بسمه تعالیٰ

تمرین اول درس نظریه زبان‌ها و ماشین‌ها

نیم‌سال دوم تحصیلی ۰۵–۱۴۰۴

مسئول تمرین: درسا شریفی

دانشکدهٔ مهندسی کامپیوتر
مدرس: دکتر محمد ایزدی

موعد تحویل پاسخ‌ها سه‌شنبه ۱۵ اردیبهشت‌ماه ۱۴۰۵ ساعت ۲۳:۵۵ است.

۱. این تمرین مجموعا ۱۲۰ نمره است که کسب ۱۰۰ نمره به معنای نمره کامل تمرین است.

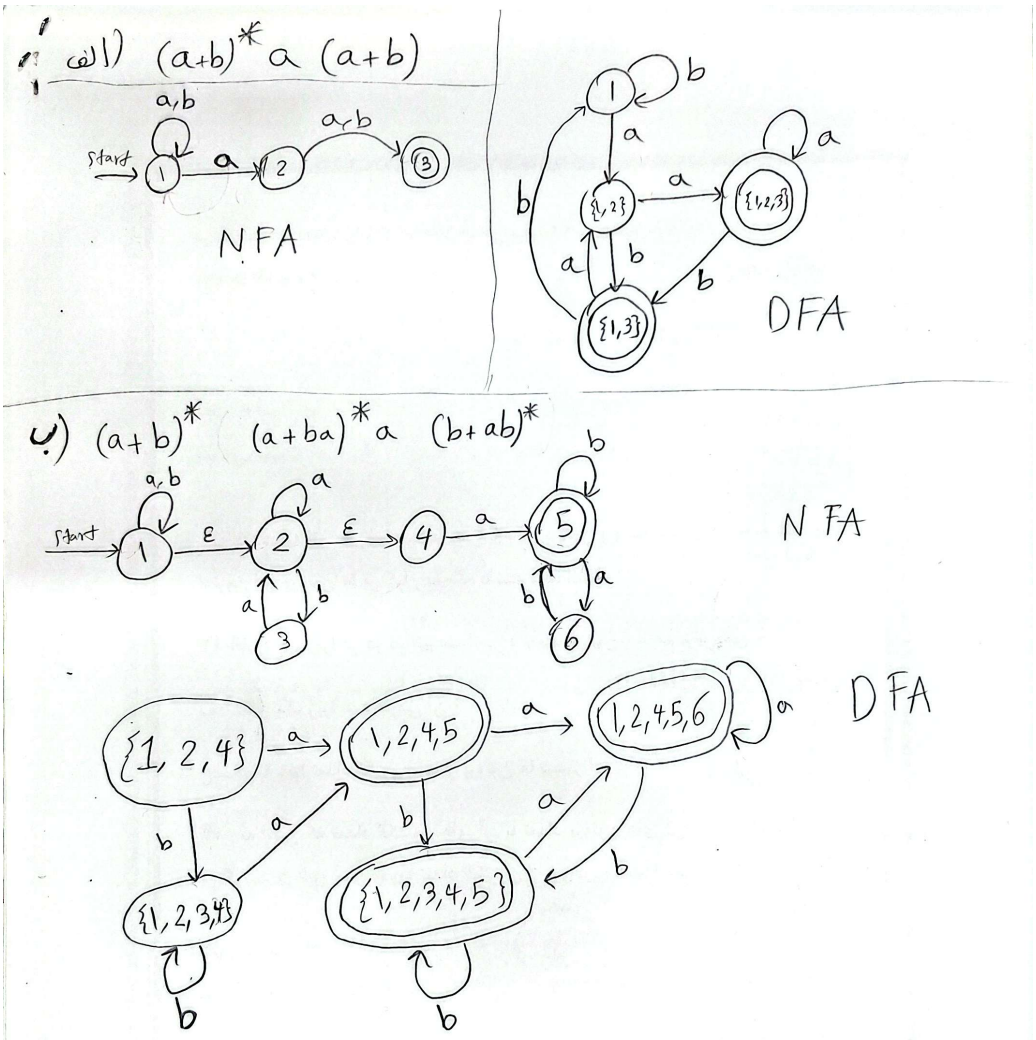
۲. هم‌فکری در حل تمرین‌ها بلامانع است، اما با مشابهت‌های غیرمتعارف در پاسخ‌نامه‌ها مطابق با آئین‌نامه‌ی دانشکده برخورد خواهد شد.

۱. (۱۵ نمره) هر یک از عبارات منظم زیر را ابتدا به *NFA* و سپس به *DFA* معادل آن تبدیل کنید.

(آ) $(a+b)^*a(a+b)$

(ب) $(a+b)^*((a+ba)^*a(b+ab)^*)$

پاسخ:



توجه کنید که راه‌حل‌های دیگری نیز وجود دارد.

۲. (۱۰ نمره) فرض کنید R, S دو عبارت منظم هستند. درستی یا نادرستی تساوی‌های زیر را بررسی کنید. برای تساوی‌های درست، اثبات ارائه دهید و برای تساوی‌های نادرست، مثال نقض بیاورید.

(آ) $(R^*S^*)^* = (R+S)^*$

(ب) $S(RS+S)^*R = RR^*S(RR^*S)^*$

پاسخ:

زبان عبارت منظم X را با $L(X)$ نشان می‌دهیم.

