دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

هوش مصنوعی

نیم‌سال دوم ۹۹ـ۹۸

استاد: محمدحسین رهبان

پاسخ‌نامه تمرین اول — جست‌وجو

---

۱. (آ) deterministic - single-agent- sequential - discrete - fully observable

(ب) stochastic - multi-agent - sequential - continuous - partially observable

(ج) deterministic - multi-agent- sequential - discrete - fully observable

۲. در تمامی بخش‌ها منظور از $d(v)$ فاصله‌ی واقعی راس v تا هدف است.

(آ) هیچ کدام از دو شرط لزوما برقرار نیست.
شکل ۱ را در نظر بگیرید که در آن S مبدا و T هدف است.

شکل ۱: مثال نقض admissable بودن سوال ۲ الف

در این شکل h_iها هر دو هم admissable هستند و هم monotonic. admissable بودن به علت

$$h_i(S) = ۹ \leq ۱۰ = d(S), h_i(T) = ۰ \leq ۰ = d(T)$$

است و monotonic بودن به علت

$$h_i(S) = ۹ \leq ۱۰ = h_i(T) + w(S,T), \quad h_i(Goal) = h_i(T) = ۰$$

اما $h = h_۱ + h_۲$ admissable نیست زیرا $h_۱(S) + h_۲(S) = ۱۸ > ۱۰$. monotonic هم نیست زیرا

$$h(S) = ۱۸ > ۱۰ = h(T) + w(S,T)$$

(ب) هر دو شرط برقرار است.

- admissable است زیرا اگر v یک راس دلخواه باشد.

$$\forall i : h_i(v) \leq d(v) \implies min_i h_i(v) \leq d(v)$$

- monotonic است. برای اثبات باید ثابت کنیم که اولا $h(G) = ۰$ که چون $\forall i : h_i(G) = ۰$، بدیهی است. همچنین باید ثابت کنیم که اگر $a \to b$ یک یال در گراف باشد، داریم

$$h(a) \leq h(b) + w(a,b)$$

یا معادلا

$$h(a) - h(b) \leq w(a,b)$$

به این منظور، داریم

$$h(a) - h(b) = min_i h_i(a) - min_i h_i(b) \tag{۱}$$

قرار دهید $j = argmin_i h_i(b)$. ۱ برابر است با

$$min_i h_i(a) - h_j(b) = min_i h_i(a) - h_j(a) + h_j(a) - h_j(b)$$

اما $min_i h_i(a) - h_j(a)$ که طبق تعریف کمترمساوی ۰ است و $h_j(a) - h_j(b)$ هم طبق فرض monotonicity تابع h_j حداکثر $w(a,b)$ است در نتیجه

$$h(a) - h(b) \leq w(a,b)$$

(ج) هر دو شرط برقرار است.

- admissable است زیرا اگر v یک راس دلخواه باشد.

$$\forall i : h_i(v) \leq d(v) \implies max_i h_i(v) \leq d(v)$$

- monotonic است. برای اثبات باید ثابت کنیم که اولا $h(G) = ۰$ که چون $\forall i : h_i(G) = ۰$، بدیهی است. همچنین باید ثابت کنیم که اگر $a \to b$ یک یال در گراف باشد، داریم

$$h(a) \leq h(b) + w(a,b)$$

یا معادلا

$$h(a) - h(b) \leq w(a,b)$$

به این منظور، داریم

$$h(a) - h(b) = max_i h_i(a) - max_i h_i(b) \tag{۲}$$

قرار دهید $j = argmax_i h_i(a)$. ۲ برابر است با

$$h_j(a) - max_i h_i(b) = h_j(a) - h_j(b) + h_j(b) - max_i h_i(b)$$

اما $h_j(b) - max_i h_i(b)$ که طبق تعریف کمترمساوی ۰ است و $h_j(a) - h_j(b)$ هم طبق فرض monotonicity تابع h_j حداکثر $w(a,b)$ است در نتیجه

$$h(a) - h(b) \leq w(a,b)$$

(د) بله هر دو شرط برقرار است.

