

## فصل ۲ ماشین‌های متناهی

در این فصل به ارائه ساده‌ترین مدل ماشین‌ها که دارای حافظه متناهی هستند و به نام ماشین متناهی شناخته می‌شوند می‌پردازیم. ماشین‌های متناهی را می‌توان به دو دسته ماشین‌های متناهی قطعی و غیر قطعی تقسیم‌بندی کرد که کاربرد بسیار مهمی در شناخت برخی از زبان‌ها که زبان‌های منظم نامیده می‌شوند، ایفا می‌کنند.

### پذیرنده متناهی قطعی (DFA) و زبان‌های منظم

چندتایی  $M = (Q, \Sigma, \delta, q_0, F)$  که به صورت زیر تعریف می‌شود یک DFA محسوب می‌شود.

مجموعه متناهی و غیر تهی از حالات  $Q$

الفبای ورودی ماشین  $\Sigma$

تابع تغییر وضعیت که به صورت زیر تعریف می‌شود  $\delta$

$\delta : Q \times \Sigma \rightarrow Q$

حالت اولیه ماشین  $q_0 \in Q$

مجموعه حالات نهایی ماشین  $F \subseteq Q$

ممکن است DFA را می‌توان توسط یک نمودار تغییر وضعیت نمایش داد که در آن حالات با رؤوس، تغییر وضعیت‌ها با یال‌ها، حالت اولیه با یک فلش کوچک و حالات نهایی به صورت دوایری دو خطی نمایش داده می‌شوند.

**مثال ۱:** نمودار تغییر وضعیت DFA زیر در شکل نشان داده شده است.

$$Q = \{q_0, q_1, q_2\}$$

$$\Sigma = \{a, b\}$$

$$\delta(q_0, a) = q_0$$

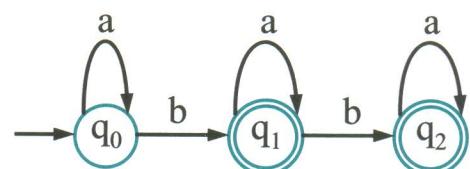
$$\delta(q_0, b) = q_1$$

$$\delta(q_1, a) = q_1$$

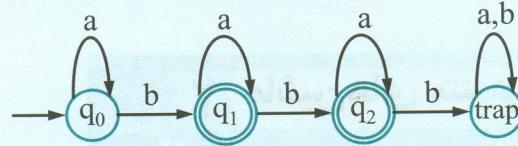
$$\delta(q_1, b) = q_2$$

$$\delta(q_2, a) = q_2$$

$$F = \{q_1, q_2\}$$



**نکته:** در ماشین‌های DFA در برخی حالتی که یال خروجی با برجسب یک حرف خاص از الفبا وجود ندارد، مرسوم است که آن یال را رسم نمی‌کنند و این بدين معنی است که آن حالت با آن ورودی به حالتی غیرنهایی که از آن مسیری به حالات نهایی وجود ندارد می‌رود. چنین حالتی با شرایط ذکر شده که معمولاً عدم رسم آن بلامانع است را حالت تله (Trap) می‌نامند. برای نمونه DFA مثال قبل را می‌توان به صورت زیر نیز رسم کرد.



تابع تغییر وضعیت توسعه‌یافته در یک DFA به صورت زیر تعریف می‌شود:

$$\delta^*: Q \times \Sigma^* \rightarrow Q$$

$$\left\{ \begin{array}{l} \delta^*(q, \lambda) = q; \forall q \in Q \\ \delta^*(q, aw) = \delta^*(\delta(q, a), w); \forall (a \in \Sigma, q \in Q, w \in \Sigma^*) \end{array} \right.$$

برای نمونه در DFA مثال قبل داریم:

$$\delta^*(q_0, aaba) = q_1$$

اگر یک DFA با دریافت یک رشته از ورودی پس از اتمام رشته در حالت نهایی متوقف شود، می‌گوییم آن رشته توسط DFA پذیرفته شده است و هر ماشین DFA یک مجموعه از رشته‌ها را می‌پذیرد که آن را زبان آن ماشین نامند. زبان پذیرفته شده توسط پذیرنده متناهی قطعی M را می‌توان به صورت زیر تعریف کرد:

$$L(M) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}$$

رشته‌هایی که یک DFA با خواندن آن‌ها در یک حالت غیرنهایی متوقف می‌شود متعلق به زبان  $L(M)$  نیستند؛ بنابراین:

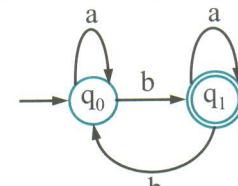
$$\overline{L(M)} = \{w \in \Sigma^* \mid \delta^*(q_0, w) \notin F\}$$

برای نمونه زبان پذیرفته شده توسط ماشین مثال قبل را می‌توان به صورت رشته‌هایی متشکل از a و b به قسمی که تعداد b‌های آن رشته برابر با ۱ یا ۲ باشد تعریف کرد. به بیان دیگر:

$$L(M) = \{w \in \Sigma^* \mid n_b(w) = 1 \text{ or } n_b(w) = 2\}$$

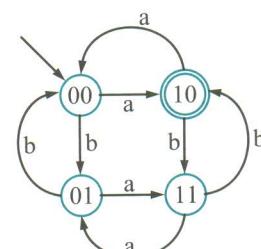
**مثال ۲:** زبان پذیرفته شده توسط DFA مقابل را می‌توان به صورت زیر تعریف کرد:

$$L(M) = \{w \in \Sigma^* \mid n_b(w) \bmod 2 = 1\}$$



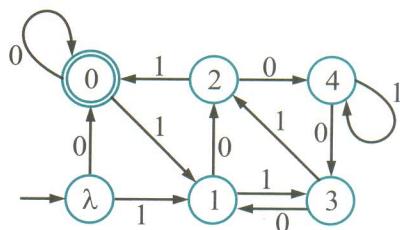
**مثال ۳:** زبان پذیرفته شده توسط DFA مقابل را می‌توان به صورت زیر تعریف کرد:

$$L(M) = \{w \in \Sigma^* \mid n_a(w) \bmod 2 = 1 \text{ and } n_b(w) \bmod 2 = 0\}$$