(آ): عبارت سمت راست، می‌گوید که به هر تعداد دلخواه می‌توانیم کلمات عبارت R یا S را پشت سر هم قرار دهیم. عبارت سمت چپ، هم از آنجایی که روی عبارت * هست، می‌توانیم هر تعداد کلمه از R یا S را پشت هم بگذاریم. در واقع هر دو عبارت زبان زیر را نشان می‌دهند:

$$A = \{x_۱x_۲\dots x_n \mid x_i \in (L(R)\cup L(S)),\ n \in (\mathbb{N}\cup\{۰\})\}$$

بنابراین دو عبارت برابر هستند.

(ب): فرض کنید $L(R) = a$ و $L(S) = b$ باشد. آنگاه هر کلمه عبارت سمت راست با حرف a شروع می‌شود اما هر کلمه عبارت سمت چپ با حرف b شروع می‌شود. بنابراین این دو عبارت با هم معادل نیستند.

۳. (۱۰ نمره) زبان L روی الفبای

$$\Sigma = \{a, +, \equiv\}$$

به صورت زیر تعریف شده است:

$$L = \{a^m + a^n \equiv a^p \mid m,n,p \in \{۰,۱,۲,\dots\}, ((m+n) \bmod ۲) = (p \bmod ۲)\}$$

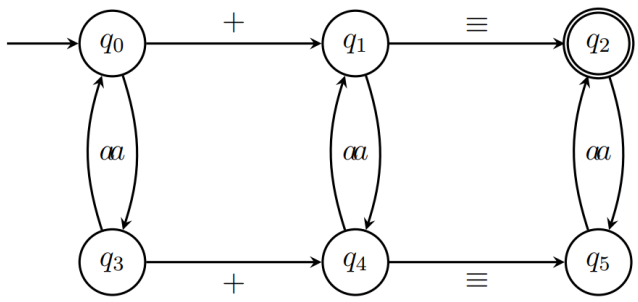
به بیان ساده، رشته‌ها باید شکل

$$a^m + a^n \equiv a^p$$

داشته باشند و زوج یا فرد بودن $m+n$ با زوج یا فرد بودن p یکسان باشد. برای نمونه، رشته‌های زیر در زبان هستند:

$$aaa + aaa \equiv aaaaaa, \quad aa + aaaa \equiv aa, \quad + \equiv, \quad a+ \equiv a, \quad +a \equiv a.$$

NFA این زبان را به صورت زیر رسم می‌کنیم:



این *NFA* را با روش حذف حالت‌ها به یک عبارت منظم معادل تبدیل کنید.

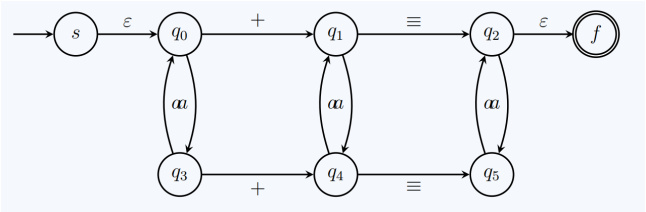
پاسخ:

ابتدا مفهوم حالت‌های ماشین را توضیح می‌دهیم. این ماشین سه بخش رشته را بررسی می‌کند:

$$a^m \qquad a^n \qquad a^p.$$

حالت‌های q_0 و q_3 زوج یا فرد بودن تعداد aها را قبل از نماد $+$ دنبال می‌کنند. حالت‌های q_1 و q_4 زوج یا فرد بودن تعداد کل aهای قبل از نماد $\equiv$ را دنبال می‌کنند. در نهایت، حالت‌های q_2 و q_5 زوج یا فرد بودن تعداد aهای بعد از نماد $\equiv$ را بررسی می‌کنند.

برای به دست آوردن عبارت منظم، NFA را به یک GNFA تبدیل می‌کنیم. برای این کار، یک حالت شروع جدید s و یک حالت نهایی جدید f اضافه می‌کنیم. سپس یک یال ε از s به q_0 و یک یال ε از q_2 به f قرار می‌دهیم.



در روش حذف حالت‌ها، اگر بخواهیم حالت r را حذف کنیم، برچسب جدید یال از حالت i به حالت j طبق رابطه‌ی زیر به دست می‌آید:

$$R'_{ij} = R_{ij} \cup R_{ir}(R_{rr})^* R_{rj}.$$

یعنی همه‌ی مسیرهایی را که از i به r می‌روند، سپس هر تعداد بار روی r می‌چرخند، و بعد از r به j می‌روند، به یال مستقیم از i به j اضافه می‌کنیم. با حذف حالت‌های میانی، در نهایت فقط حالت شروع جدید s و حالت نهایی جدید f باقی می‌مانند. برچسب یال باقی‌مانده از s به f همان عبارت منظم زبان است. پس از انجام حذف حالت‌ها، عبارت منظم زیر به دست می‌آید:

$$(aa)^*((+ \cup a + a)(aa)^*(\equiv \cup a \equiv a) \cup a+ \equiv a)(aa)^*.$$

بنابراین عبارت منظم نهایی برابر است با:

$$\boxed{(aa)^*((+ \cup a + a)(aa)^*(\equiv \cup a \equiv a) \cup a+ \equiv a)(aa)^*}$$

اکنون اجزای این عبارت را توضیح می‌دهیم. عبارت

$$(aa)^*$$

هر تعداد زوج از نماد a را تولید می‌کند؛ بنابراین زوج یا فرد بودن طول بخش مربوط را تغییر نمی‌دهد.

