

## سؤالات کنکور کارشناسی ارشد سال ۹۷

۱- در مورد الگوریتم برنامه‌ریزی گراف پلن (*Graphplan*)، کدام عبارت درست است؟

(مهندسی کامپیوتر - دولتی ۹۷)

(۱) اگر در سطح  $k$  ام از گراف برنامه‌ریزی، دو گزاره با یکدیگر ناسازگاری متقابل (*mutual exclusion*) داشته باشند، نمی‌توان با انجام  $k$  کنش از وضعیت اولیه به وضعیتی رسید که شامل هر دو گزاره باشد.

(۲) در هنگام ساخت گراف برنامه‌ریزی (*Planning Graph*)، در صورتی که یک سطح دقیقاً مشابه سطح قبلی باشد، ولی هنوز برنامه‌ای به دست نیامده باشد، الگوریتم متوقف می‌شود.  
(۳) در برنامه یافت شده توسط این الگوریتم، تمام کنش‌های انتخاب شده از هر سطح، لزوماً باید قبل از تمام کنش‌های انتخاب شده از سطح بعد قرار گیرند تا برنامه حاصل معتبر باشد.  
(۴) برنامه یافت شده توسط این الگوریتم از نظر تعداد کنش‌ها بهینه است.

۲- پایگاه دانش متشکل از عبارت‌های *Hom* زیر را در نظر بگیرید. فرض کنید روش زنجیره‌ای جلورو (*Forward Chaining*) با استفاده از جستجوی سطح اول (*Breadth First Search*) و روش زنجیره‌ای عقب رو (*Backward Chaining*) با استفاده از جستجوی عمق اول (*Depth First Search*) پیاده‌سازی شده است. در این صورت، کدام جمله نادرست است؟

(مهندسی کامپیوتر - دولتی ۹۷)

$$P(F(x)) \wedge Q(x) \Rightarrow Q(F(x))$$

$$P(x) \Rightarrow P(F(x))$$

$$P(A)$$

$$Q(A)$$

(۱) گزاره  $Q(F(F(A)))$  را نتیجه می‌دهد.

(۲)  $FC$  قبل از گزاره  $Q(F(A))$ ، گزاره  $P(F(A))$  را نتیجه می‌دهد.

(۳)  $BC$  برای پرسمانی (*query*) به صورت  $Q(F(A))$ ، مقدار درست (*True*) را برمی‌گرداند.

(۴)  $BC$  برای یک پرسمانی (*query*) به صورت  $Q(F(F(A)))$ ، مقدار نادرست (*False*) را برمی‌گرداند.



۷- در صورتی که بخواهیم با استفاده از روش رزولوشن (*Resolution*) نوع عبارت گزاره‌ای زیر را تعیین کنیم، کدام مورد در خصوص نوع این عبارت درست است؟ (مهندسی کامپیوتر - دولتی ۹۷)

$$(P \Rightarrow (Q \Rightarrow R)) \Rightarrow ((P \Rightarrow Q) \Rightarrow (P \Rightarrow R))$$

(۱) غیرقابل ارضاء (*Unsatisfiable*) است.

(۲) ارضای پذیر (*Satisfiable*) است.

(۳) نامعتبر (*Invalid*) است.

(۴) معتبر (*Valid*) است.

۸- یک درخت جستجوی *minimax* را در نظر بگیرید که دارای عمق ۳ باشد و هر گره در آن دقیقاً ۴ فرزند داشته باشد (درخت ۶۴ گره برگ دارد). اگر از روش هرس آلفا - بتا در جستجو استفاده کنیم، حداکثر چه تعداد از گره‌های این درخت ممکن است هرس شود؟ (مهندسی کامپیوتر - دولتی ۹۷)

۳۲ (۴)	۳۵ (۳)	۴۵ (۲)	۴۸ (۱)
--------	--------	--------	--------

۹- یک مسئله ارضای محدودیت با چهار متغیر  $A, B, C, D$  را در نظر بگیرید که در آن دامنه تمام متغیرها مجموعه  $\{1, 2, 3, 4\}$  است. این مسئله دارای محدودیت‌های  $A < B < C < D$  و  $C = A + 3$  است. می‌دانیم برای عدد طبیعی  $k$ ، با حذف برخی مقادیر از دامنه متغیرها، ممکن است بتوان یک مسئله ارضای محدودیت را به یک مسئله *strongly k-consistent* تبدیل کرد. حال با فرض حذف مقادیر لازم از دامنه متغیرها، کدام مورد در خصوص مسئله فوق درست است؟ (مهندسی کامپیوتر - دولتی ۹۷)

(۱) این مسئله را می‌توان به یک مسئله *strongly 2-consistent* تبدیل کرد، ولی نمی‌توان آن را به یک مسئله *strongly 3-consistent* تبدیل کرد.

(۲) این مسئله را می‌توان به یک مسئله *strongly 3-consistent* تبدیل کرد، ولی نمی‌توان آن را به یک مسئله *strongly 4-consistent* تبدیل کرد.

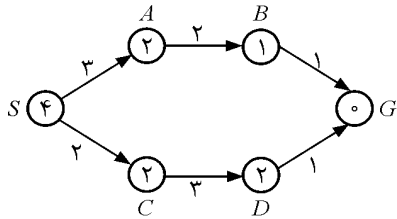
(۳) این مسئله را نمی‌توان به یک مسئله *strongly 2-consistent* تبدیل کرد.

(۴) این مسئله را می‌توان به یک مسئله *strongly 4-consistent* تبدیل کرد.

۱۰- در خصوص الگوریتم  $A^*$  در حالت استفاده از یک تابع ابتکاری سازگار (*consistent*)  $h$  (در صورتی که  $g(n)$  هزینه مسیر طی شده تا گره  $n$  باشد)، کدام مورد نادرست است؟ (مهندسی کامپیوتر - دولتی ۹۷)

(۱) همواره مسیر بهینه به هدف را (در حالت جستجوی گرافی) پیدا می‌کند.

- (۲) ممکن است گره‌هایی را که مقدار  $g(n) + h(n)$  آن‌ها بیشتر از طول مسیر بهینه است، گسترش دهد.
- (۳) ممکن است گره‌هایی را که مقدار  $g(n)$  آن‌ها بیشتر از طول مسیر بهینه است، تولید کند (یعنی در صف بگذارد).
- (۴) ممکن است گره‌هایی را که مقدار  $g(n) + h(n)$  آن‌ها بیشتر از طول مسیر بهینه است، تولید کند (یعنی در صف بگذارد).



