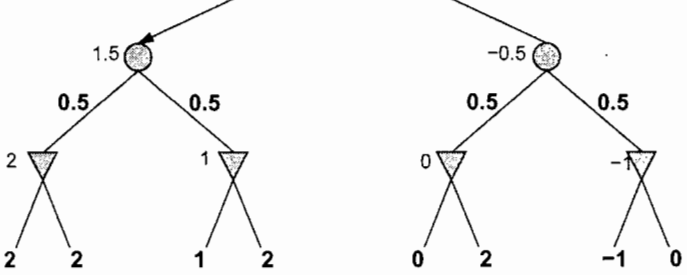
که هر یک از این x و yها از عدد 1 تا 8 مقدار می‌گیرند. از آنجا که اعداد 1 تا 8 را با سه بیت می‌توان نمایش داد. پس هر مختصات (x,y) به 6 بیت نیازمند است. پس بیان موقعیت 32 مهره به فضای 192=6×32 بیت یا 24 بایت نیازمند است. بنابراین هر موقعیت به 24 بایت فضا جهت ذخیره‌سازی نیاز دارد پس در جدولی با 2 گیگابایت، می‌توان 80 میلیون موقعیت را ذخیره نمود زیرا $2G\times2^{30}=2^{31}B$ و تقسیم این مقدار بر 24 عدد 80 میلیون را نتیجه می‌دهد (با صرف نظر از اشاره‌گرهای لیست درجدول‌های درهم). با توجه به این موضوع در طی سه دقیقه می‌توان 240 میلیون حالت را ذخیره نمود ولی طبق صورت سئوال در سه دقیقه 1800 میلیون گره ارزیابی می‌شوند که نسبت این دو مقدار 1.22 می‌باشد. برنامه‌های مدرن نظیر (Heinz, 2000) کلید درهم‌سازی هر موقعیت را گرفته و آن را به ازای هر موقعیت جدید تغییر می‌دهند. فرض کنید این کار حدوداً به اندازهٔ 20 عملگر پیچیدگی داشته باشد. بنابراین در ماشینی 2 گیگا هرتزی که برای یک ارزیابی حدود 2000 عملگر صرف می‌کند، می‌توان در هر ارزیابی، حدود 100 مراجعه انجام داد. با داشتن نموداری از یک میلی‌ثانیه از جستجوی دیسک، حتی می‌توانیم در هر مراجعه 1000 ارزیابی را انجام دهیم. واضح است که مقادیر موجود در جدول دیسک مقیم، مشکوک هستند حتی اگر جهت کاهش خواندن از دیسک، چندین مکان‌نما به مرجع اصلی ذکرکنیم.

**؟ ۱۶.۵ این تمرین هرس در بازی‌های دارای شانس را بررسی می‌کند. شکل ۱۹.۵ درخت بازی کامل را برای یک بازی نشان می‌دهد. فرض کنید که برگ‌ها به ترتیب از چپ به راست ارزیابی می‌شوند و قبل از بررسی یک برگ، هیچ اطلاعی از مقدار و محدودهٔ ممکن از مقادیر آن نداریم. الف) یک کپی از شکل برداشته و مقدار تمام گره‌های داخلی را مشخص کنید. همچنین بهترین حرکت از ریشه را با یک بردار نشان دهید. ب) با داشتن مقادیر شش برگ اولیه، آیا نیازی به بررسی برگ‌های هفتم و هشتم داریم، با داشتن مقادیر هفت برگ اولیه، آیا نیازی به بررسی برگ هشتم داریم؟ پاسخ خود را توضیح دهید. ج) فرض کنید که می‌دانید مقادیر برگ‌ها بین 2- و 2+ هستند پس از آنکه دو برگ اول بررسی شدند، محدودهٔ مقادیر برای گره شانس که در سمت چپ است، چه خواهد بود؟ د) دور تمام برگ‌هایی که تحت فرضیه قسمت ج نیازی به بررسی ندارند، دایره بکشید.**




شکل ۱۹.۵ درخت بازی کامل برای یک بازی ساده به همراه گره‌های شانس

☑ حل: **الف)** شکل ۵.۵s را ببینید. **ب)** با داشتن گره‌های 1 تا 6 دیگر نیازی به مراجعه به گره‌های 7 و 8 نداریم. اگر هر دوی این مقادیر ∞+ باشند آنگاه مقدار گرهٔ Min و گرهٔ شانس بالایی نیز ∞+ خواهد شد و بهترین حرکت متفاوت خواهد گشت. با داشتن گره‌های 1 تا 7 نیازی به مراجعه به گرهٔ 8 نداریم. حتی اگر آن برابر ∞+ باشد آنگاه گرهٔ Min نمی‌تواند بدتر از 1- شود. بنابراین گرهٔ شانس بالایی بدتر از 0.5- نخواهد گشت. بنابراین بهترین حرکت تغییر نمی‌کند. **ج)** بدترین حالت زمانی است که سومین و چهارمین برگ دارای 2- باشند که در این مورد گره شانس بالایی صفرخواهد بود. و بهترین حالت زمانی است که آنها مقدار 2 داشته باشند که گرهٔ شانس مقدار 2 خواهد داشت. بنابراین محدودهٔ موردنظر بین صفر تا 2 خواهد بود. **د)** به شکل مراجعه کنید:




شکل ۵.۵s هرسی شامل راه‌حل‌های گره‌های شانس

**؟ ۱۷.۵ الگوریتم‌های بیشینه-کمینه متوسط و آلفا- بتای ستاره‌دار که در Ballard(1983) بیان شد را جهت هرس درختان بازی با گره شانس پیاده‌سازی نمایید. سپس آنها را در بازی‌هایی چون تخته‌نرد امتحان کرده و کارایی هرس در روش آلفا- بتای ستاره‌دار را اندازه بگیرید.** ***(این تمرین مشابه تمرین ۶.۶ در ویرایش دوم است)***

☑ حل: پاسخ برعهده خواننده.

**؟ ۱۸.۵ ثابت کنید که توسط یک تبدیل خطی مثبت بر روی مقادیر برگ‌ها یعنی تبدیل یک مقدار $x$ به مقدار $ax+b$ که $a>0$ باشد در انتخاب حرکت از درخت بازی هیچ تاثیری نمی‌گذارد، حتی اگر این درخت شامل گره شانس نیز باشد.** ***(این تمرین مشابه تمرین ۷.۶ در ویرایش دوم است)***

☑ حل: استراتژی کلی آن است که با استفاده از استقراء روی عمق درخت، یک درخت بازی عمومی را به درختی تک‌لایه کاهش دهیم. مرحلهٔ استقراء را بایستی برای گره‌های Max ،Min و شانس انجام داده و سپس نشان دهیم که تبدیل مربوطه بر روی این گره‌ها انجام می‌شود. فرض کنید که مقدار فرزندان یک گره $x_1,x_2,\ldots,x_n$ باشند و تبدیل را به صورت $ax+b$ که $a>0$ است تعریف کنیم. آنگاه داریم:

$$\begin{aligned}
\min(ax_1+b, ax_2+b, \ldots, ax_n+b) &= a\min(x_1,x_2,\ldots,x_n)+b \\
\max(ax_1+b, ax_2+b, \ldots, ax_n+b) &= a\min(x_1,x_2,\ldots,x_n)+b \\
p_1(ax_1+b)+p_2(ax_2+b)+\cdots+p_n(ax_n+b) &= a(p_1x_1+p_2x_2+\cdots p_nx_n)+b
\end{aligned}$$

بنابراین کاهش درخت به درختی تک‌لایه که برگ‌های آن از مقادیر درخت اصلی برداشت شده باشند، تبدیلی خطی را نشان می‌دهد. بنابراین طبق $a>0 \Rightarrow ax+b>ay+b$ اگر $x>y$ بهترین انتخاب در ریشه با بهترین انتخاب در درخت اصلی یکسان خواهد بود.

**؟ ۱۹.۵ این مراحل را برای انتخاب حرکت در یک بازی دارای شانس درنظر بگیرید: • پرتاب‌های متوالی از پرتاب تاس (مثلا 50 بار) را تا رسیدن به عمقی خاص (مثلا 8) تولید کنید. • با دانستن اعداد روی هر تاس در هر مرتبه، درخت بازی قطعی می‌شود. به ازای هر دنباله از پرتاب‌های تاس، درخت بازی قطعی را به کمک آلفا- بتا حل کنید. • با استفاده از نتیجهٔ بدست آمده، مقدار هر حرکت را تخمین زده و بهترین آنها را انتخاب کنید. آیا این رویه به درستی کار می‌کند؟ چرا بله و چرا خیر؟** ***(این تمرین مشابه تمرین ۸.۶ در ویرایش دوم است)***

☑ حل: این رویه، نتایج نادرستی درپی‌دارد. به طور ریاضی با تشکیل دنباله پرتاب‌های تاس و تصمیم‌گیری بر مبنای آن، در واقع مقدار گره را بر اساس میانگین Min و Max می‌یابیم که در عمل این شیوه ممکن نخواهد بود. به طور شهودی این کار بدان معناست که انتخاب هر بازیکن در یک درخت قطعی معادل با داشتن دانشی کامل از آن چیزی است که تاس در آینده نشان خواهد داد. بنابراین نیازی نیست تا بازیکن حرکت خود را بر مبنای محیطی بدون‌دانش انتخاب کند زیرا آینده بازی را به طور کامل می‌داند. (به رابطهٔ بین مباحث بازی با کارت در بخش ۲.۶.۵ و مسائل کاملا و نیمه‌مشاهده‌پذیر در تصمیم‌گیری مارکوف توجه کنید). البته این روش بازی در عمل نتایج منطقی خوبی داشته و می‌تواند جهت مقایسه با بازی‌هایی که مجموع صفر دارند، بکار رود.

**؟ ۲۰.۵ درخت بیشینه، که فقط شامل گره‌های Max است و درخت میانگین‌بیشینه که علاوه بر گره‌های Max شامل گره‌های شانس نیز بوده و ریشهٔ آن Max است را درنظر بگیرید. در گره‌های شانس، تمامی احتمالات غیرصفر هستند و هدف، یافتن مقداری است که توسط جستجوی عمق‌محدود بدست می‌آید. برای قسمت‌های (الف) تا (و) در صورت صحت مثالی ذکرکنید و یا توضیح دهید که چرا غیرممکن می‌باشد. الف) فرض کنید مقادیر برگ‌ها متناهی ولی بدون‌حد است. آیا هرس یک درخت بیشینه ممکن است؟ ب) آیا در این شرایط درخت میانگین‌بیشینه در هر صورت هرس می‌شود؟ ج) اگر مقادیر تمام برگ‌ها غیرمنفی باشند، آیا همواره می‌توان درخت بیشینه را هرس نمود؟ مثالی ذکر کنید یا توضیح دهید که چرا اینگونه نیست. د) اگر مقادیر تمام برگ‌ها غیرمنفی باشند، آیا همواره می‌توان درخت میانگین‌بیشینه را هرس نمود؟ مثالی ذکر کنید یا توضیح دهید که چرا اینگونه نیست. هـ) اگر مقادیر تمام برگ‌ها در محدودهٔ [0,1] باشد، آیا همواره می‌توان درخت بیشینه را هرس نمود؟ مثالی ذکر کنید یا توضیح دهید که چرا اینگونه نیست. و) اگر مقادیر تمام برگ‌ها در محدودهٔ [0,1] باشد، آیا همواره می‌توان درخت میانگین‌بیشینه را هرس نمود؟ ی) ورودی یک گرهٔ شانس در درخت میانگین‌بیشینه را درنظر بگیرید. کدام یک از این ترتیب‌های ارزیابی، جهت بیشترین هرس مفید است؟ i) ابتدا کمترین احتمال. ii) ابتدا بیشترین احتمال. iii) هیچ تفاوتی ندارد.**

☑ حل: **الف)** هیچ هرسی انجام نمی‌شود. در یک درخت بیشینه مقدار ریشه همان مقدار بهترین برگ است ولی ممکن است برگ‌های مشاهده نشده بهتر باشند بنابراین بایستی آنها نیز بررسی شوند. **ب)** هیچ هرسی انجام نمی‌شود. یک برگ بررسی نشده ممکن است هر مقداری کمتر یا بیشتر از سایر برگ‌ها داشته باشد که (با فرض آن که احتمال هر ورودی غیرصفر باشد) بدان معناست که گره‌های Max و شانس که هنوز بسط داده نشده‌اند، هیچ محدودیتی از جهت مقدار ندارند. **ج)** هیچ هرسی انجام نمی‌شود. مشابه آنچه که برای قسمت الف گفته شد. **د)** هیچ هرسی انجام نمی‌شود. مقادیر غیر منطقی باعث ایجاد یک حد کمینه بر روی مقادیر گره‌های شانس می‌شود. ولی حد کمینه کمکی در هرس نمی‌کند. **هـ)** بله. اگر اولین پسین دارای مقدار یک باشد آنگاه ریشه نیز مقدار یک داشته و سایر پسین‌ها هرس می‌شوند. **و)** بله. فرض کنید اولین اقدام در ریشه مقدار 0.6

داشته باشد و اولین ورودی از اقدام دوم با احتمال 0.5 مقدار صفر داشته باشد. بنابراین سایر ورودی‌های اقدام دوم می‌توانند هرس شوند. **ز)** عبارت دوم که ابتدا بیشترین احتمال است، پاسخ سوال می‌باشد. زیرا این کار باعث می‌شود تا بیشترین حد بر روی مقادیر گره‌ها اعمال شود ولی برای سایر عبارات یکسان هستند.

**؟ ۲۱.۵ کدام یک از موارد زیر درست و کدام یک نادرست است؟ توضیح مختصری ارائه دهید. الف) در یک بازی چرخشی که مجموع امتیازات صفر بوده و محیط کاملامشاهده‌پذیر است، دو بازیکن با عملکرد کاملا عقلانی داریم که این شرایط محیطی کمکی به بازیکن اول در یافتن استراتژی بازیکن دوم نمی‌کند. پس از انجام حرکت بازیکن اول، نفر دوم بازی می‌کند. ب) در یک بازی چرخشی که مجموع امتیازات صفر بوده و محیط نیمه‌مشاهده‌پذیر است، دو بازیکن با عملکرد کاملا عقلانی داریم که این شرایط محیطی کمکی به بازیکن اول در داشتن حرکت بازیکن دوم نمی‌کند. ج) یک کارگزار کاملا عقلانی در بازی تخته‌نرد هرگز شکست نمی‌خورد.**

☑ حل: **الف)** صحیح است. بازیکن دوم همواره به طور بهینه بازی کرده و کاملا قابلیت پیش‌بینی دارد. بنابراین اگر در هربار دو حرکت خوب و تقریبا برابر ممکن باشد، این بازیکن حرکتی را انتخاب می‌کند که بازیکن اول نتواند در نوبت خود امتیازی کسب کند زیرا این بازیکن محیط را به طور کامل مشاهده می‌کند. **ب)** غلط است. زمانی که محیط نیمه‌مشاهده‌پذیر باشد بازیکن اول اطلاع کاملی از حالت واقعی بازی ندارد بنابراین پس از آنکه بازیکن دوم حرکت خود را انجام می‌دهد، اطلاعات بیشتری از حالت محیط را در اختیار وی می‌گذارد و به نوعی آنچه که خود می‌داند را به او منتقل می‌کند. به عنوان مثال در Kriegspiel دانستن حرکت بعدی رقیب تنها به بازیکن اول اطلاع می‌دهد که یکی از قطعات رقیب در چه مکانی واقع شده است. در یک بازی کارتی با انداختن کارت توسط رقیب، بازیکن اول تنها می‌فهمد که یکی از کارت‌های رقیب چه بوده است. مثلا بازیکن اول با توجه به دست خود تصمیم می‌گیرد بزرگترین کارت خود را انداخته و امتیاز بزرگی کسب کند ولی نمی‌داند که رقیب در دست خود کارتی بزرگتر از او را دارد بنابراین با اینکه به نظر خود کاری درست را انجام داده ولی امتیازی کسب نمی‌کند زیرا کار عقلانی آن بود که آن کارت را نمی‌انداخت. **ج)** غلط است. تخته‌نرد بعلت داشتن تاس یک بازی شانسی است. بنابراین ممکن است رقیب به طور مداوم در هر بار تاس ریختن، اعداد بهتری بیاورد بنابراین ممکن است بازیکن در هر بار با توجه به اعداد تاس خود، به طور کاملا عقلانی عمل کند ولی بازی را ببازد. بهتر است بگوییم این کارگزار به طور میانگین برای برنده‌شدن بهینه بازی می‌کند. البته هنوز مشخص نیست که اگر رقیب نیز بهینه بازی کند آیا بازیکن مقابل همچنان می‌تواند به طور میانگین برای برنده‌شدن بهینه بازی کند.

**؟ ۲۲.۵ تاثیر شانس و اطلاعات ناقص را در هر یک از بازی‌های تمرین ۴.۵ بدقت بررسی کنید. الف) برای کدام بازی، مدل استاندارد بیشینه-کمینه میانگین مناسب است؟ الگوریتم آن به همراه تغییرات موردنیاز برای محیط بازی را پیاده‌سازی کرده و بر روی کارگزار بازی اجرا کنید. ب) برای کدام بازی روش گفته شده درتمرین ۱۹.۵ مناسب است؟ ج) توضیح دهید که چگونه می‌توان با این موضوع که در برخی بازی‌ها بازیکن هیچ اطلاعی از حالت فعلی بازی ندارد برخورد نمود. *(این تمرین مشابه تمرین ۱۱.۶ در ویرایش دوم است)***

☑ حل: از رخدادهای شانسی در یک بازی می‌توان به پرتاب تاس اشاره نمود که نمی‌توان نتیجه نهایی را از بین اطلاعاتی از پیش‌مرتب‌شده مشخص نمود (مانند کارت‌های بدست آمده در یک دست از ورق‌بازی). نکته مهم آن است که بازیکنان بایستی بین اطلاعات آشکار بازی برای دو طرف، و اطلاعاتی که برای هر بازیکن متفاوت است تفاوت قائل شوند. **الف)** روش بیشینه-کمینه میانگین فقط برای بازی‌هایی مانند تخته‌نرد و مونوپولی مناسب است. در بازی بریج و اسکرابل، هر بازیکن از کارت‌های خود آگاه است ولی از کارت‌های رقیب خیر. در اسکرابل کار عقلانی آن است که به طور تصادفی درمورد وضعیت رقیب استنتاج اندکی کرده و حرکت کنیم ولی در بازی بریج دانستن اطلاعات رقیب در یک بازی خوب سهم اساسی دارد و بایستی به طور کامل انجام شود. **ب)** خیر به خاطر دلیلی که اخیرا ذکر شد. **ج)** نکتهٔ کلیدی در مورد استنتاج وضعیت رقیب آن است که تعدادی باور برای رقیب درنظرگرفته و بسته به تاثیر اقدامات مختلف، باورها را بروزرسانی کرده و با استفاده از یک شیوهٔ نمایش، این حالات‌باور را نمایش دهیم. از آن جا که این حالات‌باور تمام حالات فیزیکی ممکن برای کارگزار عقلانی را دربرمی‌گیرند این کار مفید خواهد بود.

# فصل ۶ (ویرایش سوم)

**؟ ۱.۶ تعداد راه‌حل‌های مسأله رنگ‌آمیزی در شکل ۱.۶ چقدر است؟ اگر مجاز به استفاده از چهار رنگ در این مسأله باشیم، چه تعداد راه‌حل وجود دارد؟ با دو رنگ چطور؟** ***(این تمرین مشابه تمرین ۲.۵ در ویرایش دوم است)***

☑ حل: برای رنگ‌آمیزی نقشه استرالیا با سه رنگ، تعداد 18 راه‌حل وجود دارد. با شهر $SA$ شروع می‌کنیم که می‌تواند هر یک از آن سه رنگ را داشته باشد. سپس در خلاف جهت عقربه‌های ساعت حرکت کرده و به سراغ شهر $WA$ که دو رنگ مجاز از سه رنگ دامنه‌اش را می‌تواند بگیرد می‌رویم. با رنگ‌آمیزی این دو شهر تکلیف رنگ سایر شهرها به طور دقیق مشخص می‌شود. پس تعداد راه‌حل‌ها به تعداد روش‌های مختلف رنگ‌آمیزی دو شهر $SA$ و $WA$ بستگی دارد که عبارتند از {(r,g),(r,b),(g,r),(g,b),(b,r),(b,g)}. از آنجا که شهر $T$ نیز مستقل از رنگ سایر شهرهاست، به 3 حالت مختلف رنگ می‌شود پس مجموعا به تعداد $6\times3=18$ روش رنگ‌آمیزی مختلف وجود دارد.

**؟ ۲.۶ مسأله قرار دادن $k$ اسب در یک صفحه شطرنج به طوری که هیچ دو اسبی یکدیگر را تهدید نکند درنظر بگیرید. مقدار $k$ مشخص و $k<n^2$ می‌باشد. الف) این مسأله را به صورت CSP تدوین کنید. در تدوین شما، متغیرها کدامند؟ ب) مقادیر ممکن هر متغیر چیست؟ ج) کدام مجموعه از متغیرها شرط محدودیت دارند و شرط محدودیت آنها چیست؟ د) اکنون مسأله را اینگونه درنظر بگیرید که حداکثر تعداد ممکن اسب را در صفحه قرار دهیم ولی هیچ یک، دیگری را تهدید نکند. توضیح دهید که چطور می‌توان این مسأله را با استفاده از جستجوی محلی که توابع RESULT، ACTION و شی‌گرایی آنها را به طور مناسب تعریف کرده‌ایم، حل کرد.**

☑ حل: **الف)** راه‌حل A: هر یک از $n^2$ موقعیت ممکن روی صفحه شطرنج، یک متغیر محسوب می‌شود. راه‌حل B: هر یک از اسب‌ها یک متغیر محسوب می‌شوند. **ب)** راه‌حل A: مقدار هر متغیر یکی از دو مقدار مجموعه {اشغال شده، خالی} می‌باشد. راه‌حل B: هر متغیر مجموعه‌ای از خانه‌ها را در دامنهٔ خود دارد. **ج)** راه‌حل A: وقتی اسب یک حرکت انجام می‌دهد خانهٔ شروع حرکت و خانهٔ مقصد دارای شرط محدودیت هستند و آن این است که هیچ یک نباید اشغال باشند. همچنین برای کل خانه‌ها نیز محدودیتی داریم که تعداد کل خانه‌های اشغال شده بایستی $k$ باشد. راه‌حل B: بین هر دو اسب محدودیت وجود دارد و آن این است که نباید هر دو در یک خانه باشند و همچنین نباید نسبت به هم در خانه‌ای واقع باشند که با انجام یک حرکت به دیگری برسند. راه حل B نسبت به راه حل A بهتر است زیرا یک محدودیت کلی نداریم. البته فضای حالت راه‌حل اول در زمان بزرگ بودن $k$ کوچکتر خواهد بود. **د)** هر راه‌حلی که ارائه شود بایستی یک تدوین حالت کامل را در نظر گیرد زیرا قصد استفاده از روش‌های جستجوی محلی را داریم. برای روش جستجوی شبه‌تاب‌کاری، تابع پسین بایستی به طور کامل فضا را به هم متصل کند، برای روش جستجوی شروع مجدد تصادفی حالت هدف بایستی توسط تپه‌نوردی از هر حالت شروعی قابل دستیابی باشد. دو روش اصلی برای راه‌حل این مسأله عبارتند از: راه‌حل C: در هر زمان مطمئن شوید که هیچ تهدیدی صورت نگرفته است. هر اقدام عبارتست از حذف اسبی که تهدید می‌شود و افزودن یک اسب در یک خانهٔ امن و یا آنکه یک اسب را به خانه‌ای جدید و امن منتقل کنیم. راه‌حل D: اجازه می‌دهیم تهدید موجود باشد ولی سعی در خلاص‌شدن از آن را داریم. هر اقدام عبارتست از حذف یک اسب و افزودن یک اسب در هر خانه یا حرکت یک اسب به یک خانهٔ جدید.

**؟ ۳.۶ مسئله ساخت (و نه راه‌حل) جدول کلمات متقاطع را درنظر بگیرید که منظور از این جدول، قراردادن تعدادی کلمه در یک شبکه مستطیلی شطرنجی می‌باشد[16]. دراین شبکه تعدادی از خانه‌ها بایستی خالی بوده و کلمات درآن قرار گیرند و تعدادی دیگر از خانه‌ها هاشور بخورند. فرض کنید لیستی از کلمات نظیر قسمتی از دیکشنری را داریم و می‌خواهیم با استفاده از روشی مشخص زیرمجموعه‌ای از این کلمات را درون خانه‌های خالی این جدول بگنجانیم. این مساله را به دو روش زیر تدوین کنید: الف) به عنوان یک مسئله جستجوی عمومی، یک الگوریتم جستجوی مناسب انتخاب کرده و درصورت نیاز، یک تابع اکتشافی نیز تعیین نمایید. بهتر است درخانه‌های خالی در هر بار یک کلمه قرار دهیم یا در هر بار یک حرف؟ ب) به عنوان یک مساله ارضای‌محدودیت، متغیرهای این مساله باید حروف باشند یا لغات؟ به نظر شما کدام تدوین بهتر است؟ چرا؟** ***(این تمرین مشابه تمرین ۴.۵ در ویرایش دوم است)***

☑ حل: **الف)** جدول کلمات متقاطع را می‌توان به روش‌های زیادی حل نمود که یکی از ساده‌ترین روش‌ها، جستجوی اول‌عمق است. یعنی در هر بار یک لغت را در رشته‌ای از خانه‌های خالی قرار می‌دهیم و به سراغ لغت بعدی می‌رویم. اگر به جایی رسیدیم که نتوانیم لغت بعدی را قرار دهیم (به علت سازگار نبودن با تعداد خانه‌ها خالی یا حروف مشترک) بایستی به سراغ آخرین انتساب رفته و آن را تغییر دهیم. اگر در تغییر آن نیز با مشکل روبرو بودیم باز به سراغ یک مرحله قبل‌تر رفته و آن را تغییر می‌دهیم. این رفتن به عمق و بازگشت به عقب طبق جستجویی اول‌عمق انجام می‌شود. همچنین بهتر است در هر بار یک لغت

---

[16] گینز برگ درسال 1990 چندین روش ساخت جدول کلمات‌متقاطع را بیان کرد و لیتمن (سال 1999) یافتن راه‌حل آنها که دشوارتر است را انجام داد.

ر شود تا یک حرف، زیرا تعداد مراحل موردنیاز کاهش می‌یابد. **ب)** از دید مساله ارضای‌محدودیت نیز همچنان راه‌حل‌های یادی وجود دارد. برای تدوین مساله به CSP باید سه مجموعه متغیر، دامنه و محدودیت را بیابیم. می‌توان برای هر خانه جدول، یک متغیر با دامنه یک حرف تعریف نمود و باید تعداد زیادی محدودیت تعریف شود که شرط بگذارد این حروف تشکیل یک کلمه را بدهند. در این راه‌کار بایستی قوانین محدودیت زیادی تعریف شود. به عنوان راه‌کاری دیگر، می‌توان مجموعه خانه-های متوالی قرار گرفته در یک سطر یا ستون را به عنوان یک متغیر درنظر گرفته و دامنه متغیرها همان لغات دیکشنری با آن سایز مربوطه باشند. از قوانین محدودیت می‌توان به این مورد اشاره نمود که در زمان تداخل دو کلمه بایستی حروف مشترک وجود داشته باشد. راه‌حل این تدوین نیاز به گام‌های کمتری دارد ولی دامنه متغیرها بزرگتر بوده (مخصوصا اگر دیکشنری بزرگی داشته باشیم) ولی در عوض محدودیت‌ها کمتر هستند. البته در هر صورت هر دو تدوین صحیح بوده و می‌توانند یک جدول را بسازد.

**؟ ۴.۶ برای هریک از این مسائل ارضای محدودیت، یک تدوین دقیق بیابید: الف) برنامه‌ریزی کف اتاق با خطوط مستقیم: می‌خواهیم در یک مستطیل بزرگ (مانند اتاق) تعدادی مستطیل کوچک‌تر (مانند موزاییک) قرار دهیم به طوری که هیچ‌یک از آنها با دیگری هم‌پوشانی نداشته باشد. ب) زمان‌بندی کلاس‌ها: تعداد مشخصی کلاس و استاد دریک دانشگاه، لیستی از کلاس-های ارائه شده و لیستی از زمان‌های ممکن برای هرکلاس موجود است و می‌دانیم که هر استاد می‌تواند مجموعه‌ای از کلاس‌ها را تدریس کند. ج) دور همیلتنی: با داشتن شبکه‌ای از شهرها که توسط تعدادی جاده به هم متصل شده‌اند، می‌خواهیم ترتیبی از ملاقات شهرها بیابیم که تمام شهرها سپری شده و هیچ‌یک تکرار نشوند. *(این تمرین مشابه تمرین ۵.۵ در ویرایش دوم است)***

☑ حل: **الف)** همان طور که گفته شد برای تدوین یک مساله به صورت CSP باید سه مجموعه متغیر، دامنه، و محدودیت را تعیین کنیم. برای موزاییک کردن کف اتاق، می‌توان برای هر موزاییک یک متغیر در نظر گرفت که دامنه آن، مجموعه‌ای از رشته‌های چهارتایی اعداد جهت تعریف مکان موزاییک می‌باشد. که این 4 عدد نشان‌دهنده دو مختصات $(x,y)$ گوشه‌های اصلی مستطیل یعنی نقطه بالا- چپ و نقطه پایین- راست می‌باشد. راه‌حل این مساله مجموعه‌ای از رشته‌های 4- تایی است که هر رشته، با ابعاد موزاییک‌ها همخوانی داشته و مجموعه مختصات‌ها کل اتاق را بپوشانند. محدودیت‌ها عبارتند از: هیچ دو رشته‌ای با هم همپوشانی نداشته باشند، به عنوان مثال اگر مقدار متغیر $R_1$ به صورت [0,0,5,8] باشد، آنگاه هیچ متغیر دیگری نباید مقداری بگیرد که با این موزاییک از نظر مختصاتی تداخل داشته باشد. **ب)** برای زمانبندی کلاس‌ها، می‌توان به ازای هرکلاس موجود در دانشگاه، سه متغیر تعریف نمود؛ یکی برای زمان (با مقادیری نظیر(MWF9:00 , TUTH8:00 , MWF 8: 00 ، یکی برای مکان کلاس (با مقادیری نظیر Wheeler 110, Evans330, ...) و یکی برای مدرس کلاس (با مقادیری نظیر ... Abelson, Bibel, Canny, ...). محدودیت این مساله آن است که در زمانی مشخص فقط یک کلاس می‌تواند در مکانی مشخص قرار گیرد و هر مدرس فقط در یک زمان می‌تواند یک کلاس را تدریس کند. البته می‌توان بسته به شرایط مساله محدودیت‌های بیشتری نیز تعریف کرد (به عنوان مثال یک مدرس نمی‌تواند دو کلاس پشت سر هم داشته باشد و... . ). **ج)** در مسأله دور همیلتنی، یک تدوین ممکن آن است که به ازای هر توقف در شهر یک متغیر در نظر بگیریم که شرط محدودیت اینری بوده و آن این است که شهرهای همسایه بایستی توسط جاده به هم متصل باشند همچنین در اینجا محدودیت All dif یعنی متفاوت بودن مقدار تمام متغیرها برقرار است.

**؟ ۵.۶ مساله حساب رمزی[17] شکل ۲.۶ را به شیوه‌های دستی، بازگشت به عقب، وارسی پیش‌رو و آروین‌های MRV و مقدار اکمترین‌محدودیت حل کنید. *(این تمرین مشابه تمرین ۶.۵ در ویرایش دوم است)***

☑ حل: مراحل دقیق بسته به متغیر و انتخاب مقداری دارد که شما آغاز کرده و ادامه می‌دهید. دراینجا یک راه‌حل ممکن را مشاهده می‌کنید: **الف)** متغیر $X_3$ را انتخاب کنید. دامنه آن $\{0,1\}$ می‌باشد. **ب)** مقدار 1 را برای متغیر $X_3$ انتخاب کنید. نمی‌توانستیم مقدار 0 را انتخاب کنیم زیرا طبق وارسی پیش‌رو، F برابر صفر می‌شد که منجر به غیرصفر شدن ارقام مجموع می‌گشت). **ج)** متغیر $F$ را در نظر بگیرید، زیرا فقط یک مقدار در دامنه آن باقی مانده است. **د)** مقدار 1 را برای $F$ انتخاب کنید. **ه)** اکنون متغیرهای $X_2$ و $X_1$ هر دو در دامنه خود نسبت به دیگران حداقل مقدار یعنی 2 تا را دارند. دراینجا $X_2$ را درنظر بگیرید. **و)** طبق وارسی پیش‌رو هر کدام از مقادیر مشکلی ایجاد نمی‌کند، دراینجا 0 را برای $X_2$ انتخاب کنید. **ز)** اکنون متغیر $X_1$ در دامنه خود، حداقل مقدار را دارد. **ح)** مجددا مقدار اختیاری 0 را برای $X_1$ انتخاب کنید. **ط)** متغیر $O$ بایستی یک عدد زوج و کمتر از 5 باشد. ($O$ زوج است زیرا $O=T+T$ که می‌دانیم مجموع هر عددی با خودش زوج می‌شود. همچنین $O$ باید کمتر از 5 باشد زیرا طبق $O+O=R+10X_1$ که $X_1$ صفراست، بایستی مجموع $O$ با خودش، تک رقمی باشد). این قانون باعث می‌شود این متغیر بسیار محدود شود. پس ابتدا آن را انتخاب کنید. **ی)** به طور دلخواه عدد 4 را برای متغیر $O$ در نظر بگیرید. **ک)** اکنون متغیر $R$ فقط یک مقدار مجاز در دامنه خود دارد پس آن را در نظر بگیرید. **ل)** مقدار 8 را برای $R$ انتخاب کنید. **م)** اکنون متغیر $T$ فقط یک مقدار مجاز در دامنه خود دارد. پس آن را در نظر بگیرید. **ن)** مقدار 7 را برای $T$ انتخاب کنید. **س)** $U$

---

[17] منظور از حساب رمزی، مساله‌ای است که در آن تعدادی از حروف الفبا با رابطه‌ای ریاضی نظیر جمع یا تفریق در کنار هم قرار می‌گیرند. باید به هر حرف الفبا عددی از صفر تا نه تخصیص دهیم که اولا هیچ دو حرفی عدد یکسان نداشته باشند و دوما رابطه ریاضی برقرار باشد. برای مساله گفته شده در شکل ۲.۵ یک راه‌حل به صورت {T=9,W=2,O=8,F=1,U=5,R=6} است.

متغیری است که درآن عدد زوج کمتر از 9 قرار دارد پس آن را در نظر بگیرید. **ع)** تنها مقداری برای $U$ که وارسی پیش‌رو تایید می‌کند عدد 6 است. **ف)** تنها متغیر باقی‌مانده $W$ است. پس آن را در نظر بگیرید. **ض)** تنها مقدار باقی‌مانده عدد 3 است پس برای W انتخاب کنید. **گ)** تمام متغیرها مقدار گرفته و هیچ یک از آنها محدودیتی را نقض نمی‌کنند پس یک راه‌حل یافته شد. این مساله معمایی ساده است بنابراین تعجبی ندارد که در یافتن راه‌حل به هیچ بازگشت به عقبی نیاز نشد و با همان وارسی پیش‌رو به راه‌حل رسیدیم. ولی در صورت رسیدن به تناقض باید یک مرحله به عقب پسگرد کرده و مقدار آخرین متغیر را تغییر داد.

**؟ ۶.۶ نشان دهید که چگونه می‌توان با یک متغیر کمکی محدودیتی سه‌گانه مانند A+B=C را به سه محدودیت دوگانه، تبدیل نمود. می‌توانید فرض کنید که دامنه‌ها متناهی‌اند. (راهنمایی:یک متغیر جدید در نظر بگیرید که مقادیر آن ترکیبی از مقادیر دو متغیر دیگر باشد و همچنین محدودیتی نظیر« X اولین عنصر از ترکیب Y است» را نیز تعریف کنید. سپس نشان دهید که می‌توان همین کار را با محدودیت‌های دارای بیش از سه متغیر انجام داد. در نهایت نشان دهید که چگونه می‌توان محدودیت‌های یگانی را با تغییر دامنه سایر متغیرها، حذف نمود. این بدان معناست که هر مساله CPS را می‌توان به یک مساله CSP فقط با محدودیت‌های یگانی، تبدیل نمود) *(این تمرین مشابه تمرین ۱۱.۵ در ویرایش دوم است)***

☑ حل: راهنمایی گفته شده در صورت مساله، راه‌حل را به طور کامل مشخص می‌کند. در محدودیتی سه‌گانه شامل سه متغیر $A$ و $B$ و $C$ نظیر $A+B=C$، در ابتدا یک متغیر جدید با نام $AB$ تعریف می‌کنیم، اگر دامنه متغیرهای $B$ و $A$ شامل مجموعه‌ای از اعداد $N$ باشد آنگاه دامنه $AB$ شامل مجموعه‌ای مرکب از $N$ از هر دو دامنه می‌باشد. به عنوان مثال $N\times N$ . اکنون سه محدودیت دوگانه داریم؛ یکی بین $A$ و $AB$ که می‌گوید هر مقدار از $A$ باید با اولین عنصر از زوج $AB$ برابر باشد؛ یکی بین $AB$ و $B$ که می‌گوید هر مقدار از $B$ باید با دومین عنصر از مقدار $AB$ برابر باشد؛ و محدودیت آخر که می‌گوید مجموع مقادیر بکار رفته در متغیر مرکب $AB$ بایستی با مقدار متغیر $C$ برابر باشد. برای سایر محدودیت‌های سه‌گانه به طور مشابه عمل می‌کنیم. همان طور که توانستیم محدودیت‌های سه‌گانه را به محدودیت‌های دوگانه تبدیل کنیم، می‌توانیم یک محدودیت 4- گانه شامل متغیرهای A و $B$ و $C$ و $D$ را نیز تبدیل به محدودیت‌های دوگانه کنیم. برای اینکار در ابتدا فقط متغیرهای $A$ و $B$ و $C$ را درنظرگرفته و طبق همان روش فوق، محدودیت‌های دوگانه را بدست می‌آوریم. سپس به محدودیت‌های سه‌گانه شامل $C$ و $AB$ و $D$ مراجعه کرده و با استفاده از تعریف متغیر جدید $CD$ آن را به محدودیتی دوگانه تبدیل می‌کنیم. با توجه به این توضیحات، می‌توانیم هر محدودیت $n$- گانه را در ابتدا به محدودیتی $N$-1 گانه تبدیل کنیم. می‌توانیم در همین مرحله محدودیت‌های دوگانه متوقف شده و به سراغ تبدیل به محدودیت‌های یگانی نرویم زیرا هر محدودیت یگانی را می‌توان با اعمال تغییراتی بر روی دامنه آن متغیر بدست آورده و آن محدودیت یگانه حذف می‌شود.

**؟ ۷.۶ معمای منطقی زیر را در نظر بگیرید: در 5 خانه با رنگ‌های مختلف، 5 نفر با ملیت‌های متفاوت زندگی می‌کنند و هر یک از آنها به نوع خاصی از شکلات، نوع خاصی از نوشیدنی، و نوع خاصی از حیوان علاقه‌مند هستند. با استفاده از این اطلاعات، پاسخ پرسش زیر را بیابید. «در کدام خانه حیوان گورخر نگهداری می‌شود و در کدام خانه آب نوشیده می‌شود؟». • مرد انگلیسی درخانه قرمز زندگی می‌کند. • مرد اسپانیایی یک سگ دارد. • مرد نروژی در اولین خانه سمت چپ زندگی می‌کند. • خانه سبز رنگ دقیقا سمت راست خانه شیری است. • مردی که شکلات Hershey Bars می‌خورد، پس از خانه مردی زندگی می‌کند که روباه دارد. • در خانه زردرنگ، شکلات Kit Kats خورده می‌شود. • مرد نوروژی بعد از خانه آبی زندگی می‌کند. • مردی که شکلات Smarties می‌خورد، صاحب حلزون است. • کسی که شکلات Snicker می‌خورد، آب پرتقال می‌نوشد. • مرد اوکراینی چای می‌نوشد. • مرد ژاپنی شکلات Milky Ways می‌خورد. • Kit Kats بعد از خانه‌ای مصرف می‌شود که در آنجا اسب نگهداری می‌کنند. • در خانه سبز رنگ قهوه نوشیده می‌شود. • در خانه وسط شیر نوشیده می‌شود. درمورد نمایش‌های مختلف این مساله به صورت یک CSP بحث کنید. چرا ممکن است یک نمایش بر دیگری ارجح باشد؟ *(این تمرین مشابه تمرین ۱۳.۵ در ویرایش دوم است)***

☑ حل: این معما که به «معمای گورخر»شهرت دارد می‌تواند به صورت یک مساله CSP تفسیر شود پس باید سه مجموعه متغیر، دامنه و محدودیت را در آن بیابیم. در اینجا به ازای هر یک از رنگ‌ها، حیوانات، نوشیدنی‌ها و نام شکلات‌ها یک متغیر در نظر می‌گیریم که مجموعا ۲۵ متغیر تشکیل می‌شود. حال نوبت به تعیین دامنه متغیرها می‌شود. هرمتغیر دارای یکی از اعداد 1 تا 5 که نشان‌دهنده شماره خانه است می‌باشد. این شیوه نمایش مناسب است زیرا می‌توان تمام محدودیت‌های تعریف شده در مساله را به سادگی با این متغیرها تعریف نمود. (کدی به زبان پایتون این مساله را پیاده‌سازی کرده است و می‌توان در سایر زبان‌ها مجددا آن را پیاده‌سازی نمود). دلیل انتخاب این شیوه نمایش CPS نسبت به سایرین، کارایی این روش در یافتن راه‌حل می‌باشد در برخی اجراها، جستجوی محلی با حداقل تناقضات می‌تواند راه‌حل این مساله را در طی چند ثانیه بیابد ولی سایر نمایش‌های ممکن پس از چند دقیقه این کار را انجام می‌دهند. یکی از روش‌های دیگر برای نمایش این مساله آن است که 5 متغیر برای هر خانه تعریف کنیم. یک متغیر برای رنگ با دامنه {قرمز، سبز، شیری، زرد، آبی} متغیری برای حیوان با دامنه {گورخر، حلزون، روباه، اسب و سگ}، متغیری جهت نوشیدنی با دامنه {آب پرتقال، چای، قهوه، آب}، متغیر دیگر جهت نام شکلات‌ها {Kit Kats, Milky Ways, Hershey Bars, Smarties, Snicker} و متغیر دیگر مکان خانه‌ها تعریف کنیم.

**۸.۶ گرافی شامل 8 گره با نام‌های $F_2,F_1,T,H,A_4,A_3,A_2,A_1$ را در نظر بگیرید. هر $A_i$ به ازای تمام $i$ها به $A_{i+1}$ متصل است. $A_1$ به $H$، $H$ به $T$ و $T$ به هر $F_i$ نیز متصل شده است. این گراف را با سه رنگ طبق استراتژی زیر رنگ‌آمیزی کنید: بازگشت به عقب توسط «پسگرد مستقیم به تناقضات» که در آن متغیرها به ترتیب $T,A_3,F_2,A_2,F_1,A_4,H,A_1$ و مقادیر به ترتیب R، G و B مرتب شده باشند.**

حل: **الف)** $A_1=R$ . **ب)** $H=R$ که با رنگ $A_1$ تناقض دارد. **ج)** $H=G$ . **د)** $A_4=R$ . **هـ)** $F_1=R$ . **و)** $A_2=R$ که با مقدار $A_1$ تناقض دارد، $A_2=G$ که با $H$ تناقض دارد، بنابراین $A_2=B$ . **ز)** $F_2=R$ . **ح)** $A_3=R$ با مقدار $A_4$ تناقض دارد، $A_3=G$ با مقدار $H$ تناقض دارد، $A_3=B$ با مقدار $A_2$ تناقض دارد بنابراین به متغیر قبلی باز می‌گردیم. مجموعهٔ تناقض عبارتست از $\{A_2,H,A_4\}$. بنابراین به متغیر $A_2$ جهش می‌کنیم. متغیرهای $\{H, A_4\}$ را به مجموعه تناقضات $A_2$ اضافه می‌کنیم. **ط)** $A_2$ هیچ مقدار مجازی ندارد پس به متغیر قبلی باز می‌گردیم. مجموعه تناقض عبارتست از $\{A_1,H,A_4\}$. سپس به متغیر $A_4$ می‌رویم و مجموعهٔ $\{H,A_1\}$ را به مجموعه تناقضات $A_4$ اضافه می‌کنیم. **ی)** $A_4=G$ با مقدار $H$ تناقض دارد پس $A_4=B$ . **ک)** $F_1=R$ . **ل)** $A_2=R$ با مقدار $A_1$ تناقض دارد، $A_2=G$ با مقدار $H$ تناقض دارد پس $A_2=B$ . **م)** $F_2=R$ . **ن)** $A_3=R$ . **س)** $T=R$ بامقدار $F_1$ و $F_2$ تناقض دارد، $T=G$ با مقدار $G$ تناقض دارد پس $T=B$ . **ع)** رنگ‌آمیزی با موفقیت به انجام رسید.

**۹.۶ توضیح دهید که چرا در مسئله CSP، «انتخاب متغیری که بیشترین محدودیت را ایجاد می‌کند» و «انتخاب مقداری که محدودکننده کمتری است»، توابع اکتشافی خوبی محسوب می‌شوند؟** ***(این تمرین مشابه تمرین ۳.۵ در ویرایش دوم است)***

حل: منظور از متغیری که بیش از همه محدودیت ایجاد می‌کند، متغیری است که نام آن در قوانین محدودیت بسیار استفاده شده است و بعبارتی دیگر مقدار آن متغیر با سایر متغیرها بسیار دخیل خواهد بود. به همین خاطر اگر آنها را در ابتدا مقدار دهیم معمولا در زمان انتساب مقدار به آن، مقدار مجازی نمی‌یابیم و راه‌حل فعلی شکست می‌خورد بنابراین این تابع اکتشافی پیشنهاد می‌دهد که جهت افزایش کارایی و رسیدن به راه‌حل، بهتر است در زمان انتخاب متغیر، ابتدا به این متغیرها مقداری مجاز را انتساب داده و سپس به سراغ دیگران که معمولا محدودیتی کمتر دارند، برویم. (با این کار گویا قسمتی از فضای جستجو را هرس کرده‌ایم). روش «انتخاب مقداری که محدودیت کمتری ایجاد می‌کند» نیز اکتشافی خوب محسوب می‌شود زیرا برای طولگیری از تناقض شانس بیشتری در انتساب‌های بعدی داریم. لازم بذکر است که این تابع کاری با انتخاب متغیر ندارد و فرض می‌کند متغیر موردنظر به شیوه‌ای دلخواه انتخاب شده است، و تنها بر روی مقدار انتسابی به آن کار می‌کند.

**۱۰.۶ نمونه‌های تصادفی زیر را به عنوان مسائل رنگ‌آمیزی تولید کنید: $n$ نقطه را در مربعی واحد پخش کنید، نقطهٔ تصادفی $X$ را انتخاب کرده و آن را توسط خطی مستقیم به نزدیک‌ترین نقطه با نام $Y$ که قبلا متصل نبوده است وصل کنید. این خط نباید از روی خطوط دیگر عبور کند. این گام را آنقدر تکرار کنید که دیگر هیچ اتصالی ممکن نباشد. اکنون نقاط را به عنوان شهر و خطوط را به عنوان جاده‌های اتصالی در نظر بگیرید و بکوشید گراف خود را با k رنگ مختلف یکبار به ازای $k=3$ و یکبار $k=4$ و استفاده از روش‌های حداقل‌تناقض، پس‌رو، روش پس‌رو با وارسی پیش‌رو، و روش پس‌رو با MAC رنگ‌آمیزی و حل نمائید. سپس جدولی تشکیل دهید که به ازای هر الگوریتم متوسط زمان اجرا را با افزایش مقادیر $n$ نشان دهد. در مورد نتایج خود توضیح دهید.** ***(این تمرین اندکی مشابه تمرین ۷.۵ در ویرایش دوم است)***

حل: به پاسخ تمرین ۷.۵ در ویرایش دوم مراجعه کنید.

**۱۱.۶ با استفاده از الگوریتم AC-3 نشان دهید که سازگاری کمان قادر است تا در مساله شکل ۱.۶ ناسازگاری حاصل از انتساب جزئی {WA=green , V= red} را تشخیص دهد.** ***(این تمرین مشابه تمرین ۸.۵ در ویرایش دوم است)***

حل: لازم بذکر است که منظور از انتساب کامل آن است که به تمام متغیرها مقداری را انتساب داده و هیچ تناقضی مشاهده نشود. و انتساب جزئی آن است که فقط به تعدادی از متغیرها مقدار داده و هنوز به پایان راه‌حل نرسیده باشیم. به منظور تشخیص ناسازگاری در یک انتساب جزئی، هرتکرار از حلقه While در الگوریتم $AC$=3 را دنبال می‌کنیم. (برای هر چیدمان مجاز از کمان‌ها) **الف)** حذف کمان $SA$-$WA$، برداشتن مقدار $G$ از دامنه متغیر $SA$. **ب)** حذف کمان $V$-$SA$ و برداشتن مقدار $B$ از متغیر $SA$، تنها مقدار $B$ باقی می‌ماند. **ج)** حذف کمان $NT$-WA، برداشتن مقدار $G$ از دامنه متغیر $NT$. **د)** حذف کمان $NT$-$SA$ و برداشتن مقدار $B$ از دامنه متغیر $NT$، تنها مقدار $R$ باقی می‌ماند. **هـ)** حذف کمان $SA$-$NSW$، برداشتن مقدار $B$ از دامنه متغیر $NSW$. **و)** حذف کمان $V$-$NSW$ و برداشتن مقدار $R$ از دامنه متغیر $NSW$، تنها مقدار $G$ باقی می‌ماند. **ز)** حذف کمان $NT$-$Q$، برداشتن مقدار $R$ از متغیر $Q$ . **ح)** حذف کمان $SA$ ,$Q$، برداشتن مقدار $B$ از دامنه متغیر از متغیر$Q$ . **ط)** حذف کمان $NSW$-$Q$، برداشتن مقدار $G$ از دامنه متغیر $Q$ هیچ مقداری در دامنه $Q$ باقی نمی‌ماند.

**۱۲.۶ در بدترین حالت، پیچیدگی AC-3 درمساله CSP با ساختار درختی چقدر است؟** ***(این تمرین مشابه تمرین ۹.۵ در ویرایش دوم است)***

حل: در یک گراف با ساختار درختی، هیچ کمانی بیش از یک بار درنظر گرفته نمی‌شود، بنابراین الگوریتم $AC$-3 دارای پیچیدگی $O(ED)$ خواهد بود که درآن $E$ تعداد لبه‌ها و $D$ اندازه بزرگترین دامنه متغیر می‌باشد.

**۱۳.۶ الگوریتم AC-3 در زمان حذف مقداری از دامنه متغیر $X_i$ تمام کمان‌های $(X_k,X_i)$ را درون صفی قرار می‌دهد، حتی اگر مقدار از $X_i$ با چندین مقدار از دامنه $X_i$ سازگار باشد. فرض کنید به ازای هر کمان $(X_k,X_i)$ تعداد مقادیر باقیمانده از $X_i$ که با مقداری از $X_k$ سازگار هستند را نگهداری کنیم. توضیح دهید که چگونه این اعداد را به صورت کارا آپدیت کنیم و سپس نشان دهید که سازگاری کمان بایستی در زمان $O(n^2d^2)$ انجام شود.** ***(این تمرین مشابه تمرین ۱۰.۵ در ویرایش دوم است)***

☑ حل: ایده این کار بر مبنای پیش‌پردازش محدودیت‌ها استوار است یعنی به ازای هر مقدار از $X_i$ مقادیری از $X_k$ را بیابیم که یک کمان از $X_k$ به $X_i$ با آن انتساب‌های جزئی $X_i$ ارضاء شده و در آن صدق کند. زمان محاسبه این ساختمان‌داده بستگی به اندازه مساله دارد. بنابراین زمانی که یک مقدار از دامنه $X_i$ حذف شود از تعداد مقادیر مجاز برای هرکمان $(X_i,X_k)$ تحت آن مقدار، یک واحد کم می‌کنیم این راه‌کار بسیار شبیه به الگوریتم زنجیره‌ای پیش‌رو در فصل ۷ می‌باشد.

**؟ ۱۴.۶ TREE-CSP-SOLVER شکل ۱۰.۶ سازگاری کمان را برقرار کرده است که از برگ‌ها شروع نموده و با پسگرد به عقب به ریشه رسیده است. اگر جهت این حرکت را برعکس نماییم چه اتفاقی می‌افتد؟**

☑ حل: عملیات سازگاری کمان را از پایین به بالا اعمال نمودیم تا پس از برقراری سازگاری بتوان مسأله را از بالا به پایین حل نمود. البته ممکن است راه‌حلی وجود داشته باشد که به هیچ پسگردی در آن نیاز نشود به خاطر تعریف سازگاری کمان: اگر بتوان مقداری برای گره پدر انتخاب نمود، آنگاه برای گره فرزند نیز مقداری وجود خواهد داشت

**؟ ۱۵.۶ جدول سودوکو را به عنوان یک مسأله CSP با جستجو در انتساب‌های جزئی می‌توان تعریف نمود زیرا مردم نیز معمولا این معما را اینگونه حل می‌کنند. البته می‌توان این مسأله را با جستجوی محلی بر روی انتساب کامل نیز حل نمود. یک حل‌کننده محلی با تابع اکتشاف «حداقل تناقض» چه کارایی برای انجام این مسأله خواهد داشت؟**

☑ حل: مسلم است که می‌توان مسأله سودوکو را طبق این شرایط حل نمود. ولی استفاده از راه‌کار انتساب‌های جزئی و تابع حداقل تناقضات در این مسأله خیلی مفید و کارا نخواهد بود مانند مسأله N- وزیر. شاید بدان خاطر است که در این مسأله دو نوع تناقض وجود دارد: یک تناقض به خاطر یکی از اعدادی است که در ابتدای مسأله قرار داده‌ایم و بایستی آن عدد را تصحیح کنیم ولی نوع دیگری از تناقض زمانی است که دو عدد در مکان مختلف با هم تناقض دارند که بایستی آن دو را تعویض کنیم. نسخه‌ای از روش حداقل تناقض وجود دارد که تفاوت بین این دو موقعیت را تشخیص می‌دهد و این نسخه بهتر از الگوریتم اولیهٔ حداقل تناقض خواهد بود.

**؟ ۱۶.۶ این لغات را به زبان خود تعریف کنید: مسئله ارضای‌محدودیت[18]، محدودیت، جستجوی بازگشت به عقب (پسگرد)، سازگاری کمان، پرش به عقب و تابع اکتشافی حداقل تناقاضات و کات‌ست چرخه‌ای. *(این تمرین مشابه تمرین ۵.۱ در ویرایش دوم است)***

☑ حل: یک مسئله ارضای‌محدودیت، به طور خلاصه مسئله‌ای است متشکل از تعدادی متغیر، دامنه متغیر و تعدادی قانون محدودیت. هر متغیر دارای دامنه مقدار یعنی مجموعه مقادیر مجاز برای انتساب می‌باشد و هر محدودیت، قانونی ریاضی یا تعریفی بر روی متغیرهاست که برای انتساب مقدار به متغیرها شرط می‌گذارد. این تعریف ساده‌ای از مساله CSP است که می‌توان مسائل مختلف را با یافتن این سه مجموعه (متغیر، محدودیت، دامنه) تدوین نمود. به عنوان مثال اگر بخواهیم مساله رنگ‌آمیزی نقشه استرالیا را به عنوان CSP تدوین کنیم، بایستی این سه مجموعه را بیابیم. در اینجا مجموعه متغیرها {WA,NT,Q,SA,NSW,V,T}، مجموعه محدودیت‌ها {دو همسایه همرنگ نباشند} و دامنه هر متغیر {قرمز، سبز، آبی} می‌باشد. منظور از حل یک مساله CSP آن است که به هر متغیر مقداری را از دامنه‌اش انتساب دهیم و هیچ یک از قوانین محدودیت نقض نشود. مثلا یکی از راه‌حل‌های ممکن برای مساله فوق عبارتست از {WA=red,NT=green,Q=red,SA=blue,NSW=green,V=red,T=green} که هیچ محدودیتی را نقض نمی‌کند. یک محدودیت، یک قانون یا رابطه است که بر روی مقادیر ممکن متغیرها تعریف می‌شود و مقادیر مجاز آنها را محدود می‌کند. به عنوان مثال یک محدودیت برای متغیر $A$ می‌تواند آن باشد که در صورت $B=b$ بودن، $A=a$ مجاز نیست. البته می‌توان انواع مختلف محدودیت‌ها را تعریف نمود مثلا محدودیت $A>2$، یا $A^2+B=5$ یا ... . جستجوی بازگشت‌به‌عقب یا پسگرد نوعی جستجوی اول‌عمق است که در هر بار به یک متغیر مقداری از دامنه‌اش را انتساب داده و به سراغ متغیر بعدی می‌رویم مانند پیشروی در عمق درخت جستجو. اگر نوبت به متغیری رسید که با توجه به مقادیر سایر متغیرها و محدودیت‌های موجود، هیچ مقداری برایش مجاز نباشد، پس نتیجه می‌گیریم که در انتساب‌های قبلی اشتباه کرده‌ایم لذا یک مرحله به عقب بازگشته و مقدار انتسابی به آخرین متغیر را تغییر می‌دهیم. در صورت مقدار دادن به تمام متغیرها بدون نقض هیچ محدودیتی، جستجو پایان می‌یابد. یک کمان مستقیم از متغیر $A$ به متغیر $B$ در مسائل ارضای‌محدودیت، کمان سازگار نامیده می‌شود، درصورتی که به ازای هر مقدار موجود در دامنه $A$، مقداری مجاز برای $B$ وجود داشته باشد. پرش‌به‌عقب، یکی از روش‌های افزایش کارایی جستجوی پسگرد است که در آن هر زمان که به نقطه‌ای رسیدیم که نیاز به بازگشت به عقب بود، به جای آنکه فقط یک مرحله به عقب پرش کنیم و مقدار را اصلاح نماییم، به نقطه آغاز اشتباه که معمولا چند سطح قبلتر می‌باشد، می‌پریم. «حداقل-تناقضات» یک تابع اکتشافی است که در روش‌های CSP با جستجوی محلی بکار می‌رود. منظور از این تابع آن است که در زمان انتساب مقدار به یک متغیر، مقداری را برای آن انتخاب کنیم که در صورت امکان با کمترین متغیرها تداخل و تناقض داشته باشد. یک کات‌ست چرخه‌ای مجموعه متغیرهایی است که اگر گراف محدودیت حذف شوند، گرافی بدون دور تشکیل می‌شود(مانند یک درخت). اگر به متغیرهای کات‌ست مقدار دهیم، آنگاه مابقی مسئلهٔ CSP در زمانی خطی حل می‌شود.

**؟ ۱۷.۶ فرض کنید کات‌ست چرخه‌ای در یک گراف، حداکثر $k$ گره داشته باشد. یک الگوریتم ساده برای یافتن یک کات‌ست چرخه‌ای مینیمال بیابید که زمان اجرای آن بیشتر از $O(n^k)$ در یک مساله CSP با $n$ متغیر نباشد. درمراجع مختلف جستجو**

[18] Constraint satisfaction problem (CSP)

**ده و متدهایی برای یافتن تقریبی کات‌ست چرخه‌ای مینیمال در زمانی با مرتبه چند جمله‌ای نسبت به سایز کات‌ست بیابید.**
**وجود چنین الگوریتم‌هایی روش کات‌ست‌چرخه‌ای را عملی می‌کند؟** ***(این تمرین مشابه تمرین ۱۲.۵ در ویرایش دوم است)***

ا حل: یک الگوریتم ساده برای یافتن کات‌ست با حداکثر $k$ گره آن است که تمام زیرمجموعه‌های گره‌ها با اندازه $1,2,\ldots,k$ را مرده و سپس به ازای هر زیرمجموعه، بررسی کنیم که آیا گره‌های باقی‌مانده تشکیل یک درخت را می‌دهند یا خیر. این وریتم دارای زمانی $\left(\begin{matrix}\sum_{i=1}^{k} n \\ nk\end{matrix}\right)$ به مرتبه $O(n^k)$ می‌باشد.

ر و گیگر (1994, http://citeseer.nj.nec.com/becker94approximation.html) الگوریتمی با نام MGA ارائه ند. (نوعی تغییر یافته از الگوریتم حریصانه) که می‌تواند یک کات‌ست را در حداکثر دو برابر اندازه کات‌ست مینیمال بیابد و تبه زمانی آن $O(E+V\text{Log}V)$ می‌باشد که منظور از $E$ تعداد لبه‌ها و $V$ تعداد متغیرهاست. اینکه آیا این موضوع روش ت‌ست را عملی می‌کند یا خیر بیشتر از آنکه به الگوریتم یافتن مربوط شود به گرافی مربوط است که در آن بدنبال کات‌ست ستیم. زیرا برای کات‌ست با اندازه $c$ قبل از حل CSP نیاز به فاکتوری نمایی ($d^c$) داریم. بنابراین هر گراف با کات‌ست بزرگ ی دشوار خواهد داشت. حتی اگر بتوانیم بدون هیچ کوششی، کات‌ست بیابیم.

# فصل ۷ (ویرایش سوم)

**؟ ۱.۷ فرض کنید کارگزار در موقعیت شکل ۴.۷(a) قرار دارد یعنی در خانهٔ [1,1] هیچ چیزی مشاهده نمی‌کند ولی در خانهٔ [2,1] وجود نسیم و در خانهٔ [1,2] وجود بو را احساس می‌کند. کارگزار در این موقعیت، دربارهٔ خانه‌های [1,3] و [2,2] و [3,1] نگران است زیرا ممکن است در هر کدام از آنها چاله و در یکی از آنها ومپوز قرار داشته باشد. مطابق مثال شکل ۵.۷ مجموعهٔ دنیاهای ممکن را تشکیل دهید. (بایستی 32 مورد را بیابید). سپس مدل‌هایی که طبق KB جمع‌آوری شده صحیح هستند و مدل‌هایی که عبارات زیر در آنها صحیح هستند را مشخص نمایید:**

**$\alpha_2$= هیچ چاله‌ای در خانهٔ [2,2] نیست. $\alpha_3$= یک ومپوز در خانهٔ [1,3] قرار دارد.**

**و نشان دهید که $KB \models \alpha_2$ و $KB \models \alpha_3$ می‌باشند.** ***(این تمرین مشابه تمرین ۲.۷ در ویرایش دوم است)***

☑ حل:

| Model | $KB$ | $\alpha_2$ | $\alpha_3$ |
|---|---|---|---|
| | | *true* | |
| $P_{1,3}$ | | *true* | |
| $P_{2,2}$ | | | |
| $P_{3,1}$ | | *true* | |
| $P_{1,3}$, $P_{2,2}$ | | | |
| $P_{2,2}$, $P_{3,1}$ | | | |
| $P_{3,1}$, $P_{1,3}$ | | *true* | |
| $P_{1,3}$, $P_{3,1}$, $P_{2,2}$ | | | |
| $W_{1,3}$ | | *true* | *true* |
| $W_{1,3}$, $P_{1,3}$ | | *true* | *true* |
| $W_{1,3}$, $P_{2,2}$ | | | *true* |
| $W_{1,3}$, $P_{3,1}$ | *true* | *true* | *true* |
| $W_{1,3}$, $P_{1,3}$, $P_{2,2}$ | | | *true* |
| $W_{1,3}$, $P_{2,2}$, $P_{3,1}$ | | | *true* |
| $W_{1,3}$, $P_{3,1}$, $P_{1,3}$ | | *true* | *true* |
| $W_{1,3}$, $P_{1,3}$, $P_{3,1}$, $P_{2,2}$ | | | *true* |
| $W_{3,1}$, | | *true* | |
| $W_{3,1}$, $P_{1,3}$ | | *true* | |
| $W_{3,1}$, $P_{2,2}$ | | | |
| $W_{3,1}$, $P_{3,1}$ | | *true* | |
| $W_{3,1}$, $P_{1,3}$, $P_{2,2}$ | | | |
| $W_{3,1}$, $P_{2,2}$, $P_{3,1}$ | | | |
| $W_{3,1}$, $P_{3,1}$, $P_{1,3}$ | | *true* | |
| $W_{3,1}$, $P_{1,3}$, $P_{3,1}$, $P_{2,2}$ | | | |
| $W_{2,2}$ | | *true* | |
| $W_{2,2}$, $P_{1,3}$ | | *true* | |
| $W_{2,2}$, $P_{2,2}$ | | | |
| $W_{2,2}$, $P_{3,1}$ | | *true* | |
| $W_{2,2}$, $P_{1,3}$, $P_{2,2}$ | | | |
| $W_{2,2}$, $P_{2,2}$, $P_{3,1}$ | | | |
| $W_{2,2}$, $P_{3,1}$, $P_{1,3}$ | | *true* | |
| $W_{2,2}$, $P_{1,3}$, $P_{3,1}$, $P_{2,2}$ | | | |

شکل ۱.۷s گزاره‌هایی که بعنوان true در خطوط مشخص شده لیست نشده‌اند، false هستند.

در این جدول، تنها ورودی‌های true ذکر شده است.

منظور از مدل در یک دنیا، تمام حالات ممکن از وضعیت‌هاست که یکی از آنها در واقعیت صحت دارد ولی ما به دلیل اطلاعات ناقص خود نمی‌توانیم مدل حقیقی که بر دنیا منطبق است را بیابیم. برای یافتن آن مدل حقیقی، به مشاهدات و استنتاج بیشتر نیازمندیم. با انجام مشاهدات بیشتر، دانش ما از دنیای ومپوز بیشتر خواهد شد و اصطلاحا گفته می‌شود که دانشی به پایگاه‌دانش افزوده شد. در ابتدای این تمرین در مورد سه خانه [1,3] و [2,2] و [3,1] هیچ اطلاعی نداریم و هر یک از این خانه‌ها دارای چاله، ومپوز، نسیم و یا بو هستند. بنابراین برای هر خانه: یا چاله وجود دارد یا ندارد (برای سه خانه، 8 حالت مختلف می‌شود)، یا

مپوز وجود دارد یا ندارد (برای سه خانه، 8 حالت مختلف می‌شود)، یا نسیم وجود دارد یا ندارد (برای سه خانه، 8 حالت مختلف می‌شود)، یا بو وجود دارد یا ندارد (برای سه خانه، 8 حالت مختلف می‌شود). باتوجه به مجموع موارد فوق بایستی در کل تعداد 32 مدل ممکن ترسیم شود. در کتاب پیشنهاد می‌دهد که تمام این مدل‌ها را بـه طـور دیـاگرامی ترسـیم کـرده و سـپس دور مدل‌هایی که طبق دانش شما صحیح هستند خط بکشید. تعدادی از این مدل‌ها با توجه به پایگـاه‌دانـش یعنـی اطلاعـات مـا از محیط، رد می‌شوند: به عنوان مثال می‌دانیم که در این محیط فقط یک ومپوز وجود دارد پس مدل‌هایی که دارای بیش از یـک ومپوز هستند در محدوده پایگاه نیستند؛ همچنین با توجه به اینکه محتوای سه خانه از محـیط را مـی‌دانـیم، پایگـاه دانـش مـا تکمیل‌تر است مثلا می‌دانیم که اگر در خانه‌ای بو باشد در مجاور آن حتما ومپوز وجود دارد پس مدل‌هایی که این قانون را نقض می‌کنند از محدوده خط KB کنار می‌روند. با ادامه همین روال مدل‌هایی که طبق دانش ما از محـیط صـحیح هسـتند را درون محدوده‌ای بنام KB قرار می‌دهیم که تنها 12 مدل می‌باشند (8 حالت مختلف برای نمایش چاله در سه خانـه و چهـار حالـت مختلف برای نمایش ومپوز البته عدم وجود ومپوز نیز حساب شده است). در اینجا جهت صرفه‌جویی در فضای مصرفی، به جـای ترسیم دیاگرام‌های دنیا می‌توانید از جدول استفاده کنید. اگر بخواهیم اثبات کنیم که می‌توان $\alpha_2$ را از این پایگاه‌دانـش ایجـاب کرد باید بتوان تمام مدل‌های KB را درون محدوده مدل‌های $\alpha_2$ قرار داد (یعنی محـدوده KB زیرمجموعـه‌ای از محـدوده $\alpha_2$ باشد) که در این مثال محدوده مدل KB، زیرمجموعه هر کدام از $\alpha_2$ و $\alpha_3$ است. پس داریم: $KB \models \alpha_2$ و $KB \models \alpha_3$.

**؟ ۲.۷ (با اقتباس از Barwise وEtchemendy (سال 1993)) و فرضیهٔ زیر، آیا می‌توانید اثبات کنید که تک شاخ موجـودی اسطوره‌ای است؟ در مورد سحرآمیز بودن آن چطور؟ در مورد شاخ‌داری آن چطور؟ «اگر تک شاخ موجودی اسـطوره‌ای اسـت، آنگاه فناناپذیر خواهد بود ولی اگر اسطوره‌ای نباشد، آنگاه یک پستاندار فناپذیر خواهد بود. اگر یک تک شـاخ، پـستاندار یـا فناناپذیر باشد، آنگاه شاخ‌دار خواهد بود و اگر تک شاخ یک موجود شاخ‌دار باشد حتما سحرآمیز است». *(ایـن تمـرین مـشابه تمرین ۷.۹ در ویرایش دوم است)***

☑ حل: طبق دو عبارت اول می‌یابیم که در صورتی که تک شاخ اسطوره‌ای باشد فناناپذیر است و در غیـر ایـن صـورت فنـاپـذیر است. بنابراین بایستی یا پستاندار یا فناناپذیر باشد و بنابراین شاخ‌دار است. این بدان معناست که این موجود سحرآمیز نیز هست. به هر حال نمی‌توان در مورد اسطوره‌ای بودن آن تصمیم گرفت. طبق کد استدلال گزاره‌ای داریم:

```
> (setf kb (make-prop-kb))
#S(PROP-KB SENTENCE (AND))
> (tell kb "Mythical => Immortal")
T
> (tell kb "~Mythical => ~Immortal ^ Mammal")
T
> (tell kb "Immortal | Mammal => Horned")
T
> (tell kb "Horned => Magical")
T
> (ask kb "Mythical")
NIL
> (ask kb "~Mythical")
> (ask kb "Magical")
T
> (ask kb "Horned")
T
```

**؟ ۳.۷ مسأله تصمیم‌گیری در مورد صحت یک گزاره منطقی با توجه به مدلی مشخص را در نظر بگیرید: الـف) یـک الگـوریتم بازگشتی به صورت ?PL-True(s,m) بنویسید که اگر و فقط اگر عبارت s در مدل m صحیح باشد، کلمهٔ true را برمی‌گرداند. m برای هر نماد در عبارت s، مقداری حقیقی را برمی‌گرداند). این الگوریتم بایستی نسبت به اندازهٔ عبارت، در زمانی خطی اجرا شود. ب) سه جمله مثال بزنید که درست یا غلط بودن آنها را بطور قطعی می‌دانیم ولی مدل آنها جزئی است، یعنی فقط درستی برخی از نمادهای جملات مشخص است. ج) نشان دهید که در حالت کلی درستی یک جمله (در صورت وجود) در مدلی‌جزئـی نمی‌تواند به طور کارا تعیین شود. د) الگوریتم ?PL–True را تغییر دهید تا ضمن حفظ ساختار بازگشتی زمان اجرای خطـی، بتواند گاهی مواقع در مورد درستی یک عبارت در مدلی جزئی قضاوت کند. سه جمله مثال بزنید که الگوریتم شما نمـی‌توانـد صحت آنها را در مدلی‌جزئی شناسایی کند. هـ) بررسی کنید که الگوریتم تغییریافته، چه تـأثیری در کـارایی TT–ENTAIL دارد؟ *(این تمرین مشابه تمرین ۳.۷ در ویرایش دوم است)***

☑ حل: **الف)** شکل ۲.۷S را ببینید. فرض کرده‌ایم که در زبان مربوطه عملگرهای not، and، or و iff موجود باشند.

```
function PL-TRUE?(s, m) returns true or false
  if s = True then return true
  else if s = False then return false
  else if SYMBOL?(s) then return LOOKUP(s, m)
  else branch on the operator of s
    ¬: return not PL-TRUE?(ARG1(s), m)
    ∨: return PL-TRUE?(ARG1(s), m) or PL-TRUE?(ARG2(s), m)
    ∧: return PL-TRUE?(ARG1(s), m) and PL-TRUE?(ARG2(s), m)
    ⇒: (not PL-TRUE?(ARG1(s), m)) or PL-TRUE?(ARG2(s), m)
    ⇔: PL-TRUE?(ARG1(s), m) iff PL-TRUE?(ARG2(s), m)
```

شکل ۲.۷s شبه برنامه‌ای برای ارزیابی صحت جملهٔ wrt یک مدل

**ب)** سئوال کمی ابهام دارد: منظور از «یک مدل جزئی» به معنای تمام مدلها و یا برخی مدلها است! منظور از مدل‌جزئی آن است که مقدار تمام متغیرهای موجود در جمله مشخص نبوده و تنها مقدار بخشی از آنها را می‌دانیم. عباراتی نظیر $P \wedge \text{False}$, $P \vee \neg P$, $\text{True} \vee \neg P$, مقداری قطعی دارند یعنی یا قطعا درست و یا نادرست هستند حتی اگر مقدار متغیر P را ندانیم (مدل جزئی). برای تفسیر دوم، می‌توان عباراتی نظیر $A \wedge P$ با مقدار False داشته باشیم که در مدل جزئی $\{A=\text{False}\}$ باشد.**ج)** دو جمله با $K$ متغیر را در مدلی‌جزئی یعنی شرایطی که مقدار درستی تمام متغیرها را نمی‌دانیم درنظربگیرید. یک جمله به ازای تمام $2^k$ مقدار ممکن، صحیح است (مانند مثال‌های موجود در بخش قبل) ولی جملهٔ دیگر به ازای یک حالت از $2^k$ مقدار ممکن نادرست می‌باشد. بنابراین بایستی در حالت کلی تمام $2^k$ حالت ممکن را بررسی کنیم که زمانی نمایی صرف می‌شود. از جنبه دیگر، الگوریتم کلی برای مدلهای جزئی باید مدل جزئی تهی که هنوز هیچ انتسابی در آن انجام نشده است را نیز شامل شود. در اینصورت بایستی مفاهیم اعتبارسنجی و غیرارضاپذیری بکار گرفته شود که پیچیدگی NP-کامل دارند. **د)** اینکار میسر است اگر عملگرهای and و or بتوانند آرگومانهای خود در عبارت را ارزیابی کرده و به ترتیب به False و True ختم گردند. پیاده‌سازی PL–True در زبان پاتیون بگونه‌ای است که در صورتی مقدار True را برمی‌گرداند که تمام بخشهای یک ترکیب عطفی صحیح باشند؛ و در صورتیکه تمام بخشهای یک ترکیب‌فصلی غلط باشند مقدار False را برمی‌گرداند. این اتفاق حتی در صورتی که فصل/عطف شامل متغیرهای بدون‌مقدار باشد نیز همین است. در این مورد این الگوریتم همچنان ویژگیهایی ناخواسته دارد: در مدل جزئی که $P$ درست و $Q$ ناشناخته است، $P \vee Q$ درست و عبارت $P \wedge Q$ نادرست می‌شود. ولی مقادیر درستی $Q \vee \neg Q$، $Q \vee \text{True}$, $Q \wedge \neg Q$ قابل تعیین نیست. زیرا متغیر $Q$ مقدار نداشته و این برنامه نمی‌تواند درصورت مقدار نداشتن هر دو بخش یک ترکیب آنرا ارزیابی کند. **هـ)** نسخهٔ tt–entail موجود از تابع PL–True تغییر یافته استفاده می‌کند که درصورتیکه از آن استفاده نکنیم، عملکردی کندتر خواهد داشت.

**؟ ۴.۷ کدامیک از موارد زیر صحیح می‌باشد: الف)** $\text{True} \models \text{False}$ **ب)** $\text{False} \models \text{True}$ **ج)** $\text{True} \models (B) \models (A \wedge B)$ **د)** $(A \wedge B) \models (A \Leftrightarrow B)$ **هـ)** $A \Leftrightarrow B \models A \vee B$ **و)** $A \Leftrightarrow B \models \neg A \vee B$ **ز)** $(A \wedge B) \Rightarrow C \models (A \Rightarrow C) \vee (B \Rightarrow C) \equiv (C \vee (\neg A \wedge \neg B)) \equiv ((A \Rightarrow C) \wedge (B \Rightarrow C))$ **ح)** $(A \vee B) \models (A \vee B) \wedge (\neg C \vee \neg D \vee E)$ **ط)** $(A \vee B) \wedge (\neg D \vee E) \models (A \vee B) \wedge (\neg C \vee \neg D \vee E)$ **ی)** $(A \vee B) \wedge \neg (A \Rightarrow B)$ **ارضاپذیر است. ک)** $(\neg A \vee B) \wedge (A \Leftrightarrow B)$ **ارضاپذیر است. ل)** **عبارت** $C \Leftrightarrow (A \Leftrightarrow B)$ **دارای تعداد مدلی برابر با** $(A \Leftrightarrow B)$ **به ازای هر مجموعه نماد A و B و C می‌باشد.**

☑ حل: در تمامی موارد، با مراجعه به تعریف استلزام می‌توان به راحتی مساله را حل نمود. **الف)** $\text{True} \models \text{False}$ صحیح است زیرا False هیچ مدلی ندارد بنابراین هر جمله‌ای را استلزام می‌کند و از آنجا که Trueدر تمام مدلها صحیح است، توسط هر جمله‌ی می‌تواند استلزام شود. **ب)** $\text{False} \models \text{True}$ طبق همان توضیحات قسمت الف، نادرست است. **ج)** عبارت $(A \wedge B) \models (A \Leftrightarrow B)$ درست است زیرا عبارت سمت چپ دقیقا یک مدل از دو مدل سمت راست می‌باشد. **د)** $A \Leftrightarrow B \models A \vee B$ نادرست است زیرا در یکی از مدلهای $A \Leftrightarrow B$، هم A و هم B نادرست هستند که نمی‌توانند $(A \vee B)$ را ارضا کنند. **هـ)** $A \Leftrightarrow B \models \neg A \vee B$ درست است زیرا سمت راست همان $A \Rightarrow B$ است که یکی از عطفهای عبارت سمت چپ یعنی $A \Leftrightarrow B$ می‌باشد. **و)** عبارت $(A \wedge B) \Rightarrow C \models (A \Rightarrow C) \vee (B \Rightarrow C)$ درست است زیرا فقط در صورتی عبارت سمت راست نادرست می‌شود که هردو عبارت فصلی نادرست باشند بعنوان مثال زمانی که A و B درست و C نادرست باشد، سمت چپ نیز نادرست می‌شود. البته این عبارت کمی گمراه‌کننده است و اگر نماد $\Rightarrow$ بصورت علت محسوب شود، این عبارت اینگونه نخواهد بود. **ز)** $(C \vee (\neg A \wedge \neg B)) \equiv ((A \Rightarrow C) \wedge (B \Rightarrow C))$ درست است، می‌توانید توسط جدول‌درستی و یا قواعد هم‌ارزی (شکل ۱۱.۷) آن را اثبات کنید. **ح)** $(A \vee B) \models (A \vee B) \wedge (\neg C \vee \neg D \vee E)$ درست است. حذف یک عطف مدلهای بیشتری را دربرمی‌گیرد. **ط)** $(A \vee B) \wedge (\neg D \vee E) \models (A \vee B) \wedge (\neg C \vee \neg D \vee E)$ نادرست است. حذف یک فصل مدلهای کمتری را در برمی‌گیرد. **ی)** $(A \vee B) \wedge \neg (A \Rightarrow B)$ ارضاپذیر بوده و مدل دارای A و $\neg B$ می‌باشد. **ک)** $(\neg A \vee B) \wedge (A \Leftrightarrow B)$ ارضاپذیر است. عبارت سمت راست از سمت چپ استلزام می‌شود بنابراین مدلها از $A \Leftrightarrow B$ هستند. **ل)** $C \Leftrightarrow (A \Leftrightarrow B)$ دارای همان تعداد مدل با

عبارت A⇔B می‌باشد. نیمی از مدلهای A⇔B عبارت C⇔(A⇔B) را ارضا می‌کنند و نیمی دیگر از مدلها ارضا نمی‌کنند. بنابراین تعداد مدلها و غیر مدلها برابر است.

