اتوماتا

نیمسال دوم ۹۹ـ۰۰

نام و نام خانوادگی: محمدامین شریفی

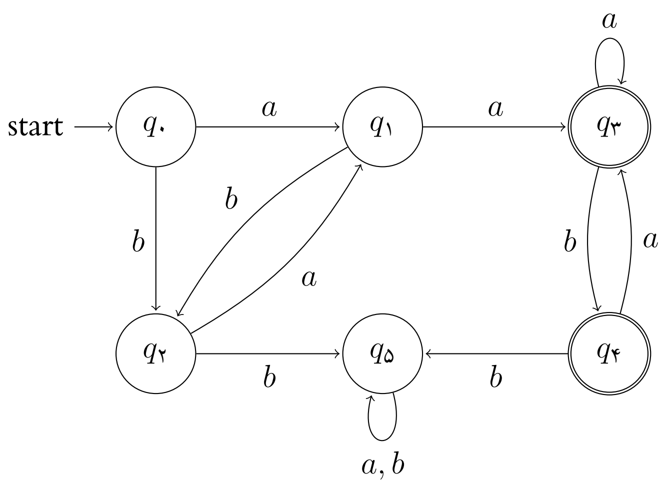
دانشکده‌ی مهندسی کامپیوتر

پاسخ تمرین سری دوم

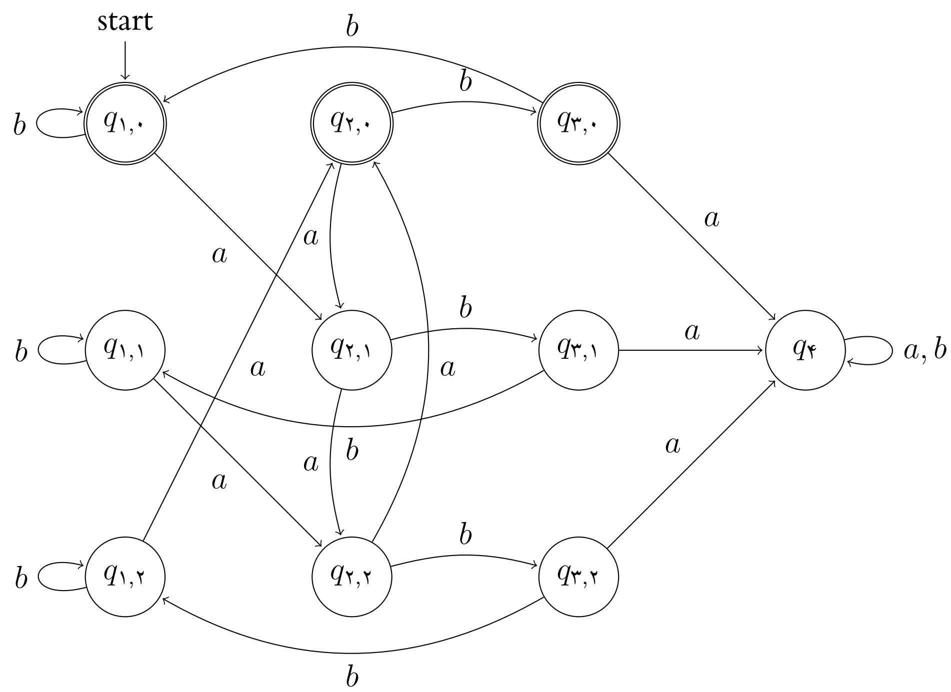
۱ توصیف ماشین‌ها و مفاهیم اولیه زبان‌های منظم

۱.۱

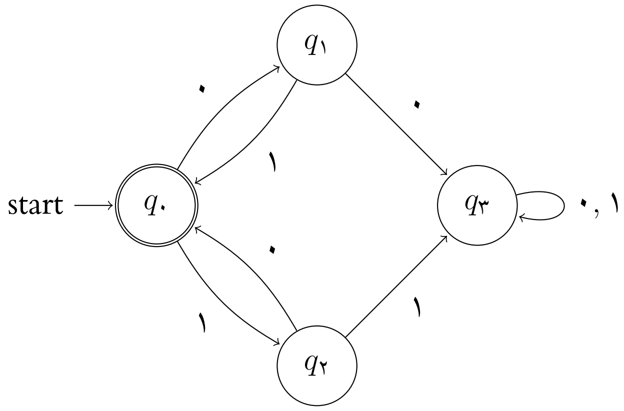
الف) در اتوماتای زیر در هر صورتی اگر دو کاراکتر b متوالی بیایند به استیت $q_۵$ می‌رویم و در آن استیت گیر می‌افتیم. اگر دو کاراکتر a متوالی نیز بیاید به استیت $q_۳$ می‌رسد و از این استیت به بعد در استیت‌های نهایی $q_۳$ و $q_۴$ می‌مانیم مگر اینکه دو کاراکتر b متوالی بیاید و به استیت $q_۵$ برویم.