۱۱- در شکل زیر هزینه کنش‌ها روی یال‌ها و مقدار تابع ابتکاری روی گره‌ها نوشته شده است. اگر  $S$  گره شروع و  $G$  گره هدف باشد، کدام مورد درست است؟ (در شرایط یکسان برای دو گره از ترتیب الفبایی استفاده شود) (مهندسی کامپیوتر - دولتی ۹۷)

- (۱) تابع ابتکاری استفاده شده قابل قبول (*admissible*) است.
- (۲) ترتیب گسترش گره‌ها در الگوریتم  $A^*$ ، از چپ به راست  $S, C, A, B, G$  است.
- (۳) ترتیب تولید گره‌ها در الگوریتم  $A^*$ ، از چپ به راست  $S, A, C, B, D, G$  است.
- (۴) ترتیب تولید گره‌ها در الگوریتم  $UCS$ ، از چپ به راست  $S, A, C, B, D, G$  است.

۱۲- فرض کنید در یک مسئله جستجو، فضای جستجو یک درخت محدود باشد که در آن هزینه هر یال یک عدد گویا است (هزینه‌ها می‌توانند منفی باشند). کدام عبارت در مورد یافتن مسیر بهینه توسط سه روش *Breadth First Search*، *Depth First Search* و *Uniform Cost Search* درست است؟ (مهندسی کامپیوتر - دولتی ۹۷)

- (۱) هر سه روش، یافتن مسیر بهینه را برای مسئله گفته شده تضمین می‌کنند.
- (۲) فقط دو روش، یافتن مسیر بهینه را برای مسئله گفته شده تضمین می‌کنند.
- (۳) فقط یکی از این سه روش، یافتن مسیر بهینه را برای مسئله گفته شده تضمین می‌کند.
- (۴) هیچکدام از این سه روش، یافتن مسیر بهینه را برای مسئله گفته شده تضمین نمی‌کنند.

۱۳- کدام عبارت در مورد خصوصیات محیط (*environment*)، درست است؟

(مهندسی کامپیوتر - دولتی ۹۷)

- (۱) یک محیط پویا (*dynamic*) نمی‌تواند کاملاً مشاهده‌پذیر (*fully observable*) باشد.
- (۲) هر محیط کاملاً مشاهده‌پذیر (*fully observable*) حتماً قطعی (*deterministic*) است.

۳) یک محیط ناشناخته (*unknown*) ممکن است کاملاً مشاهده‌پذیر (*fully observable*) باشد.

۴) در یک محیط *episodic*، هر کنش (*action*) ممکن است به کنش‌های انجام شده در مرحله قبل وابسته باشد.

۱۴- وظیفه بخش مولد مسئله (*Problem generator*) در عامل‌های یادگیر، کدام است؟

(مهندسی IT - دولتی ۹۷)

۱) دریافت تعریف مسئله از ورودی

۲) فرموله کردن اطلاعات مسئله، به شکل مفید برای عامل

۳) ایجاد امکان کاوش در محیط، برای کسب اطلاعات بیشتر

۴) به روز کردن پایگاه دانش با توجه به اطلاعات به دست آمده از محیط

۱۵- کدام مورد در خصوص مسائل برنامه‌ریزی (*Planning*)، درست است؟ (مهندسی IT - دولتی ۹۷)

۱) برای حل هر مسئله برنامه‌ریزی، می‌توان نسخه ساده شده‌ای از آن مسئله را با در نظر نگرفتن لیست حذف (*Delete List*)، کنش‌ها به سادگی (و در زمان چندجمله‌ای) حل کرد و از تعداد گام‌های برنامه بهینه حاصل به عنوان یک تابع ابتکاری برای حل مسئله اصلی استفاده کرد.

۲) گراف برنامه‌ریزی (*Planning Graph*)، ساختاری است که هم می‌تواند در راستای پیدا کردن تابع ابتکاری برای مسئله برنامه‌ریزی و هم در جهت پیدا کردن گام‌های برنامه مورد استفاده قرار گیرد.

۳) الگوریتم *POP* با شروع از وضعیت اولیه مسئله و به صورت جلورو در هر دور، گزاره‌های هدفی که پیدا نشده‌اند را در یک صف نگه می‌دارد تا اینکه نهایتاً آن صف خالی شود.

۴) در هر وضعیت، شمردن تعداد گزاره‌های هدف تولید نشده یک تابع ابتکاری، قابل قبول (*admissible*) است.

۱۶- کدام یک از عبارات زیر درست است؟ (مهندسی IT - دولتی ۹۷)

۱) برای استنتاج در منطق مرتبه اول، روش زنجیره‌ای جلورو (*forward chaining*)، برای پایگاه‌های دانش به شکل عبارات معین (*definite clauses*)، یک روش استنتاج کامل (*complete*) است.

۲) برای استنتاج در منطق گزاره‌ای، روش زنجیره‌ای جلورو (*forward chaining*)، یک روش

کامل (complete) است.

۳) برای استنتاج در منطق گزاره‌ای، روش رزولوشن (Resolution) یک روش کامل (complete) نیست.

۴) استنتاج در منطق مرتبه اول، یک مسئله تصمیم‌پذیر (decidable) نیست.

۱۷- سه عبارت  $\alpha$ ،  $\beta$  و  $\gamma$  را در منطق گزاره‌ای (propositional logic) در نظر بگیرید. در صورتی که عبارت  $\gamma$  حاصل اعمال قاعده رزولوشن (resolution) بر عبارت‌های  $\alpha$  و  $\beta$  باشد، کدام مورد درست است؟ (مهندسی IT - دولتی ۹۷)

$$(1) \neg\alpha \wedge \neg\beta \models \gamma \quad (2) \alpha \wedge \beta \models \gamma \quad (3) \alpha \vee \beta \models \gamma \quad (4) \alpha \Rightarrow \beta \models \gamma$$

۱۸- برای سه جمله داده شده  $\{(A \wedge B) \vee (B \wedge C), (D \vee C), (A \leftrightarrow B \leftrightarrow C)\}$  و با در نظر گرفتن مقادیر ممکن برای چهار متغیر گزاره‌ای  $\{A, B, C, D\}$ ، (به ترتیب از چپ به راست) چند مدل (Model) وجود دارد؟ (مهندسی IT - دولتی ۹۷)

$$(1) \{6, 3, 8\} \quad (2) \{6, 12, 4\} \quad (3) \{12, 3, 8\} \quad (4) \{8, 12, 4\}$$

۱۹- کدام مورد، نا درست است؟ (مهندسی IT - دولتی ۹۷)

- ۱) اگر فرض ثابت بودن مجموع امتیازات دو بازیکن در انتهای بازی برقرار نباشد، الگوریتم *Minimax* ممکن است حرکت‌های غیربهبینه‌ای را به بازیکن پیشنهاد کند.
- ۲) در الگوریتم *Minimax*، گره‌های پایانی بسته به اینکه آخرین حرکت مربوط به کدام بازیکن باشد، امتیاز مربوط به آن بازیکن را برمی‌گردانند.
- ۳) الگوریتم هرس آلفا - بتا هیچ تأثیری بر نتیجه نهایی الگوریتم *Minimax* ندارد و فقط می‌تواند زمان این الگوریتم را بهبود دهد.
- ۴) در الگوریتم *Minimax*، هر بازیکن به دنبال حداکثر کردن امتیاز خودش است.

