Chapter 4

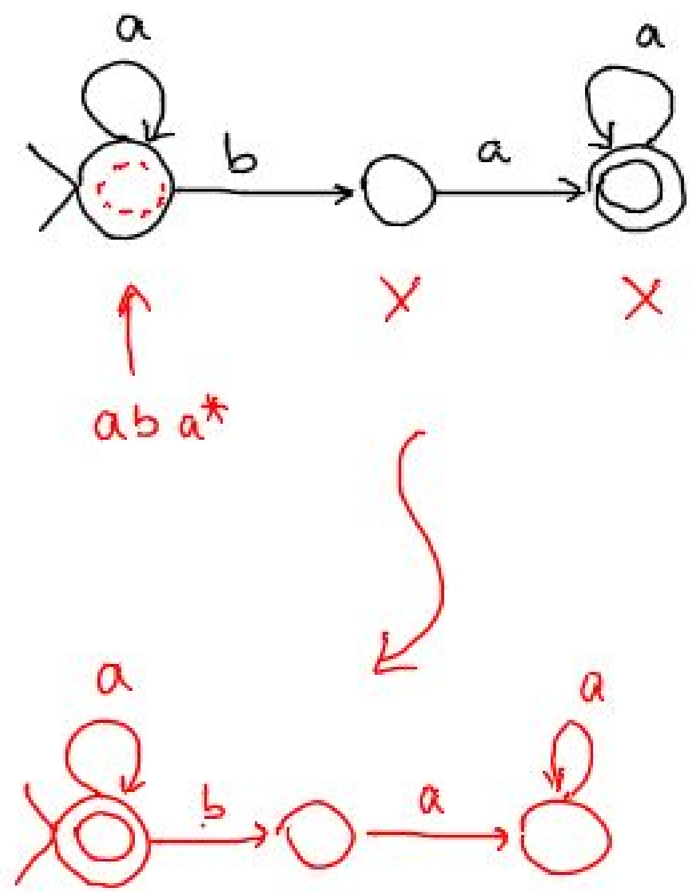
1) $L_1 = L(a^* b a a^*)$

$L_2 = L(a b a^*)$

L_1 / L_2

$= \{a\}^*$

$= L(a^*)$



۲) $L_1 = L_1 L_2 / L_2$ X

$$L_1 = \{a\}\{a\}^* = \{a\}^+$$

$$L_2 = \{a\}^*$$

$$L_1 L_2 = \{a\}^+$$

$$L_1 L_2 / L_2 = \{a\}^+ / \{a\}^*$$

$$= \{a\}^* \neq L_1$$

۳) $L_2 \backslash L_1 = \{ y : \exists x \in L_2 , xy \in L_1 \}$

L_1, L_2 are regular $\Longrightarrow L_2 \backslash L_1$ is also regular

می‌توان برای $L_2 \backslash L_1$ از روی DFA ی L_1 یک NFA برای $L_2 \backslash L_1$ ارائه داد:

از روی DFA ی L_1 یک کپی تهیه می‌کنیم. برای هر حالت $q \in Q$ بررسی می‌کنیم که آیا از حالت شروع به آن حالت مسیری با برچسب $v \in L_2$ وجود دارد یا خیر. اگر این مسیر وجود داشت، هر رشته‌ای که با $\delta(q, x) = q_f$ در $L_2 \backslash L_1$ خواهد بود. DFA را به گونه‌ای تغییر می‌دهیم که q حالت شروع باشد.

بنابراین اگر L_1 و L_2 منظم باشند، $L_2 \backslash L_1$ (یا $L_1 \backslash L_2$) نیز منظم خواهد بود.