بخش

$$(+ \cup a + a)(aa)^*(\equiv \cup a \equiv a)$$

حالت‌هایی را پوشش می‌دهد که با قرار دادن صفر یا یک a در اطراف نمادهای $+$ و $\equiv$، زوج یا فرد بودن دو طرف رابطه با هم هماهنگ می‌شود. همچنین بخش

$$a+ \equiv a$$

حالتی را پوشش می‌دهد که بخش اول و بخش سوم هر دو فرد هستند و بخش میانی زوج است. پس کل عبارت منظم دقیقاً رشته‌هایی از شکل

$$a^m + a^n \equiv a^p$$

را تولید می‌کند که در آن‌ها زوج یا فرد بودن $m+n$ با زوج یا فرد بودن p یکسان است.

۴. (۱۵ نمره) منظم بودن یا نبودن زبان‌های زیر را اثبات کنید. (فرض کنید L منظم است و الفبا مرجع در دو مورد اول Σ است.)

(آ) $L_{\text{prefix}} = \{x \mid xy \in L \textit{ for some } y \in \Sigma^*\}$

(ب) $L_{\frac{1}{2}} = \{x \mid \exists y \textit{ such that } |x| = |y| \textit{ and } xy \in L\}$

(ج) $\{a^n.b^m.a^k \mid n = m \text{ or } m \neq k\}$

پاسخ:

(آ): ثابت می‌کنیم این زبان منظم است. برای آن DFA تولیدکننده آن را معرفی میکنیم. فرض کنید $D = \{Q, \Sigma, \delta, q_0, F\}$ DFA زبان L باشد. $D' = \{Q, \Sigma, \delta, q_0, F'\}$ را اینگونه تعریف میکنیم که F' شامل تمام استیت‌هایی است که به استیت‌های F راه دارند. (با الگوریتمی مانند BFS می‌توان این استیت‌ها را به دست آورد.) این DFA دقیقا رشته‌هایی را می‌پذیرد که در استیت نهایی خودش راهی به استیت‌های F داشته باشد. بنابراین زبان این ماشین دقیقا L_{prefix} است، پس این زبان منظم است.

(ب): ثابت می‌کنیم این زبان منظم است. برای اثبات NFA ارائه می‌دهیم که این زبانش برابر $L_{\frac{1}{2}}$ باشد. فرض کنید $D = \{Q, \Sigma, \delta, q_0, F\}$ DFA زبان L باشد. ماشین غیرقطعی $N = \{Q', \Sigma, \delta', Q_s, F'\}$ را اینگونه معرفی می‌کنیم:

$$\begin{aligned}
Q' &= Q \times Q \\
\delta'^{((q_i,q_j),a)} &= \{(q_k, q_l) | q_k \in \delta(q_i, a),\ \exists b \in \Sigma \cup \{\epsilon\}\ s.t\ \delta(q_l, b) = q_j\} \\
Q_s &= \{(q_0, q) | q \in F\} \\
F' &= \{(q_i, q_i) | q_i \in Q\}
\end{aligned}$$

این NFA دقیقا زبان خواسته شده را مشخص می‌کند. فرض کنید رشته x در این زبان باشد. در این صورت y هم‌اندازه x وجود دارد که xy در L است. NFA با خواندن x یکی از استیت هایی که می‌تواند به آن برسد (q_i, q_i) است که q_i استیتی است که D با خواندن x به آن می‌رسد، و چون xy در L است، پس با خواندن y از q_i به یکی از استیت‌های نهایی می‌رسیم. با توجه به این توضیحات و تعریف δ با خواندن x به استیت نهایی NFA رسیدیم پس این رشته در زبان است. از طرفی هم اگر با خواندن رشته‌ای دلخواه مانند x به استیتی نهایی مانند (q_i, q_i) برسیم، یعنی y هم‌اندازه با x وجود دارد که xy در L باشد. پس $L_{\frac{1}{2}}$ زبانی منظم است.

به طور شهودی، انگار یک مامور در استیت اولیه و یک مامور در استیت های نهایی قرار داده‌ایم و با خواندن هر کاراکتر مامور اول بر اساس آن یک گام بر میدارد، ولی مامور دوم بدون توجه به آن مسیر برعکس فلش هارا طی میکند. اگر در نهایت این دو به هم برسند، رشته در زبان مدنظر است. NFA ما عملا این روند را شبیه‌سازی می‌کند.

(ج): ثابت می‌کنیم این زبان نامنظم است. از برهان خلف استفاده می‌کنیم. فرض می‌کنیم زبان داده شده (L) منظم باشد. بنابراین طبق لم تزریق عددی مانند $p \geq 1$ وجود دارد که به ازای هر رشته $w \in L$ که $|w| \geq p$ می‌توان آن

را به شکل $w = xyz$ نوشت به طوری که:

- $|y| \geq ۱$ (بخش y نمی‌تواند تهی باشد)
- $|xy| \leq p$
- برای هر $i \geq ۰$ داشته باشیم $xy^iz \in L$

رشته‌ی $w = a^pb^pa^p$ را در نظر می‌گیریم. این رشته متعلق به زبان L است، زیرا شرط اول زبان یعنی $n = m$ ($p = p$) در آن صدق می‌کند. از طرفی طول آن $۳p$ است که از p بزرگتر است پس در شرایط لم تزریق نیز صدق می‌کند. با توجه به شرط $|xy| \leq p$، بخش‌های x و y حتماً در قسمت اول رشته (یعنی در میان aهای اول) قرار می‌گیرند. بنابراین y فقط از حروف a تشکیل شده است. فرض می‌کنیم $y = a^j$ که $j \geq ۱$.
طبق لم تزریق باید $xy^۲z \in L$ باشد. پس داریم:

$$s = xy^۲z = xyyz = a^{p+j}b^pa^p \in L$$

از طرفی برای آنکه $s \in L$ باشد باید در حداقل یکی شروط زبان صدق کند:

- شرط $n = m$: تعداد aهای اول برابر $p + j$ و تعداد bها برابر p است. چون $j \geq ۱$، پس $p + j \neq p$. در نتیجه این شرط برقرار نیست.
- شرط $m \neq k$: تعداد bها برابر p و تعداد aهای آخر نیز برابر p است. بنابراین $m = k$ ($p = p$) و این شرط هم برقرار نیست.