ب) در اتوماتا زیر استیت‌های با شماره $q_{i,j}$ بصورتی هستند که اگر در این استیت باشیم تعداد a های ورودی داده شده مود ۳ برابر j بوده و عدد i نیز تعداد کارکترهایی است که از رشته aba با انتهای رشته مچ شده‌است.

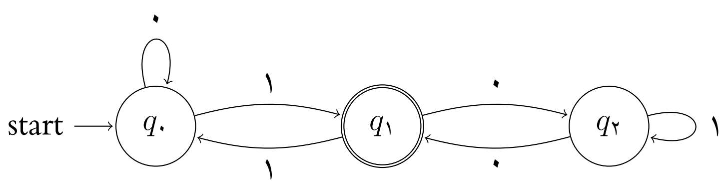


پ) در ۴ استیت این اتومات ۴ حالت در مورد اختلاف تعداد ۱ و ۰های رشته نگهداری می‌شود. در استیت $q_{۰}$، $q_{۱}$ و $q_{۲}$ بترتیب احتلاف تعداد ۰ و ۱ها برابر ۰، ۱ و ۱− است. استیت $q_{۳}$ نیز حالتی غیر از این سه حالت دارد که نشان‌دهنده این است که رشته نباید قبول شود. به همین علت وقتی وارد این استیت شویم در آن باقی می‌مانیم.

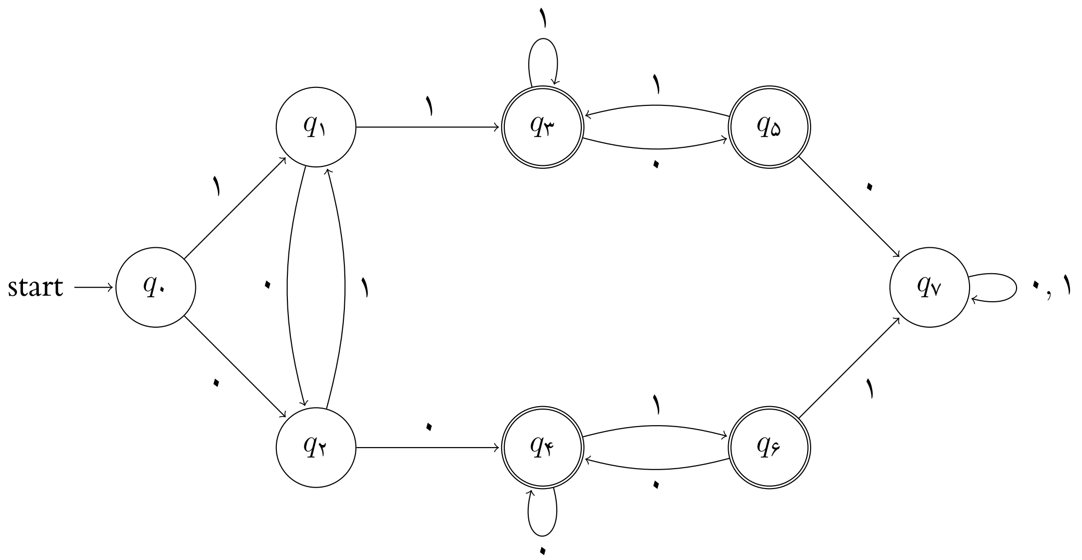


ت) در این قسمت اتوماتا طوری طراحی شده‌است که شماره استیت باقیمانده تقسیم عدد بر ۳ را نشان می‌دهد. به‌طور دقیق تر اگر رشته وارد شده تا لحظه i برابر $a_{۱} \cdots a_i$ باشد و ورودی بعدی $a_{i+۱}$ باشد داریم:

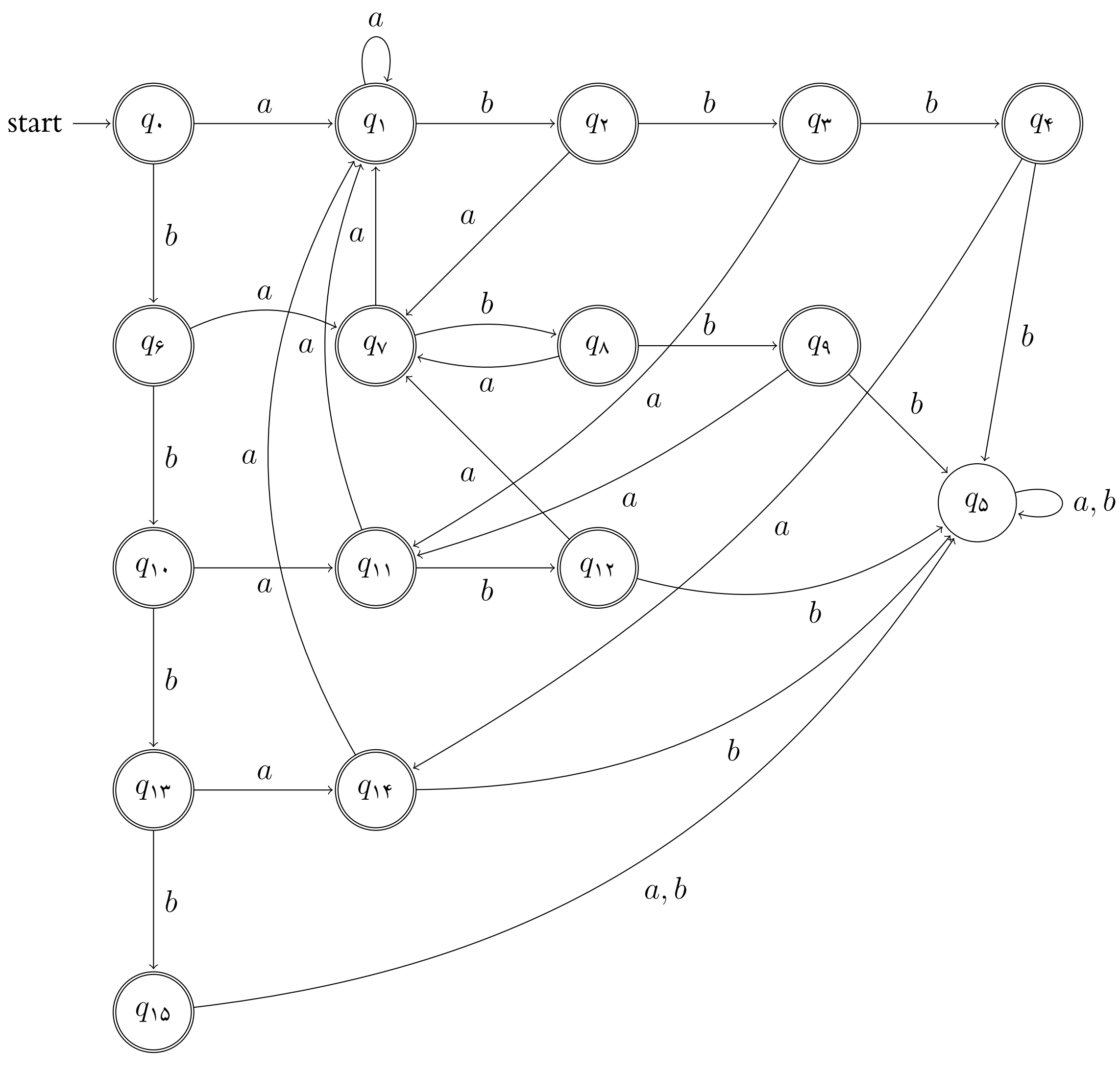
$$\overline{(a_{۱} \cdots a_{i+۱})_{۲}} \equiv ۲ \times \overline{(a_{۱} \cdots a_i)_{۲}} + a_{i+۱} \pmod{۳}$$



ث) این اتوماتا به‌صورتی طراحی شده‌است که هرگاه ۰۰ و ۱۱ هردو بیایند به استیت $q_{۷}$ می‌رسیم و در آنجا گیر می‌افتیم. در بقیه استیت‌ها نیز رشته قابل قبول است.



ج) اتوماتای ۱۶ استیتی زیر سعی می کند دنبال پترن‌های $bbbbb$، $abbbb$، $babbb$، $bbabb$، $bbbab$ و $bbbba$ بگردد و هرکدام را دید وارد یک استیت غیرنهایی می‌شود و دیگر رشته را قبول نمی‌کند. بقیه استیت‌ها نیز استیت اکسپت شوند.



۲.۱

الف) این اتوماتا به‌صورت ۵ تایی

$$(Q = \{۱, ۲, ۳, ۴, ۵\}, \Sigma = \{a, b\}, \delta, q_۰ = ۱, F = \{۵\})$$

قابل نمایش است که تابع δ توسط جدول زیر تعریف می‌شود.

	a	b
۱	۲	۱
۲	۳	۱
۳	۳	۴
۴	۵	۱
۵	۳	۱

این اتوماتا زبان تمام رشته‌هایی که با $aaba$ تمام می‌شوند را قبول می‌کند. بطور دقیقتر این اتوماتا همان اتوماتایی است که الگوریتم شناسایی رشته با استفاده از اتوماتای متناهی، برای رشته $aaba$ ایجاد می‌کند.