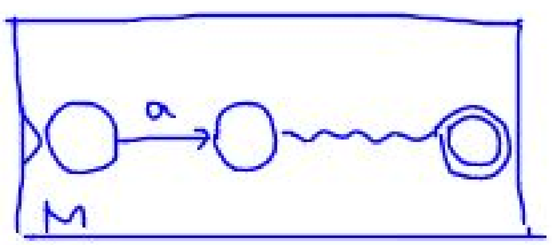
f)

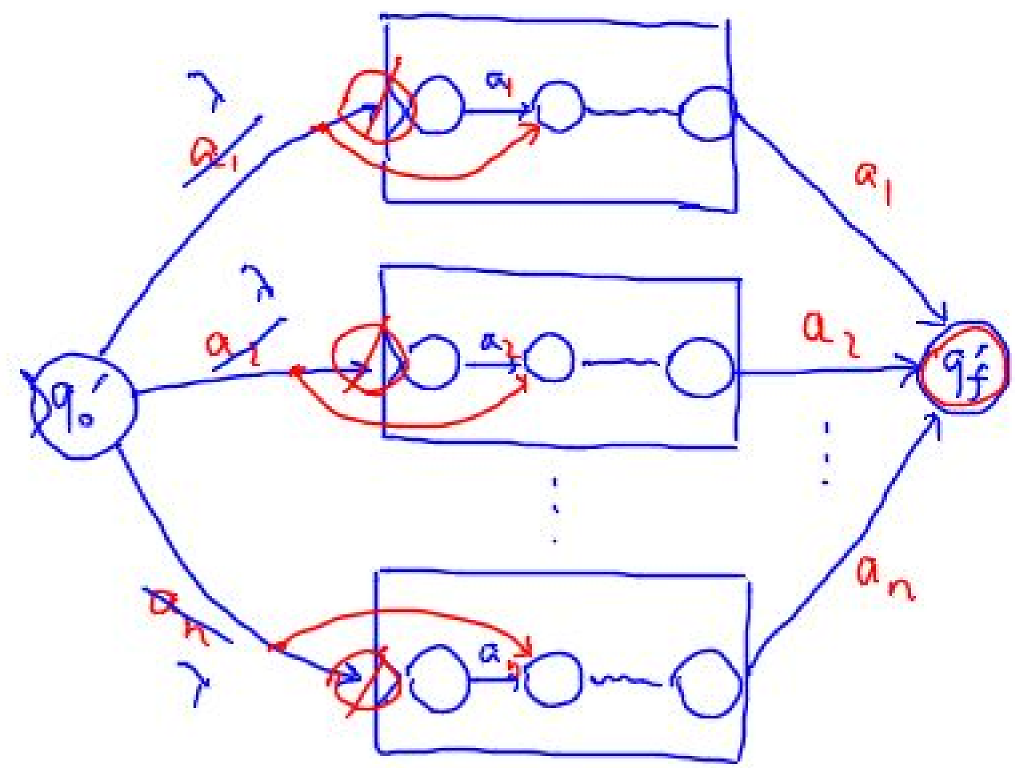
$$\text{shift}(a_1 a_2 \dots a_n) = a_2 a_3 \dots a_n a_1$$

$$\text{shift}(L) = \{v : v = \text{shift}(w),\ w \in L\}$$

L is regular $\Longrightarrow$ shift(L) is also regular

از آنجا که L منظم است برای آن DFA ای M وجود دارد. از این ماشین به اندازه‌ی $|\Sigma|$ کپی تهیه می‌کنیم.
با اضافه کردن یک حالت آغازین جدید برای هر $a \in \Sigma$ گذر $\delta(q_0', \overset{\lambda}{\cancel{a}}) = q_a'$ را به NFA ای جدید اضافه می‌کنیم که در آن q_a' جایی است که در DFA از q_0 با a به آن می‌رویم.
حال از حالت نهایی هر ماشین با همان حرف الفبا وارد حالت نهایی جدید q_f' می‌شویم.





زبان shift توسط این NFA پذیرفته می‌شود
پس منظم است.

۵) $\text{leftside}(L) = \{v : vv^R \in L\}$

بله، بسته است.