**؟ ۵.۷ هر یک از این عبارات را اثبات کنید: الف) a صحیح است اگر و فقط اگر $a \models$ True. ب) به ازای هر $a$، داریم $a \models$ False. ج) $\beta \models a$ برقرار است اگر و فقط اگر عبارت ($a \Rightarrow \beta$) برقرار باشد. د) $a \equiv \beta$ برقرار است اگر و فقط اگر عبارت ($a \Leftrightarrow \beta$) برقرار باشد. هـ) $\beta \models a$ برقرار است اگر و فقط اگر عبارت ($\beta \wedge \neg a$) برقرار باشد. *(این تمرین مشابه تمرین ۴.۷ در ویرایش دوم است)***

☑ حل: یادآوری می‌کنیم که منظور از عبارت $a \models \beta$ آن است که اگر در دنیای مساله، مدلی طبق جمله $a$ صحیح بود، آنگاه در جمله $\beta$ نیز صحیح خواهد بود. (به عنوان مثال برای دنیای جاروبرقی جملاتی نظیر «$a$- در خانهٔ سمت چپ آشغال نیست و جاروبرقی در خانه سمت راست قرار دارد» و «$\beta$- جاروبرقی در خانه سمت راست قرار دارد» دو جمله‌ای هستند که $a$ دارای دو مدل‌ممکن و $\beta$ دارای چهار مدل‌ممکن می‌باشد. منظور از مدل هر چیدمانی از وضعیت‌های دنیاست که با جمله مربوطه مطابقت داشته باشد. در مورد این مثال $a \models \beta$ برقرار است. می‌توانید با ترسیم این چهار مدل و خط‌کشیدن دور محدوده مدل‌های هر جمله به این موضوع پی ببرید. اگر محدوده هر جمله درون محدوده جمله دیگر واقع شود، آنگاه آنرا ایجاب می‌کند.) بنابراین:
**الف)** اگر هر دنیای دلخواهی را در نظر بگیرید، جمله True فقط به ازای مدل واقعی دنیا صحیح است. به عنوان مثال در دنیای جاروبرقی، مدل‌هایی از دنیا در محدوده جمله True قرار می‌گیرند که حقیقت واقعی دنیا هستند یعنی موقعیت حقیقی آشغال و جاروبرقی (می‌توانید آن را برای خود در نظر بگیرید). حال فرض کنید برای جمله «$a$- در خانهٔ سمت‌چپ آشغال نیست و جاروبرقی در خانه سمت‌راست قرار دارد»، 2 مدل‌ممکن آنرا درون محدوده‌ای با نام $a$ قرار می‌دهیم. حال اگر جمله $a$ صحیح باشد، یعنی با واقعیت دنیا تطبیق دارد پس مسلم است که می‌توان مدل‌واقعی دنیا را درون محدوده $a$ پیدا کرد؛ پس داریم $a \models$ True. اگر در مورد درستی مدل‌های واقع در محدوده $a$ اطلاعی نداشته باشیم ولی بدانیم که $a \models$ True، پس نتیجه می‌شود که مدل حقیقی دنیا زیرمجموعه‌ای از محدوده $a$ است پس حتما باید $a$ صحیح باشد. پیش‌رو: اگر آلفا در تمام مدلها معتبر باشد، آنگاه در تمام مدلهای True برقرار است. پس‌رو: اگر $a \models$ True آنگاه $a$ بایستی در تمام مدلهای True یعنی در تمام مدلها برقرار باشد. پس آلفا معتبر است. **ب)** False در هیچ مدلی برقرار نیست و طبق تعریف نماد ایجاب گفتیم که اگر مدلی در بخش سمت چپ وجود داشت، باید در سمت راست هم وجود داشته باشد تا بگوییم رابطه برقرار است. ولی در مورد عدم‌وجود چیزی گفته نشد. بنابراین $a$ در هر مدلی که False برقرار نباشد، وجود دارد. **ج)** عبارت $a \Rightarrow \beta$ در مدلی برقرار است که یا $\beta$ یا $\neg a$ برقرار باشد که با تعریف ایجاب یکسان است. **د)** اعمال اثبات قسمت ج در دو طرف. **هـ)** این مورد همان عبارت قسمت ج است زیرا جدول درستی $a \wedge \neg \beta$ دقیقا با جدول درستی $a \Rightarrow \beta$ یکسان است. و یا اینکه هر دو طرف معادل‌اند با اینکه مدلی وجود ندارد که آلفا درست و بتا نادرست باشد.

**؟ ۶.۷ برای هر یک از این موارد مثالی ذکر کنید و یا آن را اثبات کنید. الف) اگر $\gamma \models \alpha$ یا $\gamma \models \beta$ (یا هر دو) آنگاه $\gamma \models (\alpha \wedge \beta)$ ب) اگر $(\beta \wedge \gamma) \models \alpha$ آنگاه $\beta \models \alpha$ و $\gamma \models \alpha$ ج) اگر $(\beta \vee \gamma) \models \alpha$ آنگاه $\beta \models \alpha$ یا $\gamma \models \alpha$ (یا هر دو)**

☑ حل: **الف)** درست. این عبارت از خاصیت یکنواختی[19] نتیجه می‌شود. **ب)** درست. اگر $\beta \wedge \gamma$ در هر مدلی از $\alpha$ درست باشد آنگاه هم $\beta$ و هم $\gamma$ در هر مدلی از $\alpha$ درست خواهند بود بنابراین $\beta \models \alpha$ و $\gamma \models \alpha$. **ج)** نادرست. فرض کنید $A \equiv \beta$ و $A \equiv \neg \gamma$ باشد.

**؟ ۷.۷ فرض کنید تنها 4 گزاره با نام‌های D,C,B,A داریم. چه تعداد مدل برای این عبارات وجود دارد: الف) $(A \wedge B) \vee (B \wedge C)$ ب) $A \vee B$ ج) $A \Leftrightarrow B \Leftrightarrow C$ *(این تمرین مشابه تمرین ۵.۷ در ویرایش دوم است)***

☑ حل: برای این منظور کافی است برای هر عبارت، جدول درستی تشکیل داده و تعداد سطرهای هر عبارت که نتیجه True دارد را بشماریم. ولی به موارد زیر دقت کنید: **الف)** یک عبارت فقط درصورتی نادرست است که B و C نادرست باشند. که در 4 مورد برای A و D رخ داده است که 12 بار می‌شود. **ب)** یک عبارت فقط درصورتی نادرست است که A و B و C و D نادرست باشند.که در یک مورد رخ داده و 15 بار می‌شود. **ج)** چهار عطف آخر مدلی را نشان می‌دهد که در آن عطف اول نادرست است، پس 0.

**؟ ۸.۷ در این فصل چهار رابط منطقی باینری برای اتصال عبارات تعریف شد. الف) آیا رابط دیگری نیز وجود دارد که مفید باشد؟ ب) تعداد رابطهای منطقی باینری چیست ؟ ج) چرا برخی رابطها خیلی مفید نیستند؟ *(این تمرین مشابه تمرین ۶.۷ در ویرایش دوم است)***

☑ حل: منظور از رابط منطقی نمادی است که بتواند دو عبارت منطقی مانند $p$ و $q$ را به هم وصل کرده و طبق تعریف خود دارای یک جدول درستی باشد تا به ازای 4 حالت مختلف دو عبارت ([p=F,q=F]، [p=T,q=F]، [p=F,q=T]، [p=T,q=T]) نشان دهد که نتیجه درست و یا نادرست است. از آنجا که برای دو عبارت نظیر $p$ و $q$ جدولی با 4 سطر داریم، می‌توانیم به ازای هر ترکیب مختلف T و F بودن نتیجه، یک رابط تعریف کنیم یعنی آن رابط باعث می‌شود تا آن ترکیب یکتا رخ دهد. با توجه به اینکه هر کدام از این 4 سطر می‌توانند درست یا نادرست باشند بنابراین $2^4=16$ حالت مختلف برای ستون نتیجه خواهیم داشت. هر یک از این جداول درستی را، یک رابط می‌نامیم پس 16 رابط ممکن وجود خواهد داشت. شش مورد از این رابطها اهمیت ندارند زیرا یک یا هر دو ورودی را نادیده می‌گیرند مانند $Q$، $\neg P$، $P$، False و True. چهار مورد دیگر

از رابطها را در این فصل مطالعه کردیم که عبارتند از ∧، ∨، ⇒، ⇔. از شش مورد مابقی نیز استلزام معکوس (⇐ به جای ⇒) مفید می‌باشد. پنج مورد دیگر نقیض چهار عمل اصلی یعنی ∨، ∧، ⇒، ⇔، ⇐ هستند.(به نقیض دو مورد اول به ترتیب نام‌های NAND و NOR داده‌اند).

**؟ ۹.۷ با استفاده از روش انتخابی خود در مورد اعتبار هر یک از هم‌ارزی‌های شکل ۱۱.۷ تحقیق کنید.** ***(این تمرین مشابه تمرین ۷.۷ در ویرایش دوم است)***

☑ حل: در اینجا جهت صرفه‌جویی در پاسخ، فقط چهار جدول درستی اولیه را نشان می‌دهیم:

```
> (truth-table "P ^ Q <=> Q ^ P")
----------------------------------------
 P  Q  P ^ Q  Q ^ P  (P ^ Q) <=> (Q ^ P)
----------------------------------------
 F  F    F      F          \(true\)
 T  F    F      F           T
 F  T    F      F           T
 T  T    T      T           T
----------------------------------------
NIL
```

```
> (truth-table "P | Q <=> Q | P")
----------------------------------------
 P  Q  P | Q  Q | P  (P | Q) <=> (Q | P)
----------------------------------------
 F  F    F      F           T
 T  F    T      T           T
 F  T    T      T           T
 T  T    T      T           T
----------------------------------------
NIL
```

```
> (truth-table "P ^ (Q ^ R) <=> (P ^ Q) ^ R")
------------------------------------------------------------------------
 P  Q  R  Q ^ R  P ^ (Q ^ R)  P ^ Q ^ R  (P ^ (Q ^ R)) <=> (P ^ Q ^ R)
------------------------------------------------------------------------
 F  F  F    F         F           F                    T
 T  F  F    F         F           F                    T
 F  T  F    F         F           F                    T
 T  T  F    F         F           F                    T
 F  F  T    F         F           F                    T
 T  F  T    F         F           F                    T
 F  T  T    T         F           F                    T
 T  T  T    T         T           T                    T
------------------------------------------------------------------------
NIL
```

```
> (truth-table "P | (Q | R) <=> (P | Q) | R")
------------------------------------------------------------------------
 P  Q  R  Q | R  P | (Q | R)  P | Q | R  (P | (Q | R)) <=> (P | Q | R)
------------------------------------------------------------------------
 F  F  F    F         F           F                    T
 T  F  F    F         T           T                    T
 F  T  F    T         T           T                    T
 T  T  F    T         T           T                    T
 F  F  T    T         T           T                    T
 T  F  T    T         T           T                    T
 F  T  T    T         T           T                    T
 T  T  T    T         T           T                    T
------------------------------------------------------------------------
NIL
```

برای سایر عبارات، تنها نشان می‌دهیم که آنها طبق تابع Validity معتبر هستند:

```
> (validity "~~P <=> P")
VALID
> (validity "P => Q <=> ~Q => ~P")
VALID
> (validity "P => Q <=> ~P | Q")
VALID
> (validity "(P <=> Q) <=> (P => Q) ^ (Q => P)")
VALID
> (validity "~(P ^ Q) <=> ~P | ~Q")
VALID
> (validity "~(P | Q) <=> ~P ^ ~Q")
VALID
> (validity "P ^ (Q | R) <=> (P ^ Q) | (P ^ R)")
VALID
> (validity "P | (Q ^ R) <=> (P | Q) ^ (P | R)")
VALID
```

**؟ ۷. ۱۰ کدامیک از عبارات زیر معتبر، ارضاپذیر و یا هیچ‌کدام هستند. نظر خود را با استفاده از جدول‌درستی و یا قواعد هم ارزی شکل ۱۱.۷ اثبات کنید:**

**الف) Smoke⇒Smoke ب) Smoke⇒Fire ج) (¬Smoke⇒¬Fire) ⇒ (Smoke⇒Fire)**

**د) Smoke ∨ Fire ∨¬ Fire هـ) ((Smoke⇒Fire) ∨ (Heat ⇒Fire)) ⇔ ((Smoke ∧Heat) ⇒Fire)**

**و) ((Smoke∧Heat) ⇒Fire) ⇒ (Smoke ⇒Fire) ز) Big ∨ Dumb ∨ (Big ⇒Dumb)**

***(این تمرین مشابه تمرین ۸.۷ در ویرایش دوم است)***

☑ حل: قسمت الف، د، هـ و، ز معتبر بوده و قسمت ب و ج هیچکدام می‌باشد. بسیاری از افراد در مورد قسمت هـ و ز به خاطر وجود علت و معلول در استلزام دچار اشتباه می‌شوند. بنابراین در قسمت هـ احساس میشود که ترکیب دود و گرما منجر به آتش می‌شود و دلیلی وجود ندارد که یکی از آنها به تنهایی بتواند منجر به آتش شود. بطور مشابه در قسمت ز، به نظر می‌رسد که هیچ رابطه‌ای بین Dumb و Big وجود ندارد بنابراین عبارت بایستی ارضاپذیر بوده ولی معتبر نیست. به هر حال این موارد اشتباه هستند زیرا استلزام نوعی علت و معلول نیست فقط نوعی فصل محسوب می‌شود. P⇒Q همان ¬P∨Q می‌باشد. بنابراین Big∨Dumb∨(Big⇒Dumb) معادل است با Big∨Dumb∨¬Big∨Dumb که هم‌ارز است با Big∨¬Big∨Dumb و همواره صحیح است چه Big درست باشد و چه نادرست، بنابراین عبارت معتبر است.

**؟ ۱۱.۷ هر جمله در منطق گزاره‌ای معادل است با آنکه دنیایی داشته باشیم که مورد نادرست در آن وجود نداشته باشد. طبق این گفته اثبات کنید که می‌توان هر جمله را به صورت فرم CNF نوشت. *(این تمرین مشابه تمرین ۱۰.۷ در ویرایش دوم است)***

☑ حل: هر دنیا می‌تواند به عنوان ترکیب عطفی نمادها نظیر (A∧C∧E) نوشته شود. به طوری که نماد نقیض در عطف نداشته باشیم و عبارتی نظیر ¬(A∧C∧E) باید به صورت (¬A∨¬C∨¬E) بازنویسی شود. در این صورت یک جزء از فرم CNF تشکیل می‌شود و با ترکیب عطفی این اجزا یک عبارت کامل به فرم CNF شکل می‌گیرد و می‌توان تمام جملات ممکن برای دنیا را ذکر نمود.

**؟ ۱۲.۷ با استفاده از رزولوشن (قاعده تحلیل)، عبارت ¬A∧¬B را از بندهای تمرین ۲۰.۷ اثبات کنید.**

☑ حل: برای اثبات یک ترکیب‌عطفی، بایستی هر کدام از الفاظ را جداگانه اثبات کنیم. برای اثبات ¬B طبق برهان‌خلف، نقیض آن یعنی S7:B را اضافه کنید. اگر در انتها به بندی تهی منجر شود یعنی این فرض نادرست بوده و ¬B برقرار است. • S7 را توسط S5 حل کنید و S8 :F نتیجه می‌شود. • S7 را توسط S6 حل کنید و S9 :C نتیجه می‌شود. • S8 را توسط S3 حل کنید و S10 :(¬C∨¬B) نتیجه می‌شود. • S9 را توسط S10 حل کنید و S11:¬B نتیجه می‌شود. • S7 را توسط S11 حل کنید و بند خالی نتیجه می‌شود.

مجددا برای اثبات ¬A، نقیض آن یعنی S7:A را به مجموعه بندها اضافه کنید. • S7 را توسط اولین بند از S1 حل کنید و S8:(B∨E) نتیجه می‌شود. • S8 را توسط S4 حل کنید وS9:B نتیجه می‌شود. • طبق مراحل فوق پیش روید تا بند خالی حاصل شود. پس هر دو بند برقرار هستند و نتیجه حاصل از اشتراک آنها نیز برقرار میباشد.

**؟ ۱۳.۷ این تمرین به بررسی رابطهٔ بین بند و جملات استلزام می‌پردازد. الف) نشان دهید که بند $(\neg P_1 \vee \ldots \vee \neg P_m \vee Q)$ به طور منطقی با جملهٔ استلزام $(P_1 \wedge \ldots \wedge P_m \Rightarrow Q)$ هم ارز است. ب) نشان دهید که هر بند (برخلاف تعداد الفاظ مثبت) می‌تواند به صورت $(P_1 \wedge \ldots \wedge P_m) \Rightarrow (Q_1 \vee \ldots \vee Q_n)$ نوشته شود که $Q$ها و $P$ها نمادهای گزاره‌ای می‌باشند. به پایگاه‌دانشی شامل این جملات، شکل نرمال استلزامی یا شکل کوالسکی می‌گویند. ج) قاعده تحلیل‌کامل را برای جملاتی به فرم نرمال استلزامی، بنویسید. *(این تمرین مشابه تمرین ۱۲.۷ در ویرایش دوم است)***

☑ حل: **الف)** عبارت $P \Rightarrow Q$ هم‌ارز است با $\neg P \vee Q$ که در آن علامت استلزام حذف شده است (شکل ۱۱.۷) و عبارت $(P_1 \wedge \ldots \wedge P_m) \Rightarrow Q$ طبق قاعده مورگان هم‌ارز است با $(\neg P_1 \vee \ldots \vee \neg P_m \vee Q)$. بنابراین عبارت $(\neg P_1 \vee \ldots \vee \neg P_m)$ هم‌ارز است با $(P_1 \wedge \ldots \wedge P_m) \Rightarrow Q$ **ب)** یک بند می‌تواند الفاظ مثبت و منفی داشته باشد. آنها را به صورت $(\neg P_1 \vee \ldots \vee P_m \vee Q_1 \vee \ldots \vee Q_n)$ مرتب کرده و با قرار دادن $Q = Q_1 \vee \ldots \vee Q_n$ داریم: $(\neg P_1 \vee \ldots \vee P_m \vee Q_1 \vee \ldots \vee Q_n)$ که هم‌ارز با $(P_1 \wedge \ldots \wedge P_m) \Rightarrow Q_1 \vee \ldots \vee Q_n$ است. **ج)** برای عبارات اتمیک $p_i$ ،$q_i$ ،$r_i$ و $s_i$ که در آنها $\text{UNIFY}(p_j, q_k) = \theta$ داریم:

$$\frac{p_1 \wedge \ldots p_j \ldots \wedge p_{n_1} \Rightarrow r_1 \vee \ldots r_{n_2} \qquad s_1 \wedge \ldots \wedge s_{n_3} \Rightarrow q_1 \vee \ldots q_k \ldots \vee q_{n_4}}{\text{SUBST}(\theta, (p_1 \wedge \ldots p_{j-1} \wedge p_{j+1} \wedge p_{n_1} \wedge s_1 \wedge \ldots s_{n_3} \Rightarrow r_1 \vee \ldots r_{n_2} \vee q_1 \vee \ldots q_{k-1} \vee q_{k+1} \vee \ldots \vee q_{n_4}))}$$

**؟ ۱۴.۷ بر طبق نظر برخی دانشمندان سیاسی، یک شخص سیاستمدار افراطی (R) در صورتی انتخاب (E) می‌شود، اگر پیرو سنتهای قدیمی (C) باشد. و درغیراینصورت انتخاب نمی‌شود. الف) کدامیک از این موارد نمایش صحیحی از این ادعاست؟ (i** $(R \wedge E) \Leftrightarrow C$ **(ii** $R \Rightarrow (E \Leftrightarrow C)$ **(iii** $R \Rightarrow ((C \Rightarrow E) \vee \neg E)$ **ب) کدامیک از عبارات قسمت الف را می‌توان به فرم هورن بیان نمود.**

☑ حل: **الف)** نمایش صحیح جملهٔ «یک سیاستمدار افراطی در صورتی انتخاب میشود که پیرو سنتهای قدیمی باشد و در غیراینصورت انتخاب نمی‌شود» عبارتست از:

i) $(R \wedge E) \iff C$

خیر این عبارت مربوط به ادعای دیگری است که «تمام پیروان سنتهای قدیمی سیاستمدار افراطی هستند» که این جمله مورد نظر ما نیست.

ii) $R \Rightarrow (E \iff C)$

بله، این عبارت می‌گوید اگر شخصی سیاستمدار افراطی است آنگاه منتخب است اگر و فقط اگر پیرو سنت باشد.

iii) $R \Rightarrow ((C \Rightarrow E) \vee \neg E)$

خیر، این عبارت معادل است با $\neg R \vee \neg C \vee E \vee \neg E$ که با هر فرضیه‌ای صحیح می‌باشد.

**ب)** فرم هورن: i) بله:

$$\begin{aligned}(R \wedge E) \iff C &\equiv ((R \wedge E) \Rightarrow C) \wedge (C \Rightarrow (R \wedge E)) \\ &\equiv ((R \wedge E) \Rightarrow C) \wedge (C \Rightarrow R) \wedge (C \Rightarrow E)\end{aligned}$$

ii) بله:

$$\begin{aligned}R \Rightarrow (E \iff C) &\equiv R \Rightarrow ((E \Rightarrow C) \wedge (C \Rightarrow E)) \\ &\equiv \neg R \vee ((\neg E \vee C) \wedge (\neg C \vee E)) \\ &\equiv (\neg R \vee \neg E \vee C) \wedge (\neg R \vee \neg C \vee E)\end{aligned}$$

iii) بله بعنوان مثال $True \Rightarrow True$

**؟ ۱۵.۷ این تمرین بازنمایی مسائل ارضاپذیر (SAT) را در قالب CSP در نظر می‌گیرد. الف) گراف محدودیت را برای مسأله SAT با $n = 5$ ترسیم کنید: $(\neg X_1 \vee X_2) \wedge (\neg X_2 \vee X_3) \wedge \ldots \wedge (\neg X_{n-1} \vee X_n)$ ب) تعداد راه‌حل‌های این مسأله SAT در حالت کلی بصورت تابعی بر حسب $n$ چیست؟ ج) فرض کنید BACKTRACKING-SEARCH را برای یافتن تمام راه‌حلهای مسئله ارضای محدودیت SAT در قسمت الف استفاده کرده‌ایم. (برای یافتن تمام راه‌حلهای CSP، باید روش الگوریتم پایه را کمی تغییر داد تا پس از یافتن هر راه‌حل، مجددا به کار خود ادامه دهد. فرض کنید متغیرها به ترتیب $X_1$و ... و$X_n$ و همچنین false قبل از true مرتب شده باشند). چقدر زمان می‌برد تا الگوریتم خاتمه یابد؟ (یک عبارت $O(.)$ بصورت تابعی بر حسب $n$ بنویسید) د) می‌دانیم که یک مسأله SAT به شکل هورن می‌تواند توسط زنجیره‌ای پیش‌رو (انتشار واحد) در زمانی خطی حل شود. همچنین می‌دانیم هر مسأله‌ی ارضامحدودیت‌دودویی با ساختار درختی که دامنه‌ای گسسته و متناهی داشته باشد، نیز در زمانی خطی برحسب تعداد متغیرها حل می‌شود. (بخش ۵.۶). آیا این دو حقیقت به یکدیگر مرتبط هستند؟ بحث کنید.**

☑ حل: **الف)** این گراف، بطور ساده زنجیره‌ای متصل از 5 گره است که به ازای هر متغیر یک گره لحاظ شده باشد. **ب)** تعداد $n+1$ راه‌حل وجود دارد. به ازای هر $X_i$ که صحیح باشد، تمام زیردنباله‌های $X_j$ باید صحیح باشند. بنابراین راه‌حلها عبارتند از: رخدادن i بار غلط بعد از $(n\text{-}i)$ بار درست برای $i = 0, \ldots, n$ ، که تعداد آنها همان $n+1$ خواهد شد. **ج)** پیچیدگی بصورت $O(n^2)$ می‌باشد که کمی نیاز به مهارت دارد، قسمتی از درخت باینری کامل که در طی جستجو بررسی شده است را در نظر بگیرید. الگوریتم بایستی تمام دنباله راه‌حلهایی که قسمتی از درخت را با اندازهٔ درجهٔ 2 پوشش می‌دهند، سپری کند. شاخه‌های شکست‌خورده پس از انتساب true به متغیر، false شده‌اند. بنابراین تناقضات به سرعت مشخص می‌شوند بنابراین هزینه از درجهٔ 2 بیشتر نمی‌شود. **د)** این حقایق بطور آشکار به یکدیگر مربوط نیستند. مسأله استنتاج منطقی فرم هورن نیازی به شرط درخت بودن گراف ندارد. پیچیدگی خطی از طبیعت محدودیت (استلزام) ایجاد میشود نه از ساختار مسأله.

**؟ ۱۶.۷ توضیح دهید که چرا هر بند گزاره‌ای غیرتهی، به خودی خود ارضاپذیر است؟ اثبات کنید که هر مجموعه بند 3-SAT ارضاپذیر است با این فرض که هر بند دقیقا سه متغیر مجزا داشته باشد. کوچکترین مجموعه بندهایی که ارضا ناپذیر هستند، چیست؟ یک مجموعه مثال بزنید.**

☑ حل: یک بند، ترکیب فصلی الفاظ است که مدل‌های آن، اجتماع مجموعهٔ مدل‌های هر لفظ است و هر لفظ نیمی از مدلهای ممکن را ارضا می‌کند.(توجه داشته باشید که false ارضاپذیر نیست و نام دیگری از بند تهی محسوب می‌شود.). پس به خودی خود ارضاپذیر است. یک بند 3-SAT با سه متغیر مجزا، باعث می‌شود تا 1/8 از مدل‌های ممکن غیرمحتمل شوند بنابراین 5 بند نمی‌تواند بیش از 5/8 از مدلها را غیرمحتمل کند. پس هشت بند برای محتمل بودن تمام مدلها نیاز است. فرض کنید متغیرهای C و B و A را داشته باشیم. بنابراین 8 مدل موجود بوده و می‌توان یک بند برای غیرمحتمل ساختن هر مدل بیان نمود. بعنوان مثال مدل {A=false,B=false,C=false} با بند (¬A∨¬B∨¬C) غیرمحتمل خواهد شد.

**؟ ۱۷.۷ یک عبارت گزاره‌ای به فرم 2-CNF شامل ترکیب‌عطفی بندهایی است که هر یک از آنها دقیقا دو لفظ داشته باشد. بعنوان مثال: (A∨B)∧(¬A∨C)∧(¬B∨D)∧(¬C∨G)∧(¬D∨G) الف) با استفاده از رزولوشن اثبات کنید که جملهٔ فوق، G را ایجاب می‌کند. ب) دو بند را «مفهومی مجزا» می‌نامیم اگر بطور منطقی هم‌ارز نباشند. چه تعداد بند مفهومی مجزا 2-CNF در مجموعه‌ای با $n$ نماد گزاره‌ای ممکن است؟ ج) با استفاده از پاسخ قسمت ب، اثبات کنید که در یک عبارت 2-CNF که بیش از $n$ نماد مجزا نداشته باشد، رزولوشن همواره در زمان چندجمله‌ای خاتمه می‌یابد. د) توضیح دهید که چرا ادعای قسمت ج برای 3-CNF کار نمی‌کند؟**

☑ حل: **الف)** ابتدا فرض می‌کنیم نقیض هدف یعنی ¬G درست باشد. از حل دو بند آخر، D¬ و C¬ تولید می‌شود. با حل بندهای سوم و دوم، B¬و A¬ تولید می‌شود. با حل دوبارهٔ بند اول، بندی تهی ایجاد می‌شود. پس فرض ¬G نادرست بوده و G ایجاب می‌شود. **ب)** می‌توان با نمادهای false، true و یا هیچکدام پاسخ داد که در اینجا جهت سادگی حذف شده‌اند. در ابتدا هر بند 2-CNF دو مکان برای گذاشتن الفاظ دارد. پس $2n$ ترکیب‌فصلی الفاظ وجود دارد که موجب $(2n)^2$ بند مجزا می‌شود. حال، بسیاری از این بندها از نظر مفهومی یکسانند که می‌توان آنها را در یک گروه قرار داد. اگر از ترتیب صرف‌نظر کنیم تعداد $C(2n,2)=(2n)(2n-1)/2=2n^2-n$ بند با دو لفظ متفاوت وجود دارد. تمام این بندها از نظر مفهوم مجزا هستند به جز آنهایی که معادل با true میباشند (بعنوان مثال A∨¬A). که تعداد آنها $n$ است. بنابراین تعداد $2n^2-2n+1$ بند بدون الفاظ مجزا وجود دارد. و $2n$ بند مجزا با الفاظ تکراری. بنابراین تعداد $2n^2+1$ بند مجزا در کل وجود دارد. **ج)** حل دو بند 2-CNF اندازهٔ بند را افزایش نمی‌دهد بنابراین رزولوشن فقط می‌تواند $O(n^2)$ بند مجزا قبل از پایان یافتن، تولید کند. **د)** در ابتدا توجه داشته باشید که تعداد بندهای 3-CNF به صورت $O(n^3)$ می‌باشد بنابراین نمیتوان در مورد غیرچندجمله‌ای بودن پیچیدگی بر حسب تعداد بندهای مختلف، بحث نمود. نکته کلیدی آن است که حل دو بند 3-CNF می‌تواند اندازهٔ بند را به 4 افزایش دهد و غیره. بنابراین اندازهٔ بند میتواند با داشتن $O(2^n)$ بند ممکن، به $O(n)$ افزایش یابد.

**؟ ۱۸.۷ این جملهٔ را در نظر بگیرید: [(Food∧Drinks) ⇒Party] ⇒[(Food ⇒Party)∨(Drinks ⇒Party)] الف) با استفاده از سرشماری، معتبر بودن، ارضاپذیری (ولی غیرمعتبربودن)، و غیرارضاپذیربودن جملهٔ فوق را تعیین کنید. ب) سمت چپ و سمت راست استلزام اصلی در عبارت فوق را به شکل CNF تبدیل نموده و هر مرحله را نشان دهید. توضیح دهید که نتایج بدست آمده چگونه پاسخ قسمت الف را تأیید می‌کند. ج) با استفاده از رزولوشن پاسخ قسمت الف را اثبات کنید.**

☑ حل: **الف)** یک جدول درستی برای این سه عنصر Party، Food و Drinks دارای هشت سطر با نتیجه درست می‌باشد و نشان می‌دهد که این عبارت برای تمام مدلها درست بوده و بنابراین معتبر است(جمله ای معتبر است که در تمام مدلها درست باشد مانند جملات بدیهی). **ب)** برای بخش سمت چپ داریم:

$$(Food \Rightarrow Party) \vee (Drinks \Rightarrow Party)$$
$$(\neg Food \vee Party) \vee (\neg Drinks \vee Party)$$
$$(\neg Food \vee Party \vee \neg Drinks \vee Party)$$
$$(\neg Food \vee \neg Drinks \vee Party)$$

و برای بخش سمت راست داریم:

$$(Food \wedge Drinks) \Rightarrow Party$$
$$\neg(Food \wedge Drinks) \vee Party$$
$$(\neg Food \vee \neg Drinks) \vee Party$$
$$(\neg Food \vee \neg Drinks \vee Party)$$

دو طرف در CNF یکسان هستند و بنابراین عبارت اصلی به شکل P⇒P است که برای تمام Pها معتبر است. **ج)** برای اثبات معتبر بودن یک عبارت می‌توان اثبات کرد که نقیض آن ارضاپذیر نیست. بعنوان مثال، نقیض آن را درنظربگیرید، آن را به فرم CNF تبدیل کنید و با استفاده از رزولوشن، تناقض را اثبات کنید. می‌توان از نتیجهٔ CNF فوق برای بخش سمت چپ استفاده کرد.

$$\neg[[(Food \Rightarrow Party) \vee (Drinks \Rightarrow Party)] \Rightarrow [(Food \wedge Drinks) \Rightarrow Party]]$$
$$[(Food \Rightarrow Party) \vee (Drinks \Rightarrow Party)] \wedge \neg[(Food \wedge Drinks) \Rightarrow Party]$$
$$(\neg Food \vee \neg Drinks \vee Party) \wedge Food \wedge Drinks \wedge \neg Party$$

هر یک از سه بند واحد با استفاده از اولین بند مجددا حل می‌شوند و بندی تهی می‌سازند.

**؟ ۱۹.۷ یک جمله به فرم نرمال‌فصلی (DNF) است اگر شامل ترکیب‌فصلی از عطف الفاظ باشد. بعنوان مثال جملهٔ (A∧B∧¬C)∨(¬A∧C)∨(B∧¬C) به فرم DNF است. الف) هر جملهٔ منطق گزاره‌ای هم‌ارز است با این گفته که برخی دنیاها در حقیقت درست هستند. با توجه به این گفته، اثبات کنید که هر جمله می‌تواند به فرم DNF نوشته شود. ب) الگوریتمی ارائه دهید که هر جملهٔ منطق گزاره‌ای را به فرم DNF تبدیل کند. (راهنمایی: این الگوریتم مشابه الگوریتم تبدیل به فرم CNF در بخش ۲.۵.۷ خواهد بود) ج) الگوریتم ساده‌ای ارائه دهید که بعنوان ورودی یک جملهٔ DNF را گرفته و در خروجی درصورت وجود، یک انتساب ارضاپذیر را برگرداند و در غیراینصورت بگوید که هیچ انتساب ارضاپذیری برای آن وجود ندارد. د) الگوریتم‌های قسمت ب و ج را به مجموعهٔ این جملات اعمال کنید: A⇒B و B⇒C و C⇒¬A هـ) با توجه به آنکه الگوریتم قسمت ب بسیار شبیه به الگوریتم تبدیل CNF است و الگوریتم قسمت ج بسیار ساده‌تر از هر الگوریتم حل CNF است چرا این تکنیک در استنتاج‌های خودکار بکار نمی‌رود؟**

☑ حل: **الف)** هر دنیای ممکن می‌تواند به صورت ترکیب‌عطفی تمام الفاظ موجود در مدل بیان شود و عبارت معادل است با ترکیب‌فصلی تمام این عطفها که یک عبارت DNF شکل می‌گیرد. **ب)** هر الگوریتم ساده‌ای می‌تواند تمام مدلهای ممکن که شامل اصطلاحاتی است که عبارت درست می‌شود را بشمارد. ولی این کار به زمانی نمایی نیاز دارد. می‌توان همان روش تبدیل به فرم CNF را برای تبدیل عبارات به فرم DNF بکار برد با این تفاوت که در این حالت بایستی در انتها، ∧ بر ∨ گسترش یافته باشد. **ج)** یک عبارت DNF در صورتی ارضاپذیر است که حداقل دارای یک بخش بدون الفاظ‌متناقض باشد. می‌توان این بررسی را در زمانی خطی انجام داد و یا حتی در همان زمان فرآیند تبدیل انجام داد. هر ترکیبی از آن بخش که سایر الفاظ آن را مقدار داده باشیم، یک مدل محسوب می‌شود. **د)** در اولین مرحله داریم:

$$(\neg A \vee B) \wedge (\neg B \vee C) \wedge (\neg C \vee \neg A)$$

تبدیل به DNF به معنای آن است که یک لفظ از هر بند بر داشته و به تمام روشهای ممکن، سایر اصطلاحات را تولید کنیم (8 در کل). انتخاب هر لفظ معادل است با انتخاب مقدار صحیح برای هر متغیر. بنابراین این فرآیند بسیار شبیه به شمردن تمام مدلهای ممکن خواهد بود. در اینجا اولین بخش (¬A∧¬B∧¬C) است که ارضاپذیر بودن آن بدیهی است. **هـ)** مشکل در گام نهایی حصول نتیجه در عبارت DNF با اندازهٔ نمایی است بنابراین هم به زمانی‌نمایی و هم فضایی‌نمایی نیازمندیم.

**؟ ۲۰.۷ مجموعه جملات زیر را به شکل بندی تبدیل کنید:**

**S1:** A⇔(B∨E)
**S2:** E⇒D
**S3:** C∧F⇒¬B
**S4:** E⇒B
**S5:** B⇒F
**S6:** B⇒C

**اجرای DPLL بر روی ترکیب‌عطفی این بندها را دنبال کنید.**

☑ حل: نمایش CNF به صورت زیر است:

S1: $(\neg A \vee B \vee E) \wedge (\neg B \vee A) \wedge (\neg E \vee A)$
S2: $(\neg E \vee D)$
S3: $(\neg C \vee \neg F \vee \neg B)$
S4: $(\neg E \vee B)$
S5: $(\neg B \vee F)$
S6: $(\neg B \vee C)$

در اینجا از ردیابی DPLL جهت آزمودن آن صرف‌نظر کردیم.

**؟ ۲۱.۷ به نظر شما یک جملهٔ 4-CNF که بطور تصادفی با $n$ نماد و $m$ بند تولید شده است نسبت به یک جملهٔ 3-CNF که بطور تصادفی با $n$ نماد و $m$ بند تولید شده است قابلیت حل بیشتری دارد یا کمتر؟ توضیح دهید.**

☑ حل: این جمله قابلیت حل بیشتری خواهد داشت: زیرا افزودن الفاظ به بندهای فصلی، ارضاپذیری آنها را ساده‌تر می‌کند.

**؟ ۲۲.۷ بازی رایانه‌ای مین یاب[۲۰] که بسیار مشهور است، ارتباط نزدیکی با دنیای وومپوز دارد. دنیای مین‌یاب شامل یک شبکه مستطیلی از N مربع با M مین مخفی درون آنهاست که هر مربع می‌تواند توسط کارگزار بررسی شود ولی در صورت وجود مین در آن خانه، کارگزار سریعا می‌میرد. در این بازی کارگزار پس از معاینهٔ هر خانه، عددی که نشان‌دهنده مجموع تعداد مین مجاور با آن خانه به طور افقی، عمودی و قطری است، را در آن خانه قرار می‌دهد. هدف این بازی آن است که تمام خانه‌های بدون مین را بدون مردن مشخص کنیم. الف) مقدار $X_{i,j}$ درست است اگر و فقط اگر مربع [i,j] دارای مین باشد. با استفاده از ترکیب منطقی گزاره‌های $X_{i,j}$، جمله‌ای بنویسید که نشان دهد دقیقا دو مین در مجاورت خانه [1,1] وجود دارد. ب) ادعای قسمت الف را تعمیم دهید و توضیح دهید که چگونه می‌توان یک جمله CNF ایجاد کرد که نشان دهد $k$ خانه از n همسایه شامل مین هستند. ج) به دقت توضیح دهید که یک کارگزار چگونه می‌تواند با استفاده از DPLL اثبات کند که یک خانه دارای مین است (یا نیست). با صرف نظر از اینکه تعداد کل مین‌ها در طی یافت‌ها باید M باشد. د) فرض کنید محدودیت تعداد کل مین‌ها به قسمت ب اضافه**

**شود، تعداد بندها چه ارتباطی با M و N دارند؟ روشی ارائه دهید که با تغییر DPLL نیازی به نمایش صریح محدودیت تعداد کل نباشد. هـ) در صورت درنظرگرفتن محدودیت تعداد کل مین‌ها، آیا تمام نتایج حاصل از قسمت ج نامعتبر است؟ و) مثال‌هایی از پیکره‌بندی مقادیر هر کاوش که محدودیت تعدادکل داشته باشد ارائه دهید به طوری که محتوای هر خانهٔ بدون مین دارای اطلاعاتی از یک خانهٔ دورتر نیز باشد. (راهنمایی: یک صفحهٔ N×1 را در نظربگیرید). *(این تمرین مشابه تمرین ۷.۱۱ در ویرایش دوم است)***

☑ حل: **الف)** این مورد یک ترکیب فصلی با 28 عملکرد فصل می‌باشد که هر کدام از آنها نشان‌دهندهٔ آن است که دو همسایه درست بوده و سایرین نادرست باشند. اولین ترکیب فصلی عبارتست از:

$$X_{2,2} \wedge X_{1,2} \wedge \neg X_{0,2} \wedge \neg X_{0,1} \wedge \neg X_{2,1} \wedge \neg X_{0,0} \wedge \neg X_{1,0} \wedge \neg X_{2,0}$$

هر کدام از 27 ترکیب فصلی دیگر، دو $X_{i,j}$ مختلف را جهت درست بودن انتخاب می‌کنند. **ب)** تعداد $\binom{n}{k}$ ترکیب فصلی وجود دارد که هر کدام نشان‌دهندهٔ آن است که $k$ تعداد از $n$ نماد درست بوده و سایرین غلط باشند. **ج)** به ازای هر سلول بررسی شده، عدد $n$ توسط بازی نتیجه می‌شود و یک جمله با $\binom{n}{8}$ فصل (به خاطر 8 همسایه هر خانه) ساخته می‌شود. تمام جملات را به هم وصل کنید. سپس از DPLL برای پاسخ به سؤال اینکه آیا به ازای جفت i,j دلخواه، این جمله شامل $X_{i,j}$ هست یا خیر، استفاده کنید. **د)** برای اعمال شرط «وجود تعداد کل مین‌ها به اندازهٔ M» باید یک ترکیب فصلی با $\binom{M}{N}$ عملکرد فصل که هر کدام به اندازهٔ N هستند، تشکیل دهیم. یادآوری می‌شود که: $\binom{M}{N} = M!/(M!-N!)$ است، بنابراین برای یک بازی مین‌یاب با 100 سلول و 20 مین، این مقدار بیش از $10^{39}$ خواهد بود و نمی‌تواند در کامپیوتر نشان داده شود. به هر حال می‌توان محدودیت تعداد کل مین‌ها را با الگوریتم DPLL اعمال نمود. پارامترهای *min* و *max* را به تابع DPLL اضافه می‌کنیم به طوری که آنها نشان‌دهندهٔ حداقل و حداکثر تعداد نمادهای انتساب نیافته در مدل باشند که بایستی در نهایت true شوند. برای حالتی که این محدودیت اعمال نشود بایستی برای این دو پارامتر به ترتیب مقادیر 0 , N اعمال شود. برای مسأله مین‌یاب، مقدار M هم برای *min* و هم برای *max* استفاده می‌شود. درون تابع DPLL اگر *min* کمتر از تعداد نمادهای باقی‌مانده باشد و یا *max* کمتر از صفر باشد، فورا تابع شکست خورده و مقدار false را برمی‌گرداند. برای هر فراخوانی بازگشتی از DPLL بایستی *max* و *min* با توجه به انتساب مقدار true به یک نماد، بروز شوند. هـ) با افزودن چنین قابلیتی به DPLL و رمزگذاری محدودیت تعداد کل مین‌ها هیچ نتیجه‌ای نامعتبر نیست. **و)** رشته‌ای از 1ها و خانه‌های بررسی‌نشده(که با علامت خط تیره مشخص شده است) را در نظر بگیرید:

|-|1|-|1|-|1|-|1|-|1|-|1|-|1|-|

اکنون دو مدل ممکن وجود دارد: مین‌ها در خانه‌هایی با تعداد زوج خط تیره قرار دارند یا در خانه‌هایی که تعداد خط تیره فرد است. با بررسی هر پایان، تعیین می‌شود که کدام سلول‌ها در پایان دور، خالی یا دارای مین هستند.

**؟ ۲۳.۷ چقدر طول می‌کشد تا عبارت $KB \models \alpha$ را با استفاده از DPLL اثبات کنید، به طوری که α یک لفظ موجود در KB باشد؟ توضیح دهید. *(این تمرین مشابه تمرین ۱۶.۷ در ویرایش دوم است)***

☑ حل: این زمان بستگی به تعداد نمادهای خالص به علاوه تعداد بندهای واحد دارد. طبق برهان خلف فرض می‌کنیم $KB \models \alpha$ نادرست باشد و سپس اثبات می‌کنیم که این فرض غلط است. می‌دانیم که $(KB \models \alpha)\neg$ معادل است با $KB \wedge \neg \alpha$ . طبق این جمله، الگوریتم در ابتدا تمام نمادهای خالص را حذف کرده سپس بر روی بندهای واحد کار می‌کند تا زمانی که به α برسد(که آن نیز بندی واحد است). در این زمان الگوریتم به سرعت تشخیص می‌دهد که هر انتخاب(درست یا نادرست) برای α منجر به شکست خواهد شد. این بدان معناست که فرض نقیض ابتدایی غلط بوده و مسأله اثبات شده است.

**؟ ۲۴.۷ عملکرد DPLL بر روی پایگاه‌دانش شکل ۱۶.۷ در راه اثبات Q را پیگیری نموده و عملکرد آن را با رفتار الگوریتم زنجیره‌ای پیش‌رو مقایسه کنید. *(این تمرین مشابه تمرین ۱۷.۷ در ویرایش سوم است)***

☑ حل: در اینجا از آوردن جزئیات پیگیری DPLL خودداری شده است ولی می‌توانید کد آنرا بیابید. ولی روال کار به طور ساده این گونه است: قاعده بند واحد در DPLL ما را مطمئن می‌سازد که تمام اتم‌های شناخته شده به سایر بندها منتشر شده‌اند.

**؟ ۲۵.۷ یک قاعدهٔ کلی حالت پسین برای پیش‌بینی قفل‌بودن بنویسید که قابل اعمال به تمام درها باشد. فرض کنید که تنها اقدامات ممکن شامل قفل‌بودن و غیرقفل‌بودن در است.**

☑ حل:

$$Locked^{t+1} \Leftrightarrow [Lock^t \vee (Locked^t \wedge \neg Unlock^t)]$$

**؟ ۲۶.۷ بخش ۱.۷.۷ برخی قواعد حالت‌پسین موردنیاز برای دنیای ومپوز را بیان می‌کند. قواعدی برای تمام نمادهای fluent باقیمانده بنویسید.**

☑ حل: موارد باقیمانده شامل چرخش(مانند $FacingEast$) و زنده‌بودن ومپوز($WumpusAlive$) می‌شود. قواعد حالت پسین عبارتند از:

$$FacingEast^{t+1} \Leftrightarrow (FacingEast^t \wedge \neg(TurnLeft^t \vee TurnRight^t)) \vee (FacingNorth^t \wedge TurnRight^t) \vee (FacingSouth^t \wedge TurnLeft^t)$$

$$WumpusAlive^{t+1} \Leftrightarrow WumpusAlive^t \wedge \neg(WumpusAhead^t \wedge HaveArrow^t \wedge Shoot^t)$$

برای $WumpusAhead$ نیازی به قاعده حالت پسین نداریم زیرا می‌توان آن را بطور همزمان از مکان کارگزار، مقدار چرخش و مکان ومپوز تعریف کرد. این تعریف بسیار خسته‌کننده خواهد بود وضعف منطق گزاره‌ای را نشان می‌دهد. لازم به ذکر است که در ویرایش دوم یک قاعدهٔ حالت‌پسین برای $WumpusAlive$ تعریف کردیم(به شکل مدار) که از $Scream$ جهت استدلال مرگ ومپوز استفاده می‌کرد و دیگر نیازی به توصیف پیچیدهٔ فیزیک پرتابی نیست. مانند قاعده ای که برای تخمین حالت کافی باشد ولی نه برای بازی کردن.

**؟ ۲۷.۷ کارگزار HYBRID-WUMPUS-AGENT را طوری تغییر دهید تا از روش تخمین حالـــت 1-CNF طبق بخش ۷. ۷. ۳ استفاده کند. در آن بخش از کتاب توضیح دادیم که یک کارگزار قادر به یافتن، نگهداری و استفاده از حالتهای باور پیچیده نظیر ترکیب $P_{3,1} \vee P_{2,2}$ نمی‌باشد. روشی ارائه دهید که بتوان این مشکل را با تعریف تعدادی نماد گزاره‌ای برطرف نموده و سپس آن را بر روی دنیای‌ومپوز امتحان کنید. آیا این روش کارایی کارگزار را بهبود می‌بخشد؟**

☑ حل: برای تعریف قواعد بایستی تغییراتی بیافزاییم مانند: $P_{3,1 \text{ or } 2,2} \Leftrightarrow P_{3,1} \vee P_{2,2}$

همچنین بایستی تعدادی لفظ به لیست الفاظ اضافه کنید که مقدار درست آنها در هر مرحله زمانی استنتاج شود. یک روش طبیعی برای گسترش نمایش 1-CNF آن است که عبارت بدون لفظ را بعنوان تست بیافزاییم. عبارتی که برای تست انتخاب می‌کنیم بستگی به استنتاج فعلی KB دارد. این راه‌کار در صورتی مفید است که تعداد جملاتی که برای تست موردنیاز است خیلی زیاد نباشد. بعنوان مثال می‌توان از پایگاه‌دانش سؤال پرسید که کدام خانه‌های مشاهده‌شده چاله دارند، کدام خانه‌های مشاهده‌شده احتمال وجود چاله دارند و کدام حالات دارای نسیم هستند (برای تکمیل آن نیاز به حالت باور 1-CNF داریم). بنابراین به‌ازای هر خانه دارای نسیم، عبارت «یکی از همسایگان این خانه احتمالا چاله است» را بررسی می‌کنیم. بعنوان مثال اگر در خانهٔ (2,1) نسیمی باشد بایستی عبارت $P_{3,1} \vee P_{2,2}$ را بررسی کنیم. طبق فیزیک ومپوز، این الفاظ درست است اگر و فقط اگر خانه‌های دارای نسیم در اطراف خود، چاله‌های ناشناخته داشته باشند.

# فصل ۸ (ویرایش سوم)

**؟ ۱.۸ یک پایگاه دانش منطقی، جهان را توسط مجموعه‌ای از جملات که هیچ ساختار روشنی ندارند، توصیف می‌کند. از طرف دیگر، در بازنمایی‌قیاسی، ساختار فیزیکی و ساختار آن چیزی که نمایش می‌دهد مرتبط می‌باشد. یک نقشه از جاده‌های کشورتان را به عنوان یک بازنمایی قیاسی از حقایق کشور در نظر بگیرید(این نقشه حقایق را توسط زبان نقشه بیان می‌کند). ساختار دوبعدی نقشه با ساختار دوبعدی سطح جاده متناظر است. الف) 5 نمونه از نمادهای موجود در زبان نقشه را بیان کنید. ب) یک جملهٔ صریح، جمله‌ای است که توسط سازندهٔ بازنمایی بیان شود. و یک جملهٔ غیرصریح، جمله‌ای است که با توجه به مشخصات بازنمایی‌قیاسی از جملات‌صریح نتیجه گرفته شود. سه نمونه از هر یک از جملات صریح و غیرصریح در زبان نقشه را بیان کنید. ج) سه نمونه از حقایقی در مورد ساختار فیزیکی کشور که قابل نمایش در زبان نقشه نیست، ارائه دهید. د) دو نمونه از حقایقی مثال بزنید که بیان آنها در زبان نقشه ساده‌تر از منطق مرتبهٔ اول باشد. هـ) دو نمونهٔ دیگر از بازنمایی‌قیاسی مفید مثال بزنید. مزایا و معایب هر یک از این زبان‌ها را بیان کنید. *(این تمرین مشابه تمرین ۱.۸ در ویرایش دوم است)***

☑ حل: این سؤال، راه‌حل‌های مختلفی دارد. تفاوت اصلی بین بازنمایی قیاسی و بازنمایی جمله‌ای در آن است که وقتی یک موضوع بدرستی رمزنگاری شود آنگاه بازنمایی قیاسی به طور خودکار نتایجی تولید می‌کند که به راحتی قابل درک است. **الف)** بسته به مقیاس و نوع نقشه، معمولا نمادهای زبان نقشه شامل این موارد می‌باشد: علامت‌های شهر و حومه، نمادهای جاده (انواع مختلف)، فانوس دریایی، بناهای تاریخی، جریان رودخانه، تقاطع آزاد راه‌ها و غیره ... **ب)** جملات صریح و غیرصریح: تفاوت آنها کمی نیاز به مهارت دارد. ولی ایده اصلی در آن است که زمانی که یک نماد در مکانی خاص از نقشه قرار می‌گیرد مسلم است که چیزی را به صراحت نشان می‌دهد(مثلا می‌گوید برج *Coit* در اینجا واقع است) ولی ساختار قیاسی برای بازنمایی نقشه از چندین جملهٔ غیرصریح برای بیان این موضوع استفاده می‌کند. جملات صریح: بنایی تاریخی بنام برج *Coit* در این مکان واقع است؛ خیابان *Lombard* حدودا از شرق به سمت غرب ادامه دارد؛ خلیج سان‌فرانسیسکو وجود داشته و به این شکل است. جملات غیرصریح: *Van Ness* بزرگتر از *Willard* شمالی است؛ اسکلهٔ *Fisherman* در شمال ناحیهٔ *Mission* قرار دارد؛ کوتاهترین مسیر حرکت بین برج *Coit* تا *Twin Peak* به فلان صورت است. **ج)** جملاتی که در زبان نقشه قابل بازنمایی نیستند: تپه *Telegraph* تقریبا مخروطی شکل بوه و 430 فوت ارتفاع دارد(با این فرض که نقشه نمادهای توپوگرافی را دربرنداشته باشد)؛ در سال 1980 هیچ پل ارتباطی بین سان‌فرانسیسکو تا کشور *Marin* وجود نداشت(نقشه نمی‌تواند تغییرات اطلاعات را نشان دهد)؛ فاصلهٔ *Walnut Greek* تا ایالت‌های شرقی یا غربی آن 680 می‌باشد(نقشه دارای اطلاعات مجزا نیست). **د)** جملاتی که در زبان نقشه ساده‌تر بیان می‌شوند: هر جمله‌ای که به سادگی در زبان نوشته شود برای این سؤال پاسخ خوبی نخواهد بود. هر عبارت زبانی که مربوط به ساختار فیزیکی سان‌فرانسیسکو شود (به عنوان مثال سان‌فرانسیسکو در انتهای یک شبه‌جزیره و مدخل خلیج واقع شده است) می‌تواند به سادگی در ساختار پیش‌بینی‌شدهٔ نقشه بگنجد زیرا نقشه برای این هدف طراحی شده است. حقایقی چون شکل خط‌ساحلی و یا مسیر یک جاده در زبان نقشه به بهترین نحو ممکن بیان می‌شوند. حتی می‌توان در مورد خط ساحلی روی نقشه جملات بسیار زیادی بیان نمود. یعنی به ازای هر نقطه جوهر بر روی نقشه جمله‌ای داشته باشیم به خصوص اگر نقشه به صورت دیجیتال ترسیم شده باشد. در این مورد مزیت نقشه آن است که علاوه بر سهولت استنتاج، برای محاسبات بصری انسان نیز مفید است. **هـ)** مثال‌هایی از سایر بازنمایی‌های قیاسی: • ضبط صدای آنالوگ بر روی نوار. مزایا: مدارهایی ساده می‌توانند صداها را ضبط و پخش مجدد کنند. معایب: ممکن است دچار خطا یا نویز شود، برای مجزا کردن صداهای مختلف یا حذف نویز به فرآیندی پیچیده نیازمندیم. • ساعت‌های باستانی. مزایا: به سرعت و راحت خوانده می‌شوند، جهت تعیین زمان در دسترس نیاز به محاسبات زیاد ندارد. معایب: خواندن زمان دقیق دشوار است، به سادگی نمی‌تواند واحدهای زمانی کوچک (مانند میلی ثانیه) را نشان دهد. • تمام انواع گراف‌ها، نمودار میله‌ای و نمودار دایره‌ای. مزایا: داده‌ها را بسیار فشرده می‌کند، تحلیل آنها ساده‌تر می‌شود، ارتباطات اطلاعاتی که بتوان به سادگی آنها را تفسیر کرد بهتر می‌شود. معایب: غیردقیق بوده و نمی‌تواند اطلاعات مجزا یا متناقض را نمایش دهد.

**؟ ۲.۸ پایگاه‌دانشی فقط شامل دو جمله $P(a)$ و $P(b)$ را در نظر بگیرید. آیا این پایگاه‌دانش $\forall x\ P(x)$را ایجاب می‌کند؟ پاسخ خود را بر اساس مدل‌ها توضیح دهید. *(این تمرین مشابه تمرین ۲.۸ در ویرایش دوم است)***

☑ حل: این پایگاه دانش نمی‌تواند $\forall x\ P(x)$را ایجاب کند. برای نشان دادن این موضوع باید مدلی ارائه دهیم که در آن خاصیت‌های $P(a)$ و $P(b)$ برقرار باشد ولی جمله $\forall x\ P(x)$ در آن غلط باشد. برای این منظور، مدلی با سه عنصر $a$، $b$ و $c$ در نظر بگیرید که فقط برای دو عنصر $a$ و $b$، خاصیت $P$ برقرار باشد. پس این جمله در حالت کلی برقرار نمی‌باشد.

**؟ ۳.۸ آیا جملهٔ $\exists x,y\ x=y$ معتبر است؟ توضیح دهید. *(این تمرین مشابه تمرین ۳.۸ در ویرایش دوم است)***

☑ حل: جملهٔ $\exists x,y\ x=y$ معتبر است. زیرا یک جمله در صورتی معتبر نامیده می‌شود که در تمام مدل‌ها و به ازای هرگونه مقداردهی صحیح باشد. جملاتی که دارای سور وجودی هستند در صورتی صحیح هستند که شرایطی یافت شود که به هر متغیر مقداری از دامنه آن تخصیص داده و آن جمله برقرار شود. بر طبق استاندارد مفاهیم FOL که در این فصل گفته شد، هر مدل

حداقل دارای یک دامنهٔ عنصر می‌باشد بنابراین می‌توان تفسیر کرد که به دو متغیر $x$ و $y$ اولین دامنهٔ عنصر را انتساب دهیم آنگاه $x=y$ صحیح خواهد شد.

**؟ ۴.۸ یک جملهٔ منطقی بنویسید که اگر در یک دنیا درست باشد، آن دنیا دقیقا دارای یک شی باشد.** ***(این تمرین مشابه تمرین ۴.۸ در ویرایش دوم است)***

☑ حل: عبارت $\forall x,y\ x=y$ قید می‌کند که دقیقا یک شی در دنیا موجود است. اگر فرض کنیم که در این دنیا دو شی موجود باشند یعنی $x$ و $y$ به اشیایی متفاوت منتسب شوند، آنگاه این جمله غلط خواهد شد.

**؟ ۵.۸ یک فرهنگ‌لغت نمادها شامل $c$ نماد ثابت، $P_k$ نماد مسند با درجهٔ $k$، $F_k$ نماد تابعی با درجهٔ $k$ که $1\le k\le A$ را در نظر بگیرید. اندازهٔ دامنه، ثابت و برابر $D$ می‌باشد. برای هر ترکیب مدل- تفسیر، هر نماد مسندی یا تابعی به ترتیب به یک رابطه یا تابع با همان درجه نگاشت می‌یابد. می‌توانید فرض کنید که توابع در مدل، ورودی‌های چندگانه دارند که در تابع هیچ مقداری نگرفته‌اند (به عنوان مثال این مقدار، شیء نامرئی است). فرمولی برای تعداد ترکیبات ممکن مدل- تفسیر در دامنهٔ $D$ بیابید. در مورد حذف ترکیب‌های تکراری نگران نباشید.** ***(این تمرین مشابه تمرین ۵.۸ در ویرایش دوم است)***

☑ حل: ما از ساده‌ترین روش شمارش که در آن کاری با ترکیبات تکراری ندارد استفاده می‌کنیم. از آنجا که تعداد نمادهای ثابت $c$ بوده و دامنه هریک دارای $D$ عضو است، پس برای نمادهای ثابت تعداد $D^c$ انتساب وجود دارد(به عنوان مثال فرض کنید نمادهای ثابت $a$ و $b$ و $c$ هر یک دارای دامنه true و false باشند. پس می‌توان به تعداد $2^3$ حالت مختلف به آنها مقدار داد). هر مسند با درجهٔ $k$ به یک رابطهٔ $k$- تایی یعنی رابطه‌ای با $k$ آرگومان نگاشته می‌شود که تعداد کل حالات ممکن $D^k$ خواهد بود. تعداد زیرمجموعه‌های $k$- عضوی آن، همان تعداد مورد نظر ما می‌باشد که برابر $2^{D^k}$ است. هر نماد تابعی با درجهٔ $k$ به یک تابع $k$- تایی نگاشته می‌شود و از آنجا که در دامنه ورودی‌های تابع می‌توان علاوه بر عناصر دامنه، عنصر نامرئی نیز داشته باشیم پس تعداد اعضای دامنه را $D+1$ در نظر می‌گیریم. پس تعداد $(D+1)^{D^k}$ تابع مختلف وجود خواهد داشت. بنابراین تعداد کل ترکیبات ممکن عبارتست از:

$$D^c \cdot \left(\sum_{k=1}^{A} 2^{D^k}\right) \cdot \left(\sum_{k=1}^{A} (D+1)^{D^k}\right)$$

به دو نکته توجه کنید: اول آنکه این تعداد متناهی است و دوم آنکه حداکثر درجهٔ $A$ پیچیده‌ترین پارامتری است که باید تعیین شود.

**؟ ۶.۸ کدامیک از این عبارات معتبر است؟**

**الف)** $(\forall z\ \exists y\ y=z)(\exists x\ x=x)$ **ب)** $\forall x\ P(x)\vee\neg P(x)$ **ج)** $\forall x\ Smart(x)\vee(x=x)$

☑ حل: معتبر بودن در منطق مرتبهٔ اول به معنای آن است که در تمام مدل‌های ممکن صحیح باشد:

**الف)** $(\exists x\ x=x) \Rightarrow (\forall y\ \exists z\ y=z)$

معتبر است. سمت چپ عبارت به‌خودی‌خود معتبر است در استاندارد FOL هر مدل حداقل دارای یک شی است بنابراین کل عبارت در صورتی معتبر است که اگر و فقط اگر سمت راست نیز معتبر باشد. (در غیراینصورت مدلی داریم که سمت چپ آن درست و سمت راست آن نادرست است). عبارت سمت‌راست معتبر است زیرا به ازای هر مقدار از $y$، متغیری با نام $z$ که همان مقدار $y$ را دارد، وجود دارد.

**ب)** $\forall x\ P(x)\vee\neg P(x)$

معتبر است. به ازای هر رابطه که با نام $P$ نمایش داده شود، هر شی $x$ یا در خود رابطه و یا در نقیض آن وجود دارد.

**ج)** $\forall x\ Smart(x)\vee(x=x)$

معتبر است. در هر مدلی با شی $x$ همواره رابطهٔ $x=x$ برقرار است بنابراین طبق خاصیت ترکیب فصلی، با درست بودن این بخش، کل عبارت درست بوده و کاری نداریم که آیا $Smart(x)$ برقرار هست یا خیر.

**؟ ۷.۸ نسخه‌ای از منطق مرتبهٔ اول را در نظر بگیرید که در آن دامنهٔ تهی مجاز باشد. حداقل دو مثال از جملاتی ارائه دهید که طبق نسخهٔ استاندارد معتبر هستند ولی در نسخهٔ جدید خیر. توضیح دهید که کدام نتایج، مثالهای شما را شهودی‌تر می‌کند.**

☑ حل: این نسخه از FOL، اولین بار توسط Mostowski(1951) با عنوان منطق آزاد (Lambert,1967) مطالعه شد. با توجه به طبیعت مقادیر درستی که استفاده از تهی در ترکیبات عطفی منجر به true و استفاده از تهی در ترکیبات فصلی منجر false می‌شود، لذا هر عبارتی با سور عمومی در مدلهای تهی، درست میباشد و هر عبارتی با سور وجودی نادرست. این نسخه باید برای این موضوع که نمادهای ثابت هیچ مرجعی در یک مدل تهی ندارد، تدبیری بیاندیشد. اگر بخواهیم جملاتی مثال بزنیم که در نسخهٔ استاندارد معتبر بوده و در نسخهٔ منطق‌آزاد معتبر نباشند می‌توان به $\exists x\ x=x$ و جملهٔ $[\forall x\ P(x)]\Rightarrow[\exists x\ P(x)]$ اشاره نمود. هم‌ارزی $\exists x\phi\vee\psi$ و $\phi\vee\exists x\psi$ در صورت آزاد نبودن $x$ درست نمی‌باشد و نمیتوان از آن برای تبدیل به فرم CNF استفاده نمود. عبارتی نظیر $\exists x\ x=x$ نشان می‌دهد که مدل غیرتهی است ولی عبارتی معتبر نمی‌باشد و ممکن است دنیا را بدون شی نشان دهد. در هر حال، تجربه نشان داده است که منطق‌آزاد نیازمند کار و قواعد بیشتر بر روی مدلهای تهی در موارد رایج برای نمایش و استنتاج منطقی می‌باشد.

**۸.۸ آیا حقیقت ¬*Spouse*(George,Laura) از عبارات Jim≠George و *Spouse*(Jim,George) می‌تواند نتیجه شود؟ اینطور است آن را اثبات کنید و در غیر اینصورت قانونی بیافزایید تا اثبات شود. چه اتفاقی می‌افتد اگر *Spouse* را بعنوان نماد تابع یگانی به جای مسند دوگانه تعریف کنیم؟**

حل: نمی‌توان حقیقت ¬*Spouse*(George,Laura) را از موارد گفته شده نتیجه گرفت. حداقل بایستی نشان دهیم که شخص می‌تواند همسر شخص دیگر باشد:

$$\forall x, y, z \;\; Spouse(x, z) \wedge Spouse(y, z) \;\Rightarrow\; x = y$$

ق این قانون، می‌توان عبارت ¬*Spouse*(George,Laura) را اثبات رزولوشن نمود. اگر *Spouse* یک تابع‌یگانی باشد، آنگاه ال آن است که آیا ¬*Spouse*(Laura)=George از عبارات Jim≠George و *Spouse*(Laura)=Jim نتیجه می‌شود خیر. پاسخ این سؤال مثبت است، و می‌توان این عبارت را نتیجه گرفت. زیرا در صورت متفاوت بودن دو شی یعنی Jim≠Geor، هر دوی آنها نمی‌توانند خروجی تابع *Spouse* به ازای یک شی مشخص باشند.

**۹.۸ این تمرین از تابع *MapColor*، مسندهای In(x,y),Border(x,y),Country(x) با آرگومانهای مناطق جغرافیایی، و دهای ثابتی برای مناطق مختلف استفاده می‌کند. در ادامه چند عبارت و تعدادی عبارت منطقی پیشنهادی برای هر یک از ا ذکر شده است. به ازای هر عبارت منطقی اعداد 1 تا 3 را قرار دهید بطوری که مفهوم اعداد بصورت زیر باشد (1): عبارت لقی به درستی، همان عبارت را نشان می‌دهد؛ (2): از نظر نحو نامعتبر است و بنابراین معنایی ندارد؛ (3): نحو آن صحیح است ی معنای عبارت را در بر ندارد.**

**) Paris و Marseills هر دو در France قرار دارند.**

**In(Paris∧Marseills,France)**

**In(Paris,France)∧In(Marseills,France)**

**In(Paris,France)∨In(Marseills.France) )**

**کشوری وجود دارد که در آن Iraq و Pakistan هم‌مرز باشند.**

**∃c Country(c)∧Border(c,Iraq)∧Border(c,Pakistan)**

**∃c Country(c) ⇒ [Border(c,Iraq)∧Border(c,Pakistan)]**

**[∃c Country(c)] ⇒ [Border(c,Iraq)∧Border(c,Pakistan)]**

**∃c Border(Country(c),Iraq∧Pakistan)**

**تمام کشورهای هم مرز با Ecuador در South America قرار دارند.**

**∀c Country(c)∧Border(c,Ecuador)⇒In(c,SouthAmerica)**

**∀c Country(c) ⇒[Border(c,Ecuador) ⇒In(c,SouthAmerica)**

**∀c [Country(c) ⇒Border(c,Ecuador)] ⇒In(c,SouthAmerica)**

**∀c Country(c) ∧ Border(c,Ecuador) ∧In(c,SouthAmerica)**

**هیچ منطقه‌ای از South America با هیچ منطقه‌ای از Europe هم‌مرز نیست.**

**¬[∃c,d In(c,SouthAmerica)∧In(d,Europe)∧Borders(c,d)**

**∀c,d [In(c,SouthAmerica)∧In(d,Europe)] ⇒¬[Borders(c,d)]**

**¬∀c In(c,SouthAmerica) ⇒∀d In(d,Europe) ∧ ¬Borders(c,d)**

**∀c In(c,SouthAmerica) ⇒∀d In(d,Europe) ⇒ ¬Borders(c,d)**

**هیچ دو کشورمجاوری، رنگ یکسان در نقشه ندارند.**

**∀x,y ¬Country(x)∨¬Country(y)∨¬Borders(x,y)∨¬(MapColor(x)=MapColor(y)**

**∀x,y Country(x) ∧Country(y) ∧Borders(x,y) ∧¬(x=y) ⇒¬(MapColor(x)=MapColor(y))**

**∀x,y Country(x) ∧Country(y) ∧Borders(x,y) ∧¬(MapColor(x)=MapColor(y))**

**∀x,y (Country(x) ∧Country(y) ∧Borders(x,y)) ⇒MapColor(x≠y)**

حل: الف) Paris و Marseills هر دو در France قرار دارند.

$$In(Paris \wedge Marseilles, France)$$

نحو این عبارت نامعتبر است. نمی‌توان از ترکیب عطفی در یک اصطلاح استفاده کرد.

$$In(Paris, France) \wedge In(Marseilles, France)$$

صحیح است.

$$In(Paris, France) \vee In(Marseilles, France)$$

نادرست است. ترکیب فصلی نمی‌تواند کلمهٔ «هردو» را دربربگیرد.

کشوری وجود دارد که در آن Iraq و Pakistan هم‌مرز باشند.

$$\exists c \;\; Country(c) \wedge Border(c, Iraq) \wedge Border(c, Pakistan)$$

نادرست

$$\exists c \;\; Country(c) \;\Rightarrow\; [Border(c, Iraq) \wedge Border(c, Pakistan)]$$

(3) نادرست. بعلت استفاده از ایجاب درون سور وجودی.

iii: $[\exists c \;\; Country(c)] \Rightarrow [Border(c, Iraq) \wedge Border(c, Pakistan)]$

(2) نحو آن نامعتبر است. متغیر c در خارج از محدودهٔ اعتبارش استفاده شده است.

iv: $\exists c \;\; Border(Country(c), Iraq \wedge Pakistan)$

(2) نحو آن نامعتبر است. نمی‌توان از ترکیب عطفی درون یک اصطلاح استفاده کرد.

**ج)** تمام کشورهای هم مرز با Ecuador در South America قرار دارند.

i: $\forall c \;\; Country(c) \wedge Border(c, Ecuador) \Rightarrow In(c, SouthAmerica)$

(1) درست

ii: $\forall c \;\; Country(c) \Rightarrow [Border(c, Ecuador) \Rightarrow In(c, SouthAmerica)]$

(1) درست. هم‌ارز عبارت (i) می‌باشد.

iii: $\forall c \;\; [Country(c) \Rightarrow Border(c, Ecuador)] \Rightarrow In(c, SouthAmerica)$

(3) نادرست. نماد ایجاب در سمت چپ همان ایجاب در سور وجودی است. که عبارت سمت راست را برای تمام غیرکشوری‌ها تایید می‌کند.

iv: $\forall c \;\; Country(c) \wedge Border(c, Ecuador) \wedge In(c, SouthAmerica)$

(3) نادرست. بعلت استفاده از ترکیب عطفی بعنوان اتصال اصلی یک سور عمومی.

**د)** هیچ منطقه‌ای از South America با هیچ منطقه‌ای از Europe هم‌مرز نیست.

i: $\neg[\exists c, d \;\; In(c, SouthAmerica) \wedge In(d, Europe) \wedge Borders(c, d)]$

(1) درست

ii: $\forall c, d \;\; [In(c, SouthAmerica) \wedge In(d, Europe)] \Rightarrow \neg Borders(c, d)]$

(1) درست

iii: $\neg\forall c \;\; In(c, SouthAmerica) \Rightarrow \exists d \;\; In(d, Europe) \wedge \neg Borders(c, d)$

(3) نادرست. این جمله می‌گوید کشورهایی در آمریکای جنوبی وجود دارند که با هر کشوری در اروپا مجاورند.

iv: $\forall c \;\; In(c, SouthAmerica) \Rightarrow \forall d \;\; In(d, Europe) \Rightarrow \neg Borders(c, d)$

(1) درست.

**هـ)** هیچ دو کشورمجاوری، رنگ یکسان در نقشه ندارند.

i: $\forall x, y \;\; \neg Country(x) \vee \neg Country(y) \vee \neg Borders(x, y) \vee \neg(MapColor(x) = MapColor(y))$

(1) درست

ii: $\forall x, y \;\; (Country(x) \wedge Country(y) \wedge Borders(x, y) \wedge \neg(x = y)) \Rightarrow \neg(MapColor(x) = MapColor(y))$

(1) درست. نامساوی نیاز نیست زیرا هیچ کشوری با خودش مرز ندارد.

iii: $\forall x, y \;\; Country(x) \wedge Country(y) \wedge Borders(x, y) \wedge \neg(MapColor(x) = MapColor(y))$

(3) نادرست. بعلت استفاده از ترکیب عطفی بعنوان اتصال اصلی سور عمومی.

iv: $\forall x, y \;\; (Country(x) \wedge Country(y) \wedge Borders(x, y)) \Rightarrow MapColor(x \neq y)$

(2) نحو آن نامعتبر است. نمی‌توان از نماد نامساوی درون یک اصطلاح استفاده کرد.

