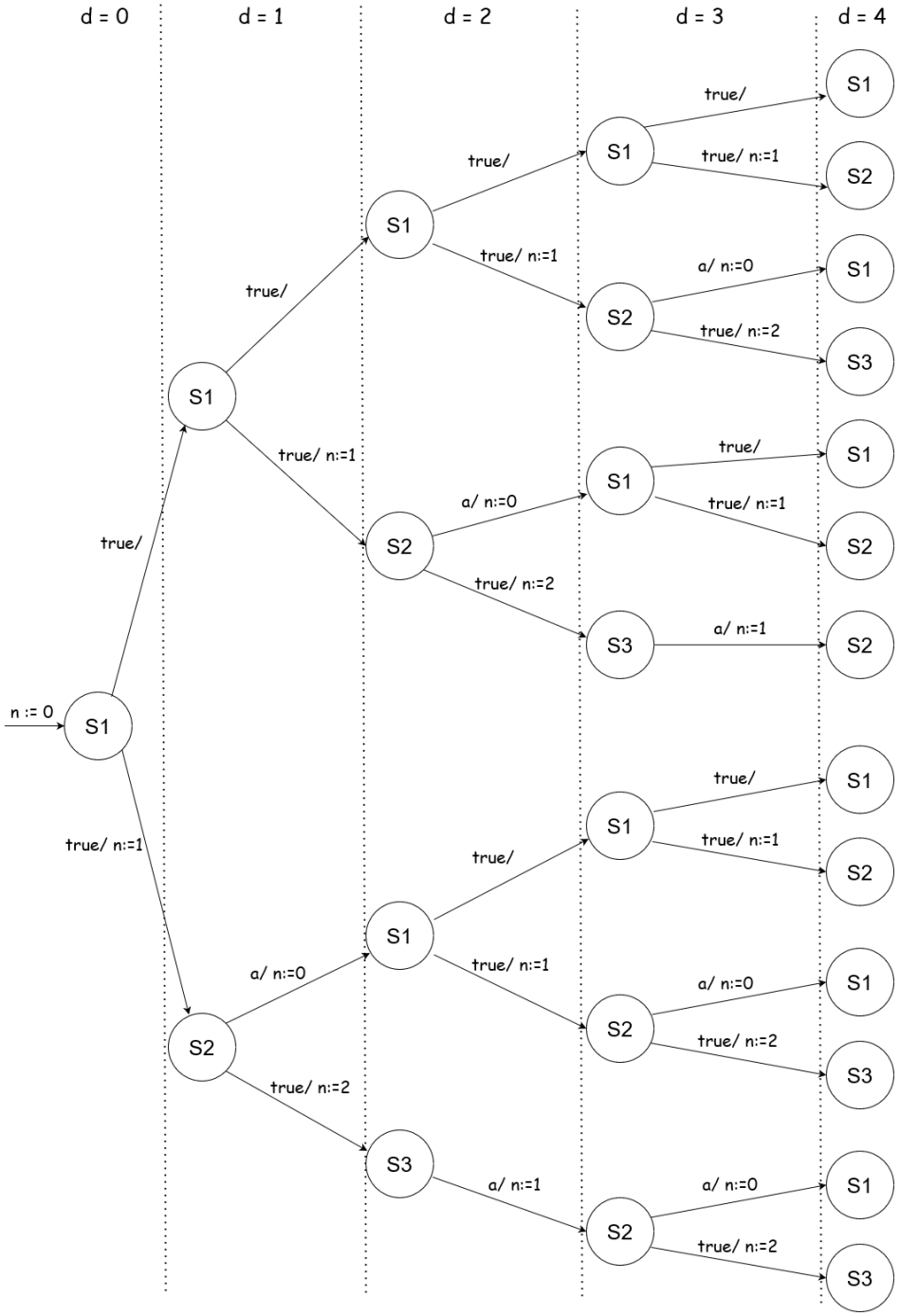
# پاسخ تمرین سری سوم مبانی سیستم‌های نهفته و بیدرنگ

۱. الف) زمانی که در استیت S1 هستیم، با شرط true (یعنی همیشه) می‌توانیم درون S1 بمانیم یا به استیت S2 برویم و مقدار n را یکی زیاد کنیم. زمانی که در استیت S2 هستیم، همواره می‌توانیم به S3 برویم و مقدار n را یکی زیاد کنیم یا می‌توانیم زمانی که مقدار present a بود، به استیت S1 برویم و یکی از n کم کنیم. زمانی که در استیت S3 هستیم، اگر present a بود، می‌توانیم به استیت S2 برویم و از مقدار n یکی کم کنیم.