**مثال ۴:** برای طراحی یک پذیرنده متناهی که بتواند زبان  $L$  (رشته‌هایی متشکل از ارقام ۰ و ۱ که در مبنای ۲ دارای باقی‌مانده ۳ بر ۵ باشند) را پذیرد، باید به این موضوع توجه داشت که باقی‌مانده هر عدد باینری به ۵ یکی از پنج حالت ۰، ۱، ۲، ۳ و ۴ ممکن است باشد. همچنین با اضافه شدن یک ۰ در سمت راست هر عدد باینری مقدار آن دوباره و با اضافه شدن یک ۱ در سمت راست هر عدد باینری مقدار آن دوباره شده و با یک واحد جمع می‌شود؛ از این‌رو ماشین مذکور را می‌توان به صورت زیر ساخت:

$$L = \left\{ x \in \{0,1\}^* \mid x \bmod 5 = 3 \right\}$$



زبان منظم: زبان  $L$  منظم نامیده می‌شود اگر یک DFA مانند  $M$  وجود داشته باشد که  $L = L(M)$ .

### پذیرنده متناهی غیر قطعی (NFA)

چندتایی  $M = (Q, \Sigma, \delta, q_0, F)$  که به صورت زیر تعریف می‌شود یک NFA محسوب می‌شود.

مجموعه متناهی و غیر تهی از حالت  $Q$

الفبای ورودی ماشین  $\Sigma$

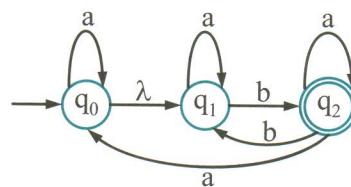
تابع تغییر وضعیت که به صورت زیر تعریف می‌شود

$$\delta: Q \times \Sigma \cup \{\lambda\} \rightarrow 2^Q$$

حالت اولیه ماشین  $q_0 \in Q$

مجموعه حالاتنهایی ماشین  $F \subseteq Q$

**مثال ۵:** ماشین زیر یک NFA را نشان می‌دهد:



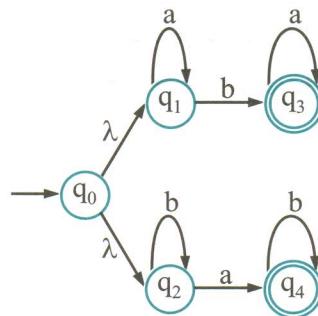
ملاحظه می‌شود که تعریف NFA مانند DFA است با این تفاوت که عملکرد آن غیر قطعی است. عدم قطعیت در این ماشین از دو روش ایجاد می‌شود:

● ورودی  $\lambda$ : در این حالت عدم قطعیت از آنجا حاصل می‌شود که ماشین می‌تواند در یک حالت بدون دریافت ورودی تغییر حالت دهد. برای نمونه در ماشین مثال قبلی، ماشین می‌تواند از حالت  $q_0$  بدون دریافت ورودی به حالت  $q_1$  تغییر حالت دهد؛ بنابراین می‌توان این‌گونه تصور کرد که ماشین وقتی در حالت  $q_0$  قرار دارد، در حالت  $q_1$  نیز قرار دارد.

● تغییر حالت از یک حالت خاص با یک ورودی خاص به بیش از یک حالت: در این حالت عدم قطعیت از آنجا حاصل می‌شود که ماشین می‌تواند با دریافت یک ورودی خاص به بیش از یک حالت تغییر حالت دهد. برای نمونه در ماشین مثال قبلی، ماشین می‌تواند در حالت  $q_2$  با دریافت ورودی  $a$  به هر دو حالت  $q_0$  و  $q_2$  برود.

مراحل طراحی یک NFA برای یک زبان خاص معمولاً ساده‌تر از طراحی DFA برای آن زبان است. برای مثال زبان NFA

- $L = \{a^m b a^n \mid m, n \geq 0\} \cup \{b^m a b^n \mid m, n \geq 0\}$  به صورت زیر است. در حالی که طراحی DFA آن پیچیده‌تر خواهد بود.



شکل ۱-۲

زبان پذیرفته شده توسط یک پذیرنده متناهی غیر قطعی  $M$  را می‌توان به صورت زیر تعریف کرد:

$$L(M) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \cap F \neq \emptyset\}$$

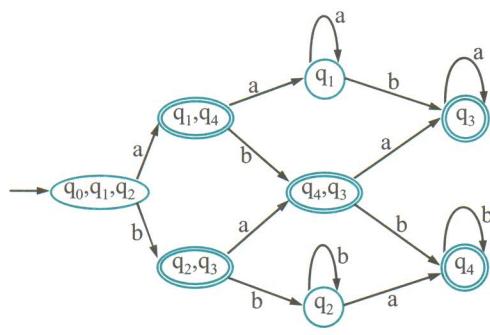
به بیان دیگر رشته‌هایی توسط یک NFA پذیرفته می‌شوند که ماشین با دریافت آن‌ها بتواند در یک حالت نهایی قرار گیرد. یک راه دیگر برای بیان این جمله این است که مسیری از حالت اولیه به یک حالت نهایی وجود داشته باشد که برچسب‌های این مسیر برابر با رشته ورودی باشد.

**نکته:** دو پذیرنده متناهی  $M_1$  و  $M_2$  معادل‌اند اگر و فقط اگر  $L(M_1) = L(M_2)$

### DFA به NFA تبدیل

از آنجاکه هر ماشین نشان‌دهنده یک الگوریتم است، هر NFA تبدیل به یک الگوریتم غیر قطعی خواهد شد و برای پیاده‌سازی باید به DFA تبدیل شود تا به راحتی به یک الگوریتم قطعی تبدیل شود. مراحل تبدیل یک NFA به DFA به این صورت