- admissable است چون برای یک v دلخواه

$$\frac{1}{n}\sum h_i(v) \leq max_i h_i(v) \leq d(v)$$

- monotonic است زیرا اگر $a \to b$ یک یال در گراف باشد، طبق فرض monotonic بودن تمامی h_iها،

$$\forall i : h_i(a) - h_i(b) \leq w(a,b) \implies \sum_i h_i(a) - h_i(b) \leq \sum_i w(a,b) = nw(a,b) \implies$$

$$\frac{1}{n}\sum_i h_i(a) - \frac{1}{n}\sum h_i(b) \leq w(a,b) \implies h(a) - h(b) \leq w(a,b)$$

۳. (آ) درست ـ با تغییر عمق تمام راس های قبلی باز می‌شوند.
(ب) درست ـ در حالت خاص این امکان وجود دارد.
(ج) غلط ـ سودوکو یک بازی deterministic می‌باشد.
(د) غلط ـ به راحتی می توان با یک مثال نقض نشان داد.
(ه) درست (اثبات در لکچر ۳ درس)

۴. (آ) $d(v) < d*$
(ب) $k(v) < c*$
(ج) $k(v) + h(v) < c*$
(د) بله می‌توان گفت ـ کافیست تا طبق قسمت قبل دو مجموعه رئوسی که توسط هریک باز می‌شوند را بنویسید.

$$k(v) + h_۱(v) < c*$$

$$k(v) + h_۲(v) < c*$$

از آنجایی که $h_۱ > h_۲$ پس تمام رئوسی که با $h_۱$ باز شوند توسط $h_۲$ نیز باز می‌شوند.
(ه) خیر نمی‌توان گفت ـ طبق دو رابطه قسمت قبل لزوما این گزاره برقرار نیست.

۵. (آ) می‌توان برای هر کدام از میله‌ها یک لیست در نظر گرفت که نشان‌دهنده‌ی دیسک‌هایی است که روی آن قرار گرفته است. هر کدام از لیست‌ها مرتب است (با فرض بالا به پایین، به شکل نزولی) و هر حالت مساله وضعیت این ۳ لیست است یعنی یک ۳تایی که عضو اولش لیست اول، عضو دومش لیست دوم و عضو سومش لیست سوم است. در حالت ابتدایی تمامی میله‌ها روی دیسک اول هستند پس نمایشش به فرم زیر است.

$$([n, n-۱, ..., ۲, ۱], [], [])$$

در انتها هم تمامی دیسک‌ها روی میله‌ی انتهایی هستند پس نمایشش به فرم زیر است.

$$([], [], [n, n-۱, ..., ۲, ۱])$$

(ب) در هر وضعیتی که باشیم، می‌توان یک دیسک از روی یک میله‌ها برداشت و روی یک میله‌ی دیگر گذاشت به شرط این که شرط ترتیب در میله‌ی بزرگتر به هم نخورد.
پس اگر عنصر آخر لیست a را با $a[-۱]$ نشان دهیم، می‌توان عنصر انتهایی لیست iام را از لیست i حذف و در لیست j درج کرد به شرط این که شرط ترتیب به هم نخورد یعنی $State[i][-۱] \leq State[j][-۱]$.

(ج) 3^n حالت ممکن وجود دارد چون هر دیسک می‌تواند در هر کدام از میله‌ها قرار بگیرد و با توجه به این که اعضای لیست هر میله باید نزولی باشند و اندازه دیسک‌ها متمایزند، ترتیب اعضا درون لیست به طور یکتا مشخص می‌شود.

۶. (آ) با توجه به این که تنها زرافه‌ها می‌توانند تغییر کنند، می‌توان از یک آرایه‌ی مرتب (چون ترتیب خانه‌ی زرافه‌ها مهم است) از دوتایی‌های i, j که $1 \leq i \leq m, 1 \leq j \leq n$ برای مدل‌سازی استفاده کرد. حالت نهایی هم حالتی است که در آن موقعیت هر زرافه، خانه‌ی متناظر با همان زرافه باشد.

(ب) فرض کنید که k تعداد زرافه‌ها و r تعداد خانه‌های خالی جدول (به جز بلوک‌های پر) باشد. چون هر زرافه r حالت دارد و k زرافه داریم، تعداد حالت‌های مساله r^k است. از آن‌جایی که در کل داریم $r \leq mn$، می‌توان از کران $(mn)^k$ هم استفاد کرد.

(ج) چون برای هر زرافه، حداکثر ۵ حرکت می‌توان متصور شد (ایستادن سر جای خود هم مجاز است)، حداکثر 5^k حالت همسایه وجود دارد که k تعداد زرافه‌هاست.

(د) با ریلکس کردن شرط وجود موانع، می‌توان دید که هر زرافه حداقل به تعداد فاصله‌ی منهتنش از خانه‌اش مرحله نیاز دارد تا به هدف برسد. پس فاصله‌ی منهتن هر زرافه از هدف می‌تواند یک تابع اکتشاف باشد. این فاصله به وضوح admissable است زیرا در بازی ریلکس شده به این تعداد حرکت نیاز است. monotonic هم هست زیرا فاصله‌ی منهتن یک متر است و نامساوی مثلث برای آن برقرار است. پس طبق بخش ج سوال ۲، ماکسیمم این فواصل یک تابع اکتشاف است.