همانطور که مشاهده می‌کنیم s در هیچ کدام از شروط صدق نمی‌کند پس نمی‌تواند عضو L باشد در صورتی که طبق لم تزریق باید باشد. بنابراین به تناقض رسیده‌ایم و فرض خلف باطل است و زبان L نامنظم است.

۵. (۱۵ نمره) یک *DFA* روی الفبای

$$\Sigma = \{۰, ۱\}$$

برای زبان زیر بسازید:

$$L = \{w \in \{۰, ۱\}^* \mid w \in \{۰, ۱\}^*۱۰۱ \text{ and } (|w|_۰ \bmod ۳) = ۲\}.$$

- با ۱۰۱ تمام شود.
- تعداد صفرهای رشته در تقسیم بر ۳ باقی‌مانده‌ی ۲ بدهد.

پاسخ:
هدف این است که ماشینی بسازیم که هم‌زمان دو شرط را بررسی کند:

(آ) رشته با ۱۰۱ تمام شود.

(ب) تعداد صفرهای رشته در تقسیم بر ۳ باقی‌مانده‌ی ۲ بدهد؛ یعنی

$$(|w|_۰ \bmod ۳) = ۲.$$

پس حالت‌های ماشین باید دو نوع اطلاعات را در خود نگه دارند. اطلاعات اول مربوط به انتهای رشته است و اطلاعات دوم مربوط به تعداد صفرهاست.
ابتدا بخش مربوط به انتهای رشته را می‌سازیم. چون می‌خواهیم بررسی کنیم که رشته با ۱۰۱ تمام می‌شود یا نه، کافی است بدانیم آخرین بخش مفید رشته تا چه اندازه با ۱۰۱ هماهنگ است. برای این کار چهار حالت زیر را در نظر می‌گیریم:

$$A, \quad B, \quad C, \quad D.$$

معنای این حالت‌ها چنین است:

- A: هنوز هیچ بخش مفیدی از انتهای رشته با شروع الگوی ۱۰۱ هماهنگ نیست.
- B: رشته‌ی خوانده‌شده در حال حاضر با ۱ تمام می‌شود.
- C: رشته‌ی خوانده‌شده در حال حاضر با ۱۰ تمام می‌شود.
- D: رشته‌ی خوانده‌شده در حال حاضر با ۱۰۱ تمام می‌شود.

پس اگر در حالت D باشیم، یعنی شرط اول، یعنی تمام شدن رشته با ۱۰۱، برقرار است.
اکنون انتقال‌های این بخش را بررسی می‌کنیم. اگر در حالت A باشیم و ۱ بخوانیم، انتهای رشته برابر ۱ می‌شود؛ پس به حالت B می‌رویم. اما اگر ۰ بخوانیم، هنوز هیچ پیشوندی از ۱۰۱ ساخته نشده است؛ پس در A می‌مانیم:

$$A \xrightarrow{۰} A, \quad A \xrightarrow{۱} B.$$

اگر در حالت B باشیم، یعنی رشته با ۱ تمام شده است. اگر ۰ بخوانیم، انتهای رشته ۱۰ می‌شود؛ پس به C می‌رویم. اگر دوباره ۱ بخوانیم، رشته همچنان با ۱ تمام می‌شود؛ پس در B می‌مانیم:

$$B \xrightarrow{۰} C, \quad B \xrightarrow{۱} B.$$

اگر در حالت C باشیم، یعنی رشته با ۱۰ تمام شده است. اگر ۱ بخوانیم، انتهای رشته ۱۰۱ می‌شود؛ پس به D می‌رویم. اما اگر ۰ بخوانیم، انتهای رشته ۱۰۰ می‌شود و دیگر پسوند مفیدی برای ساختن ۱۰۱ نداریم؛ پس به A برمی‌گردیم:

$$C \xrightarrow{۰} A, \quad C \xrightarrow{۱} D.$$

اگر در حالت D باشیم، یعنی رشته در حال حاضر با ۱۰۱ تمام می‌شود. اگر ۰ بخوانیم، انتهای رشته‌ی جدید ۱۰۱۰ است که پسوند مفید آن ۱۰ است؛ پس به C می‌رویم. اگر ۱ بخوانیم، انتهای رشته‌ی جدید ۱۰۱۱ است که پسوند مفید آن ۱ است؛ پس به B می‌رویم:

$$D \xrightarrow{۰} C, \quad D \xrightarrow{۱} B.$$

اکنون باید شرط دوم را نیز وارد حالت‌ها کنیم. شرط دوم مربوط به تعداد صفرهاست. چون فقط باقی‌مانده‌ی تعداد صفرها در تقسیم بر ۳ مهم است، سه حالت شمارشی کافی است:

$$۰, \quad ۱, \quad ۲.$$

اگر تعداد صفرهای خوانده‌شده باقی‌مانده‌ی i در تقسیم بر ۳ داشته باشد، آن را با اندیس i نشان می‌دهیم. با خواندن ۰، تعداد صفرها یکی زیاد می‌شود؛ بنابراین اندیس‌ها به صورت زیر تغییر می‌کنند:

$$۰ \to ۱, \quad ۱ \to ۲, \quad ۲ \to ۰.$$

اما با خواندن ۱، تعداد صفرها تغییر نمی‌کند؛ پس اندیس ثابت می‌ماند. بنابراین هر حالت نهایی ماشین از ترکیب دو بخش ساخته می‌شود:

باقی‌مانده‌ی تعداد صفرها $+$ وضعیت پسوندی

پس مجموعه‌ی حالت‌ها برابر است با

$$Q = \{A_i, B_i, C_i, D_i \mid i \in \{۰, ۱, ۲\}\}.$$

برای مثال، حالت $C_۲$ یعنی:

- رشته‌ی خوانده‌شده با ۱۰ تمام می‌شود؛
- تعداد صفرهای آن در تقسیم بر ۳ باقی‌مانده‌ی ۲ دارد.

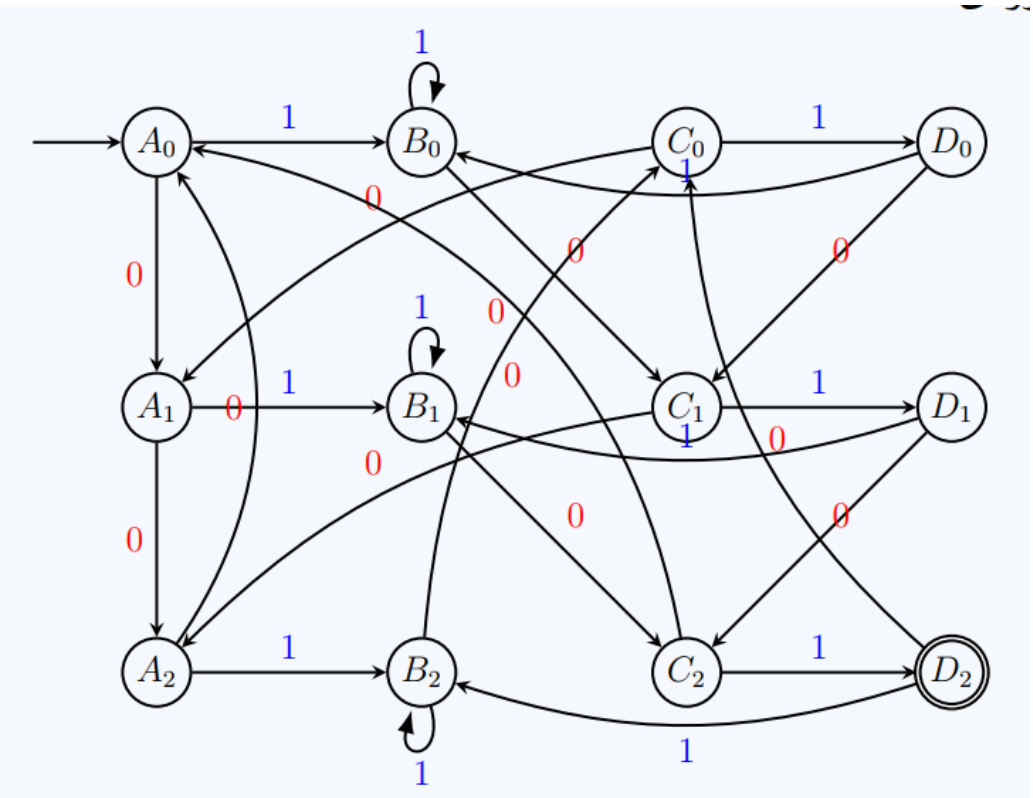
اکنون DFA را به صورت زیر تعریف می‌کنیم:

$$M = (Q, \Sigma, \delta, A_۰, F),$$

که در آن

$$\Sigma = \{۰, ۱\}, \quad F = \{D_۲\}.$$

حالت شروع $A_۰$ است؛ چون در ابتدا هنوز هیچ حرفی نخوانده‌ایم، پس نه پسوند مفیدی داریم و نه صفر دیده‌ایم. تنها حالت پذیرنده $D_۲$ است. دلیل آن این است که در حالت $D_۲$، بخش D نشان می‌دهد رشته با ۱۰۱ تمام شده است و اندیس ۲ نشان می‌دهد تعداد صفرها در تقسیم بر ۳ باقی‌مانده‌ی ۲ دارد. شکل DFA کامل به صورت زیر است. در این شکل، برچسب‌های آبی مربوط به ورودی ۱ هستند و برچسب‌های قرمز مربوط به ورودی ۰ هستند.



در پایان، درستی ماشین روشن است. بخش حرفی حالت، یعنی یکی از A, B, C, D، مشخص می‌کند رشته در حال حاضر با چه پسوندی تمام می‌شود. بنابراین قرار گرفتن در یکی از حالت‌های D_i یعنی رشته با ۱۰۱ تمام شده است.

از طرف دیگر، اندیس حالت نشان‌دهنده‌ی باقی‌مانده‌ی تعداد صفرها در تقسیم بر ۳ است. بنابراین قرار گرفتن در حالتی با اندیس ۲ یعنی تعداد صفرها باقی‌مانده‌ی ۲ دارد.

پس یک رشته دقیقاً زمانی پذیرفته می‌شود که ماشین در حالت $D_۲$ متوقف شود. این دقیقاً همان حالتی است که رشته با ۱۰۱ تمام شده و تعداد صفرهای آن در تقسیم بر ۳ باقی‌مانده‌ی ۲ دارد. در نتیجه، DFA بالا دقیقاً زبان مورد نظر را می‌پذیرد.