**؟ ۱۰.۸ فرهنگ واژگانی شامل این نمادها را در نظر بگیرید: Occupation(P,o): مسند، شخص P به کار O اشتغال دارد. Customer($P_1$,$P_2$): مسند، شخص$P_1$ یک مشتری شخص $P_2$ است. Boss($P_1$,$P_1$): مسند، شخص $P_1$ یک رئیس مشتری $P_2$است. Lawyer,Surgeon,Doctor: نمادهای ثابتی جهت اشتغالهای مختلف. Emily یا سایر نام‌ها: نمادهای ثابت برای معرفی افراد.**

**طبق نمادهای فوق، این جملات را به منطق مرتبهٔ اول بنویسید: الف) امیلی یا جراح و یا وکیل است. ب) جو یک بازیگر است ولی در کنار آن شغل دیگری نیز دارد. ج) تمام جراحان، دکتر هستند. هـ) امیلی رئیسی دارد که او یک وکیل است. و) وکیلی وجود دارد که تمام مشتریانش، دکتر هستند. ز) هر جراح یک وکیل دارد.**

☑ حل: **الف)** $O(E, S) \vee O(E, L)$ **ب)** $O(J, A) \wedge \exists p \;\; p \neq A \wedge O(J, p)$

**ج)** $\forall p \;\; O(p, S) \Rightarrow O(p, D)$ **د)** $\neg\exists p \;\; C(J, p) \wedge O(p, L)$

**هـ)** $\exists p \;\; B(p, E) \wedge O(p, L)$ **و)** $\exists p \;\; O(p, L) \wedge \forall q \;\; C(q, p) \Rightarrow O(q, D)$

**ز)** $\forall p \;\; O(p, S) \Rightarrow \exists q \;\; O(q, L) \wedge C(p, q)$

**؟ ۱۱.۸ طبق جملات منطقی، موارد زیر را تکمیل کنید.**

**الف) عبارت زیر را با جمله‌ای روان تعبیر کنید. (نه با نمادهای ریاضی و $x_s$ و $y_s$)**

$\forall x,y,l$ SpeaksLanguage$(x,l) \wedge$ SpeaksLanguage$(y,l) \Rightarrow$ Understands$(x,y) \wedge$ Understands$(y,x)$
توضیح دهید که چرا این جمله توسط جملهٔ زیر ایجاب می‌شود:
$\forall x,y,l$ SpeaksLanguage$(x,l) \wedge$ SpeaksLanguage$(y,l) \Rightarrow$ Understands$(x,y)$
جملات زیر را به منطق مرتبهٔ اول ترجمه کنید:
درک منجر به دوستی می‌شود. **ii)** دوستی دوطرفه است.
خاطر داشته باشید که بایستی تمام مسندها، توابع و نمادهای ثابت استفاده شده را تعریف کنید.
حل: **الف)** مردمی که به یک زبان سخن می‌گویند، همدیگر را درک می‌کنند. **ب)** فرض کنید $x \rightarrow$A و $y \rightarrow$B که برای
ی $l$ها صدق می‌کند: $SpeaksLanguage(x,l) \wedge SpeaksLanguage(y,l)$

ق جملهٔ دوم می‌توان نتیجه گرفت که $x \rightarrow$B نمایش $Understands$(A,B) و $y \rightarrow$A باید ارضا کند:
$SpeaksLanguage(x,l) \wedge SpeaksLanguage(y,l)$

به ما اجازه می‌دهد تا نتیجه بگیریم $Understands$(A,B). بنابراین هرگاه که جملهٔ دوم برقرار باشد جملهٔ اول نیز برقرار
ت. **ج)** اگر $Understands(x,y)$ به معنای آن باشد که $x$ $y$ را درک می‌کند و $Friends(x,y)$ به معنای آن باشد که $x$
دوست $y$ است. **i)** نمی‌توان بطور کامل از جمله برداشت کرد که منظور آن درک متقابل و دوستی دوطرفه باشد ولی فرض
ید که اینچنین باشد.
$\forall x,y \;\; Understands(x,y) \wedge Understands(y,x) \Rightarrow (Friend(x,y) \wedge Friend(y,x))$

$\forall x,y,z \;\; Friend(x,y) \wedge Friend(y,z) \;\Rightarrow\; Friend(x,z)$

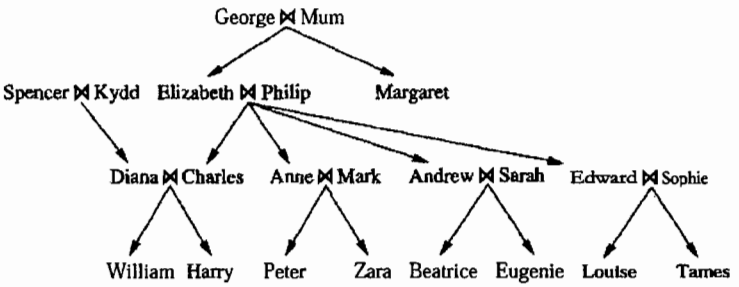
**۱۲.۸ دو قاعدهٔ اول Peano در بخش ۳.۳.۸ را بعنوان یک قاعدهٔ واحد بازنویسی کنید که در آن از NatNum(x) بعنوان**
**ادطبیعی ممکن به جز موارد تولید شده توسط تابع پسین، استفاده شود.**
حل: این تمرین نیازمند بازنویسی مشابه Clark از دو بند هورن می‌باشد:

$\forall n \;\; NatNum(n) \;\Leftrightarrow\; [n = 0 \vee \exists m \;\; NatNum(m) \wedge n = S(m)]$

**۱۳.۸ معادلهٔ ۴.۸ شرایطی را جهت وجود نسیم در یک خانه تعریف می‌کند. در اینجا دو روش دیگر برای معرفی این ویژگی از**
**ای ومپوز را در نظر می‌گیریم: الف) می‌توان از «قواعد شناختی» استفاده کرد تا به کمک موارد مشاهده شده بتوان موارد**
**ان را تشخیص داد. برای یافتن چاله‌ها، قواعد شناختی می‌گوید اگر یک خانه دارای نسیم است برخی همسایگان مجاورش**
**ه دارند و اگر یک خانه دارای نسیم نیست، هیچ همسایهٔ مجاورش چاله ندارد. این دو قانون را در منطق مرتبهٔ اول نوشته و**
**ن دهید که ترکیب عطفی آن دو بطور منطقی با معادلهٔ ۴.۸ هم ارز است. ب) می‌توان از «قواعد سببی» استفاده کرد که در**
**از روابط علت و معلول استفاده می‌شود. یک قاعدهٔ سببی در دنیای ومپوز می‌تواندآن باشد که یک چاله سبب می‌شود تمام**
**ه‌های مجاورش دارای نسیم شوند. این قاعده را به منطق مرتبهٔ اول نوشته و توضیح دهید که چرا در مقایسه با معادلهٔ ۴.۸**
**مل است، قواعد لازم برای تکمیل آن را ذکر کنید.**
حل: **الف)** دو عبارت ایجاب عبارتند از:
$\forall s \;\; Breezy(s) \;\Rightarrow\; \exists r \;\; Adjacent(r,s) \wedge Pit(r)$
$\forall s \;\; \neg Breezy(s) \;\Rightarrow\; \neg\exists r \;\; Adjacent(r,s) \wedge Pit(r)$

$\forall s \;\; \exists r \;\; Adjacent(r,s) \wedge Pit(r) \;\Rightarrow\; Breezy(s)$
وس عبارت دوم عبارتست از:

$\forall s \;\; Breezy(s) \;\Leftrightarrow\; \exists r \;\; Adjacent(r,s) \wedge Pit(r)$
درترکیب با عبارت اول داریم:
برای نشان دادن اینکه هر چاله موجب ایجاد نسیم در تمام خانه‌های مجاور می‌شود، داریم:
$\forall s \;\; Pit(s) \;\Rightarrow\; [\forall r \;\; Adjacent(r,s) \;\Rightarrow\; Breezy(r)]$
قانون اجازه می‌دهد تا در یک خانه نسیم باشد ولی در خانهٔ مجاور آن چاله نباشد زیرا می‌دانیم که جملهٔ «خانهٔ بدون چاله
ث می‌شود تمام خانه‌های مجاور بدون نسیم باشند» عبارتی غلط است زیرا ممکن است وجود آن نسیم به خاطر وجود چاله‌ای
یک خانهٔ دیگر باشد. ولی اگر تمام خانه‌های مجاور بدون چاله باشند آنگاه حتما خانه نیز بدون نسیم خواهد بود:
$\forall s \;\; [\forall r \;\; Adjacent(r,s) \;\Rightarrow\; \neg Pit(r)] \;\Rightarrow\; \neg Breezy(s)$
**۱۴.۸ اصولی بنویسید که مسندهای زیر را توصیف کند:**
**BrotherInLaw،FirstCousin،Son،Daughter،Sister،Brother،Ancestor،GreatGrandparent،Grandch**
**Uncle،Aunt،SisterInL. تعریف مناسبی برای mامین پسرخاله (دخترعمو یا پسردایی یا پسرعمه و....) بیابید که n بار**
**گزین شده است و آن تعریف را به منطق مرتبهٔ اول بازگو کنید. سپس حقایقی در مورد شجره‌نامهٔ موجود در شکل ۷.۸**
**سید. با استفاده از سیستم استدلال منطقی مناسب، تمام جملاتی که باید نوشته شود را به آن Tell نموده و سپس از آن**
**A کنید که نوهٔ Elizabeth، برادرخوانده Diana، جد بزرگ Zara و اجداد Eugenie کیست؟** ***(این تمرین مشابه تمرین***
***۱ در ویرایش دوم است)***




شکل ۷.۸ یک درخت خانواده رایج. نماد ⋈ نشان‌دهنده وصلت دو نفر و بردارها نشان‌دهنده فرزندان ماحصل از آن وصلت می‌باشد.

☑ حل: مطمئن شوید که در نوشتار خود از $\Leftrightarrow$ استفاده شده باشد. اگر شما از $\Rightarrow$ استفاده کنید، فقط قیودی را تحمیل کرده‌اید و تعریفی حقیقی ارائه نداده‌اید.

$$Grandchild(c,a) \Leftrightarrow \exists b\ Child(c,b) \wedge Child(b,a)$$
$$Greatgrandparent(a,d) \Leftrightarrow \exists b,c\ Child(d,c) \wedge Child(c,b) \wedge Child(b,a)$$
$$Ancestor(a,x) \Leftrightarrow Child(x,a) \vee \exists b\ Child(b,a) \wedge Ancestor(b,x)$$
$$Brother(x,y) \Leftrightarrow Male(x) \wedge Sibling(x,y)$$
$$Sister(x,y) \Leftrightarrow Female(x) \wedge Sibling(x,y)$$
$$Daughter(d,p) \Leftrightarrow Female(d) \wedge Child(d,p)$$
$$Son(s,p) \Leftrightarrow Male(s) \wedge Child(s,p)$$
$$FirstCousin(c,d) \Leftrightarrow \exists p_1,p_2\ Child(c,p_1) \wedge Child(d,p_2) \wedge Sibling(p_1,p_2)$$
$$BrotherInLaw(b,x) \Leftrightarrow \exists m\ Spouse(x,m) \wedge Brother(b,m)$$
$$SisterInLaw(s,x) \Leftrightarrow \exists m\ Spouse(x,m) \wedge Sister(s,m)$$
$$Aunt(a,c) \Leftrightarrow \exists p\ Child(c,p) \wedge [Sister(a,p) \vee SisterInLaw(a,p)]$$
$$Uncle(u,c) \Leftrightarrow \exists p\ Child(c,p) \wedge [Brother(a,p) \vee BrotherInLaw(a,p)]$$

روشهای مختلفی برای حذف $m$امین پسرخاله در $n$ بار، وجود دارد.یکی از این روشها، آن است که فاصله هر شخص با نزدیکترین جدش را بیابیم. عبارت $Distance(c,a)$ عبارتست از:

$$Distance(c,c) = 0$$
$$Child(c,b) \wedge Distance(b,a) = k \Rightarrow Distance(c,a) = k+1$$

بنابراین فاصله هر فرد با پدربزرگش 2 و فاصله هر فرد با پدرپدربزرگ 4 می‌باشد وغیره. بنابراین داریم:

$$MthCousinNTimesRemoved(c,d,m,n) \Leftrightarrow$$
$$\exists a\ Distance(c,a) = m+1 \wedge Distance(d,a) = m+n+1$$

حقایق موجود در شجره‌نامه بسیار ساده هستند:
هر بردار نشان‌دهندهٔ یک $Child$ است.(به عنوان مثال $Child(William,Diana)$ و $Child(William,Charles)$). هر نام نشان‌دهندهٔ یک گزارهٔ جنسی است (به عنوان مثال $Male(William)$ یا $Female(Diana)$. هر خط دوتایی نشان‌دهندهٔ یک گزارهٔ ازدواج یا $Spouse$ (به عنوان مثال $Spouse(Charles,Diana)$). پرس‌وجو از سیستم استدلال منطقی یکی از راه‌های اشکال‌زدایی تعاریف است.

**؟ ۱۵.۸ تعریف زیر برای مسند عضویت مجموعه، $\in$ ، پیشنهاد شده است. توضیح دهید که چه مشکلی دارد:**

$$\forall x,y\ \ x \in \{x|s\}$$
$$\forall x,s\ \ x \in s \Rightarrow \forall\ y\ \ x \in \{y|s\}$$

***(این تمرین مشابه تمرین ۱۳.۸ در ویرایش دوم است)***

☑ حل: اگر چه این اصول برای اثبات زیرمجموعه‌بودن در زمانی که $x$ واقعا یک عضو از مجموعه باشد، کافی است ولی در مورد مواقعی که $x$ یک عضو نباشد هیچ چیزی نمی‌گوید. به عنوان مثال، نمی‌توان اثبات کرد که $x$ عضوی از مجموعهٔ تهی نیست. این اصول ممکن است برای یک سیستم منطقی نظیر پرولوگ مناسب باشد که از تناقضات به عنوان شکست استفاده می‌کند.

**؟ ۱۶.۸ با استفاده از اصول‌مجموعه به عنوان مثال، اصولی برای دامنهٔ فهرست شامل تمام ثابت‌ها، توابع و مسندهایی بنویسید که در این فصل بیان شد. *(این تمرین مشابه تمرین ۱۴.۸ در ویرایش دوم است)***

☑ حل: منظور از $List?$ به معنای فهرست مناسب در اصطلاحات $Lisp$ است. به عنوان مثال ساختاری که در سمت راست‌ترین رقم خود $Nil$ داشته باشد.

$$List?(Nil)$$
$$\forall x,l\ List?(l) \Leftrightarrow List?(Cons(x,l))$$
$$\forall x,y\ First(Cons(x,y)) = x$$
$$\forall x,y\ Rest(Cons(x,y)) = y$$
$$\forall x\ Append(Nil,x) = x$$
$$\forall v,x,y,z\ List?(x) \Rightarrow (Append(x,y) = z \Leftrightarrow Append(Cons(v,x),y) = Cons(v,z))$$
$$\forall x\ \neg Find(x,Nil)$$
$$\forall x\ List?(z) \Rightarrow (Find(x,Cons(y,z)) \Leftrightarrow (x = y \vee Find(x,z))$$

**؟ ۱۷.۸ تعریف زیر برای خانه‌های مجاور در دنیای ومپوز، پیشنهاد شده است. توضیح دهید که چه مشکلی دارد:**

$$\forall x,y\ \text{Adjacent}([x,y],[x+1,y]) \wedge \text{Adjacent}([x,y],[x,y+1])$$

***(این تمرین مشابه تمرین ۱۵.۸ در ویرایش دوم است)***

☑ حل: این تعریف پیشنهادی چندین مشکل دارد. یکی از آنها این است که طبق این تعریف، عبارت:

$$\text{Adjacent}([1,1],[1,2])$$

این دو خانه را مجاور معرفی می‌کند در حالیکه $\text{Adjacent}([1,2],[1,1])$ طبق این تعریف برقرار نیست و مجاور نیستند در حالیکه در دنیای ومپوز مجاورند. لذا بایستی یک اصل برای تقارن عبارات اضافه شود. همچنین نمی‌توان با این تعریف، نادرستی عبارت $\text{Adjacent}([1,1],[1,3])$ را اثبات نمود. بنابراین بایستی آن را به صورت زیر نوشت:

$\forall\ s_1,s_2 \Leftrightarrow \ldots$

در نهایت، این تعریف تمام محدودیت‌های دنیا را شامل نمی‌شود و بایستی تعدادی شرایط به آن اضافه شود.

**؟ ۱۸.۸ با استفاده از نماد ثابت *Wumpus* و یک مسند دودویی *In(Wumpus,Location)* اصولی جهت استدلال در مورد مکان ومپوز بنویسید. به خاطر داشته باشید که فقط یک ومپوز در دنیا وجود دارد. *(این تمرین مشابه تمرین ۱۶.۸ در ویرایش دوم است)***

☑ حل: به این جملات نیاز داریم:

$$\forall s_1\ Smelly(s_1) \Leftrightarrow \exists s_2\ Adjacent(s_1,s_2) \wedge In(Wumpus,s_2)$$
$$\exists s_1\ In(Wumpus,s_1) \wedge \forall s_2\ (s_1 \neq s_2) \Rightarrow \neg In(Wumpus,s_2)$$

**؟ ۱۹.۸ مسندهای *Parent(p,q)* و *Female(p)* و نمادهای ثابت Joan وKevin را درنظربگیرید که معنای هر یک با توجه به کلمهٔ آن مشخص است. هر یک از جملات زیر را به منطق مرتبهٔ اول بیان کنید (می‌توانید از نمادهای خلاصه برای بیان «فقط یکی وجود دارد» استفاده کنید). الف) Joan یک دختر دارد (ممکن است بیش از یک دختر و یا چندین پسر هم داشته باشد) ب) Joan فقط یک دختر دارد (ولی ممکن است چندین پسر داشته باشد) ج) Joan فقط یک بچه یعنی یک دختر دارد. د) Joan,Kevin فقط یک بچه مشترک دارند. هـ) Joan حداقل یک بچه مشترک با Kevin دارد و هیچ بچه‌ای از فرد دیگر ندارد.**

☑ حل: **الف)** $\exists x\ Parent(Joan,x) \wedge Female(x)$ **ب)** $\exists^1 x\ Parent(Joan,x) \wedge Female(x)$

**ج)** $\exists x\ Parent(Joan,x) \wedge Female(x) \wedge [\forall y\ Parent(Joan,y) \Rightarrow y = x]$

گاهی مواقع آن را به صورت $Female(i(x)\ Parent(\text{Joan},x))$ نیز خلاصه می‌کنند.

**د)** $\exists^1 c\ Parent(Joan,c) \wedge Parent(Kevin,c)$

**هـ)** $\exists c\ Parent(Joan,c) \wedge Parent(Kevin,c) \wedge \forall d,p\ [Parent(Joan,d) \wedge Parent(p,d)] \Rightarrow [p = Joan \vee p = Kevin]$

**؟ ۲۰.۸ اصول حساب را می‌توان توسط نماد مسند $<$، نمادهای تابع $+$ ، $x$ و نمادهای ثابت 0 و 1 در منطق مرتبهٔ اول تعریف نمود. سایر مسندها می‌توانند بصورت دوشرطی تعریف و اضافه شوند. الف) عبارت «$x$ یک عدد زوج است» را بازنمایی کنید. ب) عبارت «$x$ عددی اول است» را بازنمایی کنید. ج) فرضیهٔ Goldbach (که هنوز اثبات نشده است) می‌گوید هر عدد زوج با مجموع دو عدد اول برابر است. این فرضیه را بعنوان جمله‌ای منطقی نشان دهید.**

☑ حل: **الف)** $\forall x\ Even(x) \Leftrightarrow \exists y\ x = y + y$

**ب)** $\forall x\ Prime(x) \Leftrightarrow \forall y,z\ x = y \times z \Rightarrow y = 1 \vee z = 1$

**ج)** $\forall x\ Even(x) \Rightarrow \exists y,z\ Prime(y) \wedge Prime(z) \wedge x = y + z$

**؟ ۲۱.۸ در فصل ۶، برای بیان رابطهٔ بین متغیر و مقدار آن از تساوی استفاده کردیم. بعنوان مثال WA=red به معنای آن است که شهر Western Australia به رنگ قرمز رنگ‌آمیزی شده است. این شیوهٔ بازنمایی در منطق مرتبهٔ‌اول نیازمند نوشتن تعداد زیادی جملهٔ نظیر ColorOf(WA)=red خواهد بود. چه استنتاج اشتباهی رخ می‌دهد، اگر جمله‌ای نظیر WA=red مستقیما به صورت یک قانون منطقی نوشته شود؟**

☑ حل: اگر داشته باشیم WA=red و $Q$=red آنگاه می‌توان نتیجه گرفت که WA=$Q$ که در نقشهٔ استرالیا نمی‌توان شهرهای Western Australia و Queenlander را همیشه همرنگ دانست.

**؟ ۲۲.۸ ادعای «هر کلید و حداقل یکی از لنگه جورابها سرانجام برای همیشه گم می‌شوند» را در منطق مرتبهٔ اول بیان کنید. برای این کار از این واژگان استفاده کنید: $Key(x)$: $x$ یک کلید است. $Sock(s)$: $x$ یک جوراب است. $Pair(x,y)$: $x,y$ یک جفت هستند. $Befor(t_1,t_2)$: زمان $t_1$ قبل از زمان $t_2$ است. $Lost(x,t)$: شی $x$ در زمان $t$ گم شده است.**

☑ حل:

$$\forall k \;\; Key(k) \Rightarrow [\exists t_0 \;\; Before(Now, t_0) \wedge \forall t \;\; Before(t_0, t) \Rightarrow Lost(k, t)]$$
$$\forall s_1, s_2 \;\; Sock(s_1) \wedge Sock(s_2) \wedge Pair(s_1, s_2) \Rightarrow$$
$$[\exists t_1 \;\; Before(Now, t_1) \wedge \forall t \;\; Before(t_1, t) \Rightarrow Lost(s_1, t)] \vee$$
$$[\exists t_2 \;\; Before(Now, t_2) \wedge \forall t \;\; Before(t_2, t) \Rightarrow Lost(s_2, t)]$$

دقت کنید که ترکیب فصلی اجازه می‌دهد که هر دو جوراب گم شوند، همانطور که از جمله برداشت می‌شود.

**؟ ۲۳.۸ برای هر یک از جملات زیر، کدام جمله به خوبی به منطق مرتبهٔ اول ترجمه شده و کدامیک خیر. اگر ترجمه درست نیست توضیح دهید که چرا اینگونه بوده و آن را اصلاح کنید. (برخی عبارات ممکن است بیش از یک خطا داشته باشند) الف) هیچ دو نفری، شماره امنیت اجتماعی یکسان ندارند. $\neg\exists x,y,n \; Person(x)\wedge Person(y)\Rightarrow[HasSS\#(x,n)\wedge HasSS\#(y,n)]$**

**ب) شمارهٔ امنیت اجتماعی جان، همان شمارهٔ مساوی است. $\exists n \; HasSS\#(John,n) \wedge HasSS\#(Mary,n)]$**

**ج) شمارهٔ امنیت اجتماعی هر فردی، 9 رقمی است. $\forall x,n \; Person(x)\Rightarrow[HasSS\#(x,n)\wedge Digits(n,9)]$**

**د) در هر یک از عبارات نادرست فوق، به جای استفاده از مسند $HasSS\#$ از نماد تابعی $SS\#$ استفاده کرده و آنها را بازنویسی کنید.**

☑ حل: **الف)** «هیچ دو نفری، شماره امنیت اجتماعی یکسان ندارند»

$$\neg\exists x, y, n \;\; Person(x) \wedge Person(y) \Rightarrow [HasSS\#(x, n) \wedge HasSS\#(y, n)]$$

این عبارت از $\Rightarrow$ و $\exists$ استفاده می‌کند. این عبارت می‌گوید که هیچ‌کس شمارهٔ امنیت اجتماعی ندارد زیرا خود را به مواردی که $x$ و $y$ نابرابر باشند محدود نمی‌کند. نسخهٔ صحیح این جمله عبارتست از:

$$\neg\exists x, y, n \;\; Person(x) \wedge Person(y) \wedge \neg(x = y) \wedge [HasSS\#(x, n) \wedge HasSS\#(y, n)]$$

**ب)** «شمارهٔ امنیت اجتماعی جان همان شمارهٔ ماری است»

$$\exists n \;\; HasSS\#(John, n) \wedge HasSS\#(Mary, n)$$

این عبارت صحیح است.

**ج)** «شمارهٔ امنیت اجتماعی هر فردی 9 رقمی است» $\forall x, n \;\; Person(x) \Rightarrow [HasSS\#(x, n) \wedge Digits(n, 9)]$

این عبارت می‌گوید هر کسی می‌تواند هر عددی داشته باشد. $HasSS\#(x,n)$ باید بصورت زیر فرض شود:

$$\forall x, n \;\; Person(x) \wedge HasSS\#(x, n) \Rightarrow Digits(n, 9)$$

**د)** در اینجا منظور از $SS\#(x)$ همان شمارهٔ امنیت اجتماعی $x$ می‌باشد. با استفاده از یک تابع، قاعده را مجبور می‌کنیم که هر فردی فقط یک شماره داشته باشد. $\neg\exists x, y \;\; Person(x) \wedge Person(y) \Rightarrow [SS\#(x) = SS\#(y)]$

$$SS\#(John) = SS\#(Mary)$$
$$\forall x \;\; Person(x) \Rightarrow Digits(SS\#(x), 9)$$

**؟ ۲۴.۸ این جملات را با استفاده از واژگان سازگار (طبق تعریف شما) به منطق مرتبهٔ اول بازنمایی کنید: الف) برخی دانشجویان درس فرانسه را در بهار 2001 اخذ کردند. ب) هر دانشجویی که درس فرانسه را اخذ کند، آن را می‌گذراند. ج) فقط یک دانشجو درس یونانی را در بهار 2001 اخذ کرد. د) بهترین نمرهٔ درس یونانی همیشه بالاتر از بهترین نمرهٔ درس فرانسه است. هـ) هرکسی که یک بیمه‌نامه می‌خرد، باهوش است. و) هیچ‌کسی بیمه‌نامهٔ گران نمی‌خرد. ز) کارگزاری وجود دارد که فقط به کسانی که بیمه نیستند، بیمه‌نامه می‌فروشد. ح) آرایشگری وجود دارد که تمام مردان شهر که خودشان را اصلاح نمی‌کنند، اصلاح می‌کند. ط) شخصی که متولد انگلستان باشد و یکی از والدینش یا شهروند انگلستان و یا مقیم در انگلستان باشد، به واسطهٔ زادگاهش شهروند انگلستان محسوب می‌شود. ی) شخصی که خارج از انگلستان متولد شود و یکی از والدینش به واسطهٔ زادگاه شهروند انگلستانی باشد، آنگاه به واسطهٔ نژادش شهروند انگلستان محسوب می‌شود. ک) سیاستمداران می‌توانند همیشه، برخی مردم را فریب دهند و در برخی اوقات، همهٔ مردم را ولی نمی‌توانند همهٔ مردم را همیشه فریب دهند. ل) تمام یونانی‌ها به زبان یکسانی صحبت می‌کنند. *(این تمرین مشابه تمرین ۶.۸ در ویرایش دوم است)***

☑ حل: در این تمرین بهتر است در مورد جزئیات زمان افعال و سازگار بودن مقایسه‌ها و غیره نگران نباشیم. نکتهٔ اصلی آن است که مطمئن شویم کلمات ربط، کمیت‌سنج و استفاده از مسند، تابع، ثوابت و هم‌ارزی را می‌دانیم. فرهنگ واژگان زیر را در نظر بگیرید:

$Takes(x,c,s)$ : دانشجوی $x$ درس $c$ را در نیمسال $s$ اخذ می‌کند.

$Passes(x,c,s)$ : دانشجوی $x$ درس $c$ را در نیمسال $s$ می‌گذراند.

$Score(x,c,s)$ : نمره کسب‌شده توسط دانشجوی $x$ در درس $c$ در نیمسال $s$ را نشان می‌دهد.

$x>y$ : $x$ بزرگتر از $y$ است.

$G$ و $F$: دروس فرانسه و یونانی را نشان می‌دهد (هر یک از آنها می‌تواند به عنوان هر درس در جملات تفسیر شود در آنصورت
ک مسند بنام $Subject(c,f)$ برای نشان دادن اینکه درس $c$ در رشتهٔ $f$ است، استفاده می‌شود).
$Buys(x,y,z)$: $x$ $y$ را از $z$ می‌خرد.
$Sells(x,y,z)$: $x$ $y$ را به $z$ می‌فروشند.
$Shaves(x,y)$: شخص $x$ شخص $y$ را اصلاح می‌کند.
$Born(x,c)$: شخص $x$ در کشور $c$ متولد می‌شود.
$Parent(x,y)$: $x$ یکی از والدین $y$ است.
$Citizen(x,y,r)$: $x$ یک شهروند کشور $y$ به دلیل $r$ می‌باشد.
$Resident(x,c)$: $x$ مقیم کشور $c$ است.
$Fools(x,y,t)$: شخص $x$ شخص $y$ را در زمان $t$ می‌فریبد.
$Politician(x), Smart(x), Insured(x), Agent(x), Expensive(x), Barber(x), Man(x), Person(x), Student(x)$:
سندهایی هستند که برای اعضای دسته‌های مختلف استفاده می‌شود.

**ف)** برخی دانشجویان درس فرانسه را در بهار 2001 اخذ کردند.

$$\exists x \ \ Student(x) \wedge Takes(x, F, Spring2001)$$

**ب)** هر دانشجویی که درس فرانسه را اخذ کند، آن را می‌گذراند.

$$\forall x, s \ \ Student(x) \wedge Takes(x, F, s) \Rightarrow Passes(x, F, s)$$

**ج)** فقط یک دانشجو درس یونانی را در بهار 2001 اخذ کرد.

$$\exists x \ \ Student(x) \wedge Takes(x, G, Spring2001) \wedge \forall y \ \ y \neq x \Rightarrow \neg Takes(y, G, Spring2001)$$

**د)** بهترین نمرهٔ درس یونانی همیشه بالاتر از بهترین نمرهٔ درس فرانسه است.

$$\forall s \ \exists x \ \forall y \ \ Score(x, G, s) > Score(y, F, s)$$

**ـه)** هرکسی که یک بیمه‌نامه می‌خرد، باهوش است.

$$\forall x \ \ Person(x) \wedge (\exists y, z \ \ Policy(y) \wedge Buys(x, y, z)) \Rightarrow Smart(x)$$

**و)** هیچ‌کسی بیمه‌نامه گران نمی‌خرد.

$$\forall x, y, z \ \ Person(x) \wedge Policy(y) \wedge Expensive(y) \Rightarrow \neg Buys(x, y, z)$$

**ز)** کارگزاری وجود دارد که فقط به کسانی بیمه‌نامه می‌فروشد که بیمه نباشند.

$$\exists x \ \ Agent(x) \wedge \forall y, z \ \ Policy(y) \wedge Sells(x, y, z) \Rightarrow (Person(z) \wedge \neg Insured(z))$$

**ح)** آرایشگری وجود دارد که تمام مردان شهر که خودشان را اصلاح نمی‌کنند، اصلاح می‌کند.

$$\exists x \ \ Barber(x) \wedge \forall y \ \ Man(y) \wedge \neg Shaves(y, y) \Rightarrow Shaves(x, y)$$

**ط)** شخصی که متولد انگلستان باشد و یکی از والدینش یا شهروند انگلستان و یا مقیم در انگلستان باشد، به واسطهٔ زادگاهش
هروند انگلستان محسوب می‌شود.

$$\forall x \ \ Person(x) \wedge Born(x, UK) \wedge (\forall y \ \ Parent(y, x) \Rightarrow ((\exists r \ \ Citizen(y, UK, r)) \vee Resident(y, UK))) \Rightarrow Citizen(x, UK, Birth)$$

**ی)** شخصی که خارج از انگلستان متولد شود و یکی از والدینش به واسطهٔ زادگاه شهروند انگلستانی باشد آنگاه به واسطهٔ نژادش
هروند انگلستان محسوب می‌شود.

$$\forall x \ \ Person(x) \wedge \neg Born(x, UK) \wedge (\exists y \ \ Parent(y, x) \wedge Citizen(y, UK, Birth)) \Rightarrow Citizen(x, UK, Descent).$$

**ک)** سیاستمداران می‌توانند همیشه برخی از مردم را فریب دهند و در برخی اوقات همهٔ مردم را. ولی نمی‌توانند همهٔ مردم را
میشه فریب دهند.

$$\forall x \ \ Politician(x) \Rightarrow (\exists y \ \forall t \ \ Person(y) \wedge Fools(x, y, t)) \wedge (\exists t \ \forall y \ \ Person(y) \Rightarrow Fools(x, y, t)) \wedge \neg(\forall t \ \forall y \ \ Person(y) \Rightarrow Fools(x, y, t))$$

**ل)** تمام یونانی‌ها به زبان یکسانی صحبت می‌کنند:

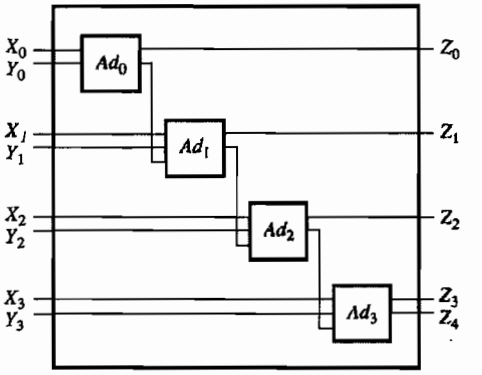
$$\forall x, y, l \ \ Person(x) \wedge [\exists r \ \ Citizen(x, Greece, r)] \wedge Person(y) \wedge [\exists r \ \ Citizen(y, Greece, r)] \wedge Speaks(x, l) \Rightarrow Speaks(y, l)$$

**۲۵.۸ مجموعه‌ای کلی از حقایق و قوانین بنویسید که ادعای «ولینگتون خبر مرگ ناپلئون را شنید» را بازنمایی کند و پاسخ
حیح سؤال «آیا ناپلئون خبر مرگ ولینگتون را شنید؟» را بیابید.** ***(این تمرین مشابه تمرین ۹.۸ در ویرایش دوم است)***

حل: این تمرین بسیار جنبهٔ آموزشی دارد ولی خیلی بدیهی نیست. مسندهای اصلی عبارتند از: $Heard(x,e,t)$ (x رخداد
را در زمان $t$ شنید)، $Occurred(e,t)$ (رخداد $e$ در زمان $t$ اتفاق افتاد)، $Alive(x,t)$ ($x$ در زمان $t$ زنده است):

$\exists t\ \ Heard(W, DeathOf(N), t)$
$\forall x, e, t\ \ Heard(x, e, t) \Rightarrow Alive(x, t)$
$\forall x, e, t_2\ \ Heard(x, e, t_2) \Rightarrow \exists t_1\ \ Occurred(e, t_1) \wedge t_1 < t_2$
$\forall t_1\ \ Occurred(DeathOf(x), t_1) \Rightarrow \forall t_2\ \ t_1 < t_2 \Rightarrow \neg Alive(x, t_2)$
$\forall t_1, t_2\ \ \neg(t_2 < t_1) \Rightarrow ((t_1 < t_2) \vee (t_1 = t_2))$
$\forall t_1, t_2, t_3\ \ (t_1 < t_2) \wedge ((t_2 < t_3) \vee (t_2 = t_3)) \Rightarrow (t_1 < t_3)$
$\forall t_1, t_2, t_3\ \ ((t_1 < t_2) \vee (t_1 = t_2)) \wedge (t_2 < t_3) \Rightarrow (t_1 < t_3)$

**؟ ۲۶.۸ فرهنگ واژگان بخش ۴.۸ را جهت تعریف جمع‌کننده $n$- بیتی اعداد دودویی توسعه دهید. سپس جمع‌کننده 4- بیتی موجود در شکل ۸.۸ را رمزنگاری کرده و با پرسیدن سئوالات موردنیاز از آن، نشان دهید که درعمل بدرستی کار می‌کند.** ***(این تمرین مشابه تمرین ۱۷.۸ در ویرایش دوم است)***




$$\begin{array}{cccccc} & X_3 & X_2 & X_1 & X_0 \\ + & Y_3 & Y_2 & Y_1 & Y_0 \\ \hline Z_4 & Z_3 & Z_2 & Z_1 & Z_0 \end{array}$$

شکل ۸.۸ یک جمع‌کننده چهار بیتی. هر $Ad$ یک جمع‌کننده یک بیتی می‌باشد.

☑ حل: به سه مرحله نیاز است. در مرحلهٔ اول، جمع‌کنندهٔ 1- بیتی و $n$- بیتی را تعریف می‌کنیم سپس مدارهای یک بیتی و $n$ بیتی را مشخص می‌سازیم. در نهایت اثبات می‌کنیم که مدار جمع‌کنندهٔ $n$- بیتی، عملیات جمع $n$ بیت را انجام می‌دهد. • جمع‌کنندهٔ یک بیتی بسیار ساده است. فرض کنید $Add_1$ تابعی با سه آرگومان یک بیتی باشد که دو آرگومان اولیه همان اعدادی هستند که قصد جمع کردن آنها را داریم و سومین آرگومان، بیت نقلی است. نتیجهٔ حاصلجمع، لیستی از بیت‌هاست که یک عدد 2 بیتی دودویی را نشان می‌دهد. کم‌ارزش‌ترین رقم در ابتدا است.

$Add_1(0,0,0) = [0,0]$
$Add_1(0,0,1) = [0,1]$
$Add_1(0,1,0) = [0,1]$
$Add_1(0,1,1) = [1,0]$
$Add_1(1,0,0) = [0,1]$
$Add_1(1,0,1) = [1,0]$
$Add_1(1,1,0) = [1,0]$
$Add_1(1,1,1) = [1,1]$

• جمع‌کنندهٔ $n$- بیتی درون خود ساختار جمع‌کنندهٔ 1- بیتی را دارد. فرض کنید $add_n(x_1,x_2,b)$ تابعی باشد که دو عدد باینری به طول $n$ (کم‌ارزش‌ترین رقم در ابتدا قرار دارد) و یک بیت نقلی (در ابتدا صفر لحاظ می‌شود) را گرفته و عددی باینری به طول $n+1$ که نشان‌دهندهٔ مجموع آن دو رقم است را برمی‌گرداند. (این مقدار همواره به طول $n+1$ بیت خواهد بود حتی زمانی که بیت مقدم صفرباشد. منظور از بیت مقدم، همان بیت سرریز است):

$Add_n([],[],b) = [b]$
$Add_1(b_1,b_2,b) = [b_3,b_4] \Rightarrow Add_n([b_1|x_1],[b_2|x_2],b) = [b_3|Add_n(x_1,x_2,b_4)]$

• در این مرحله باید ساختار مداری جمع‌کنندهٔ یک بیتی را طبق آنچه که در کتاب گفته شد، تعریف کنیم. فرض کنید $Add_1Circuit(c)$ در صورتی برای یک مدار درست است که مؤلفه‌ها و اتصالات مناسب باشد.

$$\forall c \ \ Add_1Circuit(c) \Leftrightarrow$$
$$\exists x_1, x_2, a_1, a_2, o_1 \ \ Type(x_1) = Type(x_2) = XOR$$
$$\wedge Type(a_1) = Type(a_2) = AND \wedge Type(o_1) = OR$$
$$\wedge Connected(Out(1, x_1), In(1, x_2)) \wedge Connected(In(1, c), In(1, x_1))$$
$$\wedge Connected(Out(1, x_1), In(2, a_2)) \wedge Connected(In(1, c), In(1, a_1))$$
$$\wedge Connected(Out(1, a_2), In(1, o_1)) \wedge Connected(In(2, c), In(2, x_1))$$
$$\wedge Connected(Out(1, a_1), In(2, o_1)) \wedge Connected(In(2, c), In(2, a_1))$$
$$\wedge Connected(Out(1, x_2), Out(1, c)) \wedge Connected(In(3, c), In(2, x_2))$$
$$\wedge Connected(Out(1, o_1), Out(2, c)) \wedge Connected(In(3, c), In(1, a_2))$$

لازم به ذکر است که این شیوه به مدار اجازه می‌دهد که گیت‌ها و اتصالات بیشتری داشته باشد ولی در انجام عملیات جمع اشکالی ایجاد نمی‌کنند.

• اکنون می‌خواهیم تعریف کنیم که طبق طراحی شکل ۸.۸ یک مدار جمع‌کنندهٔ $n$- بیتی چه معنایی دارد. دقت داشته باشید که یک جمع‌کنندهٔ $n$- بیتی، همان جمع‌کنندهٔ $n$-1 بیتی به علاوه یک جمع‌کنندهٔ یک بیتی نیست و بایستی بیت سرریز از جمع‌کنندهٔ $n$-1 بیتی را به ورودی بیت‌نقلی از جمع‌کنندهٔ یک بیتی متصل کنیم. در ابتدا با حالت پایه یعنی $n = 0$ آغاز می‌کنیم:

$$\forall c \ \ Add_nCircuit(c, 0) \Leftrightarrow Signal(Out(1, c)) = 0$$

به طور بازگشتی، اولین خروجی سرریز از مدار $n$-1 بیتی را به بیت نقلی یعنی آخرین بیت متصل کنید:

$$\forall c, n \ \ n > 0 \Rightarrow [Add_nCircuit(c, n) \Leftrightarrow$$
$$\exists c_2, d \ \ Add_nCircuit(c_2, n-1) \wedge Add_1Circuit(d)$$
$$\wedge \forall m \ \ (m > 0) \wedge (m < 2n-1) \Rightarrow In(m, c) = In(m, c_2)$$
$$\wedge \forall m \ \ (m > 0) \wedge (m < n) \Rightarrow \wedge Out(m, c) = Out(m, c_2)$$
$$\wedge Connected(Out(n, c_2), In(3, d))$$
$$\wedge Connected(In(2n-1, c), In(1, d)) \wedge Connected(In(2n, c), In(2, d))$$
$$\wedge Connected(Out(1, d), Out(n, c)) \wedge Connected(Out(2, d), Out(n+1, c))$$

• جهت اثبات اینکه یک مدار جمع‌کنندهٔ یک بیتی واقعا عملیات جمع را به درستی انجام می‌دهد، تعدادی ورودی به آن وارد کرده و خروجی آن را با مقدار واقعی حاصلجمع آنها مقایسه می‌کنیم تا به صحت عملکرد آن پی ببریم:

$$\forall c \ \ Add_1Circuit(c) \Rightarrow$$
$$\forall i_1, i_2, i_3 \ \ Signal(In(1, c)) = i_1 \wedge Signal(In(2, c)) = i_2 \wedge Signal(In(3, c)) = i_3$$
$$\Rightarrow Add_1(i_1, i_2, i_3) = [Out(1, c), Out(2, c)]$$

اگر این جمله توسط پایگاه دانش ایجاب شود، آنگاه هر مداری با طراحی $Add_1Circuit$ نیز در حقیقت یک جمع‌کننده است. سؤالی که از یک $n$- بیتی می‌توان پرسید، می‌تواند به صورت زیر باشد:

$$\forall c, n \ \ Add_nCircuit(c, n) \Rightarrow$$
$$\forall x_1, x_2, y \ \ InterleavedInputBits(x_1, x_2, c) \wedge OutputBits(y, c)$$
$$\Rightarrow Add_n(x_1, x_2, y)$$

عبارات $InterleavedInputBits$ و $OutputBits$ جهت نگاشت دنباله بیت‌ها به ترمینال‌های واقعی مداری استفاده شده‌اند. (توجه: این تدوین منطقی به طور تئوری اثبات نشده است لذا نمی‌توان صحت آن را تضمین نمود).

**؟ ۲۷.۸ یک فرم درخواست پاسپورت در کشورتان را بدست آورده و قواعد مربوط به واجدین شرایط آن را مطالعه نمایید. سپس طبق مراحل گفته شده در بخش ۴.۸، آن قواعد را به منطق مرتبهٔ اول تبدیل کنید. *(این تمرین مشابه تمرین ۱۹.۸ در ویرایش دوم است)***

☑ حل: پاسخ این سؤال در کشورهای مختلف متفاوت خواهد بود ولی دو قاعدهٔ اصلی برای پاسپورت‌های ایالت متحده را می‌توانید در پاسخ تمرین ۲۴.۸ بیابید.

**؟ ۲۸.۸ یک پایگاه دانش منطقی مرتبهٔ اول را در نظر بگیرید که این دنیا شامل شخص،آهنگ،آلبوم(مانند Meet the Beatle و دیسک(مانند نمونه های فیزیکی CD) می‌باشد. فرهنگ واژگان شامل این نمادها است:**

**CopyOf(d,a): مسند. دیسک d یک کپی از آلبوم a است. Owns(p,d): مسند. شخص p صاحب دیسک d است. Sings(p,s,a): آلبوم a شامل آهنگ s است که توسط شخص p خوانده شده است. Wrote(p,s): شخص p، آهنگ s را نوشته**

است. Revolver،TheManILove،EleanorRigby،Joe،BHolding،Gershwin،McCartney: نمادهای ثابتی با همین معنا هستند.

این جملات را به منطق مرتبهٔ اول بیان کنید: الف) Gershwin، آهنگ «The man I Love» را نوشته است. ب) Gershwin، آهنگ «Eleanor Rigby» را ننوشته است. ج) Gershwin یا Mccartney، آهنگ «The man I Love»را نوشته‌اند. د) Joe حداقل یک آهنگ نوشته است. هـ) Joe صاحب یک کپی از آلبوم Revolver است. و) هر آهنگی که Mccartney در آلبوم Revolver خوانده، توسط Mccartney نوشته شده است. ز) Gershwin، هیچیک از آهنگ‌های آلبوم Revolver را ننوشته است. ح) هر آهنگی که Gershwin نوشته است در همان آلبوم ثبت شده است(ممکن است آهنگهای مختلف درآلبوم‌های مختلف ثبت شده باشند) ط) یک آلبوم واحد وجود دارد که شامل تمام آهنگهایی است که Joe نوشته است. ی) joe صاحب یک کپی از آلبومی است که در آن Billie Holiday آهنگ The man I Love را خوانده است. ک) joe صاحب یک کپی از آلبومی است که در آن یکی از آهنگهای خوانده شده توسط Mccartney وجود دارد.(البته هر آلبوم مختلف بر روی یک CD مختلف قرار دارد.) ل) Joe صاحب تمام آلبوم‌هایی است که در آنها تمام آهنگها توسط Billie Holiday خوانده شده است.

☑ حل: **الف)** $W(G,T)$ **ب)** $\neg W(G,E)$ **ج)** $W(G,T) \vee W(M,T)$ **د)** $\exists s\ W(J,s)$

**هـ)** $\exists x\ C(x,R) \wedge O(J,x)$ **و)** $\forall s\ S(M,s,R) \Rightarrow W(M,s)$ **ز)** $\neg[\exists s\ W(G,s) \wedge \exists p\ S(p,s,R)]$

**ح)** $\forall s\ W(G,s) \Rightarrow \exists p,a\ S(p,s,a)$ **ط)** $\exists a\ \forall s\ W(J,s) \Rightarrow \exists p\ S(p,s,a)$

**ی)** $\exists d,a,s\ C(d,a) \wedge O(J,d) \wedge S(B,T,a)$ **ک)** $\forall a\ [\exists s\ S(M,s,a)] \Rightarrow \exists d\ C(d,a) \wedge O(J,d)$

**ل)** $\forall a\ [\forall s,p\ S(p,s,a) \Rightarrow S(B,s,a)] \Rightarrow \exists d\ C(d,a) \wedge O(J,d)$

# فصل ۹ (ویرایش سوم)

**؟ ۹.۱ با استفاده از اصول اولیه، اثبات کنید که نمونه‌سازی عمومی صحیح است و نمونه‌سازی وجودی، پایگاه‌دانشی تولید می‌کند که در استنتاج هم‌ارز است. *(این تمرین مشابه تمرین ۱.۹ در ویرایش دوم است)***

☑ حل: در نمونه‌سازی عمومی، به جای متغیری که در کنار سور عمومی آمده است، تمام اصطلاحات موجود در محیط را جایگزین کرده و به ازای هریک جمله‌ای جدید تولید می‌شود. برای اثبات صحت اینکار، باید نشان دهیم که هر عبارت به شکل $\forall v\ a$ می‌تواند تمام نمونه‌سازی‌های عمومی ایجاد شده از آنرا ایجاب کند. عبارت $\forall v\ a$ درصورتی درست نامیده می‌شود که در تمام تفسیرهای ممکن صحیح باشد. به عنوان مثال $v$ را با یک اصطلاح پایه‌ای $g$ جایگزین می‌کنیم که یک تفسیر ممکن است آنگاه اگر عبارت اصلی صحیح باشد، عبارت نمونه‌سازی شده نیز باید صحیح باشد. حالات EI: برای هر جملهٔ $\alpha$ متغیر $v$ و نماد ثابت $k$ که در هیچ جای دیگری از پایگاه دانش استفاده نشده است:

$$\frac{\exists v\ \alpha}{\text{SUBST}(\{v/k\},\alpha)}$$

اگر پایگاه دانش با جملهٔ سور وجودی را با $KB$ و نتیجهٔ حاصل از EI را با $KB'$ نمایش دهیم، باید اثبات کنیم که $KB$ ارضاپذیر است، اگر و فقط اگر $KB'$ ارضاپذیر باشد. روش پیش‌رو: اگر $KB$ ارضاپذیر باشد، آنگاه یک مدل $M$ ناشی از انتساب $v$ به شی $o$ دارد که در آن $\alpha$ درست باشد. بنابراین می‌توانیم مدل $M'$ را بسازیم که در آن $KB'$ با انتساب $k$ به شی $o$ ارضاپذیرباشد. از آن جا که $k$ در هیچ جای دیگری استفاده نشده است، بر روی مقدار درستی هیچ جمله‌ای تأثیر نمی‌گذارد. روش پس‌رو: اگر $KB'$ ارضاپذیر باشد، آنگاه یک مدل $M'$ ناشی از انتساب $k$ به شی $o$ دارد. بنابراین اگر یک مدل $M$ بسازیم که $KB$ با انتساب $v$ به $o$ در آن ارضاپذیر باشد، از آن جا که $k$ در هیچ جای دیگری استفاده نشده است، حذف آن از مدل نیز بر روی مقدار درستی سایر جملات هیچ تأثیری نمی‌گذارد.

**؟ ۹.۲ طبق عبارت $Likes(Jerry,Icecream)$ منطقی است که عبارت $\exists x\ Likes(x,IceCream)$ را نتیجه بگیریم. یک قاعدهٔ کلی به نام «معرفی وجودی» بنویسید که این نتیجه‌گیری را تایید کند. شرایطی که متغیرها و اصطلاحات این قاعده باید رعایت کنند را به دقت بیان کنید. *(این تمرین مشابه تمرین ۲.۹ در ویرایش دوم است)***

☑ حل: برای هر عبارت $a$ که شامل اصطلاح پایه‌ای $g$ بوده و برای هر متغیر $v$ که در $a$ نباشد داریم:

$$\frac{\alpha}{\exists v\ \text{SUBST}_1(\{g/v\},\alpha)}$$

که $\text{SUBST}_1$ تابعی است که یک رخداد $g$ را با $v$ جایگزین می‌کند. دقت داشته باشید که جایگزینی یک رخداد و استفاده چندباره از یک قانون نتیجه یکسانی ندارند. زیرا اینکار موجب ایجاد نتیجه‌ای ضعیف‌تر می‌گردد. بعنوان مثال، $P(a,a)$ باید عبارت $\exists x\ P(x,x)$ را ایجاب کند ولی نتیجه ضعیف‌تر آن عبارتست از: $\exists x,y\ P(x,y)$.

**؟ ۹.۳ فرض کنید یک پایگاه‌دانش تنها شامل یک عبارت به صورت $\exists x\ AsHighAs(x,Everest)$ می‌باشد. کدامیک از این موارد نتیجهٔ صحیح اعمال نمونه‌سازی وجودی می‌باشد: الف) $AsHighAs(Everest,Everest)$ ب) $AsHighAs(Kilimanjaro,Everest)$ ج) $AsHighAs(Kilimanjaro,Everest) \wedge AsHighAs(BenNevis,Everest)$ (پس از اعمال دوبار) *(این تمرین مشابه تمرین ۳.۹ در ویرایش دوم است)***

☑ حل: هردو مورد ب و ج معتبر می‌باشند زیرا در طی عملیات نمونه‌سازی وجودی باید نامی انتخاب شود که در هیچ جای پایگاه استفاده نشده باشد. مورد الف نامعتبر است زیرا از نماد $Everest$ که قبلا بکار رفته بود، استفاده کرده است. دقت کنید که قسمت ج نمی‌گوید دو کوه هم‌ارتفاع با $Everest$ داریم زیرا در هیچ جای آن نگفته است که نمادهای $BenNevis$ و $Kilimanjaro$ متفاوت هستند.

**؟ ۹.۴ برای هریک از جملات ساده زیر، در صورت وجود، عمومی‌ترین یکسان‌ساز را بیان کنید.**
**الف) $P(x,y,z),P(z,y,x)$ ب) $P(A,B,B),P(x,y,z)$ ج) $Q(y,G(A,B)),Q(G(x,x),y)$ د) $Older(Father(y),y),Older(Father(x),John)$**
***(این تمرین مشابه تمرین ۴.۹ در ویرایش دوم است)* $Knows(Father(y),y),Knows(x,x)$**

☑ حل: منظور از یکسان‌سازی آن است که دو جمله را گرفته و برای آنها یک مجموعه جایگزینی بیابیم که با اعمال آن، هر دو جمله منطقا یکسان باشند. اگر دو جمله داشته باشیم که بتوان به روش‌های مختلف آنها را یکسان‌سازی نمود (چندین مجموعه جایگزینی ممکن باشد)، قانون عمومی‌ترین یکسان‌ساز، موردی را انتخاب می‌کند که کلی‌تر بوده و محدودیت کمتری برای متغیرها ایجاد نماید. **الف)** $\{x/A\ ,\ y/B\ ,z/B\}$ (ویا سایر جایگشت‌های مشابه). **ب)** یکسان‌سازی نمی‌شود (یک متغیر نمی‌تواند برای هر دو نماد $A$ و $B$ بکار رود). **ج)** $\{y/John\ ,\ x/John\}$ **د)** یکسان‌سازی نمی‌شود (زیرا «بررسی رخدادها» مانع

از یکسان‌سازی $y$ با $Father(y)$ می‌شود. این روش بررسی می‌کند که یک متغیر با عبارتی شامل خود متغیر یکسان‌سازی نشود پس $y/Father(y)$ اشتباه است.)

**؟ ۵.۹ شبکه شمول شکل ۲.۹ را در نظر بگیرید: الف) برای عبارت ((*Employs*(*Mother*(*John*) , *Father*(*Richard*)) شبکه شمولی تشکیل دهید. ب) برای عبارت *Employs*(*IBM*,*y*) که نشان‌دهندهٔ آن است که «همه برای *IBM* کار می‌کنند»، شبکه شمولی تشکیل دهید. به خاطر داشته باشید که تمام پرس‌وجوهایی که با این عبارت یکسان‌سازی می‌شوند را به حساب آورید. ج) فرض کنید *STORE* باعث می‌شود تا هرجمله به گره‌ای در شبکه شمول مرتبط شود. توضیح دهید که اگر برخی از این جملات حاوی متغیرهایی باشند، *FETCH* چگونه باید کارکند به عنوان مثال از عبارات قسمت الف و ب و پرس‌وجوی (*Employs*(*x*,*Father*(*x*))) استفاده کنید. *(این تمرین مشابه تمرین ۵.۹ در ویرایش دوم است)***

☑ حل: برای ترسیم شبکه شمول قسمت الف به صورت شکل ۲.۹ نیاز به فضای زیادی داریم. لذا در اینجا به علت کمبود فضا آن را به صورت گره‌های فرزند در زیر هم ترسیم می‌کنیم:

```
[1] Employs(x, y)
  [2] Employs(x, Father(z))
  [3] Employs(x, Father(Richard))
    [4] Employs(Mother(w), Father(Richard))
      [5] Employs(Mother(John), Father(Richard))
  [6] Employs(Mother(w), Father(z))
    [4] ...
    [7] Employs(Mother(John), Father(z))
      [5] ...
    [8] Employs(Mother(w), y)
      [9] Employs(Mother(John), y)
        [10] Employs(Mother(John), Father(z)
          [5] ...
      [6] ...
```

شبکه شمول یک عبارت، شامل تمام پرس‌وجوهایی است که می‌توانند با آن یکسان‌سازی شوند. در شبکه شمول، پایین‌ترین گره همان چیزی است که سئوال از ما خواسته است. یک سطح بالاتر کمی کلی تر بوده و تنها با یک جایگزینی در این سطح، به سطح بعد می‌رسیم. اینکار آنقدر تا مراحل بالا ادامه می‌یابد که به عمومی‌ترین حالت عبارت برسیم. یعنی بالاترین نسل مشترک هر دو گره در این شبکه، نتیجه اعمال عمومی‌ترین یکسان‌ساز است. **ب)** برای عبارت قسمت ب، دو گره *Employs*(*x*,*y*) و *Employs*(*y*,*y*) خواهیم داشت. **ج)** ابتدا بایستی عبارات *Employs*(*Mother*(*John*) و *Father*(*Richard*) و *Employs*(*IBM*,*y*) واکشی شوند تا بتوانیم به سئوال *Employs*(*x*,*Father*(*x*)) پاسخ دهیم. که جواب آن *Richard* *Father* یعنی «پدر ریچارد» می‌گردد.

**؟ ۶.۹ برای جملات زیر بازنمایی منطقی بنویسید که قابل استفاده در قیاس‌استثنایی تعمیم‌یافته باشد. الف) اسب‌ها، گاوها و خوک‌ها پستاندار هستند. ب) یک کره اسب، یک اسب است. ج) ریش آبی یک اسب است. د) ریش آبی والد چارلی است. هـ) کره بودن و والد بودن، روابطی معکوس هستند. و) هر پستاندار یک والد دارد. *(این تمرین مشابه تمرین ۹.۹ در ویرایش دوم است)***

☑ حل: در اینجا با استفاده از علم موجودات و هستی‌شناسی، مثال‌ها را بسیار ساده‌تر می‌کنیم:

**الف)**
$$Horse(x) \Rightarrow Mammal(x)$$
$$Cow(x) \Rightarrow Mammal(x)$$
$$Pig(x) \Rightarrow Mammal(x)$$

**ب)**
$$Offspring(x,y) \wedge Horse(y) \Rightarrow Horse(x)$$

**ج)**
$$Horse(Bluebeard)$$

**د)**
$$Parent(Bluebeard, Charlie)$$

**د)**
$$Offspring(x,y) \Rightarrow Parent(y,x)$$
$$Parent(x,y) \Rightarrow Offspring(y,x)$$

ـ $Offspring(x,y) \Leftrightarrow Parent(y,x)$

توجه داشته باشید که نمی‌توانیم قاعده‌ای به صورت $Offspring(x,y) \Leftrightarrow Parent(y,x)$ داشته باشیم زیرا به شکل قاعدهٔ یاس استثنایی تعمیم‌یافته نیست).

) $Mammal(x) \Rightarrow Parent(G(x),x)$ ($G$ در اینجا یک تابع اسکولم است).

**؟ ۹.۷ این تمرین نتایج حاصل از جایگزینی و اسکولم‌سازی را بررسی می‌کند. الف) با توجه به مقدم $\forall x\ \exists y\ P(x,y)$ نمی‌توان لی $\exists q\ P(q,q)$ را از آن نتیجه گرفت. مثالی از مسند P ارائه دهید که در آن بخش ابتدایی درست بوده ولی بخش ثانویه درست باشد. ب) فرض کنید یک موتور استنتاج که با بررسی رخدادها کار میکند، اشتباه نوشته شده است و اجازه میدهد تا ک لفظ مانند $P(x,F(x))$ با $P(q,q)$ یکسان‌سازی شود. (بیشتر پیاده‌سازی‌های استاندارد پرولوگ این کار را مجاز می‌دانند) نشان دهید که این موتور استنتاج اجازه می‌دهد تا تالی $\exists y\ P(q,q)$ از مقدم $\forall x\ \exists y\ P(x,y)$ نتیجه گرفته شود. ج) فرض کنید ک تابع که منطق مرتبهٔ اول را به شکل بندی تبدیل می‌کند، اشتباها عبارت $\forall x\ \exists y\ P(x,y)$ را به $P(x,SK0)$ اسکولم کرده ست. (در واقع $y$ را با یک نماد اسکولم به جای یک تابع اسکولم از $x$ جایگزین کرده است). نشان دهید که یک موتور استنتاج ه از این تابع استفاده می‌کند اجازه می‌دهد تا عبارت $\exists q\ P(q,q)$ از مقدم $\forall x,y\ P(x,y)$ نتیجه شود. د) یکی از اشتباهات رایج نشجویان در طی یکسان‌سازی آن است که به جای جایگزینی یک اصطلاح با متغیر، از جایگزینی اصطلاح با ثابت اسکولم ستفاده می‌کنند. بعنوان نمونه، آنها فرمولهای $P(A)$ و $P(SK1)$ را تحت جایگزینی $\{SK1/A\}$ یکسان‌سازی می‌کنند. مثالی ائه دهید که نشان دهد این کار منجر به استنتاج نامعتبر می‌شود.**

حل: **الف)** فرض کنید $P(x,y)$ نشان‌دهندهٔ رابطهٔ «$x$ کمتر از $y$ است» بر روی اعداد صحیح باشد. آنگاه عبارت $\forall x\ \exists y\ P(x,y)$ درست بوده و عبارت $\exists x\ P(x,x)$ نادرست می‌باشد. **ب)** تبدیل مقدم به شکل بندی عبارتست از $P(x,SK0)$ و تبدیل نقیض هدف به شکل‌بندی عبارتست از $\neg P(q,q)$. اگر این دو فرمول را یکسان کنیم، به بندی تهی جر می‌شوند. **ج)** اگر مقدم را به صورت $P(x,SK0)$ نشان دهیم و نقیض هدف به درستی به صورت $\neg P(q,q)$ تبدیل شود، گاه تحت جایگزینی $\{q/SK0, x/SK0\}$ به بندی تهی می‌رسیم. **د)** فرض کنید مقدم به صورت $\exists x\ Cat(x)$ بوده و قصد ات $Cat(Socrates)$ را دارید. تبدیل مقدم به شکل‌بندی عبارتست از $Cat(SK1)$ که اگر آن را با $Cat(Socrates)$ سان‌سازی کنید، آنگاه می‌توانید آن را با نقیض هدف یعنی $\neg Cat(Socrates)$ حل کنید که به بندی تهی منجر می‌شود.

**؟ ۹.۸ توضیح دهید که چگونه می‌توان مساله 3-SAT با هراندازهٔ دلخواه را به صورت یک بند متناهی مرتبهٔ اول و حداکثر 30 قعیت پایه بیان نمود.** ***(این تمرین مشابه تمرین ۹.۸ در ویرایش دوم است)***

حل: یک مساله 3-SAT به فرم زیر را در نظر بگیرید:

$$(x_{1,1} \vee x_{2,1} \vee \neg x_{3,1}) \wedge (\neg x_{1,2} \vee x_{2,2} \vee x_{3,2}) \vee \ldots$$

ن مساله را به عنوان یک بند متناهی منفرد به شکل زیر بازنویسی می‌کنیم که حاوی چند بند پایه است:

$$A \wedge B \wedge C \wedge \ldots \Rightarrow Z$$

بق بند متناهی داریم:

$$OneOf(x_{1,1}, x_{2,1}, Not(x_{3,1})) \wedge OneOf(Not(x_{1,2}), x_{2,2}, x_{3,2}) \wedge \ldots \Rightarrow Solved$$

ته آن است که بندهای متناهی باید به هر رخداد از یک متغیر، مقدار یکسانی انتساب دهند، حتی اگر آن متغیر در برخی از دها نقیض باشد و در سایرین خیر. در اینجا نیاز به تعریف $OneOf$ داریم. این مورد را در زیر مشاهده می‌کنید:

$$OneOf(True, x, y)$$
$$OneOf(x, True, y)$$
$$OneOf(x, y, True)$$
$$OneOf(Not(False), x, y)$$
$$OneOf(x, Not(False), y)$$
$$OneOf(x, y, Not(False))$$

**؟ ۹.۹ فرض کنید قواعد زیر را دارید:**

**$0 \le 3$ .2 $7 \le 9$ .3 $\forall x\ x < x$ .4 $\forall x\ x \le x+0$ .5 $\forall x\ x+0 \le x$**

**.6 $\forall x,y\ x+y \le y+x$ .7 $\forall w,x,y,z\ w \le y \wedge x \le z \Rightarrow w+x \le y+z$ .8 $\forall x,y,z\ x \le y \wedge y \le z \Rightarrow x \le z$**

**ب) یک اثبات زنجیره‌ای پس‌رو برای جملهٔ $7 \le 3+9$ ارائه دهید. (فقط از قواعد مندرج در فوق استفاده کنید و از دانستنی‌های ود در مورد حساب استفاده نکنید). فقط مراحل لازم جهت رسیدن به نتیجه را نشان دهید و مراحل نامربوط را ذکر نکنید. ب) ک اثبات زنجیره‌ای پیش‌رو برای جملهٔ $7 \le 3+9$ ارائه دهید. این بار نیز فقط مراحل مرتبط را نشان دهید.**

حل: این تمرین با کمی مهارت قابل حل است که با بررسی دقیق هر مرحله می‌توانید روند پاسخ را دریابید. **الف)** (توجه: در رزولوشن، متغیرهای قواعد را تغییر نام داده‌ایم.)

Goal G0: $7 \le 3+9$ — با عبارت8، $\{x_1/7\ ,\ z_1/3+9\}$ حل کنید.

Goal G1: $7 \le y_1$ — با عبارت4، $\{x_2/7\ ,\ y_1/7+0\}$ حل کنید.

Goal G2: $7+0 \le 3+9$ — با عبارت8، $\{x_3/7+0\ ,\ z_3/3+9\}$ حل کنید.

با عبارت6، $\{x_4/7, y_4/0, y_3/0+7\}$ حل کنید. Goal G3: $7+0 \leq y_3$
با عبارت7، $\{w_5/0, x_5/7, y_5/3, z_5/9\}$ حل کنید. Goal G4: $0+7 \leq 3+9$
با عبارت1، حل کنید. نتیجه بدست می‌آید. Goal G5: $0 \leq 3$
با عبارت2، حل کنید. نتیجه بدست می‌آید. Goal G6: $7 \leq 9$
موفق شد G4
موفق شد G2
موفق شد G0

**؟ ۱۰.۹ یکی از معروف‌ترین معماهای کودکان عبارتست از: «من برادر و خواهری ندارم ولی پدرآن مرد پسر پدر من است». با استفاده از قواعد دامنهٔ خویشاوندی(بخش ۸. ۳. ۲) نشان دهید که «آن مرد چه کسی است؟». می‌توانید از روش‌های استنتاج گفته شده در این فصل نیز استفاده کنید. به نظر شما چرا این معما دشوار است؟** ***(این تمرین مشابه تمرین ۱۱.۹ در ویرایش دوم است)***

☑ حل: دشوارترین بخش این معما مربوط به جملهٔ «آن مرد چه کسی است» می‌شود. می‌خواهیم سوال «چه رابطه خویشاوندی بین آن مرد و شخص شناخته شده در معما وجود دارد؟ » را بپرسیم ولی اگر این رابطه خویشاوندی به صورت یک مسند مانند $Parent(x,y)$ در نظر بگیریم، آنگاه نمی‌توان رابطهٔ خویشاوندی را به صورت یک متغیر در منطق مرتبهٔ اول بیان نمود. بنابراین ناچاریم به روابط خویشاوندی جسمیت داده و آنها را به صورت شی در نظر بگیریم. از عبارت $Rel(r,x,y)$ برای بیان این نکته استفاده می‌شود که رابطهٔ خویشاوندی $r$ بین فرد $y$ و $x$ وجود دارد. برای معرفی خود شخص از $Me$ و معرفی آن مرد از $MrX$ استفاده می‌کنیم. همچنین از ثابت اسکولم $FM$ برای معرفی پدر شخص (پدر $Me$) و از $FX$ برای معرفی پدر آن مرد (پدر $MrX$) استفاده می‌کنیم. حقایق این مساله به شکل نرمال استلزامی عبارت است از:

(1) $Rel(Sibling, Me, x) \Rightarrow False$
(2) $Male(MrX)$
(3) $Rel(Father, FX, MrX)$
(4) $Rel(Father, FM, Me)$
(5) $Rel(Son, FX, FM)$

باید نشان دهیم که $Me$ تنها پسر پدرم می‌باشد و $Me$ پدر $MrX$ است که آن مذکر بوده و بنابراین آن مرد، پسر من می‌باشد. این تعاریف طبق حوزهٔ روابط خویشاوندی عبارت است از:

(6) $Rel(Parent, x, y) \wedge Male(x) \Leftrightarrow Rel(Father, x, y)$
(7) $Rel(Son, x, y) \Leftrightarrow Rel(Parent, y, x) \wedge Male(x)$
(8) $Rel(Sibling, x, y) \Leftrightarrow x \neq y \wedge \exists p \ Rel(Parent, p, x) \wedge Rel(Parent, p, y)$
(9) $Rel(Father, x_1, y) \wedge Rel(Father, x_2, y) \Rightarrow x_1 = x_2$

و سوالی که ما داریم عبارت است از: $(Q)\ Rel(r, MrX, y)$

به پاسخ $\{r/Son, y/Me\}$ می‌رسیم. با ترجمه جملات 1 تا 9 و $Q$ به فرم INF و (نقیض $Q$ و تعریف $\neq$) داریم:

(6a) $Rel(Parent, x, y) \wedge Male(x) \Rightarrow Rel(Father, x, y)$
(6b) $Rel(Father, x, y) \Rightarrow Male(x)$
(6c) $Rel(Father, x, y) \Rightarrow Rel(Parent, x, y)$
(7a) $Rel(Son, x, y) \Rightarrow Rel(Parent, y, x)$
(7b) $Rel(Son, x, y) \Rightarrow Male(x))$
(7c) $Rel(Parent, y, x) \wedge Male(x) \Rightarrow Rel(Son, x, y)$
(8a) $Rel(Sibling, x, y) \Rightarrow x \neq y$
(8b) $Rel(Sibling, x, y) \Rightarrow Rel(Parent, P(x,y), x)$
(8c) $Rel(Sibling, x, y) \Rightarrow Rel(Parent, P(x,y), y)$
(8d) $Rel(Parent, P(x,y), x) \wedge Rel(Parent, P(x,y), y) \wedge x \neq y \Rightarrow Rel(Sibling, x, y)$
(9) $Rel(Father, x_1, y) \wedge Rel(Father, x_2, y) \Rightarrow x_1 = x_2$
$(N)\ True \Rightarrow x = y \vee x \neq y$
$(N')\ x = y \wedge x \neq y \Rightarrow False$
$(Q')\ Rel(r, MrX, y) \Rightarrow False$

جه کنید که (۱) به شکل هورن نیست بنابراین باید از رزولوشن استفاده کنیم تا از دستیابی به راه‌حل مطمئن شویم. همچنین ان می‌دهد که بایستی از مدولاسیون معکوس برای سروکار داشتن با تساوی استفاده کنیم. در ادامه لیستی از مراحل اثبات را ی‌بینید که جواب هر مرحله درون پرانتز بیان شده است:

| | |
|---|---|
| (10) $Rel(Parent, FM, Me)$ | $(4, 6c)$ |
| (11) $Rel(Parent, FM, FX)$ | $(5, 7a)$ |
| (12) $Rel(Parent, FM, y) \wedge Me \neq y \Rightarrow Rel(Sibling, Me, y)$ | $(10, 8d)$ |
| (13) $Rel(Parent, FM, y) \wedge Me \neq y \Rightarrow False$ | $(12, 1)$ |
| (14) $Me \neq FX \Rightarrow False$ | $(13, 11)$ |
| (15) $Me = FX$ | $(14, N)$ |
| (16) $Rel(Father, Me, MrX)$ | $(15, 3, demodulation)$ |
| (17) $Rel(Parent, Me, MrX)$ | $(16, 6c)$ |
| (18) $Rel(Son, MrX, Me)$ | $(17, 2, 7c)$ |
| (19) $False$ $\{r/Son, y/Me\}$ | $(18, Q')$ |

**۱۱.۹ فرض کنید بخشی از اطلاعات آماری مردم ایالت متحده شامل سن، شهرسکونت، تاریخ تولد و نام مادر هر شخص را رون پایگاه‌داده منطقی قرارداده‌ایم که شماره امنیت اجتماعی هرشخص یکتا بوده و جهت شناسایی او بکار می‌رود. بنابراین ن George به صورت $Age$(443-65-1285-56) بیان می‌شود. که عدد درون پرانتز همان شماره امنیت اجتماعی جرج ت.کدامیک از راه‌حل‌های $S_1$ تا $S_5$ برای پاسخ به سوالات $Q_1$ تا $Q_4$ راه‌حلی مناسب است؟ (با فرض استفاده از روش زنجیره ای س گرد نرمال).**

**$S_1$: شاخصی برای هر بسیط در هرموقعیت • $S_2$: شاخصی برای هر آرگومان اولیه • $S_3$: شاخصی برای هر مسند بسیط • $S_4$: اخصی برای هر ترکیب مسند و آرگومان اولیه • $S_5$: شاخصی برای هر ترکیب مسند و آرگومان ثانویه و شاخصی برای هر گومان اولیه (غیراستاندارد) • $Q_1$: $(x,Houston)$ $Q_2$: $Age(443\text{-}44\text{-}4321,x)$ • $Q_3$: $ResidesIn$ • $Q_4$: $Mother(x,y)$ • $Age(x,34) \wedge ResidesIn(x,TinyTownUSA)$ *(این تمرین مشابه تمرین ۶.۹ در ویرایش دوم است)***

حل: در جدول زیر پیچیدگی زمانی در حالت متوسط برای هر ترکیب پرس‌وجو/راه‌حل بیان شده است. (منظور از «$1;n$» آن ت که برای یافتن اولین راه‌حل به $O(1)$ و یافتن تمام راه‌حل‌ها به $O(n)$ نیازمندیم). فرضیه زیر را در نظر بگیرید: دسترسی جدول، پیچیدگی $O(1)$ را نیاز دارد. $n$ فرد در بانک اطلاعاتی موجود است. $O(n)$ فرد با سنی مشخص داریم. هر فرد یک در دارد. $H$ فرد در $Houston$ و $T$ فرد در $TinyTown$ وجود دارند. $T$ خیلی کمتر از $n$ است در پرس وجوی $Q_4$ بخش دوم ابتدا ارزیابی می‌شود.

| | Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|---|
| S1 | 1 | $1; H$ | $1; n$ | $T; T$ |
| S2 | 1 | $n; n$ | $1; n$ | $n; n$ |
| S3 | $n$ | $n; n$ | $1; n$ | $n^2; n^2$ |
| S4 | 1 | $n; n$ | $1; n$ | $n; n$ |
| S5 | 1 | $1; H$ | $1; n$ | $T; T$ |

رچیزی با $O(1)$ وحتی $O(t)$ می‌تواند کارا محسوب شود. توجه داشته باشید که $S_1$ و $S_5$ سایر روش‌ها برای این مجموعه س‌وجو را نشان می‌دهند. همچنین دقت کنید که شاخص‌گذاری مسندها هیچ نقشی در جدول ایفا نمی‌کند.(مگر درترکیب با آرگومان) زیرا فقط سه مسند داریم (که پیچدگی آن $O(1)$ است) که در فاکتور ثابت اندکی تغییر ایجاد می‌کند.

**۱۲.۹ به نظر می‌رسد که بتوان از مشکل تناقض متغیرها در طی عملیات زنجیره‌ای پیش‌رو برای استانداردسازی تمام جملات گاه جلوگیری کرد. نشان دهید که این شیوه برای برخی جملات کارساز نیست. (راهنمایی: جمله‌ای را در نظر بگیرید که یک ش از آن با بخش دیگر یکسان‌سازی شود) *(این تمرین مشابه تمرین ۷.۹ در ویرایش دوم است)***

حل: در صورتی این شیوه کارساز خواهد بود که در پایگاه دانش، قواعد بازگشتی نداشته باشیم. فرض کنید پایگاه دانشی مل جملات زیر داشته باشیم:

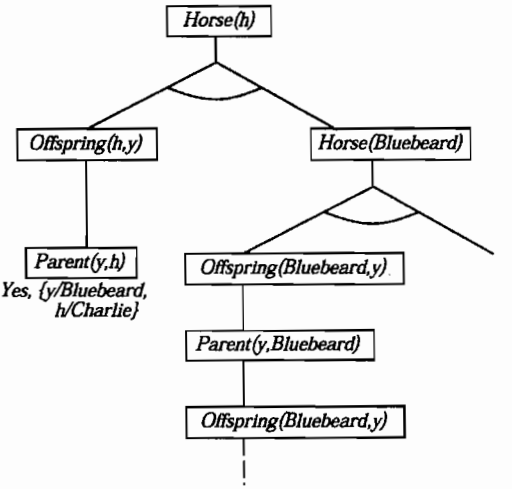
$$Member(x, [x|r])$$
$$Member(x, r) \Rightarrow Member(x, [y|r])$$

ون پرس‌وجوی $Member(3,[1,2,3])$ را با روش زنجیره‌ای پس‌رو اعمال کنید. توسط جایگزین‌های $\theta=\{x/3,y/1,r/[2,3$ پرس‌وجو را یکسان‌سازی می‌کنیم. سپس آن را به بخش سمت چپ اعمال می‌کنیم تا عبارت $Member(3,[2,3$ حاصل شود و سعی می‌کنیم با زنجیره‌ای پس‌رو به جایگزین $\theta$ بازگردیم. زمانی که سعی در اعمال مجدد یم با شکست مواجه می‌شویم زیرا $y$ نمی‌تواند هر دو مقدار 1 و 2 را درخود جای دهد. به عبارت دیگر اگر در جایی که قواعد

بازگشتی برای یک راه‌حل نیاز است از استانداردسازی اجزا استفاده کنیم شکست در استانداردسازی یک بخش، موجب شکست در آن مورد می‌شود.

**؟ ۱۳.۹ در این تمرین از جملاتی که شما در تمرین ۶.۹ نوشتید استفاده می‌کنیم تا با استفاده از الگوریتم زنجیره‌ای پس‌رو به یک سوال پاسخ دهیم. الف) درخت اثبات تولید شده توسط الگوریتم زنجیره‌ای پس‌رو برای سوال $\exists h\ Horse(h)$ را ترسیم کنید که در آن، بندها طبق ترتیب داده شده منطبق باشند. ب) دربارهٔ این دامنه به چه چیزی باید توجه کنید. ج) طبق جملات شما، در حقیقت چه تعداد راه‌حل برای h وجود دارد؟ د) فکر می‌کنید بتوانید تمام آنها را بیابید؟ (راهنمایی: شاید نیاز به استفاده از مقالهٔ Smith (1986) داشته باشید). *(این تمرین مشابه تمرین ۱۰.۹ در ویرایش دوم است)***

☑ حل: این تمرین با حلقه‌ها در اثبات زنجیره‌ای پس‌رو سروکار دارد. یک حلقه، محدود به رخدادی است که یک زیرهدف با جایگذاری در یکی از هدف‌های موجود در پشته بدست می‌آید. البته تمام حلقه‌ها به این شیوه به دست نمی‌آیند و برای حل این گونه مسائل راه‌هایی داریم. **الف)** درخت اثبات را در شکل ۱.۹s مشاهده می‌کنید. شاخه‌های $Parent(y,Bluebeard)$ و $Offspring(Bluebeard,y)$ به طور بی‌نهایت تکرار می‌شوند مابقی درخت هرگز به دست نمی‌آید. **ب)** به خاطر قاعدهٔ ب یعنی $Offspring(x,y) \wedge Horse(y) \Rightarrow Horse(x)$ یک حلقهٔ بی‌نهایت داریم. به خاطر ترتیبی که برای بندها در نظر گرفتیم حلقه‌ای که در شکل مشخص است رخ می‌دهد. بهتر بود بند $Horse(Bluebeard)$ قبل از قاعدهٔ ب قرار می‌گرفت. به هرحال اگر بخواهیم برای اثبات تئوری از تمام راه‌حل‌ها استفاده کنیم، رخداد یک حلقه، ربطی به ترتیب قواعد ندارد. **ج)** باید به گونه‌ای باشد که اثبات نماید هر دوی $Charlie$ و $Bluebeard$ اسب هستند. **د)** Smith (1986) در مقالهٔ خود روش زیر را پیشنهاد می‌دهد: زمانی که یک هدف حلقوی رخ می‌دهد (که آن هدف یک نمونه جایگذاری در یک ابرهدف بالای پشته است) سعی کنید موقتا برای اثبات آن زیرهدف، آن را کنار بگذارید. با سایر شاخه‌های اثبات ابرهدف ادامه دهید و به راه‌حل‌ها دست یابید. سپس این راه‌حل‌ها را به عنوان راه‌حل‌های همان زیرهدف‌های کنارگذاشته در نظر بگیرید و با شاخه‌های اثبات ادامه دهید تا درصورت وجود بتوانید راه‌حل‌های بیشتری بیابید. طبق اثبات نمایش داده شده در شکل، عبارت $Offspring(Bluebeard,y)$ یک هدف تکراری است و باید کنار گذاشته شود. از آنجا که هیچ راه دیگری برای اثبات وجود آن نداریم. آن شاخه با شکست پایان می‌یابد. در این مورد روش Smith جهت یافتن هر دو راه‌حل مناسب است.