ب) این اتوماتا به‌صورت ۵ تایی

$$(Q = \{۱, ۲, ۳, ۴\}, \Sigma = \{a, b\}, \delta, q_۰ = ۱, F = \{۱, ۳\})$$

قابل نمایش است که تابع δ توسط جدول زیر تعریف می‌شود.

	a	b
۱	۲	۴
۲	۲	۳
۳	۲	۳
۴	۴	۴

این اتوماتا زبان تمام رشته‌هایی که با a شروع شده و با b تمام می‌شوند بعلاوه ϵ را قبول می‌کند.

۳.۱

مسئله را با استقرا روی طول رشته $\omega_۲$ حل می‌کنیم.

ابتدا به پایه استقرا می‌پردازیم که طول رشته $\omega_۲$ برابر ۱ است. به طور معادل می‌توان گفت این رشته یک کاراکتر مانند a است. پس باید رابطه زیر را ثابت کنیم. (توضیح مصحح: البته پایه استقرا باید برابر رشته تهی با طول صفر گرفته شود)

$$\delta^*(q, \omega_۱ a) = \delta(\delta^*(q, \omega_۱), a) = \delta^*(\delta^*(q, \omega_۱), a)$$

این رابطه هم واضح است زیرا اولی همان تعریف بازگشتی δ^* است و دومی نیز به این علت درست است که توابع δ و δ^* زمانی که ورودی دوم یک کاراکتر است یکسان رفتار می‌کنند.

حال به سراغ گام استقرا می‌رویم.

فرض می کنیم مسئله برای زمانی که طول رشته $\omega_۲$ برابر k است درست است و مسئله را برای رشته‌های با طول $k+۱$ ثابت می‌کنیم. چون طول $\omega_۲$ برابر $k+۱$ است پس می‌توان آنرا بصورت $\omega_۲ = \omega'_۲ a$ نوشت که a یک کاراکتر است و $\omega'_۲$ یک رشته با طول k است. حال طبق فرض استقرا و رابطه بازگشتی δ^* داریم

$$\delta^*(q,\omega_۱\omega_۲) = \delta(\delta^*(q,\omega_۱\omega'_۲),a) = \delta(\delta^*(\delta^*(q,\omega_۱),\omega'_۲),a) = \delta^*(\delta^*(q,\omega_۱),\omega'_۲ a) = \delta^*(\delta^*(q,\omega_۱),\omega_۲)$$

که همان حکم استقرا است. پس مسئله اثبات شد.

۴.۱

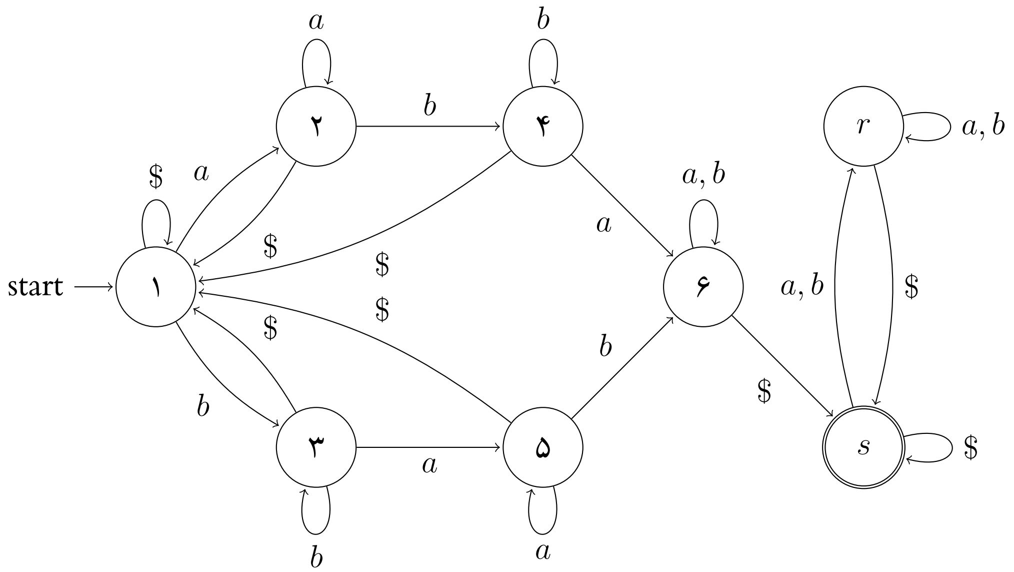
الف) ابتدا ساختار اتوماتا جدید را توضیح داده و سپس زبان اتوماتا جدید را توصیف می‌کنیم.

این اتوماتا از دو بخش تشکیل شده است که بخش اول شامل اتوماتا اولیه است و بخش دوم شامل دو استیت جدید r و s است. اگر در بخش اول باشیم حروف داخل σ بصورت قبلی عمل می‌کنند. ولی کاراکتر \$ وابسته به استیتی که هستیم مانند یک دکمه ریست عمل می‌کند و مارا به استیت اولیه می‌برد اگر در استیتی نهایی نباشیم و یا مارا به بخش دوم اتوماتا می‌برد اگر در استیتی نهایی باشیم. بخش دوم اتوماتا نیز دو حالت دارد که وقتی کاراکتر ورودی \$ باشد به استیت s می‌رود و در غیر اینصورت به استیت r می‌رود. لازم به ذکر است که پس از ورود به این قسمت هیچوقت از آن خارج نمی‌شویم.

پس می‌توان گفت اگر در رشته ω تمام کاراکترهای \$ را حذف کنیم و به چند بخش $\omega_۱, \cdots, \omega_k$ تقسیم شود حداقل یکی از این قسمت‌ها باید عضوی از $L(A)$ باشد تا به بخش دوم اتوماتا برسیم. همچنین برای اینکه در بخش دوم اتوماتا در استیت نهایی s باشیم این است که رشته ω با کاراکتر \$ تمام شود. پس می‌توان گفت زبان ماشین A' بصورت زیر است.

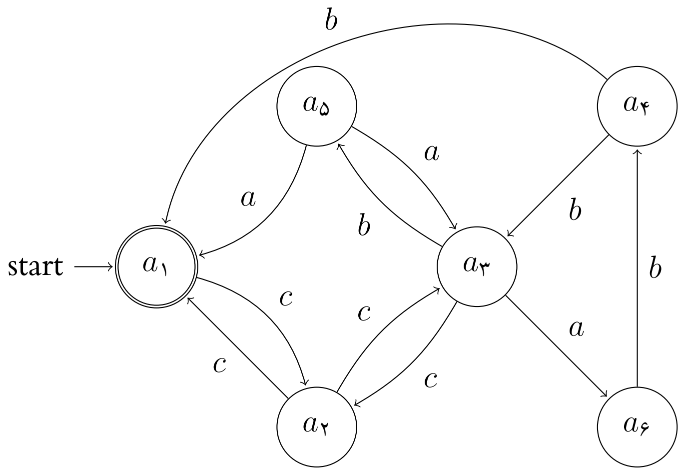
$$L(A') = (\{\epsilon\} \cup (\{a,b,\$\}^*\$))L(A)\$(\{\epsilon\} \cup (\{a,b,\$\}^*\$))$$

ب) دیاگرام ماشین A' بصورت زیر است.

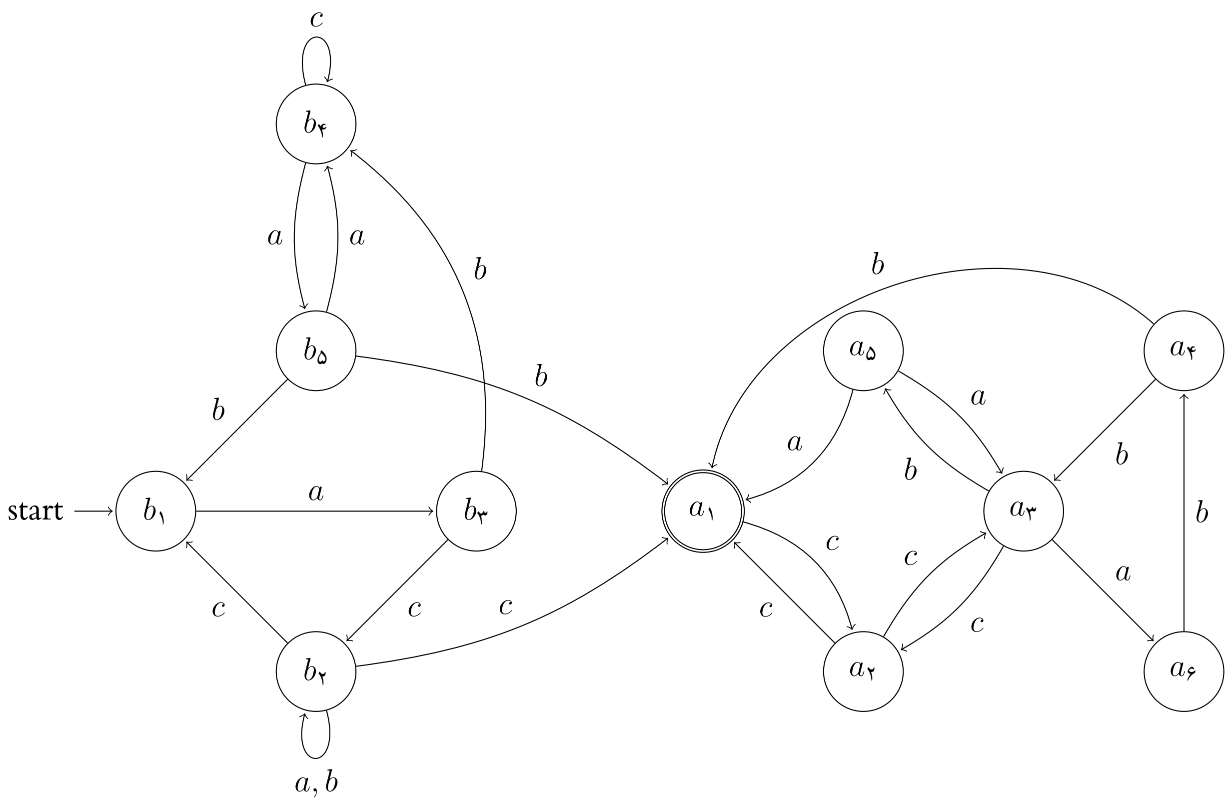


۵.۱

الف) تغییرات روی ماشین به این صورت است که اگر از یک استیت q با کاراکتر t به استیتی نهایی نرسیم با همین کاراکتر یک یال به استیت $a_۱$ می‌کشیم و استیت $a_۱$ را نیز تبدیل به تنها استیت نهایی می‌کنیم.

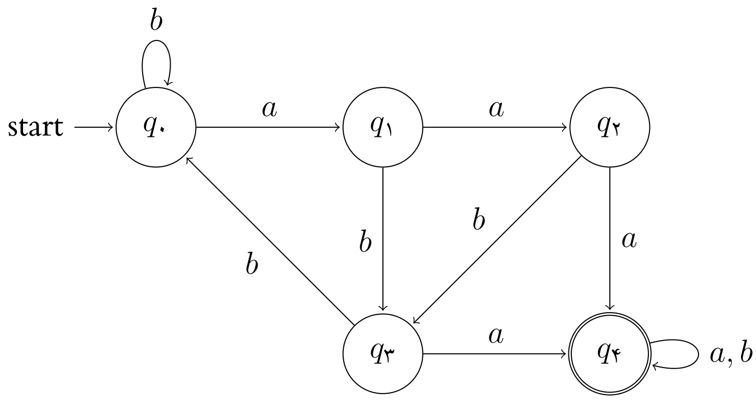


ب) دیاگرام این اتوماتا بصورت زیر است.