۲۰- فرض کنید  $h$  یک تابع مکاشفه‌ای (heuristic function) است و تابع  $h^*$  مقدار بهینه هزینه رسیدن به هدف را از هر وضعیت در فضای وضعیت‌ها می‌دهد. همچنین فرض کنید برای یکی از وضعیت‌ها به نام  $s$  که روی یک مسیر بهینه از وضعیت اولیه به یک وضعیت هدف قرار دارد  $h(s) = 2 \times h^*(s)$  و برای سایر وضعیت‌ها مقدار تابع  $h$  حداکثر برابر مقدار تابع  $h^*$  است. کدام مورد در خصوص الگوریتم  $A^*$  که از تابع مکاشفه‌ای  $h$  و روش جستجوی درختی (tree Search) استفاده می‌کند، درست است؟ (مهندسی IT - دولتی ۹۷)

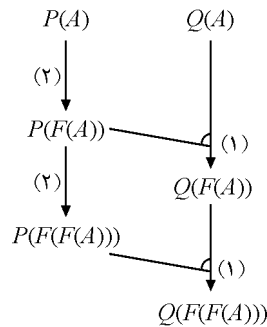
- 
- 
- (۱) چنانچه در فضای جستجو تنها یک مسیر بهینه به وضعیت هدف وجود داشته باشد، این الگوریتم آن مسیر را می یابد.
- (۲) از آنجایی که  $h$  یک تابع قابل قبول (*admissible*) نیست، این الگوریتم هیچگاه مسیر بهینه را نمی یابد.
- (۳) هزینه مسیر یافت شده توسط این الگوریتم حداکثر دو برابر هزینه مسیر بهینه است.
- (۴) وضعیت  $s$  با اجرای مسیر یافت شده توسط الگوریتم ملاقات نخواهد شد.
-

۱- گزینه (۱) صحیح است.

در الگوریتم *Graph Plan*، ناسازگاری‌های متقابل بین گزاره‌ها، از یک سطح به سطح دیگر بطور یکنواخت کاهش می‌یابند. به عبارت دیگر، اگر دو گزاره در سطح  $k$  دارای ناسازگاری متقابل باشند حتماً در سطح‌های قبلی هم با هم ناسازگار بوده‌اند. و این به این معنی است که تمام کنش‌هایی (*action* هایی) که از سطح صفر تا سطح  $k$  پیش شرط‌هایشان برقرار بوده‌اند و به گراف اضافه شده‌اند نتوانسته‌اند ما را به وضعیتی برسانند که آن دو گزاره موردنظر ناسازگاری متقابل نداشته باشند.

۲- گزینه (۴) صحیح است.

دو قانون پایگاه دانش را به ترتیب با شماره‌های ۱ و ۲ نشان می‌دهیم. مراحل زنجیره‌سازی به جلو به صورت زیر است:



دیده می‌شود که گزاره‌های (۱) و (۲) عبارات صحیحی هستند.

الگوریتم *BC* را می‌توان از روی شکل فوق و با دنبال کردن مسیر عکس فلش‌ها (از جمله پرسیمان تا رسیدن به جملات ساده و موجود در *KB*) بررسی کرد. دیده می‌شود که هر دو جمله  $Q(F(A))$  و  $Q(F(F(A)))$  توسط الگوریتم *BC* قابل استنتاج‌اند. پس گزینه (۳) عبارت درستی است اما گزینه (۴) عبارت نادرستی است.

۳- گزینه (۱) صحیح است.

ابتدا عبارت داده شده را کمی ساده می‌کنیم:



(الف) سورهای وجودی  $\exists y \exists z$  در ابتدای دو عبارت را می توان در کنار  $P(x,y,z)$  بعد از آنها نوشت چون متغیرهای  $z$  و  $y$  فقط در این عبارت بکار رفته اند (متغیرهای  $y$  و  $z$  در جاهای دیگر مربوط به سورهای دیگرند). همچنین برای سادگی نوشتاری، عبارت  $\exists y \exists z P(x,y,z)$  را با  $R(x)$  نشان می دهیم.

(ب) عبارت  $(\neg \exists z \exists u (\neg P(x,z,u)))$  معادل عبارت زیر است:

$$\forall z \forall u P(x,z,u)$$

برای سادگی بیشتر، این عبارت را با  $Q(x)$  نشان می دهیم.

(ج) دقت کنید که اگر  $Q(x)$  برقرار باشد حتماً  $R(x)$  هم برقرار است چون  $Q(x)$  یک سور عمومی روی  $P(x,z,u)$  است و  $R(x)$  یک سور وجودی روی همان عبارت است، اگر سور عمومی روی  $P$  برقرار باشد سور وجودی هم برقرار است (البته با فرض اینکه دنیا شامل حداقل یک شیء باشد). پس  $R(x) \vee Q(x)$  معادل با  $R(x)$  است.

(د)  $f(a)$  یک تابع اسکولم با مقدار ثابت  $a$  بعنوان پارامتر است. این تابع قطعاً به یک شیء ثابت اشاره می کند. آن شیء را  $C$  فرض می کنیم و به جای  $f(a)$  می نویسیم.

با توجه به توضیحات فوق عبارات (i) و (ii) به صورت زیر در می آیند:

$$(i) (\forall x R(x)) \Rightarrow R(C)$$

$$(ii) \forall x [R(x) \Rightarrow R(C)]$$

جمله (i) می گوید اگر  $R(x)$  به ازای همه اشیاء برقرار باشد به ازای شیء  $C$  هم برقرار است، که جمله ای بدیهی و صحیح است.

اما جمله (ii) می گوید به ازای هر شیء دلخواه  $x$  اگر  $R(x)$  برقرار باشد آنگاه  $R(C)$  هم برقرار است که جمله نادرستی است.

۴- گزینه (۳) صحیح است.

اطلاعات زیر از صورت سؤال بدست می آید:

(الف) چون  $\alpha \models \beta$  است پس  $\alpha \Rightarrow \beta$  عبارت درستی است (ارزش آن همیشه True است)

(ب) چون  $\alpha \models \beta$  است پس  $M_\alpha \subseteq M_\beta$

(ج) چون  $\alpha \not\models \gamma$  پس  $M_\alpha \not\subseteq M_\beta$  نیست، پس  $M_\alpha$  تهی نیست (چون اگر تهی باشد مجموعه تهی

زیر مجموعه هر مجموعه ای از جمله  $M_\gamma$  است و بنابراین می بایست  $\alpha \models \gamma$  می بود!).