شکل ۱.۹s قسمتی از درخت اثبات برای یافتن اسبها

**؟ ۱۴.۹ اجرای الگوریتم زنجیره‌ای پس‌رو را در شکل ۶.۹ برای حل مساله جرم، دنبال کنید. دنباله مقادیر کسب‌شده توسط متغیرهای هدف را نشان داده و آنها را در یک درخت مرتب کنید (این تمرین مشابه تمرین ۱۲.۹ در ویرایش دوم است)**

☑ حل: یک درخت هدف را مشاهده می‌کنید:

```
goals = [Criminal(West)]
  goals = [American(West), Weapon(y), Sells(West, y, z), Hostile(z)]
  goals = [Weapon(y), Sells(West, y, z), Hostile(z)]
    goals = [Missle(y), Sells(West, y, z), Hostile(z)]
    goals = [Sells(West, M1, z), Hostile(z)]
      goals = [Missle(M1), Owns(Nono, M1), Hostile(Nono)]
      goals = [Owns(Nono, M1), Hostile(Nono)]
    goals = [Hostile(Nono)]
goals = []
```

**؟ ۱۵.۹ کد پرولوگ زیر یک مسند P را تعریف می‌کند:**

```
P(X,[X|Y])
P(X,[Y|Z]) :- P(X,Z)
```

**ف) برای سوالات P(2,[1,A,3]) و P(A,[1,2,3]) درخت اثبات و راه‌حل‌ها را نمایش دهید. ب) کدام عملگر لیست استاندارد، ای بازنمایی P استفاده می‌شود؟** ***(این تمرین مشابه تمرین ۱۳.۹ در ویرایش دوم است)***

حل: **الف)** در اینجا، هر خط جلو آمده یک مرحله عمیق‌تر در درخت اثبات را نشان می‌دهد و دو خطی که یکسان جلو ده‌اند، دو روش برای اثبات هدفی هستند که جلو نیامده است. $P_1$ و $P_2$ که در یک خط بکار رفته‌اند، به ترتیب بند اول و بند وم از P را نشان می‌دهند:

```
P(A, [1,2,3])                goal
  P(1, [1|2,3])              P1 => solution, with A = 1
  P(1, [1|2,3])              P2
    P(2, [2,3])              P1 => solution, with A = 2
    P(2, [2,3])              P2
      P(3, [3])              P1 => solution, with A = 3
      P(3, [3])              P2

P(2, [1, A, 3])              goal
  P(2, [1|2, 3])             P1
  P(2, [1|2, 3])             P2
    P(2, [2|3])              P1 => solution, with A = 2
    P(2, [2|3])              P2
      P(2, [3])              P1
      P(2, [3])              P2
```

ب) بهتر است P را Member بنامیم زیرا در زمانیکه اولین آرگومان یک عنصر از لیستی باشد که آرگومان دوم است، نتیجه ده و موفق می‌شود.

**؟ ۱۶.۹ در این تمرین به مرتب‌سازی در پرولوگ می‌پردازیم. الف) بندهایی به زبان پرولوگ بنویسید که مسند sorted(L) را ریف کند به طوری که اگر و فقط اگر لیست L به صورت صعودی مرتب باشد، true را برگرداند. ب) مسند perm(L,M) را به ان پرولوگ تعریف کنید به طوری که گر L یک جایگشت از M باشد، true را برگرداند. ج) sort(L,M) را با استفاده از per و sorted تعریف کنید (M یک نقطهٔ مرتب شده از L است). د) Sort را برروی لیست‌هایی بزرگ و بزرگتر آنقدر را اجرا ید که خسته شوید. پیچیدگی زمانی برنامهٔ شما چقدر است؟ هـ) الگوریتم مرتب‌سازی سریع‌تری مانند مرتب‌سازی درجی با تب‌سازی سریع را به زبان پرولوگ بنویسید.** ***(این تمرین مشابه تمرین ۱۴.۹ در ویرایش دوم است)***

حل: نسخه‌های متفاوت sort می‌توانند تفاوت بین مفاهیم منطقی و تابعی در پرولوگ را نشان دهند:

**ف)**

```
sorted([]).
sorted([X]).
sorted([X,Y|L]) :- X<Y, sorted([Y|L]).
```

**(**

```
perm([],[]).
perm([X|L],M) :-
  delete(X,M,M1),
  perm(L,M1).

delete(X,[X|L],L).          %% deleting an X from [X|L] yields L
delete(X,[Y|L],[Y|M]) :- delete(X,L,M).

member(X,[X|L]).
member(X,[_|L]) :- member(X,L).

sort(L,M) :- perm(L,M), sorted(M).
```

ی شکل از مرتب‌سازی، همان شکل مشخص و قابل اجرایی است که می‌توان تعریف کرد. حداقل تعریف مفیدی که از تب‌سازی می‌توان ارائه داد، آن است که برای آن که M مرتب شده‌ی L باشد، بایستی M همان عناصر L را داشته و همگی ترتیب چیده شوند. **د)** متاسفانه نمی‌توان یک برنامه مشخص داشت که این کار را انجام دهد و باید به شیوه تولید/ بررسی این را انجام دهیم: perm در هر زمان یک جایگشت مناسب را تولید می‌کند و sorted آنها را بررسی می‌کند. در بدترین حالت انی که فقط یک جایگشت مرتب موجود باشد و آن آخرین مورد تولیدی باشد)، دارای پیچیدگی $O(n!)$ برای تولید خواهد

بود. از آنجا که هر perm دارای پیچیدگی $O(n^2)$ و هر sorted دارای پیچیدگی $O(n)$ می‌باشد، کل sort در بدترین حالت پیچیدگی $O(n!\ n^2)$ خواهد داشت. هـ) در این جا یک مرتب‌سازی درجی با پیچیدگی $O(n^2)$ را مشاهده می‌کنید:

```
isort([],[]).
isort([X|L],M) :- isort(L,M1), insert(X,M1,M).

insert(X,[],[X]).
insert(X,[Y|L],[X,Y|L]) :- X=<Y.
insert(X,[Y|L],[Y|M]) :- Y<X, insert(X,L,M).
```

**؟ ۱۷.۹ در این تمرین به کاربرد بازگشتی قواعد بازنویسی‌شده با استفاده از برنامه‌نویسی منطقی می‌پردازیم. یک قاعدهٔ بازنویسی‌شده (یا مدولاسیون معکوس در اصطلاحات OTTER) معادله‌ای با جهت مشخص می‌باشد. به عنوان مثال، قاعدهٔ بازنویسی‌شده $x+0 \rightarrow x$ پیشنهاد می‌دهد که هر عبارت منطبق با $x+0$ را با عبارت $x$ جایگزین کنید. کاربرد قواعد بازنویسی شده، مرکزی‌ترین بخش سیستم استنتاج می‌باشد. برای نمایش قواعد بازنویسی‌شده از مسند *rewrite*(x,y) استفاده می‌کنیم. به عنوان مثال، همان قاعدهٔ فوق را می‌توان به صورت *rewrite*(x+0,x) بیان کرد. برخی اصطلاحات اصلی بوده و نمی‌توانند ساده‌تر شوند بنابراین *primitive*(0) بیان می‌کند که 0 یک اصطلاح اصلی است. الف) یک تعریف از مسند *simplify*(x,y) بیان کنید که درصورتی که y نسخه ساده شده‌ای از x باشد، true را برگرداند. (دیگر نتوان هیچ قاعدهٔ بازنویسی شده‌ای را به هر زیر بخش از y اعمال نمود). ب) مجموع قواعدی برای ساده‌سازی عبارات حاوی عملگرهای محاسباتی بنویسید و سپس الگوریتم ساده‌سازی خود را به برخی عبارات به عنوان نمونه اعمال کنید. ج) مجموعه‌ای از قواعد بازنویسی‌شده برای مشتق‌گیری نمادین نوشته و سپس با استفاده از آنها و قواعد ساده‌سازی خود از عبارات محاسباتی و نمایی مشتق گرفته و آنها را ساده کنید. *(این تمرین مشابه تمرین ۱۵.۹ در ویرایش دوم است)***

☑ حل: این تمرین قدرت تطبیق الگو که در پرولوگ گنجانده شده است را نشان می‌دهد.

**الف)** که ساده‌سازی به نظر ساده می‌رسد ولی دانشجویان ممکن است در انتخاب بین ساده‌سازی و چرخیدن بی‌نهایت گیر کنند:

```
simplify(X,X) :- primitive(X).
simplify(X,Y) :- evaluable(X), Y is X.
simplify(Op(X)) :- simplify(X,X1), simplify_exp(Op(X1)).
simplify(Op(X,Y)) :- simplify(X,X1), simplify(Y,Y1), simplify_exp(Op(X1,Y1)).

simplify_exp(X,Y) :- rewrite(X,X1), simplify(X1,Y).
simplify_exp(X,X).

primitive(X) :- atom(X).
```

**ب)** چند قاعدهٔ بازنویسی‌شده از لیستی که Norving در سال 1992 ارائه داد را در زیر می‌بینید:

```
Rewrite(X+0,X).
Rewrite(0+X,X).
Rewrite(X+X,2*X).
Rewrite(X*X,X^2).
Rewrite(X^0,1).
Rewrite(0^X,0).
Rewrite(X*N,N*X) :- number(N).
Rewrite(ln(e^X),X).
Rewrite(X^Y*X^Z,X^(Y+Z)).
Rewrite(sin(X)^2+cos(X)^2,1).
```

**ج)** در زیرقواعدی جهت مشتق‌گیری مشاهده می‌کنید که d(X,Y) نشان‌دهندهٔ مشتق عبارت Y نسبت به متغیر X می‌باشد:

```
Rewrite(d(X,X),1).
Rewrite(d(U,X),0) :- atom(U), U /= X.
Rewrite(d(U+V,X),d(U,X)+d(V,X)).
Rewrite(d(U-V,X),d(U,X)-d(V,X)).
Rewrite(d(U*V,X),V*d(U,X)+U*d(V,X)).
Rewrite(d(U/V,X),(V*d(U,X)-U*d(V,X))/(V^2)).
Rewrite(d(U^N,X),N*U^(N-1)*d(U,X)) :- number(N).
Rewrite(d(log(U),X),d(U,X)/U).
Rewrite(d(sin(U),X),cos(U)*d(U,X)).
Rewrite(d(cos(U),X),-sin(U)*d(U,X)).
Rewrite(d(e^U,X),d(U,X)*e^U).
```

**؟ ۱۸.۹ در این تمرین پیاده‌سازی الگوریتم‌های جستجو در زبان پرولوگ را بررسی می‌کنیم. فرض کنید *successor*(X,Y) زمانی درست است که حالت Y یک پسین حالت X باشد و *goal*(X) زمانی درست است که x یک حالت هدف باشد. تعریفی برای *solve*(X,P) ارائه دهیم که درآن P یک مسیر (لیستی از حالات) با شروع از X بوده و به یک حالت هدف ختم شود و نباله مراحل صحیح سپری شده توسط *succersor* مشخص می‌شود. روش جستجوی اول‌عمق، ساده‌ترین راه برای انجام این کار خواهد بود. چگونه می‌توان به سادگی کنترل اکتشافی را به جستجو اضافه کنیم؟** ***(این تمرین مشابه تمرین ۱۶.۹ در ویرایش دوم است)***

☑ حل: اگر از کارکرد پرولوگ آگاه باشید پاسخ بسیار ساده خواهد بود:

```
solve(X,[X]) :- goal(X).
solve(X,[X|P]) :- successor(X,Y), solve(Y,P).
```

ی‌توان این روش را به زبان ساده این گونه بیان نمود: «از حالت اولیه شروع به حرکت می‌کنیم. اگر همان یک حالت هدف بود گاه مسیر شامل فقط یک حالت اولیه، راه‌حل مساله می‌باشد. درغیر اینصورت، پسینی می‌یابیم که یک مسیر از آن پسین به دف وجود داشته باشد. آنگاه یک راه‌حل عبارتست از شروع از حالت اولیه و ادامهٔ راه توسط آن مسیر». دقت داشته باشید که solv نمی‌تواند به تنهایی مسیر P که راه‌حل است را بیابد و فقط می‌تواند بگوید که یک مسیر داده شده راه‌حل هست یا خیر. ر می‌خواهید تابع اکتشافی (یا حتی جستجوی اول سطح) را اضافه کنید، به یک صف مشخص نیاز دارید. الگوریتم این کار شابه نسخه‌ای از کد به زبان لیسپ، پایتون و یا جاوا که در کتاب ذکر شده است می‌باشد.

**؟ ۱۹.۹ فرض کنید پایگاه دانشی شامل فقط بندهای هورن مرتبه اول زیر باشد:**

**Ancestor(Mother(x),x)**

**Ancestor(x,y)∧Ancestor(y,z)⟹Ancestor(x,z)**

**بق الگوریتم زنجیره‌ای پیش‌رو، در مرحلهٔ iامین تکرار در صورتی کار را خاتمه میدهیم که KB شامل جمله ای باشد که با هر ؤالی یکسان‌سازی شود و در غیر اینصورت باید هر جملهٔ بسیطی که از جملات موجود در پایگاه استنتاج می‌شود را در مرحلهٔ i+j به آن اضافه کنیم. الف) برای هر یک از سؤالات زیر با گذاشتن عدد (۱) نشان دهید که این الگوریتم می‌تواند یک پاسخ برای ن بیابد(در صورت وجود پاسخ، پاسخ را نیز بنویسید). و با گذاشتن عدد (۲) نشان دهید که الگوریتم خاتمه یافته ولی پاسخی ی‌یابد. و با گذاشتن عدد (۳) نشان دهید که هرگز خاتمه نمی‌یابد.**

**Ancestor(Mother(y),John)**

**Ancestor(Mother(Mother(y)),John) (**

**Ancestor(Mother(Mother(Mother(y))),Mother(y)) (i**

**Ancestor(Mother(john),Mother(Mother(john))) (i**

**ب) آیا الگوریتم رزولوشن می‌تواند جملهٔ ¬Ancestor(john,john) را از پایگاه دانش اصلی اثبات کند؟ در صورت مثبت بودن سخ چگونگی آن را توضیح داده و در غیر اینصورت بگویید چرا ممکن نیست. ج) فرض کنید جملهٔ ¬(Mother(x)=x) را به ملات قبل اضافه کرده و از الگوریتم رزولوشن برای هم‌ارزی استفاده می‌کنیم. اکنون پاسخ قسمت ب چه خواهد بود.**

☑ حل: **الف)** نتایج حاصل از زنجیره‌ای پیش‌رو: (i) بله. {y/john} (در اولین مرحله). (ii) بله. {y/john} (پس از دو مرحله کرار). (iii) بله. { } (پس از دو مرحله تکرار). (iv) خاتمه نمی‌یابد. **ب)** با این که الگوریتم رزولوشن کامل است ولی نمی‌تواند ن جمله را اثبات کند زیرا این جمله از آن موارد نتیجه نمی‌شود و هیچ قاعده‌ای نداریم که امکان جد بودن هر کسی با هر کس بگر را بیان کند. **ج)** همان پاسخ قسمت قبل

**؟ ۲۰.۹ فرض کنید C زبان مرتبهٔ اول با یک مسند S(p,q) باشد بدان معنا که«p ریش q را اصلاح می‌کند». گروه مردمی زیر را ر نظر بگیرید. الف) جملهٔ «یک شخص P وجود دارد که هر کسی که خودش را اصلاح نمی‌کند، اصلاح می‌کند» را با این زبان ان کنید. ب) جملهٔ قسمت الف را به شکل بندی بنویسید. ج) یک اثبات رزولوشن برای نشان دادن اینکه بندهای قسمت ب تا ناسازگار هستند، ارائه دهید. (توجه: نیاز به تعریف قواعد اضافه ندارید).**

☑ حل: **الف)** $\exists p \ \forall q \ \ S(p,q) \ \Leftrightarrow \ \neg S(q,q)$

**ب)** برای هر جهت از استلزام یک بند و در نتیجه در کل دو بند داریم: $\text{C1: } \neg S(Sk1,q) \vee \neg S(q,q)$

$\text{C2: } S(Sk1,q) \vee S(q,q)$

**ج)** با اعمال فاکتورگیری به $C_1$ و استفاده از جایگزینی q/SK1 داریم C3: $\neg S(Sk1,Sk1)$

با اعمال فاکتورگیری به $C_2$ و استفاده از جایگزینی q/SK1 داریم C4: $S(Sk1,Sk1)$

ل کردن $c_3$ توسط $C_4$، بندی تهی را ایجاد میکند.

**؟ ۲۱.۹ چگونه می‌توان از رزولوشن جهت تعیین اعتبار یک جمله استفاده کرد. جهت تعیین ارضاپذیرنبودن چطور؟** ***(این تمرین مشابه تمرین ۱۷.۹ در ویرایش دوم است)***

☑ حل: این تمرین به فهم دانشجویان از رزولوشن و توانایی آنها در تفکر در سطوح بالاتر روابط مجموعه‌ای می‌پردازد. یادآوری ی‌کنیم که رزولوشن برای اثبات $KB \models \alpha$ اثبات می‌کند که $KB \wedge \neg\alpha$ نادرست است (مشابه برهان خلف). معمولا در سیستم رولوشن سوال ASK(KB,α) استفاده می‌شود. اکنون می‌خواهیم نشان دهیم که جمله‌ای نظیر β معتبر یا غیرارضاپذیر هست

یا خیر. جملهٔ $\beta$ معتبر است اگر بتوانیم بدون هیچ اطلاعات اضافه‌ای آن را نمایش دهیم. این کار را با پرسیدن سوال $ASK(KB_0,\beta)$ انجام می‌دهیم که درآن $KB_0$ پایگاه دانش تهی می‌باشد. جملهٔ $\beta$ که غیرارضاپذیر است با خودش ناسازگار است. بنابراین اگر پایگاه دانش را مجددا تهی کرده و بپرسیم $ASK(KB_0,\neg\beta)$، آنگاه سیستم رزولوشن نقیض آن یعنی $\neg\neg\beta$ را در نظر می‌گیرد. اگر بتواند این کار را انجام دهد آنگاه $\neg\neg\beta$ ممکن شده و $\beta$ ناسازگار است.

**؟ ۲۲.۹ مثالی از دو بند ارائه دهید که حل آنها به دو روش مختلف باعث ایجاد دو نتیجهٔ متفاوت شود.**

☑ حل: دو راه برای این کار وجود دارد: یک لفظ در یک بند وجود داشته باشد که با دو لفظ مختلف در بند دیگر مکمل باشد. مانند:

$$P(x) \quad \neg P(a) \vee \neg P(b)$$

یا دو جفت مکمل از الفاظ مانند:

$$P(x) \vee Q(x) \quad \neg P(a) \vee \neg Q(b)$$

البته در منطق گزاره‌ای کار نخواهد کرد: در اولین مثال، دو لفظ مربوط در بند دوم باید یکتا باشند و در دومین مثال سایر جفتهای باقیمانده که حل نشده‌اند، پس از رزولوشن ممکن است نتیجه‌ای زائد داشته باشند.

**؟ ۲۳.۹ طبق جملهٔ «اسب‌ها حیوان هستند» می‌توان نتیجه گرفت که «سریک اسب، سریک حیوان است». نشان دهید که این استنتاج توسط مراحل زیر معتبر است. الف) مقدم و تالی را به زبان منطق مرتبهٔ اول ترجمه کنید. از سه مسند زیر استفاده کنید: *Horse(x)* و *Animal(x)*، *HeadOf(h,x)* (*h* سر *x* است).**

**ب) بخش تالی را نقیض کرده و سپس مقدم و تالی نقیض شده را به شکل نرمال عطفی (CNF) تبدیل کنید. ج) با استفاده از رزولوشن نشان دهید که تالی از مقدم نتیجه می‌شود. *(این تمرین مشابه تمرین ۱۸.۹ در ویرایش دوم است)***

☑ حل: این شکلی از استنتاج است که برای اثبات آن که «قیاس ارسطو نمی‌تواند درهمهٔ استنتاجات دقیق باشد» استفاده شد:

**الف)**

$$\forall x \ \ Horse(x) \Rightarrow Animal(x)$$
$$\forall x,h \ \ Horse(x) \wedge HeadOf(h,x) \Rightarrow \exists y \ \ Animal(y) \wedge HeadOf(h,y)$$

**ب)**

$$A.\ \neg Horse(x) \vee Animal(x)$$
$$B.\ Horse(G)$$
$$C.\ HeadOf(H,G)$$
$$D.\ \neg Animal(y) \vee \neg HeadOf(H,y)$$

(در اینجا $A$ از اولین جمله قسمت الف و سایرین از دومین جمله می‌آیند. $G$ و $H$ ثابت‌های اسکولم هستند.)

**ج)** $C$ و $D$ را حل کنید تا به $\neg Animal(G)$ دست یابید. سپس آن را با $A$ حل کنید تا به $\neg Horse(G)$ برسید. سپس آن را با B حل کنید تا به نتیجهٔ دلخواه برسید.

**؟ ۲۴.۹ در اینجا دو جمله به زبان منطق مرتبهٔ اول را مشاهده می‌کنید. (A) $\forall x\ \exists y\ (x \geq y)$ (B) $\exists y\ \forall x\ (x \geq y)$ الف) فرض کنید محدودهٔ متغیرها اعداد طبیعی $0,1,2,\ldots,\infty$ و بوده ومسند «≥» به معنای بزرگتر و مساوی بودن باشد. طبق این جملات (A)، (B) را به جملات صریح تبدیل کنید. ب) آیا طبق این تفسیر (A) صحیح است؟ ج) آیا طبق این تفسیر (B) صحیح است؟ د) آیا (A) به طور منطقی (B) را ایجاب می‌کند؟ هـ) آیا (B) به طور منطقی (A) را ایجاب می‌کند؟ و) با استفاده از رزولوشن اثبات کنید که (A) از (B) نتیجه می‌شود، حتی اگر فکر می‌کنید که (B) به طور منطقی (A) را ایجاب نمی‌کند، باز هم اثبات را ادامه دهید تا با شکست مواجه شوید. برای هر مرحله از رزولوشن، جایگزینی یکسان‌سازی را نشان دهید. اگر اثبات با شکست مواجه شده است به طور دقیق توضیح دهید که کجا، چرا و چگونه به شکست مواجه شده است. ز) اکنون اثبات کنید که (B) از (A) نتیجه می‌شود. *(این تمرین مشابه تمرین ۱۹.۹ در ویرایش دوم است)***

☑ حل: این تمرین فهم دانشجویان از مدل و ایجاب را نشان می‌دهد. **الف)** ترجمهٔ (A) به این صورت است: «برای هر عدد طبیعی، عدد طبیعی دیگری وجود دارد که کوچکتر و مساوی با آن باشد.» ترجمهٔ (B) به این صورت است: «یک عدد طبیعی خاص وجود دارد که کوچکتر و مساوی با هر عدد طبیعی دیگر است.» **ب)** بله. عبارت (A) تحت این تفسیر صحیح است. شما می‌توانید همواره خود عدد را به عنوان آن عدد دیگر در جمله در نظر بگیرید. **ج)** بله. عبارت (B) تحت این تفسیر صحیح است عدد صفر را به عنوان آن عدد طبیعی خاص در نظر بگیرید. **د)** خیر. عبارت (A) نمی‌تواند به طور منطقی (B) را ایجاب کند. **هـ)** بله. عبارت (B) به طور منطقی (A) را ایجاب می‌کند. **و)** می‌خواهیم با استفاده از رزولوشن اثبات کنیم که عبارت (A)، (B) را ایجاب می‌کند. برای این منظور، پایگاه دانشی شامل عبارت (A) و نقیض (B) را در نظر می‌گیریم که آن را با (-B) نمایش می‌دهیم. در ابتدا باید (-B) و (A) را به صورت متعارف تبدیل کنیم. برای (-B) باید نماد $\neg$ را در پشت دو سور قرار دهیم و برای هر دو جمله نیاز به تعریف یک تابع اسکولم داریم:

$$(A)\ \ x \geq F_1(x)$$

$$(-B)\ \neg F_2(y) \geq y$$

ن سعی می‌کنیم هر دوی آنها را با هم حل کنیم ولی باید از یکسان‌سازی استفاده شود. به نظر می‌رسد که جایگزینی باید به
ت $\{x/F_2(y), y/F_1(F_2(y)\}$ باشد ولی این معادل است با $\{x/F_2(y),y /F_1(x)\}$ که با شکست مواجه می‌شود زیرا $y$
در عبارات شامل $y$ کاربرد دارد. بنابراین رزولوشن شکست می‌خورد و هیچ مرحله دیگری از رزولوشن برای امتحان کردن
نمانده است، پس (B) از عبارت (A) نتیجه نمی‌شود. ز) برای اثبات آن که عبارت (B)، (A) را ایجاب می‌کند با پایگاه
ی شامل (B) و نقیض (A) شروع می‌کنیم که آن را با (A-) نمایش می‌دهیم.

(A) $\neg F_1 \geq y$

(-B) $x \geq F_2(x)$

مال رزولوشن و جایگزینی $\{x/F_1,y/F_2(F_1)\}$ به $False$ می‌رسیم و اثبات می‌شود که (B)، عبارت (A) را ایجاب می‌کند.
**۲۵.۹ رزولوشن می‌تواند برای سوالاتی حاوی متغیر، اثبات‌های بی‌فایده تولید کند. بنابراین باید مکانیزمی معرفی شود تا
خ‌های متناهی استخراج کند. توضیح دهید که چرا این نتیجه در پایگاه دانش‌هایی شامل فقط بندهای متناهی، رخ نمی‌دهد.
*ی تمرین مشابه تمرین ۲۰.۹ در ویرایش دوم است)***

حل: یکی از این روش‌ها آن است که به رزولوشن اجازه دهیم از روی موارد استنتاج کند که دراینصورت می‌توان از روی
ت بودن $B$ و $A$ مورد $C$ را اثبات نمود بدون آن که هر یک از آنها را بشناسیم. در پایگاه دانش‌هایی که فقط بند متناهی
د، تنها با یک زنجیره واحد از استنتاجات روبرو هستیم که می‌توان زنجیره را دنبال نموده و به متغیرها نمونه داد.

**۲۶.۹ در این فصل گفته شد که رزولوشن نمی‌تواند برای تولید تمام نتایج منطقی حاصل از یک مجموعه جملات به کار رود. آیا
ریتم دیگری وجود دارد که این کار را انجام دهد. *(این تمرین مشابه تمرین ۲۱.۹ در ویرایش دوم است)***

حل: خیر. یکی از بخش‌های تعریف الگوریتم، پایان‌پذیربودن آن است. از آنجا که برای یک مجموعه جملات، می‌توانند
نهایت نتیجه منطقی وجود داشته باشد، هیچ الگوریتمی قادر به تولید تمامی آنها نیست. یکی دیگر از دلایل موجود برای
ن چنین الگوریتمی آن است که ایجاب برای FOL تصمیم‌ناپذیر است. اگر الگوریتمی وجود داشته باشد که بتواند تمام
ج منطقی حاصل از مجموعه جملات $S$ را بیابد، آنگاه اگر بخواهیم بدانیم که آیا $B$ توسط $S$ ایجاب می‌شود یا خیر، یک راه
ست که بررسی کنیم B در مجموعه تولیدی هست یا خیر. ولی می‌دانیم که این کار غیرممکن است و تولید ایسن مجموعه
ممکن است.

# فصل ۱۰ (ویرایش سوم)

**؟ ۱.۱۰ تفاوت‌ها و شباهت‌های بین حل مسأله و برنامه‌ریزی را بیان کنید. *(این تمرین مشابه تمرین ۱.۱۱ در ویرایش دوم است)***

☑ حل: حل‌کننده مسأله و برنامه‌ریز، هر دو از نقطه‌ای شروع کرده و با استفاده از مجموعه عملگرها و اقدامات تعریف شده، قصد رسیدن به هدف را دارند که معمولا آنها را در محیط‌های قطعی، گسسته و مشاهده‌پذیر بکار می‌بریم. در برنامه‌ریزی، بازنمایی حالات، اهداف و برنامه را داریم که موجب می‌شود انواع الگوریتم‌ها بتوانند فضای جستجو را با روش‌های پس‌رو یا پیش‌رو جستجو کرده و همچنین از تولید خودکار تابع هیورستیک بهره ببرند.

**؟ ۲.۱۰ نمودار واکنش و حالت شروع که در شکل ۱.۱۰ ترسیم شده است، موجود است. اگر حالت به صورت زیر توصیف شود، تمام نمونه‌های کاربردی واقعی از *Fly(p,from,to)* را بیابید.**

$$At(P_1,JFK) \wedge At(P_2,SFO) \wedge Plane(P_1) \wedge Plane(P_2) \wedge Airport(JFK) \wedge Airport(SFO)?$$

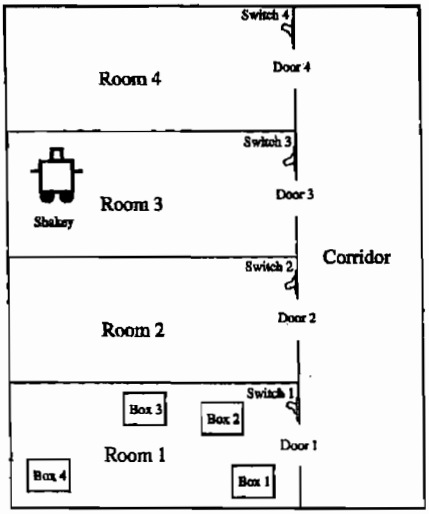
***(این تمرین مشابه تمرین ۲.۱۱ در ویرایش دوم است)***

☑ حل: منظور از کلمهٔ «نمونه کاربردی» آن است که نمونه تولیدی پیش‌شرط‌ها را ارضا کند و قابل اجرا باشد و منظور از «نمونه واقعی» آن است که متغیرهای آن توسط ثوابت جایگزین شده باشند. واکنش‌های کاربردی عبارتند از:

$$Fly(P_1, JFK, SFO)$$
$$Fly(P_1, JFK, JFK)$$
$$Fly(P_2, SFO, JFK)$$
$$Fly(P_2, SFO, SFO)$$

یک نکتهٔ ظریف در این مبحث قرار دارد و آن «پرواز به هیچ کجا» است یعنی از یک فرودگاه به خودش، که طبق تعریف *Fly* مجاز بوده و قابل‌اجرا است (اگر چه بی فایده است).

**؟ ۳.۱۰ مسأله میمون و موزها بدین صورت است که یک میمون در یک آزمایشگاه با تعدادی موز که از سقفی بلند آویخته شده است قرار می‌گیرد. یک جعبه در آنجا قرار دارد که میمون با بالا رفتن از آن، می‌تواند به موزها دسترسی یابد. در ابتدا میمون در نقطه A، موزها در B و جعبه در C قرار دارد. قد میمون و جعبه هر دو کوتاه است ولی با رفتن میمون بر روی جعبه، مجموع قد آنها به قدری بلند می‌شود که به اندازهٔ موزها برسد. اقداماتی که میمون می‌تواند انجام دهد شامل حرکت از یک مکان به مکان دیگر(*Go*)، هل دادن یک شی از یک مکان به مکان دیگر (*Push*)، بالا رفتن (*ClimbUp*) و پایین آمدن از یک شی(*ClimbD*)، گرفتن(*Grasp*) و یا رها کردن (*Ungrasp*) یک شی، می‌باشد. اگر میمون و شی‌ای که می‌گیرد در یک ارتفاع قرار داشته باشند، میمون پس از گرفتن آن شی، آن را نگه می‌دارد. الف) توصیفی برای حالت شروع بنویسید. ب) شش واکنش فوق را بطور شماتیک (طرح‌واره) بیان کنید.**




شکل ۱۴.۱۰ دنیای Shakey.
این روبات می‌تواند درون یک اتاق مابین خطوط مشخص شده بر زمین حرکت کند، می‌تواند از در مابین دو اتاق گذر نماید، همچنین از اشیایی که قابل بالا رفتن هستند بالا رفته و اشیای قابل هل دادن را هل دهد. در ضمن قادر است دکمه چراغ را بزند.

**ج) فرض کنید میمون قصد فریب دانشمندان را دارد یعنی زمانی که آنها در حال چای خوردن و استراحت هستند بدون آنکه جعبه را در جای اصلی‌اش بگذارد، موزها را چنگ انداخته و برمی‌دارد، این موضوع را به عنوان یک هدف کلی (یعنی لزوما جعبه**

در محل C نباشد) به زبان محاسبات موقعیت بیان کنید. آیا این هدف می‌تواند توسط یک سیستم کلاسیک برنامه‌ریزی حل شود؟ د) احتمالا طرحی که شما از اقدام هل دادن دارید، نادرست است. زیرا اگر شیئی خیلی سنگین داشته باشیم، با اعمال اقدام ***Push*** هیچ تأثیری در موقعیت آن حاصل نمی‌شود. طرح خود را از اقدامات به گونه‌ای اصلاح کنید که اشیای سنگین را نیز در بربگیرد. *(این تمرین مشابه تمرین ۴.۱۱ در ویرایش دوم است)*

☑ حل: این مساله مشابه مساله Shakey (۳.۱۱) می‌باشد. بنابراین تنها کافیست این دو مساله را با هم تطبیق دهید تا به پاسخ برسید. **الف)** حالت شروع عبارتست از:

$$At(Monkey, A) \wedge At(Bananas, B) \wedge At(Box, C) \wedge$$
$$Height(Monkey, Low) \wedge Height(Box, Low) \wedge Height(Bananas, High) \wedge$$
$$Pushable(Box) \wedge Climbable(Box)$$

**ب)** اقدامات عبارتند از:

$Action($ACTION:$Go(x, y)$, PRECOND:$At(Monkey, x)$,
EFFECT:$At(Monkey, y) \wedge \neg(At(Monkey, x)))$
$Action($ACTION:$Push(b, x, y)$, PRECOND:$At(Monkey, x) \wedge Pushable(b)$,
EFFECT:$At(b, y) \wedge At(Monkey, y) \wedge \neg At(b, x) \wedge \neg At(Monkey, x))$
$Action($ACTION:$ClimbUp(b)$,
PRECOND:$At(Monkey, x) \wedge At(b, x) \wedge Climbable(b)$,
EFFECT:$On(Monkey, b) \wedge \neg Height(Monkey, High))$
$Action($ACTION:$Grasp(b)$,
PRECOND:$Height(Monkey, h) \wedge Height(b, h)$
$\wedge At(Monkey, x) \wedge At(b, x)$,
EFFECT:$Have(Monkey, b))$
$Action($ACTION:$ClimbDown(b)$,
PRECOND:$On(Monkey, b) \wedge Height(Monkey, High)$,
EFFECT:$\neg On(Monkey, b) \wedge \neg Height(Monkey, High)$
$\wedge Height(Monkey, Low)$
$Action($ACTION:$UnGrasp(b)$, PRECOND:$Have(Monkey, b)$,
EFFECT:$\neg Have(Monkey, b))$

**ج)** طبق محاسبات موقعیت، هدف در یک حالت $s$ قرار دارد. نظیر:

$$Have(Monkey, Bananas, s) \wedge (\exists x \;\; At(Box, x, s_0) \wedge At(Box, x, s))$$

در STRIPS ناچاریم که فقط در مورد حالت هدف صحبت کرده و راهی برای بازنمایی حقایق مرتبط بین دو حالت درونی برنامه نداریم. (مانند هم‌ارزی‌های مکانی یک شی). بنابراین راهی برای بازنمایی این هدف نداریم. **د)** در واقع باید پیش شرط *Pushable* را به راه‌حل فوق بیافزاییم.

**؟ ۴.۱۰ برنامه اصلی STRIPS برای کنترل روبات Shakey طراحی شد. شکل ۴.۱۰ نسخه‌ای از دنیای Shakey را نشان می‌دهد که شامل چهار درب بوده که با یک راهرو به هم متصل هستند. هر اتاق دارای یک درب و یک کلید لامپ می‌باشد. اقداماتی که در دنیای Shakey مسیر است شامل حرکت از یک مکان به مکان دیگر، هل دادن اشیای قابل حرکت (مانند جعبه)، بالا رفتن و پایین آمدن از روی اشیای سخت و محکم (مانند جعبه) و روشن و خاموش کردن لامپ با کلید آن می‌شود. روبات به تنهایی قادر به بالا رفتن از جعبه یا زدن کلید نبوده ولی برنامه‌ریز STRIPS که تواناتر از روبات بود می‌توانست برنامه‌ها را یافته و چاپ کند. شش اقدام Shakey عبارتند از: • *Go(x,y)*: یعنی Shakey در موقعیت $x$ بوده و به محل $y$ می‌رود که هم $x$ و هم $y$ در یک اتاق واقع هستند. طبق قرارداد، اگر دری بین دو اتاق وجود داشته باشد بایستی آنرا در هر دو اتاق مشاهده کنیم. • *Push(b,x,y)*: به معنای هل دادن جعبه $b$ از محل $x$ به محل $y$ در همان اتاق می‌باشد. در اینجا به مسند *Box* و ثابت‌هایی برای جعبه‌ها نیازمندیم. • *ClimbUp(b)*: به معنای بالا رفتن از جعبهٔ $b$ است. • *ClimbDown(b)*: به معنای پایین آمدن از جعبهٔ $b$ است، در اینجا نیز به مسند *On* و ثابت *Floor* نیازمندیم. • *TurnOn(s)*: به معنای روشن کردن لامپ $s$ با زدن کلید آن است. • *TurnOff(s)*: به معنای خاموش کردن لامپ $s$ با زدن کلید آن است.برای خاموش یا روشن کردن لامپ، روبات Sakey باید در همان موقعیت کلید، جعبه‌ای گذاشته و بر روی آن قرار بگیرد. شش اقدام Shakey و حالت شروع شکل ۱۴.۱۰ را با جملات PDDL توصیف کنید. برنامه‌ای طراحی کنید که روبات Shakey، جعبهٔ *Box2* را به اتاق *Room2* ببرد. *(این تمرین مشابه تمرین ۱۳.۱۱ در ویرایش دوم است)***

☑ حل: اقدامات این مساله بسیار شبیه به مساله میمون و موز است. اقدامات عبارتند از:

$Action(\text{ACTION}:Go(x,y), \text{PRECOND}:At(Shakey,x) \wedge In(x,r) \wedge In(y,r),$
$\quad \text{EFFECT}:At(Shakey,y) \wedge \neg(At(Shakey,x)))$
$Action(\text{ACTION}:Push(b,x,y), \text{PRECOND}:At(Shakey,x) \wedge Pushable(b),$
$\quad \text{EFFECT}:At(b,y) \wedge At(Shakey,y) \wedge \neg At(b,x) \wedge \neg At(Shakey,x))$
$Action(\text{ACTION}:ClimbUp(b), \text{PRECOND}:At(Shakey,x) \wedge At(b,x) \wedge Climbable(b),$
$\quad \text{EFFECT}:On(Shakey,b) \wedge \neg On(Shakey,Floor))$
$Action(\text{ACTION}:ClimbDown(b), \text{PRECOND}:On(Shakey,b),$
$\quad \text{EFFECT}:On(Shakey,Floor) \wedge \neg On(Shakey,b))$
$Action(\text{ACTION}:TurnOn(l), \text{PRECOND}:On(Shakey,b) \wedge At(Shakey,x) \wedge At(l,x),$
$\quad \text{EFFECT}:TurnedOn(l))$
$Action(\text{ACTION}:TurnOff(l), \text{PRECOND}:On(Shakey,b) \wedge At(Shakey,x) \wedge At(l,x),$
$\quad \text{EFFECT}:\neg TurnedOn(l))$

• حالت شروع عبارتست از:

$In(Switch_1,Room_1) \wedge In(Door_1,Room_1) \wedge In(Door_1,Corridor)$
$In(Switch_1,Room_2) \wedge In(Door_2,Room_2) \wedge In(Door_2,Corridor)$
$In(Switch_1,Room_3) \wedge In(Door_3,Room_3) \wedge In(Door_3,Corridor)$
$In(Switch_1,Room_4) \wedge In(Door_4,Room_4) \wedge In(Door_4,Corridor)$
$In(Shakey,Room_3) \wedge At(Shakey,X_S)$
$In(Box_1,Room_1) \wedge In(Box_2,Room_1) \wedge In(Box_3,Room_1) \wedge In(Box_4,Room_1)$
$Climbable(Box_1) \wedge Climbable(Box_2) \wedge Climbable(Box_3) \wedge Climbable(Box_4)$
$Pushable(Box_1) \wedge Pushable(Box_2) \wedge Pushable(Box_3) \wedge Pushable(Box_4)$
$At(Box_1,X_1) \wedge At(Box_2,X_2) \wedge At(Box_3,X_3) \wedge At(Box_4,X_4)$
$TurnwdOn(Switch_1) \wedge TurnedOn(Switch_4)$

• برنامهٔ دسترسی به هدف عبارتست از:

$Go(X_S,Door_3)$
$Go(Door_3,Door_1)$
$Go(Door_1,X_2)$
$Push(Box_2,X_2,Door_1)$
$Push(Box_2,Door_1,Door_2)$
$Push(Box_2,Door_2,Switch_2)$

**؟ ۵.۱۰ یک ماشین تورینگ متناهی، دارای یک نوار یک بعدی متناهی از سلول‌هاست که هر سلول آن شامل یکی از نمادهای متناهی می‌شود. هر سلول در بالای خود یک هد برای خواندن و نوشتن دارد. مجموعه حالاتی که ماشین می‌تواند در آن قرار بگیرد متناهی بوده و هر یک از آنها حالاتی قابل‌قبول هستند، در هر مرحلهٔ زمانی، بسته به نماد موجود در سلول زیر هد و نیز حالت فعلی ماشین، یک مجموعه اقدام می‌تواند انتخاب شود. هر اقدام به معنای نوشتن یک نماد در سلول زیر هد، انتقال ماشین به یک حالت و حرکت هد به چپ و یا راست می‌باشد. این نگاشت مشخص می‌کند که کدام اقدامات برای برنامهٔ ماشین تورینگ مجاز است. هدف شما کنترل ماشین در حالتی قابل قبول است. برنامهٔ قابل قبول ماشین تورینگ را بعنوان یک مساله برنامه‌ریزی بازنمایی کنید. اگر موفق به انجام این کار شدید، به این موضوع پی خواهید برد که یافتن راه‌حل برای یک مساله برنامه‌ریزی حداقل به دشواری مساله قابل‌قبول تورینگ (PSPACE-HARD) است.**

☑ حل: در این پاسخ به یک بازنمایی می‌پردازیم. از مسندهای زیر استفاده شده است: • $HeadAt(c)$: هر نوار در موقعیت سلول $c$ قرار دارد که فقط برای یک سلول صحیح است. • $State(s)$: ماشین در حالت $s$ قرار دارد که فقط برای یک سلول صحیح است. • $ValueOf(c,v)$: مقدار سلول c، $v$ می‌باشد. • $LeftOf(c_1,c_2)$: سلول $c_1$ به اندازهٔ یک سلول در سمت چپ سلول $c_2$ قرار دارد. • $TransitionLeft(s_1,v_1,s_2,v_2)$: ماشین در حالت $s_1$ در حال خواندن سلولی با مقدار $v_1$ است که مقدار $v_2$ را در آن قرار داده، ماشین را به حالت $s_2$ برده و یک گام به سمت چپ منتقل می‌شود. • $TransitionRight(s_1,v_1,s_2,v_2)$: ماشین در حالت $s_1$ در حال خواندن سلولی با مقدار $v_1$ است که مقدار $v_2$ را در آن قرار داده، ماشین را به حالت $s_2$ برده و یک گام به سمت راست منتقل می‌شود. معنای مسندهای $HeadAt$، $State$ و $ValueOf$ از نام آنها پیداست و سایر نمادهای استفاده شده، اطلاعاتی در مورد ماشین و نوار مربوطه در اختیار ما می‌گذارند. دو اقدام مورد نیاز عبارتند از:

$Action(RunLeft(s_1, c_1, v_1, , s_2, c_2, v_2)$,
PRECOND: $State(s_1) \wedge HeadAt(c_1) \wedge ValueOf(c_1, v_1)$
; $\wedge TransitionLeft(s_1, v_1, s_2, v_2) \wedge LeftOf(c_2, c_1)$
EFFECT: $\neg State(s_1) \wedge State(s_2) \wedge \neg HeadAt(c_1) \wedge HeadAt(c_2)$
$\wedge \neg ValueOf(c_1, v_1) \wedge ValueOf(c_1, v_2))$

$Action(RunRight(s_1, c_1, v_1, , s_2, c_2, v_2)$,
PRECOND: $State(s_1) \wedge HeadAt(c_1) \wedge ValueOf(c_1, v_1)$
; $\wedge TransitionRight(s_1, v_1, s_2, v_2) \wedge LeftOf(c_1, c_2)$
EFFECT: $\neg State(s_1) \wedge State(s_2) \wedge \neg HeadAt(c_1) \wedge HeadAt(c_2)$
$\wedge \neg ValueOf(c_1, v_1) \wedge ValueOf(c_1, v_2))$

معمولا هدف از یک حالت پذیرفته شدهٔ مشخص، قابل دسترسی است. بعنوان مثال یک مساله ساده عبارتست از:

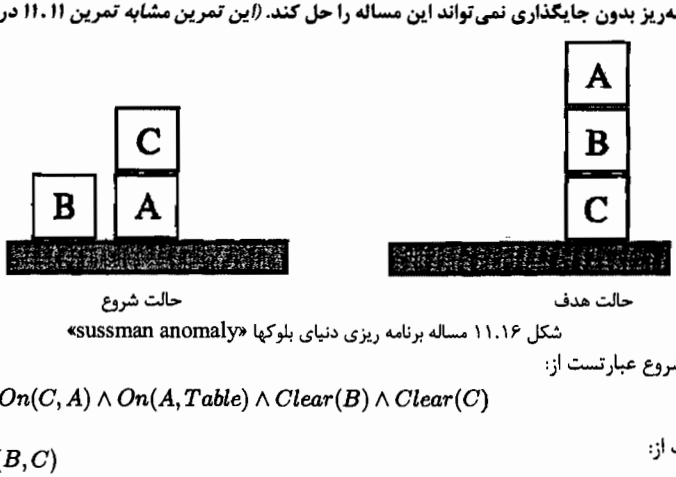
$Init(HeadAt(C_0) \wedge State(S_1) \wedge ValueOf(C_0, 1) \wedge ValueOf(C_1, 1)$
$\wedge ValueOf(C_2, 1) \wedge ValueOf(C_3, 0) \wedge LeftOf(C_0, C_1) \wedge LeftOf(C_1, C_2)$
$\wedge LeftOf(C_2, C_3) \wedge TransitionLeft(S_1, 1, S_1, 0) \wedge TransitionLeft(S_1, 0, S_{accept}, 0)$
$Goal(State(S_{accept}))$

لازم به ذکر است که تعداد الفاظ در یک حالت، بر حسب تعداد سلولها خطی است. بنابراین یک ماشین فضای چند جمله‌ای برای بازنمایی نیاز به حالتی چند جمله‌ای دارد.

**؟ ۶.۱۰ توضیح دهید که چرا در یک مساله برنامه‌ریزی، حذف تأثیر منفی برای هر طرح واکنش، منجر به یک مسالهٔ ساده‌تر می‌گردد.** ***(این تمرین مشابه تمرین ۱۱.۶ در ویرایش دوم است)***

☑ حل: اهداف و پیش شروط، بایستی الفاظی مثبت باشند. بنابراین یک تأثیر منفی فقط رسیدن به هدف را دشوارتر می‌سازد (و یا رسیدن از پیش‌شرط به اقدامی که به هدف می‌رسد). لذا حذف تمام اثرات منفی فقط مساله را ساده‌تر می‌کند. البته اگر پیش‌شروط و اهداف مجاز به منفی باشند، آنگاه دیگر این موضوع درست نخواهد بود.

**؟ ۷.۱۰ شکل ۴.۱۰ مسالهٔ دنیای بلوک‌ها که مشهور به مساله «ساسمن بی‌قاعده» است، را نشان می‌دهد. این مساله از آن جهت بی‌قاعده نامیده می‌شود که هیچ یک از برنامه‌ریزان بدون جایگذاری در اوایل 1970 نتوانستند این مساله را حل کنند. تعریفی از این مساله ارائه داده و آن را یا با دست و یا با نرم‌افزاری برنامه‌ریز حل کنید. یک برنامه‌ریز بدون جایگذاری، برنامه‌ریزی است که اگر دو زیر هدف $G_1$ و $G_2$ را داشته باشد، می‌تواند برای $G_1$ یک برنامه متصل به برنامه $G_2$ تولید کند و یا برعکس. توضیح دهید که چرا یک برنامه‌ریز بدون جایگذاری نمی‌تواند این مساله را حل کند.** ***(این تمرین مشابه تمرین ۱۱.۱۱ در ویرایش دوم است)***




شکل ۱۱.۱۶ مساله برنامه ریزی دنیای بلوکها «sussman anomaly»

☑ حل: حالت شروع عبارتست از:

$On(B, Table) \wedge On(C, A) \wedge On(A, Table) \wedge Clear(B) \wedge Clear(C)$

و هدف عبارتست از:

$On(A, B) \wedge On(B, C)$

در ابتدا توضیح می‌دهیم که چرا این مورد برای برنامه‌ریز بدون جایگذاری، یک بی‌قاعدگی محسوب می‌شود. دو زیرهدف وجود دارد: فرض کنید در ابتدا بر روی $On(A,B)$ کار می‌کنیم. می‌توانیم از $A$، $C$ را حذف کنیم سپس $A$ را به $B$ حرکت می‌دهیم. ولی با انجام این کار، دیگر هیچ راهی برای رسیدن به $On(B,C)$ نداریم. به طور مشابه اگر در ابتدا بر روی زیرهدف $On(B,C)$ کار کنیم باز هم پس از یک مرحله فورا به همین نتیجه می‌رسیم پس باید مجددا به همان حالت $A$ در $B$ برگردیم. اکنون نشان می‌دهیم که یک برنامه‌ریز بدون جایگذار نظیر POP چگونه کار می‌کند. از آنجا که $On(A,B)$ در حالت شروع صحیح نمی‌باشد، پس تنها یک راه داریم: برای برخی xها $Move(A,x,B)$ به طور مشابه، به مرحلهٔ $Move(B,x',C)$ برای برخی x'ها

می‌رسیم. اکنون بیایید نظری به مرحلهٔ $Move(A,x,B)$ بیاندازیم. نیاز است تا پیش‌شرط آن یعنی $Clear(A)$ بدست آید. که این امر توسط $Move(b,A,y)$ و یا $MoveToTable(b,A)$ حاصل می‌شود. حال به مرحلهٔ بعدی نظری بیاندازیم. اگر $b$ را به $C$ بچسبانیم، آنگاه تمام پیش‌شرط‌های مرحلهٔ $MoveToTable(C,A)$ در حالت شروع، صحیح هستند. و می‌توانیم لینک‌هایی غیرمهم به آنها بیافزاییم. البته توجه داشته باشید که یک موضوع وجود دارد: مرحلهٔ $\text{Move}(B,x',C)$ از شرط $Clear(C)$ خبر می‌دهد که خود به مرحلهٔ $MoveToTable$ نیازمند است. این نکته را می‌توان با قرار دادن $Move(B,x',C)$ پس از مرحلهٔ $MoveToTable$ رفع نمود. در نهایت، دقت داشته باشید که تمام پیش‌شرط‌های $Move(B,x',C)$ در حالت شروع صحیح هستند. بنابراین یک برنامهٔ کامل داریم که تمام پیش‌شرط‌ها را ارضا می‌کند. لذا بایستی این سه مرحله به ترتیب طی شوند:

$MoveToTable(C, A)$
$Move(B, Table, C)$
$Move(A, Table, B)$

**؟ ۸.۱۰ اثبات کنید که جستجوی پس‌رو برای مسائل PDDL کامل است.**

☑ حل: دلیل آن برای جستجوی پیش‌رو نیز یکسان خواهد بود: در غیاب نمادهای تابعی، یک فضای حالت PDDL متناهی می‌باشد بنابراین هر الگوریتم جستجوی کاملی برای برنامه‌ریزی PDDL کامل خواهد بود، چه با روش پس‌رو و چه با روش پیش‌رو استفاده شود.

**؟ ۹.۱۰ برای گراف برنامه‌ریزی در مسألهٔ شکل ۱.۱۰، سطوح 0، 1 و 2 را تشکیل دهید.** ***(این تمرین مشابه تمرین ۸.۱۱ در ویرایش دوم است)***

☑ حل: ترسیم این سطوح کمی پیچیده و فراتر از فضای کنونی است لذا به بیان برخی نکات اساسی برای این کار بسنده می‌کنیم: ۱) هر دو اقدام Load و Fly در سطح A0 ممکن هستند زیرا هواپیما با وجود خالی بودن هم می‌تواند پرواز کند. ۲) اثرات منفی در S1 بکار می‌روند و در سایر اجزای مثبت بکار نمی‌رود.

**؟ ۱۰.۱۰ اصول زیر را بری گراف برنامه‌ریزی اثبات کنید: الف) لفظی که در مرحلهٔ نهایی گراف بکار نمی‌رود، نمی‌تواند قابل دسترس باشد. ب) برای دسترسی به یک لفظ، هزینهٔ سطح آن لفظ در گراف سریال، بیشتر از هزینهٔ واقعی یک برنامهٔ بهینه نیست.** ***(این تمرین مشابه تمرین ۹.۱۱ در ویرایش دوم است)***

☑ حل: **الف)** الفاظ ماندگار هستند بنابراین اگر در مرحلهٔ نهایی بکار نروند آنگاه در هیچ جای دیگر نیز بکار نمی‌روند و در این صورت هرگز قابل‌دسترس نیستند. **ب)** در یک گراف برنامه‌ریزی سریال، در هر مرحلهٔ زمانی، فقط یک اقدام انجام می‌شود. هزینهٔ سطح (مرحله‌ای که آن لفظ اولین بار بکار رفته است) همان حداقل تعداد اقداماتی است که در یک برنامه صورت می‌گیرد که شاید در آن صورت به آن لفظ دست یابیم.

**؟ ۱۱.۱۰ هیورستیک «مجموعه- سطح» برای تخمین هزینهٔ دسترسی به یک هدف عطفی از حالت فعلی، از گراف برنامه‌ریزی استفاده می‌کند. چه مسائل تعدیل یافته‌ای هستند که هیورستیک مجموعه- سطح بتواند در آنها به پاسخ دست یابد؟**

☑ حل: طبیعت مسائل تعدیل یافته را در این فصل مشاهده کنید.

**؟ ۱۲.۱۰ تعریف جستجوی دوطرفه در فصل ۳ را بررسی می‌کنیم: الف) آیا جستجوی دوطرفه فضای حالت برای برنامه‌ریزی، ایده‌ای مناسب است؟ ب) جستجوی دوطرفه فضاهایی با برنامه‌ریزی جزئی چطور است ؟ ج) نسخه ای از برنامه ریزی نیمه مرتب ارائه دهید که در آن یک واکنش در صورتی به یک برنامه افزوده شود که پیش‌شرط آن توسط اثرات واکنش‌های موجود در برنامه، بدست آید. توضیح دهید که چطور با محدودیت‌های تناقض و ترتیب روبرو شویم؟ آیا این الگوریتم برای جستجوی فضای حالت پیش‌رو یکتاست؟** ***(این تمرین مشابه تمرین ۷.۱۱ در ویرایش دوم است)***

☑ حل: **الف)** استفاده از جستجوی دوطرفه امکان پذیر است زیرا می‌توان در آن واکنش‌ها را معکوس نمود. با این حال بیشتر کارهایی که در این زمینه انجام شده است، نشان می‌دهد که جستجوی دوطرفه در حالت کلی کارا نیست زیرا جستجوهای پس‌رو و پیش‌رو تمایل به عدم وجود دیگری دارند. این موضوع در یک فضای حالت بزرگ مشخص خواهد شد. برخی برنامه‌ریزان نظیر PRODIGY (فنیک و بلیس،۱۹۹۸) از جستجوی دوطرفه استفاده کرده‌اند. **ب)** مجددا می‌گوییم که این موضوع ممکن می‌باشد ولی پرکاربرد و عمومی نیست. PRODIGY در حقیقت جزئی از یک برنامه‌ریز نیمه‌مرتب است: در جهت پیش‌رو می‌تواند یک برنامه کاملا مرتب ارائه دهد و در جهت پس‌رو یک برنامهٔ نیمه‌مرتب با ساختار درختی ارائه می‌دهد. **ج)** واکنش $A$ در صورتی افزوده می‌شود که تمام پیش‌شروط $A$ توسط سایر مراحل برنامه، حاصل شده باشند. زمانیکه $A$ افزوده شد، محدودیت‌های ترتیب و لینک‌های اتفاقی نیز افزوده می‌شوند تا مطمئن شویم که $A$ پس از تمام واکنش‌هایی که آن را فعال می‌کنند، ظاهر می‌شود. همین طور برای پیش‌شرطی که قبل از اجرای $A$ نباید مستقر شود. این الگوریتم جستجوی خود را به صورت پیش‌رو انجام می‌دهد ولی مشابه جستجوی فضای حالت پیش‌رو نیست زیرا در زمانیکه آنها تناقض ندارند، نیز می‌توانند واکنش‌ها را به طور موازی بسط دهد. به عنوان مثال اگر $A$ دارای سه پیش‌شرط باشد که توسط واکنش‌های بدون تناقض $D$ و $C$ و $B$ ارضا شوند، آنگاه راه‌حل برنامه می‌تواند توسط یک برنامه نیمه‌مرتب بازنمایی شود که در آن یک برنامه‌ریز فضای حالت باید تمام !3 جایگشت $D$ و $C$ و $B$ را در نظر بگیرد.

**؟ ۱۳.۱۰ ما برنامه‌ریزان جستجوی فضای حالت پیش‌رو و پس‌رو را با برنامه‌ریز نیمه‌مرتب مقایسه کرده و متوجه شدیم که مورد آخر یک جستجوگر فضای برنامه است. توضیح دهید که چگونه جستجوهای فضای حالت پیش‌رو و پس‌رو می‌توانند به عنوان مثال جستجوگرهای فضای برنامه نیز در نظر گرفته شوند و همچنین بگویید که عملگرهای پالایش برنامه چیستند؟** ***(این تمرین مشابه تمرین ۱۰.۱۱ در ویرایش دوم است)***

☑ حل: یک برنامه‌ریز فضای حالت پیش‌رو، یک برنامه جزئی که همان دنباله خطی واکنش‌هاست را نگهداری می‌کند. عملگرپالایش برنامه باید یک واکنش قابل اجرا را به پایان دنباله بیافزاید. و بروزرسانی الفاظ مطابق با تأثیر واکنش انجام می‌شود. یک برنامه‌ریز فضای حالت پس‌رو، یک برنامهٔ جزئی که همان دنبالهٔ معکوس از واکنش‌هاست را نگهداری می‌کند. عملگر پالایش برنامه باید یک واکنش را به ابتدای دنباله به گونه‌ای بیافزاید که اندازهٔ تأثیر واکنش با اندازهٔ حالت آغازین دنباله سازگار باشد.

**؟ ۱۴.۱۰ تاکنون فرض نمودیم که پیش‌شروط اقدامات در برنامه‌هایی که می‌سازیم همواره باید ارضا شوند. اکنون می‌خواهیم بدانیم چه قوانین حالت پسین گزاره‌ای نظیر:** $HaveArrow^{t+1} \Leftrightarrow (HaveArrow^t \wedge \neg Shoot^t)$ **باید در مورد اقدامات بیان شود تا پیش‌شروط ارضا نشوند. الف) نشان دهید که این قانون پیش‌بینی می‌کند که در زمانیکه یک واکنش در حالتی اجرا شود که پیش‌شروط آن ارضا نشده باشند، هیچ اتفاقی نمی‌افتد. ب) یک برنامهٔ $p$را به طوری در نظر بگیرید که شامل اقداماتی جهت رسیدن به یک هدف به همراه اقداماتی غیرمجاز باشد. آیا این مورد به صورت زیر است:**

$$initial\ state \wedge successor\text{-}state\ axioms \wedge p \models goal?$$

**ج) توسط قوانین حالت پسین مرتبهٔ اول در محاسبات موقعیت، آیا ممکن است اثبات شود که یک برنامه با اقدامات غیرمجاز به یک هدف دست یابد؟** ***(این تمرین مشابه تمرین ۱۵.۱۱ در ویرایش دوم است)***

☑ حل: **الف)** ایده اصلی را به کمک قانونی که داده شده است، نشان می‌دهیم. فرض کنید $Shoot^t$ درست ولی $HaveArrow^t$ نادرست باشد. آنگاه سمت راست قانون نادرست می‌شود بنابراین $HaveArrow^{t+1}$ نیز نادرست می‌شود. در حالت کلی، اگر یک پیش‌شرط واکنش نقض شود، آنگاه هر دو عبارت $ActionCausesNotF^t$ و $ActionCausesF^t$ نادرست می‌شوند. بنابراین قانون حالت پسین کلی به این صورت تعدیل می‌شود:

$$F^{t+1} \Leftrightarrow False \vee (F^t \wedge True)$$

که معادل است با آنکه بگوییم: $F^{t+1} \Leftrightarrow F^t$ ، که هیچ اتفاقی نمی‌افتد. **ب)** بله. برنامه به همراه قوانین می‌تواند رضایت هدف را ایجاب کند. قوانین هر جریان در جهت یک اقدام غیرمجاز را کپی کرده و مابقی برنامه همچنان کار می‌کند. دقت داشته باشید که ایجاب هدف تقریبا در صورت افزودن قوانین پیش‌شرط حاصل می‌شود زیرا در آن صورت برنامه بطور منطقی با قوانین ناسازگار شده و هر جمله توسط یک تناقض مشخص می‌شود. قوانین پیش‌شروط، یک راه پیش‌گیری از قوانین غیرمجاز در متدهای برنامه‌ریزی مبتنی بر ارضاپذیری است. **ج)** خیر. همانطور که در این فصل گفته شد، قوانین حالت پسین توسط اقدامات غیرمجاز از هر آن چه بیرون برنامه است، جلوگیری می‌کند. زمانی که $Poss(a,s)$ نادرست است، قوانین هیچ چیزی در مورد موقعیت‌های حاصل از اقدامات بیان نمی‌کنند.

**؟ ۱۵.۱۰ چگونگی تبدیل یک مجموعه طرح به قوانین حالت پسین در محاسبات موقعیت را در نظر بگیرید. الف) طرح‌وارهٔ $Fly(p,from,to)$ را در نظر بگیرید. تعریفی منطقی برای مسند $FlyPrecond(p,from,to,s)$ بنویسید که در صورتی درست باشد که پیش شرط‌های $Fly(p,from,to)$ در موقعیت $s$ ارضا شوند. ب) سپس، فرض کنید $Fly(p,from,to)$ تنها طرح واکنش قابل دسترس برای کارگزار باشد. یک قانون حالت پسین برای $At(p,x,s)$ بنویسید که همان اطلاعات طرح واکنش را دربرگیرد. ج) اکنون فرض کنید یک روش دیگر برای حمل‌ونقل موجود باشد: $Teleport(p,from,to)$. این عبارت پیش‌شرط $\neg Warped(p)$ و تاثیر $Warped(p)$ را اضافه بر قبل دارد. توضیح دهید که پایگاه دانش محاسبات موقعیت چگونه باید تغییر کند. د) در نهایت یک روال کلی و دقیق برای انجام حمل‌ونقل بین یک مجموعه طرح STRIPS به مجموعه دیگری از حالت پسین ارائه دهید.** ***(این تمرین مشابه تمرین ۳.۱۱ در ویرایش دوم است)***

☑ حل: نکته آن جاست که برای نوشتن هر قانون حالت پسین بطور صحیح، نیازمند داشتن تمام اقداماتی هستیم که باید به یک حقیقت افزوده یا حذف شوند. به بیان دیگر نوشتن یک قانون STRIPS نیازمند داشتن تمام حقایقی است که با داشتن یک واکنش باید اضافه یا حذف شوند. **الف)**

$$Poss(Fly(p,from,to),s) \Leftrightarrow At(p,from,s) \wedge Plane(p) \wedge Airport(from) \wedge Airport(to)$$

**ب)**

$$Poss(a,s) \Rightarrow (At(p,to,Result(a,s)) \Leftrightarrow (\exists from\ \ a = Fly(p,from,to)) \vee (At(p,to,s) \wedge \neg\exists new\ \ new \neq to \wedge a = Fly(p,to,new)))$$

**ج)** بایستی قانون ممکن برای واکنش جدید را بیافزاییم:

$$Poss(Teleport(p,from,to),s) \Leftrightarrow At(p,from,s) \wedge \neg Warped(p,s) \wedge Plane(p) \wedge Airport(from) \wedge Airport(to)$$

قانون حالت پسین برای موقعیت بایستی اصلاح شود:

$$
\begin{aligned}
&Poss(a, s) \Rightarrow \\
&\quad (At(p, to, Result(a, s)) \Leftrightarrow \\
&\qquad (\exists\, from \;\; a = Fly(p, from, to)) \vee \\
&\qquad (\exists\, from \;\; a = Teleport(p, from, to)) \vee \\
&\qquad (At(p, to, s) \wedge \neg \exists\, new \;\; new \neq to \wedge \\
&\qquad\quad (a = Fly(p, to, new) \vee a = Teleport(p, to, new))))
\end{aligned}
$$

سرانجام باید برای $Warped$ یک قانون حالت پسین بیافزاییم:

$$
\begin{aligned}
&Poss(a, s) \Rightarrow \\
&\quad (Warped(p, Result(a, s)) \Leftrightarrow \\
&\qquad (\exists\, from, to \;\; a = Teleport(p, from, to)) \vee Warped(p, s))
\end{aligned}
$$

**د)** روال پایه همان است که در توضیحات برنامه‌ریزی کلاسیک بعنوان ارضاپذیری بولین بحث شد، به جز آنکه هیچ مرحلهٔ پایه‌ای وجود ندارد؛ قوانین پیش‌شرطی برای هر واکنش همان تعریف $Poss$ هستند؛ و قوانین حالت پسین از ساختار گفته شده در این فصل توسط کمیت‌های اساسی برای تمام متغیرهای آزاد در واکنش‌ها استفاده می‌کند. که در مثالهای فوق نشان داده شده است.

**؟ ۱۶.۱۰ در الگوریتم SATPLAN (شکل ۲۲.۷) هر فراخوانی الگوریتم ارضاپذیری به یک هدف $\mathbf{g}^{\mathbf{T}}$ منجر می‌شود که T دارای محدوده‌ای است که از 0 تا $T_{\max}$ است. فرض کنید که به جای اینکار الگوریتم ارضاپذیری فقط یکبار و با هدف**

$$g^0 \vee g^1 \vee \ldots \vee g^{T_{\max}}$$

**فراخوانی شود. الف) اگر طولی کمتر و یا مساوی با $T_{\max}$ وجود داشته باشد، آیا این کار همچنان یک برنامه را تولید می‌کند؟ ب) آیا این راه‌کار، هر راه‌حل جدید جعلی را تولید می‌کند؟ ج) بحث کنید که چطور می‌توان یک الگوریتم ارضاپذیری را نظیر WALKSAT تغییر داد تا با داشتن یک هدف فصلی به این شکل، بتواند راه‌حلی کوتاه بیابد(در صورت وجود). *(این تمرین مشابه تمرین ۱۱.۱۷ در ویرایش دوم است)***

☑ حل: **الف)** بله. این کار در هر زمان SATPLAN نرمال یک برنامه با طول حداقل $T_{\max}$ بیابد، خواهد توانست یک برنامه بیابد. **ب)** خیر. این کار باعث نمی‌شود که SATPLAN یک راه‌حل نادرست برگرداند. ولی ممکن است در برنامه‌ریزی آن این اثر وارد شود. به عنوان مثال دست‌یابی و یا عدم دست‌یابی به هدف برای چندین مرتبه. **ج)** نمی‌توان راه‌حل ساده و روشن برای آنکه WALKSAT به راه‌حلی کوتاه برسد ارائه داد. زیرا طول یک برنامه را نمی‌دانیم- در واقع مسأله آنجاست که یک مساله برنامه‌ریزی، قسمتی از رمزنگاری است و نه قسمتی از WALKSAT. ولی با اعمال یک استراتژی تقریبی بر روی WALKSAT می‌توانیم به راه‌حلی کوتاهتر توسط روش زیر دست یابیم. برای هر اقدام متغیری گرفته و به هر متغیر اقدام به طور تصادفی مقدار False را برای شروع انتساب می‌دهیم. (به خصوص آنهایی که جنب آن هستند) و به طور تصادفی یک متغیر واکنش را نسبت به متغیر کناری‌اش معکوس می‌کنیم.

# فصل ۱۱ (ویرایش سوم)

**؟۱.۱۱ اهدافی که تا کنون درنظرگرفتیم بدان صورت بود که از برنامه‌ریز خواسته می‌شد تا دنیا را به گونه‌ای بسازد که در یک مرحله زمانی، هدف ارضا شود (به هدف دست یابیم). ولی بیان تمام اهداف با این روش ممکن نیست: اگر هدف شما آویزان کردن لوستر به سقف باشد نمی‌توانید با پرتاب کردن آن به بالا به آن هدف دست یابید. یا برای مثالی دیگر شما نمی‌خواهید سیستم حفظ جان در سفینه فضایی فقط برای یک روز اکسیژن‌رسانی کند ولی از جهت دیگر بهتر از عدم اکسیژن‌رسانی است. یک «هدف بقا» زمانی بدست می‌آید که برنامهٔ کارگزار موجب ایجاد شرایطی برای حفظ مدام یک حالت به سمت جلو شود. توضیح دهید که چگونه فرمول گفته شده در این فصل را گسترش دهیم تا تعریف هدف بقا را نیز شامل شود.**

☑ حل: برای درنظرگرفتن هدف بقا از حالت شروع و امتداد آن در طی برنامه، میتوان با یک گسترش ساده به آن دست یافت. اهداف ایمن معمولا به این شکل هستند و می‌توانند مسائل کلاسیک برنامه‌ریزی را بطوری گسترش دهند که هدف بقا میسر شود. یک برنامه در صورتی می‌تواند یک مسأله را حل کند که حالت پایانی شرط اهداف قاعده‌مند را ارضا کرده و تمام حالات ملاقات شده نیز شرط هدف بقا را ارضا کنند. برای سیستم حفظ جان در سفینه فضایی، یک برنامهٔ متناهی نمی‌تواند آن را حل کند. و نیاز به گسترش داریم که برنامه‌ای نامتناهی تولید کند. یک برنامه نامتناهی در صورتی یک مسأله برنامه‌ریزی را حل میکند که سرانجام هدف توسط برنامه ارضا شود. راه‌حلهای نامتناهی را می‌توان بطور متناهی توسط لوپ‌ها توصیف نمود. برای مثال آویختن لوستر از سقف، می‌توانیم واکنش NoOP تعریف کنیم بدان معنا که هیچ کاری به جز آنچه در مدل فیزیکی است، انجام نشود. بنظر می‌رسد یک راه‌حل این مسئله دارای یک پیشوند متناهی و سپس در انتهای خود بی‌نهایت NoOP داشته باشد. (لوپ بی‌نهایت) این موضوع به مساله اجازه می‌دهد تا پس از چندین مرحلهٔ زمانی که نتوان لوستر را آویزان نمود، آنگاه بی‌ثباتی لوستر پرتاب شده را ثبت کند.

**؟۲.۱۱ شما تعدادی کامیون دارید که باید مجموعه‌ای از بسته‌بندی‌ها را تحویل دهند. هر بسته‌بندی از مکانی خاص بر روی نقشه شبکه‌ای شروع به حرکت کرده و دارای یک مقصد مشخص در جایی دیگر از نقشه است. هر کامیون را می‌توان با حرکت به سمت جلو و چرخاندن، بطور کامل کنترل نمود. برای این مساله یک سلسله‌مراتب از واکنشهای سطح بالا تشکیل دهید. چه اطلاعاتی از راه‌حل هستند که سلسله‌مراتب شما را رمزنگاری می‌کنند؟**

☑ حل: در ابتدا باید واکنشهای اصلی را مشخص کنیم. برای حرکت کامیون $t$، واکنشهای $Forward(t)$، $TurnLeft(t)$ و $TurnRight(t)$ را تعریف می‌کنیم و برای تحویل بسته‌بندی‌ها، واکنشهای $Load(p,t)$ و $UnLoad(p,t)$ که $p$ یک بسته‌بندی و t یک کامیون است تعریف می‌شود. بنابراین می‌توان از PDDL در حالت معمول استفاده نمود. این سلسله‌مراتب می‌تواند به چندین روش ساخته شود که یکی از آنها از HLA $Nagivate(t,[x,y])$ به منظور بودن کامیون $t$ در مختصات $[x,y]$ و $Deliver(t,p)$ به منظور تحویل بسته $p$ به مقصد مربوطه توسط کامیون $t$، استفاده می‌کند. فرض می‌کنیم $At(o,[x,y])$ برای کامیون حاوی بسته $o$، موقعیت $[x,y]$ آن را نشان داده و مسند $Destination(p,[x',y'])$، مقصد بسته را مشخص می‌کند. سلسله‌مراتب (شکل ۱.۱۱S) اطلاعاتی را رمزنگاری کرده است که: کامیونها فقط در هر زمان بتوانند یک بسته را حمل کنند؛ بسته‌ها نیز فقط به مقصد مربوطه تحویل داده شوند و نه در میانه راه؛ می‌توان تحویل‌ها را سریال نمود. (در واقعیت ممکن است کامیونها بطور موازی و همزمان حرکت کنند ولی در اینجا هیچ واکنشی جهت بازنمایی این موضوع لحاظ نشده است.) برای یک سطح بالاتر، سلسله مراتب می‌گوید که برنامه‌ریز فقط نیاز دارد تا انتخاب کند که کدام کامیون، کدام بسته را به چه ترتیبی حمل کند و کامیون‌ها باید برای رسیدن به آن مقصد مسیریابی کنند.

```
Refinement(Deliver(t, p),
  PRECOND: Truck(t) ∧ Package(p) ∧ At(p, [x, y]) ∧ Destination(p, [x', y'])
  STEPS: [Navigate(t, [x, y]), Load(p, t), Navigate(t, [x', y']), Unload(p, t)] )
Refinement(Navigate(t, [x, y]),
  PRECOND: Truck(t) ∧ At(t, [x, y])
  STEPS: [] )
Refinement(Navigate(t, [x, y]),
  PRECOND: Truck(t)
  STEPS: [Forward(t), Navigate(t, [x, y])] )
Refinement(Navigate(t, [x, y]),
  PRECOND: Truck(t)
  STEPS: [TurnLeft(t), Navigate(t, [x, y])] )
Refinement(Navigate(t, [x, y]),
  PRECOND: Truck(t)
  STEPS: [TurnRight(t), Navigate(t, [x, y])] )
```

شکل ۱۱.۱S سلسله‌مراتب کامیون

**؟ ۳.۱۱ فرض کنید یک واکنش سطح بالا فقط به صورتی پیاده‌سازی شده است که شامل دنباله‌ای از واکنشهای اصلی باشد. الگوریتمی ارائه دهید که با داشتن سلسله‌مراتب کامل و طرح‌واره واکنشهای اصلی، بتوان پیش‌شروط و تأثیرات آن واکنش را محاسبه نمود؟**

☑ حل: به منظور ساده‌سازی فرض می‌کنیم که در هر زمان حداکثر یک پالایش از واکنش سطح بالا قابل‌اجرا باشد. الگوریتمی که در زیر آمده است در هر نقطه پیش‌شروط شبکه و تأثیرات پیشوند $h$ که تاکنون پردازش شده است را نشان می‌دهد. این الگوریتم هم پیش‌شروط و تأثیرات واکنشهای اصلی و هم پیش‌شروط پالایش. توجه داشته باشید که هر لفظی که در تأثیر نباشد، توسط پیشوند کنونی که در حال پردازش است، لمس نمی‌شود:

```
net_preconditions <- {}
net_effects <- {}
remaining <- [h]

while remaining not empty:
  a <- pop remaining

  if a is primitive:
    add to net_preconditions any precondition of a not in effects
    add to net_effects the effects of action a, first removing any
        complementary literals
  else:
    r <- the unique refinement whose preconditions do not include
        literals negated in net_effect or net_preconditions
    add to net_preconditions any preconditions of r not in effect
    prepend to remaining the sequence of actions in r
```

**؟ ۴.۱۱ فرض کنید یک مجموعه که در حالت خوش‌بینانه از یک برنامهٔ سطح‌بالا قابل دسترسی است، یک ابرمجموعه برای هدف باشد. آیا می‌توان از این موضوع که برنامه به هدف می‌رسد، هر چیزی را نتیجه گرفت؟ مجموعه‌ای که در حالت بدبینانه هیچ اشتراکی با مجموعهٔ هدف ندارد چیست؟ توضیح دهید.**

☑ حل: نمی‌توانیم هیچ نتیجهٔ خاصی بگیریم. داشتن این موضوع که یک مجموعه درحالت خوش‌بینانه، قابل دسترس یک ابرمجموعهٔ هدف است، هیچ کمکی بیشتر از آنکه بدانیم با هدف اشتراک دارد، نمی‌کند. مجموعه خوش‌بینانه قابل‌دسترس، فقط ضمانت می‌کند که نمی‌توانیم به حالات خارج از آن دسترسی یابیم ولی ضمانت نمی‌کند که بتوانیم به هر حالت درونی آن دسترسی یابیم. بطور مشابه مجموعهٔ بدبینانه قابل دسترس فقط می‌گویند که می‌توانیم به حالات درون آن دسترسی داشته باشید و نه حالات خارج از آن.