۶. (۱۵ نمره) یک ربات روی یک مسیر دایره‌ای با ۵ ایستگاه (شماره‌گذاری شده از ۰ تا ۴ در جهت عقربه‌های ساعت) حرکت می‌کند. ربات در ابتدا در ایستگاه ۰ قرار دارد. الفبای ورودی $\{۱, ۲\}$ است که نشان‌دهنده تعداد گام‌هایی است که ربات باید به جلو بردارد.

ایستگاه شماره ۲ یک میدان مغناطیسی است. اگر ربات دقیقا در ایستگاه ۲ **فرود بیاید**، قطبیت آن معکوس می‌شود. (منظور از فرود آمدن توقف در ایستگاه ۲ می‌باشد.) وقتی قطبیت ربات معکوس است، تمام دستورات بعدی را به صورت دنده‌عقب (خلاف عقربه‌های ساعت) اجرا می‌کند. اگر ربات در حالت معکوس دوباره دقیقا در ایستگاه ۲ فرود بیاید، قطبیت آن به حالت عادی (حرکت به جلو) برمی‌گردد. (دقت کنید که فقط فرود آمدن در ایستگاه ۲ قطبیت را عوض می‌کند، نه صرفا عبور از روی آن).

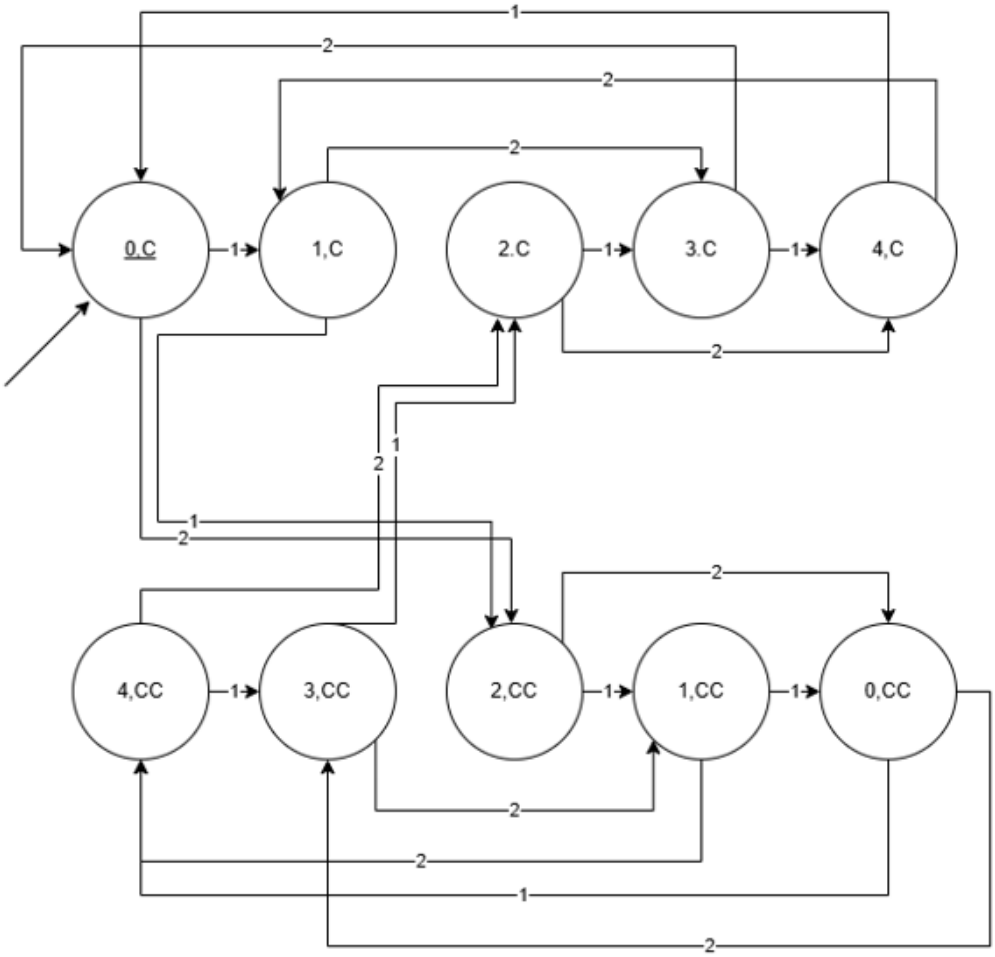
ماشین متناهی قطعی *(DFA)* رسم کنید که تمام دنباله دستوراتی را بپذیرد که باعث می‌شوند ربات در نهایت دقیقا در ایستگاه ۰ و با **قطبیت عادی متوقف شود.** رسم ماشین یا توصیف دقیق و ریاضیاتی پنج‌تایی ماشین (به‌خصوص تابع گذار) الزامی است.