(د) با توجه به (ب) و (ج)، چون  $M_\alpha$  تهی نیست و  $M_\alpha \subseteq M_\beta$  است پس  $M_\beta$  هم تهی نیست.

حال گزینه ها را بررسی می کنیم:

گزینه (۱) نادرست است: می‌دانیم که  $M_{\beta \vee \gamma} = M_{\beta} \cup M_{\gamma}$  و چون  $M_{\beta}$  تهی نیست پس  $M_{\beta \vee \gamma}$  هم تهی نیست، بنابراین  $\beta \vee \gamma$  حتماً ارضا پذیر است.

گزینه (۲) نادرست است: طبق نتیجه (الف)  $\alpha \Rightarrow \beta$  یک عبارت همیشه درست است، پس ارضا پذیر هم هست.

گزینه (۳) درست است: طبق نتیجه (ج)  $M_{\alpha}$  تهی نیست، بنابراین  $\alpha$  حتماً ارضا پذیر است.

گزینه (۴) نادرست است: با توجه به اینکه  $\alpha \Rightarrow \beta$  عبارت همیشه درستی است می‌توان دید که عبارت گزینه (۴) برابر  $False$  و بنابراین نادرست است:

$$\gamma \not\models (\alpha \Rightarrow \beta) \equiv \gamma \not\models True \equiv \neg(\gamma \models True) \equiv \neg(\gamma \Rightarrow True) \equiv \neg True \equiv False$$

۵- گزینه (۱) صحیح است.

گزینه (۱) عبارت صحیحی است که در متن درس به آن اشاره شده است.

گزینه (۲) نادرست است: وقتی می‌گوییم یک جمله ارضا پذیر است یعنی حداقل به یک صورت می‌توان به متغیرهای آن مقداردهی کرد بطوریکه آن جمله برابر درست شود. به عبارت دیگر، ارضا پذیر بودن یک جمله به این معنی است که اگر برای آن جمله، جدول درستی رسم کنیم، آن جمله در حداقل یک سطر از جدول درستی برابر  $True$  است. اما وقتی می‌گوییم یک جمله معتبر ( $Valid$ ) یا «همیشه درست» است یعنی به هر صورت به متغیرهای آن جمله مقدار بدهید آن جمله برابر  $True$  خواهد شد (هیچگاه  $False$  نمی‌شود). به عبارت دیگر، آن جمله در تمام سطرها جدول درستی خود برابر  $True$  است.

گزینه (۳) نادرست است: در صورتی که از الگوریتم  $DPLL$  به همراه هیوریستیک‌های ذکر شده در متن درس استفاده کنیم معمولاً الگوریتم سریع‌تر از نمایی خواهد بود. همچنین الگوریتم  $Walk Sat$  معمولاً تمام فضای جستجو را بررسی نمی‌کند و بنابراین پیچیدگی زمانی آن کمتر از نمایی است.

گزینه (۴) نادرست است: هر مسئله  $SAT$  یک  $CSP$  است: باید به متغیرها به گونه‌ای مقدار بدهیم که جمله  $True$  شود (محدودیت مسئله، همان  $True$  شدن جمله به ازای مقدار متغیرها است).

۶- گزینه (۲) صحیح است.

عبارت داده شده ۶ متغیر دارد که کلاً  $2^6 = 64$  حالت (۶۴ مدل یا دنیا یا سطر جدول درستی) می‌توانند داشته باشند. می‌خواهیم ببینیم در چه کسری از این ۶۴ مدل، عبارت داده شده درست است.

برای این عبارت می توان به صورت زیر تحلیل کرد: عملگری که بعد از همه اعمال می شود عملگر  $\leftrightarrow$  قبل از  $F$  است. متغیر  $F$  به نصف سطرهای جدول درستی برابر  $True$  و در نصف دیگر آن  $False$  است. این موضوع را اینگونه بیان می کنیم که احتمال درست بودن  $F$  برابر  $\frac{1}{4}$  و احتمال نادرست بودن آن هم  $\frac{1}{4}$  است. عبارت سمت چپ  $\leftrightarrow$  فوق الذکر، یک عبارت با ۵ متغیر است که  $2^5 = 32$  حالت مختلف می توانند مقدار داشته باشند. فرض کنید این عبارت (که آن را  $L$  می نامیم) در  $n$  حالت از این ۳۲ حالت ارزش  $True$  داشته باشد. پس احتمال درستی  $L$  برابر  $\frac{n}{32}$  و احتمال نادرست بودن آن  $\frac{32-n}{32}$  است.

حال به خود عملگر  $\leftrightarrow$  توجه کنید: برای اینکه حاصل این عملگر برابر  $True$  شود باید یا دو سمت آن هر دو  $True$  باشند، یا هر دو  $False$ . بنابراین احتمال درست بودن کل عبارت به صورت زیر بدست می آید:

$$p(L \text{ نادرست بودن } F) \times p(F \text{ نادرست بودن } L) + p(L \text{ درست بودن } F) \times p(F \text{ درست بودن } L) \\ = \frac{n}{32} \times \frac{1}{4} + \frac{32-n}{32} \times \frac{1}{4} = \frac{32}{64}$$

یعنی احتمال درست بودن کل عبارت برابر  $\frac{32}{64}$  یا  $0.5$  است، یعنی از کل ۶۴ مدل ممکن، عبارت صورت سؤال در نصف آنها، یعنی در ۳۲ مدل، ارزش  $True$  خواهد داشت.

برای یادگیری بیشتر نحوه حل چنین سؤالاتی، توجه به روش حل طولانی تر زیر مفید است: با توجه به اولویت عملگرها و ترتیب اجرای آنها می دانیم که ابتدا  $A \Rightarrow B$  ارزیابی می شود، سپس  $(A \Rightarrow B) \wedge C$  ارزیابی می شود و ... می دانیم که جدول درستی عبارت  $A \Rightarrow B$  چهار سطر دارد و این عبارت در ۳ سطر آن  $True$  است، یعنی احتمال درست بودن آن  $\frac{3}{4}$  است. حال به سراغ

عبارت  $(A \Rightarrow B) \wedge C$  می رویم. احتمال درست بودن  $C$  برابر  $\frac{1}{4}$  است، پس احتمال درست بودن عبارت  $(A \Rightarrow B) \wedge C$  برابر است با  $\frac{3}{4} \times \frac{1}{4} = \frac{3}{16}$ . (یادآوری:  $p(\alpha \wedge \beta) = p(\alpha) \cdot p(\beta)$ )

حال به سراغ  $D \leftrightarrow ((A \Rightarrow B) \wedge C)$  می رویم: احتمال درستی  $D$  برابر  $\frac{1}{4}$  است و عملگر  $\leftrightarrow$  وقتی درست است که دو سوی آن هر دو  $True$  یا هر دو  $False$  باشند، پس احتمال درستی کل این