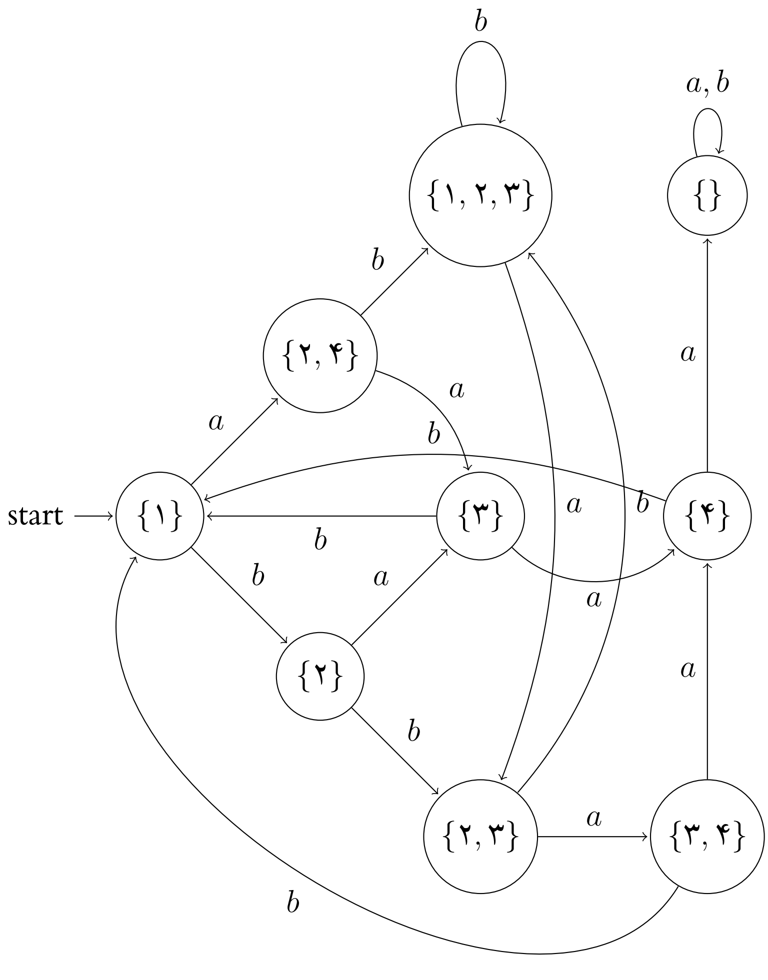


۲ هم‌ارزی ماشین‌های متناهی قطعی و غیرقطعی

الف) در این قسمت به‌جای ایجاد DFA ۱۶ استیتی و ساده‌سازی آن ابتدا زبان ماشین را فهمیده و DFA متناظرش را طراحی می‌کنیم. به‌راحتی داریم تنها رشته‌هایی که یکی از aaa یا aba را به‌عنوان زیررشته دارند قبول می‌شوند. تصویر ماشین متناظر با زبان این رشته‌ها در زیر آمده است.

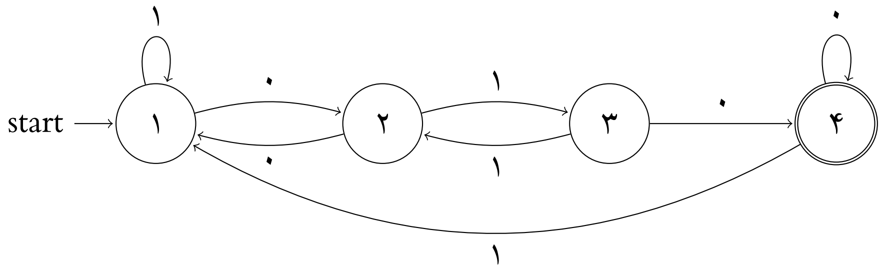


ب) در این مسئله در ابتدا تمام زیرمجموعه‌های ۴ استیت را گرفتیم و ماشین گفته شده در کلاس را ایجاد کردیم. سپس در این ماشین ۱۶ استیتی، استیت‌های غیر قابل دسترسی را حذف کردیم. در انتها نیز دو استیت یکسان مربوط به مجموعه‌های $\{۲, ۳\}$ و $\{۲, ۳, ۴\}$ را یکی کردیم و در انتها به ماشین ۹ استیتی زیر رسیدیم.

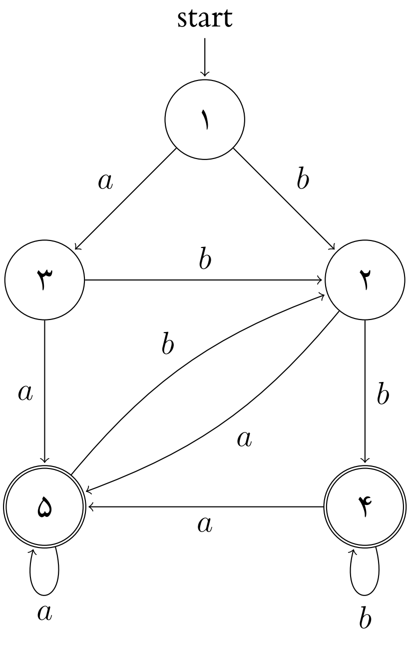


۳ کمینه‌سازی ماشین‌های متناهی قطعی

الف) ابتدا استیت‌های غیرقابل رسیدن که ۵، ۶، ۷ و ۸ بودند را حذف کردیم. سپس الگوریتم هاپکرافت را اجرا کردیم که استیت‌های معادلی وجود نداشتند. پس DFA نهایی شکل زیر می‌شود.



ب) در این قسمت در ابتدا دسترسی به استیت‌ها را بررسی کردیم که هیچ‌کدام غیر قابل دسترسی نبودند. سپس الگوریتم هاپکرافت را اجرا کردیم که دو دسته $\{۲, ۶\}$ و $\{۵, ۷\}$ بدست‌آمدند. اتوماتا جدید بدست‌آمده بصورت زیر است.