پاسخ:
جهت حرکت اولیه را clockwise و جهت معکوس را counter-clockwise می گیریم.

$$M = (Q, \Sigma, \delta, q_\circ, F)$$

- $Q = \{(p, f) \mid p \in \{\circ, ۱, ۲, ۳, ۴\}, f \in \{c, cc\}\}$
- $\Sigma = \{۱, ۲\}$
- $q_\circ = (\circ, c)$
- $F = \{(\circ, c)\}$
- $\delta =$

Current (p, f)	Input a	New p'	New f'
(p, c)	a	(p + a) mod 5	c if p'≠2, else cc
(p, cc)	a	(p – a) mod 5	cc if p'≠2, else c



۷. (۲۰ نمره) الفبای $\Sigma = \{۱, \#\}$ را در نظر بگیرید. با این الفبا، زبان L را برابر با مجموعه‌ی رشته‌های ساخته شده از تعدادی بلوک ۱ با طول غیر یکسان تعریف می‌کنیم، به طوری که بین هر دو بلوک مجاور یک بار کاراکتر # آمده باشد. به بیان دقیق‌تر:

$$L := \{۱^{x_۱}\#۱^{x_۲}\#\cdots\#۱^{x_k} : k \geq ۱, x_i \geq ۱, x_i \neq x_j \text{ (if } i \neq j)\ \}$$

برای نمونه،

$$
\begin{aligned}
&۱ \in L,\\
&۱۱۱\#۱۱۱ \notin L,\\
&\varepsilon \notin L,\\
&۱\#۱۱۱۱۱\#۱۱ \in L.
\end{aligned}
$$

نشان دهید زبان L منظم نیست.

پاسخ:

مسئله از ما می‌خواهد نشان دهیم زبان L روی الفبای $\Sigma = \{۱, \#\}$ منظم نیست.

به زبان ساده، رشته‌های این زبان از چند بلوک ساخته‌شده از کاراکتر ۱ تشکیل شده‌اند که با علامت # از هم جدا شده‌اند؛ به طوری که طول هیچ دو بلوکی با هم برابر نیست (تمام x_iها متمایز هستند).

برای اثبات منظم نبودن این زبان، از لم پمپاژ برای زبان‌های منظم استفاده می‌کنیم.

فرض خلف می‌گیریم که زبان L منظم است. بنابر قضیه پمپاژ، ثابت پمپاژی مانند $p \geq ۱$ وجود دارد به طوری که هر رشته $w \in L$ با طول $|w| \geq p$ را می‌توان به سه بخش $w = xyz$ تجزیه کرد به طوری که:

(آ) $|xy| \leq p$

(ب) $|y| \geq ۱$ (یعنی $y \neq \epsilon$)

(ج) به ازای هر $i \geq ۰$، رشته $xy^iz \in L$ باشد.

حالا باید رشته‌ای مثل $w \in L$ انتخاب کنیم که طول آن حداقل p باشد و نشان دهیم که با پمپ کردن آن (مثلاً با انتخاب یک i خاص)، رشته حاصل از زبان خارج می‌شود و به تناقض می‌رسیم.

رشته زیر را با $k = ۲$ بلوک انتخاب می‌کنیم:

$$w = ۱^p\#۱^{p+p!}$$

بررسی شرایط رشته انتخاب شده:

- تعداد کاراکترهای بلوک اول p و بلوک دوم $p+p!$ است. از آنجا که $p \geq ۱$ است، قطعاً $p \neq p+p!$ می‌باشد. پس هر دو بلوک طول‌های متمایزی دارند و $w \in L$ است.
- طول رشته واضح است که از p بزرگتر است ($|w| = ۲p + p! + ۱ \geq p$).

طبق شرط اول قضیه پمپاژ، داریم $|xy| \leq p$. از آنجا که رشته با p تا کاراکتر ۱ شروع می‌شود، بخش xy کاملاً درون بلوک اول قرار می‌گیرد. بنابراین:

- $x = ۱^a$
- $y = ۱^b$ که در آن $b \geq ۱$ (طبق شرط دوم $y \neq \epsilon$) و $a + b \leq p$.
- $z = ۱^{p-(a+b)}\#۱^{p+p!}$

- به این ترتیب، بخش پمپ‌شونده y تنها شامل کاراکترهای ۱ از بلوک اول است. طول بلوک اول پس از i بار پمپ شدن برابر خواهد بود با:

$$p + (i - ۱)b$$

می‌خواهیم i را طوری انتخاب کنیم که طول بلوک اول دقیقاً برابر با طول بلوک دوم (یعنی $p + p!$) شود تا شرط متمایز بودن بلوک‌ها نقض شده و رشته از زبان خارج شود. بنابراین قرار می‌دهیم:

$$p + (i - ۱)b = p + p!$$

$$(i - ۱)b = p! \implies i - ۱ = \frac{p!}{b} \implies i = \frac{p!}{b} + ۱$$

از آنجا که b طول بخشی از xy است و $|xy| \leq p$، پس قطعاً $۱ \leq b \leq p$ است. چون b یک عدد طبیعی کوچکتر یا مساوی p است، مطمئناً $p!$ بر b بخش‌پذیر است. در نتیجه $\frac{p!}{b}$ یک عدد صحیح و مثبت است .پس $i \geq ۲$ عددی صحیح و معتبر برای پمپ کردن خواهد بود. با انتخاب این i، رشته پمپ شده برابر است با:

$$w' = xy^iz = ۱^{p+p!}\#۱^{p+p!}$$

در رشته w'، طول بلوک اول و بلوک دوم هر دو برابر با $p+p!$ شده است. از آنجا که دو بلوک هم‌طول در رشته وجود دارد، این رشته به زبان L تعلق ندارد:

$$w' \notin L$$

این یک تناقض با فرض قضیه پمپاژ است (که می‌گفت به ازای هر $i \geq ۰$ باید رشته حاصل درون زبان باشد). بنابراین فرض اولیه ما غلط بوده و **زبان L منظم نیست**.

۸. (۱۲ نمره امتیازی)

(آ) فرض کنید A یک زبان منظم نامتناهی باشد. ثابت کنید که A می‌تواند به دو زیرمجموعه منظم مجزای نامتناهی تقسیم شود.

