دانشکده‌ی علوم ریاضی

نظریه زبان‌ها و اتوماتا — ۷ آبان ۱۳۹۱

# جلسه‌ی ۱۳: ویژگی‌های بستاری زبان‌های منظم

*مدرّس: دکتر شهرام خزائی* — *نگارنده: کیارش رحمانی*

## ۱ نکات جلسه گذشته

ماشین حالت متناهی زیر را در نظر می‌گیریم:

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

با استفاده از الگوریتم‌های مطرح شده، $Q$ را به کلاس‌های هم‌ارزی ($S_i$)، افراز می‌کنیم. این کلاس‌ها، حالت‌های DFA کمینه معادل را تشکیل خواهندداد.

$$A_{eq} : (Q_{eq}, \Sigma, \delta_{eq}, q_{eq}, F_{eq})$$

- اگر به ازای یک $i$:  $q_\circ \in S_i \Leftarrow q_{eq} = S_i$
- $F_{eq}$ شامل $S_i$ هایی است که $S_i$ شامل یک حالت نهایی باشد. یعنی:  $F_{eq} = \{S_i | S_i \cap F \neq \varnothing\}$
- به ازای $S \in Q_{eq}$ تعریف می‌کنیم:  $\delta_{eq}(S, a) = T$ ، که $p$ یک عضو دلخواه $S$ است، و مجموعه $T$ شامل $\delta(p, a)$ است.

**قضیه ۱** *اگر قبل یا بعد از محاسبه DFA معادل، حالت‌های غیرقابل دسترسی را حذف کنیم، حاصل، DFA کمینه خواهدبود.*

## ۲ ویژگی‌های بستاری زبان‌های منظم

می‌خواهیم منظم بودن زبان‌های منظم را بعد از انجام اعمال مختلف روی آن‌ها نشان دهیم.

### ۱.۲ اجتماع

**قضیه ۲** *اگر زبان‌های $M$ و $N$ منظم باشند، آنگاه، $M \cup N$ منظم است.*

**برهان.** فرض می‌کنیم ماشین حالت متناهی (DFA) های $A_M$ و $A_N$ را که به ترتیب زبان‌های $M$ و $N$ را می‌پذیرند، داریم.

$$A_M = (Q_M, \Sigma, \delta_M, q_M, F_M)$$

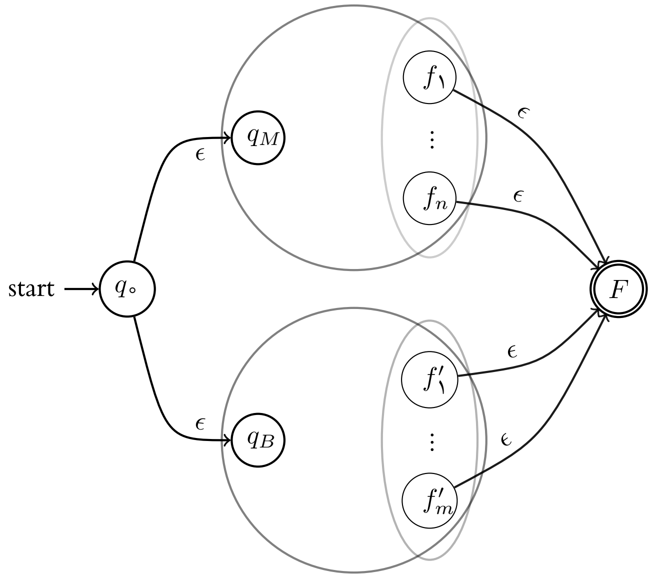
$$A_N = (Q_N, \Sigma, \delta_N, q_N, F_N)$$

با استفاده از آنها، DFA زیر را می‌سازیم. نشان می‌دهیم که این DFA اجتماع $M$ و $N$ را می‌پذیرد.

$$A_{M\cup N} = (Q, \Sigma, \delta, q_\circ, F)$$

- $Q = Q_M \cup Q_N \cup \{q_\circ, F\}$
- $\delta$ شامل همه‌ی قوانین $\delta_M$ و $\delta_N$ به علاوه‌ی قوانین زیر است:

$$\delta(q_\circ, \epsilon) = q_M$$
$$\delta(q_\circ, \epsilon) = q_N$$
$$\delta(f_i, \epsilon) = F \text{ ، } i \leq n$$
$$\delta(f'_j, \epsilon) = F \text{ ، } j \leq m$$




از حالت شروع، بدون خواندن ورودی با یال‌های $\epsilon$ به حالت‌های شروع ماشین‌های حالت متناهی $A_M$ و $A_N$ می‌رویم. و تنها راه رسیدن به حالت پایانی، رسیدن به حداقل یکی از حالت‌های نهایی $A_M$ یا $A_N$ است. پس فقط در صورتی که رشته‌ی ورودی از حالت ابتدایی یکی از DFA ها (یا هردو) به حالت نهایی آن ماشین برسد، رشته قابل قبول خواهد بود. واین یعنی اجتماع دو زبان $M$ و $N$ زبانِ اوتوماتای ساخته‌شده است.