**؟ ۵.۱۱ الگوریتمی بنویسید که یک حالت شروع(که توسط مجموعه‌ای از الفاظ گزاره‌ای بیان شده است) و یک دنباله HLA (هر کدام توسط پیش‌شروط و میزان خوشبینانه و بدبینانه بودن مجموعه قابل‌دسترس، تعریف می‌شوند) را گرفته و توصیف‌های خوشبینانه و بدبینانهٔ مجموعهٔ قابل دسترس از آن دنباله را با هم مقایسه کند.**

☑ حل: به منظور ساده‌سازی، در اینجا پیش‌شروط HLA را مدل نمی‌کنیم. عملیات انتشار توصیف 1-CNF در طی توصیفات برای حالت خوش‌بینانه و بدبینانه یکسان بوده و به صورت زیر است:

```
state <- initial state

for each HLA h in order:
  for each literal in the description of h:
    choose case depending on form of literal:
      +l:              state <- state - {-l} + {l}
      -l:              state <- state - {l} + {-l}
      poss add l:      state <- state + {l}
      poss del l:      state <- state + {-l}
      poss add del l: state <- state + {l,-l}

description <- conjunction of all literals which are
               not part of a complementary pair in state
```

**؟ ۶.۱۱ در شکل ۲.۱۱ دیدیم که چگونه می‌توان واکنشها را در یک مساله زمان‌بندی توسط فیلدهای مجزا برایDURATION ، USE و CONSUME توصیف نمود. اکنون فرض کنید بخواهیم زمان‌بندی را با برنامه‌ریزی غیرقطعی ترکیب کنیم که نیازمند تأثیرات غیرقطعی و شرطی خواهد بود. هر یک از این سه فیلد را در نظر گرفته و در صورتی که آنها بطور مجزا باقی بمانند، یا اثر یک واکنش شوند، توصیفات خود را ارائه دهید. برای هر یک از این سه مورد مثالی ذکر کنید.**

☑ حل: شکل غیرقطعی و طبیعیDURATION ، USE و CONSUME به جای یک مقدار تک، بازه‌ای از مقادیر ممکن برای هر مورد است. الگوریتمی که بتواند با مقادیر مجزا کار کند، می‌تواند با کمی تغییرات برای مدیریت بازه‌ها نیز استفاده شود.

بنابراین اگر کارگزار با 10 پیچ شروع به حرکت کند و در اولین واکنش در برنامه، 2 الی 4 پیچ مصرف شود، آنگاه واکنش بعدی که نیازمند 5 پیچ است همچنان قابل‌اجرا خواهد بود. وقتی پای اثرات شرطی به میان آید، بایستی با فیلدها متفاوت رفتار شود. فیلد USE به محدودیتی که واکنش *during* برقرار می‌کند اشاره می‌کند. بنابراین باید یک فیلد مجزا باشد و نباید با آن مشابه یک تاثیر رفتار شود. هر دو فیلد DURATION و CONSUME اثرات را تعریف می‌کنند. بنابراین آنها می‌توانند در توصیف شرطی اثر برای واکنش آمیخته شوند.

**؟۷.۱۱ برخی عملگرهای موجود در زبان برنامه‌نویسی استاندارد، می‌توانند به عنوان واکنش‌هایی جهت تغییر حالت دنیا مدل شوند. به عنوان مثال عملگر انتساب در برنامه‌نویسی، محتوای یک محل از حافظه را تغییر داده و عملگر print(چاپ) حالت رشته خروجی را تغییر می‌دهد. یک برنامه‌نویسی شامل چنین عملگرهایی می‌تواند به عنوان برنامه‌ای در نظر گرفته شود که هدفش توسط آن برنامه‌نویسی مشخص می‌شود. بنابراین الگوریتم‌های برنامه‌نویسی می‌توانند برای ساخت برنامه‌هایی با هدف مشخص، استفاده شوند. الف) یک طرح‌واره واکنش برای عملگر انتساب (انتساب یک مقدار از یک متغیر به متغیر دیگر) بنویسید. یادآوری می‌شود که در این حالت باید روی مقدار اصلی، مقداری ریخته شود. ب) نشان دهید که تولید اشیاء چگونه می‌تواند توسط یک برنامه‌ریز بکار رود تا یک برنامه برای جابجایی یک مقدار بین دو متغیر با استفاده از یک متغیر کمکی تولید شود. *(این تمرین مشابه تمرین ۷.۱۲ در ویرایش دوم است)***

☑ حل: به یک واکنش *Assign* نیازمندیم که یک مقدار $d$ را از رجیستر منبع $sr$ به رجیستر مقصد $dr$ انتساب دهد. (می‌توانید به جای اصطلاح رجیستر از اصطلاح متغیر استفاده کنید ولی استفاده از رجیستر روشن می‌سازد که ما از محل‌های فیزیکی استفاده می‌کنیم):

$$\begin{aligned}&Action(\text{ACTION:}Assign(dr, sr),\\ &\quad \text{PRECOND:}Register(dr) \wedge Register(sr) \wedge Value(dr, dv) \wedge Value(sr, sv),\\ &\quad \text{EFFECT:}Value(dr, sv) \wedge \neg Value(dr, dv))\end{aligned}$$

اکنون فرض کنید از یک حالت شروع به صورت زیر آغاز می‌کنیم:

$$Register(R_1) \wedge Register(R_2) \wedge Value(R_1, V_1) \wedge Value(R_2, V_2)$$

و هدف ما عبارتست از:

$$Value(R_1, V_2) \wedge Value(R_2, V_1)$$

متاسفانه برای حل مساله به این صورت، هیچ راهی نداریم. باید شرط $Register(R_3)$ را به حالت شروع بیافزاییم و یا آنکه روشی ارائه دهیم که یک امکان ایجاد رجیستر جدید فراهم شود. برای حل این مشکل، واکنشی تعریف می‌کنیم که رجیستری جدید تولید نماید:

$$\begin{aligned}&Action(\text{ACTION:}Allocate(r),\\ &\quad \text{EFFECT:}Register(r))\end{aligned}$$

بنابراین دنباله مراحل زیر یک برنامهٔ معتبر را ارائه می‌دهد:

$$\begin{aligned}&Allocate(R_3)\\ &Assign(R_3, R_1)\\ &Assign(R_1, R_2)\\ &Assign(R_2, R_1)\end{aligned}$$

**؟۸.۱۱ فرض کنید واکنش *Flip* همواره مقدار درستی متغیر L را تغییر دهد. نشان دهید که چگونه باید اثرات این واکنش را توسط اثرات شرطی تعریف نماییم. نشان دهید که با وجود استفاده از اثرات شرطی، یک بازنمایی حالت باور 1-CNF پس از *Flip* همچنان بصورت 1-CNF باقی می ماند.**

☑ حل: توصیف Flip توسط اثرات شرطی عبارتست از:

$$\begin{aligned}&Action(Flip,\\ &\quad \text{EFFECT:}\textbf{when } L\text{: } \neg L \wedge \textbf{when } \neg L\text{: } L)\end{aligned}$$

برای نشان دادن اینکه یک حالت باور به شکل 1-CNF پس از *Flip* نیز همچنان به شکل 1-CNF باقی می‌ماند، نشان می‌دهیم که 3 مورد وجود دارد: • اگر $L$ در حالت باور درست باشد، آنگاه پس از *Flip* نادرست می‌گردد. • اگر نادرست باشد بعکوس این مورد رخ می‌دهد. • اگر $L$ قبلا ناشناخته باشد، پس از اعمال نیز ناشناخته باقی می‌ماند یعنی $L$ یا $\neg L$ می‌تواند حاصل شود. و سایر اجزای حالت باور بدون تغییر باقی می‌مانند پس همچنان در شکل 1-CNF هستیم.

**؟۹.۱۱ در دنیای بلوک‌ها دو واکنش STRIPS بنام‌های *Move* و *MoveToTable* را جهت برقرای مسند *Clear* تعریف می‌کنیم. نشان دهید که چگونه باید از اثرات شرطی استفاده کنیم تا بتوان هر دوی این موارد را توسط یک واکنش بازنمایی نمود. *(این تمرین مشابه تمرین ۱۰.۱۲ در ویرایش دوم است)***

☑ حل: طبق تعریف دومی که از *Clear* در این فصل بیان گشت، (یک مربع خالی برای یک بلوک وجود داشته باشد)، تنها تغییر موردنیاز آن است که اگر یک جدول داریم مقصد، خالی باقی بماند:

$Action(Move(b,x,y)$,
PRECOND:$On(b,x) \wedge Clear(b) \wedge Clear(y)$,
EFFECT:$On(b,y) \wedge Clear(x) \wedge \neg On(b,x) \wedge$ (**when** $y \neq Table$: $\neg Clear(y)$))

**؟ ۱۰.۱۱ اثر شرطی که برای واکنش Suck در دنیای جاروبرقی گفته شد آن است که خانه‌های محیط بسته به مکانی که روبات در آن قرار دارد، تمیز می‌شوند. آیا شما می‌توانید یک مجموعهٔ جدید از متغیرهای گزاره‌ای برای تعریف حالات دنیـای جـاروبرقی نظیر Suck که یک توصیف غیرشرطی دارد، بیابید؟ با استفاده از گزارهٔ خود توصیفاتی برای Suck، Left، Right ارائه داده و نشان دهید که این گزاره‌ها برای توصیفات تمام حالات ممکن در دنیا کافی هستند. *(این تمرین مشابه تمرین ۱۱.۱۲ در ویرایش دوم است)***

☑ حل: عبارت $CleanH$ درست است اگر و فقط اگر مکان فعلی روبات تمیز باشد و $CleanO$ درست است اگر و فقط اگر خانه دیگر محیط تمیز باشد. بنابراین $Suck$ عبارتست از:

$Action(Suck,$ PRECOND:, EFFECT:$CleanH)$

متاسفانه حرکت کردن بر روی این الفاظ جدید تأثیر می‌گذارد. برای $Left$ داریم:

$Action(Left,$ PRECOND:$AtR,$
EFFECT:$AtL \wedge \neg AtR \wedge$ **when** $CleanH$: $CleanO \wedge$ **when** $CleanO$: $CleanH$
$\wedge$ **when** $\neg CleanO$: $\neg CleanH \wedge$ **when** $\neg CleanH$: $\neg CleanO)$

برای $Right$ نیز همین است.

**؟ ۱۱.۱۱ فرشی کثیف که روی آن مانعی قرار نداشته باشد، را جاروبرقی بکشید. مسیر پیموده شده توسط تمیزکننده جاروبرقی را با دقت ممکن ترسیم کنید. این مسیر را با توجه به برنامه‌ریزی‌های گفته شده در این فصل توضیح دهید. *(این تمرین مشابه تمرین ۱۳.۱۲ در ویرایش دوم است)***

☑ حل: نکته مهم آن است که تمیزکننده جاروبرقی حرکت خود را در مکان‌های کثیف، چندین بار را تکرار می‌کند تا آن مکان کاملا تمیز شود. همچنین حرکت‌های به سمت جلو معمولا کوتاه و سریع بوده و جهت رسیدن به ابتدای منطقه صورت می‌گیرد. این موضوع بر حسب خروجی‌های فصلی عبارتست از: هر منطقه ممکن است کاملا تمیز باشد و یا خیر؛ حرکت برگشتی کارگزار را قادر به بررسی می‌کند؛ و تکرار کردن کارگزار را از تکمیل کار خود مطمئن سازد (البته اگر کثیفی دیرینه بوده و مربوط به نخ قالی شود آنگاه با تکرار جاروکشیدن نیز نمی‌توان آن را تمیز کرد). بنابراین یک برنامه حلقوی قوی با واکنش‌های حسی داریم.

**؟ ۱۲.۱۱ به مسأله تجویز دارو برای زمان‌هایی که *Disease* درست است و مواردی که اثر *Know(CultureGrowth)* دارد، یک واکنش *Test* بیافزایید. یک برنامه شرطی ترسیم کنید که این مسأله را حل کرده و استفاده از واکنش *Medicate* را به حداقل برساند.**

☑ حل: برنامه‌ای که می‌تواند بعنوان راه‌حل این مساله تلقی گردد عبارتست از:

[$Test$, **if** $CultureGrowth$ **then** [$Drink$,$Medicate$]]

# فصل ۱۲ (ویرایش سوم)

**؟ ۱.۱۲ یک آنتولوژی در منطق مرتبهٔ اول برای بازی دوز تعریف کنید. این آنتولوژی باید شامل موقعیت‌ها، واکنش‌ها، مربع‌ها، بازیکنان، علامت‌ها (X و یا 0 و یا خالی) و همچنین نمادهایی برای حالت برنده، بازنده و یا قرعه‌کشی در بازی باشد. همچنین نمادهایی برای برنده حتمی و یا قرعه‌کشی شده، تعریف کنید یعنی موقعیتی که یک بازیکن می‌تواند با انجام دنباله واکنشهای صحیح، بطور حتمی برنده شود (و یا قرعه‌کشی شده). قوانینی برای این دامنه بیان کنید. (راهنمایی: اگر قوانینی دارید که بر حسب مربع‌های مختلف بوده و باید برای هر مربع باید تعریف شود، آنگاه تعداد قوانین بسیار زیاد می‌گردد. پس نیازی نیست که تمام آنها را به طور کامل بیان کنید و صرفا به طور واضح بیان کنید که این قوانین چگونه خواهند بود).**

☑ حل: • مسندهای *Sortal*:

$Player(p)$
$Mark(m)$
$Square(q)$

• ثوابت:

$Op,Xp$: بازیکنان
$Blank,O,X$: علامتهای بازی
$Q11, Q12, \ldots, Q33$: مربعهای بازی
$S0$: موقعیت

• *Atemporal*:

$MarkOf(p)$: تابع. تابعی که علامت بازیکن $p$ را برمی‌گرداند.
$Winning(q1,q2,q3)$: مسند. مربعهای $q1,q2,q3$ حاوی یک موقعیت برنده هستند.
$Opponent(p)$: تابع. تابعی که رقیب بازیکن $p$ را برمی‌گرداند.

• محاسبات موقعیت:

$Result(a,s)$
$Poss(a,s)$

• حالت:

$TurnAt(s)$: تابعی که نشان می‌دهد در موقعیت $s$ نوبت کدام بازیکن است
$Marked(q,s)$: تابعی که در موقعیت $s$ علامت $q$ را قرار می‌دهد.
$Wins(p,s)$: بازیکن $p$ در موقعیت $s$ برنده شده است.

• واکنشها:

$Play(p,q)$: تابعی که به ازای بازیکن $p$ و خانهٔ $q$، واکنشی که بازیکن $p$ باید برای علامت‌گذاری $q$ استفاده کند، را نشان می‌دهد.

• قوانین *Atemporal*:

A1. $MarkOf(Xp) = X$
A2. $MarkOf(Op) = O$
A3. $Opponent(Xp) = Op$
A4. $Opponent(Op) = Xp$
A5. $\forall p \;\; Player(p) \;\Leftrightarrow\; p = Xp \vee p = Op$
A6. $\forall m \;\; Mark(m) \;\Leftrightarrow\; m = X \vee m = O \vee m = Blank$
A7. $\forall q \;\; Square(q) \;\Leftrightarrow\; q = Q11 \vee q = Q12 \vee \ldots \vee q = Q33$
A8. $\forall q1,q2,q3 \;\; WinningPosition(q1,q2,q3) \;\Leftrightarrow$
$[q1 = Q11 \wedge q2 = Q12 \wedge q3 = Q13] \vee$
$[q1 = Q21 \wedge q2 = Q22 \wedge q3 = Q23] \vee$
(برای سایر شش موقعیت برنده نیز مشابه همین خواهد بود) ...
$[q1 = Q31 \wedge q2 = Q22 \wedge q3 = Q13].$

• تعریف برنده:

A9. $\forall p,s \;\; Wins(p,s) \;\Leftrightarrow$
$\exists q1,q2,q3 \;\; WinningPosition(q1,q2,q3) \wedge$
$MarkAt(q1,s) = MarkAt(q2,s) = MarkAt(q3,s) = MarkOf(p)$

• قوانین سببی:

A10. $\forall p,q \;\; Player(p) \wedge Square(q) \Rightarrow MarkAt(q, Result(Play(p,q),s)) = MarkOf(p).$

A11. $\forall p,a,s \;\; TurnAt(p,s) \Rightarrow TurnAt(Opponent(p), Result(a,s))$

• قوانین پیش‌شرطی:

A12. $Poss(Play(p,q),s) \Rightarrow TurnAt(s)=p \wedge MarkAt(q,s)=Blank$

• قوانین چهارچوب:

A13. $q1 \neq q2 \Rightarrow MarkAt(q1, Result(Play(p,q2),s)) = MarkAt(q1,s)$

نام‌های یکتا:

A14. $X \neq O \neq Blank$

(توجه: ویژگی یکتایی بازیکنان $X_p \neq O_p$ از A14 و A1 و A2 نتیجه می‌شود.)

برای هر $i,j,k,m$ بین 1 و 3 که $i \neq k$ و یا $j \neq m$ است $Q_{ij} \neq Q_{km}$ را داریم A15-A50.

توجه: در بسیاری از نظریه‌ها بهتر است تا قواعد نام بین موجودیت‌های گوناگون، یکتا باشد.

بعنوان مثال: $\forall p,q \; Player(p) \wedge Square(q) \Rightarrow p \neq q$

البته در این نظریه نیاز نیست، اگر فرض کنید که بودن بازیکن $X_p$ در خانهٔ $Q23$ به عنوان یک موجودیت با واکنش $Play(X_p,Q23)$ یکسان باشد، آنگاه هیچ آسیبی به راه‌حل شما وارد نمی‌شود.

**؟ ۱۲.۲ شکل ۱.۱۲ مراحل فوقانی از سلسله مراتب هر چیزی را نشان می‌دهد. آن را گسترش دهید تا در حد امکان، طبقات حقیقی بیشتری را شامل شود. یک روش خوب آن است که تمام چیزهایی که در روزهای زندگیتان با آنها مواجه می‌شوید را در آن قرار دهید. این موضوع هم شامل اشیا و هم رویدادها می‌شود. از آغاز بیدار شدن خود شروع کرده و بترتیب تمام چیزهایی که می‌بینید، لمس می‌کنید، انجام می‌دهید و دربارهٔ آنها می‌اندیشید را ثبت کنید. به عنوان مثال، در یک قسمت تصادفی از زندگی داریم: موزیک، اخبار، شیر، پیاده‌روی، رانندگی، بنزین، ساختمان، Soda Hall، فرش، صحبت کردن، پروفسور فیتمن، ادویه مرغ، زبان، هفت دلار، خورشید، روزنامه روزانه، و ... . شما باید هم یک چارت سلسله مراتبی (در یک برگه بزرگ) و هم لیستی از اشیا و طبقات به همراه رابطه‌ای که بین اعضای هر طبقه وجود دارد تهیه کنید. هر شی باید در یک طبقه بوده و هر طبقه نیز در یک سلسله مراتب واقع باشد.** ***(این تمرین مشابه تمرین ۲۱.۱۰ در ویرایش دوم است)***

☑ حل: این تمرین و دو تمرین بعدی نسبتا پیچیده بوده و برای انتخاب به عنوان پروژه مناسب هستند. بهتر است خواننده برای امور روزمره زندگی، سلسله مراتب طبقات را یافته و سپس کار خود را با سلسله مراتب‌های آنلاین نظیر سایت یاهو مقایسه کنند.

**؟ ۳.۱۲ یک سیستم بازنمایی برای استدلال در مورد پنجره‌ها در محیط رایانه‌ای مبتنی بر پنجره طراحی کنید. بازنمایی شما باید بتواند این موارد را توصیف کند: • حالت یک پنجره: «کمینه(Minimze)»، «در حال نمایش(Display)» و «عدم وجود(Nonexistent)» • کدام پنجره (اگر وجود داشته باشد) «پنجرهٔ فعال(Active)» است • «موقعیت(Position)» هر پنجره در هر لحظه • ترتیب پنجره‌هایی که با یکدیگر همپوشانی دارند (جلو و پشت هم) • واکنشهای «ایجاد»، «بستن»، «تغییر اندازه» و «حرکت پنجره»، «تغییر در حالت پنجره» و «آوردن یک پنجره به جلو». این واکنشها را به صورت اتمیک در نظر بگیرید. به ارتباط این واکنشها با واکنشهای ماوس کاری نداشته باشید. قواعدی روان و سلیس برای توصیف اثر این واکنشها بر fluentها بیان کنید. می‌توانید از رخدادها و یا محاسبات موقعیت استفاده نمایید. فرض کنید یک آنتولوژی شامل موقعیت‌ها، واکنشها، اعداد صحیح (برای مختصات‌های $x,y$) و پنجره‌ها باشد. یک زبان برای این آنتولوژی تعریف نمایید که شامل لیستی از ثوابت، نمادهای تابعی و مسندها بوده و برای هر یک توصیفی ارائه دهید. در صورت نیاز می‌توانید دسته‌های بیشتری را به این آنتولوژی بیافزایید (مانند پیکسل‌ها). ولی حتما در نوشته های خود آنها را مشخص کنید. بایستی از نمادهای تعریف شده در متن استفاده کنید ولی اطمینان حاصل کنید که آنها را بطور دقیق ذکر کرده‌اید.**

☑ حل: یکی از زبانهایی که می‌تواند برای این مساله چاره‌ساز باشد، شامل موارد پایه زیر است:

• مسندهای موقت:

$Poss(a,s)$ مسند: واکنش $a$ در موقعیت $s$ ممکن است. مانند بخش ۳.۱۰

$Result(a,s)$ تابعی از واکنش $a$ و موقعیت $s$ به موقعیت است. مانند بخش ۳.۱۰

• محاسبه:

$x<y$ , $x \leq y$ , $x+y$ , $0$

• حالت پنجره:

$Active(w,s)$ ،$Nonexistent(w,s)$ ،$Displayed(w,s)$ ،$Minimmized(w,s)$

مسند: در تمام این مسندها، $w$ پنجره و $s$ یک موقعیت است

منظور از $Displayed(w,s)$ آن است که پنجره وجود دارد و کمینه نیست

این عبارت همچنین شامل مواردی میشود که w توسط سایر پنجره‌ها مسدود باشد

• موقعیت پنجره:

*BottomEdge*(w,s) ،*TopEdge*(w,s) ،*LeftEdge*(w,s) ،*Rightedge*(w,s)
توابعی بر حسب پنجرهٔ $w$ و موقعیت $s$ که در خروجی یک موقعیت را برمی‌گردانند
*Screentheight* ،*Screenwidth* ثوابت

تیب پنجره‌ها:
*Infront*($w_1$,$w_2$,$s$) مسند: پنجرهٔ $w_1$ در جلوی پنجرهٔ $w_2$ در موقعیت $s$ است

کنشها:
*BringToFront*(w) ،*Destory*(w) ،*MakeVisivble*(w) ،*Minimized*(w)
توابعی که برای پنجرهٔ $w$، یک واکنش را برمی‌گرداند
*Move*(w,dx,dy)
حرکت پنجرهٔ $w$ به اندازهٔ $dx$ به چپ و $dy$ به بالا
($dx$ و $dy$ می‌توانند منفی بوده و نشان از خلاف جهت حرکت باشند)
*Resize*(w,dxl,dxr,dyb,dyt)
تغییر اندازهٔ پنجرهٔ $w$ به اندازهٔ $dxl$ به چپ و $dxr$ به راست، $dyb$ به پایین و $dyt$ به بالا

**۴.۱۷ طبق زبانی که برای تمرین قبل طراحی کردید این موارد را بیان کنید: الف) در موقعیت $S_0$، پنجرهٔ $W_1$ پشت پنجرهٔ $W_2$ ب ولی این پنجره از راست و چپ آن کمی بیرون آمده است. مختصات خاصی را برای این مورد در نظر بگیرید و موقعیت کلی وصیف کنید. ب) اگر یک پنجره در حال نمایش است آنگاه لبه بالایی آن بیشتر از لبهٔ پایینی است. ج) پس از ایجاد یک رهٔ W، آن پنجره نمایش داده می‌شود. د) اگر یک پنجره در حال نمایش باشد، می‌تواند کمینه شود.**

حل: **الف)** $LeftEdge(W1,S0) < LeftEdge(W2,S0) \wedge RightEdge(W2,S0) < RightEdge(W1,S0) \wedge$
$TopEdge(W1,S0) \leq TopEdge(W2,S0) \wedge BottomEdge(W2,S0) \leq BottomEdge(W1,S0) \wedge$
$InFront(W2,W1,S0)$

$$\forall w,s \;\; Displayed(w,s) \Rightarrow BottomEdge(w,s) < TopEdge(w,s)$$
$$\forall w,s \;\; Poss(Create(w),s) \Rightarrow Displayed(w,Result(Create(w),s))$$
$$Displayed(w,s) \Rightarrow Poss(Minimize(w),s)$$

**۵.۱۷ (با اقتباس از مثال Doug Lenat) شما باید با توجه به جمله زیر دانش را به شکل منطقی بیان کرده و سپس به ش‌هایی که در زیر آمده است پاسخ دهید. «دیروز جان به سوپر مارکت Safeway در شعبهٔ شمالی برکلی مراجعه کرده و دو گوجه‌فرنگی و یک پوند گوشت چرخ‌کرده خریداری نمود.» در ابتدا مفهوم جملهٔ فوق را با تعدادی جمله بازنمایی کنید. لات شما باید ساختار منطقی آسانی داشته باشند (به عنوان مثال، عباراتی که اشیای آن ویژگی‌های مشخص داشته باشند؛ و باراتی که اشیای آن به هم مرتبط باشند؛ و یا عباراتی که تمام اشیای مشترک در یک ویژگی، در ویژگی دیگر هم صدق ند). این موارد برای شروع به شما کمک می‌کنند: • به چه کلاس‌ها، اشیا و روابطی نیازمندید؟ والدین، هم نژادها و ....کدام ند؟ (شما به رخدادها و ترتیب زمانی بین آنها نیاز خواهید داشت). • در چه مکانی می‌توان آنها را با یک سلسله مراتب می‌تر تطبیق داد؟ • محدودیت ها و روابط متقابل بین آنها چیست؟ • به چه جزئیاتی از مفاهیم مختلف باید توجه داشت؟ باید با سئوالاتی چون: چه چیزهایی در یک سوپرمارکت وجود دارد؟ فردی که برای انتخاب یک چیز مراجعه می‌کند، چه هایی سفارش می‌دهد؟ چه چیزهایی برای سفارش استفاده می‌شوند و ... مواجه شوید. بکوشید تا حد امکان بازنمایی خود را می کنید. به یک مثال ساده توجه کنید: به عنوان مثال بازنمایی جملهٔ «مردم غذا را از Safeway می‌خرند» کم کاربرد است برای افرادی که از دیگر مغازه‌ها خرید می‌کنند بکار نمی‌رود و یا بازنمایی جملهٔ « جو اسپاگتی را از گوجه و گوشت‌چرخکرده ت می‌کند» نیز کم‌اهمیت است زیرا درست کردن اسپاگتی با هر چیز دیگر را شامل نمی‌شود. سعی کنید این نکته را در ایی سئوالات رعایت کنید مثلا سئوال ج می‌پرسد «آیا جان مقداری گوشت خرید؟» نه اینکه «آیا جان یک پوند گوشت کرده خرید»؟ زنجیره استدلال خود را گسترش دهید تا بتوانید به سئوالات پاسخ دهید. در طی فرآیند در مورد افزودن یم، ایجاد قوانین بیشتر و غیره دچار تردید نشوید. در صورت امکان از یک سیستم استنتاج منطقی برای نمایش کارایی اه دانش خود کمک بگیرید. بسیاری از آنچه شما می‌نویسید ممکن است در دنیای حقیقی تخمینی باشد ولی نگران نباشید هدف ما فقط استخراج چیزی است که بتواند به این سئوالات پاسخ دهد و پاسخگویی دقیق و کامل به این سئوالات واقعا ار بوده و شاید فراتر از بازنمایی فعلی دانش باشد. ولی شما باید بتوانید یک مجموعه سازگار از قوانین برای این سئوالات ود تشکیل دهید: الف) جان یک کودک است و یا یک بزرگسال؟ [بزرگسال] ب) آیا هم‌اکنون جان حداقل دو گوجه‌فرنگی ؟[بله] ج) آیا جان مقداری گوشت خرید؟ [بله] د) اگر ماری در همان زمانی که جان گوجه‌فرنگی خرید در حال خرید بوده ، آیا جان او را دیده است؟ [بله] هـ) آیا گوجه‌فرنگی‌ها در سوپرمارکت تولید می‌شوند؟ [خیر] و) جان قصد دارد با گوجه- ی‌ها چه کند؟ [آنها را بخورد] ز) آیا Safeway، بوگیر می‌فروشد؟ [بله] ح) آیا جان مقداری پول به سوپرمارکت آورده ؟ [بله] ط) آیا مقدار پول جان کمتر از مقدار پول او قبل از آمدن به سوپرمارکت است؟ [بله] *(این تمرین مشابه تمرین ۱۰. ر ویرایش دوم است)***

حل: این مسأله به بازنمایی خیلی زیادی نیاز دارد لذا پیشنهاد می‌شود تیم‌های 2 الی 3 نفره تشکیل شده و با حداقل 2 زمان، این کار را انجام دهند. برای درنظرگرفتن زمان و دانش پس‌زمینه کافی بایستی از محاسبات موقعیت، محاسبات

جریان، یا محاسبات رخداد در پاسخ خود استفاده نمایید. اگر از یک سیستم برنامه‌ریزی منطقی یا اثباتگر قضیه استفاده نمی‌کنید، قوانینی که تهیه می‌کنید حداقل باید به اندازه‌ای باشد که پاسخگوی سئوالات گفته شده باشد.

**؟ ۶.۱۲ پایگاه دانش تمرین قبل را به گونه‌ای تغییر دهید یا اضافه کنید که پاسخگوی سئوالات زیر باشد. در توضیحات خود، در مورد تغییرات ایجاد شده، دلیل نیاز ما به آنها، اصلی و فرعی بودن آنها و اینکه کدام سئوالات ما را ناگزیر به ایجاد تغییرات میکنند، بحث کنید. الف) آیا در زمانی که جان در Safeway بود، افراد دیگری هم آنجا بودند؟ [بله، کارمند] ب) آیا جان گیاه‌خوار است؟ [خیر] ج) چه کسی صاحب بوگیر در Safeway است؟ [شرکت Safeway] د) آیا جان مقدار کمی گوشت چرخکرده دارد؟ [بله] هـ) آیا جایگاه بعدی Shell، بنزین دارد؟ [بله] و) آیا گوجه‌فرنگی‌ها در صندوق ماشین جان، جا می‌شوند؟ [بله] *(این تمرین مشابه تمرین ۱۰.۲۳ در ویرایش دوم است)***

☑ حل: پاسخ این تمرین در گرو پاسخ مناسب به تمرین قبل است زیرا با تغییر و کمی افزودنی به آن به این پاسخ دست خواهیم یافت. دانشجویان سعی کنند در بازنمایی خود تا حد امکان عمومی و درست بنویسند تا در این تمرینات نیز قابل استفاده باشد و تشخیص اشتباه و تغییرات ساده‌تر شود.

**؟ ۷.۱۲ هفت عبارت زیر را با استفاده و گسترش از بازنمایی گفته شده در این فصل، بازنمایی کنید: الف) آب مایعی با درجهٔ بین 0 تا 100 است. ب) آب در 100 درجه می‌جوشد. ج) آب موجود در بطری آب John، یخ زده است. د) Perrier نوعی آب است. هـ) John در بطری آب خود Perrier دارد. و) تمام مایعات یک نقطهٔ انجماد دارند. ز) وزن یک لیتر آب، بیشتر از وزن یک لیتر الکل است. *(این تمرین مشابه تمرین ۱۰.۵ در ویرایش دوم است)***

☑ حل: در ابتدا بایستی مواد گفته شده در این تمرین را تعریف کنیم. بنابراین «آب» یک طبقه بوده و عناصر وجودی آن باعث می‌شوند تا به آن آب بگوئیم. یکی از مهمترین مشکلات ما با زبان نوشتاری رایج، کژتابی کلمات است. به عنوان مثال به نظر می‌رسد بخشی از کلمهٔ «آب» شامل مفهوم «یخ» هم شود (یخ همان آب منجمد است) ولی از جنبهٔ دیگر کلمهٔ «آب» و «یخ» مجزا هستند (این آب است و آن یخ). به نظر می‌رسد منظور جملات این تمرین همان مورد اول باشد. بنابراین ما پاسخ خود را بر این فرض بنا می‌داریم. همچنین به نظر می‌رسد که در این جملات منظور از «آب» همان $H_2O$ باشد. از دیگر مشکلات ما تغییر ماهیت اشیا در طی زمان است (انجماد و تبخیر). بنابراین اگر w مقداری آب باشد، نمی‌توانیم بگوییم $w \in Liquid$. زیرا $w$ ممکن است در یک زمان مایع و در زمان دیگر جامد باشد. به منظور ساده‌سازی، از بازنمایی محاسبات‌موقعیت مانند $T(w \in Liquid)$ استفاده می‌کنیم. در این صورت ممکن است پاسخ‌های صحیح زیادی برای هر یک از موارد پیدا کنید ولی نکتهٔ مهم آن است که اطلاعات باید به صورت «سازگار» بازنمایی شوند. به عنوان مثال اگر $Water$ به عنوان یک دسته از مواد استفاده شده باشد نباید $Liquid$ را به عنوان یک مسند در شی بکار ببرید. **الف)** می‌توان جملهٔ اول را به این صورت ترجمه نمود: « هر نوع آب در هر موقعیتی مایع است، اگر و فقط اگر دمای آب در موقعیتی بین 0 تا 100 درجهٔ سانتیگراد باشد». پس داریم:

$$\forall w, s \;\; w \in Water \Rightarrow (Centigrade(0) < Temperature(w, s) < Centigrade(100)) \Leftrightarrow T(w \in Liquid, s)$$

**ب)** می‌توان از ابزارهای ساخت استفاده نمود. در این فصل از $MeltingPoint$ به عنوان یک مسند استفاده کردیم که به نمونه‌هایی مشخص از مواد اعمال شد. در اینجا $SBoilingPoint$ را به عنوان نقطهٔ جوش تمام نمونه‌های یک ماده تعریف می‌کنیم. مفهوم کلی جوشیدن آن است که نمونه‌های ماده به اندازهٔ نقطهٔ جوش حرارت داشته باشند:

$$SBoilingPoint(c, bp) \Leftrightarrow \forall x, s \;\; x \in c \Rightarrow (\forall t \;\; T(Temperature(x, t), s) \wedge t > bp \Rightarrow T(x \in Gas, s))$$

بنابراین تنها نیاز است بگوییم:

$$SBoilingPoint(Water, Centigrade(100))$$

**ج)** از ثابت $Now$ به منظور بازنمایی موقعیت فعلی یعنی موقعیتی که جمله در آن بیان می‌شود، استفاده می‌کنیم. البته دقت داشته باشید که ممکن است به سادگی در بیان عبارت «فقط برخی از آب‌های داخل بطری منجمد هستند» دچار اشتباه شوید:

$$\exists b \;\; \forall w \;\; w \in Water \wedge b \in WaterBottles \wedge Has(John, b, Now) \wedge Inside(w, b, Now) \Rightarrow (w \in Solid, Now)$$

**د)** Perrier نوعی آب است. $Perrier \subset Water$

**هـ)** John در بطری آب خود Perrier دارد.

$$\exists b \;\; \forall w \;\; w \in Water \wedge b \in WaterBottles \wedge Has(John, b, Now) \wedge Inside(w, b, Now) \Rightarrow w \in Perrier$$

) احتمالا منظور این عبارت آن است که «تمام موادی که در دمای اتاق مایع هستند، دارای نقطهٔ انجماد می‌باشند». اگر *RTLiquidSubstance* برای بیان این دسته از مواد استفاده کنیم آنگاه داریم:

$$\forall c \;\; RTLiquidSubstance(c) \Rightarrow \exists t \;\; SFreezingPoint(c,t)$$

ته در این عبارت *SFreezingPoint* تعریفی مشابه *SBoilingPoint* دارد. بد نیست بدانید که این عبارت در جهان حقیقی ادرست است: می‌توان دسته‌هایی مانند «blue liquid» اختراع کرد که یک نقطهٔ انجماد یکتا ندارند. می‌توان برای تمرین، مادهٔ خالصی تعریف نمود که تمام نمونه‌های آن دارای ترکیب شیمیایی یکسان باشند.

) وزن یک لیتر آب، بیشتر از وزن یک لیتر الکل است.

$$\forall w,a \;\; w \in Water \wedge a \in Alcohol \wedge Volume(w) = Liters(1) \wedge Volume(a) = Liters(1) \Rightarrow Mass(w) > Mass(a)$$

**؟ ۸.۱۲ برای این موارد تعریف‌هایی ارائه دهید: الف) *ExhaustivePartDecomposition* ب) *PartPartition* ج) *PartwiseDisjoint*. تعریف این موارد، مشابه تعریف *ExhaustiveDecomposition*، Partition و Disjoint می‌باشد. آیا مواردی وجود دارد که در آن *PartPartition(s,BunchOf(s))* باشد؟ اگر وجود دارد آن را اثبات کنید وگرنه مثالی نقض ارائه کرده و شرایط لازم جهت برقراری آن را تعریف کنید. *(این تمرین مشابه تمرین ۶.۱۰ در ویرایش دوم است)***

☑ حل: **الف)** *ExhaustivePartDecomposition* بین یک مجموعه اجزا و یک عنصر کامل قرار می‌گیرد و می‌گوید «هر چیزی که یکی از اجزای کامل باشد باید یکی از اجزای موجود در مجموعهٔ اجزا باشد»:

$$\forall s,w \;\; ExhaustivePartDecomposition(s,w) \Leftrightarrow (\forall p \;\; PartOf(p,w) \Rightarrow \exists p_2 \;\; p_2 \in s \wedge PartOf(p,p_2))$$

ب) *PartPartition* زمانی بین یک مجموعه اجزا و عنصر کامل قرار می‌گیرد که آن مجموعه مجزا و تجزیه‌ای جامع باشد:

$$\forall s,w \;\; PartPartition(s,w) \Leftrightarrow PartwiseDisjoint(s) \wedge ExhaustivePartDecomposition(s,w)$$

ج) یک مجموعه اجزا در صورتی *PartwiseDisjoint* است که اگر هر دو عنصر آن مجموعه را بردارید، دیگر هیچ چیزی باشد که جزئی از آن دو عنصر باشد:

$$\forall s \;\; PartwiseDisjoint(s) \Leftrightarrow \forall p_1,p_2 \;\; p_1 \in s \wedge p_2 \in s \wedge p_1 \neq p_2 \Rightarrow \neg\exists p_3 \; PartOf(p_3,p_1) \wedge PartOf(p_3,p_2)\, hOf(s))$$

رای هر $s$ عبارت $(BunchOf(s))$ چیزیکی با هم همپوشانی داشته باشند، مانند «یک دست» و «انگشتان یک دست». آنگاه در این مثال، $BunchOf(s)$ معادل ست با دست ولی $s$ یک قسمتی از آن نیست. بنابراین نیاز است تا مطمئن شویم که عناصر $s$ مجزا بوده و همپوشانی نداشته اشند:

$$\forall s \;\; PartwiseDisjoint(s) \Rightarrow PartPartition(s, BunchOf(s))$$

**؟ ۹.۱۲ یک راه کار ثانویه برای بازنمایی اندازه‌گیری آن است که تابع واحد(unit) را به شی انتزاعی طول اعمال کنیم. به عنوان ثال در این بازنمایی $Inches(Length(L_1))=1.5$ است. این راه‌کار در مقایسه با آنچه که در این فصل گفته شد، چگونه خواهد ود؟ نتایج بحث شامل تبدیل اصول، نامگذاری کمیت‌های انتزاعی(مانند 50 دلار) و مقایسه معیارهای انتزاعی در واحدهای ختلف (مثلا 50 اینچ بیشتر از 50 سانتی‌متر است) می‌باشد. *(این تمرین مشابه تمرین ۸.۱۰ در ویرایش دوم است)***

☑ حل: در راهکار ارائه شده در این فصل، یک اصل تبدیل مشابه این است:

$$\forall x \;\; Centimeters(2.54 \times x) = Inches(x)$$

50 دلار» به صورت $(50)$\$ می‌باشد که نام یک کمیت انتزاعی و پولی است. برای هر تابع اندازه‌گیری نظیر \$، می‌توان از $>$ به صورت زیر استفاده کرد:

$$\forall x,y \;\; x > y \Rightarrow \$(x) > \$(y)$$

صل تبدیل برای دلار و سنت عبارت است از:

$$\forall x \;\; Cents(100 \times x) = \$(x)$$

ه سرعت می‌توان نتیجه گرفت که:

$$\$(50) > Cents(50)$$

ر راه‌کار جدید که این تمرین پیشنهاد می‌دهد، باید اشیایی را معرفی کنیم که طول آنها تبدیل شده باشد:

$$\forall x \;\; Centimeters(Length(x)) = 2.54 \times Inches(Length(x))$$

هیچ روش مشخصی برای بیان مستقیم «50 دلار» و یا ارتباط آن با «50 سنت» وجود ندارد. مجددا باید اشیایی را تعریف کنیم که ارزش پولی آنها 50 دلار یا 50 سنت باشد:

$$\forall x, y \ \$(Value(x)) = 50 \wedge Cents(Value(y)) = 50 \Rightarrow \$(Value(x)) > \$(Value(y))$$

**؟ ۱۰.۱۲ قواعدی بیافزایید که با استفاده از آنها بتوان تعریف مسند Name(S,C) را به طوری گسترش داد که رشته‌ای نظیر «Laptop Cpmputer» با نام طبقاتی از فروشگاههای مختلف منطبق شود. سعی کنید تعریفی عمومی ارائه دهید. به ده فروشگاه آنلاین مراجعه کرده و با توجه به نام طبقاتی که آنها برای سه دسته مختلف گذاشته‌اند، تعریف خود را تست کنید. بعنوان مثال، برای طبقهٔ لپ‌تاپ، نام‌هایی چون «Note Book»، «Laptop»، «Note Book Computer»، «Note Books» و «Note Books and Laptop» مشاهده می‌شود. برخی از این نام‌ها صریحا آن نام مربوطه را پوشش می‌دهند، ولی برخی دیگر با استفاده از قواعد جمع، عطف و ... تولید شده‌اند. *(این تمرین مشابه تمرین ۱۵.۱۰ در ویرایش دوم است)***

☑ حل: ترکیبات جمع توسط یک رابطهٔ *Plural* بین رشته‌ها تولید می‌شود: $Plural("computer", "computers")$

به علاوه می‌دانیم که ترکیب جمعی یک اسم، باز هم برای آن دسته یک اسم محسوب می‌شود:

$$\forall c, s_1, s_2 \ \ Name(s_1, c) \wedge (Plural(s_1, s_2) \vee Plural(s_2, s_1)) \Rightarrow Name(s_2, c)$$

ترکیب عطفی، در صورتی نام یک طبقه خواهد بود که یکی از عطف‌ها، نامی برای یک طبقه باشد:

$$\forall c, s, s_2 \ \ Conjunct(s_2, s) \wedge Name(s_2, c) \Rightarrow Name(s, c)$$

که *Conjunct* از روی اصطلاحات الصاق رشته‌ای تعریف می‌شود. البته بهتر بود که به جای این کار، عبارت *RelevantCategoryName* را مجددا تعریف می‌کردیم.

**؟ ۱۱.۱۲ قواعد محاسبات رخدادی بنویسید که واکنشهای دنیای ومپوز را توصیف کند.**

☑ حل: بخش ۳.۱۲ شامل زوج قوانینی برای دنیای ومپوز می‌باشد:

$$Initiates(e, HaveArrow(a), t) \Leftrightarrow e = Start$$
$$Terminates(e, HaveArrow(a), t) \Leftrightarrow e \in Shootings(a)$$

در اینجا قاعده‌ای برای چرخش وجود دارد و سایر موارد دیگر مشابه آن ولی پیچیده‌تر خواهند بود. فرض کنید اصطلاح $TurnRight(a)$ نشان‌دهندهٔ دسته رخدادی باشد که کارگزار به سمت راست می‌چرخد. اکنون می‌خواهیم در مورد مواردی نظیر زیر بحث کنیم:

«کارگزار در آغاز واکنش به سمت جنوب می‌رود و پس از اتمام واکنش به غرب می‌رود و ...»

$$T(TurnRight(a), i) \Leftrightarrow$$
$$[\exists h \ \ Meets(h, i) \wedge T(FacingSouth(a), h) \Rightarrow$$
$$Clipped(FacingSouth(a), i) \wedge Restored(FacingWest(a), i)]$$
$$\vee \ \ \ldots$$

**؟ ۱۲.۱۲ روابط جبر بازه‌ای که بین هر جفت از رخدادهای جهان واقعی در زیر وجود دارد را بیان کنید:**

**LK**: عمر رئیس‌جمهور کندی — **IK**: طفولیت رئیس‌جمهور کندی

**PK**: ریاست‌جمهوری رئیس‌جمهور کندی — **LJ**: عمر رئیس‌جمهور جانسون

**PJ**: ریاست‌جمهوری رئیس‌جمهور جانسون — **LQ**: عمر رئیس‌جمهور

☑ حل:

| | | |
|---|---|---|
| $Starts(IK, LK)$ | $During(IK, LJ)$ | $During(PJ, LJ)$ |
| $Finishes(PK, LK)$ | $Before(IK, PJ)$ | $Overlap(LJ, LC)$ |
| $During(LK, LJ)$ | $Before(IK, LC)$ | $During(PJ, LC)$ |
| $Meets(LK, PJ)$ | $During(PK, LJ)$ | |
| $Overlap(LK, LC)$ | $Meets(PK, PJ)$ | |
| $Before(IK, PK)$ | $During(PK, LC)$ | |

**؟ ۱۳.۱۲ در مورد روش‌هایی تحقیق کنید که می‌توانند با گسترش محاسبات رخدادی[21]، رخدادهای همزمان را پوشش دهند. آیا می‌توان از ترکیب فراوان اصول اجتناب کرد؟ *(این تمرین مشابه تمرین ۴.۱۰ در ویرایش دوم است)***

☑ حل: قسمت دشوار رخدادها و واکنش‌های همزمان، مربوط به چگونگی درست استفاده‌کردن از استنتاج‌های ممکن است. برای شروع بد نیست نظری به مقاله Shanahan که در سال 1999 نوشته است، نظری بیافکنید. بخش پنجم از این مقاله چگونگی مدیریت واکنش‌های همزمان را نشان می‌دهد که برای اینکار مسندهای *Cancels* و *Canceled* را اضافه کرده و چگونگی استنتاج واکنش‌ها با یکدیگر را به تفضیل بیان می‌کند.

[21] Event Calculus

**؟ ۱۴.۱۲ یک بازنمایی برای تبدیل نرخ‌های پولی بنویسید که در آن نوسانات روزانه مجاز باشد. *(این تمرین مشابه تمرین ۹.۱۰ در ویرایش دوم است)***

☑ حل: برای کمیتهایی نظیر زمان و طول، اصول تبدیل زیر همیشه (بجز چند مورد خاص) برقرار می‌باشند:

$$Centimeters(2.54 \times d) = Inches(d)$$

مشابه این قانون برای تبدیل واحدهای پولی برقرار است، بعنوان مثال $US\$(1)=US¢(100)$. جهت ساده‌سازی مبحث فرض کرده‌ایم که در زمان تبدیل دو نرخ پولی؛ مقدار نرخ تبدیل آنها، مشخص و ثابت باشد:

$$T(ExchangeRate(UK£(1), US\$(1)) = 1.55, t)$$

این نرخ تبدیل دوطرفه می‌باشد:

$$ExchangeRate(UK£(1), US\$(1)) = 1/ExchangeRate(US\$(1), UK£(1))$$

داریم:

$$T(UK£(1) = US\$(1.55), t)$$

با استفاده از *equating* می‌توان مقدار پول کشورهای مختلف را اختصارا بیان نمود. البته چنین نیست که نرخ تبدیلات در یک زمان مشخص ثابت و سازگار باشد و می‌توان با استفاده از معادلات تبدیل بین نرخ‌های مختلف پولی، یک ناسازگاری منطقی تعریف نمود. نرخ‌های تبدیل معمولا توسط کمیسیون‌هایی با معنای تمایل به تبدیل ارز، معنا می‌شوند و ناسازگاری نرخ تبدیل، فرصتی برای داوری در این مورد فراهم می‌سازد. البته مدلهای خبره‌ای وجود دارد که تمام این نرخها، محدودیتها و مسائل آنها را دربرمی‌گیرد.

**؟ ۱۵.۱۲ مسند *Fixed* را به طوری تعریف کنید که *Fixed(Location(x))* به معنای آن باشد که مکان شی $x$ در طی زمان ثابت است. *(این تمرین مشابه تمرین ۱۱.۱۰ در ویرایش دوم است)***

☑ حل: هر شی $x$ یک رویداد است و $Location(x)$ رویدادی است که برای هر زیربازه از زمان، به مکان x اشاره می‌کند. به عنوان مثال $Location(Peter)$ یک رویداد پیچیده است و شامل خانهٔ او از نیمه شب تا ساعت 9 امروز، اجزای مختلف مسیر جاده و دفتر کارش از ساعت 10 تا 1:30 می‌شود. برای گفتن آنکه یک رویداد ثابت است باید بگوییم که در هر دو لحظه از رویداد یکسان هستند:

$$\forall e \;\; Fixed(e) \Leftrightarrow$$
$$(\forall a, b \;\; a \in Moments \wedge b \in Moments \wedge Subevent(a, e) \wedge Subevent(b, e)$$
$$\Rightarrow SpatialExtent(a) = SpatialExtent(b))$$

**؟ ۱۶.۱۲ رویداد «معاملهٔ یک چیز یا چیز دیگر» را توصیف کنید. همچنین خرید را به عنوان نوعی معامله توصیف کنید که شیئی که در آن داد و ستد می‌شود، مقداری پول است. *(این تمرین مشابه تمرین ۱۸.۱۰ در ویرایش دوم است)***

☑ حل: فرض کنید $Trade(b,x,a,y)$ نشان دهندهٔ طبقه‌ای از رخدادها باشد که شخص $b$، شی $y$ خود را با شی $x$ از شخص $a$ معامله می‌کند:

$$T(Trade(b, x, a, y), i) \Leftrightarrow$$
$$T(Owns(b, y), Start(i)) \wedge T(Owns(a, x), Start(i)) \wedge$$
$$T(Owns(b, x), End(i)) \wedge T(Owns(a, y), End(i))$$

اکنون برای آنکه خرید را به عنوان نوعی معامله توصیف کنیم باید قیمت را از روی مجموعه‌ای حقیقی از پول‌ها مشخص کنیم:

$$T(Buy(b, x, a, p), i) \Leftrightarrow \exists m \;\; Money(m) \wedge Trade(b, x, a, m) \wedge Value(m) = p$$

**؟ ۱۷.۱۲ مفهوم مالکیت در دو تمرین قبل بسیار ابتدایی بحث شد. به عنوان مثال، در آغاز خریدار مالک اسکناس‌های دلار خود.این تصور در زمانی که مقداری از اسکناس‌ها در بانک باشد اشتباه خواهد شد زیرا شخص به مجموعه اسکناسی بیشتر از آنچه در اختیار دارد، مالکیت ندارد. با افزودن مفاهیمی چون قرض دادن، اجاره کردن، کرایه دادن، و امانت، این تصور دچار مشکلات بیشتری خواهد شد. دربارهٔ انواع مفاهیم قانونی و بدیهی مالکیت تحقیق کرده و طرحی ارائه دهید که بتوان این موارد را در آن به طور رسمی بازنمایی نمود. *(این تمرین مشابه تمرین ۱۹.۱۰ در ویرایش دوم است)***

☑ حل: برخی از نکات کلیدی عبارتند از: • مالکیت در هر زمانی اتفاق می‌افتد پس به روش‌های محاسبات موقعیت یا محاسبات تازه‌ای نیازمندیم. • شاید بتوان مالکیت و مالکیت حقوقی را به هم پیوند داد در آنصورت مالک، گروهی هم نوع تعریف می‌شود ولی در حالت ساده، مالک گروهی متشکل از یک نفر می‌باشد. • مالکیت شامل برخی کارها خواهد بود: استفاده کردن، فروش مجدد، بخشیدن و ..... بسیاری از این کارها به خودی خود در تعریف گنجانده نمی‌شوند ولی میزان مناسب بودن یک پاسخ، در گرو بازنمایی تعداد بیشتری از این موارد است. • مالکیت می‌تواند به معنای مالکیت اشیای مرتبط تفسیر شود. این ایده‌ی بانکها و سوپرمارکت‌های آینده خواهد بود. زمانی که شما یک دلار در بانک سپرده می‌گذارید، شما مالک آن دلار خاص هستید و در زمان پس گرفتن، مالک دلار دیگری هستید که به شما داده می‌شود. (البته تصادفا ممکن است همان دلار خاص به شما برگردانده شود). مفاهیمی نظیر اجاره به همین روش تفسیر می‌شوند که نوعی بازنمایی هستند زیرا باید تبادلاتی این چنینی تعریف شوند. در این مورد $Withdraw(person,money,bank,time)$ یک شی بوده و مسند نیست.

**؟ ۱۸.۱۲ (با اقتباس از Fagin در سال 1995) بازی را در نظر بگیرید که هر دست آن تنها شامل 8 کارت است، یعنی 4 آس (A) و 4 شاه (K). سه بازیکن با نام‌های کیومرث و بابک و الیاس داریم که با دو کارت یکدیگر سروکار دارند. تمام بازیکنان این دو کارت خود را بدون آنکه مشاهده کنند بر روی پیشانی‌شان قرار می‌دهند ولی سایر بازیکنان می‌توانند کارتهای روی پیشانی یکدیگر را مشاهده کنند. سپس نوبت بین بازیکنان به ترتیب می‌چرخد و هر کس که در نوبت خود بر حسب شواهد بازی می‌داند که بر روی پیشانی‌اش چه کارتی قرار دارد آن را اعلام و برنده آن نوبت می‌شود وگرنه می‌گوید «من نمی‌دانم» و نوبت به نفر بعدی می‌رسد. همهٔ بازیکنان راستگو بوده و در استنتاج باورها ماهر هستند. الف) بازی 1: الیاس و بابک هر دو اعلام کردند که «من نمی‌دانم». کیومرث مشاهده می‌کند که الیاس دارای دو کارت آس(AA) و بابک دارای دو کارت شاه (k-k) است. کیومرث چه باید بگوید؟ (راهنمایی: تمام حالات ممکن برای کیومرث را در نظر بگیرید:A-A, K-K, A-A). ب) هر مرحله از بازی 1 را با استفاده از نمادهای منطق کیفی (model logic) بیان کنید. ج) بازی 2: کیومرث، الیاس و بابک همگی در چرخش اول بازی اعلام کردند که « من نمی‌دانم ». الیاس کارتهای K-K و بابک کارتهای A-K دارد.کیومرث در نوبت دوم بازی خود چه باید بگوید؟ د) بازی 3: الیاس، کیومرث و بابک در نوبت اول بازی خود همگی اعلام کردند که « من نمی‌دانم ». الیاس این جمله را در نوبت دوم بازی خود نیز مجددا اعلام کرد. الیاس و بابک هر دو دارای کارتهای A-K هستند.کیومرث چه باید بگوید؟ هـ) اثبات کنید که همواره یک برنده برای این بازی وجود خواهد داشت.**

☑ حل: خواننده را به مقالهٔ Fagin در سال 1995 ارجاع می‌دهیم تا به مطالعه مثالهای متعددی از استدلال در آنجا بپردازد ولی برای شروع به بیان راه‌حل بازی اول می‌پردازیم: در بازی 1 الیاس می‌گوید «من نمیدانم». اگر کیومرث k-k داشته باشد آنگاه با توجه به آنکه بابک نیز k-k دارد و الیاس آن را دیده است، پس مشخص است که 4 تا شاه بین آن دو بوده و الیاس باید در نوبت خود می‌فهمید که آس دارد و اعلام می‌کرد A-A. با توجه به اینکه الیاس اینکار را نکرده پس کیومرث K-K ندارد. همچنین می‌دانیم که بابک گفته است «من نمی‌دانم». اگر کیومرث A-A داشته باشد، از آنجا که بابک می‌توانست کارتهای روی پیشانی الیاس را مشاهده کند و ببیند که او A-A است پس می‌فهمید که همهٔ 4 آس بین الیاس و کیومرث بوده و خود او دارای K-K است. بعلت اینکه بابک اینکار را نکرده است پس کیومرث دارای A-A نبوده است. بنابراین کیومرث نه دارای A-A و نه دارای K-K است و کارتهای روی پیشانی وی A-K می‌باشد.

**؟ ۱۹.۱۲ فرضیه وجود « دانای‌کل منطقی» که قبلا در این کتاب گفته شد، طبق هیچیک ازاستدلال‌گرهای واقعی، حقیقت ندارد ولی ایده‌آلی برای فرآیندهای استدلال محسوب میشود که بسته به استفاده و کاربردآنها، می‌تواند کمی بیشتر یا کمتر قابل‌قبول باشد. در مورد معقول بودن فرضیه، برای هریک از کاربردهای استدلالی دانشی که در زیر آمده است بحث کنید. الف) بازی‌های رقابتی نیمه‌دانشی مانند بازی‌های کارتی. در اینجا یک بازیکن می‌خواهد در مورد میزان اطلاعات رقیبش در مورد حالت بازی استدلال کند. ب) بازی شطرنج زمان‌دار. در این بازی، بازیکن باید محدودیتهای رقیب و قابلیت های خودش را برای انجام بهترین حرکت در زمان مربوطه درنظرگیرد. بعنوان نمونه اگر بازیکن A زمان بیشتری نسبت به بازیکن B داشته باشد آنگاه بازیکن A حرکتی خواهد کرد که موقعیت بازی پیچیده شود تا بتواند به نتایج بهتری دست یابد. زیرا این بازیکن زمان بیشتری برای کار بر روی استراتژی خود را داشته است. ج) یک کارگزار فروشگاه که در محیطی دارای هزینه های دریافت اطلاعات قرار دارد. د) استدلال در مورد کلید عمومی رمزنگاری که یکی از مسائل لجوجانه محاسباتی (کامپیوتری) است.**

☑ حل: **الف)** فرضیه وجود «دانای کل منطقی» ایده‌آل است. فاکتور محدودکننده اطلاعاتی است که بازیکنان به آنها دسترسی دارند و نه پیچیده بودن محیط اجرا. **ب)** این بازی نمی‌تواند با فرضیه دانای کل منطقی تطبیق یابد. اگر دانای کل منطقی حقیقت داشته باشد، آنگاه هر بازیکن می‌توانست در هر نوبت همواره حرکت بهینه را بازی کند. **ج)** دانایی کل منطقی یک ایده‌آل است. هزینهٔ گرفتن اطلاعات همواره بیشتر از هزینهٔ استدلال در مورد آنهاست. **د)** این مورد بسته به نوع استدلالی دارد که شما قصد انجام آن را دارید. اگر شما می‌خواهید در مورد رابطهٔ رمزنگاری با مسائل محاسباتی خاص استدلال کنید، آنگاه نمی‌توان از دانای کل منطقی استفاده نمود زیرا این فرضیه ایجاب می‌کند که هر مسئله محاسباتی می‌تواند حل شود. بعبارت دیگر اگر شما میخواهید رمزنگاری (رمزگشایی) را بعنوان یک فرآیند جادویی به صورت ایده‌آل درآورید و هیچ پایهٔ محاسباتی نداشته باشد، آنگاه معقول است که فرضیهٔ دانای کل منطقی را به سایر جنبه‌های تئوری اعمال کنید.

**؟ ۲۰.۱۲ عبارات منطقی زیر را به منطق مرتبهٔ اول تبدیل کرده و در مورد نتایج آن توضیح دهید.**

***And(Man,Atleast(3,Son),AtMost(2,Daughter),***
***All(Son,And(Unemployed,Married,All(Spouse,Docter))),***
***All(Daughter,And(Professor,Fills(Department,Physics,Moth))))).***

☑ حل: این مورد بستگی به فرمول زیر دارد:

$$Man(x) \wedge \exists s_1, s_2, s_3 \;\; Son(s_1, x) \wedge Son(s_2, x) \wedge Son(s_3, x)$$
$$\wedge s_1 \neq s_2 \wedge s_1 \neq s_3 \wedge s_2 \neq s_3$$
$$\wedge \neg \exists d_1, d_2, d_3 \;\; Daughter(d_1, x) \wedge Daughter(d_2, x) \wedge Daughter(d_3, x)$$
$$\wedge d_1 \neq d_2 \wedge d_1 \neq d_3 \wedge d_2 \neq d_3$$
$$\wedge \forall s \;\; Son(s, x) \Rightarrow Unemployed(s) \wedge Married(s) \wedge Doctor(Spouse(s))$$
$$\wedge \forall d \;\; Daughter(d, x) \Rightarrow Professor(d) \wedge$$
$$(Department(d) = Physics \vee Department(d) = Math)$$

**؟ ۲۱.۱۲ می‌دانیم که اطلاعات وراثتی در شبکه‌های معنایی می‌تواند با عبارات استلزامی مناسب به طور منطقی بیان شود. در این تمرین، کارایی استفاده از چنین عباراتی را برای وراثت بررسی می‌کنیم: الف) اطلاعات موجود در کاتالوگ‌های ماشین‌های قدیمی نظیر Kelly's Blue Book را در نظر بگیرید. به عنوان مثال، «ون‌های Dodge سال 1973 کمتر از 575 دلار می‌ارزیدند». فرض کنید تمام این اطلاعات را (برای 11000 مدل) طبق آنچه که در این فصل گفته شد، با قواعد رمزنگاری کرده‌ایم. اکنون سه قانون مشابه آنها بنویسید که برای ون Dodge سال 1973 باشد. اگر یک اثبات‌کنندهٔ قضیه زنجیره‌ای پس‌رو نظیر پرولوگ در اختیار داشته باشید، چگونه می‌توانید با استفاده از این قواعد، ارزش یک ماشین خاص را بیابید؟( مثلا برای ماشین JB که نوعی ون Dodge در سال 1973 است). ب) پیچیدگی زمانی روش زنجیره‌ای پس‌رو را با روش وراثتی که در شبکه‌های معنایی استفاده می‌شود، برای این مسأله مقایسه کنید. ج) توضیح دهید که چگونه در یک سیستم مبتنی بر منطق، روش زنجیره‌ای پیش‌رو می‌تواند چنین مسائلی را به طور کارآمد حل کند؟ فرض کنید پایگاه دانش فقط شامل 11000 قانون در مورد قیمت ماشین‌ها باشد. د) موقعیتی را تشریح کنید که در آن اگر از قیمت ماشینی خاص پرسش شود، نه روش زنجیره‌ای پیش‌رو و نه زنجیره‌ای پس رو نتوانند به طور کارا پاسخگو باشند؟ هـ) آیا می‌توانید یک راه‌حل برای این نوع سئوالات ارائه دهید که در تمام موارد در سیستم منطقی به طور کارا حل شوند؟ [ راهنمایی: یادآوری می‌شود که دو ماشین از یک دسته، قیمت یکسان دارند]. *(این تمرین مشابه تمرین ۲۴.۱۰ در ویرایش دوم است)***

☑ حل: در بسیاری از کتاب‌های هوش‌مصنوعی و پرولوگ می‌بینیم که از استلزام برای پیاده‌سازی وراثت استفاده کرده‌اند. این کار به طور منطقی درست است ولی در عمل خیر. **الف)** در اینجا سه قانون پرولوگ را می‌بینید. البته در حقیقت باید به سمت راست این عبارت، بندهای بیشتری اضافه کنیم تا بتوان تمایز بین مدل‌ها، اختیارات و ... را بیان نمود.

```
worth(X,575) :- year(X,1973), make(X,dodge), style(X,van).
worth(X,27000) :- year(X,1994), make(X,lexus), style(X,sedan).
worth(X,5000) :- year(X,1987), make(X,toyota), style(X,sedan).
```

برای دانستن قیمت JB با فرض آنکه در پایگاه داده عبارات زیر موجود باشد:

style(jb,van) , make(jb,dodge),yaer(jb,1973)

آنگاه اگر بخواهیم با روش زنجیره‌ای پس‌رو به جواب برسیم از هدف worth(jb,D) شروع کرده و در نهایت به قیمت D می‌رسیم. **ب)** پیچیدگی زمانی این پرسش $O(n)$ است که $n$ همان 11000 ورودی کاتالوگ Blue Book است، یک شبکه معنایی با وراثت اجازه می‌دهد که با پیگیری لینک‌ها از JB به 1973-dodge-van داشته باشیم و سپس از آنجا لینکی به worth که موجب می‌شود به قیمت مربوطه با زمان $O(1)$ برسیم. **ج)** با فرض آنکه پایگاه دانش شامل سه عبارتی است که قبلا گفتیم اگر بخواهیم با روش زنجیره‌ای پیش‌رو به جواب برسیم باید عبارت worth (jb,575) را به مجموعه حقایق بافزاییم. سپس به سئوال worth(gb,D) می‌رسیم که با $O(1)$ به پاسخ دست می‌یابیم. این کار با ایندکس‌گذاری روی مسند و اولین آرگومان صورت می‌گیرد. به نظر می‌رسد استنتاج منطقی شبیه شبکه معنایی باشد با دو تفاوت: استنتاج منطقی از یک جدول جستجوی هش به جای پیگیری اشاره‌گرها استفاده می‌کند. همچنین استنتاج منطقی عبارات worth را برای هر ماشین ذخیره می‌کند. لذا اگر تعداد ماشین‌ها زیاد باشد فضای زیادی مصرف می‌شود. (در اینگونه موارد ترجیح می‌دهیم به جای 11000 مدل، تعداد مدل کمتری را لحاظ کنیم). **د)** اگر ماشین‌های هر دسته مشخصات زیادی داشته باشند به عنوان مثال: برای هر قطعه کامیون بتوان چندین جایگزین گذاشت آنگاه روش زنجیره‌ای پیش‌رو با متد استلزام نمی‌تواند راه‌کاری عملی در تعیین قیمت یک ماشین باشد. **هـ)** اگر قانونی از نوع زیر داشته باشیم:

```
worth(X,D) :- year-make-style(X,Yr,Mk,St),
              year-make-style(Y,Yr,Mk,St), worth(Y,D)
```

به همراه سایر حقایق موجود در بانک اطلاعاتی در مورد سایر انواع ماشین‌های هم‌نوع با J B ،آنگاه سئوال worth (jb,D) در زمان $O(1)$ ایندکس‌گذاری مناسب حل خواهدشد. صرف نظر از سایر حقایقی که در مورد آن ماشین می‌دانیم و تعداد انواع ماشین‌ها.

**؟ ۲۲.۱۲ ممکن است گمان کنید تشخیص نحو بین لینک‌های غیرمربعی و لینک‌های تک‌مربعی در شبکه‌معنایی نیاز نباشد زیرا لینک‌های تک‌مربعی همواره به طبقات متصل اند. لذا یک الگوریتم وراثت به سادگی می‌تواند متوجه شود که یک لینک غیرمربعی اگر به یک طبقه متصل شود، قصد دارد تا به تمام اعضای آن طبقه اعمال شود. نشان دهید که این استدلال غلط، سفسطه آمیز بوده و مثال‌هایی از خطاهایی که ممکن است بروز دهد را بیان کنید. *(این تمرین مشابه تمرین ۲۵.۱۰ در ویرایش دوم است)***

☑ حل: زمانی که طبقات به شی تبدیل می‌شوند، می‌توانند به عنوان اشیای فردی دارای مشخصاتی باشند (مانند *Cardinality* و *Superset*) که به عناصر داخلی آنها اعمال نمی‌شود. بدون تشخیص بین لینک‌های مربعی و غیرمربعی، عبارت *Cardinality*(*SingletonSets*,1) ممکن است بدان معنا تفسیر شود که هر مجموعه یکتا فقط یک عنصر دارد و یا آنکه فقط یک مجموعه یکتا وجود دارد.

**؟ ۲۳.۱۲ یکی از بخش‌های فرآیند خرید که در این فصل بحث نشد، بررسی سازگاری بین کالاهای خریداری شده است. به عنوان مثال، اگر مشتری یک کامپیوتر سفارش می‌دهد آیا قطعات آن هم‌خوانی دارند؟ اگر یک دوربین دیجیتال سفارش**

**می‌دهد،آیا میزان حافظه و باطری آن مناسب است؟ پایگاه دانشی بنویسید که تشخیص دهد یک مجموعه کالا سازگار هستند یا خیر و بتواند پیشنهاد دهد که در صورت ناسازگاری، کدام کالا باید جایگزین یا اضافه شود. اطمینان حاصل کنید که پایگاه دانش شما به گونه‌ای باشد که حداقل با یک کالا نیز بدرستی کار کرده و بتوان آن را به سادگی برای اقلام بیشتر گسترش داد.** ***(این تمرین مشابه تمرین ۱۴.۱۰ در ویرایش دوم است)***

☑ حل: در اینجا طرح کلی از یک روش را بیان می‌کنیم ولی روش‌های دیگری نیز وجود دارد. هرشی که سفارش داده می‌شود ممکن است برخی اجزای اضافی نیاز داشته باشد (مانند باطری) تا بتواند بدرستی کار کند و همچنین ممکن است بتوان اجزائی دلخواه را به آن افزود. در اینجا نیازمندی را به عنوان رابطهٔ بین یک شی و یک کلاس از شی نمایش می‌دهیم که با تعداد اشیایی که به آن نیاز دارند، مقدار می‌گیرد:

$$\forall x \;\; x \in Coolpix995DigitalCamera \;\Rightarrow\; Requires(x, AABattery, 4)$$

همچنین نیاز به روشی داریم تا بدانیم یک شی سازگار هست یا خیر. یعنی نقش مورد نظر خود را ایفا می‌کند یا خیر. به عنوان مثال:

$$\forall x, y \;\; x \in Coolpix995DigitalCamera \wedge y \in DuracellAABattery \Rightarrow Compatible(y, x, AABattery)$$

بنابراین به سادگی می‌توان مجموعه‌ای از اقلام سفارش داده شده را از جهت سازگاری (یعنی اشیای مورد نیاز هر شی)، بررسی نماییم.

**؟ ۲۴.۱۲ یافتن یک راه‌حل کامل برای مسأله غیر دقیق انطباق در توصیف خریدار در فروشگاه کاری بس دشوار بوده و نیاز به مجموعه کاملی از فرآیندهای پردازش زبان طبیعی و تکنیک‌های بازیابی اطلاعات دارد. یک کار ساده آن است که از کاربر بخواهیم برای خصوصیات مختلف، مقدار مینیمم و ماکزیمم را مشخص کند. همچنین از خریدار می‌خواهیم از گرامر زیر برای توصیف محصولات استفاده کند:**

***Description→Category[Connector Modifier]****
***Connector→"with" | "and" | ","***
***Modfier→ Attribute | Attribute Op Value***
***Op→*** **"=" | ">" | "<"**

**در اینجا *Category* همان طبقه کالا، *Attribute* یک مشخصه از کالا نظیر *CPU* یا قیمت، و *Value* مقدار نهایی آن مشخصه (خصوصیت) می‌باشد.بنابراین پرس‌وجویی نظیر «کامپیوتری با حداقل یک *CPU* 2.5 گیگا هرتزی زیر 1000 دلار» به صورت «کامپیوتری با *CPU*>2.5GHz و price<1000$» تبدیل می‌شود. یک کارگزار فروشگاه پیاده‌سازی کنید که توصیفات این زبان را بپذیرد.** ***(این تمرین مشابه تمرین ۱۶.۱۰ در ویرایش دوم است)***

☑ حل: می‌توان یک رشته متنی را پیمایش کرده و اطلاعات معنایی خاصی را از آن استخراج کنیم. خروجی این فرآیند، تعریفی از اشیای مورد قبول برای کاربر در یک خرید خاص خواهد بود. این کار به روبات کمک می‌کند تا به فروشگاه مراجعه کرده و اقلام منطبق با نیازهای خود را بیابد. در اینجا از تعریف کامل این کار گزار صرف نظر کردیم ولی مراجعه به عملیات موجود در پروژه AIMA صفحات وب، برایتان مفید خواهد بود.

**؟ ۲۵.۱۲ در توصیفی که ما از خرید اینترنتی ارائه دادیم، تمام مراحل مهم خرید واقعی کالا را حذف کردیم. یک توصیف منطقی رسمی برای خرید با استفاده از محاسبات رویدادها ارائه دهید که در آن دنبالهٔ رویدادها از زمانی که کاربر یک سفارش را با کارت اعتباری ارسال می‌کند تا زمانی که صورتحساب و سرانجام کالا را تحویل می‌گیرد، گنجانده شده باشد.** ***(این تمرین مشابه تمرین ۱۷.۱۰ در ویرایش دوم است)***

☑ حل: در اینجا نسخهٔ ساده‌ای از پاسخ را می‌بینید زیرا اگر به جزئیات بپردازیم، پاسخ بسیار طولانی و شاید نامتناهی گردد. اصطلاح $Buy(b,x,s,p)$ را به عنوان طبقه رویدادی تعریف می‌کنیم که خریدار $b$، شی $x$ را از فروشنده s با قیمت $p$ می‌خرد. می‌خواهیم بگوییم که $b$ پولی به $s$ داده و $s$ مالکیت $x$ را به $b$ می‌دهد:

$$T(Buy(b,x,s,p),i) \Leftrightarrow T(Owns(s,x),Start(i)) \wedge \exists m \;\; Money(m) \wedge p = Value(m) \wedge T(Owns(b,m),Start(i)) \wedge T(Owns(b,x),End(i)) \wedge T(Owns(s,m),End(i))$$

# ضمیمه ۱ (سؤالات تستی امتحانی)

*** توجه کنید که در سؤالات منظور از عامل همان کارگزار است.**

۱ – قیاس صوری ارسطو کدام طبقه از تعاریف هوش مصنوعی را به یاد می‌آورد؟

۱) تفکر انسان‌گونه ۲) عملکرد انسان‌گونه ۳) تفکر عقلانی ۴) عملکرد عقلانی

۲ – کدام یک جزء ویژگی‌های محیط کار تخته نرد نمی‌باشد؟

۱) قطعی ۲) کاملا رویت‌پذیر ۳) ایستا ۴) گسسته

۳ – کدامیک از عامل‌های زیر درجه هوشمندی ضعیف‌تری دارند؟

۱) عامل مبتنی بر جدول ۲) عامل‌های واکنشی ساده

۳) عامل‌های مبتنی بر هدف ۴) عامل‌های مبتنی بر سودمندی

۴ – کدام عامل تنها در محیط *کاملا رویت‌پذیر* امکان تصمیم‌گیری صحیح را دارد؟

۱) واکنشی ساده ۲) واکنشی مبتنی بر مدل

۳) مبتنی بر هدف مبتنی بر مدل ۴) مبتنی بر سودمندی مبتنی بر مدل

**** با در نظر گرفتن شرایط زیر به سؤالات ۵ و ۶ پاسخ دهید:**
**الف) در هر شرایطی ب) در شرایطی که هزینه اقدامات در یک سطح برابر باشد. ج) به شرطی که فاکتور انشعاب متناهی باشد. د) هزینه هر اقدام از $\varepsilon$ بزرگ‌تر باشد. هـ) در هر دو جهت از جستجوی اول سطح استفاده شود.**

۵ – روش حل عمیق *شونده تکراری* در چه شرایطی بهینه است؟

۱) 4 ۲) 3 ۳) 2 ۴) 1

۶ – در چه شرایطی جستجوی دو طرفه کامل است؟

۱) 3 و 5 ۲) 5 ۳) 3 و 4 ۴) 1

۷ – کدام جستجو از لحاظ پیچیدگی زمانی ارجح است؟

۱) اول سطح ۲) اول عمق ۳) عمق‌شونده تکراری ۴) دو طرفه

۸ – در کدام نوع از مسائل حالت‌های تکراری غیرقابل اجتناب هستند؟

۱) مسائل دارای اقدامات معکوس‌پذیر ۲) مسائل اقتضایی

۳) مسائل بدون حسگر ۴) مسائل اکتشافی

۹ – در مورد *Graph search* با جستجوی هزینه یکنواخت کدام گزینه صحیح است؟

۱) کامل و غیربهینه است. ۲) نه کامل و نه بهینه است. ۳) کامل نیست و بهینه است. ۴) کامل و بهینه است.

۱۰ – در محیط کاملا رویت‌پذیر و قطعی برای عامل جاروبرقی "بدون‌حسگر" در همان محیط 2 مکانه با عمل L ، R ، S کدام گزینه صحیح نیست؟

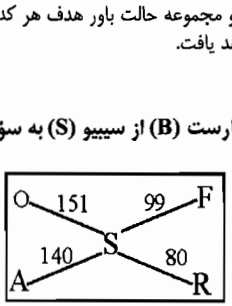
۱) حالت اولیه – مجموعه حالت باور شامل 8 حالت ممکن

۲) حالت هدف – دو حالت هدف مجزا وجود دارد. (دو مجموعه حالت باور هدف هر کدام یک حالت هدف را دربردارند)

۳) به دلیل نداشتن حسگر عامل کافی هدف را نخواهد یافت.

۴) تنها 12 حالت باور دسترس‌پذیر وجود دارند.

**** با توجه به اشکال زیر (با هدف رسیدن به بخارست (B) از سیبیو (S)) به سؤالات ۱۱ و ۱۲ پاسخ دهید:**




| $N$ | $H(n)$ |
|---|---|
| $S$ | 253 |
| $O$ | 380 |
| $A$ | 366 |
| $F$ | 176 |
| $R$ | 193 |

۱۱ – الگوریتم جستجوی حریصانه بعد از $S$ ، کدام گره را اول بسط می‌دهد؟

۱) $O$ ۲) $F$ ۳) $R$ ۴) $A$

۱۲ – جستجو با هزینه یکسان بعد از $S$ ، کدام گره را اول بسط می‌دهد؟

۱) $O$ ۲) $F$ ۳) $R$ ۴) $A$

۱۳ – کدام یک جزء جستجوهای محلی نمی‌باشند؟

۱) تپه‌نوردی ۲) سخت‌سای شبیه‌سازی شده (*Simulated Annealing*)

۳) الگوریتم ژنتیک ۴) *RBFS*

۱۴ – با کدام شرایط در *Graph search* با جستجوی $A^*$ مسیر بهینه به هر حالت تکراری همیشه اولین مسیری است که دنبال می‌شود؟

۱) قابل قبول بودن $h(n)$ ۲) سازگاری $h(n)$

۳) هیچ‌گاه امکان‌پذیر نیست. ۴) بدون شرط همواره امکان‌پذیر است.

۱۵ – کدام گزینه در مورد $SMA^*$ صحیح نیست؟

۱) تا هنگامی که حافظه پر نشده همانند $A^*$ عمل می‌کند.
۲) اگر حافظه پر باشد گره با بیشترین $f$ را حذف می‌کند.
۳) جد یک زیردرخت از کیفیت بهترین مسیر در آن زیردرخت آگاه است.
۴) اگر تمام گره‌های برگی دارای مقدار $f$ یکسانی باشند با شکست روبه‌رو می‌شوند.