عبارت برابر است با  $\frac{3}{16} \times \frac{1}{4} + \frac{5}{16} \times \frac{1}{4} = \frac{8}{16}$ . در ادامه، باید عملگر  $\vee$  ارزیابی شود. احتمال درستی

$$E \text{ برابر } \frac{1}{4} \text{ است، پس ارزش عبارت شامل } \vee \text{ برابر است با: } \frac{8}{16} + \frac{1}{4} - \frac{8}{16} \times \frac{1}{4} = \frac{9}{16} \\ \text{(یادآوری: } p(\alpha \vee \beta) = p(\alpha) + p(\beta) - p(\alpha \wedge \beta)\text{)}$$

نهایتاً به سراغ عملگر  $\leftrightarrow$  و متغیر  $F$  می رویم: احتمال درستی  $F$  برابر  $\frac{1}{4}$  است پس احتمال درستی عملگر  $\leftrightarrow$  و کل عبارت برابر است با:

$$\frac{8}{32} \times \frac{1}{2} + \frac{24}{32} \times \frac{1}{2} = \frac{32}{64}$$

توجه کنید که این شیوه حل فقط در شرایطی که در عبارت داده شده هر متغیر فقط یک بار ظاهر شده باشد (چه بدون  $\neg$  و چه با آن) قابل استفاده است.

#### ۷- گزینه (۴) صحیح است.

نحوه استفاده از رزولوشن برای اثبات درستی یک جمله تنها - بدون داشتن جملات دیگر در  $KB$  - روشن نیست. اما ارضایپذیری و معتبر بودن یک جمله را می توان با روش های دیگری تشخیص داد (روش مدنظر طراح هم اگر یک روش صحیح باشد باید نتیجه مشابهی بدهد). برای تعیین معتبر (یا همان «همیشه درست») بودن یک عبارت می توانید کارهای مختلفی انجام دهید: جدول درستی تشکیل دهید و چک کنید آن عبارت به ازای تمام سطرها  $True$  باشد؛ سعی کنید با بکارگیری قوانین هم ارزی گزاره ها، عبارت را ساده کنید تا کل عبارت با  $True$  معادل شود؛ نشان دهید این عبارت را نمی توان  $False$  کرد؛ و غیره. برای تشخیص ارضایپذیر بودن یا نبودن یک جمله هم می توان کارهای مشابهی انجام داد: به متغیرها به شکلی مقدار بدهیم که عبارت  $True$  شود؛ یا نشان دهیم عبارت داده شده همیشه  $False$  نیست یا ...

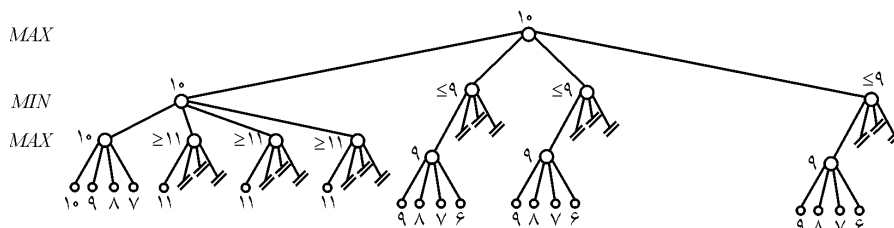
برای مثال، برای بررسی معتبر بودن عبارت سؤال، سعی می کنیم آن را  $False$  کنیم. سطح بالاترین عملگر (یعنی عملگری که آخر همه اعمال می شود) عملگر  $\rightarrow$  است و عملگر  $\rightarrow$  تنها در حالت  $False \rightarrow True$  برابر  $False$  می شود. پس برای اینکه کل عبارت  $False$  شود باید  $P \rightarrow (Q \rightarrow R)$  برابر  $True$  و  $(P \rightarrow Q) \rightarrow (P \rightarrow R)$  برابر  $False$  شود.

برای اینکه  $(P \rightarrow Q) \rightarrow (P \rightarrow R)$  برابر  $False$  شود باید  $P \rightarrow Q$  برابر  $True$  و  $P \rightarrow R$  برابر  $False$  شود. از  $False$  بودن  $P \rightarrow R$  می فهمیم که باید  $P = True$  و  $R = False$  باشد. حال که  $P = True$  شد برای اینکه  $P \rightarrow Q$  برابر  $True$  باشد باید  $Q = True$  باشد. حال اگر مقادیر بدست آمده برای متغیرها را در عبارت  $P \rightarrow (Q \rightarrow R)$  بگذاریم این عبارت (که طبق استدلالمان باید  $True$  باشد) برابر  $False$  می شود. پس به تناقض رسیدیم و نتوانستیم کل عبارت را برابر  $False$  کنیم. یعنی عبارت داده شده در سؤال هیچگاه برابر  $False$  نمی شود، یعنی «همیشه درست» یا معتبر است (گزینه ۴).

#### ۸- گزینه (۲) صحیح است.

از متن درس به یاد داریم که در مسائلی که حداکثر ارزش گره ها نامحدود است، بیشترین هرس

زمانی اتفاق می افتد که فرزندان گره‌های  $MAX$  از چپ به راست به ترتیب نزولی ارزش‌شان و فرزندان گره‌های  $MIN$  از چپ به راست به ترتیب صعودی ارزش‌شان قرار گرفته باشند. یک درخت نمونه با عمق ۳ و فاکتور انشعاب ۴ در شکل زیر آمده است و گره‌هایی که با روش هرس آلفا - بتا هرس می شوند نشان داده شده‌اند.



دیده می شود که از سمت چپ‌ترین زیردرخت ریشه ۹ گره و از هر یک از سه زیردرخت دیگر ریشه ۱۵ گره حذف می شود، پس جمعاً  $9 + 15 \times 3 = 54$  گره حذف می شود. اما اگر تنها گره‌های برگ هرس شده را بشمریم جمعاً  $9 + 3 \times 12 = 45$  گره برگ حذف می شود. با توجه به گزینه‌های داده شده می بایست «تعداد برگ‌های هرس شده» مدنظر طراح بوده باشد. سازمان سنجش در کلید اولیه و نهایی گزینه (۲) را بعنوان پاسخ صحیح اعلام کرده است.

۹- گزینه (۳) صحیح است.

این مسئله  $1$ -consistent هست چون هیچ محدودیت یکانی (محدودیت روی تنها یک متغیر) نداریم.