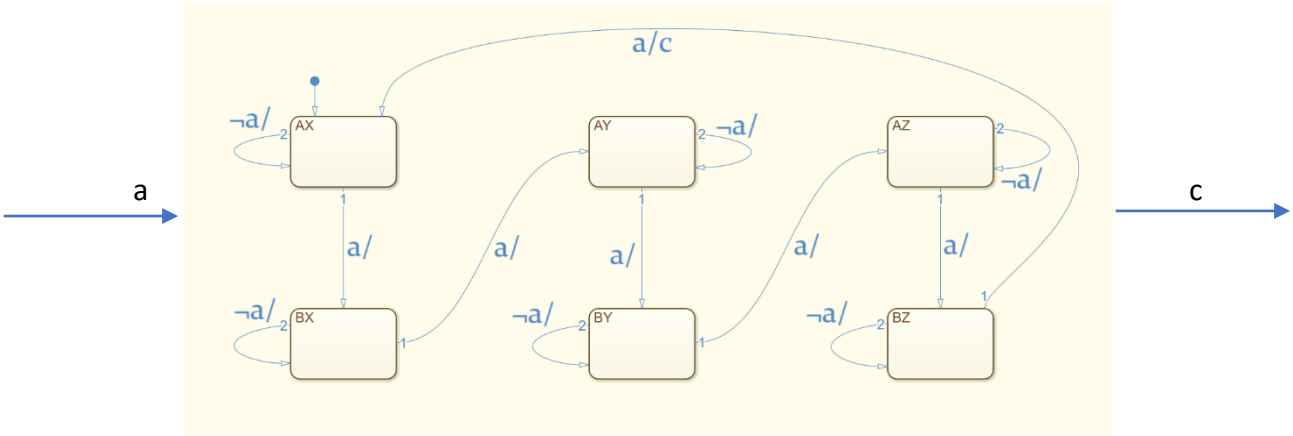
ب) هر سه استیت reachable هستند (البته به دلیل غیرقطعی بودن ماشین، ممکن است به برخی استیت‌ها کلا نرسیم ولی بهرحال باز هم می‌توان به هر سه استیت دسترسی داشت).

۲. تعداد استیت‌های ماشین حالت ترکیبی: 6 = 3 × 2 که این استیت‌ها به صورت روبه‌رو خواهند بود: AX, BX, AY, BY, AZ, BZ

ماشین ترکیبی آن به صورت زیر خواهد بود:

زمانی که در استیت A هستیم، اگر present a شود، به استیت B می‌رویم و مقدار absent b خواهد بود. پس در نتیجه در استیت ماشین سمت راست در استیت X می‌مانیم و مقدار خروجی absent می‌شود. پس در ماشین معادل آن، انگار از استیت AX با مقدار a به استیت BX رفته‌ایم. اگر در استیت A باشیم و مقدار absent a بود، در همین استیت میمانیم و مقدار b نیز absent می‌ماند و در استیت ماشین سمت راست هم در همان استیت X می‌مانیم و خروجی هم absent می‌گردد.

همین روند را برای حالت‌های BX، AY، BY، AZ و BZ هم انجام می‌دهیم تا به ماشین حالت معادل زیر برسیم:



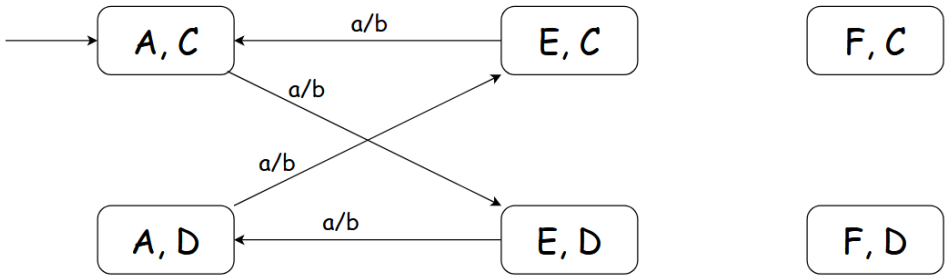

همانطور که میبینیم، در ماشین حالت معادل به تمامی استیت‌ها دسترسی داریم. پس هر ۶ استیت reachable هستند.