از روی ماشین DFA پذیرنده‌ی زبان L، یک FA برای پذیرش زبان فوق ارائه می‌دهیم، که به صورت زیر عمل می‌کند:

(۱) حالت نهایی $q \in F$ و یک y را حدس می‌زنیم به طوری که $y = v^R$ و $x = v$ در رشته‌ی ورودی $w = xy$.

(۲) توسط FA، پذیرنده‌ی زبان L را به صورت زیر شبیه‌سازی می‌کنیم: - حرکت روی x از q_0 به سمت جلو

- حرکت روی y از q به سمت عقب

(۳) حالت پذیرش در FA جایی است که هر دو شبیه‌سازی فوق در یک حالت DFA خاتمه یابد.

$M = \langle Q, \Sigma, \delta, q_0, F \rangle$ M is DFA $L(M) = L$

construct M'

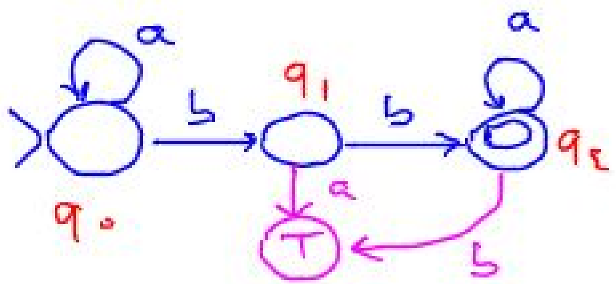
$M' = \langle Q', \Sigma, \delta', q_0', F' \rangle$ M' is NFA $L(M') = \text{leftside}(L)$

$$Q' = (Q \times Q) \cup \{q_0'\}$$
$$F' = \{(q, q) : q \in Q\}$$
$$\delta'(q_0', \lambda) = \{(q_0, q_j) : q_j \in F\}$$
$$\delta'((q_i, q_j), a) = \{(q_i', q_j') : q_i' = \delta(q_i, a),\ q_j = \delta(q_j', a)\} \quad , a \in \Sigma$$

د) حل

$L = L(a^* b b a^*)$

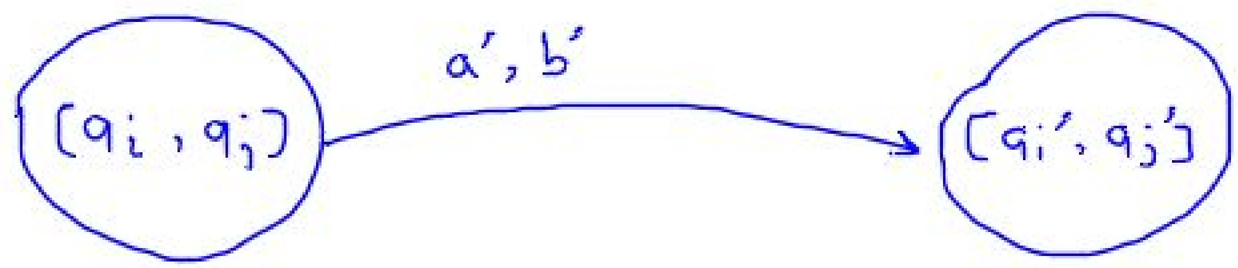
$\text{Leftside}(L) = \{a^n b : n \geq 0\} = L(a^* b)$



$x = a^* b$

$y = b a^*$

aabbaa



$\delta'(q_0', >) = \{(q_0, q_i)\}$

$\delta'((q_i, q_j), a) = \{(q_i', q_j') : q_i' = \delta(q_i, a),\ q_j = \delta(q_j', a)\}$

$((q_0, q_2), aabbaa) \vdash ((q_0, q_2), \underline{a}bbaa)$

$\vdash ((q_0, q_2), \underline{b}baa)$

$\vdash (\underbrace{(q_1, q_1)}_{\in F}, baa)$ accept ✓

$\boxed{aab}$

۹) $\text{Palindrome}(L) \iff (L = L^R)$

L is regular $\Rightarrow$ L^R is regular

وجود الگوریتم برای تساوی دو زبان منظم

$\Downarrow$

وجود الگوریتم برای تساوی L و L^R

$\Downarrow$

وجود الگوریتم برای تعیین آینه بودن L

✓

۷) $L_1 \subseteq L_2$

$$L_1 \subseteq L_2 \iff \underbrace{L_1 \cup L_2}_{\text{منظم}} = \underbrace{L_2}_{\text{منظم}}$$