■

## ۲.۲ الحاق

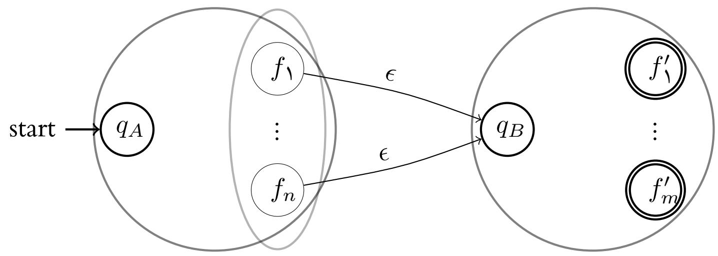
**قضیه ۳** *اگر زبان‌های $M$ و $N$ منظم باشند، آنگاه، $MN$ منظم است.*

**برهان.** برای این قسمت هم مانند بخش قبل، فرض می‌کنیم ماشین‌های $A_M$ و $A_N$ را داریم که به ترتیب زبان‌های منظم $M$ و $N$ را می‌پذیرند.

$$A_M = (Q_M, \Sigma, \delta_M, q_M, F_M)$$

$$A_N = (Q_N, \Sigma, \delta_N, q_N, F_N)$$

یک ماشین حالت متناهی تعریف‌می‌کنیم و نشان می‌دهیم که زبانی که می‌پذیرد، $MN$ است. و می‌دانیم که هر ماشین حالت متناهی، نمادی از یک زبان منظّم است، پس $MN$ یک زبان منظم است.




$$A_{M\cup N} = (Q, \Sigma, \delta, q_A, F_N)$$

- $Q = Q_M \cup Q_N$
- $\delta$ شامل همه‌ی قوانین $\delta_M$ و $\delta_N$ به علاوه‌ی قوانین زیر است:
  $\delta(f_i, \epsilon) = q_B$ ، $i \leq n$

■

## ۳.۲ اشتراک

**قضیه ۴** *اگر زبان‌های $M$ و $N$ منظم باشند، آنگاه، $M \cap N$ منظم است.*

**برهان.** می‌دانیم ماشین‌های حالت متناهی A و B وجود دارند که به ترتیب زبان‌های $M$ و $N$ را می‌پذیرند. (چون برای هر زبان منظم، یک DFA وجود دارد.)
ماشین حالت متناهی $A \times B$ را به صورت زیر تعریف می‌کنیم:

$$A = (Q_M, \Sigma, \delta_M, q_M, F_M)$$

$$B = (Q_N, \Sigma, \delta_N, q_N, F_N)$$

$$A \times B = (Q_M \times Q_N, \Sigma, \delta, [q_M, q_N], F_M \times F_N)$$

که $\delta$ را به صورت زیر تعریف می‌کنیم:

$$\delta([p, q], a) = [\delta_M(p, a), \delta_N(q, a)]$$

می‌توان به استقرا اثبات کرد که ماشین حالت متناهی $A \times B$ ، زبان $N \cap M$ را می‌پذیرد.

■

## ۴.۲ $\overline{M}$ یا $(\Sigma^* - M)$

**قضیه ۵** *اگر زبان $M$ ، منظم باشد، آنگاه، مکمّل آن، $\overline{M}$ ، منظّم است.*

**برهان.** فرض می‌کنیم ماشین حالت متناهی $A = (Q, \Sigma, \delta, q, F_A)$ زبان $M$ را می‌پذیرد. ماشین حالت متناهی $A'$ را تعریف می‌کنیم به صورتی که همه‌ی حالت‌های نهاییِ $A$، تشکیل حالت‌های غیرنهایی آن را بدهند و حالت‌های غیرنهایی $A$ ، حالت‌های نهاییِ آن باشند. بدین ترتیب هر رشته‌ای که در $A$ پذیرفته می‌شود (به یک حالت نهایی می‌رسد)، در $A'$ به یک حالت غیرنهایی می‌رسد و پذیرفته نمی‌شود. پس زبانی که $A'$ می‌پذیرد $\overline{M}$ خواهد بود.

$$A' = (Q, \Sigma, \delta, q, Q - F_A)$$

بدین ترتیب منظم بودن $\overline{M}$ را با تشکیل DFA آن اثبات کرده‌ایم.