۱۶ – با فرض این که مسئله‌ای حاوی سه شرط محدودکننده باشد حداکثر چند مسئله تعدیل شده (*Relaxed*) برای آن می‌توان تولید نمود؟

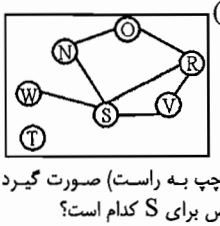
۱) 6 ۲) 4 ۳) 8 ۴) 7

۱۷ – کدامیک جزء روش‌های مرسوم ایجاد توابع هیوریستیک نمی‌باشند؟
۱) به دست آوردن هزینه یک راه‌حل بهینه برای یک مسئله تعدیل شده (*relaxed*)
۲) مجموع (یا ترکیب، هزینه راه‌حل‌های زیر مسائل ( با استفاده از بانک‌های اطلاعاتی الگو)
۳) فراگیر هیوریستیک از تجارب قبلی و یا تعمیم آن برای حالات مشابه
۴) در نظر گرفتن مقدارهای تصادفی برای $h(n)$ و ثبت مقادیری که جستجوی را به سمت هدف هدایت می‌کند.

۱۸ – در حل مسئله ۸ و زیر توسط الگوریتم ژنتیک، مقدار تابع برازش برای یک راه‌حل کدام است؟ (راهنمایی: تابع برازش تعداد جفت وزیرهایی که به هم حمله نمی‌کنند.

۱) 24 ۲) 28 ۳) 56 ۴) 25

۱۹ – اگر در گراف زیر، از هیوریستیک مقدار با حداقل محدودیت در جستجوی پس‌روی استفاده شود و به ترتیب انتساب‌های $W = red$ و $N = green$ را انجام داده‌ایم، در انتساب بعدی به $Q$ چه مقداری را تخصیص خواهیم داد؟ (*CSP*)




۱) *blue*
۲) *green*
۳) *red*
۴) مقدار قابل تخصیصی وجود ندارد.

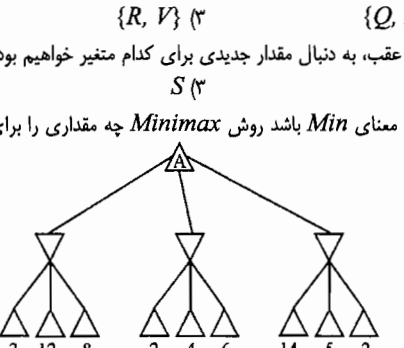
۲۰ – اگر در پس‌گرد هوشمندانه مقداردهی به متغیرهای به ترتیب $T$، $S$ ، $W$ ، $N$ ، $T$ ، $V$ ، $R$ ، $Q$ (از چپ به راست) صورت گیرد و انتساب مقابل صورت گرفته باشد: $\{Q = red, R = qreen, V = blue, T = red\}$ مجموع تناقض برای S کدام است؟

۱) $\{Q, R, V\}$ ۲) $\{Q, R\}$ ۳) $\{R, V\}$ ۴) $\{Q, V\}$

۲۱ – در سؤال قبل پس از پرش رو به عقب، به دنبال مقدار جدیدی برای کدام متغیر خواهیم بود؟

۱) $Q$ ۲) $V$ ۳) $S$ ۴) $T$

۲۲ – اگر $\Delta$ به معنی *Max* و $\nabla$ به معنای *Min* باشد روش *Minimax* چه مقداری را برای (A) در نظر خواهد گرفت؟




۱) 12
۲) 14
۳) 3
۴) 2

۲۳ – اگر در سؤال قبل از روش هرس آلفا و بتا استفاده شود، گره‌ها با چه مقادیری بررسی نمی‌شوند؟

۱) 12 و 8 ۲) 6 و 4 ۳) 2 و 5 ۴) 2 و 5 و 14

۲۴ – در بازی‌های چند نفره در صورتی که بازیکنان $A$ و $B$ نسبت به بازیکن $C$ وضعیت ضعیف‌تری دارند معمولا چه رفتاری صورت می‌گیرد؟

۱) معمولا $A$ و $B$ به جای حمله به یکدیگر به $C$ حمله می‌کنند.
۲) هر کس به‌دنبال برد خود به بقیه حمله می‌کند.
۳) ممکن است یکی از آنها با $C$ در جهت پیشرفت خود همکاری کند.
۴) $A$ و $B$ تا انتهای بازی متحد می‌شوند.

۲۵ – الگوریتم *TT-Entails* برای تصمیم‌گیری در مورد ایجاب گزاره‌ای از چه روشی استفاده می‌کند؟

۱) جدول درستی ۲) تحلیل (*Resolution*) ۳) زنجیره‌ای پیش‌رو ۴) زنجیره‌ای پس‌رو

۲۶ – دانش در کدام عامل غیرانعطاف‌تر می‌باشد؟

۱) حل مسئله – جستجوگر – (هدف‌گرا) ۲) مبتنی بر منطق
۳) عامل‌های سودمند ۴) مبتنی بر دانش

۲۷ – یک جمله ارضا شدنی (*Satisfiable*) است اگر و فقط اگر:

۱) در هر مدلی از جهان صحیح باشد. ۲) با قوانین نحوی یک زبان منطقی ساخته شده باشد.
۳) بتواند توسط یک روال استنتاجی اثبات شود. ۴) تفسیری از جهان وجود داشته باشد که جمله تحت آن صحیح باشد.

۲۸ – کدامیک از جملات زیر به صورت هورن (*Horn*) نوشته شده است؟

۱) $P_1 \wedge P_2 \wedge P_3 \wedge Q_1$ ۲) $P_1 \wedge P_2 \wedge P_3 \Rightarrow Q_1 \wedge Q_2$
۳) $P_1 \wedge P_2 \wedge P_3 \Rightarrow Q_1$ ۴) $P_1 \vee P_2 \vee P_3 \Rightarrow Q_1 \wedge Q_2$

۲۹ – در سه بند زیر کدام یک محض (*pure*) می‌باشد.

$$\begin{cases} 1) \quad A \vee \neg B \\ 2) \neg A \vee C \vee D \\ 3) \neg C \vee \neg B \vee \neg D \end{cases}$$

۱) $A$ ۲) $B$ ۳) $C$ ۴) $D$

۳۰ – کدام ترتیب از جملات زیر باعث جستجوی نامحدود با حالت‌های تکراری در پرولوگ می‌شود؟

۱) $path(X,Z) :- link(X,Z).$
$path(X,Z) :- path(X,Y), link(Y,Z).$

۲) $path(X,Z) :- link(X,Z).$
$path(X,Z) :- path(X,Y), link(Y,Z).$

۳) $append([],Y,Y).$
$append([A|X],Y,[A|Z]) :- append(X,Y,Z).$

۴) $p(X,[X|Y]).$
$P(X,[Y|Z]) :- p(X,Z).$

۳۱ – برای این که یک عامل در آزمون تورینگ پذیرفته شود به کدام قابلیت نیازی ندارد؟

۱) پردازش زبان طبیعی ۲) استدلال خودکار ۳) الگوریتم جستجو ۴) علم روباتیک

۳۲ – کدام یک جزء ویژگی‌های محیط عامل شطرنج زمان‌دار است؟

۱) قطعی ۲) اتفاقی ۳) راهبردی ۴) پیوسته

۳۳ – پیچیده‌ترین محیط کار کدام است؟

۱) نیمه رویت‌پذیر، اتفاقی، ترتیبی، پویا، پیوسته و چند عاملی ۲) نیمه رویت‌پذیر، راهبردی، ترتیبی، پویا، پیوسته و چند عاملی
۳) نیمه رویت‌پذیر، اتفاقی، ترتیبی، نیمه پویا، پیوسته و چند عاملی ۴) نیمه رویت‌پذیر، راهبردی، مرحله‌ایی، پویا، پیوسته و چند عاملی

۳۴ – کدام یک جزء مشکلات عامل‌های مبتنی بر جدول (table driven)، نمی‌باشد؟

۱) حافظه‌ای بیش از حد مورد نیاز جدول ۲) زمان بیش از حد مورد نیاز جهت پر کردن جدول توسط طراح
۳) پیچیدگی زیاد پیاده‌سازی ۴) عدم خودمختاری

۳۵ – کدام یک از عامل‌ها به صورت "تدوین، جستجو، اجرا" طراحی می‌شوند؟

۱) واکنشی ساده ۲) حل مسئله ۳) مبتنی بر دانش ۴) مبتنی بر جدول

۳۶ – فضای حالت مسئله به طور ضمنی توسط کدام گزینه قابل تعریف است؟

۱) حالت‌ها و اقدامات ۲) حالت ابتدایی و اقدامات ۳) حالت شروع و حالت هدف ۴) حالت ابتدایی و تابع پسین

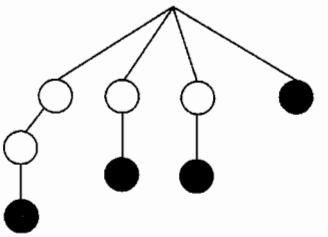
۳۷ – کدام گزینه تعریفی از تجرید (*bstraction*) را ارائه می‌کند؟
۱) در نظر گرفتن یک هدف واحد قصد رسیدن به آن را داریم.
۲) کنار گذاشتن حالت‌هایی که برای جستجوی هدف فعلی به ما کمکی نمی‌کنند.
۳) حذف جزئیات از یک بازنمایی
۴) کنار گذاشتن مسیرهای انحرافی که ما را از هدف دور می‌کند.

**** با در نظر گرفتن شرایط زیر به سؤالات ۳۸ و ۳۹ پاسخ دهید:**
**الف) در هر شرایطی ب) در شرایطی که هزینه اقدامات در یک سطح برابر باشد. ج) به شرطی که فاکتور انشعاب متناهی باشد. د) هزینه هر اقدام از $\varepsilon$ بزرگ‌تر باشد. هـ) در هر دو جهت از جستجوی اول سطح استفاده شود.**

۳۸ – جستجوی اول سطح در چه شرایطی بهینه است؟
۱) 1 ۲) 2 ۳) 3 ۴) 4

۳۹ – در چه شرایطی جستجوی دو طرفه بهینه است؟
۱) 2 و 4 ۲) 2 و 3 ۳) 3 و 4 ۴) 2 و 5

۴۰ – در حین انجام یک جستجو، درخت جستجوی حاصل به شکل زیر رشد یافته است. راس‌هایی که نامزد بسط داده شدن هستند به رنگ سیاه مشخص شده‌اند. این جستجو چه روشی می‌تواند باشد؟



۱) اول عمق (*Depth first*)
۲) اول سطح (*Breadth first*)
۳) جستجوی هزینه یکنواخت (*Uniform cost*)
۴) عمیق شونده تکراری (*Iterative deepening*)

۴۱ – اگر در مسئله $n$ وزیر، هر وزیر بتواند در هر خانه‌ایی که وزیری در آن نباشد قرار گیرید، در آن صورت یک حالت که شامل $n$ وزیر باشد در درخت فضای حالت در چند گره درخت تکرار می‌شود؟
۱) $n$ ۲) $n!$ ۳) تنها یک گره ۴) $\log n$

۴۲ – در عامل جاروبرقی اگر اقدام *Suck* به این صورت تعریف شود که در صورت کثیف بودن مکش و در صورت تمیز بودن، گاهی آشغال‌ها را روی فرش خالی می‌کند (محیط مورفی .. در این صورت عامل در کدام محیط می‌تواند هدف را بیابد؟
۱) محیط رویت‌ناپذیر (مسائل بدون حسگر)
۲) محیط نیمه‌رویت‌پذیر (مسائل اقتضایی)
۳) محیط کاملا رویت‌پذیر
۴) در محیط مورفی، عامل جاروبرقی، حتی با محیط کاملا رویت‌پذیر گاهی ناموفق خواهد بود.

۴۳ – نقطعه ضعف اصلی $A^*$ چیست؟
۱) کامل نبودن ۲) نیمه بهینگی ۳) پیچیدگی زمانی ۴) پیچیدگی حافظه

۴۴ – اگر به ازای هر گره درخت، رابطه $h1 < h2 < h3 < h4$ برای 4 هیوریستیک $h1$ تا $h4$ برقرار باشد، بین ضرایب مؤثر انشعاب مربوط به هر هیوریستیک (یعنی $b1^*$ تا $b4^*$) چه رابطه‌ای برقرار است؟
۱) $b1^* < b2^* < b3^* < b4^*$ ۲) $b4^* < b3^* < b2^* < b1^*$
۳) رابطه مشخصی بین ضرایب برقرار نمی‌باشد. ۴) به ازای $d$ های مختلف روابط متفاوت خواهد بود.

۴۵ – برای مسئله معمای 8 برای وضعیت *start* مقدار $h1$ و $h2$ (فاصله منهتن) به ترتیب کدام است؟
۱) 7و 9 ۲) 9 و 7 ۳) 6 و 10 ۴) 7 و 10

| | | |
|---|---|---|
| | 1 | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

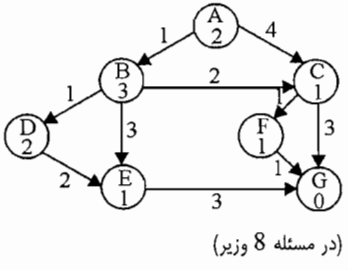
Start state

| | | |
|---|---|---|
| 3 | | 2 |
| 4 | 1 | 8 |
| 5 | 6 | 7 |

Goal state

۴۶ – اگر با فضای حالت نامتناهی طرف هستید و مسیر رسیدن به هدف برای شما اهمیتی ندارد، از کدام جستجو استفاده می‌کنید؟
۱) جستجوی عمق ۲) جستجوی سطحی ۳) جستجوی $A^*$ ۴) جستجوی محلی

۴۷ – مسیر یافت شده توسط الگوریتم جستجو $A^*$ برای گراف مقابل چیست؟




( $A\ B\ C\ F\ G$
( $A\ C\ G$
( $A\ B\ C\ G$

( $A\ B\ E\ G$

۴ – مقدار تابع برازش برای فرد (کروموزوم) 32543213 کدام است؟ (در مسئله 8 وزیر)
راهنمایی: تابع برازش = تعداد جفت وزیرهایی که به هم حمله نمی‌کنند)

( 20 ( 11 ( 23 ( 24

۴ – اگر جمعیت اولیه در الگوریتم ژنتیک حاوی 4 فرد (کروموزوم) باشد و مقدار تابع برازش برای آن‌ها به ترتیب 2، 8 ، 6 و 4 باشد، احتمال انتخاب افراد به ترتیب کدام است؟

( %10، %40 ، %30 و %20 ( %20، %30 ، %40 و %10
( %25، %25، %25 و %25 ( %10 ، %10، %10و %10

۵ – تدوین حالت کامل (که هر حالت یک انتساب کامل است) در مسائل $CSP$ در کدام روش‌های جستجو می‌توانند مفید باشد؟

( جستجوهای پس‌رو ( جستجوهای پیش‌رو ( پس‌رو + $MRV$ ( جستجوهای محلی

۵ – در محدودیت منبع (محدودیت از درجه بالا) برای $Atmost\ (10, PA1, PA2, PA3, PA4)$ (یعنی حداکثر 10 نفر برای 4 کار نیاز داریم)، کدام یک از مجموعه‌های زیر اگر به عنوان دامنه برای هر یک از 4 متغیر در نظر گرفته شود، سازگار خواهند بود؟

( $\{3,4,5,6\}$ ( $\{2,3,4,5,6\}$ ( $\{2,3,4\}$ ( $\{2,3,4,5\}$

۵ – کدام یک از روش‌های زیر در بازی شطرنج می‌توانند پیچیدگی زمانی هرس آلفا و بتا را با *تصمیمات* بهینه به‌طور قابل توجهی کاهش دهند؟

الف) مرتب‌سازی پسین‌ها از بهترین به بدترین. ب) ذخیره مقدار سودمندی برای هر گره در یک جدول هش و استفاده از آنها در حالات مشابه (تکراری) بدون محاسبه مجدد. ج) جایگزینی $Terminal-test$ با $Cutoff-test$ و تابع سودمندی با تابع ارزیاب.

( 1، 2 و 3 ( 1 ( 1و 2 ( 1و 3

۵ – یک مدل (*model*) در منطق چیست؟

( مجموعه‌ایی از قواعد استنتاجی *Sound*
( دنباله‌ایی از اعمال روال‌های استنتاجی برای اثبات یک جمله
( دنیایی است که در آن یک جمله تحت تفسیر خاصی معتبر است.
( مجموعه جملاتی که از روی آنها می‌توان قابل نتیجه‌گیری بودن یک جمله خاص را اثبات نمود.

۵ – اگر $G$ مجموعه جملات یک پایگاه دانش به زبان منطق باشد و $P$ یک جمله به زبان منطق، گوییم $P$ نتیجه منطقی $G(Entailment$ است اگر و فقط اگر:

( مدلی وجود داشته باشد که هم همه جملات $G$ و هم $P$ را ارضا (*Satisfy*) کند.
( هر مدلی که $P$ را ارضا (*Satisfy*) می‌کند، همه جملات $G$ را هم ارضا کند.
( هر مدلی که حداقل یکی از جملات $G$ را ارضا می‌کند، $P$ را هم ارضا کند.
( هر مدلی که همه جملات $G$ را ارضا کند، $P$ را هم ارضا کند.

۵ – الگوریتم *TT-Entalist* برای تصمیم‌گیری در مورد ایجاب گزاره‌ای از چه روشی استفاده می‌کند؟

( جدول درستی ( تحلیل (*Resolution*) ( زنجیره‌ای پیش رو ( زنجیره‌ای پس‌رو

۵ – جمله $P \vee \neg P$ کدام است؟

( معتبر (*Valid*) ( ارضاپذیر (*Satisfiable*)
( ارضاناپذیر (*Unsatisfiable*) ( نامعتبر (*Invalid*)

۵ – استدلال با زنجیره‌ای پیش‌رو (*Forward Chaining*) :

( همیشه کامل است.
( روی بندهای معین (*Definite Clause*) کامل است.
( اصلا کامل نیست.
( روی *CNF* (*Conjunctive Normal Form*) کامل است.

۵ – در الگوریتم *DPLL*، در سه بند زیر کدام نماد محض (*pure*) می‌باشد؟

( A ( B ( C ( D

$$\begin{cases} 1) \quad A \vee \neg B \\ 2) \neg A \vee C \vee D \\ 3) \neg C \vee \neg B \vee \neg D \end{cases}$$

۵۹ – کدام گزینه در مورد الگوریتم *WALKSAT* صحیح نیست؟
۱) اگر مدلی را برگرداند، جمله ورودی واقعا ارضاءپذیر است.
۲) اگر *failure* برگرداند، جمله ورودی ارضاءپذیر نیست.
۳) در هر مرحله یک بند ارضاء نشده را انتخاب کرده و مقدار نمادی از آن را (در مدل) عوض می‌نماید.
۴) به روش حداقل تناقضات در *CSP* ها شباهت زیادی دارد.

۶۰ – کدام زوج از عبارات زیر قابل یکسان‌سازی (*Unification*) هستند؟

۱) $P(f(x),y)$ و $P(y,f(x))$
۲) $P(f(x),f(x))$ و $P(x,x)$
۳) $P(x,y)$ و $P(y,f(x))$
۴) $P(f(x),x)$ و $P(y,f(y))$

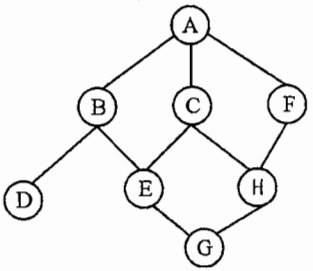
۶۱ – کامپیوتری که بتواند در آزمون تورینگ پذیرفته شود به کدام امکان نیاز ندارد؟ (منظور آزمون جامع نمی‌باشد.)
۱) پردازش زبان طبیعی ۲) استدلال خودکار ۳) بینایی کامپیوتری ۴) یادگیری ماشین

۶۲ – محیط کار سیستم تشخیص پزشکی چگونه است؟
۱) کاملا رویت‌پذیر، قطعی، نیمه پویا، گسسته ۲) کاملا رویت‌پذیر، راهبردی، پویا، گسسته
۳) بعضا رویت‌پذیر، اتفاقی، ایستا، گسسته ۴) بعضا رویت‌پذیر، اتفاقی، پویا، پیوسته

۶۳ – پیچیدگی زمانی جستجوی عمیق شونده تکراری به کدام عامل زیر بستگی دارد؟
۱) بیشترین عمق درخت ۲) اندازه فضای حالت ۳) تابع هیوریستیک (اکتشافی) ۴) عمق کم عمق‌ترین کره هدف

۶۴ – کدام گزینه در مورد الگوریتم جستجوی گراف (*Graph-Search*) صحیح نیست؟
۱) در مسائلی که حالت‌های تکراری زیادی دارد مؤثرتر از *Tree-Search* عمل می‌کند.
۲) خطر از دست دادن بهینگی را دارد.
۳) تمام گره‌های گسترش یافته را در حافظه ذخیره می‌کند.
۴) جستجوی هزینه یکنواخت با الگوریتم Graph-Search بهینه نیست.

۶۵ – در گراف زیر با انجام جستجوی اول عمق و شروع از رأس C ، کدام گره‌های به ترتیب از چپ به راست گسترش می‌یابند؟ (فرزندان یک گروه بر اساس ترتیب حروف الفبا انتخاب می‌شوند.) (با استفاده از *Graph-search*)

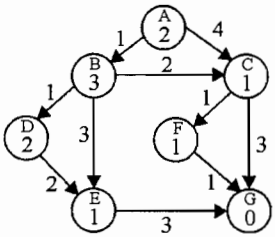



۱) *C, A, H, B, F, D, E, G*
۲) *C, A, E, H, B, F, G, D*
۳) *C, A, B, D, F, E, G, H*
۴) *C, A, B, D, E, G, H, F*

۶۶ – پاسخ سؤال قبل با جستجوی اول سطح کدام گزینه است؟ (با استفاده از *Graph-search*)
۱) *C, A, H, B, F, D, E, G* ۲) *C, A, E, H, B, F, G, D*
۳) *C, A, B, D, F, E, G, H* ۴) *C, A, B, D, F, G, H, E*

۶۷ – در مورد جستجوی دو طرفه کدام گزینه درست نیست؟
۱) پیچیدگی زمانی آن $O(d^{d/2})$ است.
۲) بزرگ‌ترین نقطه ضعف آن، میزان حافظه برای آن است.
۳) اگر هر دو جستجو از نوع اول عمق باشند، الگوریتم کامل و بهینه است.
۴) انجام جستجوی دو طرفه همیشه عملی نیست.

۶۸ – در گراف زیر مسیر پیدا شده توسط الگوریتم جستجو $A^*$ کدام گزینه است؟ ( $A$ گره شروع، اعداد روی یال‌ها هزینه واقعی و اعداد داخل دایره‌ها مقدار $h$ گره مورد نظر است.) (فرض کنید $A^*$ به قابل قبول نبودن $h$ توجهی نمی‌کند.)




– در گراف سؤال قبل، مسیر پیدا شده توسط الگوریتم جستجو اول بهترین حریصانه کدام است؟

۱) $A, B, C, F, G$ ۲) $A, C, G$ ۳) $A, B, C, G$ ۴) $A, B, D, E, G$

– نقطه ضعف الگوریتم جستجوی عمق شونده $IDA^*$ تکراری چیست؟

۱) کامل نبودن ۲) دوباره‌کاری ۳) مصرف زیاد حافظه ۴) کارایی پایین

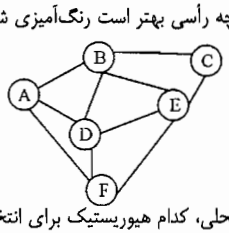
– سه تابع هیوریستیک قابل قبول $h1$ و $h2$ و $h3$ برای حل مساله‌ای به روش $A^*$ پیشنهاد شده است. بهترین انتخاب برای ادامه بر از گره $n$ استفاده از کدام هیوریستیک است؟

۱) $h_1 \times h_2 \times h_3$ ۲) $h_1 + h_2 + h_3$ ۳) $\max\{h_1\ h_2\ h_3\}$ ۴) انتخاب دلخواه یکی از آنها

– برای حل مسأله 8 وزیر به روش جستجوی تپه‌نوردی کدام روش مناسب است؟ (منظور از فرمول‌بندی افزایشی این است که در حالت ع همه وزیرها در صفحه قرار ندارند و به تدریج اضافه می‌شوند.)

۱) فرمول‌بندی حالت کامل با هیوریستیک تعداد جفت وزیرهایی که به هم حمله می‌کنند.
۲) فرمول‌بندی حالت کامل با هیوریستیک تعداد جفت وزیرهایی که به هم حمله نمی‌کنند.
۳) فرمول‌بندی افزایشی با هیوریستیک تعداد جفت وزیرهایی که به هم حمله می‌کنند.
۴) فرمول‌بندی افزایش با هیوریستیک تعداد جفت وزیرهایی که به هم حمله نمی‌کنند.

– در مسأله رنگ‌آمیزی گراف زیر، بعد از رنگ‌آمیزی رئوس $A$ و $B$ چه رأسی بهتر است رنگ‌آمیزی شود؟ (با توجه به هیوریستیک $MI$)




C
D
E
F

– در حل یک مساله ارضاء محدودیت ($CSP$) به روش جستجوی محلی، کدام هیوریستیک برای انتخاب یک مقدار برای متغیر انتخاب استفاده می‌شود؟

۱) هیوریستیک حداقل مقادیر باقی‌مانده ($MRV$) ۲) هیوریستیک حداقل تناقضات
۳) هیوریستیک متغیر با حداکثر محدودیت ۴) هیوریستیک درجه

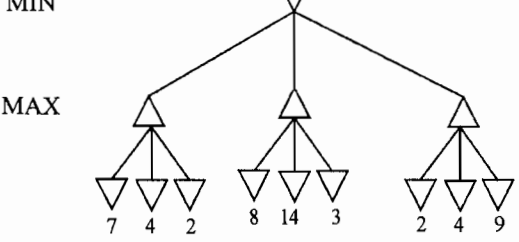
– در حل یک $CSP$ که با روش فرمول‌بندی افزایشی تعریف شده است کدام جستجو مناسب عمل می‌کند؟

۱) اول عمق ۲) اول سطح ۳) جستجوی محلی ۴) $A^*$

– کدام یک از مسائل زیر برای حل توسط الگوریتم $MIN\text{-}CONFLICT$ مناسبت نیست؟

۱) مسائل زمان‌بندی ۲) $n$ وزیر
۳) زمان‌بندی رصدهای فضایی ۴) مسأله فروشنده دوره‌گرد

– در درخت زیر اعمال هرس آلفا – بتا، منجر به حذف چه شاخه‌هایی می‌شود؟




۱) ۹ و ۴
۲) ۳ و ۱۴
۳) ۲ و ۴
۴) ۹ و ۴ و ۳ و ۱۴

۷۸ – الگوریتم استنتاجی که فقط جملات ایجابی را به دست آورد ..... نامیده می‌شود.
۱) کامل ۲) صحیح ۳) ارضاپذیر ۴) تحلیل

۷۹ – کدام عبارت یک بند هورن است؟
۱) $\neg L_{1,1} \vee \neg Breeze \vee B_{1,1}$ ۲) $\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$
۳) $\neg L_{1,1} \wedge \neg Breeze \wedge B_{1,1}$ ۴) $\neg B_{1,1} \vee P_{1,2} \wedge P_{2,1}$

۸۰ – معنی جمله $\neg \exists x \neg Likes(x, Icecream)$ در زبان طبیعی کدام است؟
۱) هیچ کس بستنی دوست ندارد. ۲) همه بستنی دوست دارند.
۳) هیچ کس وجود ندارد که بستنی دوست داشته باشد. ۴) کسی وجود دارد که بستنی دوست ندارد.

۸۱ – بازنمایی جمله "پادشاه جان تاجی روی سرش داشت" در منطق مرتبه اول کدام گزینه است؟
۱) $\exists x\ Crown(x) \wedge OnHead(x, John)$ ۲) $\exists x\ Crown(x) \Rightarrow OnHead(x, John)$
۳) $\forall x\ Crown(x) \Rightarrow OnHead(x, John)$ ۴) $\forall x\ Crown(x) \vee OnHead(x, John)$

۸۲ – کدام یک از گزینه‌های زیر، عمومی‌ترین یکسان‌ساز دو عبارت زیر است؟ *Unify (Knows (Ali, x), Knows(y,t))*
۱) $\{y/ali, t/f(x)\}$ ۲) $\{y/ali, x/t\}$
۳) $\{y/ali, x/ali, t/ali\}$ ۴) این عبارات قابل یکسان‌سازی نیستند.

۸۳ – الگوریتم "زنجیره‌ای پیشرو در منطق مرتبه اول" بر روی کدام یک از پایگاه‌های دانش زیر مناسب است؟

۱) $P(x) \wedge \neg Q(x) \Rightarrow R(x)$, $P(A)$, $\neg Q(y)$
۲) $P(x) \vee Q(x) \Rightarrow R(x)$, $P(A)$, $Q(x)$
۳) $P(x) \wedge Q(x) \Rightarrow R(x) \vee A$, $P(A)$, $\neg Q(y)$
۴) $P(x) \wedge Q(x) \Rightarrow R(x)$, $P(A)$, $Q(y)$

۸۴ – در زبان پرولوگ (*prolog*) کدام یک از روش‌های استنتاجی زیر استفاده می‌شود؟
۱) استدلال زنجیره پیشرو ۲) استدلال زنجیره پسرو
۳) اثبات با برهان خلف ۴) استدلال پس‌گرد (*backtracking*)

۸۵ – الگوریتم ریتی جهت بهبود کدام یک از معایب الگوریتم زنجیره‌ای پیشرو در منطق مرتبه اول ایجاد شده است؟
۱) کاهش هزینه مرتبه انطباق الگو
۲) جلوگیری از انطباق‌های جزیی فراوانی که در هر تکرار ساخته و دور ریخته می‌شود.
۳) جلوگیری از تولید واقعیت‌های زیادی که با هدف مرتبط نیستند.
۴) کامل نبودن

۸۶ – انجام عمل واکنشی از قبیل "فوری عقب کشیدن دست از روی اجاق داغ" در کدام طبقه از تعاریف هوش مصنوعی مطرح می‌شود؟
۱) تفکر انسان‌گونه ۲) عملکرد انسان‌گونه ۳) تفکر عقلانی ۴) عملکرد عقلانی

۸۷ – عملکرد رضایت بخش عامل (کارگزار) از کدام دیدگاه قابل قبول‌تر است؟
۱) خود عامل ۲) طراح و سازنده عامل ۳) کاربر عامل ۴) تماشاگر عامل

۸۸ – کدام یک جزو ویژگی‌های محیط کار شطرنج زمان‌دار است؟
۱) ایستا ۲) پویا ۳) نیمه پویا ۴) تک عاملی

۸۹ – با تغییر راهبرد جستجو (*search strategy*) در (*Tree search*)، کدام جستجو را نمی‌توان تولید کرد؟
۱) اول سطح ۲) هزینه یکنواخت ۳) اول عمق ۴) جستجوی دوطرفه

۹۰ – بزرگترین مشکل جستجوی اول سطح (مثلا با $b=10$) کدام است؟
۱) کامل نبودن ۲) عدم بهینگی ۳) پیچیدگی زمانی ۴) پیچیدگی حافظه

۹۱ – با در نظر گرفتن لیست شرایط زیر، روش جستجوی هزینه یکنواخت در چه شرایطی کامل است؟
الف) در هر شرایطی ب) در شرایطی که هزینه اقدامات در یک سطح برابر باشد. ج) به شرطی که فاکتور انشعاب متناهی باشد. د) هزینه هر اقدام از $\varepsilon$ بزرگ‌تر باشد ( $\varepsilon$ مثبت است) هـ) در هر دو جهت از جستجوی اول سطح استفاده شود.

۴) 3 و 4 ۳) 3 و 2 ۲) 3 2

– استفاده از *Graph search* در چه مسائلی بسیار مؤثرتر از *Tree search* است؟

مسائل با حالت‌های تکراری زیاد در درخت ۲) در مسائل پیچیدگی زمانی بالا

در مسائل با پیچیدگی زمانی حافظه بالا ۴) در مسائل اکتشافی

– در محیط کاملا رویت‌پذیر و قطعی برای عامل (کارگزار) جاروبرقی "بدون حسگر" کدام گزینه صحیح نیست؟

حالت اولیه = مجموعه حالت باور شامل 8 حالت ممکن

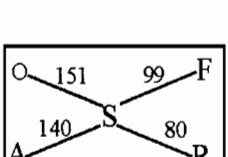
حالت هدف = دو حالت هدف مجزا وجود دارد (دو مجموعه حالت باور هدف هر کدام یک حالت هدف را در بر دارند)

به دلیل نداشتن حسگر عامل گاهی هدف را نخواهد یافت.

تنها 12 حالت باور دسترس‌پذیر وجود دارند.

– با توجه به اشکال مقابل (با هدف رسیدن به بخارست ($N$) از سیبیو ($S$))، الگوریتم $A^*$ بعد از $S$ کدام گره را اول بسط می‌دهد؟

| $n$ | $H(n)$ |
|---|---|
| $S$ | 253 |
| $O$ | 380 |
| $A$ | 366 |
| $F$ | 176 |
| $R$ | 193 |




O
F
R
A

– به کارگیری $A^*$ با *Graph search* به شرطی که $h(n)$ قابل قبول باشد ولی سازگار نباشد، چگونه است؟

کامل و نیمه بهینه ۲) کامل و بهینه ۳) غیرکامل و غیربهینه ۴) غیرکامل و بهینه

– کدام گزینه در مورد *RBFS* صحیح است؟

پیچیدگی حافظه آن $O(b+d)$ است.

تا حدودی از $IDA^*$ مؤثرتر است.

از تولید مجدد افراطی گره‌ها سود می‌برد.

اگر حافظه بیشتری در دسترس باشد (خیلی بیشتر) از، عملکرد بهتری خواهد داشت.

– اگر ضریب مؤثر انشعاب ($b^*$) برای 4 تابع هیوریستیک (آروینی) $h4$ ، $h3$ ، $h2$ ، $h1$ به ترتیب 1.5 ، 1.4 ، 1.3 ، 1.2 باشد
م یک بر بقیه ارجحیت دارد؟

$h4$ ۴ $h3$ ۳ $h2$ ۲ $h1$

– در محاسبه هیوریستیک (تابع آروینی) از هزینه راه حل زیر مسائل، بانک‌های اطلاعاتی الگو چگونه ساخته می‌شود؟

با جستجوی رو به عقب از حالت هدف و ثبت هزینه هر الگوی جدید

حل هر زیر مسئله و به دست آورد و ثبت هزینه آن هنگامی که با آن روبه‌رو می‌شویم.

تخمین هزینه زیر مسائل توسط مسئله تعدیل شده

با استفاده از یک الگوریتم فراگیری استقرایی

– الگوریتم‌های جستجوی محلی برای حل کدام مسئله مناسب نمی‌باشد؟

مسئله $n$ وزیر ۲) چیدمان دستگاه‌های کارخانه

به دست آوردن مسیر بهینه برای رسیدن به شهر بخارست ۴) مدارهای مجتمع

– برای حل مسئله $n$ وزیر کدام روش مؤثرتر است؟

تپه‌نوردی ۲) تپه‌نوردی با حرکات کناره

تپه‌نوردی اولین گزینه ۴) تپه‌نوردی با شروع مجدد تصادفی

– از دو فرد (کروموزوم، مقابل با عمل *Cross-over* و یک جهش (*mutation*) ، کدام فرد نمی‌تواند ایجاد شود؟

| | |
|---|---|
| 3 2 7 | 5 2 4 1 1 |

| | |
|---|---|
| 2 4 7 | 4 8 5 5 2 |

32748152
24752415
34748152
21752411

– کدام گزینه در مورد جستجوی سخت‌سازی شبیه‌سازی (شبه تابکاری) صحیح نیست؟

در اکثر موارد قادر به فرار از اکسترمم‌های محلی است.

نتیجه تلاش برای ترکیب تپه‌نوردی با یک راهپیمایی تصادفی می‌باشد.

یک حرکت بد با احتمال $e^{AE/T}$ امکان وقوع خواهد داشت.

۴) این الگوریتم همواره بهینه سراسری را خواهد یافت.

۱۰۳ – در *Online-DFS-Agent* ، عامل (کارگزار) در چه صورت جستجویش کامل شده است؟

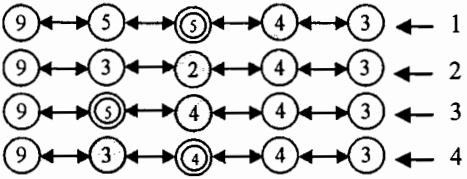
۱) در صورتی که از وضعیت فعلی اقدام اکتشاب نشده‌ایی وجود نداشته باشد (*Unexplored*[*s*] خالی باشد.)

۲) در صورتی که حالتی برای عقبگرد عامل وجود نداشته باشد (*Unbacktracked*[*s*] خالی باشد.)

۳) در حالی که هدف یافته شود (*Goal-test*[*s*] برابر *True* باشد.)

۴) در صورتی که شرایط (۲) یا (۳) برقرار باشد.

۱۰۴ – اگر الگوریتم *LRTA** در فضای حالت یک بعدی 4 تکرار انجام داده باشد و شکل مقابل فضای حالت را بعد از تکرارها نمایش دهد، ترتیب درست تکرارها را بیان کنید؟



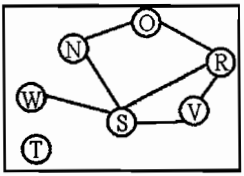

۱) 2و 4 و 3 و 1

۲) 2و 3 و 4 و 1

۳) 4و 3 و 2 و 1

۴) 4و 2 و 3 و 1

۱۰۵ – در مسئله رنگ‌آمیزی گراف مقابل (با سه رنگ)، با فرض این که در سطح اول درخت جستجوی *BT + MRV*، به متغیر *W* رنگ قرمز انتساب یافته باشد، اولین متغییری که در سطح بعد جهت رنگ‌آمیزی انتخاب خواهد شد، کدام است؟




۱) *S*

۲) *N* یا *V*

۳) *Q* یا *R*

۴) *T*

۱۰۶ – برای گراف سؤال ۱۰۵، در صورتی که در جستجوی پیشرو و مقادیر *R*، *G* و *B* به معنای رنگ‌های قرمز، سبز و آبی باشند و *R* به معنای انتساب رنگ قرمز به متغیر باشد، با توجه به گراف و جدول زیر، کدام کمان سازگار است؟

| *W* | *N* | *Q* | *R* | *S* | *V* | *T* |
|---|---|---|---|---|---|---|
| *R* | *R B* | *G* | *R B* | *B* | *R G B* | *R G B* |

۱) *R* به *S*

۲) *S* به *R*

۳) *N* به *S*

۴) *V* به *S*

۱۰۷ – اگر حداکثر ظرفیت‌های پروازهای 271 و 272 به ترتیب 165 و 385 نفر باشند (یعنی $Flight\,272 \in [0,385]$ و $Flight\,271 \in [0,165]$) با شرط این که مجموع تعداد مسافرینی که به وسیله هر دو پرواز انتقال می‌یابند باید 420 نفر باشد، کران‌های سازگار برای دو پرواز کدامند؟

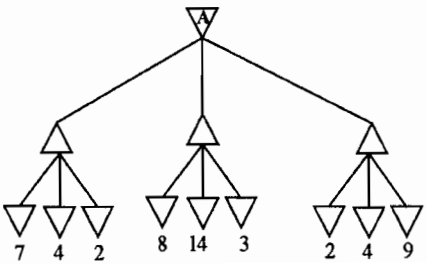
۱) $Flight\,272 \in [0,385]$ ، $Flight\,271 \in [0,165]$

۲) $Flight\,272 \in [0,255]$ ، $Flight\,271 \in [0,165]$

۳) $Flight\,272 \in [165,255]$ ، $Flight\,271 \in [0,165]$

۴) $Flight\,272 \in [255,385]$ ، $Flight\,271 \in [35,165]$

۱۰۸ – با فرض این که Δ به معنی Max و ∇ به معنای Min باشد و از روش هرس آلفا و بتا استفاده شود کدام گره‌ها بررسی نمی‌شوند؟



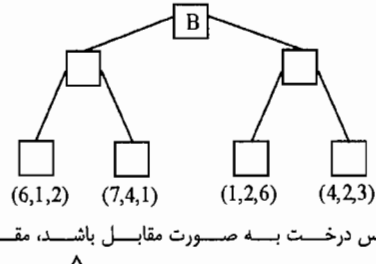

۱) 9 و 4

۲) 3 و 14

۳) 2 و 4

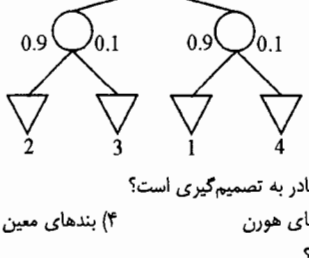
۴) 9 و 4 و 3 و 14

۱۰۹ – در درخت بازی چند نفره زیر، مقدار بردار مربوط به گره $B$ کدام است؟ مقادیر بردارها به این ترتیب چیده شده‌اند: $(A, B, C)$




۱) (6، 2، 3)
۲) (4، 6، 2)
۳) (1،2، 2)
۴) (1،2، 6)

۱۱۰ – اگر در بازی‌های دارای گره شانس درخت به صورت مقابل باشد، مقدارگره $\Delta$ کدام است؟ $(\text{Min} = \nabla, \text{chance} = O, \text{Max} = \Delta)$




۱) 2.1
۲) 1.3
۳) 4
۴) 3

۱۱۱ – الگوریتم زنجیره‌ای پیش‌رو (PL-FC-Entalis) با چه جملاتی قادر به تصمیم‌گیری است؟

۱) جملات گزاره‌ای دلخواه ۲) بندهای CNF ۳) بندهای هورن ۴) بندهای معین

۱۱۲ – کدام یک شکلی از استدلال هدفگرا (Goal-directed) می‌باشد؟

۱) الگوریتم زنجیره‌ای پس‌رو ۲) الگوریتم زنجیره‌ای پیش‌رو
۳) الگوریتم تحلیل (Resolution) ۴) TT-Entails

۱۱۳ – در الگوریتم DPLL اگر مدل دارای $A = \text{False}$ ، $D = \text{True}$ و $C = \text{True}$ باشد، کدام یک از بندهای سؤال زیر، بند واحد (unit_clause) است؟

$$\begin{cases} 1)\ A \vee \neg B \\ 2)\ \neg A \vee C \vee D \\ 3)\ \neg C \vee \neg B \vee \neg D \end{cases}$$

۱) 1
۲) 2
۳) 3
۴) 1 و 3

۱۱۴ – نتیجه حل دو بند زیر کدام بند است؟ (تحلیل Resolution)

$$1)\ \neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$$
$$2)\ \neg P_{2,1} \vee B_{1,1}$$

۱) $\neg B_{1,1} \vee P_{1,2} \vee B_{1,1}$
۲) $P_{1,2} \vee P_{2,1} \vee \neg P_{1,2}$
۳) $\neg B_{1,1} \vee P_{2,1} \vee B_{1,1}$
۴) گزینه‌ی (۱) یا (۲)

۱۱۵ – کدام یک از عبارات زیر با عبارت $R(x), F(A), F(z)$ قابل یکسان‌سازی (*Unification*) است؟ ( $x$ و $y$ متغیر و $A$ و $B$ مقادیر ثابت هستند.)

۱) $R(F(z), F(y), F(x))$ ۲) $R(F(z), x, F(B))$
۳) $R(z, F(z), F(B))$ ۴) $R(F(y), y, x)$

۱۱۶ – تست تورینگ مربوط به کدام تعریف هوش مصنوعی است؟

۱) عملکرد انسان‌گونه ۲) تفکر انسان‌گونه ۳) تفکر عقلانی ۴) عملکرد عقلانی

۱۱۷ – کدام یک از گزینه‌های زیر، صحیح نیست؟

۱) عقلانیت کارآیی مورد انتظار را بیشینه می‌کند. ۲) کمال، کارآیی واقعی را بیشینه می‌کند.
۳) عقلانیت مستلزم همه چیز دانی است. ۴) انتخاب عقلانی تنها به رشته ادراکات تا آن لحظه بستگی دارد.

۱۱۸ – کدام گزینه در مورد عاملی که فقط بر اساس دانش درونی عمل می‌کند، صحیح نیست؟

۱) ممکن است در یک محیط قطعی ساده موفق عمل کند. ۲) کاملا خودمختار است.
۳) بسیار آسیب‌پذیر است. ۴) امکان یادگیری ندارد.

۱۱۹ – کدام یک از محیط‌های کاری عامل‌های زیر، یک محیط ایستا می‌باشد؟

۱) تشخیص پزشکی ۲) جدول کلمات متقاطع ۳) شطرنج زمان‌دار ۴) معلم انگلیسی محاوره‌ای

۱۲۰ – در محیط‌های نیمه رؤیت‌پذیر کدام نوع عامل اغلب دچار حلقه‌های بی‌نهایت می‌شود؟

۱) واکنشی ساده ۲) واکنشی مبتنی بر مدل ۳) مبتنی بر هدف ۴) مبتنی بر سودمندی

۱۲۱ – کدام یک از اجزاء مفهومی یک عامل یادگیرنده نمی‌باشد؟

۱) عنصر یادگیری ۲) عنصر دانش ۳) منتقد ۴) مولد مسئله

۱۲۲ – در تدوین مسئله "حذف جزئیات از یک بازنمایی" را چه می‌نامند؟

۱) تجرید $(Abstraction)$ ۲) تعدیل شده $(Relaxation)$

۳) به اشتراک‌گذاری $(Sharing)$ ۴) معتبرسازی $(Valiation)$

۱۲۳ – کدام گزینه در رابطه با الگوریتم‌های جستجو صحیح است؟

۱) اگر فاکتور انشعاب محدود باشد، جستجوی اول عمق کامل است.

۲) جستجوی هزینه‌ی یکنواخت سربار قابل توجهی را در مقایسه با جستجوی طولانی کنندهٔ تکراری ایجاد می‌کند.

۳) میزان حافظه مورد استفاده در جستجوی اول عمق نسبت به جستجوی اول – سطح کمتر است.

۴) اگر فاکتور انشعاب متناهی باشد، آن‌گاه جستجوی هزینه یکنواخت کامل است ولی بهینه نیست.

۱۲۴ – الگوریتم ..... به جای یک حالت اطلاعات $k$ حالت را در حافظه نگهداری می‌کند؟

۱) جستجوی تپه‌نوردی ۲) الگوریتم تپه‌نوردی اتفاقی

۳) جستجوی پرتوی محلی ۴) جستجوی سخت‌سازی شبیه‌سازی شده

۱۲۵ – کدام گزینه صحیح است؟

۱) به جستجوهایی که از تابع ارزیاب $(f)$ برای انتخاب بهترین گره استفاده می‌کنند، آگاهانه گوییم.

۲) جستجوی حریصانه بهینه و کامل است.

۳) در هیوریستیک قابل قبولی سازگار است.

۴) به جستجویی که $h(n)$ بخشی از تابع ارزیاب $(f)$ آن باشد، آگاهانه گوییم.

۱۲۶ – کدام یک از شرایط زیر، شرط سازگاری یک تابع هیوریستیک را بیان می‌کنند؟ ( $n$ گره جاری، $n'$ گره پسین گره $n$ است که با اقدام $a$ تولید شده است – تابع $c$ هزینه گام رسیدن به $n'$ از $n$ با اقدام $a$ می‌باشد)

۱) $h(n)<c(n,a,n')+h(n')$ ۲) $h(n)\le c(n,a,n')+h(n')$

۳) $h(n)>c(n,a,n')+h(n')$ ۴) $h(n)\ge c(n,a,n')+h(n')$

۱۲۷ – $RBFS$ برای غلبه بر چه مشکلی در $A^*$ مطرح می‌شود؟

۱) کامل بودن ۲) عدم بهینگی ۳) پیچیدگی حافظه ۴) پیچیدگی زمانی

۱۲۸ – کدام گزینه صحیح نیست؟

۱) برای مکعب روبیک، بانک‌های پراکنده الگو بسیار مؤثر بوده است.

۲) اگر برای یک مسئله هیوریستیک‌های $h_1,\dots,h_m$ موجود باشد، $h(n)=\max\{h_1(n),\dots,h_m(n)\}$ بر همه برتری دارد.

۳) استفاده از بانک‌های اطلاعاتی پراکنده الگو سرعت پازل 24 تایی را به میزان قابل ملاحظه‌ای افزایش می‌دهد.

۴) هزینه یک راه حل بهینه برای یک مسئله تعدیل شده، یک هیوریستیک قابل قبول برای مسئله اصلی است.

۱۲۹ – برای حالتی که تعداد کمی بیشینه محلی و غلات وجود داشته باشد، کدام نوع از الگوریتم‌های تپه‌نوردی می‌تواند سریع‌تر یک راه‌حل خوب پیدا کند؟

۱) تپه‌نوردی اولین گزینه ۲) تپه‌نوری اتفاقی

۳) تپه‌نوردی با شروع مجدد تصادفی ۴) گزینه‌های (۱) و (۲) صحیح است.

۱۳۰ – کدام گزینه صحیح نیست؟

۱) جستجوی سخت‌سازی شبیه‌سازی شده با تکان‌های شدید آغاز شده و تدریجا از شدت تکان‌ها کاسته می‌شود.

۲) جستجوی پرتوی محلی همانند $k$ جستجوی محلی می‌باشد که به صورت موازی اجرا می‌شوند.

۳) در جستجوی پرتوی محلی، بهترین $k$ پسین جایگزین حالت قبلی می‌شود.

۴) جستجوی سخت‌سازی شبیه‌سازی شده، می‌تواند برای مسائلی همچون چیدمان $VLSI$ مورد استفاده قرار گیرد.

۱۳۱ – کدام گزینه در مورد جستجوی بر خط صحیح نیست؟

۱) این جستجو برای یک مسئله اکتشافی (حالت‌ها و اقدامات ناشناخته) ضروری می‌باشد.

۲) در بعضی موارد بهترین نسبت رقابتی دست‌یافتنی بی‌نهایت است.

۳) فضای حالتی که اقدامات قابل برگشت دارند مطمئنا قابل اکتشاف هستند.

۴) پیشروی $ONLIE.DFS.AGENT$ در هر فضای حالتی قابل استفاده است.

۱۳۲ – کدام گزینه در مورد $LRTA^*$ صحیح نیست؟
۱) جستجوی محلی برخطی است که محیط دارای $n$ حالت را حداکثر در $O(n^2)$ گام اکتشاف می‌کند.
۲) هزینه تخمینی رسیدن به هدف از طریق همسایه‌ای مانند $s'$ هزینه رسیدن به $H(s')+s'$ =
۳) این الگوریتم، برای فضاهای حالت نامتناهی، کامل است.
۴) همانند $ONLINE.DPS.AGENT$ با استفاده از جدول $result$ نقشه‌ای از محیط تهیه می‌کند.
۱۳۳ – کدام گزینه در مورد $CSP$ ها صحیح است؟
۱) اکثر $CSP$ها، جابه‌جایی‌پذیر هستند.
۲) یک الگوریتم استاندارد جستجو ممکن است به یک مسئله $CSP$ قابل اعمال نباشد.
۳) ساده‌ترین نوع $CSP$ شامل متغیرهایی گسسته با دامنه‌های نامحدود می‌باشند.
۴) یک راه حل برای $CSP$، می‌تواند یک انتساب ناکامل باشد.
۱۳۴ – در جستجوی پسگرد برای $CSP$ها، کدام گزینه برای انتخاب اولین متغیر جهت انتساب مقدار، مؤثرتر می‌باشد؟
۱) هیوریستیک $MRV$ ۲) سازگاری کمان ۳) انتشار محدودیت ۴) هیوریستیک درجه
۱۳۵ – برای مسئله با گرافی که دارای ساگزاری شدید مرتبه $k$ باشد، کدام گزینه نادرست است.
۱) دارای سازگاری مرتبه $1,...k-1,k$ می‌باشد.
۲) بدون انجام پسگرد قابل حل است.
۳) الگوریتمی که بخواهد سازگاری مرتبه $n$ را بررسی کند، حداکثر به مرتبه زمانی $O(n^2)$ نیاز دارد.
۴) می‌توان راه حل مسئله را در حداکثر زمان $O(nd)$ محاسبه نمود.
۱۳۶ – در مورد تصمیمات بلادرنگ ناقص در بازی‌ها کدام گزینه صحیح نیست؟
۱) هنگامی که زمان کافی برای جستجوی با تصمیمات بهینه نداریم، مورد استفاده قرار می‌گیرد.
۲) تابع ارزیاب، مقدار دقیق ارزش گره‌ها را محاسبه می‌کند.
۳) آزمون قطع جایگزین آزمون پایانی در جستجوی با تصمیمات بهینه می‌گردد.
۴) تابع ارزیاب جایگزین تابع سودمندی در جستجوی با تصمیمات بهینه می‌گردد.
۱۳۷ – کدام گزینه در مورد جستجوی بیشینه کمینه یا هرس آلفا بتا صحیح نیست؟
۱) از نوع اول عمق است.
۲) حالات تکراری در درخت می‌تواند هزینه جستجو را به طور نمایی افزایش دهد.
۳) با بررسی بهترین پسین‌ها فاکتور انشعاب مؤثر $\sqrt{b}$ خواهد بود.
۴) با بررسی تصادفی پسین‌ها پیچیدگی در حدود $O(b^n)$ می‌باشد.
۱۳۸ – در مورد تابع ارزیاب در بازی‌های دارای گره‌های شانس کدام مورد دقیق‌تر است؟
۱) این که به موقعیت‌های بهتر امتیازی بالاتری بدهی کافی است (دقت زیاد تخمین مؤثر نیست).
۲) اگر برای مقادیر سودمندی کران‌هایی قائل شویم. هرس آلفا بتا به شیوه مشابهی قابل اعمال است.
۳) ارزش یک گره شانس برابر بهترین پسین آن گره خواهد بود.
۴) مانند روش بیشینه کمینه می‌توان از هرس آلفا بتا استفاده کرد.
۱۳۹ – کدام گزینه صحیح نیست؟
۱) جمله $(A \Rightarrow B) \Rightarrow C$ با جمله $A \Rightarrow (B \Rightarrow C)$ تفاوت معنایی ندارد.
۲) الگوریتم استنتاجی که فقط جملات ایجابی را به‌دست می‌آورد، صحیح نامیده می‌شود.
۳) ترکیب $A \Leftrightarrow B \Leftrightarrow C$ به پرانتز نیازی ندارد.
۴) یک الگوریتم استنتاج کامل است در صورتی که بتواند هر جمله ایجاب شدنی را به‌دست آورد.
۱۴۰ – کدام گزینه در مورد ایجاب ( $\models$ ) صحیح نیست؟
۱) $\alpha \models \beta$ اگر و فقط اگر در هر مدلی که $\alpha$ در آن درست است، $\beta$ نیز درست باشد.
۲) $\alpha = \beta$ اگر و فقط اگر $\alpha \models \beta$ و $\beta \models \alpha$
۳) $\alpha \models \beta$ اگر و فقط اگر $\alpha \wedge \neg\beta$ ارضاء پذیر باشد.
۴) $\alpha \models \beta$ اگر و فقط اگر $\alpha \Rightarrow \beta$ معتبر ($valid$) باشد.
۱۴۱ – اگر $KB$ تهی باشد و $TELL(KB,S_1),...,TELL(KB,S_n)$ را انجام دهیم آنگاه $KB$ معادل کدام گزینه می‌باشد؟
۱) $S_1 \wedge ... \wedge S_n$ ۲) $S_1 \vee ... \vee S_n$
۳) $S_1 \Rightarrow ... \Rightarrow S_n$ ۴) $S_1 \oplus ... \oplus S_n$
۱۴۲ – کدام قاعده استنتاج به همراه هر الگوریتم جستجوی کامل می‌تواند به تنهایی یک الگوریتم استنتاج کامل ایجاد کند؟
۱) قیاس استثنایی ($Modus\ ponens$) ۲) حذف عطف ($AND\ Elimination$)

۳) تحلیل (*Resolution*) ۴) دمورگان (*de Morgan*)

۱۴۳ – شرط خروج الگوریتم تحلیل برای $KB \models \alpha$ چیست؟

۱) هیچ بند جدیدی که بتواند اضافه شود وجود نداشته باشد که در این صورت $\alpha$ از $KB$ نتیجه نمی‌شود.

۲) بندی تهی ایجاد شود که در این صورت $\alpha$ از $KB$ نتیجه می‌شود.

۳) بندی تهی ایجاد شود که به معنی عدم نتیجه‌گیری $\alpha$ از $KB$ است.

۴) گزینه‌های (۱) و (۲)

۱۴۴ – کدام گزینه یک بند هورن نمی‌باشد؟

۱) $\neg P \vee \neg Q \vee R$ ۲) $\neg P \wedge Q \Rightarrow R$ ۳) $\neg P \vee Q \vee \neg R$ ۴) $P \wedge Q \Rightarrow R$

۱۴۵ – الگوریتم زنجیره‌ای پس‌رو، شکلی از ..... است.

۱) منطق مرتبه اول ۲) منطق فازی ۳) استدلال منطقی ۴) استدلال هدف‌گرا

۱۴۶ – یک عامل عقلانی چه ویژگی‌هایی باید داشته باشد؟

۱) دانش قبلی در مورد محیط ۲) قدرت استدلال ۳) توانایی تعامل با محیط ۴) قدرت تصمیم‌گیری

۱۴۷ – کدام یک از موارد زیر در مورد ویژگی‌های محیط کار درست نیست؟

۱) قطعی در مقابل اتفاقی ۲) مرحله‌ای در مقابل ترتیب

۳) وابسته در مقابل غیروابسته ۴) تک عاملی در مقابل چند عاملی

۱۴۸ – عامل یا *agent* تلفیقی از ..... می‌باشد.

۱) هوش + تحرک ۲) برنامه + معماری ۳) حس‌گر + برنامه ۴) استنتاج + حسگر

۱۴۹ – کدام یک از موارد زیر در مورد رویکرد جدول‌گرا در برنامه‌های عامل درست است؟

۱) هیچ عامل فیزیکی در این دنیا فضای ذخیره‌سازی این جدول را ندارد.

۲) طراح دقت کافی برای ساختن جدول را داراست.

۳) عامل می‌تواند تمام داده‌های صحیح را از طریق جدولش یاد بگیرد.

۴) رویکرد جدول‌گرا در ساخت عامل، موجب موفقیت برنامه عامل خواهد شد.

۱۵۰ – کدام مورد از انواع عامل‌ها نمی‌باشد؟

۱) عامل واکنشی ساده ۲) عامل مبتنی بر هدف ۳) عامل مبتنی بر هوش ۴) عامل مبتنی بر مدل

۱۵۱ – طراحی عامل به چه شکلی است؟

۱) تدوین هدف، روال جستجو، اجرا ۲) مقداردهی اولیه، جستجو، اجرا

۳) مشاهدات، استنتاج اجزاء ۴) مشاهدات، استنتاج، ارائه راه حل

۱۵۲ – کدام یک از روش‌های ارزیابی الگوریتم محسوب نمی‌شود؟

۱) کامل بودن ۲) قابل درک بودن ۳) پیچیدگی زمانی ۴) پیچیدگی فضایی

۱۵۳ – برای طراحی عامل "راننده تاکسی" کدام یک از برنامه‌های عامل زیر مناسب‌تر خواهد بود؟

۱) عامل واکنشی ساده ۲) عامل همراه با تثبیت تغییرات دنیا

۳) عامل هدف‌گرا ۴) عامل مبتنی بر سودمندی

۱۵۴ – نحوه عمل عامل‌های واکنشی ساده در محیط‌های پیوسته چگونه است؟

۱) در بسیاری از موارد تصمیم‌گیری کامل و دقیق نیست، زیرا عامل ورودی پیوسته محیط را به کمیتی گسسته تبدیل نموده و بر اساس آن عمل خواهد کرد.

۲) کامل است، زیرا می‌توان برای ورودی، رفتاری را در عامل تعیین نمود.

۳) اگر محیط قابل دسترس باشد، عملکرد آن کامل خواهد بود.

۴) عملکرد عامل بستگی به پیوسته بودن محیط ندارد.

۱۵۵ – کدام یک از گزینه‌های زیر تفاوت معیار کارآیی و تابع سودمندی عامل را بیان می‌کند؟

۱) معیار کارآیی برای تصحیح عملکرد عامل استفاده می‌شود و تابع سودمندی در طراحی ساختار عامل دخالت دارد.

۲) معیار کارآیی چگونگی و میزان موفقیت یک عامل را نشان می‌دهد و تابع سودمندی میزان سودمندی یک وضعیت را در دنیا و از دیدگاه عامل برمی‌گرداند.

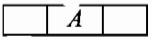
۳) معیار کارآیی، پارامتر مقایسه بین دو عامل است و تابع سودمندی پارامتر مقایسه دو وضعیت در رسیدن به هدف.

۴) معیار کارآیی، موفقیت عامل است و تابع سودمندی درجه وضعیت عامل.

۱۵۶ – فرض کنید $A$ یک جارو برقی اتوماتیک است محیط این جاروبرقی، مطابق شکل زیر، از سه خانه کنار هم تشکیل شده است. این جاروبرقی می‌توانند از هر یک از این خانه‌ها با انجام یک حرکت به خانه مجاور نقل مکان نماید و زباله موجود در آن خانه را (در صورت

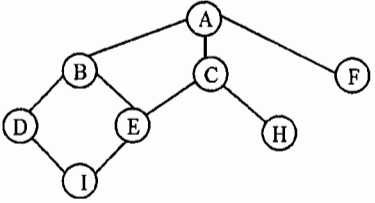
وجود) جمع‌آوری کند. با توجه به این که این جاروبرقی برای جمع‌آوری هر زباله باید در همان خانه‌ای که زباله در آن وجود دارد، قرار بگیرد، فضای حالت این مساله دارای چند وضعیت منحصر بفرد است؟

۱) 23
۲) 9
۳) 63
۴) 81



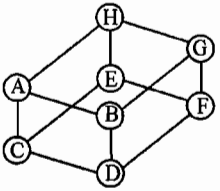

۱۵۷ – اگر در گراف زیر جستجو در عمق (*Depth first search*) را از راس $C$ شروع کنیم، کدام گره‌ها به ترتیب از چپ به راست رؤیت می‌شوند؟ (فرض کنید فرزندان یک گره بر اساس ترتیب حروف الفبا انتخاب می‌شوند.)

۱) *ABCDEFHI*
۲) *CABDIEFH*
۳) *CAEHBFID*
۴) *CABDEHIF*



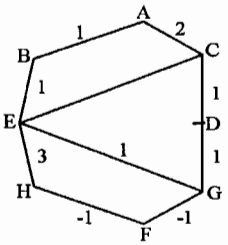

۱۵۸ – پیمایش اول عمق (*Depth first*) برای گرف مقابل با شروع از گره $A$ کدام است؟

۱) *ABDCEFHG*
۲) *ACEFDBHG*
۳) *ACDFBEGH*
۴) *AHGBDCEF*




۱۵۹ – کدام یک از الگوریتم‌های جستجو زیر از لحاظ زمان و حافظه بر روی گراف فوق بهتر عمل می‌کند؟ (گره $A$ شروع، گره $G$ گره هدف می‌باشد. هزینه عملگرها برای رفتن از یک گره به گره دیگر روی یال‌ها نوشته شده است.)

۱) جستجوی اول سطح
۲) جستجوی اول عمق
۳) جستجوی با هزینه یکسان
۴) جستجوی عمقی محدود شده با حداکثر عمق 2




۱۶۰ – کدام یک از الگوریتم‌های زیر کامل نمی‌باشند؟

۱) دو طرفه ۲) اول عمق ۳) اول سطح ۴) عمیق شونده تکراری

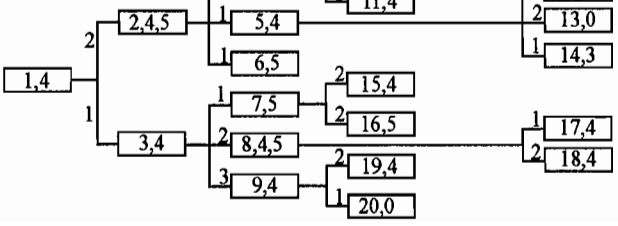
۱۶۱ – کدام یک از الگوریتم‌های زیر از لحاظ فضا مناسب‌تر است؟

۱) دو طرفه ۲) اول سطح ۳) با عمق محدود ۴) اول عمق

۱۶۲ – نقطه ضعف روشن $IDA^*$ (*Iterative Deeping* $A^*$) در چیست؟

۱) کامل نبودن ۲) دوباره کاری ۳) کارآیی پایین ۴) مصرف حافظه زیاد

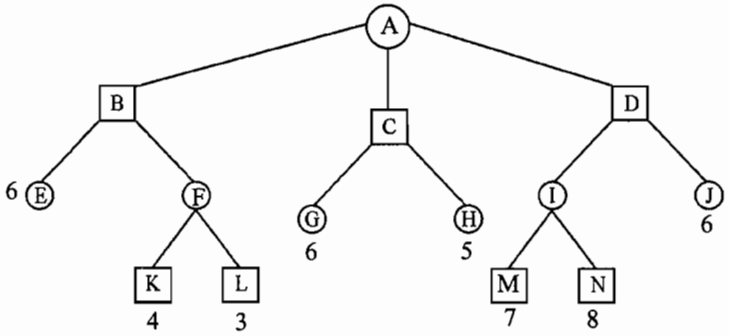
۱۶۳ – در درخت تصمیم‌گیری زیر با استفاده از جستجوی $A^*$ کدام گزینه شماره گره‌های مورد بررسی را مشخص می‌کند؟ توجه کنید هزینه هر گره در کنار شماره آن و هزینه هر شاخه روی آن نوشته شده است (در هر گره اولین عدد شماره گره و دومین عدد هزینه می‌باشد.)



۱) 1 و 2 و 3 و 4 و 5 و 6 و 12 و 13 و 14
۲) 1 و 2 و 3 و 7 و 8 و 9 و 19 و 20
۳) 1 و 2 و 3 و 7 و 8 و 9 و 17 و 18 و 19 و 20
۴) 1 و 2 و 3 و 7 و 8 و 9 و 4 و 5 و 6 و 12 و 13 و 14

۱۶۴ – کدام یک از گزینه‌های زیر درست است؟
۱) به طوری کلی در مواردی که حافظه در اختیار داریم بهتر است از روش پیمایش اول سطح برای جستجو استفاده کنیم.
۲) شرایط لازم برای آن که $A^*$، رسیدن به جواب بهینه را تضمین کند، به دامنه مسأله وابسته است.
۳) الگوریتم پیمایش اول عمق همواره رسیدن به جواب را تضمین نمی‌کند.
۴) الگوریتم $A^*$ در بدترین حالت عملکردی مشابه $Best\ first$ دارد.

۱۶۵ – اگر با استفاده از روش جستجوی $Minmax$ درخت جستجوی زیر پیمایش شود، با استفاده از روش هرس آلفا بتا کدام یک از گره‌های این درخت ملاقات نخواهند شد؟ (دوایر: معرف گره $Min$، مربع‌ها: معرف $Max$ و اعداد کنار گره‌های برگ: معرف ارزش این گره‌هاست.)

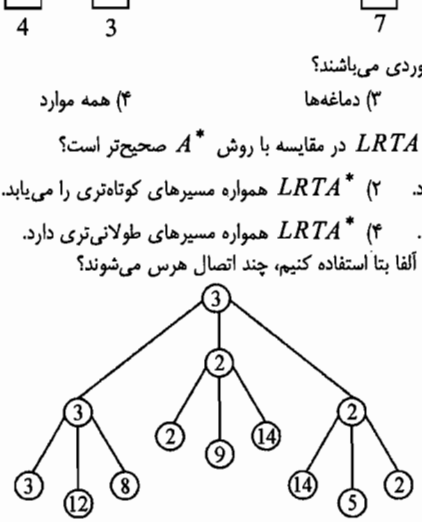



۱) $\{L, H, N\}$
۲) $\{L, N, J\}$
۳) $\{L, H, J\}$
۴) $\{K, N, H\}$

۱۶۶ – کدام یک از موارد زیر دلایل گیرکردن جستجوی تپه‌نوردی می‌باشند؟
۱) بیشینه‌های محلی ۲) فلات‌ها ۳) دماغه‌ها ۴) همه موارد

۱۶۷ – کدام یک از موارد زیر در خصوص روش جستجوی $LRTA^*$ در مقایسه با روش $A^*$ صحیح‌تر است؟
۱) $LRTA^*$ اغلب تمایل بیشتری به ادامه مسیر جاری دارد. ۲) $LRTA^*$ همواره مسیرهای کوتاه‌تری را می‌یابد.
۳) $LRTA^*$ اغلب تمایل کمتری به ادامه مسیر جاری دارد. ۴) $LRTA^*$ همواره مسیرهای طولانی‌تری دارد.

۱۶۸ – درخت $Minmax$ زیر را در نظر بگیرید. اگر از هرس آلفا بتا استفاده کنیم، چند اتصال هرس می‌شوند؟




۱) 2 اتصال
۲) 4 اتصال
۳) 6 اتصال
۴) هیچ اتصالی هرس نمی‌شود.

۱۶۹ – کدام یک از موارد زیر صحیح نیست؟
۱) الگوریتم جستجوی پرتوی محلی، اطلاعات $K$ حالت را به جای یک حالت نگه می‌دارد.
۲) الگوریتم ژنتیک از پیوند و جهش تصادفی روی جمعیت برای تولید نسل بعد کمک می‌گیرند.
۳) $Online-Dfs-Agent$ قابلیت بازگشت به عقب ندارد.
۴) $IDA^*$ و $RBFS$ از حافظه خیلی کم استفاده می‌کنند برای دسترسی فضای بیشتری از حافظه $MA^*$ ($A^*$ مقید به حافظه) و $SMA^*$ ($MA^*$ ساده شده) استفاده می‌شود.

۱۷۰ – فرض کنید مجموعه گزاره $\{q \Rightarrow False, q \Leftrightarrow (P_2 \vee P_3)\}$ درست است. آن‌گاه کدام یک از عبارات زیر را می‌توان از مجموعه گزاره بالا نتیجه گرفت؟
۱) $q$ ۲) $P_2$ و $P_3$ ۳) $P_2 \vee P_3$ ۴) $q$ و $(P_2 \vee P_3)$

۱۷۱ – با این فرض که متغیر $X$ در $Q$ به صورت آزاد ($free$) ظاهر نشده است، مقدار کدام یک از عبارت زیر در منطق مسندات ($PREDICATE\ LOGIC$) نادرست است؟

۱) $(\exists x\ (p(x)\to Q))\to(\forall x\ p(x)\to Q)$ ۲) $(\forall x\ p(x)\to Q)\to(\forall x\ (p(x)\to Q)$

۳) $(\exists x\ p(x)\to Q)\to(\forall x\ p(x)\to Q)$ ۴) $(\exists x\ p(x)\to Q)\to(\exists x\ (p(x)\to Q))$

۱۷۲ – در یک پایگاه دانش، مجموعه دانش زیر موجود است، کدام یک از گزینه‌ها از این مجموعه قابل استنتاج منطقی است؟

$$\begin{cases}\neg K(x)\vee M(x)\\ \neg K(x)\wedge Q(x)\\ L(N(x)\wedge M(g)\wedge P(x,y))\\ \neg Q(ali)\\ K(amir)\\ L(ali)\end{cases}$$

۱) $P(amir,ali)$

۲) $Q(ali)\wedge M(amir)$

۳) $N(amir)\vee\neg K(ali)$

۴) $K(amir)\Rightarrow P(ali,amir)$

۱۷۳ – ترجمه جمله "هر کس یک و فقط یک مادر دارد" به منطق مرتبه اول چیست؟

۱) $\forall x,\exists y\ mother(x,y)\wedge(\forall z\ mother)(m,z)\Rightarrow y=z)$

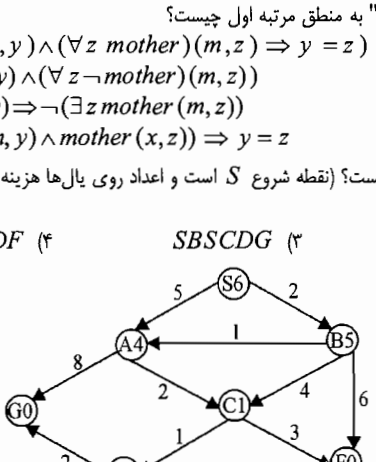
۲) $\forall x,\exists y\ mother(x,y)\wedge(\forall z\neg mother)(m,z))$

۳) $\forall x,y\ (mother(x,y)\Rightarrow\neg(\exists z\ mother(m,z))$

۴) $\forall x,y,z\ (mother(m,y)\wedge mother(x,z))\Rightarrow y=z$

۱۷۴ – در گراف مقابل حاصل جستجو با روش $A^*$ چیست؟ (نقطه شروع $S$ است و اعداد روی یال‌ها هزینه واقعی و اعداد داخل دایره‌ها مقدار $h$ گره مورد نظر است.)

۱) $SBF$ ۲) $SBAG$ ۳) $SBSCDG$ ۴) $SBACDF$



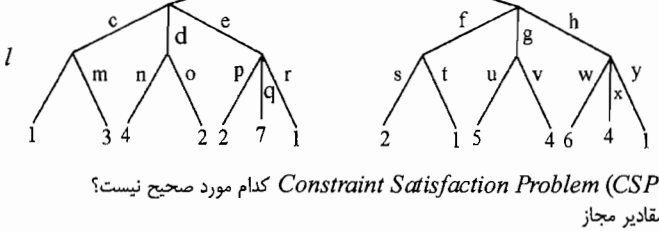

۱۷۵ – در درخت بازی زیر اگر از هرس آلفا بتا استفاده شود، کدام شاخه‌ها حذف خواهند شد؟ (فرض می‌شود حذف شاخه غیرانتهایی به طور ضمنی حذف تمام زیر درخت تحت آن را به همراه دارد و ذکر شاخه‌های زیر درخت لازم نیست.)

۱) $c-g-h$

۲) $o-r-v-h$

۳) $o-p-r-v-n-y$

۴) $o-r-g-h$




۱۷۶ – در مساله ارضاء محدودیت‌ها $Constraint\ Satisfaction\ Problem\ (CSP)$ کدام مورد صحیح نیست؟

۱) انتخاب متغیری با بزرگ‌ترین دامنه مقادیر مجاز

۲) انتخاب متغیری با کوچک‌ترین دامنه مقادیر مجاز

۳) انتخاب متغیری که کمترین میزان مقادیر مجاز را از دامنه سایر متغیرها حذف کند.

۴) انتخاب متغیری که بیشترین میزان مقادیر مجاز را از دامنه سایر متغیرها حذف کند.

۱۷۷ – کدام یک از گزینه‌های زیر نتیجه منطقی جملات مقابل است؟

۱) بهزاد قاتل ماهی است.

۲) حمید دوستدار گربه است.

۳) بهزاد قاتل پویا است یا حمید قاتل بهزاد است.

۴) حمید قاتل پویا است یا گربه قاتل پویا است.

$\exists x\, cat(x) \wedge owns(Hamid, x)$

$\forall x (\exists y, cat(y) \wedge owns(x, y) \Rightarrow animal - lover(x)$

$\forall x \forall y (Animal - lover(x) \wedge animal(y) \Rightarrow \neg Kills(x, y))$

$kills(Hamid, Puya) \vee kills(Behzad, Puya)$

$Fish(pupu)$

$\forall x (Fish(x) \Rightarrow Animal(x))$

$\neg kills(Hamid, Behzad)$

۱۷۸ – روش جستجوی $A^*$، تحت چه شرایطی یافتن پاسخ بهینه را تضمین می‌کند؟

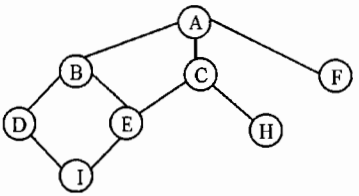
۱) اصلا روش‌های ابتکاری از جمله $A^*$ قادر به یافتن پاسخ بهینه نیستند.

۲) شرایط لازم برای این که $A^*$ پاسخ بهینه را تضمین کند، به دامنه مسئله بستگی دارد.

۳) در صورتی که تابع ابتکاری مورد استفاده، فاصله وضعیت‌های مختلف تا وضعیت هدف را، هرگز بیشتر از مقدار واقعی تخمین نزند.

۴) در صورتی که تابع ابتکاری مورد استفاده، فاصله وضعیت‌های مختلف تا وضعیت هدف را حداکثر به اندازه مقدارکوچک دلتا و بیشتر از مقدار واقعی تخمین بزند.

۱۷۹ – اگر در گراف جستجو در عمق (*depth first search*) را از رأس $C$ شروع کنیم، کدام گره‌ها به ترتیب از چپ به راست رؤیت (*Visit*) می‌شوند؟ (فرض کنید فرزندان گره بر اساس ترتیب حروف الفبا انتخاب شوند.)



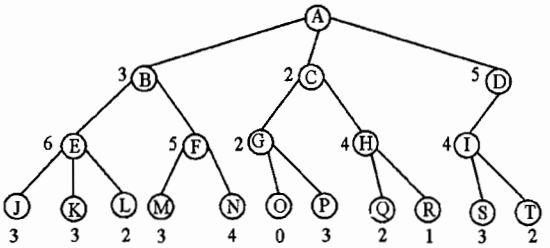

۱) $ABCDEFHI$

۲) $CABDIEFH$

۳) $CAEHBFID$

۴) $CABDEHIF$

۱۸۰ – در درخت جستجوی زیر به شرطی که گره $O$، گره هدف باشد، بر اساس الگوریتم جستجو $Best - First$، ترتیب دیدن گره‌ها کدام است؟




۱) $A, B, C, D, E, F, G, H, O$

۲) $A, B, C, D, E, F, G, O$

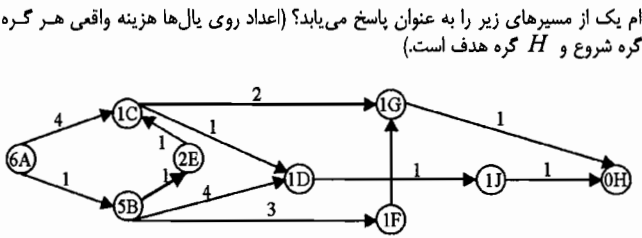
۳) $A, B, E, J, K, L, F, M, N, C, G, O$

۴) $A, B, C, D, E, F, G, H, I, J, K, L, M, N, O$

۱۸۱ – نقطه ضعف روش $IDA^*$ (*Iterative Deepening* $A^*$) در چیست؟

۱) کامل نبودن ۲) دوباره‌کاری ۳) کارایی پایین ۴) مصرف حافظه زیاد

۱۸۲ – الگوریتم $A^*$ در گراف زیر کدام یک از مسیرهای زیر را به عنوان پاسخ می‌یابد؟ (اعداد روی یال‌ها هزینه واقعی هر گره هزینه تخمینی آن گره تا گره هدف است. $A$ گره شروع و $H$ گره هدف است.)