تساوی دو زبان منظم

۸) مجموعه‌ی همه‌ی رشته‌های به طول زوج روی Σ $= (\Sigma^2)^*$

$$\underbrace{(\Sigma^2)^* \cap \overset{\text{منظم}}{L}}_{\text{منظم}} \stackrel{?}{=} \varnothing$$

وجود الگوریتم برای تعیین تهی بودن یک زبان منظم

9) $\text{tail}(L) = \{ v : uv \in L,\ u, v \in \Sigma^* \}$

$$= \Sigma^* \setminus L$$

$$\left\{ \begin{array}{ll} \Sigma^* & \text{is regular} \\ L & \text{is regular} \end{array} \right. \Rightarrow \Sigma^* \setminus L \text{ is regular} \Rightarrow \text{tail}(L) \text{ is regular}$$

$$\underset{\text{منظم}}{\underset{\downarrow}{L}} = \underset{\text{منظم}}{\underset{\downarrow}{\text{tail}(L)}}$$

وجود الگوریتم برای ت دی در زبان منظم

۱۵) آ) $L = \{ w : n_a(w) = n_b(w) \}$

استفاده از لم تزریق

خودمان/حریف	حریف/خودمان
$m > 0$	$w = a^m b^m a^m b^m$

$w = xyz$

$|xy| \le m \quad |y| \ge 1$

$w = \underbrace{a^k}_{xy} \underbrace{a^{m-k} b^m a^m b^m}_{z}$

$1 \le k \le m$

$w_i = x y^i z$

$= a^1 (a^{k-1})^i a^{m-k} b^m a^m b^m$

$i = 0 \Rightarrow w_0 = a^1 a^{m-k} b^m a^m b^m$

$n_a(w_0) \ne n_b(w_0) \Rightarrow w_0 \notin L$

پس L منظم نیست.

ب) $L = \{ a^{k^2} : k \ge 0 \}$ $\qquad w = a^{m^2}$

ج) $L = \{ a^n : n \text{ is prime} \}$ $\qquad w = a^p \leftarrow$ عدد اول

p کوچکترین عدد اول بزرگتر یا مساوی با m انتخاب می‌شود.

اگر $y = a^k$ انتخاب شود، در این صورت $w_i = a^{p+(i-1)k}$ خواهد بود.

اگر $i - 1 = p$ انتخاب شود، بدیهی است که $p + (i-1)k = p(k+1)$ که به وضوح یک عدد مرکب است، $w_{p+1} \notin L$.

پس L منظم نیست.

۱۱) $L = \{w_1 c w_2 : w_1, w_2 \in \{a,b\}^*, w_1 \neq w_2\}$

اگر L منظم باشد، باید $\bar{L}$ هم منظم باشد.

اشتراک دو زبان منظم، منظم است.

داریم:

$$\bar{L} = \dots$$

$$L_1 = \bar{L} \cap \underbrace{L(a^* c a^*)}_{\text{منظم است}} = \underbrace{\{a^n c a^n : n \geq 0\}}_{L_1 \text{ منظم نیست}}$$

چون اشتراک زبان منظم $\bar{L}$، زبان منظم $L(a^* c a^*)$ یک زبان نامنظم شده است، پس $\bar{L}$ و در نتیجه L نمی‌تواند منظم باشد.

۱۲) L_1 and L_2 are regular languages

$$L = \{w : w \in L_1 \wedge w^R \in L_2\}$$

$$= \{w : w \in L_1 \wedge w \in L_2^R\}$$

$$= \underbrace{\underbrace{L_1}_{\text{منظم}} \cap \underbrace{L_2^R}_{\text{منظم}}}_{\text{منظم}}$$