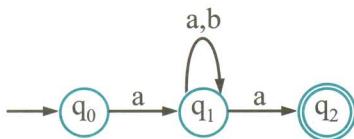
است که برای هر حالت مانند  $q$  در صورت وجود یال خروجی  $a$  برچسب  $\lambda$ ، آن را به همراه تمام حالتی در نظر می‌گیریم که از  $q$  به آن‌ها مسیری با برچسب  $\lambda$  وجود دارد و برای هر ورودی، که با آن ورودی تغییر حالتی به بیش از یک حالت وجود دارد، همه آن حالت را با هم به عنوان یک حالت در نظر می‌گیریم. هر حالتی در DFA حاصل شده که شامل حالت اولیه NFA اولیه باشد شامل حداقل یکی از حالت‌های NFA اولیه باشد حالت نهایی محاسبه شده و هر حالتی در DFA حاصل شده که شامل حداقل یکی از حالت‌های NFA اولیه باشد حالت نهایی محاسبه شود. برای نمونه DFA معادل با NFA مثال قبل به صورت رو به رو است:



شکل ۲-۲

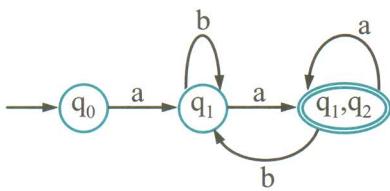
**نکته:** همان‌طور که در NFA اولیه حالت  $q_3$  با  $b$  و حالت  $q_4$  با  $a$  فقط به حالت trap می‌روند، در DFA

حاصل نیز نیازی به رسم این دو یال وجود ندارد.



**مثال ۶:** زبان NFA به صورت زیر است:

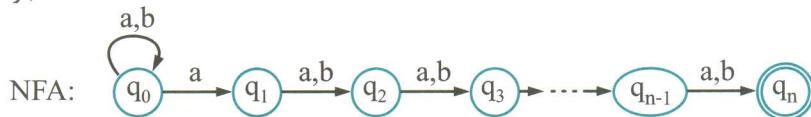
$$\left\{ awa \mid w \in \{a, b\}^* \right\}$$



در حالی که DFA آن به صورت زیر است:

**نکته:** در تبدیل یک DFA به NFA گاهی اوقات ممکن است NFA ای با n حالت به DFA ای با  $2^{n-1}$  حالت تبدیل شود.

$$(a \mid b)^* a \underbrace{(a \mid b) \cdot (a \mid b) \dots (a \mid b)}_{\text{n-1 بار}}$$



**مثال ۷:**

**نکته:** در تبدیل یک DFA به NFA گاهی اوقات ممکن است NFA ای با n حالت به DFA ای با یک حالت تبدیل شود.

### کاهش تعداد حالت DFA

در برخی از حالات ممکن است یک DFA به صورت غیر بهینه بیان شود. به بیان دیگر یک DFA مینیمال است اگر تعداد حالات آن کمینه باشد. برای تبدیل یک DFA به مینیمال ابتدا باید حالات ادغام‌پذیر آن را مشخص کرده و سپس با ادغام آنها با یکدیگر به DFA مینیمال رسید.

در یک پذیرنده متناهی قطعی دو حالت p و q ادغام‌پذیر نامیده می‌شوند اگر:

$$\exists w \in \Sigma^* : \delta(p, w) \in F \Rightarrow \delta(q, w) \notin F$$

و دو حالت p و q ادغام‌پذیر نامیده می‌شوند اگر:

$$\forall w \in \Sigma^* : \begin{cases} \delta(p, w) \in F \Rightarrow \delta(q, w) \in F \\ \quad \text{and} \\ \delta(p, w) \notin F \Rightarrow \delta(q, w) \notin F \end{cases}$$

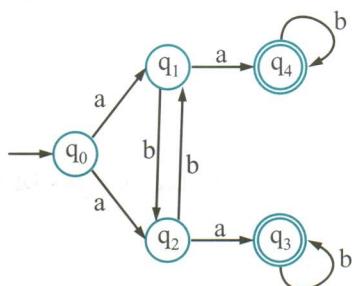
برای تشخیص حالات ادغام‌پذیر در یک DFA بر اساس الگوریتم زیر عمل می‌کنیم:

### الگوریتم کاهش حالت یک DFA

- ۱- کل حالات غیر نهایی ماشین را به عنوان یک گروه و کل حالات نهایی ماشین را به عنوان گروه دیگری فرض می‌کیم.
- ۲- دو حالت  $p$  و  $q$  از یک گروه یکسان در صورتی با هم ادغام‌ناپذیرند که برای یک حرف از الفبای ورودی مانند  $\delta(p, a) \in \sum$  و  $\delta(q, a) \in \sum$  در دو گروه متفاوت قرار داشته باشند. در چنین شرایطی دو حالت  $p$  و  $q$  باید از هم جدا شوند.
- ۳- تا زمانی که حالات ادغام‌ناپذیری وجود دارند، مرحله ۲ را اجرا کنید.

پس از اجرای الگوریتم بالا حالاتی که در یک گروه قرار گرفته‌اند، با هم ادغام‌پذیرند و باید ادغام شوند.

**مثال ۸:** در DFA زیر برای مشخص کردن حالات ادغام‌پذیر به صورت زیر عمل می‌کنیم:



حالات نهایی را در یک گروه و حالات غیر نهایی را در گروه دیگری قرار می‌دهیم:

$$G_1 = \{q_0, q_1, q_2\}$$

$$G_2 = \{q_3, q_4\}$$

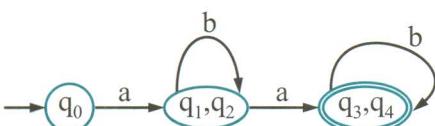
حال چون  $\delta(q_0, a) \in G_1$  و  $\delta(q_1, a) \notin G_1$  بنابراین دو حالت  $q_0$  و  $q_1$  ادغام‌ناپذیرند و همچنین چون  $\delta(q_0, a) \in G_1$  و  $\delta(q_2, a) \notin G_1$  بنابراین دو حالت  $q_0$  و  $q_2$  نیز ادغام‌ناپذیرند از این‌رو  $q_0$  از  $q_1$  و  $q_2$  جدا می‌شود؛ پس نتیجه می‌شود:

$$G_1 = \{q_0\}$$

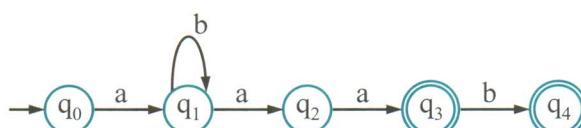
$$G_3 = \{q_1, q_2\}$$

$$G_2 = \{q_3, q_4\}$$

در این مرحله دیگر هیچ دو حالتی نمی‌تواند به عنوان ادغام‌ناپذیر مشخص شود؛ بنابراین DFA مینیمال به صورت زیر خواهد بود:



**مثال ۹:** در DFA زیر برای مشخص کردن حالات ادغام‌پذیر به صورت زیر عمل می‌کنیم:



حالات نهایی را در یک گروه و حالات غیر نهایی را در گروه دیگری قرار می‌دهیم:

$$G_1 = \{q_0, q_1, q_2\}$$

$$G_2 = \{q_3, q_4\}$$

حال چون  $\delta(q_4, b) \notin G_2$  و  $\delta(q_3, b) \in G_2$  بنابراین دو حالت  $q_3$  و  $q_4$  ادغام‌ناپذیرند و از هم جدا می‌شوند؛ پس نتیجه می‌شود:

$$G_1 = \{q_0, q_1, q_2\}$$

$$G_2 = \{q_3\}$$

$$G_3 = \{q_4\}$$

حال چون  $\delta(q_2, a) \in G_2$  و  $\delta(q_1, a) \notin G_2$  بنابراین دو حالت  $q_2$  و  $q_1$  ادغام‌ناپذیرند و همچنین چون  $\delta(q_2, a) \in G_2$  و  $\delta(q_0, a) \notin G_2$  بنابراین دو حالت  $q_2$  و  $q_0$  نیز ادغام‌ناپذیرند؛ از این‌رو  $q_2$  از  $q_1$  و  $q_0$  جدا می‌شود؛ پس نتیجه‌ای می‌شود:

$$G_1 = \{q_0, q_1\} \quad G_4 = \{q_2\} \quad G_2 = \{q_3\} \quad G_3 = \{q_4\}$$

حال چون  $\delta(q_1, a) \notin G_1$  و  $\delta(q_0, a) \in G_1$  بنابراین دو حالت  $q_0$  و  $q_1$  ادغام‌ناپذیرند؛ بنابراین از هم جدا می‌شوند؛ پس نتیجه می‌شود:

$$G_1 = \{q_0\} \quad G_5 = \{q_1\} \quad G_4 = \{q_2\} \quad G_2 = \{q_3\} \quad G_3 = \{q_4\}$$

درنهایت مشخص می‌شود که DFA اولیه در حالت مینیمال قرار داشته است و هیچ‌کدام از حالات با هم ادغام نمی‌شوند.

### نکات مربوط به ماشین‌های متناهی

۱- طراحی یک NFA برای یک زبان ساده‌تر از طراحی DFA برای آن زبان است و پیاده‌سازی DFA نیز ساده‌تر از پیاده‌سازی NFA است.

۲- در تبدیل یک DFA به NFA گاهی اوقات ممکن است  $n$  حالت به DFA ای با  $2^{n-1}$  حالت تبدیل شود.

۳- در NFA محدوده تابع  $\delta$ ، مجموعه توانی  $Q$  است؛ بنابراین مقدار آن یک عنصر از  $Q$  نیست بلکه زیرمجموعه‌ای از  $Q$  است.

۴- در DFA،  $\lambda$  به عنوان ورودی قابل قبول است؛ یعنی با  $\lambda$ ، یک تغییر وضعیت انجام می‌شود، در حالی که هیچ نمادی از ورودی خوانده نمی‌شود. این نشان‌دهنده آن است که اگرچه ورودی از چپ به راست خوانده می‌شود، اما مکانیزم خواندن می‌تواند متوقف باشد و وضعیت ماشین عوض شود.

۵- به ازای هر زبان منظم یک NFA وجود دارد که آن را می‌پذیرد.

۶- یک DFA به نام  $G_M$  گراف تغییر وضعیت مربوط به آن است. برای هر  $q_i, q_j \in Q$  داریم:  $\omega \in \Sigma^*$  اگر و فقط اگر بک راه با برچسب  $\omega$  از  $q_i$  به  $q_j$  وجود داشته باشد.

۷- در عین حال که زبان پذیرفته شده توسط یک DFA داده شده، یکتا است اما به طور معمول DFA‌های زیادی وجود دارند که یک زبان را می‌پذیرند.

۸- زبان پذیرفته شده توسط DFA منظم است.

۹- هر زبانی که توسط یک DFA پذیرفته می‌شود، منظم است.

۱۰- برای هر NFA یک DFA وجود دارد که همان زبان را می‌پذیرد.

۱۱- دو ماشین  $M_1, M_2$  در صورتی مساوی هستند که  $L(M_1) = L(M_2)$  یعنی اینکه هر دو زبان یکسانی تولید کنند.

۱۲- دو DFA ممکن است معادل باشند اما تعداد وضعیت‌های آن‌ها متفاوت باشد. اگرچه دو تا ماشین معادل هستند اما در مرحله پیاده‌سازی ممکن است که یکی را بردیگری ترجیح دهیم.

۱۳- اگر  $L$  زبان پذیرفته شده توسط DFA ای  $M_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$  باشد آن‌گاه یک DFA مانند:

$$L = L(M_D) = (Q_D, \Sigma, \delta_D, q_0, F_D)$$

۱۴- اگر  $L$  یک زبان غیر تهی باشد به طوری که حداقل طول رشته‌های  $\omega$  در  $L$  برابر با  $n$  باشد آن‌گاه هر DFA پذیرنده  $L$  حداقل دارای طول  $n+1$  حالت است.

به عنوان نمونه رشته  $\omega = aa$  را در نظر بگیرید:

$$|\omega| = n = 2$$

$$\text{تعداد حالت‌ها} = n + 1 = 3$$



۱۵- هر DFA یک NFA نیز هست اما عکس آن همیشه صادق نیست.

۱۶- گراف هر پذیرنده متناهی نامعین NFA را می‌توان به صورت یک گراف تغییر وضعیت عام نشان داد.

۱۷- اگر  $\delta_D^*(q_0, \omega)$  شامل  $q_f$  باشد آن گاه  $\delta_N^*(q_0, \omega)$  نیز شامل  $q_f$  است. به این معنا که اگر رشتہ  $\omega$  توسط DFA پذیرش شود، آن گاه توسط یک NFA معادل با این DFA نیز پذیرفته می‌شود.