۴ خواص بستاری زبان‌های منظم

۱.۴

در ابتدا DFAهایی که زبان‌های آنها A و B هستند را در نظر بگیرید.

$$DFA_۱ = (Q_۱, \Sigma, \delta_۱, q_۱, F_۱), L(DFA_۱) = A$$

$$DFA_۲ = (Q_۲, \Sigma, \delta_۲, q_۲, F_۲), L(DFA_۲) = B$$

حال ماشین زیر را در نظر بگیرید.

$$DFA = (Q = Q_۱ \times Q_۲ \times \{۰, ۱\}, \Sigma, \delta, q = (q_۱, q_۲, ۰), F = F_۱ \times F_۲ \times \{۰\})$$

که تعریف δ بصورت زیر است.

$$\forall q' = (q'_۱, q'_۲, a) \in Q, b \in \Sigma: \quad \delta(q', b) = \begin{cases} (\delta_۱(q'_۱, b), q'_۲, ۱) & a = ۰ \\ (q'_۱, \delta_۲(q'_۲, b), ۰) & a = ۱ \end{cases}$$

ابتدا دقت کنید که با توجه به خصیصه سوم استیت‌ها در میابیم که با هر کاراکتر ورودی این خصیصه برعکس می‌شود. چون این خصیصه برای استیت ابتدایی و تمامی استیت‌های نهایی برابر ۰ است، پس هر رشته‌ای که در این ماشین قبول شود طول زوج دارد. حال فرض کنید رشته ω قبول شده‌است و آنرا بصورت زیر تقسیم بندی کنید.

$$\omega = a_۱ b_۱ \cdots a_k b_k \Rightarrow \omega_۱ = a_۱ \cdots a_k, \omega_۲ = b_۱ \cdots b_k$$

حال با توجه به خاصیت δ داریم. (توضیح مصحح: نیاز به اثبات یا به کارگیری نتیجه سوال ۳.۱)

$$\delta^*(q, \omega) = (\delta^*_۱(q_۱, \omega_۱), \delta^*_۲(q_۲, \omega_۲), ۰)$$

حال رشته ω قبول می‌شود اگر و فقط اگر

$$\delta^*_۱(q_۱, \omega_۱) \in F_۱, \delta^*_۲(q_۲, \omega_۲) \in F_۲$$

یا به طور معادل $\omega_۱ \in A$ و $\omega_۲ \in B$ که همان چیزی است که دنبالش بودیم.

۲.۴

در این مسئله در ابتدا برای هر کاراکتر $b \in \Sigma$ اتوماتایی که زبان آن $f(b)$ است را در نظر می‌گیریم.

$$DFA_b = (Q_b, \Delta, \delta_b, q_{۰,b}, F_b), L(DFA_b) = f(b)$$

همچنین یک زبان منظم L و اتوماتا قطعی که زبان آن L است را در نظر می‌گیریم.

$$DFA = (Q, \Sigma, \delta, q_۰, F), L(DFA) = L$$

حال اتوماتا غیر قطعی زیر را تعریف می‌کنیم که زبان تولید شده تحت عملیات جایگزینی با تابع f را قبول کند.

$$NFA = ((\{\epsilon\} \cup (\cup_{b \in Sigma} Q_b)) \times Q, \Delta \cup \{\epsilon\}, \delta', q_۰, \{\epsilon\} \times F)$$

همچنین تابع δ' بصورت زیر تعریف می‌شود.

$$\forall q' = (q_b, q) \in (\{\epsilon\} \cup (\cup_{b \in Sigma} Q_b)) \times Q, a \in \Delta \cup \{\epsilon\}: \quad \delta(q', a) = \begin{cases} \{(q_{۰,b}, q) | b \in \Sigma\} & q_b = \epsilon \wedge a = \epsilon \\ \{\} & q_b = \epsilon \wedge a \neq \epsilon \\ \{(\delta_b(q_b, a), q)\} & q_b \in Q_b \wedge a \neq \epsilon \\ \{(\epsilon, \delta(q', b))\} & q_b \in Q_b \wedge \\ & q_b \in F_b \wedge a = \epsilon \\ \{\} & q_b \in Q_b \wedge \\ & q_b \notin F_b \wedge a = \epsilon \end{cases}$$

حال کافیست بگوئیم NFA ارائه شده همان زبان حاصل جایگزینی را می‌دهد. دقت کنید این NFA به اینصورت طراحی شده‌است که برای هر یال داخل DFA با حرف b یک اتومات جایگرین DFA_b قرار داده شده‌است و از

استیت سر یال با ϵ به استیت اولیه DFA_b وصل شده است و از تمام استیت‌های نهایی DFA_b به استیت انتهای یال وصل شده است. این تغییر به این معنی است که وقتی در اتوماتا اصلی با حرف b از یال عبور می‌کردیم در اتوماتا جدید با رشته‌ای که توسط DFA_b قبول می‌شود از روی یال عبور می‌کنیم. به همین علت این اتوماتا زبان حاصل از جایگزینی را قبول می‌کند.