■

**نکته ۱** *اکنون می‌توانیم اثبات دیگری برای منظم بودن $N \cap M$ ارائه دهیم:* $N \cap M = \overline{\overline{N} \cup \overline{M}}$
*با توجه به منظم بودن هر قسمت از سمت راست تساوی به منظم بودن سمت چپ پی می‌بریم.*

## ۵.۲ $N - M$

**قضیه ۶** *اگر زبان‌های $M$ و $N$ منظّم باشند، آنگاه، اختلاف آن‌ها، $N - M$ ، منظّم است.*

**برهان.** از روشی مشابه اثبات منظم بودن $(N \cap M)$ استفاده می‌کنیم:

$$A = (Q_M, \Sigma, , \delta_M, q_M, F_M)$$
$$B = (Q_N, \Sigma, , \delta_N, q_N, F_N)$$

ماشین‌های حالت متناهی $A$ و $B$ را تعریف کردیم که به ترتیب زبان‌های $M$ و $N$ را می‌پذیرند.
$A \times B$ را تعریف می‌کنیم:

$$A \times B = (Q_M \times Q_N, \Sigma, \delta, [q_M, q_N], F)$$

که $F$ به صورت زیر تعریف می‌شود:

$$\forall p \in Q_M, q \in Q_N$$
$$[p, q] \in F \Leftrightarrow p \in F_M \wedge q \notin F_N$$

■

## ۶.۲ $M^R$

**تعریف ۱** *معکوس[۱] یک رشته، رشته‌ای است که همه کاراکترهای آن از انتها به ابتدا برعکس نوشته‌شده باشند.*

**تعریف ۲** *معکوس یک زبان،مثل $M$ ، زبانی است که شامل معکوسِ همه رشته‌های آن زبان باشد، که آن را با $M^R$ نشان می‌دهیم.*

**قضیه ۷** *زبان‌های منظّم نسبت به عملِ معکوس‌گیری بسته‌اند. یعنی اگر زبان $M$ منظم باشد، زبان $M^R$ هم منظّم است.*

فرض می‌کنیم ماشین حالت متناهی $A$ را داریم که زبان $M$ را می‌پذیرد. به صورت شهودی یک DFA می‌سازیم که زبان $M^R$ را بپذیرد.

۱. جهت همه‌ی یال‌های $A$ را برعکس می‌کنیم.

۲. حالت شروع $A$ را تبدیل به تنها حالت نهایی در ماشینِ جدید می‌کنیم.

۳. یک حالت شروع به DFA اضافه می‌کنیم و آن را با یال‌های $\epsilon$ به حالت‌هایی که در $A$ نهایی بودند وصل می‌کنیم.

ماشین حالت متناهی حاصل، زبان $M^R$ را می‌پذیرد.

---

[۱] reversal

## ۳ یک‌ریختی

یک‌ریختی[۲]، تابعی است که به هرکدام از نمادهای الفبا، یک رشته‌ی مشخص را نسبت می‌دهد.

**تعریف ۳** *تابع $h: \Sigma_1^* \to \Sigma_2^*$ یک یک‌ریختی نامیده می‌شود، اگر برای هر $w_1, w_2 \in \Sigma_1^*$ داشته‌باشیم:*

$$h(w_1 w_2) = h(w_1)h(w_2)$$

**لم ۸** *اگر $h$، یک‌ریختی باشد، آنگاه: $h(\epsilon) = \epsilon$*

**لم ۹** *برای رشته‌ی $\omega = a_1 a_2 a_3 ... a_n$، و تابع یک‌ریختیِ $h$، داریم:*

$$h(\omega) = h(a_1)h(a_2)....h(a_n)$$

**نتیجه ۱** *اگر $h: \Sigma_1^* \to \Sigma_2^*$ یک یک‌ریختی باشد، تحدید $h$ به $\Sigma_1$، یک‌ریختی $h$ را به طور کامل توصیف می‌کند. به عبارت دیگر، کافی‌است مقادیر $h(a)$ را به ازای تمام $a \in \Sigma_1$ مشخص باشد. بنابراین، برای به دست آوردن حاصل یک تابع یک‌ریختی روی یک رشته، تابع را روی تک‌تک حرف‌های رشته اعمال می‌کنیم و در انتها، آنها را به هم الحاق می‌کنیم.*