۳. الف) در ابتدا، در استیت A و ساب استیت C هستیم. زمانی که مقدار present a شود، به صورت همزمان به استیت D رفته و از A نیز به B و ساب استیت E میرسیم. چون ساب استیت A حافظه‌دار است، پس این استیت را در حافظه نگه میداریم و به همین دلیل زمانی که در ماشین حالت معادل بین E, D و E, C بخواهیم یکی را انتخاب کنیم، باید E, D را انتخاب کنیم تا زمانی که به استیت A برگشتیم، در استیت D قرار بگیریم.

حال دوباره مقدار present a میگردد و به صورت همزمان به استیت F می‌رویم و از B به A و ساب استیت D می‌رویم. توجه شود که چون در لحظه هر دو اکشن اتفاق می‌افتد، رفتن به F با توجه به بی‌حافظه بودن ساب استیت B، انگار رخ نمی‌دهد و فراموش می‌گردد. پس در نتیجه در ماشین حالت معادل از E, D به A, D می‌رویم.

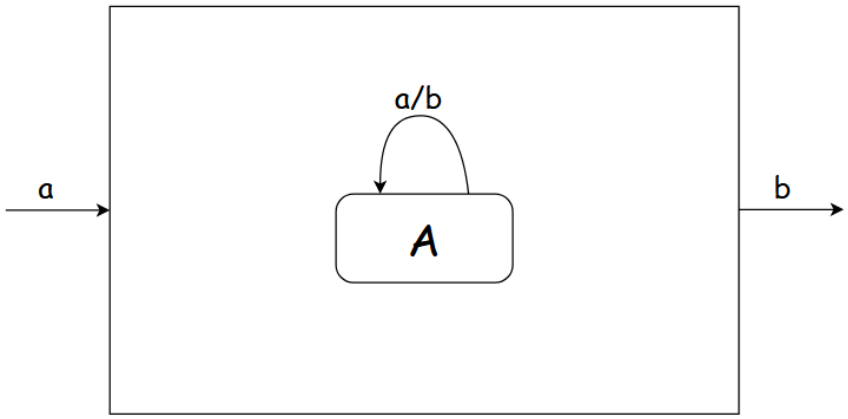
دوباره مقدار present a می‌گردد و همزمان به استیت C رفته و از A به B و ساب استیت E می‌رویم (چون این ساب استیت بی‌حافظه است). پس در نتیجه در ماشین حالت معادل از A, D به E, C می‌رویم.

دوباره مقدار present a می‌گردد و همزمان به استیت F رفته و از B به A و ساب استیت C می‌رویم. پس در نتیجه در ماشین حالت معادل از E, C به A, C که همان حالت اولیه بود، برمیگردیم.
در نتیجه ماشین حالت معادل آن به صورت زیر خواهد شد:




ب) زمانی که مقدار a، present بود، مقدار b را خروجی می‌دهد.

ج)




۴. الف)

$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{cccc} P1 & P2 & P3 & P4 \end{array} \\ \begin{bmatrix} 2 & -1 & 0 & 0 \\ 0 & 2 & -4 & 0 \\ -4 & 0 & 4 & 0 \\ -2 & 0 & 0 & 1 \\ 0 & 0 & 2 & -1 \end{bmatrix} \end{array}$$

$$\Gamma = \begin{bmatrix} 2 & -1 & 0 & 0 \\ 0 & 2 & -4 & 0 \\ -4 & 0 & 4 & 0 \\ -2 & 0 & 0 & 1 \\ 0 & 0 & 2 & -1 \end{bmatrix} = \begin{bmatrix} 2 & -1 & 0 & 0 \\ 0 & 2 & -4 & 0 \\ 0 & -2 & 4 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & 0 & 0 \\ 0 & 1 & -2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 1 \\ 0 & 0 & 2 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -\frac{1}{2} & 0 & 0 \\ 0 & 1 & -2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & 0 & 0 \\ 0 & 1 & -2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -\frac{1}{2} \end{bmatrix} \rightarrow rank(\Gamma) = 3$$