این مسئله  $2$ -consistent نیست: باید به محدودیت‌های دوتایی توجه کنیم: محدودیت  $A < B < C < D$  به ما می گوید بین هر دو متغیر دلخواه یک محدودیت هست (یکی کوچکتر از دیگری است). همچنین محدودیت  $C = A + 3$  یک محدودیت دیگر روی  $A$  و  $C$  است که با توجه به اینکه دامنه متغیرها اعداد ۱ تا ۴ است این محدودیت تنها با مقداردهی  $\{A=1, C=4\}$  برآورده می شود. یعنی برای اینکه یال بین  $A$  و  $C$  سازگار باشد باید دامنه  $A$  به  $\{1\}$  و دامنه  $C$  به  $\{4\}$  محدود شود. اما با این دامنه برای  $C$  نمی توان هیچ مقداری برای  $D$  داشت، یعنی یال از  $C$  به  $D$  سازگار نخواهد بود (یادآوری: طبق تعریف سازگاری یال، یال  $C$  به  $D$  سازگار است اگر به ازای هر مقداری از دامنه  $C$  حداقل یک مقدار مجاز برای  $D$  باقی بماند). پس این مسئله  $2$ -consistent نیست، پس  $strongly\ 2$ -consistent هم نخواهد بود. (و در نتیجه قویاً سازگار ۳ و ۴ هم نخواهد بود).

۱۰- گزینه (۲) صحیح است.

الگوریتم  $A^*$  در حالت جستجوی گرافی با تابع هیوریستیک سازگار، همواره مسیر بهینه را می‌یابد. پس گزینه (۱) عبارت درستی است.

الگوریتم  $A^*$ ، چه در حالت درختی با هیوریستیک قابل قبول و چه در حالت گرافی با هیوریستیک سازگار، گره‌هایی که  $f(n)$  آنها بیشتر از طول مسیر بهینه  $(C^*)$  باشد را گسترش نمی‌دهد  $(f(n)=g(n)+h(n))$ . پس گزینه (۲) عبارت نادرستی است.

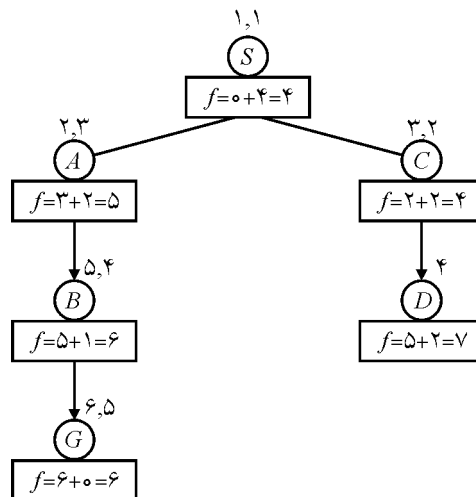
گزینه‌های (۳) و (۴) هر دو عبارت صحیحی هستند (به تفاوت تولید شدن گره و گسترش آن توجه داشته باشید).

۱۱- گزینه (۲) صحیح است.

تابع ابتکاری استفاده شده قابل قبول نیست چون برای گره  $D$  داریم  $h(D)=2$  و  $h^*(D)=1$  و

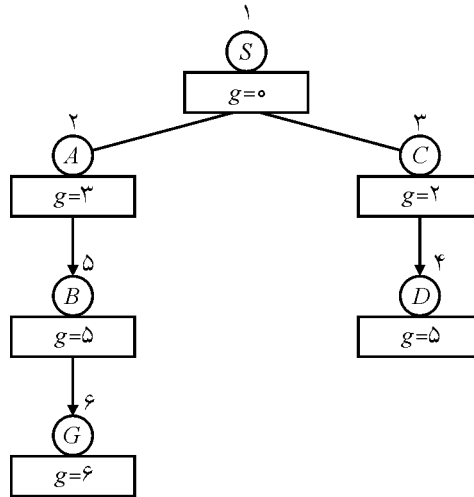
$$h(D) \not\leq h^*(D)$$

درخت جستجوی  $A^*$  به صورت زیر است: (اعداد چپ و راست بالای هر گره، ترتیب تولید و بسط گره را نشان می‌دهند)



بنابراین گره‌ها به ترتیب  $SACDBG$  تولید و به ترتیب  $SCABG$  بسط داده شدند.

درخت جستجوی  $UCS$  به صورت زیر است (عدد بالای هر گره، ترتیب تولید (و نه بسط) آن را نشان می‌دهد):



پس ترتیب تولید گره  $SACDBG$  است. توجه کنید که در درخت فوق هر چند گره  $G$  تولید شده است اما تا رسیدن به هدف نیاز دارد در ادامه، گره  $D$  و سپس گره  $G$  (که زیر  $B$  قرار دارد) برای بسط داده شدن انتخاب شوند ولی چون برای ما تنها ترتیب تولید مهم بود از رسم ادامه درخت جستجو صرف نظر کردیم. با توجه به توضیحات فوق، گزینه (۲) صحیح و سایر گزینه‌ها نادرست است.

۱۲- گزینه (۴) صحیح است.

در مسائلی مثل این سؤال که در آنها هزینه اعمال یکسان نیست الگوریتم‌های  $BFS$  و  $DFS$  بهینه نیستند. همچنین الگوریتم  $UCS$  در مسائلی با هزینه مرحله‌ای صفر و منفی بهینه نیست. بنابراین هیچیک از این الگوریتم‌ها بدست آوردن پاسخ بهینه را تضمین نمی‌کنند.

۱۳- گزینه (۳) صحیح است.

مثال نقض برای گزینه (۱)، جاروبرقی با سنسور تمام اتاق‌ها و با احتمال ظاهر شدن تصادفی گرد و خاک است، این محیط پویا است اما کاملاً قابل مشاهده است. مثال نقض برای گزینه (۲) بازی تخته نرد است: کاملاً مشاهده‌پذیر اما غیرقطعی است. گزینه (۴) نادرست است. چون طبق تعریف، در محیط اپیزودیک، تصمیم عامل در هر مرحله (در هر اپیزود) ارتباطی با تصمیم و  $action$  عامل در اپیزودهای قبلی ندارد.

۱۴- گزینه (۳) صحیح است.

بخش «مولد مسئله» مسئول پیشنهاد فعالیت‌های اکتشافی است که منجر به تجربیات آموزنده

جدیدی می شود. بنابراین گزینه (۳) صحیح است.

#### ۱۵- گزینه (۲) صحیح است.

گزینه (۱) صحیح نیست: هر چند هیوریستیک *empty-delete list* برای مسائل زیادی عملکرد مناسبی (بهبود از سایر هیوریستیک‌ها) داشته است اما قادر نیست برای تمام مسائل جواب بهینه را بدست آورد. همچنین حل مسئله طرح‌ریزی ساده شده از مرتبه نمایی است، هر چند معمولاً در عمل سربار زیادی ایجاد نمی‌کند.

گزینه (۲) صحیح است: از *Graph-Plan* هم می‌توان یک طرح استخراج کرد و هم می‌توان از روی آن به هیوریستیک‌هایی برای بکارگیری در روش‌های جستجوی دیگر دست یافت.