**مثال ۱** *تابع یک‌ریختی $h$ را به صورت زیر تعریف می‌کنیم و آن را روی رشته‌ی ۰۱۱۰۱ اعمال می‌کنیم:*

$$\begin{cases} h(0) = \epsilon \\ h(1) = ab \end{cases}$$

پس

$$h(01101) = h(0)h(1)h(1)h(0)h(1) = ababab$$

**تعریف ۴** *اگر $L$ یک زبان روی $\Sigma_1$ باشد، و $h: \Sigma_1^* \to \Sigma_2^*$ یک یک‌ریختی باشد، $h(L)$ یک زبان روی $\Sigma_2$ است و به صورت زیر تعریف می‌شود:*

$$h(L) = \{h(\omega) | \forall \omega \in L\}$$

**قضیه ۱۰** *اگر $L$ یک زبان منظّم باشد، و $h$ یک تابع یک‌ریختی، آنگاه $h(L)$ یک زبان منظّم است.*

**مثال ۲**

$$\begin{cases} E = 01^* + 10^* \\ L(E) = L \end{cases} \quad \Rightarrow \quad h(L) = (\epsilon(ab)^* + (ab)\epsilon^*) = (ab)^*$$

*تابع $h$ را روی زبان منظّم $L$ اعمال کردیم، و حاصل یک زبان منظّم شد.*

## ۴ یک‌ریختی معکوس

اگر $h: \Sigma_1^* \to \Sigma_2^*$ یک یک‌ریختی باشد، تابع معکوس آن، $h^{-1}$ را، به صورت زیر تعریف می‌کنیم و آن را یک یک‌ریختی معکوس[۳] می‌نامیم.
فرض کنیم $L$ یک زبان روی $\Sigma_2^*$ باشد، آنگاه:

$$h^{-1} = \{w | \forall w \in \Sigma_1^*, h(w) \in L\}$$

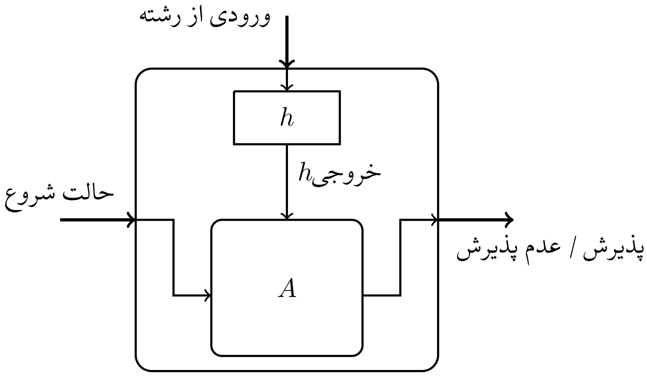
[۲] homorphism
[۳] inverse homorphism

**قضیه ۱۱** *اگر $h : \Sigma_1^* \to \Sigma_2^*$ یک یک‌ریختی باشد، و $L$ یک زبان منظّم روی الفبای $\Sigma_2^*$ باشد، آنگاه، $h^{-1}(L)$ یک زبان منظّم روی الفبای $\Sigma_1$ است.*

**برهان.** با ساختن یک ماشین حالت متناهی برای زبان $h^{-1}(L)$ قضیه را اثبات می‌کنیم.
فرض می‌کنیم ماشین $A = (Q, \Sigma_2, \delta, q_0, F)$ را داریم که زبان $L$ را می‌پذیرد. ماشین حالت متناهی $A'$ را به صورت زیر تعریف می‌کنیم:

$$A' = (Q, \Sigma_2, \gamma, q_0, F)$$

که $\gamma$ به صورت روبرو تعریف می‌شود: $\gamma(q, a) = \hat{\delta}(q, h(a))$
یعنی ماشینی که زبان $h^{-1}(L)$ را می‌پذیرد همان اتوماتایی است که زبان $L$ را می‌پذیرد با این تفاوت که روی هر کاراکتر، قبل از آن که توسط ماشین خوانده‌شود، تابع $h$ را اعمال می‌کنیم.




$A'$

■