۱۸- در آتماتای متناهی نامعین DFA اگر وضعیت ورودی و حافظه را بدانیم، می‌توانیم رفتار بعدی آتماتون را تعیین کنیم اما در آتماتای متناهی نامعین این گونه نیست. در هر زمان آتماتای نامعین می‌تواند رفتارهای متفاوتی داشته باشد.

۱۹- اگر  $G_M$  گراف DFA می‌باشد آن گاه داریم:

- اگر  $L(M)$  نامتناهی باشد، آن گاه مسیری از حالت ابتدایی به حالت نهایی وجود دارد که دارای چرخه است.

- اگر  $L(M)$  متناهی باشد، آن گاه چرخه‌ای با شرایط گفته شده وجود ندارد.

۲۰- اگر یک DFA با  $k$  حالت، رشتہ  $\omega$  با طول بیش از  $-1$  را پذیرد، آن گاه زبان پذیرفته شده توسط آن نامتناهی است.

۲۱- اگر یک DFA با  $k$  حالت، رشتہ‌ای را پذیرد، حتماً رشتہ‌ای با طول کوچکتر از  $k$  وجود دارد که DFA آن را می‌پذیرد.

۲۲- اگر یک DFA با  $k$  حالت، زبانی نامتناهی را پذیرد، رشتہ‌ای مانند  $\omega$  با طول  $|2k| \leq k$  را نیز خواهد پذیرفت.

۲۳- شرایط لازم و کافی برای آنکه یک DFA، زبان  $\Sigma^*$  را پذیرد، آن است که تمامی حالات دسترسی‌پذیر آن، حالت پایانی باشند. (حالتی دسترسی‌پذیر نامیده می‌شود که از حالت اولیه به آن مسیری وجود داشته باشد.)

۲۴- شرایط لازم و کافی برای آنکه یک DFA، زبان  $\emptyset$  را پذیرد، آن است که تمامی حالات دسترسی‌پذیر آن، حالت غیر پایانی باشند.

۲۵- شرایط لازم و کافی برای آنکه یک DFA، رشتہ پوج  $(\lambda)$  را پذیرد آن است که حالت شروع آن حالت پایانی باشد.

## نمونه سؤالات

۱. کدامیک از گزینه‌های زیر نادرست است؟

(۱) زبان پذیرفته شده توسط یک NFA به نام M عبارت است از:

$$L(M) = \left\{ \omega \in \Sigma^* : \delta^*(q_0, \omega) \cap F \neq \emptyset \right\}$$

(۲) مکمل زبان پذیرفته شده توسط یک NFA به نام M عبارت است از:

$$\overline{L(M)} = \left\{ \omega \in \Sigma^* : \delta^*(q_0, \omega) \cap (Q - F) \neq \emptyset \right\}$$

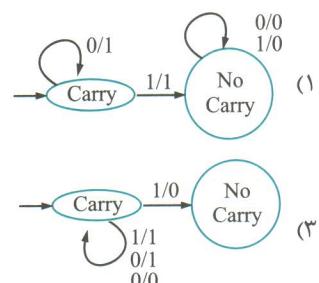
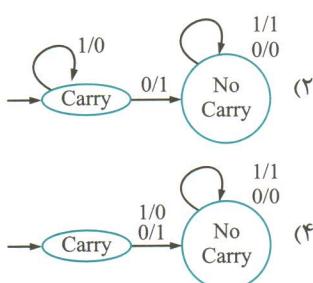
(۳) زبان پذیرفته شده توسط یک DFA به نام M عبارت است از:

$$L(M) = \left\{ \omega \in \Sigma^* : \delta^*(q_0, \omega) \in F \right\}$$

(۴) مکمل زبان پذیرفته شده توسط یک DFA به نام M عبارت است از:

$$\overline{L(M)} = \left\{ \omega \in \Sigma^* : \delta^*(q_0, \omega) \in (Q - F) \right\}$$

۲. کدامیک از ماشین‌های زیر ورودی را یک واحد افزایش می‌دهد؟ (ورودی را یک عدد باینری فرض کنید که از کم‌ارزش‌ترین بیت آن به سمت پارازش‌ترین بیت آن در ورودی ظاهر می‌شود).



۳. زبان پذیرفته شده توسط FA می‌قابل کدام است؟

(۱) تمام رشته‌هایی که دقیقاً یک ۱ دارند و یا شامل دنباله ۰۱ هستند.

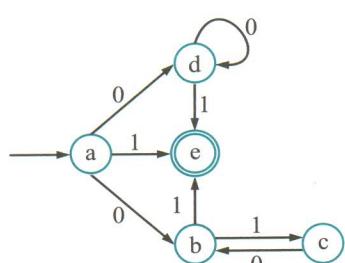
(۲) تمام رشته‌هایی که به نماد ۱ ختم می‌شوند و شامل دنباله ۰۱ هستند.

(۳) تمام رشته‌هایی که به یکی از دو صورت زیر هستند:

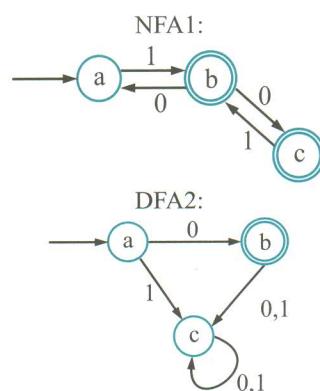
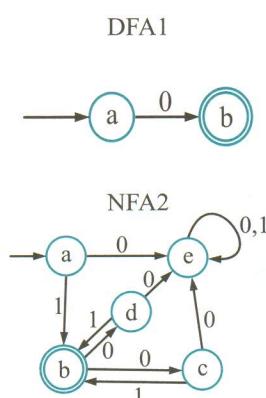
فقط یک ۱ در انتهای خود دارند.

به ۱ ختم شده و بعد از هر ۰ یک ۱ و بعد از هر ۱ یک ۰ دارند.

(۴) هیچ کدام.



۴. کدام یک از گزینه های زیر درست است؟ ( با فرض اینکه  $\Sigma = \{0,1\}$  ).