گزینه (۳) نادرست است: الگوریتم *POP* به صورت عقب‌رو عمل می‌کند، پیش‌شرط‌های هدف را در نظر می‌گیرد و سعی می‌کند کنش‌هایی را بررسی کند که آن پیش‌شرط‌ها را برآورده کنند، خود این کنش‌ها می‌توانند پیش‌شرط‌هایی داشته باشند و ...

گزینه (۴) نادرست است: همانگونه که در متن درس اشاره شد هیوریستیک «تعداد اهداف ارضا نشده» قابل قبول نیست.

#### ۱۶- گزینه (۱) و (۴) صحیح‌اند.

گزینه (۱) عبارت درستی است: این جمله عیناً در متن درس و همچنین در کتاب راسل آمده است (کتاب راسل، ویرایش سوم، فصل نهم، صفحه ۳۳۱، پاراگراف آخر، جمله سوم):

*Second, it is complete for definite clause knowledge bases; that is, it answers every query whose answers are entailed by any knowledge-base of definite clauses.*

گزینه (۲) عبارت کاملاً درستی نیست: در واقع الگوریتم *FC*، «کامل کاذب» است، یعنی برای هر جمله داده شده می‌تواند تعیین کند آیا آن جمله از *KB* قابل نتیجه‌گیری هست یا خیر. اما *FC* «کامل» نیست، یعنی نمی‌تواند تمام جملاتی که *KB* مستلزم آنها است را از *KB* نتیجه بگیرد.

گزینه (۳) عبارت نادرستی است و عکس این عبارت عیناً در متن درس و در کتاب راسل آمده است (کتاب راسل، ویرایش سوم، فصل هفتم، صفحه ۲۵۵، پاراگراف اول، جمله اول):

*To conclude our discussion of resolution, we now show why PL-RESOLUTION is complete.*

گزینه (۴) صحیح است: همانطور که در متن درس ذکر شد، استنتاج با فراکردهای معین در منطق مرتبه اول، «نیمه تعمیم‌پذیر» (*Semi-decidable*) است، یعنی الگوریتم در برخی موارد خاتمه نمی‌یابد. مثلاً اگر *KB* مستلزم  $\alpha$  نباشد ممکن است الگوریتم خاتمه نیابد (کتاب راسل، ویرایش



سوم، فصل نهم، صفحه ۳۳۳، پاراگراف سوم، جمله آخر):

*As with general first-order-logic, entailment with definite clause is semidecidable.*  
سازمان سنجش در کلید اولیه گزینه (۴) و در کلید نهایی گزینه‌های (۱) و (۴) را بعنوان پاسخ صحیح اعلام کرده است.

۱۷- گزینه (۱) صحیح است.

طبق صورت سؤال،  $\gamma$  از رزولوشن  $\alpha$  و  $\beta$  بدست آمده است. پس فرض کنیم  $\alpha = a \vee b$  و  $\beta = \neg b \vee c$  و  $\gamma = a \vee c$  باشد (این فرض ما محدودکننده نیست و روش استدلال مبتنی بر آن را می‌توان به جملات با چند متغیر تعمیم داد. مثلاً  $a$  یا  $c$  می‌توانند هر کدام ترکیب فصلی چند متغیر دلخواه باشند). این موضوع که  $\gamma$  از رزولوشن  $\alpha$  و  $\beta$  بدست آمده است بدین معنی است که  $\alpha \wedge \beta \models \gamma$  و جمله زیر یک جمله همیشه درست است (هیچگاه نمی‌توانید آن را *False* کنید؛ همچنین با ساده‌سازی آن به *True* می‌رسید):

$$[(a \vee b) \wedge (\neg b \vee c)] \Rightarrow (a \vee c)$$

حال نشان می‌دهیم عبارت گزینه (۱)، یعنی  $\neg \alpha \wedge \neg \beta \models \gamma$  یا همان  $\neg \alpha \wedge \neg \beta \Rightarrow \gamma$  هم عبارت همیشه درستی است:

$$(\neg \alpha \wedge \neg \beta) \Rightarrow \gamma \equiv [\neg(a \vee b) \wedge \neg(\neg b \vee c)] \Rightarrow (a \vee c)$$

$$\equiv (a \vee b) \vee (\neg b \vee c) \vee (a \vee c) \equiv a \vee b \vee \neg b \vee c \equiv a \vee \text{True} \vee c \equiv \text{True}$$

اما عبارت گزینه (۲) صحیح نیست: جمله  $\alpha \wedge \beta \Rightarrow \gamma$  عبارت همیشه درستی نیست:

$$\gamma \Rightarrow (\alpha \wedge \beta) \equiv (a \vee c) \Rightarrow [(a \vee b) \wedge (\neg b \vee c)]$$

عبارت فوق مثلاً به ازای  $a = \text{False}$  و  $b = \text{False}$  و  $c = \text{True}$  ارزش *False* خواهد داشت.

عبارت گزینه (۳) نیز صحیح نیست چون  $\alpha \vee \beta \Rightarrow \gamma$  جمله همیشه درستی نیست:

$$(\alpha \vee \beta) \Rightarrow \gamma \equiv [(a \vee b) \vee (\neg b \vee c)] \Rightarrow (a \vee c)$$

عبارت فوق به ازای  $a = \text{False}$  و  $c = \text{False}$  و هر مقداری از  $b$  ارزش *False* خواهد داشت.

عبارت گزینه (۴) نیز صحیح نیست چون  $\alpha \Rightarrow \beta \Rightarrow \gamma$  همیشه درست نیست:

$$(a \vee c) \Rightarrow [(a \vee b) \Rightarrow (\neg b \vee c)]$$

جمله فوق به ازای  $a = \text{True}$  و  $b = \text{True}$  و  $c = \text{False}$  ارزش *False* خواهد داشت.

۱۸- گزینه (۲) صحیح است.

چهار متغیر داریم که جمعاً  $2^4 = 16$  حالت (همان مدل یا دنیا یا سطر جدول درستی) می‌توانند داشته باشند. می‌خواهیم ببینیم هر یک از جملات داده شده در چند مدل ارزش *True* دارند. یک راه دقیق و ساده، تشکیل جدول درستی برای جملات و شمردن سطرهایی است که هر جمله ارزش *True* دارد. برای تسریع در کار، به جای جدول درستی از جدول کارنو درس مدار منطقی استفاده می‌کنیم. تنها نکته‌ای که باید به آن توجه داشت این است که برای جملاتی که شامل دو یا سه متغیر هستند هم باید جدول کارنو را با تمام ۴ متغیر رسم کنیم چون هر مدل باید شامل مقداردهی به تمام متغیرها باشد.