(ب) فرض کنید B و D دو زبان باشند. می‌نویسیم $B \Subset D$ اگر $B \subseteq D$ و D شامل تعداد نامتناهی رشته باشد که در B نیستند. نشان دهید که اگر B و D دو زبان منظم باشند که $B \Subset D$، آنگاه می‌توانیم یک زبان منظم C پیدا کنیم که $B \Subset C \Subset D$.

پاسخ:
(آ): چون A یک زبان منظم و نامتناهی است پس طبق لم تزریق در آن به ازای عددی مانند $m \geq ۱$ صدق می‌کند. رشته‌ای مانند $s \in A$ در نظر می‌گیریم که $|s| \geq m$ باشد (چنین رشته‌ای حتما وجود دارد چرا که زبان A نامتناهی است). طبق لم تزریق می‌توان آن را به صورت $s = xyz$ نوشت، به طوری که $|y| \geq ۱$ و برای هر $i \geq ۰$ داشته باشیم $xy^iz \in A$. مجموعه تمام این رشته‌های تولید شده را به دو مجموعه‌ی مجزا تقسیم می‌کنیم:

$$L_۱ = \{x(y^۲)^kz \mid k \geq ۰\}$$

$$L_۲ = \{xy(y^۲)^k z \mid k \geq ۰\}$$

این دو زبان اشتراکشان تهی است ($L_۱ \cap L_۲ = \emptyset$).

اکنون ادعا می‌کنیم که دو مجموعه‌ی زیر پاسخ مسئله هستند:

$$A_۱ = A \setminus L_۱$$

$$A_۲ = L_۱$$

برای اثبات این ادعا ۳ شرط باید بررسی شود. اول اینکه این دو مجموعه افرازی از A باشند. دوم اینکه هر کدام منظم باشند و در نهایت هر کدام نامتناهی باشند. شرط اول واضحا برقرار است. به ۲ شرط دیگر می‌پردازیم:

- منظم بودن: $L_۱$ زبانی منظم است چرا که می‌توان برای آن عبارت منظم $x(yy)^*z$ را ارائه داد. پس $A_۲$ منظم است. از طرفی چون خود A منظم است و زبان‌های منظم نسبت به عملگر تفاضل مجموعه‌ها بسته هستند بنابراین $A_۱$ نیز منظم خواهد بود.
- نامتناهی بودن: از آنجا که $|y| \geq ۱$ و k تا هر اندازه می‌تواند بزرگ شود پس $L_۱$ نامتناهی است و درنتیجه $A_۲$ نامتناهی است. به دلیل مشابه $L_۲$ نیز نامتناهی است و چون $L_۲ \subseteq A \setminus L_۱$ پس $A_۱$ نیز نامتناهی خواهد بود.

بنابراین A به دو زبان منظم و نامتناهی $A_۱$ و $A_۲$ تقسیم شده است و حکم ثابت شد.

(ب): زبان L را به صورت $L = D \setminus B$ تعریف می‌کنیم. از آنجا که $B \Subset D$ بنابراین L نامتناهی است. از طرفی چون هر دوی B و D منظم هستند و زبان‌های منظم تحت عمل تفاضل بسته‌اند، در نتیجه L نیز منظم خواهد بود.
طبق قسمت (آ) L را به دو زبان نامتناهی و منظم $L_۱$ و $L_۲$ افراز می‌کنیم. ادعا می‌کنیم که اگر $C = B \cup L_۱$ قرار دهیم، آنگاه $B \Subset C \Subset D$ و C منظم است.
زبان C از اجتماع دو زبان منظم بدست آمده پس قطعا منظم خواهد بود. کافیست نشان دهیم $B \Subset C$ و $C \Subset D$:

- $B \Subset C$: واضح است که $B \subseteq C$ است. از طرفی $C \setminus B = L_۱$ است که چون $L_۱$ نامتناهی است پس داریم: $B \Subset C$
- $C \Subset D$: از آنجا که هم $B \subseteq D$ و هم $L_۱ \subseteq L \subseteq D$ بنابراین $C = B \cup L_۱ \subseteq D$ است. از طرفی دیگر $D \setminus C = L_۲$ است که می‌دانیم $L_۲$ نامتناهی است. پس داریم: $C \Subset D$

بنابراین ثابت شد که زبان منظمی مانند C وجود دارد به طوری که $B \Subset C \Subset D$ برقرار است.

۹. (۸ نمره امتیازی) فرض کنید B و C زبان‌هایی روی الفبای $\Sigma = \{۰, ۱\}$ باشند. تعریف می‌کنیم

$$B \overset{۱}{\leftarrow} C = \{w \in B \mid \textit{for some } y \in C, \textit{ strings } w \textit{ and } y \textit{ contain equal numbers of 1s}\}.$$

نشان دهید که کلاس زبان‌های منظم تحت عملگر $\overset{۱}{\leftarrow}$ بسته است.

پاسخ:

- $Q = Q_B \times Q_C.$

- *For* $(q, r) \in Q$ *and* $a \in \Sigma_\varepsilon$*, define*

$$\delta((q,r),a) = \begin{cases} \{(\delta_B(q,\circ), r)\} & \textit{if } a = \circ \\ \{(\delta_B(q,۱), \delta_C(r,۱))\} & \textit{if } a = ۱ \\ \{(q, \delta_C(r,\circ))\} & \textit{if } a = \varepsilon. \end{cases}$$

- $q_0 = (q_B, q_C)$.
- $F = F_B \times F_C$.