(۱) DFA1 و NFA1 معادل هستند.

(۲) DFA2 و NFA1 معادل هستند.

(۳) DFA1 و NFA2 معادل هستند.

(۴) هیچ کدام.

۵. زبان پذیرفته شده توسط DFA زیر کدام است؟

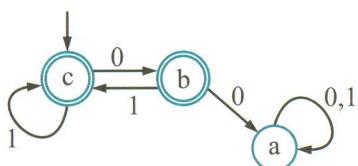
(۱) تمام رشته هایی که شامل 00 نیستند.

(۲) تمام رشته هایی که به 0 ختم شده و شامل 00 نیستند.

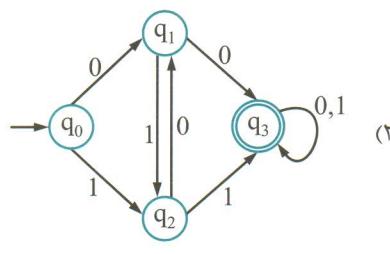
(۳) تمام رشته هایی که در آن بعد از هر زیر رشته 01 رشته 00 می آید.

(۴) هیچ کدام.

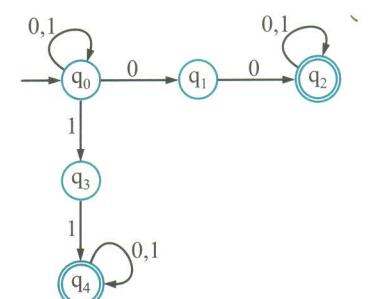
۶. کدام یک از ماشین های زیر معادل با زبان L هستند؟



$$L = \left\{ \omega \in (0|1)^* \mid w \text{ شامل } 00 \text{ یا } 11 \text{ است} \right\}$$

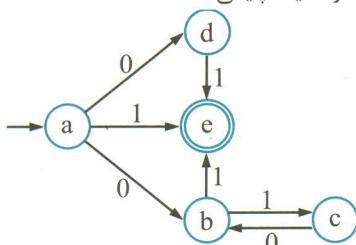


(۴) هیچ کدام



(۳) گزینه ۱ و ۲ درست است

۷. اگر DFA مقابل را به NFA معادل با آن تبدیل کنیم آن گاه حاصل دارای چند وضعیت پایانی است؟



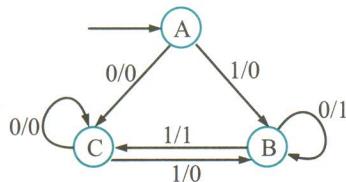
(۱) یک وضعیت پایانی

(۲) دو وضعیت پایانی

(۳) سه وضعیت پایانی

(۴) چهار وضعیت پایانی

۱۲. ماشین مبدل متناهی میلی (Finite-state transducer of Mealy sequential Machine) زیر را در نظر بگیرید:



۱) حالت A هم ارز حالت B است.

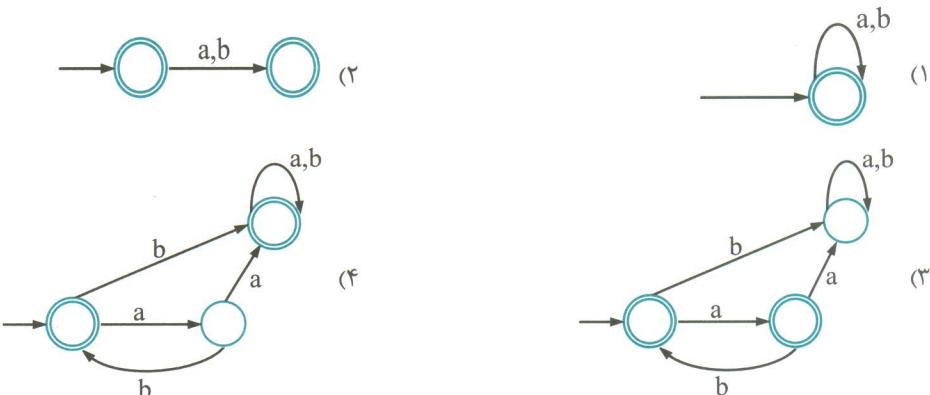
۲) حالت B هم ارز حالت C است.

۳) حالت A هم ارز حالت C است.

۴) هیچ کدام

۱۳. یک DFA برای زبان منظم  $L = (a^*(b \cup \{\lambda\})a^*)^*$  عبارت است از:

(علوم کامپیوتر ۸۵)



۱۴. اگر  $|w|_i$  تعداد حرف i در کلمه w باشد، آن گاه کدام گزاره در مورد زبان  $L = \{w \in \{0,1\}^* \mid |w|_0 = f(|w|_1)\}$  که در آن  $f(n) = n \bmod 5$  است، صحیح‌تر است؟

(علوم کامپیوتر ۸۵)

۱) مستقل از متن است.

۳) مستقل از متن نیست.

۲) متناهی است.

۴) وابسته به متن است ولی مستقل از متن نیست.

۱۵. زبانی است با الفبای  $\{0,1\}^* = \sum$  به قسمی که کلیه رشته‌های L دارای حداقل یک زیر رشته ۱۱ و فاقد زیر رشته ۰۰ هستند. کوچک‌ترین آتماتایی که این زبان را شناسایی کند دارای چند وضعیت (حالت) است؟

(دولتی مهندسی کامپیوتر ۸۸)

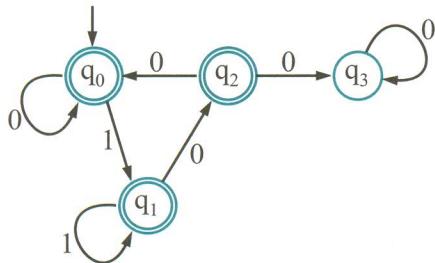
۱) ۵ وضعیت که دو وضعیت آن از نوع شناسایی است.

۳) ۵ وضعیت که سه وضعیت آن از نوع شناسایی است.

۲) ۶ وضعیت که دو وضعیت آن از نوع شناسایی است.

۴) ۶ وضعیت که سه وضعیت آن از نوع شناسایی است.