ب)

$$\Gamma q = 0 \rightarrow \begin{bmatrix} 2 & -1 & 0 & 0 \\ 0 & 2 & -4 & 0 \\ -4 & 0 & 4 & 0 \\ -2 & 0 & 0 & 1 \\ 0 & 0 & 2 & -1 \end{bmatrix} \begin{bmatrix} P1 \\ P2 \\ P3 \\ P4 \end{bmatrix} = 0, P1 = t \rightarrow P2 = 2t \rightarrow P3 = t \rightarrow P4 = 2t \rightarrow q$$

$$= \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \end{bmatrix} t \rightarrow \text{مقدار قابل قبول} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \end{bmatrix}$$

ج) همه گره‌ها برای اجرا نیاز به یک مقداری ریسورس دارند. پس در نتیجه، این سیستم بدون یک ***initial tokens*** مشخصی نمیتواند اجرا شود. بهترین نود برای قرار دادن ***initial token***، نودی است که کمترین میزان ریسورس (هم از نظر نوع و هم از نظر مقدار) را نیاز دارد (به این دلیل این کار را میکنیم تا کمترین اندازه بافر را طراحی کنیم).

به همین منظور، نود ***P2*** یا نود ***P4***، نودهای مناسبی هستند.

$$b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow (P2\ fires)\ b = \begin{bmatrix} -1 \\ 2 \\ 0 \\ 0 \end{bmatrix} \rightarrow (P2\ fires)\ b = \begin{bmatrix} -2 \\ 4 \\ 0 \\ 0 \end{bmatrix}$$

$$q = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \end{bmatrix} \rightarrow (P2\ fires)\ q = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \end{bmatrix} \rightarrow (P2\ fires)\ q = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 2 \end{bmatrix}$$

اما می‌دانیم مقادیر بافرها نمیتوانند منفی شوند! پس به بافر ***a***، یک ***initial tokens*** با مقدار ۲ قرار می‌دهیم. حال ***schedule*** ما به صورت زیر خواهد شد:

$$b = \begin{bmatrix}2\\0\\0\\0\\0\end{bmatrix} \rightarrow (P2\ fires)\ b = \begin{bmatrix}1\\2\\0\\0\\0\end{bmatrix} \rightarrow (P2\ fires)\ b = \begin{bmatrix}0\\4\\0\\0\\0\end{bmatrix} \rightarrow (P3\ fires)\ b = \begin{bmatrix}0\\0\\4\\0\\2\end{bmatrix} \rightarrow (P4\ fires)\ b$$

$$= \begin{bmatrix}0\\0\\4\\1\\1\end{bmatrix} \rightarrow (P4\ fires)\ b = \begin{bmatrix}0\\0\\4\\2\\0\end{bmatrix} \rightarrow (P1\ fires)\ b = \begin{bmatrix}2\\0\\0\\0\\0\end{bmatrix}$$

$$q = \begin{bmatrix}1\\2\\1\\2\end{bmatrix} \rightarrow (P2\ fires)\ q = \begin{bmatrix}1\\1\\1\\2\end{bmatrix} \rightarrow (P2\ fires)\ q = \begin{bmatrix}1\\0\\1\\2\end{bmatrix} \rightarrow (P3\ fires)\ q = \begin{bmatrix}1\\0\\0\\2\end{bmatrix} \rightarrow (P4\ fires)\ q$$

$$= \begin{bmatrix}1\\0\\0\\1\end{bmatrix} \rightarrow (P4\ fires)\ q = \begin{bmatrix}1\\0\\0\\0\end{bmatrix} \rightarrow (P1\ fires)\ q = \begin{bmatrix}0\\0\\0\\0\end{bmatrix}$$

➔ *Schedule = (P2; P2; P3; P4; P4; P1)*

اندازه سایز بافر نیز برابر با ماکزیمم مقدار ورودی/خروجی ریسورس گره متناظر با آن است یعنی:

*size(a) = 2, size(b) = 4, size(c) = 4, size(d) = 2, size(e) = 2 -> sum of buffer sizes = 14*

د) نتیجه‌ی اجرای یک حلقه از کد به صورت زیر است:

```
P2 fired:
a -= 1
b += 2
P2 fired:
a -= 1
b += 2
P3 fired:
d -= 4
c += 4
e += 2
P4 fired:
e -= 1
d += 1
P4 fired:
e -= 1
d += 1
P1 fired:
c -= 4
d -= 2
a += 2
```