۱) $ACGH$

۲) $ACDJH$

۳) $ABFGH$

۴) $ABCDJH$

۱۸۳ – روش جستجوی تولید آزمون (*generate and test*) برای حل کدام یک از مسائل زیر مناسب‌تر است؟

۱) شطرنج ۲) معمای هشت (*Eight puzzle*)

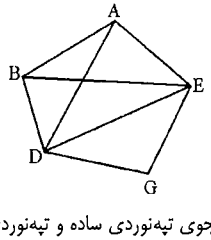
۳) هشت وزیر ۴) فروشنده دوره‌گرد

۱۸۷ – اگر بخواهیم کلید پاسخ‌های یک مسئله ارضاء محدودیت (*Constriantsatisfaction*) را بیابیم. کدام یک از روش‌های جستجوی زیر مناسب‌ترین است؟

۱) عمق اول ۲) سطح اول ۳) تپه‌نوردی ۴) $A^*$

۱۸۸ – در شکل مقابل ( $A$ نقطه شروع و $G$ هدف است) حاصل جستجو با کدام روش به مسیر $ABDG$ است؟

۱) جستجوی عرض اول
۲) جستجوی عمق اول
۳) جستجوی $A^*$
۴) جستجوی تپه‌نوردی

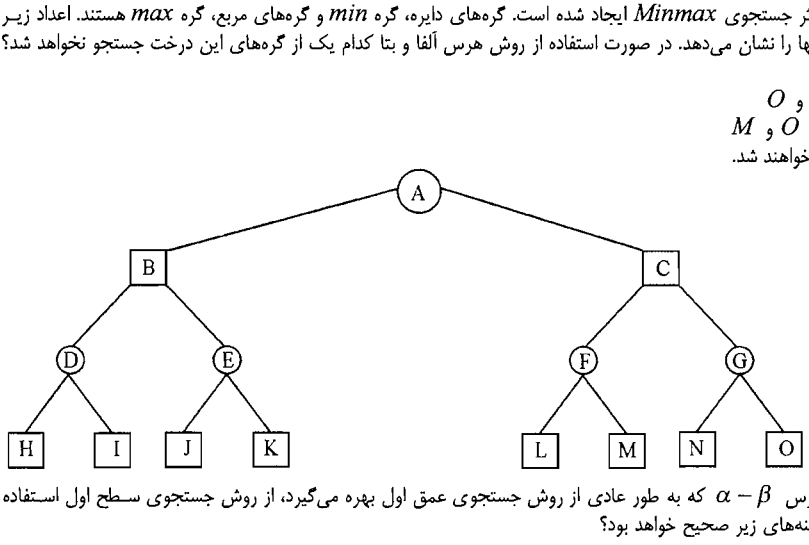



۱۸۹ – کدام یک از موارد زیر در مورد مقایسه دو روش جستجوی تپه‌نوردی ساده و تپه‌نوردی از تندترین شیب صحیح است؟

۱) تپه‌نوردی ساده کمتر در ماکزیمم محلی قرار می‌گیرد.
۲) تپه‌نوردی ساده با سرعت بیشتری حرکت می‌کند اما مسیر طولانی‌تری را می‌یابد.
۳) تپه‌نوردی از تندترین شیب با سرعت بیشتری حرکت می‌کند اما حافظه بیشتری نیز مصرف می‌کند.
۴) تپه‌نوردی از تندترین شیب پاسخ بهینه را می‌یابد، در حالی که تپه‌نوردی ساده این طور نیست.

۱۹۰ – درخت زیر در اثر جستجوی *Minmax* ایجاد شده است. گره‌های دایره، گره *min* و گره‌های مربع، گره *max* هستند. اعداد زیر گره‌های برگ، ارزش آنها را نشان می‌دهد. در صورت استفاده از روش هرس آلفا و بتا کدام یک از گره‌های این درخت جستجو نخواهد شد؟

۱) گره‌های $K$ و $M$
۲) گره‌های $K$ و $M$ و $O$
۳) گره‌های $K$ و $G$ و $O$ و $M$
۴) کلیه گره‌ها جستجو خواهند شد.




۱۹۱ – اگر در روش هرس $\alpha - \beta$ که به طور عادی از روش جستجوی عمق اول بهره می‌گیرد، از روش جستجوی سطح اول استفاده نماییم. کدام یک از گزینه‌های زیر صحیح خواهد بود؟

۱) جستجو پاسخ بهینه مسئله را می‌یابد. ۲) تغییری در عملکرد روش صورت نمی‌گیرد.
۳) کارایی جستجو د رحد روش *Minmax* کاهش می‌یابد. ۴) کارایی روش افزایش می‌یابد، اما حافظه بیشتری مصرف می‌گردد.

۱۹۲ – کدام یک از قالب‌های نمایش، سیستم‌های خبره قاعده‌گرا را بنیان نهاده است؟

۱) قالب نمایش منطقی ۲) قالب نمایش رویه‌ای ۳) قالب نمایش شبکه‌ای ۴) قالب نمایش ساخت‌یافته

۱۹۳ – پاسخ به سؤال "چه میزان حافظه برای جستجو صرف خواهد شد؟" کدام معیار در استراتژی جستجو را تعیین می‌کند؟

۱) بهینه بودن ۲) کامل بودن ۳) پیچیدگی زمانی ۴) پیچیدگی فضا

۱۹۴ – کدام جمله در خصوص جستجوی حریصانه صحیح است؟

۱) مانند جستجوی عمقی است، اما زمانی که در پیمودن یک مسیر به بن‌بست می‌رسد، برمی‌گردد.
۲) این جستجو بهینه و کامل نیست.
۳) جستجوی حریصانه تمام گره‌ها را در حافظه نگه می‌دارد. بنابراین پیچیدگی فضای آن مشابه پیچیدگی زمانی آن است.
۴) هر سه مورد

۱۹۵ – قابلیت اداره کردن داده‌های گم شده یا غیرقطعی در کدام برنامه‌نویسی صورت می‌گیرد؟

۱) برنامه‌نویسی مبتنی بر دانش ۲) برنامه‌نویسی شیءگرا ۳) برنامه‌نویسی رویه‌ای ۴) برنامه‌نویسی چند لایه

۱۹۳ – مهم‌ترین انتقاد به تست تورینگ به دلیل استفاده از کدام روش حل مسئله است؟

۱) اکتشافی ۲) غیرسمبلیک ۳) غیرالگوریتمی ۴) سمبلیک

۱۹۴ – به منظور قبولی در تست تورینگ سیستم باید چه ویژگی‌ای داشته باشد؟

۱) فهم احساس انسان ۲) درک هوشمندی ۳) فهم زبان طبیعی ۴) درک شرایط محیطی

۱۹۵ – عملگرهای دستکاری بر روی پایگاه دانش را چه می‌نامند؟

۱) موتور انبساط ۲) هوشمندی ۳) قوانین علمی ۴) دامنه دانش

۱۹۶ – کارایی یک جستجو در پاسخ به کدام یک از سؤالات زیر ارزیابی می‌گردد؟

۱) آیا این جستجو راه حلی پیدا می‌کند؟

۲) آیا راه حل به‌دست آمده بهترین پاسخ است؟

۳) هزینه جستجو از نظر زمانی و حافظه مورد نیاز در یافتن راه حل چیست؟

۴) هر سه مورد

۱۹۷ – در کدام جستجو است که استراتژی هیچ اطلاعی از تعداد مراحل یا هزینه جستجو ندارد و تنها قادر به تشخیص دادن حالت هدف از غیر حالت هدف است؟

۱) کورکورانه ۲) $\alpha - \beta$ ۳) $A^*$ ۴) حریصانه

۱۹۸ – کدام روش حل مسئله $AI$ را از سایر علوم کامپیوتر و مهندسی متمایز می‌نماید؟

۱) روش الگوریتمی ۲) روش ساده‌سازی ۳) روش اکتشافی ۴) روش محاسباتی

۱۹۹ – مهم‌ترین انتقاد به تست تورینگ به دلیل استفاده از کدام روش حل مسئله است؟

۱) اکتشافی ۲) غیرسمبلیک ۳) غیرالگوریتمی ۴) سمبلیک

۲۰۰ – کدام یک از موارد زیر از قابلیت‌های ضروری برای هوشمندی نیست؟

۱) معتادان به پیام‌های مبهم یا نادرست ۲) پیدا کردن شباهت‌ها، ولو این که موقعیت‌ها متفاوت باشند.

۳) اثبات قضیه ریاضی ۴) پیدا کردن شباهت‌ها، ولو این که موقعیت‌ها متفاوت نباشند.

۲۰۱ – کدام مورد از مزایای ارائه سمبولیک نیست؟

۱) تحمل‌پذیری بیشتر نسبت به نوفه یا نویز دارند. ۲) آگاهی و دانش با جملاتی به زبان رسمی بیان می‌شود.

۳) سازنده سیستم می‌تواند چیزی را که سیستم می‌داند بخواند. ۴) خواندن ارائه و فهمیدن معنی دانش امکان‌پذیر است.

۲۰۲ – کدام مورد ازمنابع اصلی دانش نیست؟

۱) مطالب نوشتاری ۲) مثال‌ها ۳) قوانین فیزیکی ۴) افراد خبره

۲۰۳ – طبق نظریه مایلوپولس و لوسک کدام یک از قالب‌های زیر برای ارائه دانش مطرح نیستند؟

۱) قالب نمایش منطقی ۲) قالب نمایش رویه‌ای ۳) قالب نمایش شبکه‌ای ۴) قالب نمایش شیءگرا

۲۰۴ – کدام عبارت زیر در مورد قالب‌ها صحیح عنوان نشده است؟

۱) قالب‌ها شبکه‌های معنایی را از راه‌های مختلف گسترش می‌دهند. مهم‌ترین آنها سازمان‌دهی دانش در ساختارهاست و این موضوع مهمی برای پایگاه دانش است.

۲) الحاق رویه‌ای، تا وقتی که دانش معین با نمایش‌ها به خوبی وفق داده نشده‌اند، یک خصوصیت مهم ویژه از قالب‌هاست.

۳) نمایش دادن دانش با سیستم قالب، اگر چه دارای اطلاعات ناتمام می‌باشد، حداقل تا حدی به ما اجازه استدلال کردن و استنتاج سریع حقایقی که به طور صریح مشاهده و آشکار نشده‌اند را می‌دهند.

۴) یکی از مشکلات نمایش قالبی، دشواری تعیین الگوریتم خطا برای یک قالب می‌باشد.

۲۰۵ – کدام جمله در خصوص اسکریپت‌ها صحیح نیست؟

۱) اسکریپت‌ها توانایی جهت پیشگویی وقایع را دارند.

۲) اسکریپت‌ها عمومیت بیشتری نسبت به قالب‌ها دارند.

۳) ممکن است اسکریپت‌ها برای ارائه کلیه انواع دانش مناسب نباشد.

۴) در اسکریپت‌ها تفسیر منسجم انفرادی از مجموعه‌ای از مشاهدات ساخته می‌شود.

۲۰۶ – کدام یک از موراد زیر جزو اجزای سیستم‌های جستجو نیست؟

۱) پایگاه داده ۲) استدلال پسین ۳) عملگرها ۴) استراتژی کنترل

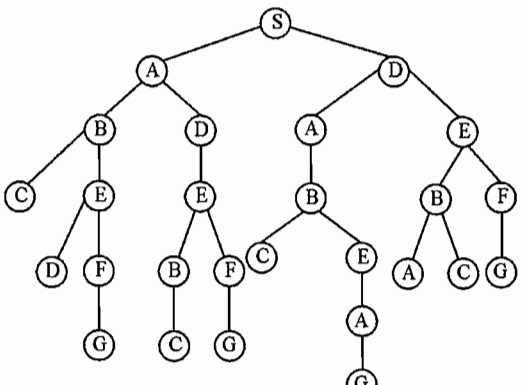
۲۰۷ – شکل زیر نمایی از کدام روش جستجو است؟

۱) کورکورانه

۲) سطحی

۳) مین مکس

۴) تپه‌نوردی




۲۰۸ – کدام مطلب در خصوص جستجوی کورکورانه صحیح نیست؟

۱) کامل است. ۲) بهینه است.

۳) منابع زیادی مصرف می‌کند. ۴) بررسی آن فقط ارزش تئوریک دارد.

۲۰۹ – کدام یک از روش‌های جستجوی زیر بیشتر در بازی‌ها به‌کار می‌رود؟

۱) سطحی ۲) مین مکس ۳) تپه‌نوردی ۴) حریصانه

۲۱۰ – کدام جمله در خصوص جستجوها صحیح نیست؟

۱) در جستجوی حریصانه هزینه رسیدن به هدف با استفاده از تابع کشف کننده کاهش می‌یابد.

۲) جستجوی حریصانه می‌تواند زمان جستجو را کاهش دهد اما نه کامل است نه بهینه.

۳) در جستجو با هزینه یکسان هزینه مسیر حداقل بوده و هم بهینه هست هم کامل.

۴) جستجوی $A^*$ از ترکیب جستجوی حریصانه و تپه‌نوردی به‌وجود آمده است.

۲۱۱ – در مورد حوزه کاربردی‌های پردازش سیگنال کدام جمله صحیح نیست؟

۱) فهم، تشخیص و شناسایی کلام از کاربردهای مهم پردازش سیگنال است.

۲) تشخیص کلام عمل نگاشت از سیگنال‌های صوتی دیجیتالی شده به رشته‌ای از کلمات می‌باشد.

۳) همه زبان‌های بشری، ترکیبی از ۴۰ تا ۵۰ صوت متمایز می‌باشند که *phone* نامیده می‌شوند.

۴) برای تشخیص کلام ابتدا باید آن را فهمید سپس از یک بانک اطلاعات مفاهیم را استخراج کنیم.

۲۱۲ – کدام یک از موارد زیر جزو عوامل مؤثر در حل کارآمد مسئله در سیستم‌های خبره نیست؟

۱) دانش قابل اجرا، صحیح و قابل تفکیک و تمایز ۲) حذف سریع دیدگاه‌های غیرثمربخش

۳) منابع دانش غیرمشترک ۴) تقسیم راه‌حل‌ها به سطوح متفاوتی از تجرد

۲۱۳ – کدام یک از موارد زیر از ضروریات یک حل کننده مسئله ایده‌آل نیست؟

۱) دانش در مورد یک حوزه که شامل حقایق، کشف‌کنندگی و عقاید است.

۲) دانش برای یافتن راه حل‌های ناپایدار

۳) دانش برای برنامه‌ریزی استراتژی راه حل مسئله بعدی

۴) دانش برای ارزیابی راه حل‌های جزئی

۲۱۴ – کدام یک از موارد زیر جزو مراحل اکتساب دانش نیست؟

۱) شناسایی ذی‌نفعان ۲) شناسایی اهداف ۳) ادراک ۴) رسمی‌سازی

۲۱۵ – کدام عبارت جزوه مسائل جدی در بازیابی، اداره کردن و استفاده مجدد از اطلاعات ذخیره شده در وب نیست؟

۱) امنیت دسترسی به اطلاعات در وب همیشه برقرار نیست.

۲) اطلاعات به کار گرفته شده توسط شرکت‌ها ممکن است بعد از مدتی تغییر کند.

۳) دسترسی به منابع اطلاعات همیشه امکان‌پذیر نیست.

۴) خریدهای ارزان ارائه شده توسط شرکت‌ها ممکن است بعد از مدتی تغییر کرده باشد.

۲۱۶ – کدام جمله در خصوص عامل‌ها صحیح است؟

۱) عامل‌ها نرم افزارهایی هستند که در زمینه خاص مهارت دارند.

۲) عامل‌ها ابزار توسعه همه سیستم‌ها چه هوشمند و چه غیرهوشمند هستند.

۳) عامل‌ها ابزارهای ورودی و خروجی سیستم‌های هوشمند هستند.

۴) عامل‌ها دانش ارائه عملیات صحیح هستند.

۲۱۷ – کدام یک از امکانات زیر توسط پوسته‌ها یا شل‌ها پیاده‌سازی نمی‌شود؟

۱) زبان بازنمایی دانش ۲) یک ویراستار پایگاه دانش

۳) امکانات ردیابی و اشکال‌زدایی ۴) یک بانک اطلاعاتی دینامیک

۲۱۸ – کدام جمله صحیح است: هوش مصنوعی بر ..... تمرکز دارد.

۱) سیستم‌هایی که به طور منطقی فکر می‌کنند ۲) سیستم‌هایی که مانند انسان فکر می‌کنند

۳) سیستم‌هایی که به طور منطقی عمل می‌کنند ۴) سیستم‌هایی که مانند انسان عمل می‌کنند

۲۱۹ – در مثال زیر کدام جمله در خصوص ارتباط بین محیط، هدف، ادراک و عملیات و نوع عامل صحیح نیست؟

۱) محیط: بیمارستان / هدف: بیمار سالم / عملیات: سؤالات / ادراک: پاسخ بیمار / نوع عامل: سیستم تشخیص پزشکی

۲) محیط: پالایشگاه / هدف: افزایش خلوص / علمیات: تعدیل دما / ادراک: دما / نوع عامل: کنترل کننده دما

۳) محیط: مجموعه دانش‌آموز / هدف: افزایش نمرات / عملیات: تمرین‌های چاپ شده / ادراک: نتایج آزمون / نوع عامل: آموزگار الکترونیکی

۴) محیط: جاده / هدف: ایمنی / عملیات: ترمز / ادراک: سرعت سنج / نوع عامل: پلیس راهنمایی

۲۲۰ – کدام گزینه از انواع عامل‌ها قابل بررسی نمی‌باشد؟

۱) عامل واکنش ساده ۲) عامل‌های فکرگرا ۳) عامل‌های سودمند ۴) عامل‌های بازگشتی

۲۲۱ – کدام جمله در خصوص بازی شطرنج صحیح است؟

۱) محیط گسسته، اپیزودیک و قطعی است. ۲) محیط پیوسته، غیراپیزودیک و غیر قطعی است.

۳) محیط گسسته، غیراپیزودیک و قابل دسترسی است. ۴) محیط پیوسته، قابل دسترسی و قطعی است.

۲۲۲ – کدام مورد از معیارهای استراتژی جستجو نیست؟

۱) پیچیدگی زمانی ۲) بهینگی ۳) پیچیدگی الگوریتم ۴) کامل بودن

۲۲۳ – کدام جمله در ارزیابی جستجوها غلط است؟

۱) جستجوی سطحی کم عمق‌ترین حالت هدف را پیدا می‌کند.

۲) جستجوی عمیق مزایای جستجوی سطحی و عمقی را با هم ترکیب می‌کند.

۳) در حالت کلی که فضای جستجوی بزرگی وجود دارد و عمق راه حل نیز مجهول است جستجوی عمیق بهترین استراتژی است.

۴) جستجوی سطحی یک جستجوی کامل و بهینه است.

۲۲۴ – جستجوی $A^*$ .....

۱) کامل است – بهینه نیست. ۲) کامل نیست – بهینه است. ۳) کامل و بهینه است. ۴) نه کامل است و نه بهینه.

۲۲۵ – کدام جمله در خصوص الگوریتم‌ها صحیح است؟

۱) پیچیدگی فضای الگوریتم $A^*$ بسیار زیاد است.

۲) جستجوی حریصانه نه بهینه است و نه کامل.

۳) الگوریتم‌ها $SMA^*$، $IDA^*$ با روش حافظه محدود پیچیدگی فضای معمولی $A^*$ را کاهش می‌دهند.

۴) هر سه مورد

۲۲۶ – در بازی‌ها کدام مورد صحیح نیست؟

۱) الگوریتم $Minmax$ در یک بازی دو نفره بهترین حرکت را توسط تعیین درخت کلی بازی تشخیص می‌دهد.

۲) در روش آلفا بتا تمام درخت بازی در نظر گرفته می‌شود.

۳) الگوریتم آلفا بتا، محاسبه‌ای همانند $Minmax$ صورت می‌دهد اما کارایی بیشتری دارد.

۴) در بازی شانسی الگوریتم $Minmax$ با استفاده از میانگین، سودمندی تمام گره‌های فرزندی را ارزیابی می‌کند.

۲۲۷ – کدام مورد در خصوص سطوح تعریف عامل، مبتنی بر دانش نیست؟

۱) سطح عملیاتی ۲) سطح پیاده‌سازی ۳) سطح منطقی ۴) سطح دانش

۲۲۸ – کدام جمله صحیح نیست؟

۱) دانش در عامل‌ها به صورت جملات، توسط یک زبان بازنمایی دانش در یک پایگاه دانش ذخیره می‌شوند.

۲) عامل مبتنی بر دانش مرکب از یک پایگاه دانش و یک مکانیزم استنتاج است.

۳) استنتاج پردازشی از استخراج جملات جدید از جملات قدیمی است.

۴) تفسیر یک جمله همان واقعیتی است که جمله به آن رجوع می‌کند و قسمتی از دنیایی مجازی است و می‌تواند درست یا غلط است.

۲۲۹ – کدام گزاره در خصوص منطق صحیح نیست؟

۱) منطق گزاره‌ای می‌تواند استنتاج‌های قطعی را توسط یک عامل منطقی مورد نیاز جا دهد و همواره عملی است.

۲) منطق‌ها برای تعهداتی که به وجود نمی‌آورند، مفید هستند، زیرا کمبود تعهد، به پایگاه دانش آزادی عمل بیشتری می‌دهد.

۳) منطق‌های متفاوت تعهدات متفاوتی در مورد این که دنیا از چه ساخته شده است و یا چه نوع از اعتقادات با توجه به حقایق می‌توانیم داشته

شیم به‌وجود می‌آورند.
منطق گزاره‌ای فقط وجود حقایقی را متعهد می‌شود که ممکن است نکته مورد نظر در دنیای بازنمایی شده باشد یا نباشد.
۲۲ – در صورتی که عاملی با استفاده از منطق مرتبه اول استدلال کند کدام صحیح نیست؟
نیازمند شرح دقیقی از حالت جاری از ادراکات است.
نیازمند ذکر و استفاده از اطلاعات در مورد مطلوب بودن عملیات در شرایط مختلف است.
نیاز به عکس‌العمل در مقابل آنچه که دریافت می‌کند دارد.
نیازمند نگهداری مدل داخلی از نمادهای مربوط دنیا که مستقیما از ادراکات فراهم می‌شوند است.
۲۲ – در ارتباط بین سور وجودی و سور عمومی کدام گزینه غلط است؟

$\exists xp \equiv \sim \forall x \sim p$ (۴ $\forall xp \equiv \exists x \sim p$ (۳ $\sim \forall xp \equiv \exists x \sim p$ (۲ $\forall x \sim p \equiv \sim \exists xp$ (

۲۲ – تعریف زیر از هوش مصنوعی جزء کدام دسته از تعاریف هوش مصنوعی است؟
شاخه‌ای از علوم کامپیوتر که با اتوماسیون رفتار هوشمند مربوط می‌شود."
سیستم‌های که به طور منطقی فکر می‌کنند. ۲) سیستم‌های که مانند انسان فکر می‌کنند.
سیستم‌های که به طور منطقی عمل می‌کنند. ۴) سیستم‌های که مانند انسان عمل می‌کنند.
۲۲ – یک عامل برای آموزش زبان انگلیسی به روش ارتباط طراحی شده است. کلمات تایپ شده، جزء کدام یک از ویژگی‌های عامل است؟
محیط ۲) ادراکات ۳) عملیات ۴) اهداف
۲۲ – اگر بازی شطرنج بدون ساعت را در نظر بگیریم، ویژگی‌های این محیط کدام است؟
پیوسته – غیرقطعی – اپیزودیک – قابل دسترسی ۲) گسسته – قطعی – اپیزودیک – غیرقابل دسترس
گسسته – قطعی – غیراپیزودیک – غیرقابل دسترس ۴) گسسته – قطعی – غیراپیزودیک – قابل دسترسی
۲۲ – کدام یک از محیط‌های زیر اپیزودیک است؟
شطرنج به همراه ساعت ۲) رانندگی تاکسی ۳) سیستم تشخیص پزشکی ۴) سیستم تحیل تصویر
۲۲ – ..... نگاشتی از ادراک به عمل در حین بهنگام‌سازی وضعیت داخلی است.
برنامه عامل ۲) معماری عامل ۳) اهداف عامل ۴) عملیات عامل
۲۲ – در تعریف کدام یک از مسایل زیر هزینه مسیر صفر است؟
ریاضیات رمزی ۲) دنای مکش ۳) مساله کشیش‌ها و آدمخوارها ۴) معمای 8
۲۲ – کدام یک از استراتژی‌های جستجوی ناآگاهانه زیر فقط در بخشی از موارد خاصیت کامل بودن را دارا می‌باشد؟
جستجوی دو طرفه ۲) جستجوی عمیق کننده تکراری
جستجوی عمقی محدود ۴) جستجوی سطحی
۲۲ – فضای مورد نیاز برای اجرای استراتژی‌های جستجوی دو طرفه و جستجوی عمیق کننده تکراری به ترتیب کدام است؟
($b$ فاکتور انشعاب، $i$ محدودیت عمق، $d$ عمق پاسخ، $m$ ماکزیمم عمق درخت)

$bi, b^d$ $bi, b^{\frac{d}{2}}$ (۲ $bm, b^{\frac{d}{2}}$ (۳ $bd, b^{\frac{d}{2}}$ (۴

۲۲ – تابع زیر معادل کدام یک از استراتژی‌های جستجو آگاهانه است؟

$Function\ X\ (problem)\ returns\ a\ solution\ or\ failure$
$Return\ BEST - First - Search(problem, g + h)$

جستجوی حریصانه
الگوریتم $A^*$
الگوریتم جستجوی اول بهترین
جستجوی $SMA^*$
۲۲ – کدام یک از جملات زیر در مورد جستجوی $SMA^*$ صحیح نیست؟
می‌تواند از تمام حافظه دسترس استفاده ببرد.
از حالات تکراری تا جایی که حافظه اجازه می‌دهد جلوگیری می‌کند.
الگوریتم کامل است.
زمانی که حافظه موجود برای درخت جستجو کامل کافی باشد، جستجو $optimally\ efficient$ خواهد بود.
۲۲ – کدام یک از موارد زیر جزء مشکلات سیاست تپه‌نوردی نیست؟
$simulated\ annealing$ ۲) $local\ maxima$
$plateaux$ ۴) $Ridges$
۲۲ – اگر عمق حداکثر درخت $m$ و $b$ حرکت قانونی در هر نقطه داشته باشیم، پیچیدگی زمانی الگوریتم $Minmax$ کدام گزینه خواهد

بود؟

۱) $b^m$ ۲) $b^{m/2}$ ۳) $m^b$ ۴) $m^{b/2}$

۲۴۴ – برای بازنمایی دنیای شامل حقایق، اشیاء، ارتباطات و زمان از کدام زبان (منطق) زیر استفاده می‌شود؟

۱) منطق گزاره‌ای ۲) منطق مرتبه اول

۳) *Temporal Logic* ۴) تئوری احتمال (*Probability theory*)

۲۴۵ – ..... می‌تواند استنتاج‌های قطعی را توسط یک عامل منطقی مورد نیاز، جا بدهد، اما سریعا برای دنیاهای کوچک، غیرعملی می‌شود.

۱) منطق گزاره‌ای ۲) منطق مرتبه اول

۳) *Temporal Logic* ۴) تئوری احتمال (*Probability theory*)

۲۴۶ – ..... یک زبان بازنمایی رایج است که بر حسب یک تعهد هستی شناسی مبتنی بر وجود اشیاء و روابط در دنیا به وجود آمده است.

۱) منطق گزاره‌ای ۲) منطق مرتبه اول ۳) منطق فازی ۴) تئوری احتمال

۲۴۷ – کدام یک از موارد زیر یک استراتژی برای راهنمایی جستجو به سمت یک اثبات در منطق مرتبه اول نیست؟

۱) *Unit preference* ۲) مجموعه *Support* ۳) *Resolution* خروجی ۴) *Subsumption*

۲۴۸ – کدام یک از موارد زیر تعریف دقیق قضیه هربرند (*Herbrand*) است؟

۱) اگر مجموعه‌ای از گزاره‌ها صدق ناپذیر باشد، سپس زیر مجموعه محدودی از $Hs(S)$ وجود دارد که صدق ناپذیر است.

۲) اگر مجموعه‌ای از گزاره‌ها صدق‌پذیر باشد، سپس زیر مجموعه محدودی از $Hs(S)$ وجود دارد که صدق‌پذیر است.

۳) اگر مجموعه‌ای از گزاره‌ها صدق ناپذیر باشد، سپس زیر مجموعه محدودی از $Hs(S)$ وجود دارد که صدق‌پذیر است.

۴) اگر مجموعه‌ای از گزاره‌ها صدق‌پذیر باشد، سپس زیر مجموعه محدودی از $Hs(S)$ وجود دارد که صدق‌ناپذیر است.

۲۴۹ – کدام یک از محیط‌های زیر غیرقابل دسترس، غیرقطعی، غیراپیزودیک، پویا و پیوسته است؟

۱) آموزش دهنده زبان انگلیسی با ارتباط متقابل ۲) سیستم تحلیل تصویر

۳) شطرنج بدون ساعت ۴) ربات جابجا کننده اشیاء

۲۵۰ – اگر محیط قابل دسترس باشد و عامل دقیقا بداند هر کدام از عملیاتش دقیقا چه انجام می‌دهد و دنباله‌ای از عملیات‌ها باعث چه وضعیتی خواهد شد، نوع مسأله کدام مورد است؟

۱) تک حالته ۲) چند حالته ۳) احتمالی ۴) مکاشفه‌ای

۲۵۱ – کدام یک از الگوریتم‌های جستجوی زیر برای جستجو در حافظه محدوده شده طراحی شده است؟

۱) $A^*$ ۲) *Minmax* ۳) $IDA^*$ ۴) *BFS*

۲۵۲ – یک سیستم نماد فیزیکی دارای امکانات ..... برای عملیات عمومی باهوش است.

۱) لازم ۲) لازم و کافی ۳) کافی ۴) لازم ولی ناکافی

۲۵۳ – کدام یک از موارد زیر جزء خصایص ناخوشایند دانش نیست؟

۱) حجیم بودن ۲) سختی توصیف ۳) مرتبا تغییر می‌کند. ۴) یادگیری آن سخت است.

۲۵۴ – پایه و اساس تمام روش‌های هوش مصنوعی برای حل مسائل چیست؟

۱) جستجو ۲) فضای حالت ۳) حذف حالات غیرممکن ۴) گسترش حالات ممکن

۲۵۵ – کدام یک از موارد زیر جزء اجزاء یک سیستم تولید نیست؟

۱) پایگاه دانش ۲) مجموعه قواعد ۳) پایگاه دادگاه‌ها ۴) استراتژی کنترل

۲۵۶ – کدام عبارت زیر صحیح نیست؟

۱) جستجوی هیوریستیک بهترین پاسخ را ارائه می‌کنند.

۲) به طور متوسط هیوریستیک‌ها کیفیت مسیرهای بازرسی شده را بهبود می‌بخشد.

۳) استراتژی‌های کنترلی که باعث حرکت نشوند هیچ‌گاه ما را به جواب نمی‌رسانند.

۴) سیستم تولید راه خوبی است برای مدل کردن طبیعت شدیدا داده‌گرای عملی هوشیارانه

۲۵۷ – اگر از یک هیوریستیک همه منظوره مثل "الگوریتم نزدیک‌ترین همسایه" برای حل مسأله تاجر سیار استفاده کنیم، برای ملاقات $N$ شهر، زمان مورد نیاز با کدام یک از روابط زیر رابطه خطی دارد؟

۱) $N!$ ۲) $N^2$ ۳) $2^N$ ۴) $N^3$

۲۵۸ – برای پیاده‌سازی مسائل جبران‌پذیر یا برگشت‌پذیر چه ساختمان داده‌ای مورد نیاز است؟

۱) صف ساده ۲) صف اولویت ۳) پشته ۴) یک ساختار کنترل ساده

۲۵۹ – بازی شطرنج جزء کدام دسته از مسائل است؟

غیرقابل جبران یا برگشت‌ناپذیر ۲) قابل اغماض
برگشت‌پذیر ۴) قابل اغماض و برگشت‌پذیر

۲ – اثبات تئوری جزء کدامیک از سیستم‌های تولید است؟
غیریکنواخت – غیر پاره‌ای جابجاپذیر ۲) غیریکنواخت – پاره‌ای جابجاپذیر
یکنواخت – غیرپاره‌ای جابجاپذیر ۴) یکنواخت – پاره‌ای جابجاپذیر

۲ – کدام جمله زیر صحیح نیست؟
در مساله معمای هشت هم می‌توان از استدلال جلورو استفاده کرد هم عقب‌رو
فاکتور شاخه شاخه شدن در انتخاب استدلال جلورو و عقب‌رو مؤثر است.
استراتژی جستجوی دو سویه همواره موفق است.
در صورتی که وضعیت آغازی یکی باشد و وضعیت هدف متعدد از استدلال جلورو استفاده می‌کنیم.

۲ – به شکل تشخیص این که کدام یک از حقایق تغییر نمی‌کند ..... می‌گویند.
شاخص‌بندی ۲) مقایسه و مطابقت ۳) *Sterotype* ۴) مسأله قاب

۲ – کدام جمله زیر صحیح است؟
هر دو استراتژی ایجاد و تست و تپه‌نوردی روال‌های جستجوی اول عمق هستند.
استراتژی ایجاد و تست جستجوی اول – پهنا و تپه‌نوردی جستجوی اول – عمق است.
استراتژی ایجاد تست جستجوی اول – عمق و تپه‌نوردی جستجوی اول – پهنا است.
هر دو استراتژی ایجاد و تست و تپه‌نوردی روال‌های جستجوی اول – پهنا هستند.

۲ – کدام جمله زیر صحیح نیست؟
از الگوریتم $AO^*$ نمی‌توان برای حل مسائل $AND-OR$ استفاده کرد.
از الگوریتم $A^*$ نمی‌توان برای حل مسائل $AND-OR$ استفاده کرد.
$AO^*$ یک روش قابل تقلیل مسأله است.
قدرت الگوریتم $AO^*$ از الگوریتم $A^*$ بیشتر است.

۲ – مسأله ریاضیات رمزی را با کدام یک از استراتژی‌های زیر می‌توان حل کرد؟
تقلیل مسأله ۲) ایجاد و تست ۳) ارضا محدودیت ۴) استراتژی تحلیل ابزار پایان

۲ – کدام جمله زیر صحیح نیست؟
ثر افق یکی از مشکلات روش $MinMax$ است.
حرکت‌های کتابی از مشکلات روش $MinMax$ است.
روش $MinMax$ رقیب را خیلی باهوش فرض می‌کند.
نتظار برای آرامش یک نسخه اصلاح شده الگوریتم روش $MinMax$ است.

۲ – کدام یک از جملات زیر صحیح نیست؟
مکانیز اثبات تئوری دارای این مشکل است که برخی از دانش‌ها را نمی‌توان به کمک منطق مسند نمایش داد.
ئوری هربرند پایه اساسی تئوریک روال اثبات در چه مسندها است.
منطق مسند دارای روال تصمیم‌گیری است.
در منطق مسندها، حقایق دنیای واقعی به صورت احکام با فرم خوش شکل نوشته می‌شوند.

۲ – کدام یک از روش‌های زیر اجازه می‌دهد مدل‌های تودرتوی مجموعه عقاید ارائه گردند؟
منطق غیریکنواخت ۲) استدلال احتمالی ۳) منطق مشکوک ۴) فضای عقاید

۲ – در مورد لیست $OUT$ در سیستم استدلال غیریکنواخت $TMS$ کدام جمله صحیح است؟
عتقاد داریم درست است. ۲) اعتقاد نداریم درست است.
یست توجیهات مربوط به هر گره است. ۴) $OUT$ و $CP$ یکی است.

۲ – توصیف مجموعه از خصایص و شخصیت‌هایی که اغلب یک جا در مردم دیده می‌شود به وسیله کدام یک از روش‌های هوش نوعی زیر قابل نمایش است؟
قاب‌ها ۲) سناریوها ۳) کلیشه‌ای‌ها ۴) مدل مجموعه قواعد

# پاسخ سؤالات تستی امتحانی

| سؤال | جواب |
|---|---|
| ۲۲۶ | ۲ |
| ۲۲۷ | ۲ |
| ۲۲۸ | ۴ |
| ۲۲۹ | ۱ |
| ۲۳۰ | ۴ |
| ۲۳۱ | ۳ |
| ۲۳۲ | ۳ |
| ۲۳۳ | ۲ |
| ۲۳۴ | ۴ |
| ۲۳۵ | ۴ |
| ۲۳۶ | ۱ |
| ۲۳۷ | ۱ |
| ۲۳۸ | ۳ |
| ۲۳۹ | ۴ |
| ۲۴۰ | ۲ |
| ۲۴۱ | ۳ |
| ۲۴۲ | ۱ |
| ۲۴۳ | ۱ |
| ۲۴۴ | ۳ |
| ۲۴۵ | ۱ |
| ۲۴۶ | ۲ |
| ۲۴۷ | ۳ |
| ۲۴۸ | ۱ |
| ۲۴۹ | ۴ |
| ۲۵۰ | ۱ |
| ۲۵۱ | ۳ |
| ۲۵۲ | ۲ |
| ۲۵۳ | ۴ |
| ۲۵۴ | ۲ |
| ۲۵۵ | ۱ |
| ۲۵۶ | ۱ |
| ۲۵۷ | ۲ |
| ۲۵۸ | ۳ |
| ۲۵۹ | ۱ |
| ۲۶۰ | ۴ |
| ۲۶۱ | ۳ |
| ۲۶۲ | ۴ |
| ۲۶۳ | ۱ |
| ۲۶۴ | ۱ |
| ۲۶۵ | ۳ |

| سؤال | جواب |
|---|---|
| ۱۸۱ | ۲ |
| ۱۸۲ | ۴ |
| ۱۸۳ | ۳ |
| ۱۸۴ | ۱ |
| ۱۸۵ | ۲ |
| ۱۸۶ | ۲ |
| ۱۸۷ | ۳ |
| ۱۸۸ | ۳ |
| ۱۸۹ | ۲ |
| ۱۹۰ | ۴ |
| ۱۹۱ | ۴ |
| ۱۹۲ | ۱ |
| ۱۹۳ | ۴ |
| ۱۹۴ | ۳ |
| ۱۹۵ | ۱ |
| ۱۹۶ | ۴ |
| ۱۹۷ | ۱ |
| ۱۹۸ | ۳ |
| ۱۹۹ | ۴ |
| ۲۰۰ | ۳ |
| ۲۰۱ | ۱ |
| ۲۰۲ | ۳ |
| ۲۰۳ | ۴ |
| ۲۰۴ | ۴ |
| ۲۰۵ | ۲ |
| ۲۰۶ | ۲ |
| ۲۰۷ | ۱ |
| ۲۰۸ | ۲ |
| ۲۰۹ | ۲ |
| ۲۱۰ | ۴ |
| ۲۱۱ | ۴ |
| ۲۱۲ | ۳ |
| ۲۱۳ | ۲ |
| ۲۱۴ | ۱ |
| ۲۱۵ | ۱ |
| ۲۱۶ | ۱ |
| ۲۱۷ | ۴ |
| ۲۱۸ | ۳ |
| ۲۱۹ | ۴ |
| ۲۲۰ | ۴ |

| سؤال | جواب |
|---|---|
| ۱۳۶ | ۲ |
| ۱۳۷ | ۴ |
| ۱۳۸ | ۲ |
| ۱۳۹ | ۱ |
| ۱۴۰ | ۳ |
| ۱۴۱ | ۱ |
| ۱۴۲ | ۳ |
| ۱۴۳ | ۴ |
| ۱۴۴ | ۲ |
| ۱۴۵ | ۴ |
| ۱۴۶ | ۱ |
| ۱۴۷ | ۳ |
| ۱۴۸ | ۲ |
| ۱۴۹ | ۱ |
| ۱۵۰ | ۳ |
| ۱۵۱ | ۱ |
| ۱۵۲ | ۲ |
| ۱۵۳ | ۴ |
| ۱۵۴ | ۱ |
| ۱۵۵ | ۱ |
| ۱۵۶ | ۱ |
| ۱۵۷ | ۲ |
| ۱۵۸ | ۴ |
| ۱۵۹ | ۲ |
| ۱۶۰ | ۲ |
| ۱۶۱ | ۳ |
| ۱۶۲ | ۲ |
| ۱۶۳ | ۴ |
| ۱۶۴ | ۳ |
| ۱۶۵ | ۳ |
| ۱۶۶ | ۴ |
| ۱۶۷ | ۱ |
| ۱۶۸ | ۱ |
| ۱۶۹ | ۳ |
| ۱۷۰ | ۲ |
| ۱۷۱ | ۲ |
| ۱۷۲ | ۴ |
| ۱۷۳ | ۱ |
| ۱۷۴ | ۴ |
| ۱۷۵ | ۴ |

| سؤال | جواب |
|---|---|
| ۹۱ | ۴ |
| ۹۲ | ۱ |
| ۹۳ | ۳ |
| ۹۴ | ۳ |
| ۹۵ | ۱ |
| ۹۶ | ۲ |
| ۹۷ | ۱ |
| ۹۸ | ۱ |
| ۹۹ | ۳ |
| ۱۰۰ | ۴ |
| ۱۰۱ | ۳ |
| ۱۰۲ | ۴ |
| ۱۰۳ | ۴ |
| ۱۰۴ | ۱ |
| ۱۰۵ | ۱ |
| ۱۰۶ | ۲ |
| ۱۰۷ | ۴ |
| ۱۰۸ | ۲ |
| ۱۰۹ | ۴ |
| ۱۱۰ | ۱ |
| ۱۱۱ | ۳ |
| ۱۱۲ | ۱ |
| ۱۱۳ | ۴ |
| ۱۱۴ | ۴ |
| ۱۱۵ | ۴ |
| ۱۱۶ | ۱ |
| ۱۱۷ | ۳ |
| ۱۱۸ | ۲ |
| ۱۱۹ | ۲ |
| ۱۲۰ | ۱ |
| ۱۲۱ | ۲ |
| ۱۲۲ | ۱ |
| ۱۲۳ | ۳ |
| ۱۲۴ | ۳ |
| ۱۲۵ | ۴ |
| ۱۲۶ | ۲ |
| ۱۲۷ | ۳ |
| ۱۲۸ | ۱ |
| ۱۲۹ | ۳ |
| ۱۳۰ | ۲ |

| سؤال | جواب |
|---|---|
| ۴۶ | ۴ |
| ۴۷ | ۱ |
| ۴۸ | ۲ |
| ۴۹ | ۱ |
| ۵۰ | ۴ |
| ۵۱ | ۳ |
| ۵۲ | ۳ |
| ۵۳ | ۳ |
| ۵۴ | ۴ |
| ۵۵ | ۱ |
| ۵۶ | ۱ |
| ۵۷ | ۲ |
| ۵۸ | ۲ |
| ۵۹ | ۲ |
| ۶۰ | ۱ |
| ۶۱ | ۳ |
| ۶۲ | ۴ |
| ۶۳ | ۴ |
| ۶۴ | ۴ |
| ۶۵ | ۳ |
| ۶۶ | ۲ |
| ۶۷ | ۳ |
| ۶۸ | ۱ |
| ۶۹ | ۲ |
| ۷۰ | ۲ |
| ۷۱ | ۳ |
| ۷۲ | ۱ |
| ۷۳ | ۲ |
| ۷۴ | ۲ |
| ۷۵ | ۱ |
| ۷۶ | ۴ |
| ۷۷ | ۲ |
| ۷۸ | ۲ |
| ۷۹ | ۱ |
| ۸۰ | ۲ |
| ۸۱ | ۱ |
| ۸۲ | ۲ |
| ۸۳ | ۴ |
| ۸۴ | ۲ |
| ۸۵ | ۲ |

| سؤال | جواب |
|---|---|
| ۱ | ۳ |
| ۲ | ۱ |
| ۳ | ۱ |
| ۴ | ۱ |
| ۵ | ۳ |
| ۶ | ۱ |
| ۷ | ۴ |
| ۸ | ۱ |
| ۹ | ۴ |
| ۱۰ | ۳ |
| ۱۱ | ۲ |
| ۱۲ | ۳ |
| ۱۳ | ۴ |
| ۱۴ | ۲ |
| ۱۵ | ۴ |
| ۱۶ | ۴ |
| ۱۷ | ۴ |
| ۱۸ | ۲ |
| ۱۹ | ۳ |
| ۲۰ | ۱ |
| ۲۱ | ۲ |
| ۲۲ | ۳ |
| ۲۳ | ۲ |
| ۲۴ | ۱ |
| ۲۵ | ۱ |
| ۲۶ | ۱ |
| ۲۷ | ۴ |
| ۲۸ | ۳ |
| ۲۹ | ۲ |
| ۳۰ | ۲ |
| ۳۱ | ۳ |
| ۳۲ | ۳ |
| ۳۳ | ۱ |
| ۳۴ | ۳ |
| ۳۵ | ۲ |
| ۳۶ | ۴ |
| ۳۷ | ۳ |
| ۳۸ | ۲ |
| ۳۹ | ۴ |
| ۴۰ | ۳ |

| | |
|---|---|
| ۲۶۶ | ۲ |
| ۲۶۷ | ۳ |
| ۲۶۸ | ۴ |
| ۲۶۹ | ۲ |
| ۲۷۰ | ۳ |

| | |
|---|---|
| ۲۲۱ | ۳ |
| ۲۲۲ | ۴ |
| ۲۲۳ | ۳ |
| ۲۲۴ | ۳ |
| ۲۲۵ | ۴ |

| | |
|---|---|
| ۱۷۶ | ۲ |
| ۱۷۷ | ۳ |
| ۱۷۸ | ۳ |
| ۱۷۹ | ۲ |
| ۱۸۰ | ۱ |

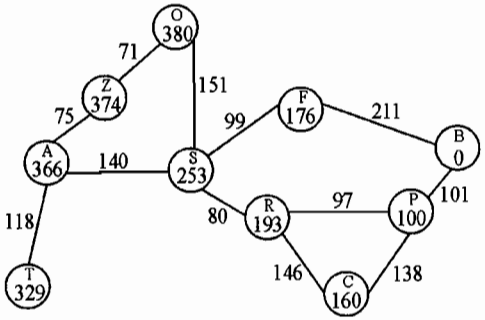
| | |
|---|---|
| ۱۳۱ | ۴ |
| ۱۳۲ | ۳ |
| ۱۳۳ | ۱ |
| ۱۳۴ | ۴ |
| ۱۳۵ | ۳ |

| | |
|---|---|
| ۸۶ | ۴ |
| ۸۷ | ۲ |
| ۸۸ | ۳ |
| ۸۹ | ۴ |
| ۹۰ | ۴ |

| | |
|---|---|
| ۴۱ | ۲ |
| ۴۲ | ۳ |
| ۴۳ | ۴ |
| ۴۴ | ۲ |
| ۴۵ | ۱ |

# ضمیمه ۲ (سؤالات تشریحی امتحانی)

۱ – توسط الگوریتم $A^*$ با شروع از راس $A$، درخت جستجو را مرحله به مرحله توسعه دهید تا هدف $B$ به‌دست آید. (مقدار داخل هر گره برابر هزینه فاصله مستقیم تا هدف $B$ می‌باشد.

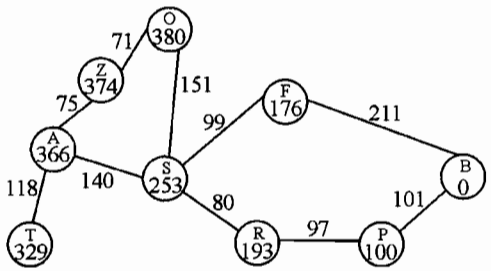



۲ – الف) روش $LRTA^*$ را در قالب مثالی شرح دهید. ب) به نظر شما این روش در کدام نوع از جستجوها قابل طبقه‌بندی است و آیا عاملی که از این روش استفاده می‌کند قابلیت یادگیری دارد؟ توضیح دهید.

۳ – جمله مقابل را در نظر بگیرید: " اسب‌ها حیوان هستند" در نتیجه " سر یک اسب، سر یک حیوان است"

الف) مقدم و تالی جمله فوق را به زبان منطق مرتبه اول بنویسید. از سه مسند $HeadOf(h,x)$ (به معنای $h$ سر $x$ است)، $Horse(x)$ و $Animal(x)$ استفاده نمایید. ب) جمله را به شکل نرمال عطفی بنویسید. ج) توسط تحلیل نشان دهید که تالی از مقدم نتیجه می‌شود.

۴ – توسط $RBFS$ با شروع از راس $A$، درخت جستجو را مرحله به مرحله توسعه دهید تا هدف $B$ به‌دست آید. (مقدار $best$ و $alternative$ و $f.limit$ را در هر مرحله مشخص نموده و دلیل تغییر هر یک را در صورت نیاز به تشریح، مختصرا بیان کنید. ضمنا مقدار داخل هر گره برابر هزینه فاصله مستقیم تا هدف $B$ می‌باشد.)




۵ – نحوه هرس شدن درخت در هرس آلفا و بتا را با مثالی تشریح نمایید. چگونه می‌توان از هرس آلفا و بتا در بازی‌های با گره شانس بهره برد؟

۶ – در الگوریتم $DPLL$ روش برخورد با نماد محض ($Pure\ symbol$) و بند واحد ($Unit\ clause$) را با مثالی تشریح نمایید و مشخص کنید که چرا این رویه باعث افزایش سرعت الگوریتم شده است؟

۷ – حذف سور عمومی، اسکولم‌سازی در سور وجودی، یکسان‌سازی و عمومی‌ترین یکسان‌ساز و جداسازی استاندارد ($standardizing\ apart$) و وارسی وقوع ($occur\ check$) و شبکه شمول ($subsumption\ lattice$) را هر یک با مثالی مختصرا توضیح دهید.

۸ – عامل مبتنی بر سودمندی را با رسم شکل توضیح دهید.

۹ – برای مساله پازل 8 تایی: الف) سه مساله تعدیل شده (با محدودیت‌های کمتر) تعریف کنید. ب) با توجه به قسمت (الف)، سه هیوریستیک قابل‌قبول برای هر مسأله، استخراج کنید.

۱۰ – الگوریتم تابکاری شبیه‌سازی شده حرارت ($Simulated\ annealing$) را شرح دهید.

۱۱ – شکل نرمال عطفی ($CNF$) عبارت زیر را به دست آورید.

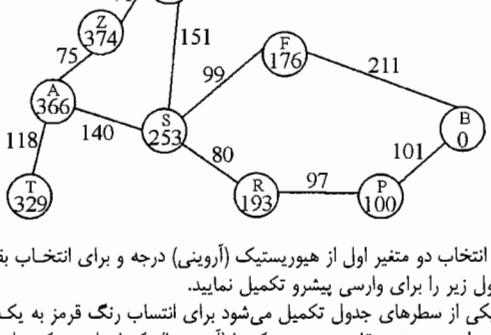
$$P \Leftrightarrow (P \vee Q)$$

۱۲ – مساله زیر را در نظر بگیرید: "برای یک آمریکایی، فروش تسلیحات به ملل متخاصم جرم است. کشور نونو که دشمن آمریکاست، تعدادی موشک دارد و تمامی موشک‌هایش را از سرهنگ وست که یک آمریکایی است، خریده است."

الف) مساله فوق را به صورت بندهای معین مرتبه اول بازنمایی کنید. ب) درخت اثبات به وسیله الگوریتم زنجیره‌ای پس‌رو، برای اثبات این

ه "وست یک مجرم است" را رسم کنید.

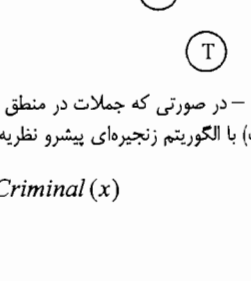
۱ – الگوریتم جستجو توسط $A^*$ با شروع از راس $Z$، درخت جستجو را مرحله به مرحله توسعه دهید تا هدف $B$ به‌دست آید هزینه مسیر قدر است و از چه شهرهایی عبور می‌کند؟ (مقدار داخل هر گره برابر هزینه فاصله مستقیم تا هدف $B$ می‌باشد.




۱ – برای مسئله رنگ‌آمیزی گراف (با سه رنگ، در انتخاب دو متغیر اول از هیوریستیک (آروینی) درجه و برای انتخاب بقیه متغیرها از یوریستیک (آروینی) $MRV$ استفاده نمایید و جدول زیر را برای وارسی پیشرو تکمیل نمایید.

**نکته ۱:** در هر مرحله با انتساب مقدار به یک متغیر، یکی از سطرهای جدول تکمیل می‌شود برای انتساب رنگ قرمز به یک متغیر در سلول ربوط به آن $(R)$ را بنویسید. **نکته ۲:** در صورتیکه برای دو متغیر مقادیر هیوریستیک‌ها (آروین‌ها) یکسان است یکی را به صورت اتفاقی خاب نمایید.

| | *WA* | *NT* | *Q* | *NSW* | *V* | *SA* | *T* |
|---|---|---|---|---|---|---|---|
| *Initial domains* | *RGB* | *RGB* | *RGB* | *RGB* | *RGB* | *RGB* | *RGB* |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |




۱ – در صورتی که جملات در منطق مرتبه اول به صورت زیر باشند. آیا می‌توان $Criminal(West)$ را نتیجه گرفت؟
ب) با الگوریتم زنجیره‌ای پیشرو نظریه خود را اثبات کنید. ب) با تحلیل $(Resolution)$ اثبات نمایید.

1) $\forall x, y, z\ American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$

2) $\exists x\ Owns(Nono, x) \wedge Missle(x)$

3) $\forall x\ Missle(x) \wedge Own(Nono, x) \Rightarrow Sells(West, x, Nono)$

4) $\forall x\ Missle(x) \Rightarrow Weapon(x)$

5) $\forall x\ Enemy(x, America) \Rightarrow Hostile(x)$

6) $American(West)$

7) $Enemy(Nono, America)$

۲ – جدول زیر را کامل کنید.

| نوع عامل | حسگرها | اقدام گرها | مقیاس کارایی | محیط |
|---|---|---|---|---|
| سیستم تشخیص پزشکی | | | | |
| ربات جابه‌جا کننده اشیاء | | | | |

۲ – الف) روش تولید سه مسئله تعدیل شده برای مسئله معمای هشت را بیان کنید. ب) مقدار هیوریستیک‌های به‌دست آمده از این مسائل دیل شده را از وضعیت مشخص شده در شکل (الف) برای رسیدن به هدف در شکل (ب) را مشخص کنید. ج) اگر تعدادی هیوریستیک قابل ول داشته باشیم، بهترین هیوریستیک ممکن را چگونه می‌توان از آنها ایجاد کرد؟ مشکل این هیوریستیک چیست؟

| | | |
|---|---|---|
| 4 | 8 | |
| 7 | 3 | 2 |
| 6 | 1 | 5 |

شکل (الف) وضعیت شروع

| | | |
|---|---|---|
| | 1 | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

شکل (ب) وضعیت هدف

۱۸ – دنیای *wumpus* زیر را در نظر بگیرید. (چاله: $PIT$، طلا: $GLD$، ومپوز: $WU$، عامل: $AG$). برای نشان دادن این هدف که در خانه [1,2] چاله‌ای وجود ندارد (یعنی $\alpha = \neg P_{1,2}$): الف) ادراک عامل به ازای حضور عامل در خانه‌های (1,1)، (1,2)، (2, 1)، (2,2) و (2,3) را به صورت نمادهای گزاره‌ای بیان کنید. ب) جملاتی از پایگاه دانش ($KB$) را بنویسید که برای رسیدن به هدف ضروری هستند. ج) توسط الگوریتم "$PL.Resolution$"، هدف را به‌دست آورید. (با ذکر قوانین استفاده شده).

| | | | |
|---|---|---|---|
| | 2,4 | | $PIT$ |
| $PIT$ | | $PIT$ | $GLD$ |
| 1,2 | | | |
| $AG$ | | $WU$ | |

۱۹ – عبارت زیر را به شکل نرمال عطفی ($CNF$) در منطق مرتبه اول تبدیل کنید.

$$\forall x\,[\forall y\,Animal(y) \Rightarrow Loves(x,y)] \Rightarrow [\exists y\,Loves(y,x)]$$

که به معنای زیر است:

*Everyone who loves all animals is loved by someone*

۲۰ – مسئله معمای 8 را در نظر بگیرید.

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 8 | | 4 |
| 7 | 6 | 5 |

حالت شروع

| | | |
|---|---|---|
| 2 | 8 | 3 |
| 1 | 6 | 4 |
| 7 | | 5 |

حال هدف

الف) دو تابع هیوریستیک قابل قبول برای این معما، طراحی کنید. (چگونگی طراحی را بیان کنید.) ب) الگوریتم جستجوی $A^*$ را روی هر یک از این هیوریستیک‌ها اعمال کنید. (از حالت شروع، درخت جستجو را تا رسیدن به هدف مرحله به مرحله رسم نمایید.) ج) نتایج این دو جستجو را مقایسه کنید. ( با ذکر دلیل)

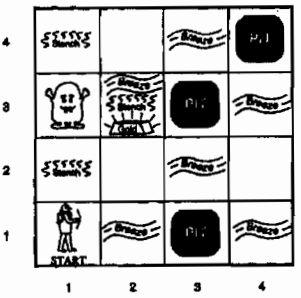
۲۱ – فرض کنید می‌خواهیم مسئله هشت وزیر را با استفاده از الگوریتم ژنتیک حل کنیم:

الف) مسئله را برای حل با الگوریتم ژنتیک، چگونه کد می‌کنید؟ ب) افرادی (کروموزوم‌هایی) که در هر جمعیت، در این مسئله وجود دارند، به چه شکلی هستند؟ ج) به صورت مثال، چگونگی اعمال عملکرد ژنتیکی *crossover* را در این مسئله که روی دو فرد اعمال شده، نشان دهید.

۲۲ – الف) در دنیای ومپوز زیر، جمله زیر را با استفاده از منطق گزاره‌ها به ازای خانه $x$ و $y$ بنویسید:

"در یک خانه نسیم می‌وزد اگر و فقط اگر چاله‌ای مجاور آن باشد."

جمله فوق را برای خانه‌های (1 و 1) و (1 و 2) بازنویسی کنید.




ب) جمله مقابل را در مسئله دنیای ومپوز، به شکل نرمال عطفی ($CNF$) تبدیل کنید $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

۲۳ – با استفاده از قاعده استنتاج تحلیل: ($Resolution$) نتیجه بگیرید. هدف: $Kills(Curiosity, Tuna)$

جملات $KB$:

$\forall x[\forall y\ Animal(y) \Rightarrow Loves(x,y)] \Rightarrow [\exists y\ Loves(x,y)]$

$\forall x[\exists y\ Animal(y) \wedge Kills(x,y)] \Rightarrow \forall z \neg Loves(z,x)]$

$\forall x\ Animal(x) \Rightarrow Loves(Jack,x)$

$Kills(Jack,Tuna) \vee Kills(Curiosity,Tuna)$

$Cat(Tuna)$

$\forall x\ Cat(x) \Rightarrow Animal(x)$

۲۴ – الگوریتم "پس‌گرد" ساده را برای مسائل ارضاء محدودیت شرح دهید.
۲۵ – الگوریتمی برای روش $Minmax$ ارائه دهید.
۲۶ – الگوریتمی برای جستجوی تپه‌نوردی ارائه دهید.
۲۷ – جستجوی اول سطح را تشریح کنید.
۲۸ – جستجوی هزینه یکنواخت را تعریف کنید.
۲۹ – عامل‌های واکنشی مبتنی بر مدل را تعریف کنید.
۳۰ – چهار مورد از راهبردهای جستجوی ناآگاهانه را نام ببرید.
۳۱ – شبکه معنایی را تعریف و چهار اصل آن را نام ببرید.
۳۲ – روش $Minmax$ را تشریح کنید.
۳۳ – رویه جستجوی تپه‌نوردی را با یک شکل توضیح دهید.
۳۴ – الف) تست تورینگ را بیان کرده ب) دو ویژگی آن را نام ببرید.
۳۵ – روش مینی‌مکس را تشریح نمایید.
۳۶ – رویه جستجوی تپه‌نوردی را با یک شکل توضیح دهید.

# پاسخ سؤالات تشریحی امتحانی

۱ – به شکل ۳.۴ در کتاب اصلی (ویرایش دوم) یا شکل ۲۴.۳ در کتاب اصلی (ویرایش سوم) مراجعه کنید.

۲ – الگوریتم LRTA* برگرفته از learning real time A* و به معنای «بی‌درنگ یادگیرنده» می‌باشد. این الگوریتم در دسته روشهای جستجوی برخط، آن هم از نوع محلی قرارمی‌گیرد. روال کار این الگوریتم بدان صورت است که در هر بار مانند A* از مجموع h(n)+g(n) گره‌های همسایه برای تصمیم‌گیری استفاده می‌کند. برای این منظور فرض کنید هم‌اکنون در گره x قرار داریم و هر همسایه را با y نشان دهیم. ابتدا h(x) را با h(y) مقایسه می‌کنیم، اگر h(x)≠h(y) مانند A* پیش می‌رویم. اگر h(x)=h(y)، یعنی از هر دوی این گره‌ها تا هدف فاصله یکسانی داریم و در مینیمم‌محلی گیر کرده‌ایم، بنابراین مقدار h گره جاری یعنی h(x) را با کمترین f(y)=h(y)+g(y) از همسایگان بروز کرده و سپس به آن همسایه منتقل می‌شویم. دوباره برای آن موقعیت جدید همین کار را کرده و در زمان خروج از هر گره مجددا h آنرا بروز می‌کنیم. این کار باعث می‌شود تا با گیرکردن در یک مینیمم محلی، در طی چند مرحله بروزرسانی، از مینیمم محلی فرار کرده و شانس رسیدن به مینیمم سراسری را داشته باشیم. برای نمونه به شکل ۲۲.۴ در کتاب اصلی (ویرایش دوم) یا شکل ۲۳.۴ در کتاب اصلی (ویرایش سوم) مراجعه کنید. در ابتدا در گره h(x)=2 قرار داریم که یکی از همسایگانش نیز h(y)=2 است، پس مقدار h(x)=h(y)+g(y)=2+1=3 شده و طبق A* به گره کمتر منتقل می‌شویم. در مرحله بعد، مجددا گره فعلی h(x)=2 دارد و در همسایگانش مجموع h و g مقادیر ۴ و ۵ می‌باشد. پس مقدار گره جاری h(x)=4 شده و به سمت گره کمتر حرکت می‌کنیم. پس از ادامه همین روال مشاهده می‌کنید که از مینیمم محلی فرار کرده و از سمت راست شکل خارج می‌شویم. این الگوریتم هر لحظه در حال یادگیری است.

۳ – به پاسخ تمرین ۱۸.۹ (ویرایش دوم) یا تمرین ۲۳.۹ (ویرایش سوم) مراجعه کنید.

۴ – به شکل ۶.۴ کتاب اصلی (ویرایش دوم) یا شکل ۲۷.۳ کتاب اصلی (ویرایش سوم) مراجعه کنید.

۵ – می‌توان الگوریتم زیر را برای هرس آلفا- بتا بیان نمود:

- سطر ها را یکی در میان Max و Min بنامید. (معمولا سطر اول Max است درغیر اینصورت در صورت سؤال ذکر خواهد شد.)
- در کنار تمام گره های واقع در سطر Max عبارت α=-∞ و در کنار تمام گره های واقع در سطر Min عبارت β=+∞ را بنویسید.
- طبق پیمایش پیشوندی به سراغ گره بعدی بروید:
  - * اگر آن گره غیرپایانی است: از آن عبور کنید.
  - + اگر آن گره پایانی است و مقدار k دارد:
  - * اگر پدرش یک گره min با مقدار β است: مقدار قبلی پدر (β) را با مقدار k مقایسه کرده و کمترین آنها را به جای مقدار قبلی β بنویسید. اگر مقدار جدید بتا≤ مقدار آلفای پدرش باشد، آنگاه سایر فرزندان بتا هرس می‌شوند.
  - * اگر پدرش یک گره max با مقدار α است: مقدار قبلی پدر (α) را با مقدار k مقایسه کرده و بیشترین آنها را به جای مقدار قبلی α بنویسید. اگر مقدار جدید آلفا≤ مقدار بتای پدرش باشد، آنگاه سایر فرزندان آلفا هرس می‌شوند.
- پس از بررسی تمام فرزندان یک گره، مقدار آن گره را به پدر مخابره کنید.
- در صورتی که طبق پیمایش پیشوندی گره‌ای باقی‌مانده است، به سراغ گره بعدی بروید.

بعنوان نمونه برای درخت شکل ۲.۶ کتاب اصلی (ویرایش دوم) یا شکل ۲.۵ کتاب اصلی (ویرایش سوم)، ترتیب مراحل و هرس را مشاهده می‌کنید.

| | شروع | مشاهده ۳ | مشاهده ۱۲ | مشاهده ۸ | مشاهده۲ | مشاهده ۱۴ | مشاهده ۵ | مشاهده۲ |
|---|---|---|---|---|---|---|---|---|
| گره A | α=-∞ | α=-∞ | α=-∞ | α=3 | α=3 | α=3 | α=3 | α=3 |
| گره B | β=+∞ | β=3 | β=3 | β=3 | β=3 | β=3 | β=3 | β=3 |
| گره C | β=+∞ | β=+∞ | β=+∞ | β=+∞ | β=2 (α>β پس هرس) | β=2 | β=2 | β=2 |
| گره D | β=+∞ | β=+∞ | β=+∞ | β=+∞ | β=+∞ | β=14 | β=5 | β=2 (α>β پس هرس) |

می‌توانیم از این روش برای هرس گره‌های شانس نیز استفاده کنیم. می‌دانیم که مقدار یک گره شانس برابر است با میانگین مقادیر زیر شاخه‌هایش. پس اگر فرض کنیم مقادیر تابع سودمندی کراندار باشند، می‌توانیم برای گره‌های شانس نیز کرانی بیابیم. با داشتن کران برای گره شانس، در واقع همان مقداری را یافتیم که در هرس آلفا- بتا برای هرس سایر زیرشاخه‌ها نیاز می‌باشد.

۶ – منظور از نماد محض، نمادی است که همواره در تمام بندهای پایگاه، به یک شکل ظاهر شده باشد (یا همیشه با ¬ و یا همیشه بدون ¬). و منظور از بند واحد، بندی است که تنها دارای یک لفظ باشد. مثلا بقیه الفاظ آن مقدار گرفته‌اند و در آن بند فقط یک لفظ بدون مقدار باقی‌مانده است. الگوریتم DPLL که برای تشخیص درستی یا نادرستی یک جمله در پایگاه بکار می‌رود و به گونه‌ای به نمادهای محض مقدار می‌دهد که الفاظش true شوند. همچنین برای بندهای واحد مقداری درنظر می‌گیرد که کل آن بند true شود. مثلا بند (¬A∨¬B) که بدانیم B=True، بندی واحد است که DPLL نتیجه می‌دهد: A=False. با اعمال این مقدار در کل پایگاه ممکن است مجددا چندین بند واحد و نماد محض تشکیل شود که به آنها مقدار می‌دهیم تا از وسعت مدلهای ممکن در پایگاه بکاهیم. اعمال این شیوه و خاتمه زودهنگام آن در کنار برخی راهکارها نظیر انتخاب متغیر و شروع مجدد تصادفی، این الگوریتم را از سریعترین الگوریتمها کرده است.

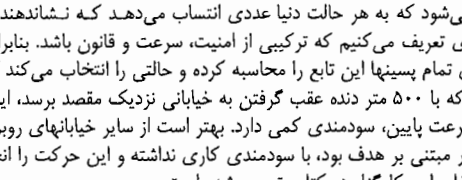
۷ – پایگاه‌های شامل سور عمومی و وجودی در منطق مرتبه‌اول قرارمی‌گیرند که برای تبدیل چنین پایگاهی به حالت گزاره‌ای باید سورها حذف شوند تا بتوان از قواعد استنتاج گزاره‌ای استفاده نمود. طبق مثالی از کتاب اصلی، پایگاه زیر در منطق مرتبه‌اول شامل دو جمله است:

a1: $\forall x\ King(x)\wedge Greedy(x)\Rightarrow Evil(x)$
a2: $\exists x\ Crown(x)\wedge Onhead(x,John)$

پس از حذف سورعمومی و وجودی یک پایگاه گزاره‌ای داریم:

b1: $King(John)\wedge Greedy(John)\Rightarrow Evil(John)$
b2: $King(Richard)\wedge Greedy(Richard)\Rightarrow Evil(Richard)$
b3: $King(Father(John))\wedge Greedy(Father(John))\Rightarrow Evil(Father(John))$
b4: $King(Father(Richard))\wedge Greedy(Father(Richard))\Rightarrow Evil(Father(Richard))$
b5: $Crown(C)\wedge Onhead(C,John)$

- اگر در پایگاهی بدانیم که «ماری، علی را می‌شناسد» یعنی Knows(mary,ali). حال اگر بپرسیم «چه کسی علی را می‌شناسد؟» یعنی Knows(x,ali)، طبق یکسان‌سازی باید جایگزینی {x/mary} را داشته باشیم. یعنی ماری.
- اگر در دو جمله متغیرها همنام باشند عملیات یکسان‌سازی شکست می‌خورد. بنابراین روش جداسازی استاندارد پیشنهاد می‌دهد که در آغاز کار نام تمام متغیرها را متفاوت قرار دهیم. اگر در پایگاهی بدانیم که «احمد، همه را می‌شناسد» یعنی Knows(ahmad,x) و سپس سؤال قبلی را بپرسیم، متغیر x هم باید ahmad و هم باید ali باشد که ناممکن است. پس طبق جداساز استاندارد، نام یکی را x و دیگری را y می‌گذاریم.
- طبق مثالی از کتاب اصلی، اگر در پایگاهی داشته باشیم: «همه، همه را می‌شناسند» یعنی Knows(y,z) و سپس سؤال قبلی را بپرسیم (Knows(x,ali))، یک جواب ممکن {x/y,z/ali} می‌باشد ولی جواب دیگر {z/ali,x/ali,y/ali} است. الگوریتم عمومی‌ترین یکسان‌ساز جواب اول را بعلت جامع بودن، قبول می‌کند.
- وارسی وقوع: در طی یکسان‌سازی در زمان هر تطبیق، باید بررسی شود که متغیر درون خود اصطلاح نباشد. در مثال Knows(x,ali) و Knows(father(x),ali)، وارسی وقوع تشخیص می‌دهد که نباید x با father(x) یکسان‌سازی شود.
- شبکه شمول برای جمله «Knows(ali,ahmad)» عبارتست از سلسله مراتب پرس‌وجوهایی که می‌توانند با این جمله یکسان‌سازی شوند:




۸ – در این کارگزار یک تابع سودمندی تعریف می‌شود که به هر حالت دنیا عددی انتساب می‌دهد که نشاندهنده سودمندی آن حالت می‌باشد. مثلا برای کارگزار تاکسی، تابع سودمندی تعریف می‌کنیم که ترکیبی از امنیت، سرعت و قانون باشد. بنابراین هر زمان که راننده بخواهد برای مسیر بعدی خود تصمیم بگیرد، برای تمام پسینها این تابع را محاسبه کرده و حالتی را انتخاب می‌کند که سودمندی آن طبق این تعریف بیشتر باشد. مثلا اگر در خیابانی باشد که با ۵۰۰ متر دنده عقب گرفتن به خیابانی نزدیک مقصد برسد، اینکار را نمی‌کند زیرا این انتخاب بعلت نداشتن امنیت، تناقض با قانون و سرعت پایین، سودمندی کمی دارد. بهتر است از سایر خیابانهای روبرو مسیری را انتخاب کند که سودمندی آن بالاتر باشد. ولی اگر این کارگزار مبتنی بر هدف بود، با سودمندی کاری نداشته و این حرکت را انجام می‌داد تا سریعتر به هدف برسد حتی اگر خطر داشته باشد. جزئیات داخلی این کارگزار در کتاب ترسیم شده است.

* – سه مساله تعدیل شده به همراه هیورستیک‌هایشان عبارتند از:

- یک کاشی می‌تواند از مربع A به مربع B حرکت کند، اگر A مجاور B باشد. (h=فاصله منهتن)
- یک کاشی می‌تواند از مربع A به مربع B حرکت کند، اگر B خالی باشد. (h= تابع گاشینگ طبق تمرین ۹.۴ کتاب اصلی (ویرایش دوم) یا تمرین ۳۱.۳ کتاب اصلی (ویرایش سوم))
- یک کاشی می‌تواند از مربع A به مربع B حرکت کند. (h= تعداد مربع‌هایی که در جای خود نیستند)

۱ – به پاسخ تمرین ۱۱.۴ کتاب اصلی (ویرایش دوم) و یا تمرین ۱.۴ کتاب اصلی (ویرایش سوم) مراجعه کنید.

۱۱ – حذف ⇔ (با جایگزینی $a\Leftrightarrow b$ با عبارت $a\Rightarrow b\wedge b\Rightarrow a$): $(p\Rightarrow(p\vee q))\wedge((p\vee q)\Rightarrow p)$

حذف ⇒ (با جایگزینی $a\Rightarrow b$ با $\neg a\vee b$): $(\neg p\vee p\vee q)\wedge(\neg(p\vee q)\vee p)$

اعمال هم ارزی های $(a\vee\neg a)\equiv True$ و $(True\vee a)\equiv True$ و $(True\wedge a)\equiv a$: $\neg(p\vee q)\vee p$

با اعمال قانون دمورگان (جایگزینی $\neg(a\vee b)$ با $\neg a\wedge\neg b$): $(\neg p\wedge\neg q)\vee p$

اعمال توزیع‌پذیری: $(\neg p\vee p)\wedge(\neg q\vee p)$

در نتیجه شکل نرمال عطفی عبارتست از: $\neg q\vee p$

۱۲ – الف) اگر داشته باشیم: American(x)=آمریکایی بودن x، Weapon(y)=اسلحه بودن y، Sell(x,y,z)=فروش اسلحه y توسط فرد x به فرد z، Hostile(z)=مخالف بودن z، Criminal(x)=مجرم بودن x، owns(Nano,x)=متعلق بودن x به نونو، Missile(x)=موشک بودن x. این مثال در کتاب اصلی ذکر شده است، طبق آن داریم:

«برای یک آمریکایی فروش اسلحه به ملل مخالف جرم است»:

$$R1: American(x)\wedge Weapon(y)\wedge Sell(x,y,z)\wedge Hostile(z)\Rightarrow Criminal(x)$$

«کشور نونو که دشمن آمریکاست، تعدادی موشک دارد»: $\exists x\ Owns(Nano,x)\wedge Missile(x)$

که با حذف سور وجودی به دو بند زیر تبدیل می‌شود:

$$R2: Owns(Nano,M1)$$
$$R3: Missile(M1)$$

«نونو تمامی موشک‌هایش را از سرهنگ وست که یک آمریکایی است، خریده است»:

$$R4: Missile(x)\wedge Owns(Nano,x)\Rightarrow Sell(West,x,Nano)$$

می‌دانیم که «موشک نوعی اسلحه است»: $R5: Missile(x)\Rightarrow Weapon(x)$

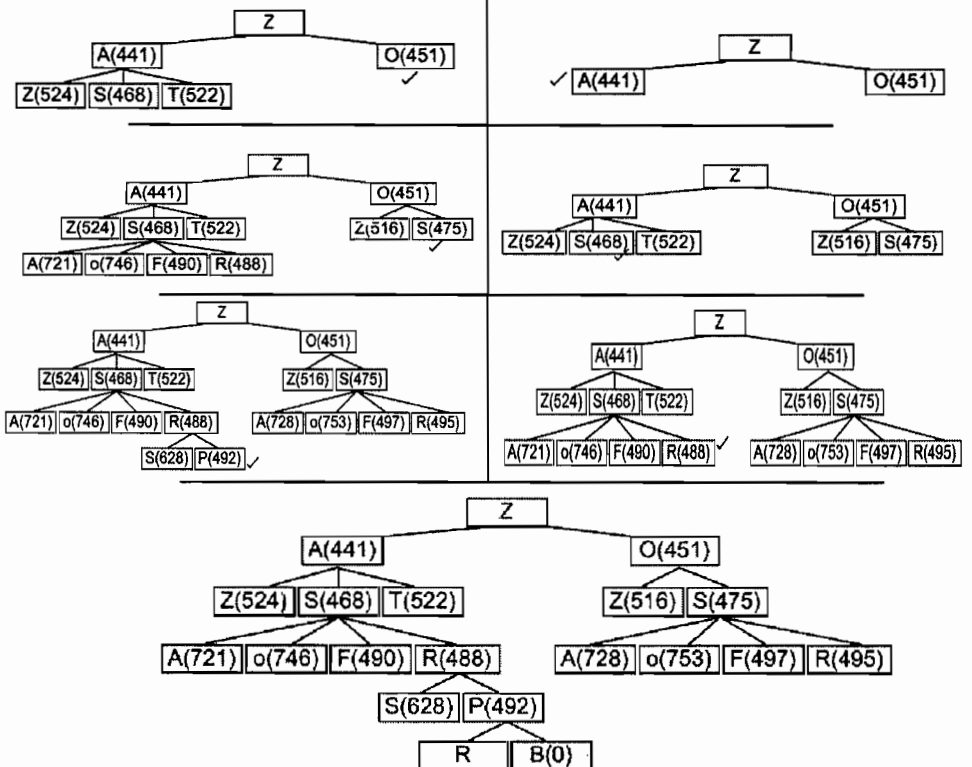
می‌دانیم که «تمام دشمنان آمریکا، مخالف هستند»: $R6: Enemy(x,America)\Rightarrow Hostile(x)$

می‌دانیم که «وست یک آمریکایی است»: $R7: American(West)$

می‌دانیم که «نونو دشمن آمریکا است»: $R8: Enemy(Nano,America)$

ب) درخت اثبات به وسیله الگوریتم زنجیره‌ای پس‌رو را در شکل ۷.۹ از کتاب اصلی (ویرایش دوم و سوم) و زنجیره پیش‌رو را در شکل ۴.۹ از کتاب اصلی (ویرایش دوم و سوم) مشاهده می‌کنید. باید درخت پس‌رو را بصورت اول‌عمق و از چپ به راست بخوانید. شکل کلی درخت اثبات برای زنجیره‌ای پس‌رو و پیش‌رو مشابه است و تفاوت آنها در مراحل تولید این درخت است.

۱۳ – درون هر گره نام آن به همراه مقدار ارزیاب یعنی $f=g+h$ را مشاهده می‌کنید. مراحل پیشروی الگوریتم به ترتیب از راست به چپ در زیر آمده است:



۱۱ – در مسائل ارضای محدودیت باید به تمام متغیرها مقداری از دامنه‌شان تخصیص دهیم که هیچ محدودیتی نقض نشود. دو روش انتخاب متغیر برای مقداردهی داریم: آروین‌درجه و آروین MRV.

آروین‌درجه، متغیری را انتخاب می‌کند که بالاترین درجه محدودیت در مقایسه با سایر متغیرهای بدون مقدار داشته باشد و بنابراین ضریب انشعاب در مراحل بعد کاهش یابد. آروین MRV متغیری را انتخاب می‌کند که در دامنه خود کمترین تعداد مقدار مجاز را داشته باشد.

- درجه متغیرها در ابتدا: SA=۵، NT=Q=NSW=۳، WA=V=۲ و T=0. در ابتدا SA با درجه ۵ انتخاب می‌شود.
- درجه متغیرها در مرحله بعد: NT=Q=NSW=۲، T=0، V=WA=۱. پس بطور تصادفی NT انتخاب می‌شود.
- در مرحله بعد متغیرهای NSW و V دارای کمترین مقدار مجاز در دامنه هستند. پس بطور تصادفی NSW را با مقدار G انتخاب می‌کنیم که باعث می‌شود دامنه متغیر Q تهی شود. پس به عقب بازگشته و مقدار NSW را B می‌دهیم.

| | WA | NT | Q | NSW | V | SA | T |
|---|---|---|---|---|---|---|---|
| Initial domains | RGB | RGB | RGB | RGB | RGB | RGB | RGB |
| After SA=R | GB | GB | GB | GB | GB | R | RGB |
| After NT=B | G | B | G | GB | GB | R | RGB |
| After NSW=G | G | B | {} | G | B | R | RGB |
| After NSW=B | G | B | G | B | G | R | RGB |
| After T=R | G | B | G | B | G | R | R |

۱۵ – قوانین را به ترتیب از R1 تا R8 مطابق با سؤال ۱۲ نامگذاری می‌کنیم. در روش پیشرو باید فرض کنیم تمام مقدمها درست هستند و سپس تالی را نتیجه بگیریم. ادامه می‌دهیم تا در تالی به عبارت Criminal(West) برسیم. برای اثبات مجرم بودن West طبق مثال کتاب، به دو مرحله نیاز داریم:

- مرحله اول:
  - + مقدمهای قاعده R1 صدق نمی‌کند.
  - + مقدم قاعده قاعده R4 با {x/M1} برقرار است.
  - پس تالی آن یعنی Sell(West,M1,Nano) به مجموعه قوانین افزوده می‌شود.
  - + مقدم قاعده قاعده R5 با {x/M1} برقرار است.
  - پس تالی آن یعنی Weapon(M1) به مجموعه قوانین افزوده می‌شود.
  - + مقدم قاعده R6 با {x/Nano} برقرار است.
  - پس تالی آن یعنی Hostile(Nano) به مجموعه قوانین افزوده می‌شود.
  - + مقدم قاعده قاعده R4 با {x/M1} برقرار است.
  - پس تالی آن یعنی Sell(West,M1,Nano) به مجموعه قوانین افزوده می‌شود.
- مرحله دوم:
  - + مقدم قاعده R1 با {x/West, y/M1, z/Nano} برقرار است.
  - پس تالی آن یعنی Criminal(West) به مجموعه قوانین افزوده می‌شود.

ه عبارت مربوطه رسیدیم، پس روش زنجیره‌ای پیشرو توانست آن را اثبات کند.

۱۷ –

| محیط | مقیاس کارایی | اقدام گرها | حسگرها | نوع عامل |
|---|---|---|---|---|
| بیمار، بیمارستان | سلامتی بیمار، کاهش هزینه | تجویز دارو، نوشتن آزمایش و پرسیدن سؤال | علائم، صحبتهای بیمار و نتایج آزمایش | سیستم تشخیص پزشکی |
| قطعات، تسمه حمل اشیا | قرار دادن اشیا در مکان صحیح | برداشتن اشیا، گذاشتن اشیا در مکان مربوطه | دریافت پیکسلها با رنگهای مختلف | ربات جابه‌جاکننده اشیاء |

۱ – جواب بخش الف و ب را می‌توانید در سؤال ۹ مشاهده کنید. مقدار آروینهای گفته شده برای شکل الف عبارتند از:

=فاصله منهتن: (12=1+2+0+1+1+2+3+2)

= تعداد مربع‌هایی که در جای خود نیستند: (7)

=تابع گاشینگ (7)

+ هر بار خانه خالی را با عدد صحیح آن جابجا می‌کنیم و اگر خانه خالی در جایی واقع شد که جای اصلی خودش باشد، یکی از خانه‌های اشتباه را با آن جابجا می‌کنیم تا بتوانیم ادامه دهیم. در اینجا باید به ترتیب، خانه خالی را با اعداد ۲، ۵، ۸، ۱، ۷، ۳ و ۴ جابجا کنیم که ۷ بار می‌شود.

ج) اگر برای یک مساله تعدادی آروین قابل‌قبول مانند $h_1(n),..,h_m(n)$ موجود بوده و ندانیم که کدامیک را انتخاب کنیم، آروین زیر یک انتخاب قابل‌قبول و سازگار از روی آنها ارائه می‌دهد:

$$h(n)=Max\{h_1(n),..,h_m(n)\}$$

۱۸ – ادراکات کارگزار شامل موارد زیر است: درک بو، درک نسیم، دیدن چاله، دیدن ومپوز، دیدن طلا و درک روشنایی طلا. در اینجا فقط از ادراکات ابتدایی کارگزار در نوشتن جملات استفاده می‌کنیم تا عبارات ساده‌تر شوند. الف)

R1: $\neg B1,1$
R2: $\neg S1,1$
R3: $\neg P1,1$
R4: $S2,1$
R5: $\neg B2,1$
R6: $\neg P2,1$
R7: $\neg B2,2$
R8: $\neg S2,2$
R9: $\neg P2,2$
R10: $B1,2$
R11: $\neg S1,2$
R12: $\neg P1,2$
R13: $B2,3$
R14: $\neg S2,3$
R15: $\neg P2,3$

ب) برای اثبات هدف یعنی $(\neg P1,2)$ فقط کافی است خانه‌های (1,1) و (2,1) بازدید شوند. پس فقط جملات R1 و R6 به همراه جملات زیر برای اثبات هدف نیاز است:

R16: $\neg B1,1 \Leftrightarrow (\neg P2,1 \wedge \neg P1,2)$
R17: $\neg B2,1 \Leftrightarrow (\neg P1,1 \wedge \neg P2,2 \wedge \neg P3,1)$

با توجه به آنکه بخش مقدم R16 همان قانون R1 است پس طبق قیاس استثنایی، می‌توان بخش تالی را نتیجه گرفت و دانش زیر را کسب کنیم: R19: $(\neg P2,1 \wedge \neg P1,2)$

طبق قاعده تحلیل (رزولوشن) از ترکیب قاعده R19 و R6 داریم: R20: $\wedge \neg P1,2$

که به هدف رسیدیم.

۱۹ – حذف نماد استلزام (با جایگذاری $\neg\alpha \vee \beta$ به جای $\alpha \Rightarrow \beta$):

$\forall x \neg [\neg \forall y\ Animal(y) \vee Loves(x,y)] \vee [\exists y\ Loves(y,x)]$

• استفاده از هم‌ارزی $(\neg \forall x\ Q \equiv \exists x\ \neg Q)$:

$\forall x \quad [\exists y\ Animal(y) \wedge \neg Loves(x,y)] \vee [\exists y\ Loves(y,x)]$

طبق قانون استانداردسازی متغیرها، نباید از نام یک متغیر برای دو منظور در یک عبارت در نظر گرفت شود (در اینجا متغیر y دوبار استفاده شده است). پس داریم:

$\forall x \quad [\exists y\ Animal(y) \wedge \neg Loves(x,y)] \vee [\exists z\ Loves(z,x)]$

حذف سور وجودی (با اسکولم‌سازی):

$\forall x\ Animal(Q(x)) \wedge \neg Loves(x,Q(x))] \vee [Loves(P(x),x)]$

حذف سور عمومی:

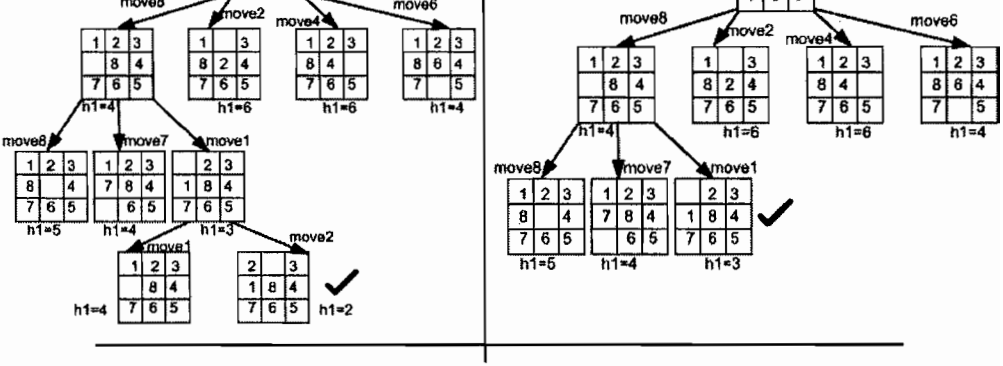
$[Animal(Q(x)) \wedge \neg Loves(x,Q(x))] \vee Loves(P(x),x)$

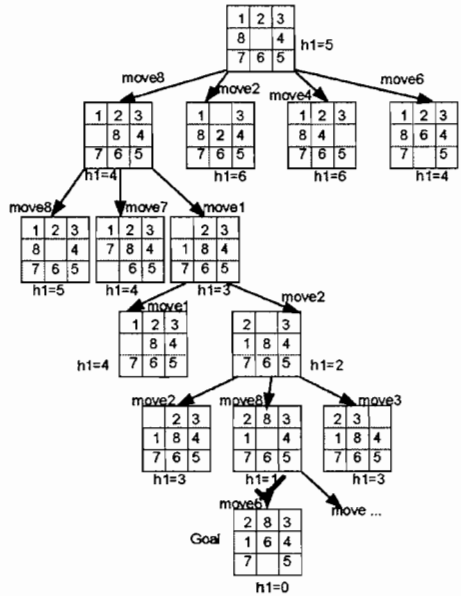
• اعمال خاصیت توزیع‌پذیری:

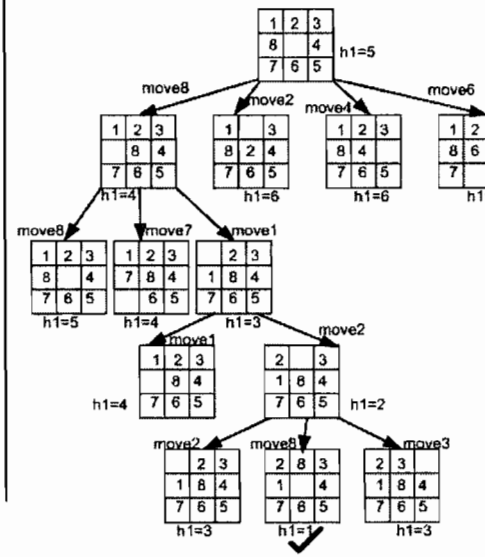
$[Animal(Q(x)) \vee Loves(P(x),x)] \wedge [\neg Loves(x,Q(x)) \vee Loves(P(x),x)]$

۲۰ – الف) توابع پیشنهادی برای هیورستیک عبارتند از: $h_1$= فاصله منهتن و $h_2$=تعداد خانه‌های اشتباه.

ب) نتیجه الگوریتم A* با تابع $h_1$=منهتن را بطور مرحله به مرحله می‌بینید. مراحل رسیدن به هدف به ازای تابع دیگر به خواننده واگذار می‌شود.






ج) بدلیل اینکه $h_1>h_2$ پس موثرتر است که این موضوع در بخش ب مشخص است.

۲۱ – الف و ب) می‌دانیم که هر وزیر باید در یک ستون قرار داشته باشد. پس برای هر وزیر فقط شماره سطری که در آن قرار دارد را می‌نویسیم. با داشتن ۸ وزیر، ۸ عدد داریم. پس هر چیدمان مختلف از ۸ وزیر را می‌توان با ۸ عدد نمایش داد. برای تولید جمعیت اولیه در ژنتیک بطور تصادفی ۸ وزیر را در ستون مربوط به خودشان حرکت داده و در هر بار رشته تولید شده را یادداشت می‌کنیم. هر رشته یک کروموزوم (یکی از افراد جمعیت) می‌باشد.

ج) اگر فرض کنیم دو نفر از جامعه (دو چیدمان مختلف ۸ وزیر) بصورت زیر کد شده باشند: 32752411 , 24748552

جایگاه برای پیوند آنها می‌توان هر یک از ۸ عدد فوق را برگزید. در اینجا پیوند را از عدد سوم انجام می‌دهیم. حاصل پیوند عبارتست از:

32748552 , 24752411

۲۲ – الف) $B_{x,y}\Leftrightarrow(P_{x+1,y}\vee P_{x-1,y}\vee P_{x,y-1}\vee P_{x,y+1})$

این عبارت برای خانه‌های (۱و۱) و (۱و۲) عبارتست از:

$$B_{1,1}\Leftrightarrow(P_{1,2}\vee P_{2,1})$$
$$B_{2,1}\Leftrightarrow(P_{1,1}\vee P_{2,2}\vee P_{3,1})$$

ب) حذف $\Leftrightarrow$ (با جایگزینی $a\Leftrightarrow b$ با عبارت $a\Rightarrow b\wedge b\Rightarrow a$):

$$(B_{1,1}\Rightarrow(P_{1,2}\vee P_{2,1}))\wedge((P_{1,2}\vee P_{2,1})\Rightarrow B_{1,1})$$

حذف $\Rightarrow$ (با جایگزینی $a\Rightarrow b$ با $\neg a\vee b$ ):

$$(\neg B_{1,1}\vee P_{1,2}\vee P_{2,1})\wedge(\neg(P_{1,2}\vee P_{2,1})\vee B_{1,1})$$

اعمال قانون دمورگان (جایگزینی $\neg(a\vee b)$ با $\neg a\wedge\neg b$) و توزیع‌پذیری شکل نرمال عطفی حاصل می‌شود:

$$(\neg B_{1,1}\vee P_{1,2}\vee P_{2,1})\wedge(\neg P_{1,2}\vee B_{1,1})\wedge(\neg P_{2,1}\vee B_{1,1})$$

۲۳ – قواعد را به ترتیب از A تا B نامگذاری می‌کنیم و نقیض هدف را بصورت $\neg G$ به مجموعه قواعد می‌افزاییم. یعنی:

$\neg G$: $\neg$Kills(Curiosity,Tuna)

قانون تمام جملات را به فرم CNF تبدیل می‌کنیم:

A1': Animal(F(x)) $\vee$ Loves(G(x),x)
A2': $\neg$Loves (x,F(x)) $\vee$ Loves(G(x),x)
B': $\neg$ Animal(y) $\vee$ $\neg$Kills(x,y) $\vee$ Loves(z,x)
C': $\neg$ Animal(x) $\vee$ Loves(Jack,x)
D': Kills(Jack,Tuna) $\vee$ Kills(Curiosity,Tuna)
E': Cat(Tuna)
F': $\neg$Cat(x) $\vee$ Animal(x)

¬G': ¬Kills(Curiosity,Tuna)

حال برای اثبات اشتباه بودن این نقیض هدف داریم:

R1: Kills(Jack,Tuna) (result of D' and ¬G')
R2: ¬ Animal(Tuna) ∨ Loves(z,Jack) (result of B' and R1)
R3: ¬ Animal(Tuna) (result of C' and R2)
R4: ¬Cat(Tuna) (result of F' and R3)
R5: {} (result of E' and R4)

۲۴ – مسئله ارضای محدودیت از سه عنصر {متغیر، دامنه و محدودیت} تشکیل شده است و منظور از حل آن این است که به تمام متغیرها، مقداری از دامنه تخصیص دهیم که هیچ محدودیتی نقض نشود. یکی از روشهای رسیدن به این راه‌حل، پسگرد ساده می‌باشد. در این روش به یک متغیر مقداری از دامنه‌اش تخصیص داده و به سراغ متغیر بعدی می‌رویم. و به آن نیز مقداری می‌دهیم که با شرایط محدودیت سازگار باشد. به همین ترتیب به سراغ متغیرهای بعدی می‌رویم تا جاییکه برای یک متغیر نتوان هیچ مقداری از دامنه‌اش انتخاب نمود. در اینصورت یک مرحله به عقب پسگرد کرده و مقدار متغیر قبلی را تغییر می‌دهیم. اگر آن متغیر هم دیگر مقدار مجازی نداشته باشد مجبوریم یک مرحله دیگر نیز به عقب برگردیم. الگوریتم زمانی پایان می‌یابد که همه مقدار گرفته و تناقض مشاهده نشود.

۲۵ – می‌دانیم که بازیکن Max همواره حرکتی را انجام می‌دهد که بیشترین امتیاز را کسب کند و بازیکن Min حرکتی می‌کند که کمترین امتیاز حاصل شود. روشی بنام Minimax وجود دارد که با توجه به امتیاز برگهای درخت، تعیین می‌کند که در ابتدا بازیکن Max چه حرکتی را کند تا به بیشترین امتیاز ممکن دست یابد. برای اینکار درخت را از پایین به بالا پیمایش می‌کنیم.

- اگر پدر امتیازها، یک گره Min باشد: از بین فرزندان هر پدر کمترین امتیاز را برای پدر یادداشت می‌کنیم.
- اگر پدر امتیازها، یک گره Max باشد: از بین فرزندان هر پدر بیشترین امتیاز را برای پدر یادداشت می‌کنیم.

سپس به سراغ یک سطر بالاتر رفته و همین کار را تکرار می‌کنیم. تا به ریشه درخت برسیم. آنگاه بازیکن آغازگر بازی می‌داند که باید کدام حرکت را انجام دهد. می‌توانید به شکل ۲.۶ کتاب اصلی (ویرایش دوم) یا شکل ۲.۵ کتاب اصلی (ویرایش سوم) مراجعه کنید.

۲۶ – جستجوی تپه‌نوردی که از دسته روشهای جستجوی‌محلی محسوب می‌شود به سان عملکرد یک انسان نابینا بر روی دامنه کوه است که موقعیت خود را نمی‌داند ولی قصد رسیدن به قله را دارد. این فرد در هر گام، با استفاده از پای خود زمین اطراف را لمس کرده و هر سمتی که به سوی بالا باشد را ادامه می‌دهد. اگر به موقعیتی برسد که همه نقاط اطرافش رو به پایین باشند، نتیجه می‌گیرد که به قله کوه رسیده است. بنابراین اگر به جایی برسد که قله اصلی کوه نباشد ولی قله‌ای کوچک محسوب شود این فرد متوقف می‌شود. ولی در روش پرتومحلی، فرض می‌کنیم k نفر بر روی نقاط مختلف کوه قرار دارند و همگی در هر گام، بهترین پسین خود را انتخاب می‌کنند و انتخاب بهتر را با یکدیگر به اشتراک می‌گذارند. (مشابه حرکت k فرد نابینا که اطلاعات گام بعدی خود را با موبایل به هم اطلاع می‌دهند). بنابراین همگی به مرور در اطراف انتخاب برتر جمع می‌شوند. می‌توان همین ایده را به جای پیمایش بر روی یک کوه، بر روی نموداری ریاضی که نشاندهنده کارایی است درنظرگرفت و یا با اعمال کمی تغییرات در آن به جای یافتن ماکزیمم مطلق، مینیمم مطلق را بیابیم.

۲۷ – یک جستجوی ناآگاهانه است که در آن یک درخت فضای‌حالت را داریم و می‌خواهیم گره هدف را بیابیم. در این الگوریتم از حالت شروع (گره ریشه) آغاز به گسترش نموده و تمامی پسین‌های ریشه را گسترش می‌دهیم. سپس آنها را با آزمون پایانی از جهت هدف/غیرهدف بودن بررسی می‌کنیم. اگر هدف بود، جستجو پایان می‌یابد و گرنه تمام پسین‌های آن گره‌ها را مجددا بسط داده و سطربعدی درخت تشکیل می‌شود. این کار آنقدر ادامه می‌یابد تا هدف یافته شود و یا دیگر پسینی وجود نداشته باشد.

۲۸ – یک روش جستجوی ناآگاهانه است که در یک درخت فضای حالت به جای گسترش تمام گره‌های هم‌سطح، گره‌ای را برای گسترش مرحله بعد انتخاب می‌کند که کمترین هزینه مسیر را داشته باشد. اگر در درخت مربوطه، هزینه تمام پسینها مساوی باشد، این روش با اول-سطح یکسان می‌شود.

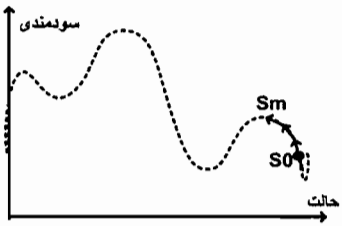
۲۹ – اگر محیطی که کارگزار در آن قرار می‌گیرد، نیمه‌مشاهده‌پذیر باشد، کارگزار بهتر است تاریخچه‌ای از آنچه مشاهده می‌کند را درخود نگه دارد. تا اگر به حالتی برسد که برخی از جنبه‌های محیط، غیرقابل‌مشاهده بودند بتواند از این تاریخچه کمک بگیرد. بعنوان مثال کارگزار راننده تاکسی را درنظربگیرید. بهتر است موقعیت سایر خودروهای مجاور را در تاریخچه نگه دارد زیرا شاید در لحظه بعدی نتواند همه انها را ببیند ولی در ذهن خود می‌داند که در فلان مکان ماشینی در حال راهنما زدن بوده است، پس شاید در این لحظه مقابل او باشد و بهتر است سرعت خود را کم کند. به آنچه کارگزار از تاریخچه محیط پیرامون ذخیره می‌کند «مدل دنیا» گفته می‌شود که باید هر از گاهی بروزرسانی شود. کارگزار برای هر تصمیم جدید، ادراک فعلی خود را با مدل دنیا ترکیب کرده و بعد تصمیم می‌گیرد. این کارگزارها مبتنی بر مدل نامیده می‌شوند.

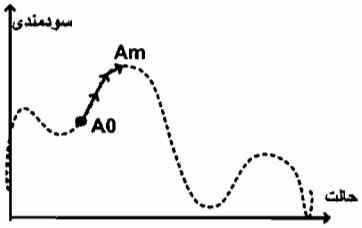
۳۰ – اول سطح، هزینه یکنواخت، اول عمق، عمق محدود، اول عمق عمیق شونده تکراری، دو طرفه.

۳۱ – یکی از روشهای بازنمایی دانش محسوب می‌شود که به جای منطق گزاره‌ای قرار می‌گیرد. در این بازنمایی یک گراف ترسیم می‌شود که در آن اشیا را درون هر گره و روابط بین اشیا را بر روی یالها می‌نویسیم.

۳۲ – به پاسخ سؤال ۲۵ مراجعه شود.

۳۳ – توضیحات مربوط به تپه‌نوردی در سؤال ۲۶ آورده شده است. در شکل زیر تابع سودمندی را برای حالات مختلف یک مساله مانند پازل ۸– تایی مشاهده می‌کنید. فرض کنید برای حالت اولیه بازی، سودمندی را محاسبه کرده و در شکل زیر با S0 نمایش داده‌ایم. جستجوی تپه‌نوردی، سودمندی S0 را با دو نقطه مجاورش مقایسه کرده و نقطه بیشتر را انتخاب می‌کند و به آنجا می‌رود. با ادامه همین روال به Sm می‌رسد که یک ماکزیمم محلی است و متوقف می‌شود. اگر فرض کنیم در همین مساله از حالت A0 شروع به حرکت کنیم، طبق تپه‌نوردی به Am می‌رسیم که ماکزیمم سراسری است.




۳ – الف) تست تورینگ یک آزمون است بطوریکه از یک انسان و یک روبات خواسته می‌شود در دو اتاق مجزا نشسته و با استفاده از مینالی که در اختیار آنها قرار می‌گیرد به سؤالات پاسخ دهند. یک نفر را مسئول می‌کنیم تا سؤالاتی را از هر دوی آنها بپرسد و جواب را یافت کند. اگر این مسئول نتواند تشخیص دهد که در کدام اتاق انسان و در کدام اتاق روبات است، تست تورینگ موفق شده است.

ب) • بدلیل اینکه تنها راه تشخیص از روی پاسخ سؤالات است پس سایر حواس در تشخیص انسان از روبات کارساز نخواهد بود.

تست تورینگ یک آزمون تعیین هوشمندی است.

۳ – به پاسخ سؤال ۲۵ مراجعه شود.

۳ – الگوریتم تپه‌نوردی در یک مساله:

ب) گره ریشه را درون صف قرار بده.

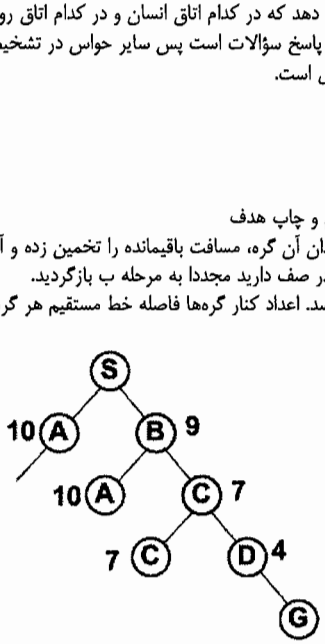
ج) گره‌ای را از صف خارج کن:

اگر آن گره هدف است -> پایان الگوریتم و چاپ هدف

اگر آن گره غیرهدف است -> برای فرزندان آن گره، مسافت باقیمانده را تخمین زده و آنها را بطور مرتب در صف قرار دهید.

د) اگر به هدف نرسیده‌اید و یا هنوز گره‌ای در صف دارید مجددا به مرحله ب بازگردید.

خت زیر طبق تپه‌نوردی به هدف G می‌رسد. اعداد کنار گره‌ها فاصله خط مستقیم هر گره تا هدف هستند:



# ضمیمه ۳ (سؤالات کنکور ارشد)

۱ – رویهٔ روبه‌رو را درنظر بگیرید.
تعداد مقایسه‌ها (تعداد <ها درسطر ۶) بر حسب n چند است؟

۱) n-1 ۲) $n-(\lceil n/2 \rceil - \lfloor n/2 \rfloor)$ ۳) n ۴) $\lceil \lg n \rceil$

```
FindMax(A,i,j)
1   n←j-i+1
2   if n=1
3      then return A[i]
4      else m1← FindMax(A,i,i+⌊n/2⌋-1)
5           m2← FindMax(A,i+⌊n/2⌋,j)
6           if m1<m2
7                then return m2
8           else return m1
```

۲ – در یک گراف جهت‌دار G=(V,E) وزن هر کدام از یال‌ها یک عدد صحیح، مثبت و حداکثر C است که C یک عدد ثابت است. می‌خواهیم طول کوتاه‌ترین مسیرها از یک راس به نام s را تا بقیهٔ راس‌های G بدست آوریم کدامیک از گزینه‌های زیر مرتبهٔ یک الگوریتم کارا برای حل این مسئله است؟

۱) O(|E| lg|V|) ۲) O(|V| lg|V|+|E|) ۳) O(|V|+|E|) ۴) O((|V|+|E|) lg|E|)

۳ – الگوریتم فلوید کوتاه‌ترین مسیر بین همهٔ زوج نقطه‌ها را درگراف جهت‌دار و وزن‌دار G محاسبه می‌کند. اگر وزن برخی (حداقل یکی از) یال‌ها منفی باشد کدامیک از گزینه‌های زیر بهترین توصیف برای عملکرد این الگوریتم به هنگام وجود یال منفی است؟ فرض کنید در این الگوریتم، فاصلهٔ هر راس تا خودش همواره صفر بوده و تغییر نمی‌کند.
۱) با تکرار این الگوریتم می‌توان وجود یا عدم وجود دور منفی را تشخیص داد.
۲) حتی اگر دور منفی نداشته باشیم ممکن است در حلقهٔ نامتناهی بیفتد.
۳) تنها در حالتی در حلقهٔ نامتناهی می‌افتد که دور منفی داشته باشیم.
۴) حتی با داشتن دور منفی هم متوقف می‌شود اما ممکن است در حالتی هم که دور منفی نداریم درست کار نکند.

۴ – آرایهٔ A به طول n بصورت حلقوی مرتب است. یعنی اگر ابتدا و انتهای A را به هم بچسبانیم از یک دریچه‌ای به بعد (بصورت حلقوی) آرایه مرتب است. مثلا آرایه

| ۵۰ | ۴۰ | ۳۰ | ۲۰ | ۱۰ | ۸۰ | ۷۰ | ۶۰ |
|---|---|---|---|---|---|---|---|

بصورت حلقوی مرتب است. بهترین الگوریتم برای یافتن کوچکترین عنصر A از چه مرتبه‌ای است؟

۱) O(lg lg n) ۲) O(lgn) ۳) $O(lg^2 n)$ ۴) O (n)

۵ – دورترین راس از یک راس داده شدهٔ v در یک گراف بدون‌وزن، راسی است که کوتاه‌ترین فاصلهٔ آن تا v بیشترین باشد. کدامیک از روش‌های زیر برای یافتن این دورترین راس مناسب‌تر و سریع‌تر است؟

۱) دایکسترا ۲) مرتب‌سازی توپولوژیکی ۳) DFS ۴) BFS

۶ – کدام گزینه درستی یا نادرستی گزاره‌های زیر را بیان می‌کند.
a) در Radix-sort اگر برای مرتب‌سازی رقم‌ها به جای مرتب‌سازی شمارشی (conut sort) از مرتب‌سازی ادغامی استفاده کنیم الگوریتم سریع‌تر خواهد بود.
b) اگر آرایه تقریبا مرتب باشد مرتب‌سازی درجی (Insersion sort) از مرتب‌سازی سریع (Quick sort) سریع‌تر مرتب می‌کند.

۱) a نادرست، b نادرست ۲) a نادرست، b درست ۳) a درست، b نادرست ۴) a درست و b درست

۷ – در یک مسئله طرح‌ریزی (planning) عاملی بناست مواد لازم را در محل‌های مختلف در آزمایشگاه یافته یا براساس فرمول های موجود با استفاده از مواد یافته شده تهیه کند. عملگرهای مجاز آن مکان‌یابی اشیا، حرکت در محیط، حمل و ادغام مواد است. در این مسئله هدف بصورت عطفی (conjunction) از گزاره‌ها معرفی شده و برای حل آن از الگوریتم pop استفاده می‌شود. کدامیک از جملات زیر صحیح است؟
۱) تعداد جملات موجود درلیست پیش شرایط باز یک هیوریستیک قابل‌قبول برای جستجوی A* در این فضاست.
۲) مسئله نمی‌تواند به زبان STRIPS بیان شود مگر آن که بجای عطف بین جملات هدف، از اپراتور فصل (disjunction) استفاده شود.

ممکن است راه‌حلی با ترتیب جزئی (partial order plan) برای مسئله پیدا شود که هیچ خطی‌سازی از آن منجر به یک حل با ترتیب کامل نشود.

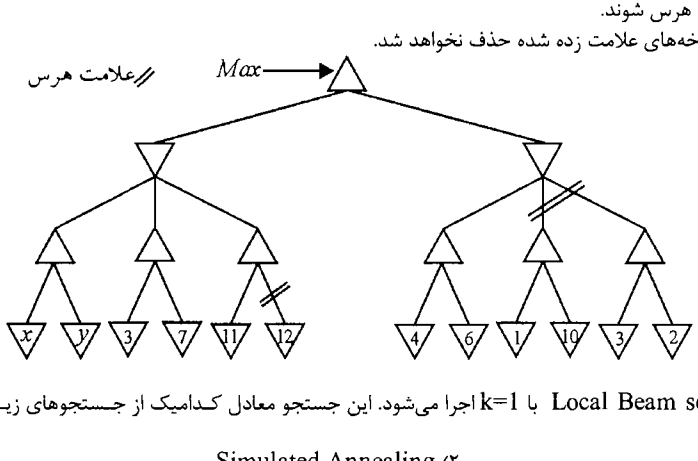
در این مسئله هر طرح با ترتیب جزئی که در آن پیش شرط بازی باقی‌نمانده باشد و هیچ تهدیدی هم وجود نداشته باشد اب مسئله است.

– فرض کنید رباتی دارید که برای حمل اشیاء سنگین استفاده می‌شود. اگر شی‌ای سنگین‌تر از آستانه تحمل ربات روی آن ر دهید، به احتمال ۸۰٪ چراغ خطر آن روشن می‌شود. در غیر اینصورت (اگر شی سنگین‌تر ازآستانه نباشد) فقط ۱۰٪ احتمال ن شدن چراغ خطر هست. روشن‌شدن چراغ خطر در ۹۰٪ موارد منجر به خاموش شدن موتور ربات می‌شود. در ۲٪ موارد کن است بدون روشن شدن چراغ، موتور ربات خاموش شود. اگر فقط ۱۰٪ اشیا قابل حمل سنگین‌تر از آستانه تحمل ربات ند احتمال این که بعد ازگذاشتن شی روی ربات، چراغ روشن نشود ولی موتور خاموش شود چقدر است؟

۱) ٪۰/۰۴ ۲) ٪۱/۶۶ ۳) ٪۲ ۴) ٪۰/۸۱

– در درخت بازی زیرکدام یک ازگزینه‌ها محدوده مناسبی برای مقادیر مثبت x و y تعیین می‌کند بطوری که شاخه‌های مت زده شده درهرس آلفا – بتا هرس شوند.

به ازای هیچ مقداری از x,y شاخه‌های علامت زده شده حذف نخواهد شد.




۱) $\forall x,y\ 0<x\leq 3\ ,\ 0<y\leq 3$

۲) $\forall x,y\ 6-x-y>0$

۳) $\forall x,y\ x<5\ ,\ y>7$

– فرض کنید الگوریتم Local Beam search با k=1 اجرا می‌شود. این جستجو معادل کدامیک از جستجوهای زیر ت؟

۱) Hill Climbing ۲) Simulated Annealing

۳) Genetic Algorithm ۴) Constraint Satisfaction

– اگر $KB_1$ یک پایگاه‌دانش دلخواه و b یک جمله دلخواه باشد با فرض اینکه افزودن b به پایگاه $KB_1$ منجر به ایجاد پایگاه K می‌شود اگر بدانیم جمله a از $KB_1$ قابل نتیجه‌گیری است کدامیک از گزینه‌های زیر درست است؟

۱) $a\wedge b$ از $KB_1$ قابل نتیجه‌گیری است. ۲) a از KB2 قابل نتیجه‌گیری است.

۳) $a\wedge b$ از $KB_2$ قابل نتیجه‌گیری است. ۴) $a\vee b$ هم از $KB_1$ و هم از $KB_2$ قابل نتیجه‌گیری است.

– فرض کنید برای مسئله‌ای با جستجوی اول‌پهنا (breadth-first) و تست هدف در لحظه تولید، نیاز به بسط دادن (expan) ۳۲ گره باشد. اگر فاکتورانشعاب (branching factor) درخت جستجو ثابت باشد و عمق‌درخت برابر ۵ و عمق ف (goal) برابر ۴ باشد، کدامیک ازگزینه‌ها مقدار فاکتور انشعاب (b) را نشان می‌دهد (فرض کنید ریشهٔ درخت در عمق ر (0) واقع شده است.)

۱) b=2 ۲) b>5 ۳) 2<b<3 ۴) 3≤b≤5

– کدامیک از گزینه‌های زیر غلط است؟

الگوریتم شبیه‌سازی حرارت حتما از بهینه محلی فرار می‌کند.

الگوریتم تپه‌نوردی حتما در بهینه محلی گیر می‌کند.

الگوریتم پرتومحلی ممکن است بهینه عمومی را پیدا کند.

الگوریتم ژنتیک ممکن است از بهینه محلی فرار کند.

– کدامیک از عبارات زیر غلط است؟

۱) در زنجیره عقب‌گرد باید از عبارات هورن استفاده کرد ۲) در زنجیره جلورو استنتاج از حقایق شروع می‌شود.

۳) سرعت زنجیره جلورو بیشتر از زنجیره عقب‌گرد است ۴) یک عبارت هورن فقط یک لیترال منفی دارد.

– کدامیک از الگوریتمهای زیر به عنوان یک الگوریتم جستجو نمی‌تواند مطرح باشد؟

۱) برش آلفا بتا ۲) ژنتیک ۳) minmax ۴) A*

– کدامیک از عبارات زیر درمورد روش رزولوشن غلط است؟

۱) حقایق را باید تبدیل به CNF کرد. ۲) در پایگاه‌دانش نباید تناقض وجود داشته باشد.

۳) پایگاه دانش باید به صورت CNF باشد. ۴) نتیجه‌ایی که باید اثبات گردد باید تبدیل به CNF گردد.

۱۷ – کدامیک از عبارات زیر در مورد الگوریتم ژنتیک غلط است؟

۱) جمعیت اولیه به صورت تصادفی یکنواخت تعیین گردد. ۲) انتخاب بازماندگان متناسب با شایستگی است.

۳) فرزندان توسط بازترکیبی و جهش تولید می‌شوند. ۴) شرط خاتمه می‌تواند ماکزیمم تعداد نسل باشد.

۱۸ – کدامیک از عبارات زیر معادل رتبه اول جمله زیر است؟

کسی که به همه بدبین است به هیچ‌کس نمی‌تواند اعتماد کند.

۱) $\exists x\ \forall y\ B(x,y) \Rightarrow \nexists z\ \sim E(x,z)$

۲) $\exists x\ \forall y\ B(x,y) \Rightarrow \forall z\ \sim E(x,z)$

۳) $\forall x\ \forall y\ B(x,y) \Rightarrow \forall z\ \sim E(x,z)$

۴) $\forall x\ \forall y\ B(x,y) \Rightarrow \nexists z\ \sim E(x,z)$

۱۹ – روی یک توری $n*n$ که هر خانه به چهار همسایه خود متصل است. خانه میانی را نقطه شروع جستجو و نقطه (0,0) در نظرمی‌گیریم. گره هدف در موقعیت (x,y) است. در این گراف الگوریتم جستجوی A بدون تست تکراری‌بودن حالات، حداکثر $(4^{x+y+1}-1)/3)-1$ گره و الگوریتم جستجوی B با تست تکراری‌بودن حالات حداکثر $2(x+y)(x+y+1)-1$ گره را قبل از یافتن جواب بسط می‌دهند. کدامیک از گزینه‌های زیر در مورد این دو الگوریتم صحیح است؟

۱) A و B هر دو الگوریتم اول پهنا (Breadth first) هستند.

۲) A و B هر دو الگوریتم اول عمق (Depth first) هستند.

۳) A الگوریتم اول پهنا (Breadth first) و B الگوریتم اول عمق (Depth first) است.

۴) A الگوریتم اول عمق (Depth first) و B الگوریتم اول پهنا (Breadth first) است.

۲۰ – کدامیک ازگزینه‌های داده شده بازنمایی جملهٔ «هر دانش‌آموزی حداقل دو دوست دارد.» به منطق مرتبه اول است؟

۱) $\forall x,y\ student(x) \Rightarrow number(friend(y,x)) \geq 2$

۲) $\forall x,y\ student(x) \wedge friend(y,x) \Rightarrow number(y) \geq 2$

۳) $\forall x,y,z\ student(x) \wedge friend(y,x) \wedge friend(z,x) \Rightarrow y \neq z$

۴) $\forall x\ student(x) \Rightarrow \exists y,z\ friend(y,x) \wedge friend(z,x) \wedge y \neq z$

۲۱ – فرض کنید فضای جستجویی دارای پنج گره A، B، C، D و E باشد. جدول زیر فواصل واقعی این گره‌ها را از هم نشان می‌دهد. (وجود عدد در هر خانه جدول نشاندهندهٔ این است که از گره مربوط به سطر به سمت گره مربوط به ستون مسیری به طول عدد وجود دارد). اگر گره A گره شروع، گره E گره هدف و تابع h تابع مکاشفه‌ای تخمین فاصله گره تا هدف باشد کدامیک از گزینه‌های زیر صحیح است.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | | ۱۰ | ۸ | ۲ | |
| B | ۱۰ | | ۲ | | ۲ |
| C | ۸ | | | ۲ | ۶ |
| D | ۲ | | ۲ | | ۹ |
| E | | ۲ | ۶ | ۹ | ۰ |

۱) اگر $h(B)=3$، $h(C)=6$ و $h(D)=9$ آنگاه تابع h یک تابع یکنواخت (monotonic) و قابل‌قبول (admissible) است.

۲) اگر $h(B)=1$، $h(C)=5$ و $h(D)=8$ آنگاه تابع h یک تابع یکنواخت (monotonic) و قابل‌قبول (admissible) است.

۳) اگر $h(B)=1$، $h(C)=5$ و $h(D)=8$ آنگاه تابع h یک تابع یکنواخت (monotonic) نیست ولی قابل قبول (admissible) است.

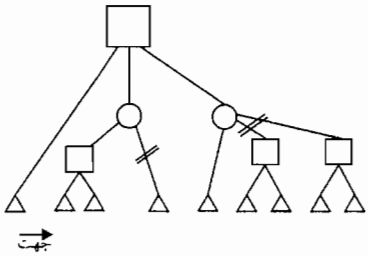
۴) اگر $h(B)=3$، $h(C)=3$ و $h(D)=8$ آنگاه تابع h یک تابع یکنواخت (monotonic) است ولی قابل قبول (admissible) نیست.

۲۲ – فرض کنید در سفری به سیبری راه خود را گم کرده‌اید و بعد از مدتی به یک شهرمی‌رسید. دقیقا نمی‌دانید نام این شهر چیست ولی با توجه به محدوده گم شدنتان حدس می‌زنید این شهر باید یکی از چهار شهر A، B، C یا D باشد. با توجه به اطلاعاتی که در مورد مکان این چهار شهر دارید احتمال حضور در هر یک از چهار شهر A، B، C و D را به ترتیب ۱۰٪ و ۴۰٪ و ۱۰٪ و ۳۰٪ می‌دانید. سپس از یکی از اهالی سؤالی می‌پرسید و او به زبان اسپرانتو پاسخ می‌دهد. شما می‌دانید ۵۰٪ شهر A،

۲۰٪ مردم شهر B، ۴۰٪ مردم شهر C و ۳۰٪ مردم شهر D به زبان اسپرانتو صحبت می‌کند. حالا با بیشترین احتمال خود را درکدام شهر می‌دانید.

۱) A ۲) B ۳) C ۴) D

۲۲ – درگراف مقابل مربع نشانه بازیکن Max، دایره نشانه بازیکن Min و مثلث نشانه حالت پایانی است. اگر مقادیر ارزیابی توانند در فاصله بسته [0,10] باشند و با هرس آلفا-بتا فقط یال‌های علامت‌زده شده با // حذف شوند، ترتیب گره‌های پایانی به‌ترتیب از چپ به راست در شکل کدامیک از گزینه‌های زیر خواهد بود؟




۱) 0,1,2,3,4,9,8,9,10 ←جهت
۲) 10,9,8,5,4,3,2,1,0 ←جهت
۳) 9,5,3,10,8,0,1,2,4 ←جهت
۴) 8,2,5,3,9,10,0,1,4 ←جهت

۲۴ – اگر بدانیم:

$E \wedge R \Rightarrow B$
$E \Rightarrow R \vee P \vee L$
$K \Rightarrow B$
$\neg(L \wedge B)$
$P \Rightarrow \neg K$

کدامیک از موارد زیر با استدلال منطقی قابل نتیجه‌گیری نیست؟

۱) $E \wedge P$ ۲) $K \wedge E \Rightarrow R$ ۳) $L \vee P \Rightarrow \neg K$ ۴) $L \Rightarrow \neg(K \wedge E)$

۲۵ – کدامیک از گزینه‌های زیر از نظر منطقی همیشه درست است.

۱) $(smoke \Rightarrow fire) \Rightarrow ((smoke \wedge heat) \Rightarrow fire)$ ۲) $(big \wedge dumb) \vee \neg dumb$
۳) $(smoke \Rightarrow fire) \Rightarrow (\neg smoke \Rightarrow \neg fire)$ ۴) $smoke \Rightarrow fire$

۲۶ – کدامیک از جملات منطقی درگزینه‌های زیر معادل «تنها یک دانشجو درس هوش‌مصنوعی را در سال ۱۳۸۱ برداشت» است؟

۱) $\exists x\ student(x) \wedge take(x, AI, 1381) \wedge \forall y\ (student(y) \wedge y \neq x) \Rightarrow \sim take(y, AI, 1381)$
۲) $\sim \forall x\ student(x) \wedge take(x, AI, 1381) \wedge \exists y\ take(y, AI, 1381)$
۳) $\sim \exists x\ student(x) \wedge take(x, AI, 1381) \wedge \exists y\ take(y, AI, 1381)$
۴) $\exists x\ student(x) \wedge take(x, AI, 1381)$

۲۷ – در الگوریتم simulated Annealing درصورتی که دما بالا باشد کدامیک از عبارات زیر صحیح است؟

۱) جستجوی عمومی و ممکن است جستجوی محلی هم انجام شود.
۲) فقط جستجوی محلی انجام می‌شود.
۳) جستجوی عمومی و جستجوی محلی انجام می‌شود.
۴) فقط جستجوی عمومی انجام می‌شود.

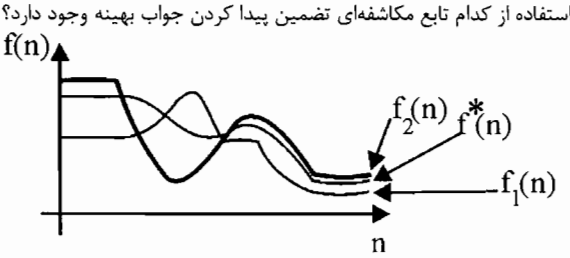
۲۸ – کدامیک از عبارات زیر درمورد Forward chaining و Backward chaining نادرست است؟

۱) BC هم به هدف و هم به حقایق نیاز دارد. ۲) BC به حقایق نیازی ندارد.
۳) سرعت BC بیشتر از FC است. ۴) FC به هدف نیاز ندارد.

۲۹ – کدامیک ازموارد زیر در مورد الگوریتم ژنتیک نادرست است؟

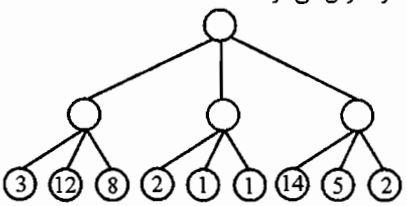
۱) شرط خاتمه در الگوریتم ژنتیک می‌تواند نسل باشد. ۲) یکی از مراحل الگوریتم ژنتیک انتخاب والدین است
۳) یکی ازمراحل الگوریتم ژنتیک انتخاب بازماندگان است. ۴) یکی ازمراحل الگوریتم ژنتیک تولید فرزندان است

۳۰ – در جستجوی A* در صورت استفاده از کدام تابع مکاشفه‌ای تضمین پیدا کردن جواب بهینه وجود دارد؟




۱) $f_2(n)+f_1(n)$
۲) $f_2(n)$
۳) $(f_2(n)+f_1(n))/2$
۴) $f_1(n)$

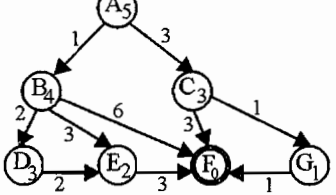
۳۱ – کدام عبارت در مورد جستجوی MIN-MAX و هرس آلفا و بتا غلط است؟
۱) هرس آلفا بتا باعث حذف بعضی از زیرشاخه ها می‌شود.
۲) هرس آلفا بتا باعث افزایش سرعت جستجو می‌شود.
۳) در جستجوی min-max بازیکن رقیب باید بهترین بازی خود را انجام دهد.
۴) در جستجوی max –min فقط بهترین راه‌حل با بیشترین امتیاز برای بازیکن max تولید می‌شود.
۳۲ – کدام عبارت در مورد زنجیر پیشرو و زنجیر عقب‌گرد غلط است؟
۱) در زنجیر پیشرو فقط از حقایق به سمت هدف حرکت باید صورت گیرد.
۲) در زنجیر پیشرو نیازی به مشخص بودن هدف نیست.
۳) در زنجیر عقب‌گرد فقط از هدف به سمت حقایق حرکت باید صورت گیرد.
۴) درزنجیر عقب‌گرد باید حتما هدف مشخص باشد.
۳۳ – کدام عبارت در مورد استنتاج به روش رزولوشن غلط است؟
۱) برای اثبات نتیجه R با داشتن حقیقت F و دانش KB باید ثابت شود که نقیض عبارت $KB \wedge F \Rightarrow R$ همیشه نادرست است.
۲) کلوز یک عبارت شامل عطف یکسری لیترالها است.
۳) در استنتاج به روش رزولوشن عبارت KB و F و 'R باید به CNF تبدیل شوند.
۴) با تولید دو کلوز متمم اثبات با موفقیت به پایان می‌رسد.
۳۴ – کدام عبارت در مورد یک روش جستجوی مناسب غلط است؟
۱) جستجو باید کامل و بهینه باشد.
۲) ضریب انشعاب جستجو باید یک باشد.
۳) پیچیدگی زمانی جستجو باید قابل قبول باشد.
۴) پیچیدگی مکانی جستجو باید قابل قبول باشد.
۳۵ – کدام عبارت در مورد جستجوی $A^*$ غلط است؟
۱) در جستجوی $A^*$ باید $h=h^*$ باشد.
۲) در جستجوی $A^*$ هزینه تا گره جاری و تخمین از گره جاری تا هدف باید مشخص باشد.
۳) جستجوی $A^*$ کامل و بهینه است.
۴) در صورتی $h=h^*$ باشد پیچیدگی زمانی برابر با حاصلضرب عمق جواب در تعداد متوسط شاخه‌ها است.
۳۶ – کدام عبارت برای حل یک مسئله با روش جستجو غلط است؟
۱) هزینه از یک حالت تا حالت بعدی باید مشخص باشد.
۲) حالت هدف باید مشخص باشد.
۳) حالتهای بعدی هر حالت باید مشخص باشند.
۴) حالت شروع باید مشخص باشد.
۳۷ – درکدامیک از گزینه‌های زیر یکسان‌سازی (unification) با شکست روبرو می‌شود؟
۱) group(x,y,mother(x)) , group(Ali,Ali,Alice)
۲) friends(x,y,z) , friends(Ali,Ali,Ali)
۳) knows(y,mother (y)) , knows (Ali,x)
۴) هیچکدام
۳۸ – مطابق با قوانین هرس درخت α-β در درخت مقابل چند گره هرس می‌شوند؟




۱) صفر ۲) ۲ ۳) ۱ ۴) ۳

۳۹ – در جستجوی روی گراف، کدام خصوصیت تابع مکاشفه‌ای (heuristic) h(n) در معادلهٔ f(n)=g(n)+h(n) قابل-قبول‌بودن (admissibility) آن را تضمین می‌کند؟
۱) h(n) همواره از g(n) کوچکتر باشد.
۲) h(n) تابعی یکنواخت (monotonic) باشد.
۳) h(n) درگره‌های متوالی غیرنزولی باشد.
۴) h(n) تابعی غیر صفر و همواره از g(n) بزرگتر باشد.
۴۰ – کدام عبارت صحیح است؟
۱) حافظه مصرفی $A^*$ از تپه‌نوردی کمتر است.
۲) $A^*$ جواب‌های بهتری نسبت به جستجو با هزینه‌یکنواخت می‌یابد.
۳) پیچیدگی فضایی جستجوی دوسویه از جستجوی عرض‌اول کمتر است.
۴) پیچیدگی فضایی جستجویی تعمیق تکراری از جستجوی عمق‌اول بیشتر است.

۴۱ – حاصل جستجوی *SMA با حداکثر ۳ خانه حافظه بر روی گراف مقابل چیست؟ (A گره شروع و F گره هدف است. اعداد روی یال‌ها هزینه مسیر و اعداد داخل گره‌ها هزینه تخمینی گره تا هدف است. ترتیب ملاقات فرزندان به ترتیب حروف الفباست.)




۱) ACF ۲) ABF
۳) ACGF ۴) SMA پاسخی برای این مسئله پیدا نمی‌کند.

۴۲ – در یک درخت بازی اگر ترتیب ملاقات گره‌ها عوض شود:
۱) هیچ تغییری رخ نخواهد داد.
۲) احتمال یافتن جواب بهینه تغییر خواهد کرد.
۳) شاخه هایی که با هرس آلفا-بتا حذف می‌شوند تغییر خواهند کرد.
۴) مسیری که با استفاده از الگوریتم mini max انتخاب می‌شود تغییر خواهد کرد.

۴۳ – اگر بدانیم: - هرکس درس بخواند قبول می‌شود.
- علی قبول شده است.
و نتیجه بگیریم: - علی درس خوانده است.
از چه نوع استنتاجی استفاده کرده‌ایم؟

۱) modus ponen ۲) Abduction
۳) Backward chaining ۴) با هیچ رویه استنتاجی نمی‌توان چنین نتیجه‌ای گرفت.

۴۴ – یک fluent (شناور) در حساب وضعیت‌ها (situation calculus) عبارتست از:
۱) عاملی که حرکت می‌کند.
۲) دنباله‌ای از اعمال که ما را به هدف موردنظر برساند.
۳) متغیری که وضعیت‌های متوالی را در خود نگه می‌دارد.
۴) تابع یا مسندی که از یک وضعیت به وضعیت بعدی تغییر می‌کند.

۴۵ – ترجمه جملهٔ «در کلاس A دانشجویی هست که یکی از نمراتش از همه نمرات دانشجویان کلاس B بیشتر است». به منطق مرتبه اول چیست؟

۱) $\forall w,z\ \exists y,x\ In(x,A)\wedge Grade(x,y)\wedge In(z,B)\wedge Grade(z,w)\wedge GT(y,w)$
۲) $\exists y,x\ \forall w,z\ (In(x,A)\Rightarrow Grade(y,x)) \Rightarrow (In(z,B) \Rightarrow Grade(w,z) \Rightarrow GT(y,w))$
۳) $\exists x\ In(x,A)\wedge \exists y\ Grade(x,y) \wedge (\forall z\ In(z,B) \Rightarrow (\forall w\ Grade(w,z) \Rightarrow GT(y,w)))$
۴) $\exists x,y\ \forall z,w\ In(x,A)\wedge Grade(y,x) \wedge ((In(z,B) \wedge Grade(w,z)) \Rightarrow GT(y,w))$

۴۶ – از دانش‌های زیر برای حل سؤال استفاده کنید. پدر x را با father(x)، مادر x را با mother(x) و همسر x را با partner(x) نشان می‌دهیم. همچنین دو فرض زیر را درنظر بگیرید.

$\forall x,y\ [father(x)=father(y)] \vee [mother(x)=mother(y)] \Rightarrow sib(x,y)$
$\forall x,y\ [father(x)=father(y)] \vee [mother(x)=mother(y)] \Leftrightarrow sib2(x,y)$

کدامیک از عبارات زیر برای نمایش گزاره «هیچ‌کسی در جامعه بیش از یک همسر ندارد» مناسب است؟

۱) $\forall x,y \quad sib(x,y) \Rightarrow [father(x)=father(y)] \wedge [mother(x)=mother(y)]$
۲) $\forall x,y \quad sib2(x,y) \Rightarrow [father(x)=father(y)] \wedge [mother(x)=mother(y)]$
۳) $\forall x,y \quad y=partner(\ partner(x)) \Rightarrow [y=x]$
۴) $\forall x,y \quad y= partner(partner(x)) \Rightarrow [x= partner(partner(x))]$

۴۷ – کدامیک از روشهای جستجوی زیر برای استفاده در یک مسئله ارضای محدودیتها مناسب‌تر می‌باشد؟

۱) تپه‌نوردی ۲) عرض نخست
۳) عمیق‌سازتکراری ۴) عمق نخست

۴۸ – کدامیک از عبارت‌های زیر صحیح‌تر است؟
۱) فرموله‌کردن مسئله همواره باید قبل از فرموله‌کردن هدف انجام گیرد.
۲) فرموله‌کردن هدف همواره باید قبل از فرموله‌کردن مسئله انجام گیرد.
۳) می‌توان فرموله‌کردن هدف را به اختیار قبل وبعد از فرموله‌کردن مسئله انجام داد.

۴) فرموله‌کردن هدف و مسئله اوامر اختیاری هستند، مهم اعمال الگوریتمهای جستجو است و فرموله‌کردن این دو همیشه لازم نیست.

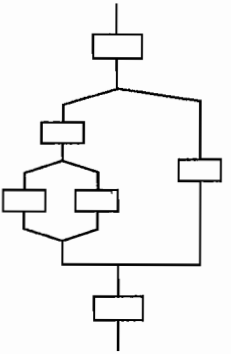
۴۹ – فرض کنید که جملات زیر در منطق مرتبهٔ اول را در اختیار داریم:
این جملات به صورت مناسب تبدیل شده و در پایگاه دانش قرار می‌گیرند. اگر P(A) به این پایگاه‌دانش اضافه شود کدامیک از جملات زیر ایجاب می‌شوند؟

$\forall x\ (p(x) \Rightarrow \exists y\ Q(x,y))$
$\forall x\ Q(x,B)$

۱) Q(G(A),B)
۲) Q(A,B)
۳) Q(A,G(A))
۴) Q(G(A),G(B))

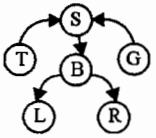
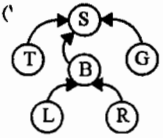
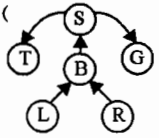
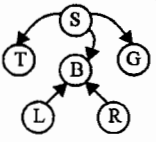
۵۰ – به کدامیک از دلایل زیر استفاده از مدل دریک عامل می‌تواند مفید باشد؟
۱) گسستگی محیط
۲) استفاده از روشهای جستجو
۳) پیوستگی محیط
۴) مشاهده ناپذیر بودن محیط

۵۱ – یک طرح جزئی درشکل مقابل رسم شده است. چند خطی‌سازی از این طرح وجود دارد؟
۱) ۱
۲) ۲
۳) ۴
۴) ۸



۵۲ – کدامیک از مسائل زیر برای استفاده از الگوریتم ارضاءمحدودیتها مناسب نیست؟
۱) جورچین ۸(8 PAZZel)
۲) زمان‌بندی امتحان‌ها
۳) حل جدول کلمات‌متقاطع
۴) چینش ساختارهای منطقی (modules) بر روی یک تراشه

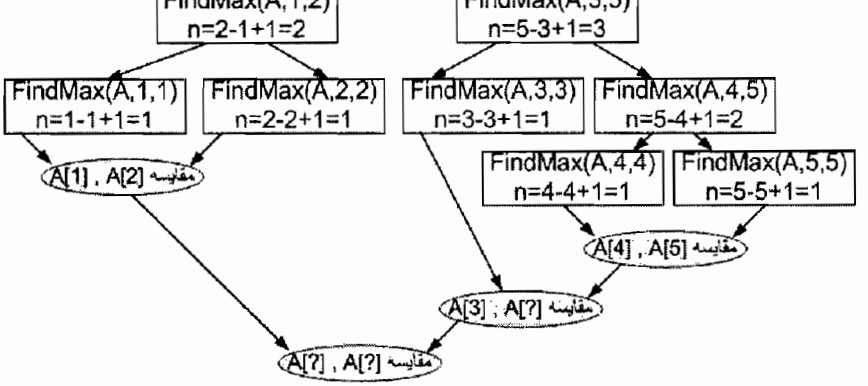
۵۳ – فرض کنید بدانیم: «وقتی ماشین روشن نمی‌شود یعنی یا باطری تمام شده یا ماشین بنزین ندارد یا استارت خراب است. روشن‌ماندن چراغ یا رادیوی ماشین به مدت طولانی منجر به تمام شدن باطری می‌شود.» کدام شبکه‌باور زیر دانش بیان شده در این سؤال را صحیح نمایش می‌دهد؟ (فرض کنید S: بیانگر روشن شدن ماشین، B: کارکردن باطری، G: بنزین داشتن ماشین، L: روشن بودن لامپ در طولانی مدت، R: روشن ماندن طولانی رادیو و T: درست کارکردن استارت است).

۱) 
۲) 
۳) 
۴) 

۵۴ – استدلال با زنجیر عقب‌رو (Backward chaining):
۱) همیشه کامل است
۲) روی definite clause ها کامل است.
۳) اصلا کامل نیست
۴) روی CNF (Conjunctive Normal Form) ها کامل است.

# پاسخ سؤالات کنکور ارشد

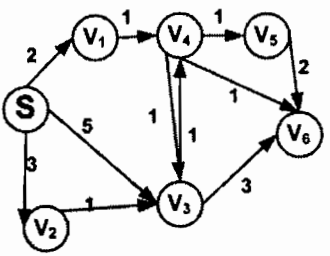
۱ – (گزینه ۱). این رویه برای یافتن ماکزیمم در یک آرایه A بکار می‌رود که i اندیس خانه اول و j اندیس خانه آخر آرایه مورد بررسی است. فرض کنید آرایه A دارای 5 خانه باشد. مراحل زیر بطور بازگشتی رخ می‌دهد. محلهای مقایسه را با رنگ خاکستری نشان داده‌ایم:




برای آرایه‌ای با ۵ خانه، باید ۴ بار مقایسه کنیم. پس n-1 صحیح است.

۲ – (گزینه ۳). فرض کنید گراف زیر را داشته باشیم که تمام وزنها کمتر و مساوی 5 می‌باشد یعنی $C \leq 5$. پس از پایان سؤال باید بگوییم که کوتاهترین مسیر از راس S تا هریک از رئوس دیگر، چقدر است. آرایه‌ای به اندازه تعداد رئوس تشکیل می‌دهیم که هر عدد، نشاندهنده فاصله آن راس از S می‌باشد. اکنون به ترتیب برای هر عنصر آرایه، بایستی:

- یالهای ورودی و خروجی آن گره را یافته و بسته به وزن آن، مقادیر آرایه فوق را آپدیت کنیم.
- اگر برای هر آپدیت مقدار جدید کمتر از قبلی باشد، آنرا یادداشت کنیم وگرنه خیر.




بعنوان مثال برای راس $V_1$ داریم:

| $V_1$ | $V_2$ | $V_3$ | $V_4$ | $V_5$ | $V_6$ |
|---|---|---|---|---|---|
| 2 | 0 | 0 | 3 | 0 | 0 |

بعنوان مثال برای راس $V_2$ داریم:

| $V_1$ | $V_2$ | $V_3$ | $V_4$ | $V_5$ | $V_6$ |
|---|---|---|---|---|---|
| 2 | 3 | 0 | 3 | 0 | 0 |

بعنوان مثال برای راس $V_3$ داریم:

| $V_1$ | $V_2$ | $V_3$ | $V_4$ | $V_5$ | $V_6$ |
|---|---|---|---|---|---|
| 2 | 3 | 4 | 3 | 0 | 7 |

بعنوان مثال برای راس $V_4$ داریم:

| $V_1$ | $V_2$ | $V_3$ | $V_4$ | $V_5$ | $V_6$ |
|---|---|---|---|---|---|
| 2 | 3 | 4 | 3 | 4 | 4 |

یک بار پیمایش بر روی آرایه رئوس و بار دیگر پیمایش بر روی آرایه یالها، پس O(|V|+|E|).

۳ – (گزینه ۱). در الگوریتم فلوید وزن یالهای گراف جهتدار درون ماتریس weight و فاصله رئوس با یکدیگر درون ماتریس distance. شبه کد زیر نحوه پیاده‌سازی الگوریتم فلوید را نشان می‌دهد:

```
Procedure Floyd
Begin
  Copy weight matrix to distance matrix
  N = number of nodes
  For k = 0 to N
    For i = 0 To N
      For j = 0 To N
        distance[i,j]=Min(distance[i,j],distance[i,k]+distance[k,j])
      End For
    End For
  End For
End
```

در الگوریتم فلوید-وارشال می‌توان وجود یال منفی و دور منفی را تشخیص داد. دقت کنید که در صورتیکه گراف شما دارای دور منفی است، می‌توانید چندین بار کار خود را تکرار کنید. درآنصورت اعداد ماتریس به سمت منفی بینهایت میل می‌کنند. البته در هیچ صورتی نمی‌توان برای این گرافها مسئله را حل نمود و در بهترین حالت فقط می‌توانیم این موضوع را تشخیص دهیم.

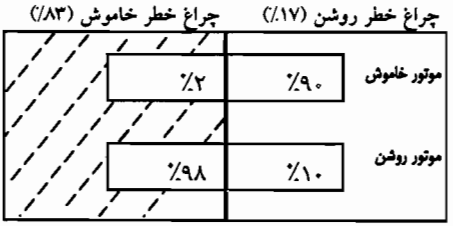
۴ – (گزینه ۲). می‌توان عنصر اول، وسط و آخر آرایه را به ترتیب با نام‌های a، b و c در نظر گرفت. اگر a<b و b>c عنصر مینیمم در نیمه دوم آرایه قرار دارد. اگر a>b و b<c عنصر مینیمم در نیمه اول آرایه قرار دارد. مجددا همین کار را برای آن نیمه انجام می‌دهیم. بنابراین در هر بار نصف آرایه انتخاب می‌شود پس O(log n).

۵ – (گزینه ۴). این گراف بدون وزن است پس گره‌هایی که در سطوح بالاتری از گراف بدون وزن باشند، به راس S نزدیکترند. با انجام پیمایش اول‌سطح، می‌توان به ترتیب رئوس نزدیک تا دور را یافت.

۶ – (گزینه ۲). به تعریف این الگوریتم‌ها در منبع درسی مراجعه کنید تا درستی و نادرستی جملات مشخص شود.

۷ – (گزینه ۴)

۸ – (گزینه ۲)




ابتدا باید احتمال روشن شدن/خاموش شدن چراغ خطر را محاسبه کنیم. از آنجا که هیچ دلیلی به جز اشیای سنگین وجود ندارد پس داریم:

احتمال(روشن شدن چراغ خطر) =

{احتمال(سنگینی شی و روشن شدن خطر)×(سنگینی شی)}+{ احتمال(سبکی شی و روشن شدن خطر)×(سبکی شی)}=

{٪۱۰×٪۸۰} +{٪۹۰×٪۱۰}=٪۱۷

احتمال(خاموش شدن چراغ خطر) =٪۱۰۰-٪۱۷=٪۸۳ (که این اعداد را بر روی شکل مشاهده می‌کنید)

حال طبق شکل فوق بایستی احتمال منطقه خاکستری به منطقه هاشورخورده را بیابیم:

احتمال(موتورخاموش و خطرخاموش)= {احتمال(خطرخاموش|موتورخاموش)×احتمال(خطرخاموش)}=٪۲×٪۸۳=٪۱,۶۶

۹ – (گزینه ۴). به پاسخ سؤال ۵ مراجعه کنید. اکنون فرض کنید رئوس را به ترتیب از A تا I نامگذاری کرده‌ایم:

| | شروع | مشاهده x | مشاهده y | مشاهده 3* | مشاهده 7* | مشاهده 11* | مشاهده 4 | مشاهده 6* |
|---|---|---|---|---|---|---|---|---|
| A | a=-∞ | a=-∞ | a=-∞ | a=-∞ | a=7 | a=7 | a=7 | a=7 |
| B | β=+∞ | β=+∞ | β=max{x,y} | β=max{x,y} | β=7 | β=7 | β=7 | β=7 |
| C | β=+∞ | β=+∞ | β=+∞ | β=+∞ | β=+∞ | β=+∞ | β=4 | β=4 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| D | a=-∞ | a=x | a=max{x,y} | a=max{x,y} | a=max{x,y} | a=max{x,y} | a=max{x,y} | a=max{x,y} |
| E | a=-∞ | a=-∞ | a=-∞ | a=3 | a=7 | a=7 | a=7 | a=7 |
| F | a=-∞ | a=-∞ | a=-∞ | a=-∞ | a=-∞ | a=11 | a=11 | a=11 |
| G | a=-∞ | a=-∞ | a=-∞ | a=-∞ | a=-∞ | a=-∞ | a=4 | a=6 |
| H | a=-∞ | a=-∞ | a=-∞ | a=-∞ | a=-∞ | a=-∞ | a=-∞ | a=-∞ |
| I | a=-∞ | a=-∞ | a=-∞ | a=-∞ | a=-∞ | a=-∞ | a=-∞ | a=-∞ |

ستونهایی که ستاره دارند حاوی نکته‌هایی هستند که در رسیدن به جواب مهم است:
- پس از مشاهده ۳: اگر $\max\{x,y\}\leq 3$ آنگاه باید شاخه ۷ هرس می‌شد. بعلت عدم هرس باید داشته باشیم: $\max\{x,y\}>3$
- پس از مشاهده ۷: اگر $\max\{x,y\}\leq 7$ آنگاه باید شاخه F هرس می‌شد. بعلت عدم هرس باید داشته باشیم: $\max\{x,y\}>7$
- پس از مشاهده ۱۱: اگر $\max\{x,y\}\leq 11$ آنگاه باید شاخه ۱۲ هرس می‌شد. بعلت وجود هرس باید داشته باشیم: $\max\{x,y\}\leq 11$.
- پس از مشاهده ۶: اگر $a(A)=7\leq\beta(C)=6$ آنگاه باید سایر شاخه‌ها هرس شوند.

پس $\max\{x,y\}>7\geq 11$ که طبق گزینه‌ها تنها مورد آخر برقرار است.

۱۰ – (گزینه ۱). این روش دقیقا همانند جستجوی حریصانه است با این تفاوت که فقط k تا از بهترین گره‌های مجاور برای بسط انتخاب شده و با بقیه کاری نداریم. اگر k=1 باشد یعنی در هر مرحله فقط بهترین گره مجاور برای مرحله بعد انتخاب می‌شود که همان تپه‌نوردی است.

۱۱ – (گزینه ۴)

۱۲ – (گزینه ۴)

۱۳ – (گزینه ۱). روشهای شبیه‌سازی حرارت، ژنتیک و پرتومحلی با احتمال زیادی از بهینه محلی فرار می‌کنند ولی اینگونه نیست که حتمی باشد. ولی روش تپه‌نوردی در صورتی که نزدیک بهینه‌محلی کار خود را شروع کند، حتما در آن گیر می‌افتد.

۱۴ – (گزینه ۳). طبق مباحث منبع درسی، سرعت روش زنجیره‌ای عقبگرد بیشتر از زنجیره‌ای جلورو است.

۱۵ – (گزینه ۱). روش برش آلفا-بتا برای هرس شاخه‌هایی از درخت بازی طراحی شده است که در طی بازی امکان انتخاب آنها وجود ندارد. ژنتیک یک جستجوی محلی، A* نوعی جستجوی اول بهترین و MiniMax برای جستجوی حرکتی که بازیکن بیشترین منفعت را ببرد مورد استفاده قرار می‌گیرد.

۱۶ – (گزینه ۴). طبق تعریف رزولوشن.

۱۷ – (گزینه ۲). در الگوریتم ژنتیک، انتخاب والدین جهت تولید نسل، بر حسب شایستگی آنهاست.

۱۸ – (گزینه ۳). E(x,z) نشاندهنده آن است که x به z اعتماد دارد. B(x,y) نشاندهنده آن است که x به y بدبین است. طبق بخش اول در سؤال هرکسی که به همه بدبین است. پس باید یک سور عمومی برای «هرکس» و یک سور عمومی برای «همه» درنظر بگیریم که در بخش مقدم گزینه‌های ۳ و ۴ بیان شده است. بخش تالی گزینه اول و چهارم بصورت «فرد غیرقابل-اعتمادی وجود ندارد یعنی می‌توان به همه اعتماد نمود» و گزینه دوم و سوم بصورت «هیچ فرد قابل‌اعتمادی وجود ندارد یعنی نمی‌توان به هیچکس اعتماد کرد» می‌باشد.

۱۹ – (گزینه ۲). اگر از نقطه (0,0) شروع به حرکت کنیم، در هر بار چهار خانه برای حرکت بعدی قابل انتخاب است. و باز با رفتن به هر خانه بعدی مجددا می‌توان ۴ انتخاب دیگر داشته باشیم. پس فاکتور انشعاب در درخت جستجو b=4 خواهد بود. و برای رسیدن از خانه (0,0) به (x,y) در این روش باید به عمق d=x+y از درخت برسیم. اگر فرض کنیم جستجو با الگوریتم اول‌سطح (اول پهنا) انجام شود، طبق مباحث عنوان شده در کتاب اصلی، تعداد گره‌های تولیدی عبارتست از:

$$b+b^2+b^3+\ldots+(b^{d+1}-b)=\frac{b^{d+1}-1}{b-1}-1$$

که طبق مقادیر b=4 و d=x+y عبارتست از:

$$\frac{4^{x+y+1}-1}{3}-1$$

۲۰ – (گزینه ۴)

۲۱ – (گزینه ۳)

طبق جدول داده شده مقدار تابع اکتشافی، عبارتست از مقدار هر گره تا E. یعنی:

$h(B^*)=2\ ,\ h(C^*)=6\ ,\ h(D^*)=9$

۴۱ – (گزینه ۱). مسیرهای رسیدن به هدف با سه گره را در گراف یافته و مابقی گراف را حذف می‌کنیم. از بین مسیرهای موجود، کوتاهترین را می‌یابیم.

۴۲ – (گزینه ۳)

۴۳ – (گزینه ۲). استنتاج abduction نوعی حدس زدن است و برای استنتاج در مورد بهترین علل رخ دادن یک پدیده بکار می‌رود. بعبارت دیگر از روی تعدادی مشاهده، بهترین فرضیه را استنتاج می‌کنیم.

مشاهده O برقرار است. (دراینجا: علی قبول شده است)

فرضیه H، مشاهده O را تبیین می‌کند. (دراینجا: هرکس درس بخواند قبول می‌شود)

فرضیه H بهترین فرضیه از میان سایر فرضیه های‌دیگر است.

آنگاه طبق استنتاج abduction باید H صادق باشد. (در اینجا: علی درس خوانده است)

ولی برای استنتاج modus ponen داریم: $\frac{P \Rightarrow Q \ , \ P}{\therefore Q}$

۴۴ – (گزینه ۴)

۴۵ – (گزینه ۴)

۴۶ – (گزینه ۳). برای اثبات نادرستی گزینه ۴ فرض کنید علی دارای دو همسر با نامهای زهرا و مریم باشد. پس در بخش مقدم، عبارت partner(partner(x=ali)) می‌تواند زهرا یا مریم باشد. و دلیلی ندارد که این عبارت در بخش تالی برابر علی شود. برای گزینه اول و دوم اگر بخش مقدم نادرست باشد یعنی xوy برادر نباشند، نمی‌توان جمله خواسته شده را اثبات نمود. گزینه سوم، با گذاشتن عبارت y=x رابطه را ملزم می‌کند که زهرا و مریم باید یک نفر باشند.

۴۷ – (گزینه ۴). در مسائل ارضای‌محدودیت، باید به تمام متغیرهای مسئله مقدار داده و سپس چک کنید که هیچیک از قوانین محدودیت نقض نشده باشند. در روش اول‌عمق، به یک متغیر (مثلا نام شهر) مقدار داده (مثلا آبی) و با توجه به آن به سایر متغیرها مقدار داده و در درخت پایین می‌رویم تا به تناقض (همرنگی دو شهر مجاور) یا پایان درخت برسیم. پس این روش بهتر است زیرا در مسئله‌ای نظیر رنگ‌آمیزی، با داشتن مقدار رنگ یک شهر، نمی‌توان به تناقض پی برد. روش عرض‌نخست، به یک متغیر (شهر) مقادیر مختلف انتساب داده و هر بار تناقض را چک می‌کند. پس اول عمق بهتر است.

۴۸ – (گزینه ۲)

۴۹ – (گزینه ۳). با استفاده از نمونه‌سازی عمومی عبارت اول (برای عنصر A) داریم: $P(A) \Rightarrow \exists y \ Q(A,y)$

طبق صورت مسئله می‌دانیم که P(A) در پایگاه برقرار است. پس طبق قاعده modus ponen از ترکیب P(A) با عبارت فوق، داریم: $\exists y \ Q(A,y)$

اکنون می‌خواهیم نمونه‌سازی وجودی عبارت فوق را انجام دهیم ولی باید دقت داشته باشیم که در صورت سؤال، عبارت دوم شامل Q(A,B) هست. پس در اینجا باید از اسکولم استفاده نماییم. پس داریم: $Q(A,G(A))$

که G(A) اسکولمی است که به جای y قرار گرفت.

۵۰ – (گزینه ۴). برداشت کارگزار از شرایط فعلی محیط، مبنای استنتاج وی در مرحله بعدی واکنش خواهد شد. بنابراین اگر محیطی مشاهده‌ناپذیر داشته باشیم کارگزار مبنایی برای تصمیم‌گیری ندارد. پس بهتر است تا برای وی مدلی تعریف شود تا آنرا مبنای کار خود قرار دهد.

۵۱ – (گزینه ۴)

۵۲ – (گزینه ۱). اگر هر خانه پازل را یک متغیر درنظربگیریم، بیان محدودیتها دشوار است. در این پازل رسیدن به چیدمان صحیح مهم است که روش ارضای‌محدودیت هیچ کمکی نمی‌کند زیرا فقط می‌تواند مقدار هر متغیر را عوض کرده و محدودیتها را چک کند. تعویض مقدار متغیرها بایستی با توجه به خانه‌های مجاور پازل انجام شود ولی در مسائل ارضای محدودیت برای تغییر مقدار هر متغیر بطور تصادفی مقداری از دامنه آنرا انتخاب می‌کنیم.

۵۳ – (گزینه ۲). با توجه به جمله «روشن ماندن زیاد رادیو و لامپ، منجر به تمام شدن باطری می‌شود»، باید دو حالت L و R به سمت B فلش داشته باشند. پس گزینه ۱ غلط است. همچنین «نبود باطری، استارت و بنزین منجر به روشن نشدن ماشین می‌شود» پس باز هم باید به سمت S فلش داشته باشیم.

۵۴ – (گزینه ۲)