AB \ CD	00	01	11	10
00			1	
01			1	
11		1	1	
10		1	1	

جمله  $(A \wedge B) \vee (B \wedge C)$  با توجه به جدول مقابل در ۶ مدل ارزش *True* دارد:

AB \ CD	00	01	11	10
00				
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

جمله  $D \vee C$  با توجه به جدول مقابل در ۱۲ مدل ارزش *True* دارد:

AB \ CD	00	01	11	10
00	1	0	1	0
01	1	0	1	0
11	1	0	1	0
10	1	0	1	0

برای جمله  $A \oplus B \oplus C$  ابتدا حالت‌های درستی  $A \oplus B$  را مشخص می‌کنیم: (یادآوری:  $A \oplus B$  وقتی *True* است که  $A$  و  $B$  هر دو *True* یا هر دو *False* باشند)

AB \ CD	00	01	11	10
00		1		1
01		1		1
11	1		1	
10	1		1	

اگر کل عبارت  $A \oplus B$  را با  $X$  نشان دهیم حال حالت‌های درستی  $X \oplus C$  را مشخص می‌کنیم (حالت‌هایی که  $X$  و  $C$  هر دو *True* یا هر دو *False* باشند). توجه کنید که  $C$  در دو سطر آخر *True* و در دو سطر اول *False* است.

پس عبارت  $A \leftrightarrow B \leftrightarrow C$  در ۸ مدل ارزش *True* دارد.  
هیچیک از گزینه‌ها پاسخ کاملاً صحیح را ندارند اما در گزینه (۲) پاسخ دو مورد از جملات بدرستی آمده است. سازمان سنجش در کلید اولیه گزینه (۲) را بعنوان پاسخ صحیح اعلام کرده بود اما در کلید نهایی اعلام کرد گزینه (۲) با تأثیر مثبت در نظر گرفته می‌شود.

#### ۱۹- گزینه (۲) صحیح است.

در مورد گزینه (۱): در زمانی که مجموع امتیازها ثابت باشد (بازی *Zero-sum* باشد) اضافه شدن امتیاز یک بازیکن به معنای کم شدن امتیاز رقیب وی است. در این شرایط، که شرایط مدنظر در کتاب راسل برای نوشتن شبه کد الگوریتم *MiniMax* بوده است، می‌توان ارزش هر گره را فقط از نظر *Max* ذکر کرد (همان مقدار *Minimax*) و بازیکن *Max* به دنبال گره با ارزش بیشتر باشد و بازیکن *Min* برعکس.

اما زمانی که مجموع امتیازها ثابت نباشد نمی‌توان ارزش هر گره را تنها با یک عدد (مقدار *Minimax*) تعیین کرد چون بیشتر شدن سودمندی یک گره برای یک بازیکن به معنای کم شدن سودمندی آن گره برای بازیکن رقیب نیست. (در این شرایط یک عدد تنها نمی‌تواند تعیین کند به هر بازیکن چه میزان سودمندی خواهد رسید و حتی یک گره می‌تواند سودمندی هر دو بازیکن را افزایش داده باشد. اساساً اگر مجموع امتیازها ثابت نباشد الگوریتم *Minimax* را نمی‌توان با داشتن تنها یک مقدار *Minimax* برای هر گره پیاده‌سازی کرد چون اگر این تنها مقدار نشان دهنده سودمندی *max* باشد برای گره‌های *Min* نمی‌توانیم تعیین کنیم بازیکن *Min* کدام حرکت را انتخاب خواهد کرد!) در چنین شرایطی سودمندی هر گره را برای هر بازیکن با یک عدد جداگانه ذکر می‌کنیم (مشابه بازی‌های چندنفره). در این صورت با فرض اینکه هر بازیکن سودمندی هر گره برای بازیکن رقیبش را هم می‌داند، الگوریتم *Minimax* قابل پیاده‌سازی است و کماکان این الگوریتم سودمندی بهینه را برای بازیکن ریشه درخت تضمین می‌کند (همانند بازی‌های چندنفره).

در مورد گزینه (۲): این جمله نادرست است: آخرین حرکت را هر بازیکنی انجام داده باشد در انتهای بازی مقداری امتیاز (سودمندی) به هر یک از دو بازیکن داده می‌شود. اگر مجموع امتیازها ثابت باشد (مثلاً برابر  $n$  باشد) مرسوم است که برای سادگی ارزش *Minimax* هر گره را معادل با سودمندی حاصل برای *Max* بیان می‌کنند (بدیهی است که سودمندی آن گره برای *Min* برابر است با  $n$  منهای سودمندی حاصل برای *Max*)، اما این عدد وابسته به بازیکنی که آخرین حرکت را انجام داده نیست.

در مورد گزینه (۳): این جمله صحیح است و در متن درس به آن اشاره شده است.  
 در مورد گزینه (۴): دقت کنید که در این گزینه گفته شده هر بازیکن «سودمندی خود» را ماکزیمم می‌کند (نه مقدار *Minimax* گره را) که جمله درستی است.  
 مقدار *Minimax* یک گره (که تنها در بازی‌های مجموع صفر قابل استفاده است) مقدار سودمندی حاصل برای بازیکن *Min* را نشان نمی‌دهد و اتفاقاً هر چه *Minimax* کمتر باشد سودمندی *Min* بیشتر است. به عبارت دیگر، بازیکن *Min* وقتی سعی می‌کند به گرهی با مقدار *Minimax* کمتر برود بدین معنی است که دارد سعی می‌کند سودمندی خودش را بیشتر کند.  
 سازمان سنجش در کلید اولیه و نهایی گزینه (۲) را بعنوان پاسخ صحیح اعلام کرده است.

#### ۲۰- گزینه (۳) صحیح است.

گزینه (۱) نادرست است: چون تابع  $h$  قابل قبول نیست تضمینی نیست که مسیر بهینه را بیابد.  
 گزینه (۲) هم نادرست است: غیر قابل قبول بودن  $h$  به این معنا نیست که قطعاً مسیر بهینه را نخواهد یافت، بلکه تضمینی برای آن نیست.  
 گزینه (۳) درست است: مسیر بهینه‌ای که از  $s$  می‌گذرد هزینه‌اش حداکثر دو برابر هزینه واقعی تخمین زده می‌شود:  $s$  می‌تواند همان گره شروع باشد پس هزینه واقعی تا هدف برابر  $h^*(s)$  و هزینه تخمینی آن برابر  $2 \times h^*(s)$  است. (اگر  $s$  گرهی غیر از گره شروع باشد هزینه تخمینی مسیر بهینه گذرنده از  $s$  کمتر از دو برابر هزینه واقعی آن خواهد بود). اگر هیچ مسیری با هزینه‌ای کمتر از مسیر بهینه گذرنده از  $s$  وجود نداشته باشد الگوریتم  $A^*$  نهایتاً این مسیر را خواهد یافت و هیچگاه مسیری با هزینه بیشتر را بررسی نخواهد کرد.  
 گزینه (۴) نادرست است: مسیر پیدا شده توسط  $A^*$  و هیوریستیک غیر قابل قبول ذکر شده (چه آن مسیر بهینه باشد چه نباشد) ممکن است از گره  $s$  عبور کند.