۸. ماشین متناهی مقابله، کدامیک از زبان‌های منظم زیر را می‌پذیرد؟



- (۱) تمام رشته‌هایی که شامل ۱۰۱ نباشند.
- (۲) تمام رشته‌هایی که شامل ۱۰۱ باشند.
- (۳) تمام رشته‌هایی که با ۰ یا ۱ شروع می‌شوند.
- (۴) تمام رشته‌هایی که به ۰۰ ختم می‌شوند.

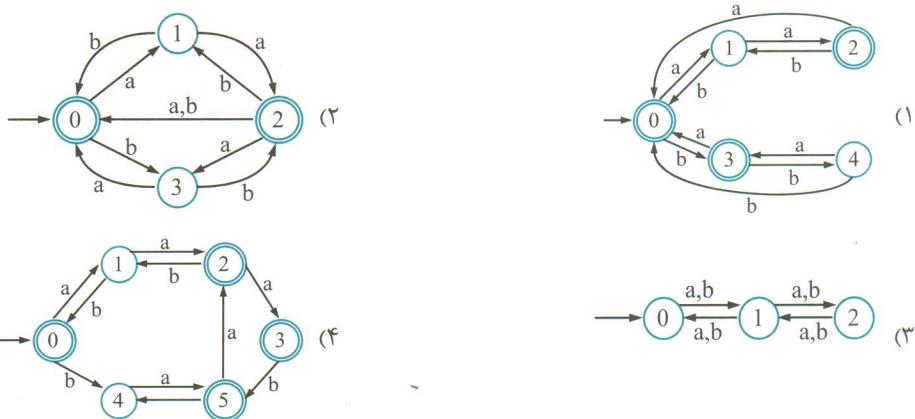
۹. اگر  $\Sigma = \{0, 1, \dots, n\}$  و  $L = \{w \mid \exists k \geq 0, |w| = 2k\} \subseteq \Sigma^*$  یک زبان با حروف  $\Sigma$  باشد، آن‌گاه تعداد حالت‌اتوماتون قطعی متناهی (DFA) مینیمال متناظر با زبان  $L$  ..... است.

(علوم کامپیوتر ۸۴)

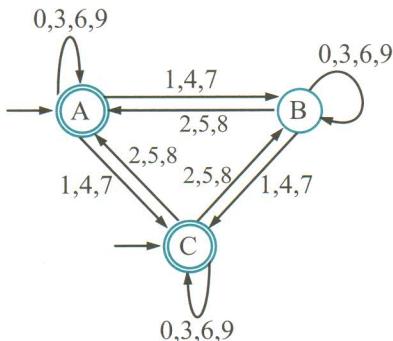
- الف)  $n$   
ب)  $2n - 1$   
ج)  $2n$   
د) عددی ثابت و مستقل از  $n$

۱۰. کدامیک از اutomات‌های متناهی زیر (Finite Automata) زبان  $L$  را شناسایی می‌کند؟ (منظور از  $(w)$  تعداد  $a$  های موجود در رشته  $w$  است).

$$L = \{w : N_a(w) - N_b(w) \bmod 3 \neq 1\}$$

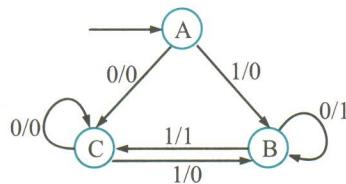


۱۱. ماشین پذیرنده متناهی (Finite State Automata) زیر را در نظر بگیرید. کدامیک از مجموعه رشته‌های زیر توسط این ماشین پذیرفته می‌شود؟



- (۱) رشته‌هایی از اعدادی که بر ۳ بخش‌پذیرند.
- (۲) رشته‌هایی از اعدادی که طولشان بر ۳ بخش‌پذیر است.
- (۳) رشته‌هایی از اعدادی که طولشان بر ۳ بخش‌پذیر است و با یکی از ارقام ۰, ۳, ۶, ۹ ختم شود.
- (۴) هیچ‌کدام

۱۲. ماشین مبدل متناهی میلی (Finite-state transducer of Mealy sequential Machine) زیر را در نظر بگیرید:



۱) حالت A هم‌ارز (Equivalent) حالت B است.

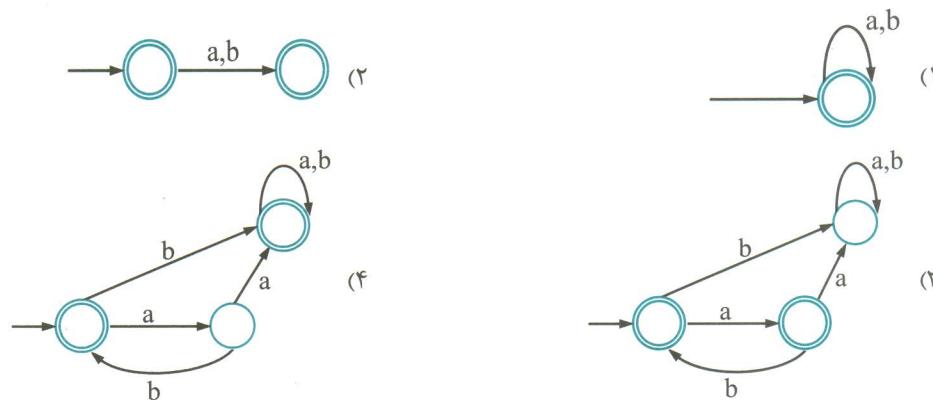
۲) حالت B هم‌ارز حالت C است.

۳) حالت A هم‌ارز حالت C است.

۴) هیچ کدام

۱۳. یک DFA برای زبان منظم  $L = (a^*(b \cup \{\lambda\})a^*)^*$  عبارت است از:

(علوم کامپیوتر ۸۵)



۱۴. اگر  $|w|_i$  تعداد حرف i در کلمه w باشد، آن‌گاه کدام گزاره در مورد زبان  $L = \{w \in \{0,1\}^* \mid |w|_0 = f(|w|_1)\}$  که در آن  $f(n) = n \bmod 5$  است، صحیح‌تر است؟

(علوم کامپیوتر ۸۵)

۱) مستقل از متن است.

۲) متناهی است.

۳) مستقل از متن نیست.

۱۵. زبانی است با الفبای  $\{0,1\}^*$  به قسمی که کلیه رشته‌های L دارای حداقل یک زیر رشته ۱۱ و فاقد زیر رشته ۰۰ هستند. کوچک‌ترین آتماتایی که این زبان را شناسایی کند دارای چند وضعیت (حالت) است؟

(دولتی مهندسی کامپیوتر ۸۸)

۱) ۵ وضعیت که دو وضعیت آن از نوع شناسایی است.

۲) ۶ وضعیت که سه وضعیت آن از نوع شناسایی است.

۳) ۶ وضعیت که دو وضعیت آن از نوع شناسایی است.

۴) ۶ وضعیت که سه وضعیت آن از نوع شناسایی است.

## حل تشریحی

۱. گزینه ۲ درست است.

در یک ماشین NFA ممکن است با یک رشته مسیری هم به حالت اولیه و هم به حالت نهایی وجود داشته باشد؛ بنابراین  $L(M)$  مجموعه رشته‌هایی است که توسط رشته‌های آن مسیری از حالت اولیه به نهایی وجود داشته باشد و  $\overline{L(M)}$  مجموعه رشته‌هایی است که توسط آن‌ها هیچ مسیری از حالت اولیه به نهایی وجود نداشته باشد که گزینه ۲ این را نشان نمی‌دهد؛ پس ۲ نادرست است.

۲. گزینه ۲ درست است.

برای افزایش یک عدد باینری به اندازه یک واحد باید آن را با یک بیت carry که مقدار اولیه ۱ دارد جمع کرد که این عمل به این معنی است که درصورتی که اولین بیت ورودی (سمت راست عدد) برابر با ۰ بود خروجی ۱ شده و بقیه بیت‌های ورودی بدون تغییر می‌مانند (carry=0) و درصورتی که اولین بیت ورودی (سمت راست عدد) برابر با ۱ بود خروجی ۰ شده و هنوز در حالتی قرار داریم که carry=1 است؛ بنابراین گزینه ۲ درست است.

۳. گزینه ۳ درست است.

ماشین رشته ۰۱۰ را نمی‌پذیرد؛ پس گزینه ۱ نادرست است. رشته ۰۱۱ را نمی‌پذیرد؛ پس گزینه ۲ نادرست است.

۴. گزینه ۳ درست است.

DFA1، DFA2 و همچنین NFA1، NFA2 معادل هستند. در DFA1 حالات a و c و d ادغام‌پذیرند.

۵. گزینه ۱ درست است.

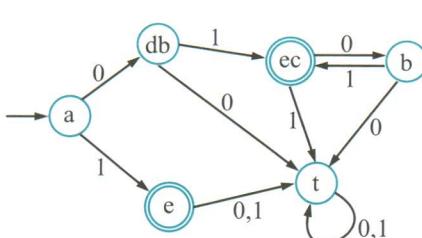
مشخص است که ماشین مورد نظر فقط با دیدن ۰۰ در هر جای رشته به حالت c رفته و دیگر از آن خارج نمی‌شود؛ بنابراین گزینه ۱ درست است.

۶. گزینه ۳ درست است.

ماشین الف یک NFA و ماشین ب یک DFA است که هر دو با دیدن رشته ۰۰ یا ۱۱ در هر جای ورودی به حالت نهایی رفته و از آن خارج نمی‌شوند؛ بنابراین گزینه ۳ درست است.

۷. گزینه ۲ درست است.

DFA معادل با NFA مذکور به این صورت است:



۸. گزینه ۱ درست است.

ماشین مذکور رشته ۱۰۱ را نمی پذیرد؛ ازین رو گزینه ۲ و ۳ نادرست است. ماشین رشته ۱۰۱۰۰ را نمی پذیرد؛ بنابراین گزینه ۴ نادرست است. گزینه ۱ درست است.

۹. گزینه ۴ درست است.

زبان مورد نظر رشته های را نشان می دهد که دارای طول زوج هستند؛ بنابراین ماشین متناهی مینیمال آن دارای دو حالت باید باشد که بتواند رشته های با طول زوج و فرد را از هم متمایز کند و گزینه ۴ درست است.

۱۰. گزینه ۱ درست است.

گزینه ۲ را نمی پذیرد؛ پس نادرست است. گزینه ۳ دارای حالت پایانی نیست و نادرست است. گزینه ۴ رشته bbb را نمی پذیرد؛ پس نادرست است.

۱۱. گزینه ۴ درست است.

رشته ۱۱۱ پذیرفته نمی شود؛ پس گزینه ۱ و ۲ نادرست هستند. رشته ۱۱۱۳۳۳ پذیرفته نمی شود؛ بنابراین گزینه ۳ نادرست است.

۱۲. گزینه ۳ درست است.

از آنجاکه حالت A و C هر دو با ورودی ۰ خروجی ۰ تولید کرده و به حالت C می روند و همچنین با ورودی ۱ خروجی ۰ تولید کرده و به حالت B می روند؛ بنابراین گزینه ۳ درست است.

۱۳. گزینه ۱ درست است.

عبارت منظم مورد نظر هر رشته ای را می تواند تولید کند و زبان آن برابر با  $\Sigma^*$  است، بنابراین گزینه ۱ صحیح است. بقیه ماشین ها چون دارای حالات غیر پایانی هستند؛ پس برخی از رشته ها را نمی پذیرند و بنابراین نادرست اند.

۱۴. گزینه ۱ درست است.

در زبان مورد نظر باید تعداد ۰ های موجود در رشته برابر با باقی مانده تعداد ۱ های رشته بر عدد ۵ باشد. چنین زبانی را می توان با یک ماشین DFA با ۵ حالت پذیرفت، پس زبان مورد نظر منظم است و گزینه ۱ درست است. از آنجاکه تعداد رشته های زبان مورد نظر نامتناهی است گزینه ۲ نادرست است. نیز چون زبان مورد نظر منظم است مستقل از متن نیز است؛ بنابراین گزینه های ۳ و ۴ نادرست اند.

۱۵. گزینه ۲ درست است.



ماشین مورد نظر به صورت رو به رو است؛ بنابراین گزینه ۲ درست است: