دانشگاه صنعتی شریف

دانشکده‌ی مهندسی کامپیوتر

درس نظریه‌ی زبان‌ها و ماشین‌ها

سوالات نمونه

پاسخ‌نامه‌ی مجموعه‌ی ۶ : زبان‌های منظم - بخش ۳

استاد: دکتر علی موقر

تیم دستیاران درس - نیم‌سال دوم ۰۲ - ۰۱

۱۰ اردیبهشت ۱۴۰۲

۱. زبان‌های نامنظم

(a) $L = \{ a^i b^j a^k \mid k > i+j \}$ 1.1

- فرض می‌کنیم طبق لم تزریق برای هر رشته با طول حداقل n، شرایط این لم برقرار باشد.

- رشته‌ی w را اینگونه درنظر می‌گیریم $\Longleftarrow w = a^n b^n a^{2n+1}$

- فرض می‌کنیم $w = xyz$ باشد $\Longleftarrow$

$\xrightarrow{\text{باید}} |xy| \leqslant n \xrightarrow{\text{پس}} y = a^k$

$\longrightarrow xy^iz = a^{n-k}(a^{ki})z$ شامل b^n است

- پس با درنظر گرفتن اعداد بزرگ برای i در xy^iz، شرط $k > i+j$ نقض می‌شود. پس لم تزریق برقرار نیست و زبان نامنظم است.

(b) $L = \{ a^i b^j \mid i = j \vee j = 2i \}$

- فرض می‌کنیم لم تزریق با طول n برقرار است.

- رشته‌ی w را این‌گونه تعریف می‌کنیم $\Leftarrow$ $w = a^n b^n$

$w = xyz \xrightarrow{\text{باید}} |xy| \leq n \xrightarrow{\text{پس}} y = a^k$

- پس با درنظر گرفتن i بزرگ در xy^iz، تعداد a ها از b بیشتر می‌شود و هر دو شرط $i = j$ و $j = 2i$ نقض می‌شوند.

$$xy^iz = \underbrace{a^m}_{x} a^{ki} a^l b^j \longrightarrow \exists i, \quad m+ki+l \neq j$$

$$8$$

$$2(m+ki+l) \neq j$$

ⓒ $L = \{w \in \{a,b\}^* \mid n_a(w) < 2n_b(w)\}$

- فرض می‌کنیم لم تزریق با طول n برقرار است.
- رشته‌ی w را این‌گونه در نظر می‌گیریم $\Leftarrow$ $w = a^n b^n$

$w = xyz, \ |xy| \leq n \xrightarrow{\text{پس}} y = a^k, \ k > 0$

- حال کافی است i را به اندازه‌ای بزرگ بگیریم که $\Leftarrow$

$ik > 2n$

$\xrightarrow{\text{پس}} xy^iz = a^{n-k} a^{ik} b^n \longrightarrow n-k+ik > 2n$

پس xy^iz عضو زبان نیست، پس زبان منظم نیست.

ⓓ $L = \{www \mid w \in \{a,b\}^*\}$

- فرض می‌کنیم لم تزریق با طول n برقرار است.
- w را اینگونه در نظر می‌گیریم

$$w = a^n b a^n b a^n b$$

$$w = xyz \,,\ |xy| \leqslant n \longrightarrow y = a^k$$

$$\longrightarrow xy^2z = \underline{a^{n+k} b a^n b a^n b}$$

این رشته شرایط www را ندارد، لم تزریق برقرار نیست و

زبان منظم نیست.

ⓔ $L = \{ a^{2^n} \mid n \geqslant 0 \}$

- فرض می‌کنیم لم تزریق با طول n برقرار است.
- w را اینگونه در نظر می‌گیریم: $w = a^{2^n}$

$$w = xyz \,,\ |xy| \leqslant n \longrightarrow y = a^k \,,\ k \leqslant n$$

فرض می‌کنیم xy^2z نیز عضو زبان باشد، یعنی:

$$\exists n_1 \quad a^{2^n + k} = a^{2^{n_1}} \longleftrightarrow 2^{n_1} - 2^n = k$$

از طرفی چون $n_1 > n$، داریم $\Leftarrow$

$$K = 2^{n_1} - 2^n > 2^n \Longleftrightarrow \underline{K > 2^n \,,\ K \leqslant n} \quad \times$$

- پس این فرض غلط است و xy^2z عضو زبان نیست و لم تزریق برقرار

نیست.

۴

- فرض می‌کنیم لم تزریق با طول n برقرار است.
- w را این گونه در نظر می‌گیریم:

$$w = \alpha^{n} \gamma \alpha^{n+1} \gamma \cdots \gamma \alpha^{2n}$$

- لم تزریق:

$$w = xyz , |xy| \leqslant n \longrightarrow y = \alpha^{k} , k \leqslant n$$

$$\xrightarrow{\text{پس}} xy^2z = \underbrace{\alpha^{n-k} \alpha^{2k}}_{w_1'} \gamma \alpha^{n+1} \cdots \gamma \alpha^{2n}$$

$$0 < k \leqslant n \longrightarrow n+1 \leqslant n+k \leqslant 2n$$

- پس یعنی:

$$\exists j \longrightarrow w_1' = \underbrace{\alpha^{n+j}}_{w_j} , 1 < j \leqslant n$$

$$\xrightarrow{\text{پس}} xy^2z \notin L$$

پس لم تزریق برقرار نیست.

۲.۱

اثبات برقراری لم تزریق $\Leftarrow$

- فرض می‌کنیم $n=1$ است. یعنی ادعا می‌کنیم که برای هر رشته‌ای با طول بزرگ‌تر از ۱، می‌توان آن را به گونه‌ای به xyz تجزیه کرد که $xy^iz \in L$ و $|xy| \leqslant n$.

حالت I :

$$i=1 \longrightarrow w = ab^Nc^N \quad N \geq 0$$

$$\longrightarrow \begin{cases} x=\varepsilon \\ y=a \\ z=b^Nc^N \end{cases} \longrightarrow \underline{xy^iz = a^i b^N c^N \in L}$$

حالت II :

$$i=N>1 \longrightarrow w = a^N b^j c^k$$

$$\begin{cases} x=\varepsilon \\ y=a^N \\ z=b^jc^k \end{cases} \longrightarrow xy^iz = \underline{a^{iN} b^j c^k \in L}$$

حالت III :

$$i=0 \longrightarrow w = b^j c^k$$

$$\longrightarrow \begin{cases} x=\varepsilon \\ y=b \\ z=b^{j-1}c^k \end{cases} \longrightarrow \underline{xy^iz \in L}$$

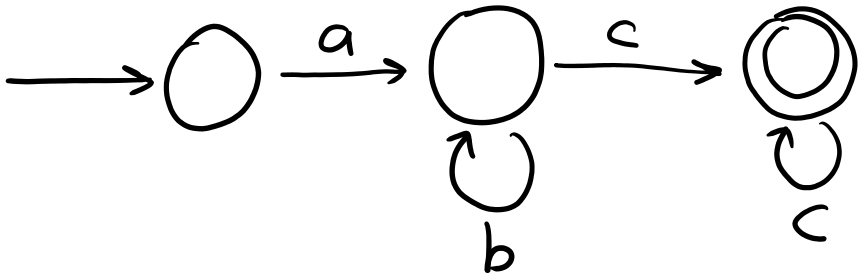
اگر $j=0$ بود، $y=c$ را می‌توان در نظر گرفت.

ـ پس لم تزریق برقرار است.

حال سعی می‌کنیم ثابت کنیم که L منظم نیست. با برهان خلف فرض می‌کنیم که L منظم است. زبان ما را اینگونه تعریف می‌کنیم $\Leftarrow$

$$L_1 = \{ab^jc^k \mid \forall\, j,k \geq 0\}$$

این زبان منظم است، زیرا DFA زیر آن را تولید می کند ⇐



- طبق فرض باید $L \cap L_1$ نیز منظم باشد ⇐

$$L \cap L_1 = \{ab^N c^N \mid N \geq 0\} = L_2$$

- اما L_2 منظم نیست. زیرا با فرض لم تزریق برای طول p، رشته‌ی روبه‌رو شرط لم تزریق را ارضا نمی کند.

$$w = ab^p c^p$$

$$w = xyz,\ |xy| \leq p \longrightarrow p = \begin{cases} ab^k \\ b^k \end{cases}$$

$$\longrightarrow xy^2z = \begin{cases} a^2b^{2k}b^m c^p \notin L_2 \\ ab^{2k}b^m c^p \notin L_2 \end{cases}$$

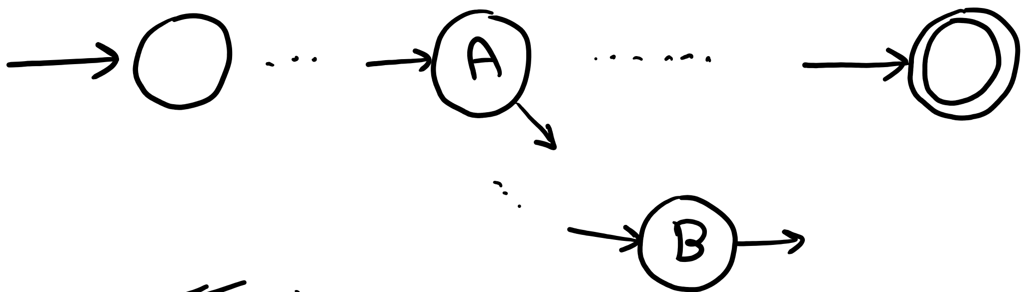
- پس یعنی L_2 منظم نیست ⟵ L منظم نیست.

۳.۱

(الف) فرض می کنیم L یک زبان منظم باشد. $w \in L$ را این گونه

در نظر می‌گیریم $\omega = z_1 z_2 z_3$, $|z_2| = n$, $n = |Q|$

که در آن $(Q, \Sigma, \delta, q_0, F)$ ، DFA منسوب به L است:



← A را آخرین state ، پس از خواندن z_1 در نظر می‌گیریم

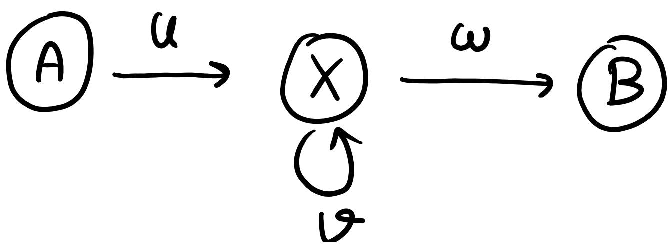
B را آخرین state ، پس از خواندن z_2 در نظر می‌گیریم

← در مسیر A به B هنگام خواندن z_2 ، به اندازه‌ی $|z_2| = n$ گام برمی‌داریم.

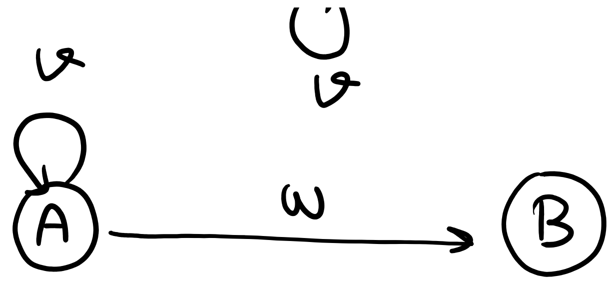
پس بدون در نظر گرفتن A به عنوان state شروع، از n تا state گذر خواهیم کرد.

چون $|Q| = n$ ، طبق اصل لانه کبوتری ، یا اینکه از یک state مثل X دو بار عبور کردیم یا اینکه بر روی A باز می‌گردیم.

- پس در مسیر A تا B یک دور خواهیم داشت ⇐



یا



- پس همانند شکل و با استدلال مشابه اثبات لم تزریق، z_2 را می‌توان به این صورت بازنویسی کرد $\Leftarrow$

$$z_2 = uvw \quad , \quad |v| \geq 1$$

به طوریکه مسیر طی شده از (A) تا (B)، برای رشته های uv^iw یکسان است. [فقط بخش حلقه چند بار طی می‌شود، i بار]

$\xrightarrow{\text{پس}}$ $\underline{z_1 u v^i w z_3 \in L}$ $\square$.

(ب) فرض می‌کنیم زبان L منتظم باشد. طبق لم بالا، هر w را باید بتوان با شرایط گفته شده به شکل $z_1 z_2 z_3$ نوشت. *

$\boxed{|w| \geq n}$ $\to$ ab^nc^n

- رشته‌ی w را در نظر می‌گیریم.

که در آن $z_1 = a$ ، $z_2 = b^n$ ، $z_3 = c^n$ است.

$$z_2 = uvw \longrightarrow v = b^k \ , \ k \leq n$$

$$\longrightarrow z_1 u v^2 w z_3 = a b^{n+\alpha} c^n \ , \ \alpha \geq 1 \quad \notin L$$

$\longrightarrow z_1 u v^2 w z_3 = \underline{a\,b \quad c^{..}}$ ، از α

پس این لم برقرار نیست، یعنی L منتظم نیست.

۴.۱

ⓐ $L' := \{ b^i c^j \mid i,j \geqslant 0 \} \longrightarrow$ منتظم است

$L'' = L \cap L' = \{ b^m c^m \mid m \geqslant 0 \} \longrightarrow$ طبق مثال های حل شده، منتظم نیست

- اگر L منتظم باشد، طبق بسته بودن $\cap$، باید L'' هم منتظم باشد، اما این طور نیست، پس L منتظم نیست.

ⓑ $L' := \{ ww \mid w \in \{a,b\}^* \} \longrightarrow$ طبق مثال های کتاب و تمرین ۱.۱، منتظم نیست.

$L'' := \{ w \in \{a,b\}^* \mid |w| \overset{2}{\equiv} 1 \} \longrightarrow$ طبق مثال های اولیه حل شده، منتظم است.

$L \cup L'' = \{a,b\}^* - L' \longrightarrow$ حاصل این اجتماع برابر است با همه‌ی رشته‌هایی که طول فرد دارند، یا

L'' طول زوج دارند اما به شکل aaa

نیستند.

L

- اگر L منظم باشد، طبق بسته بودن عمل $\cup$، با $L \cup L''$ نیز منظم باشد، و اگر $L \cup L''$ منظم باشد، طبق بسته بودن عمل مکمل با L' هم منظم باشد، اما اینگونه نیست، پس یعنی L منظم نیست.

(C)

$$L' := \{ab^j \mid j \geq 1\} \longrightarrow \text{منظم است}$$

$$L \cap L' = \{ab^{2^k} \mid k \geq 1\}$$

معکوس

$$(L \cap L')^R = \{b^{2^k}a \mid k \geq 1\} = L''$$

$$L'' / \{a\} = \underbrace{\{b^{2^k} \mid k \geq 1\}}_{=A} \longleftarrow$$

طبق تمرین‌ها، منظم نیست.

right-quotion

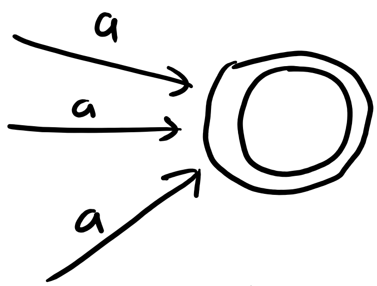
توضیح $\Longrightarrow$

- اگر L منظم باشد، طبق بسته بودن عمل اشتراک و معکوس، L'' نیز منظم خواهد بود.

- اگر L'' منظم باشد، طبق بسته بودن عمل right quotion و منظم بودن $\{a\}$، A منظم خواهد بود. *

- A منظم نیست، پس L منظم نیست.

* ← برای اثبات بسته بودن عمل right quotion، کافی است این گونه در نظر بگیریم که DFA حاصل از زبان L'' (در صورت منظم بودن)، دارای state نهایی به شکل زیر خواهد بود =>



- به ازای هرکدام از این state های نهایی، آن را حذف کرده و state ای که با a به آن وصل می شد را نهایی در نظر می گیریم. با این کار $L''\backslash\{a\}$ نیز منظم خواهد بود.

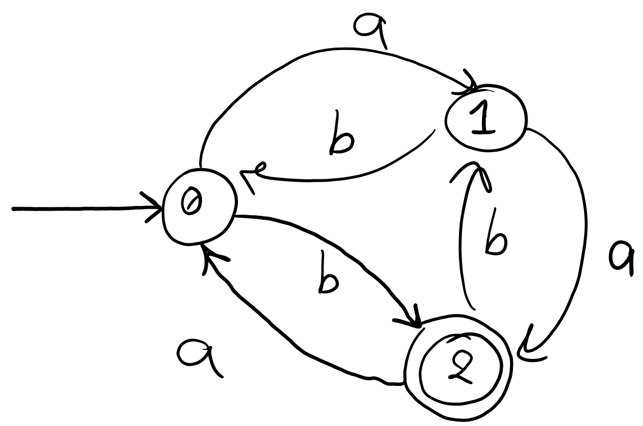
۲

۱.۲ - برای هرکدام ابتدا GNFA می سازیم و پس state ها را حذف می کنیم ⇐

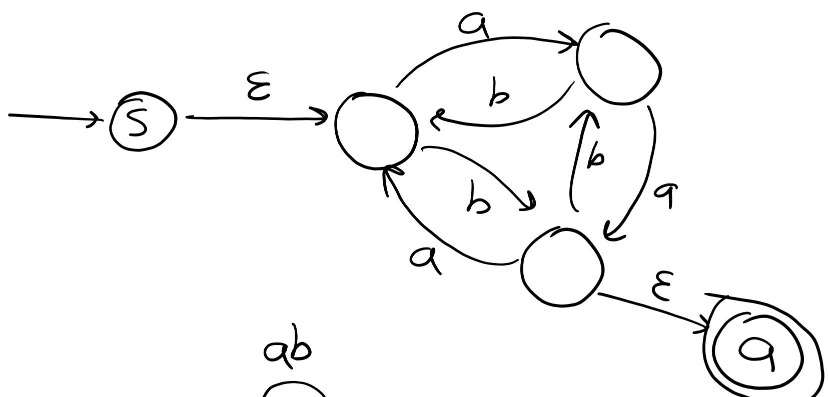
a

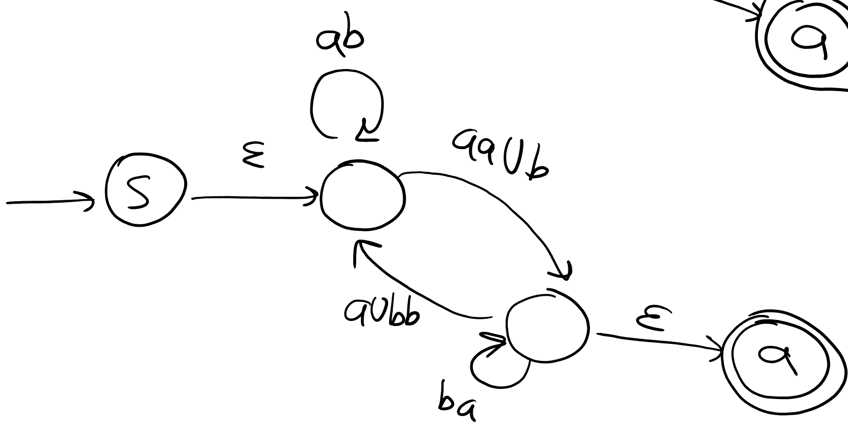
- هر state بیانگر $n_a(w) - n_b(w)$ به پیمانه‌ی ۳ است.

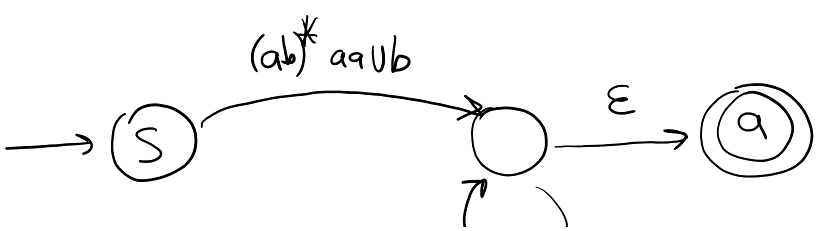
DFA ⇒



تبدیل ⇒







۲. عبارت‌های منظم

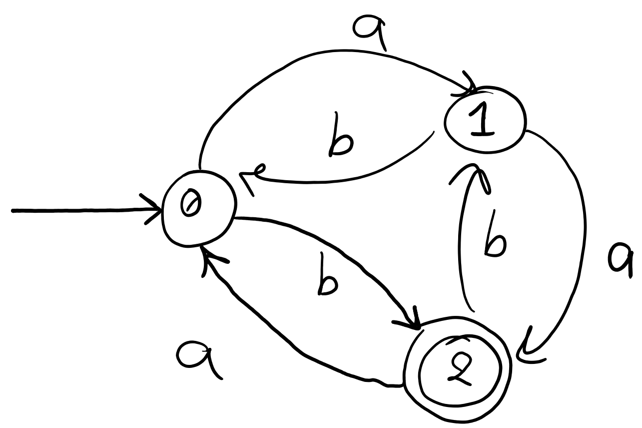
۱.۲

- برای هرکدام ابتدا GNFA می سازیم و پس state ها را حذف می کنیم ⇐

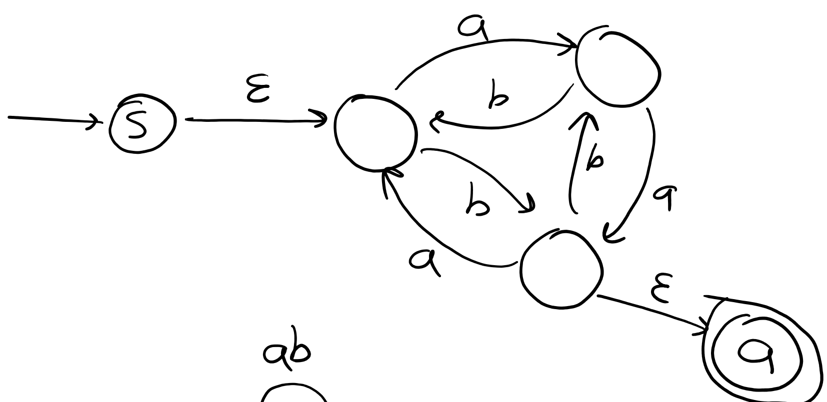
(a)

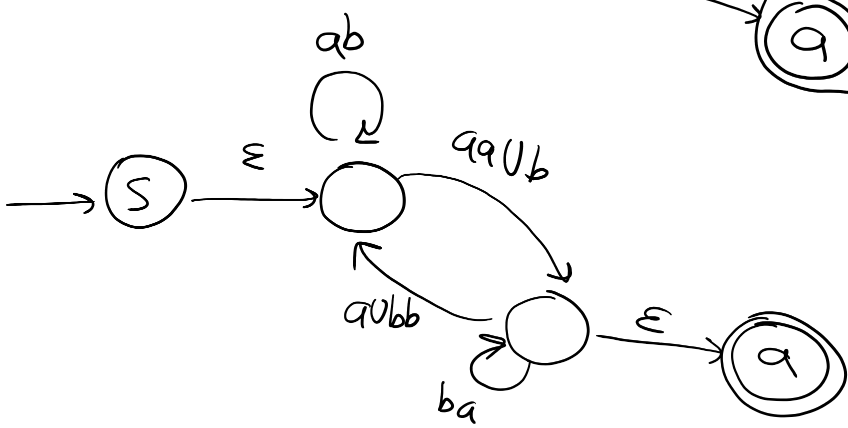
- هر state بیانگر $n_a(w)-n_b(w)$ به پیمانه‌ی ۳ است.

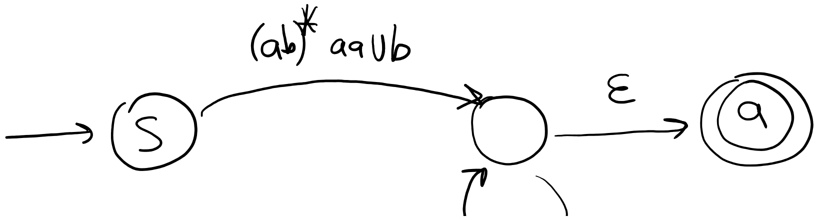
DFA ⇒

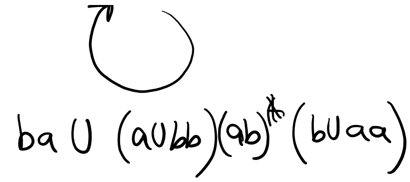


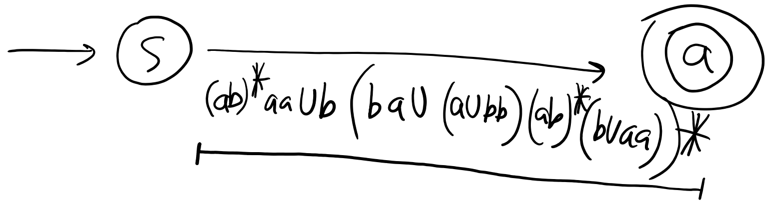
تبدیل ⇒



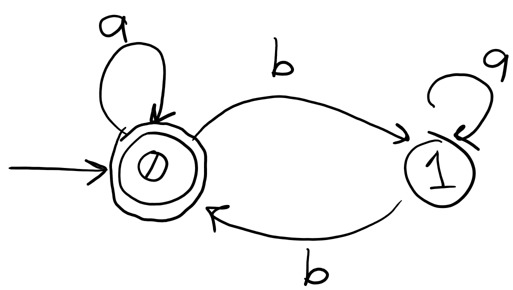




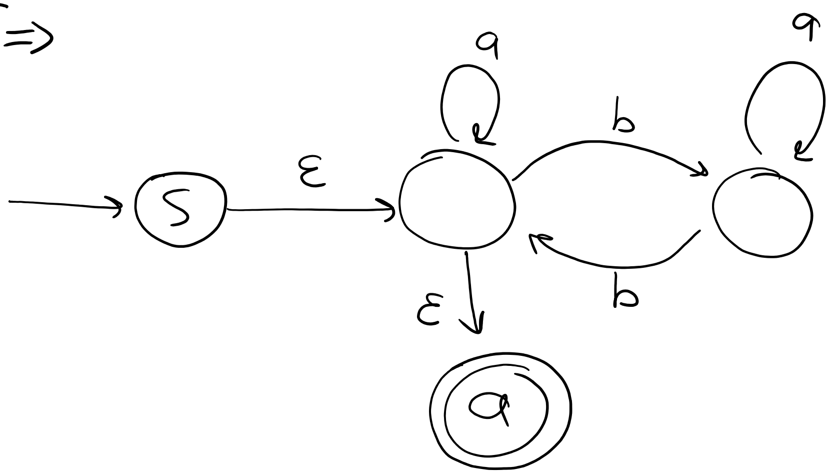

ba U (a U bb)(ab)* (b U aa)

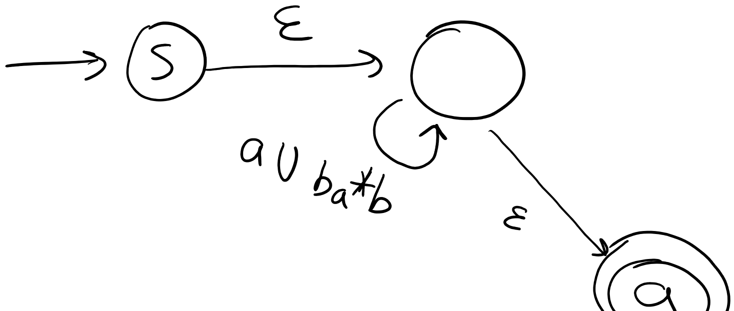

S
a
(ab)* aa U b (ba U (a U bb)(ab)* (b U aa))*

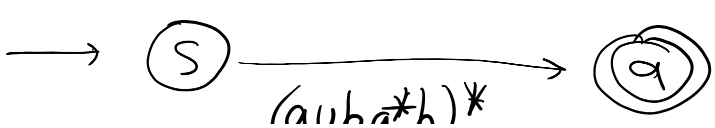
(b)


a
b
a
∅
1
b

تبدیل =>


a
a
S
ε
b
b
ε
a

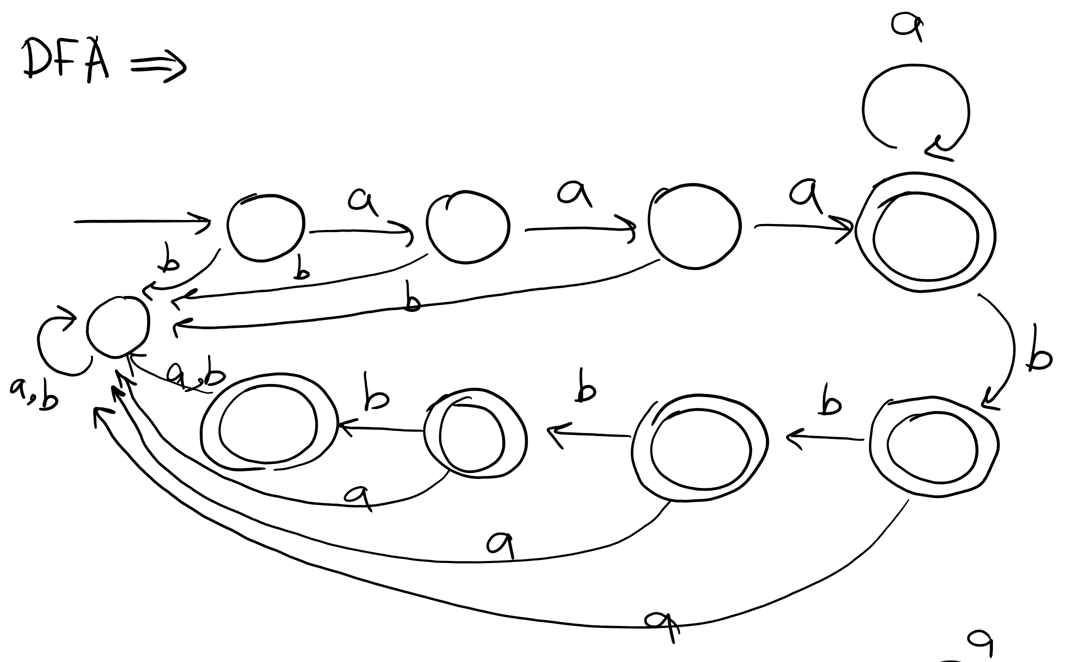

S
ε
a U ba*b
ε
a


S
a
(a U ba*b)*

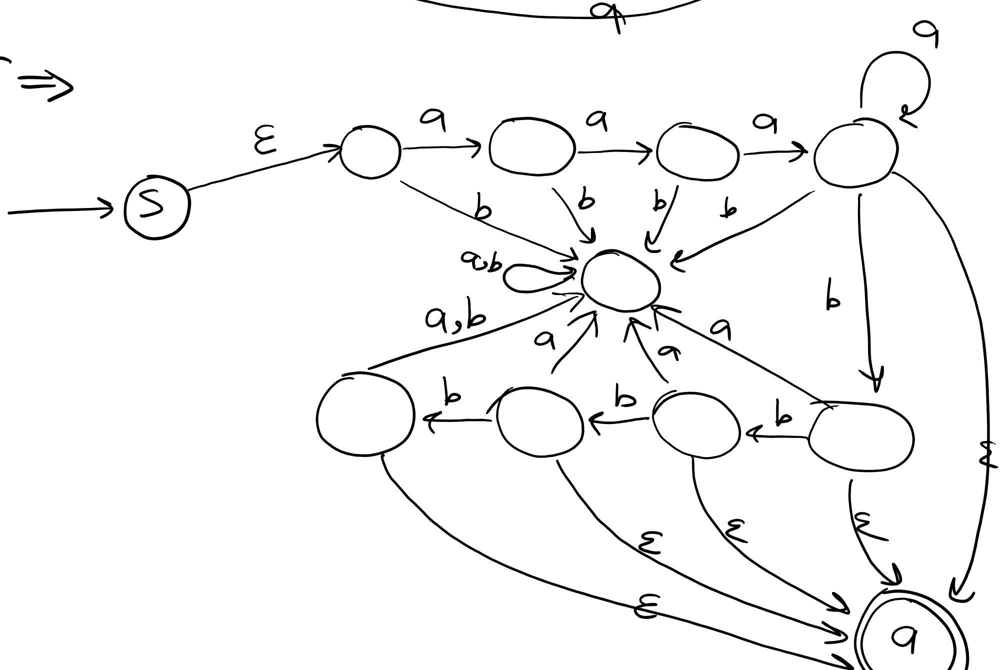
$(a \cup ba^*b)^*$

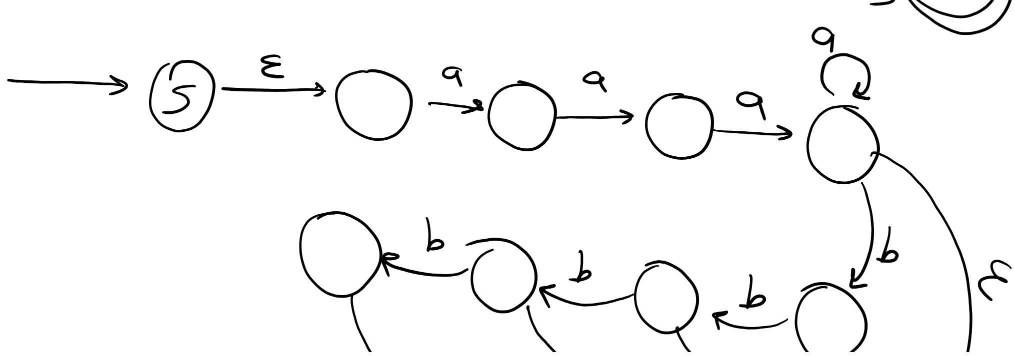
(C)

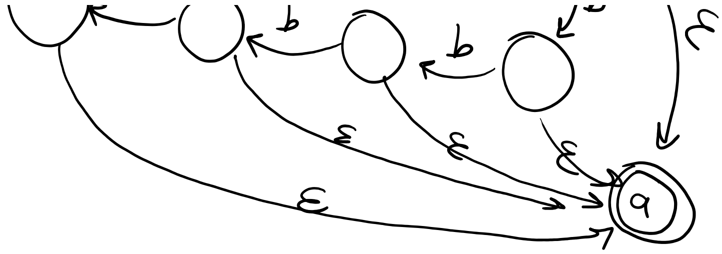
DFA ⇒

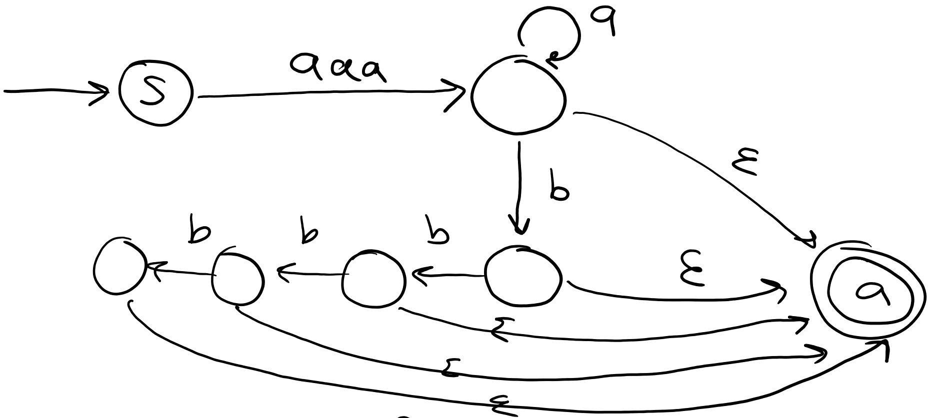


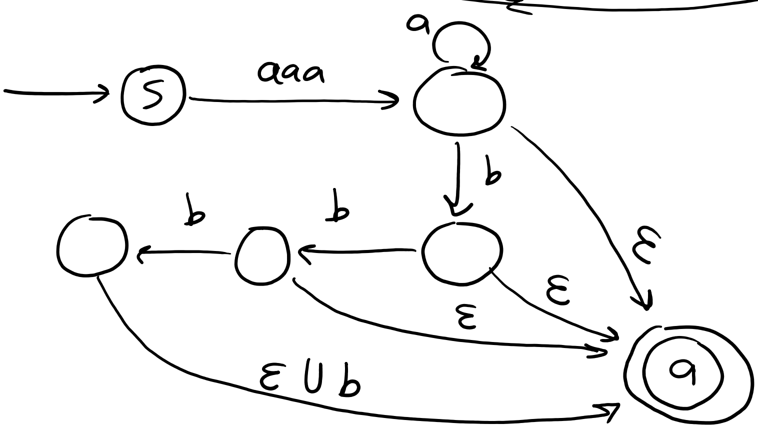
تبدیل ⇒

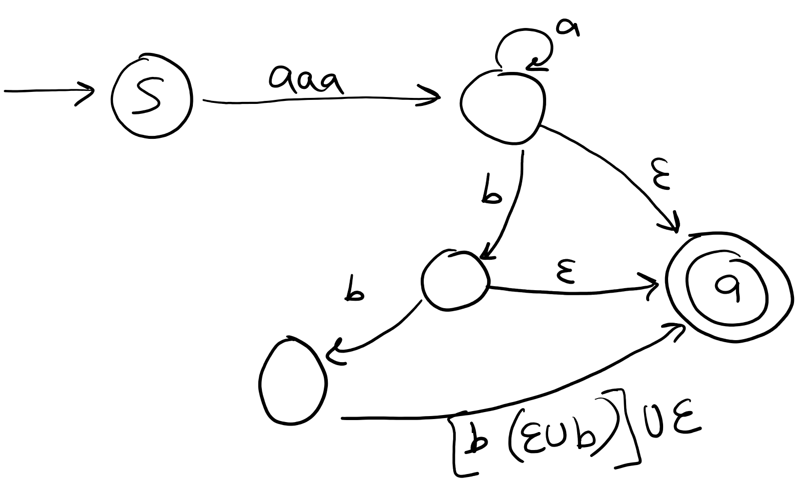


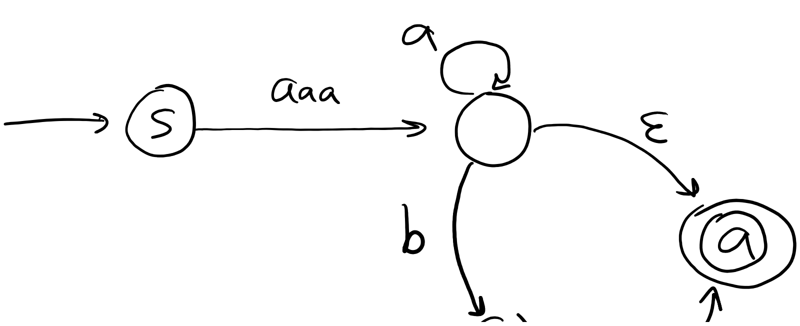


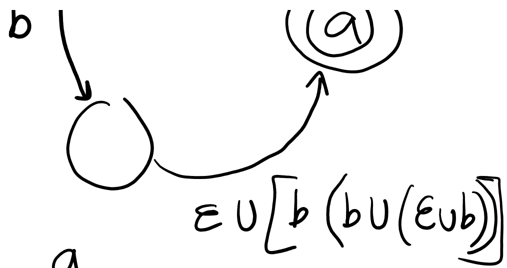

b
b
ε
ε
ε
ε
ε
a

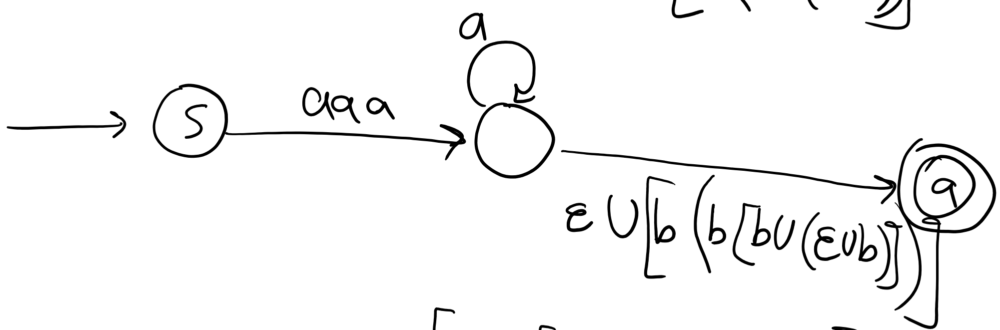

S
aaa
a
b
ε
b
b
b
ε
ε
ε
ε
a

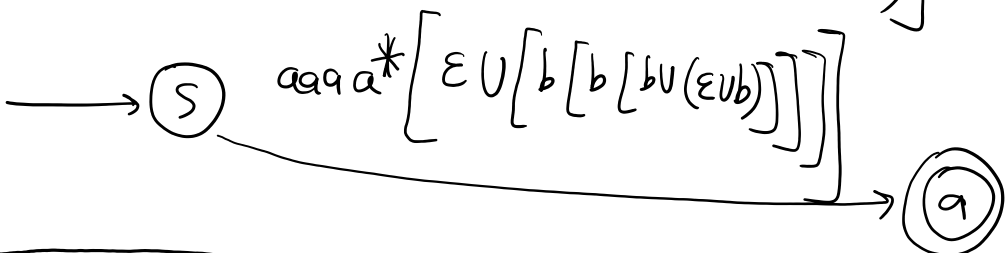

S
aaa
a
b
ε
b
b
ε
ε
ε ∪ b
a


S
aaa
a
b
ε
b
ε
[b (ε∪b)] ∪ ε
a


S
aaa
a
ε
b
a

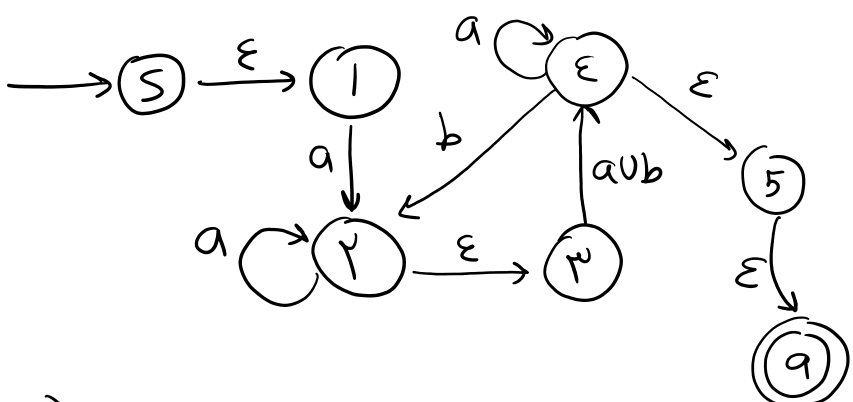




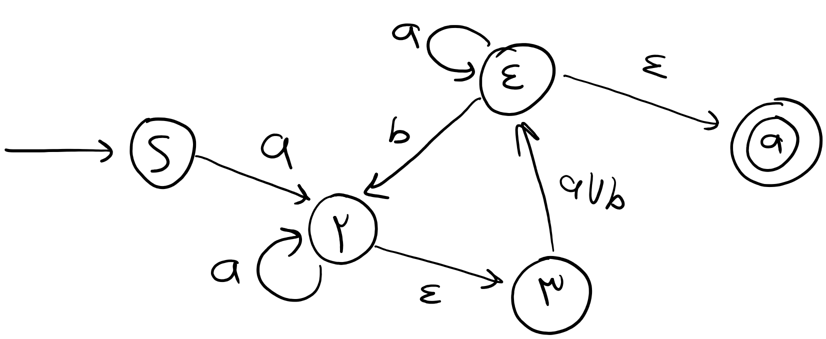


۲.۲

a



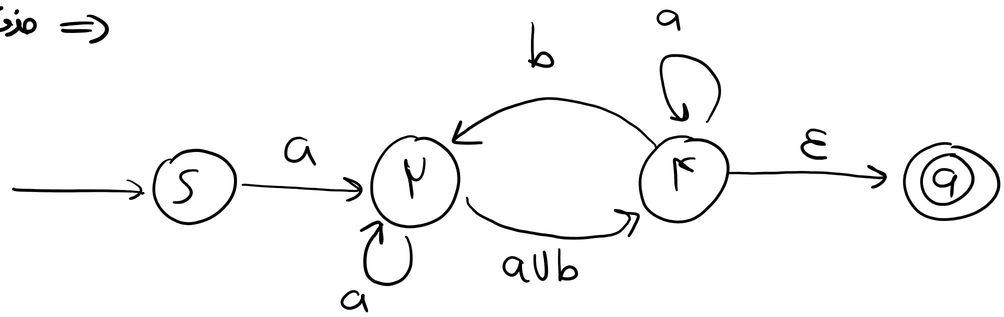
۱ و ۵ حذف ⇒



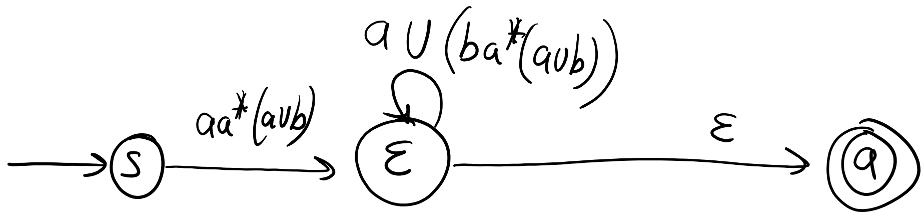
۳ حذف ⇒

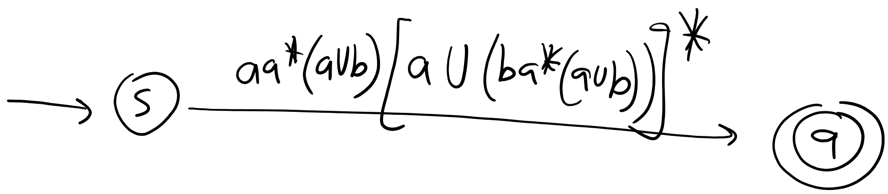
۱ a

حذف ۳ ⇒

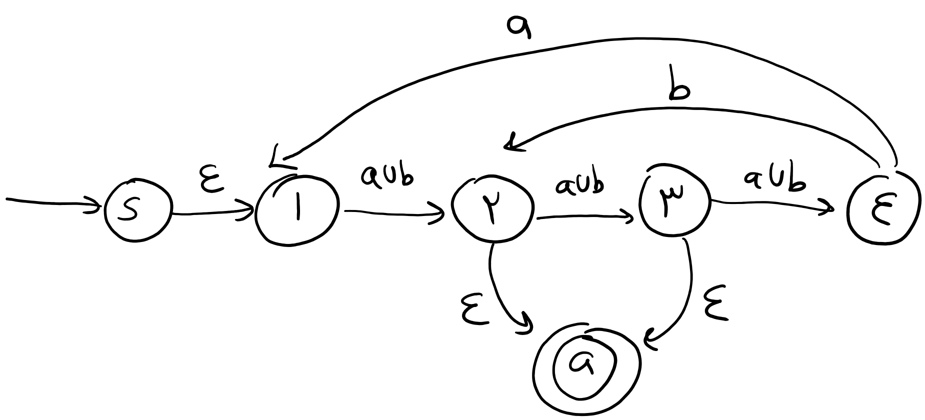


حذف ۲ ⇒

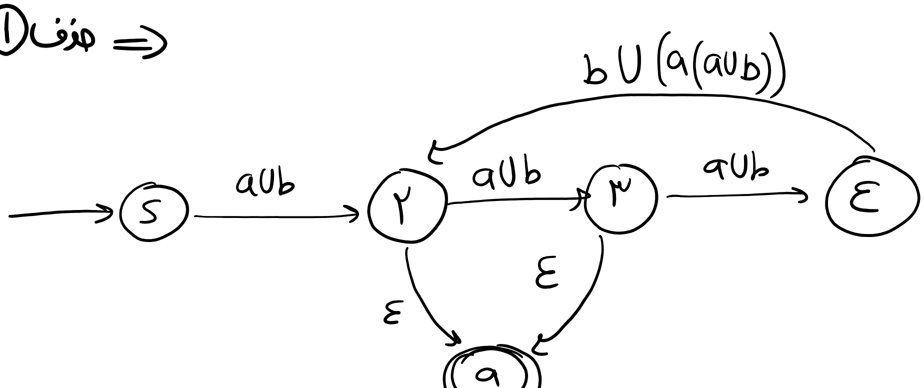




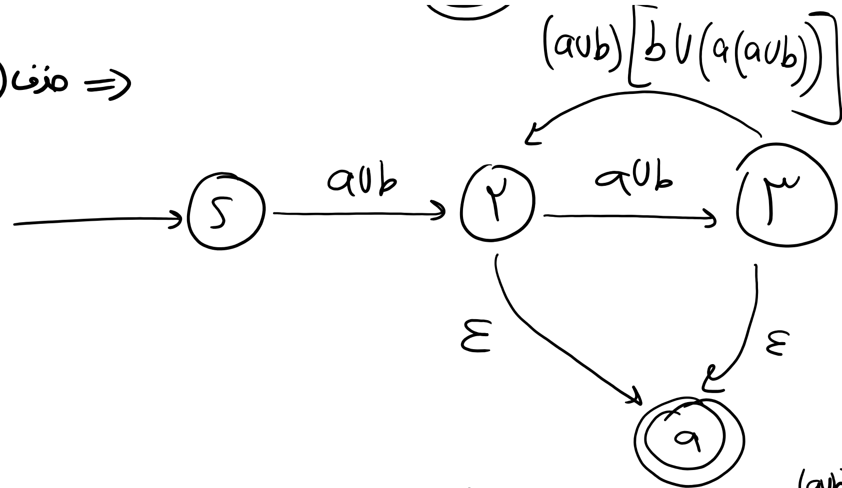
(b)



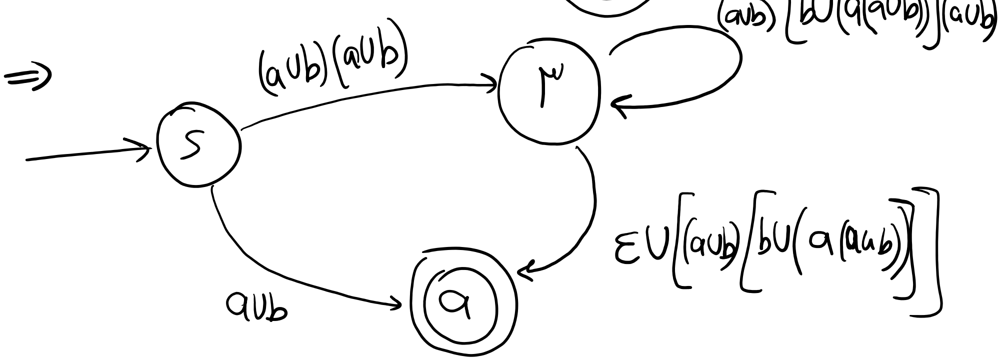
حذف ۱ ⇒



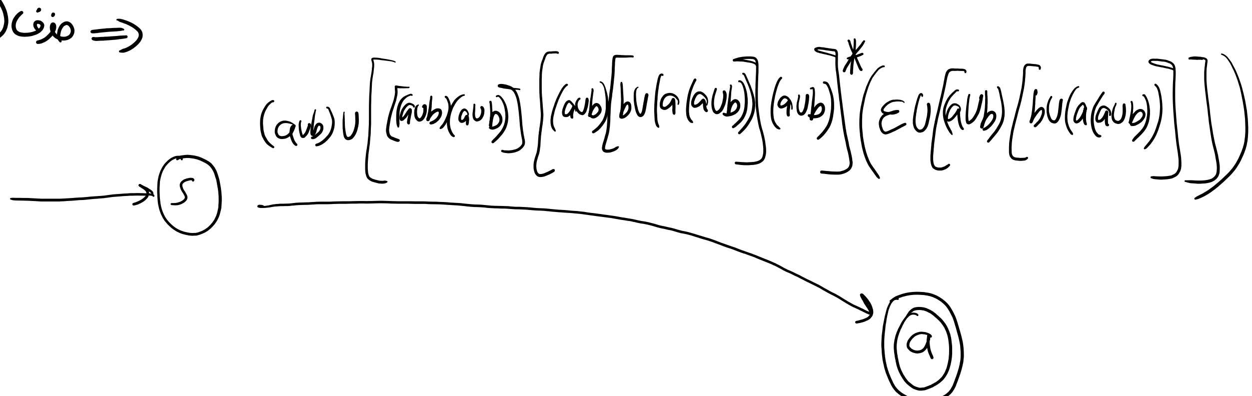
حذف 4 ⇒



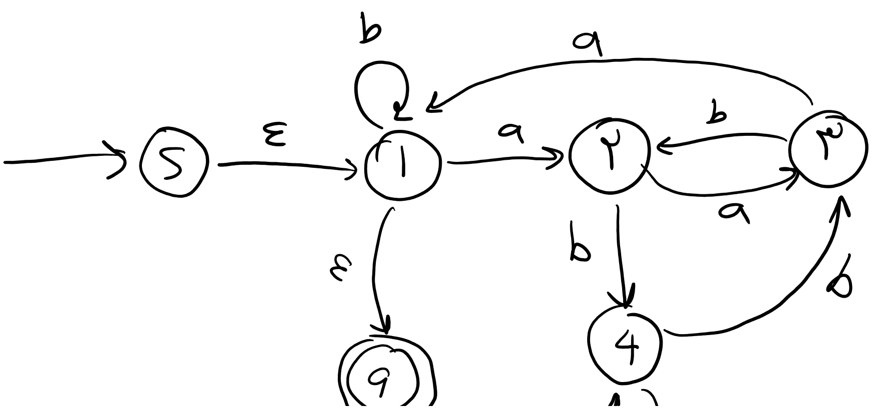
حذف ۲ ⇒

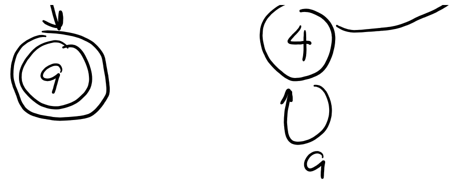


حذف ۳ ⇒

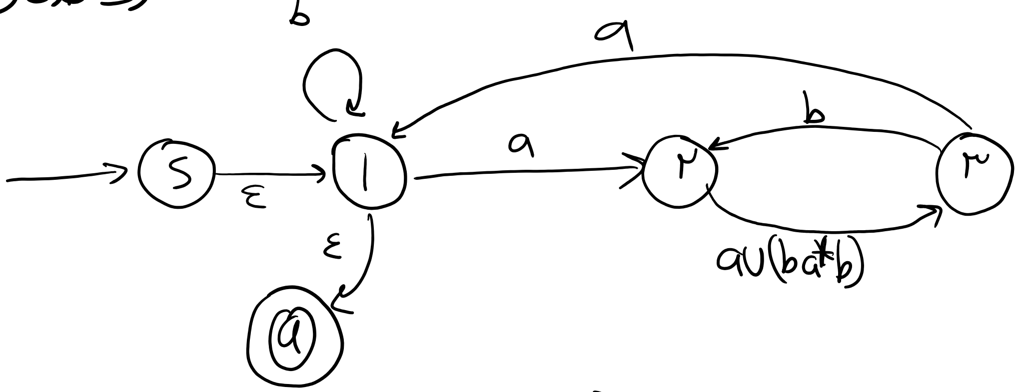


C

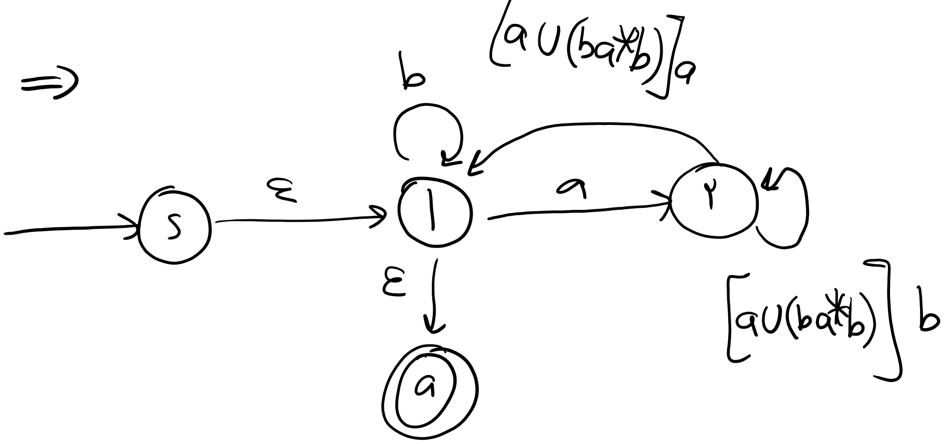




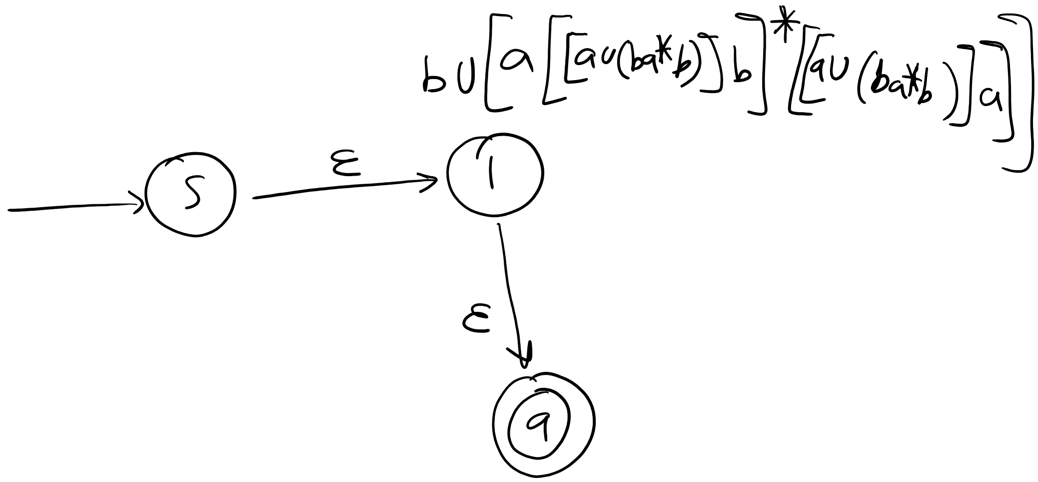
حذف ④ ⟹

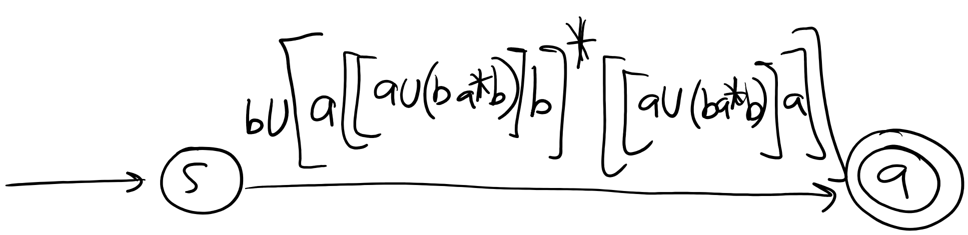


حذف ③ ⟹



حذف ۲ ⟹



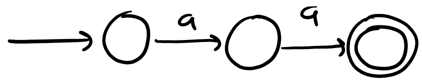


(۳.۲)

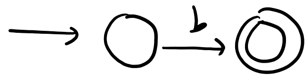
_ در هر مرحله ، از پایین به بالا DFA را می‌سازیم .

(a)

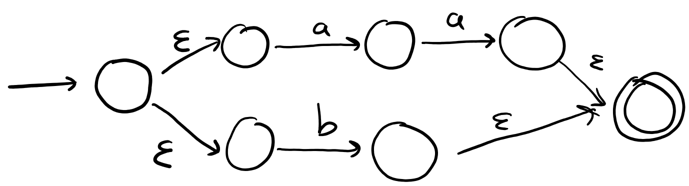
aa



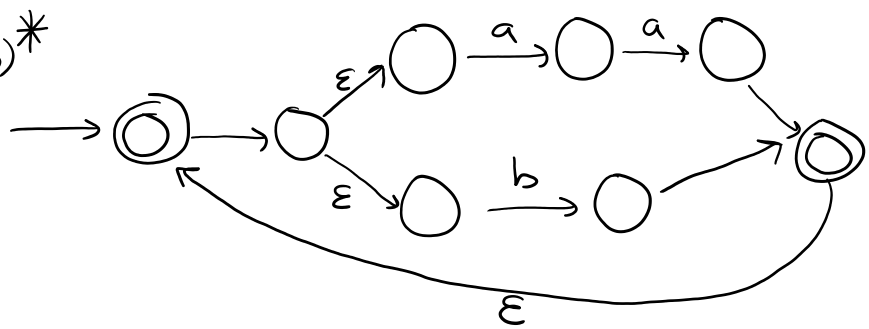
b



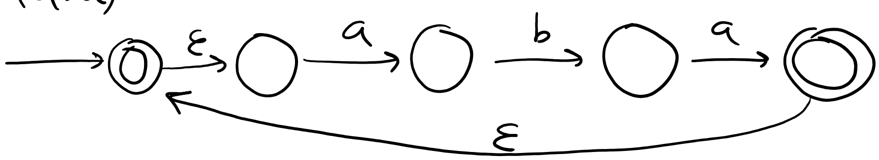
aa∪b



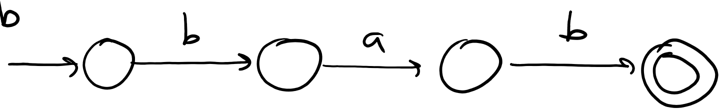
$(aa \cup b)^*$



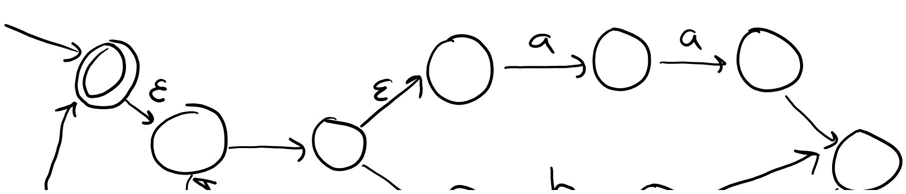
$(aba)^*$

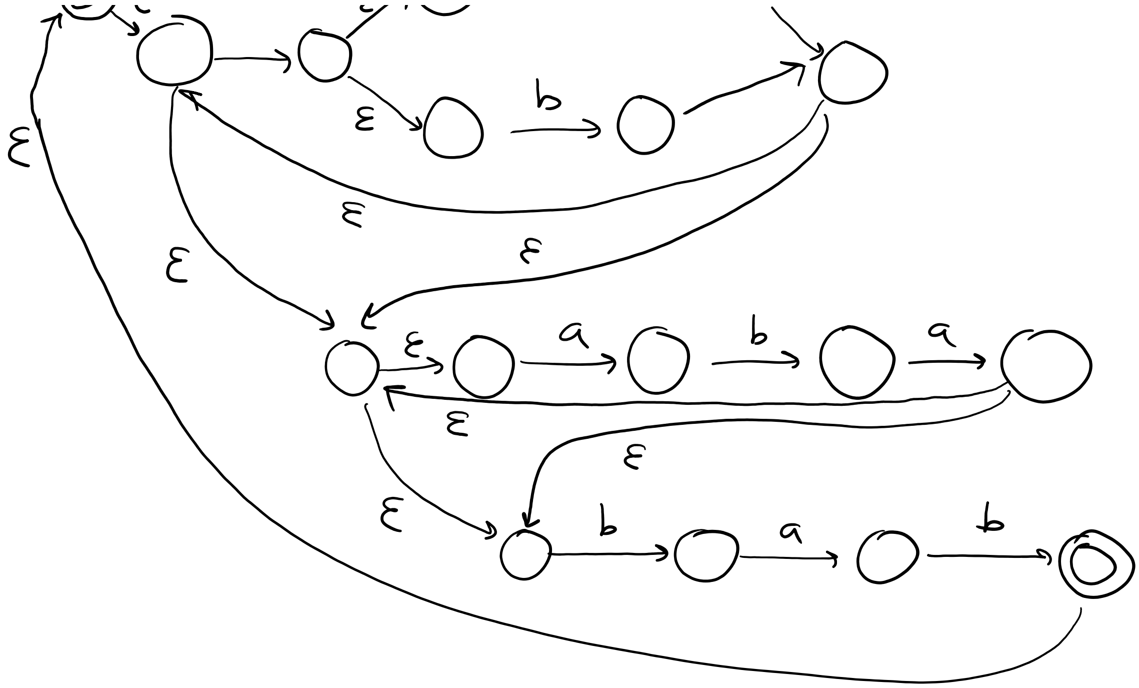


bab



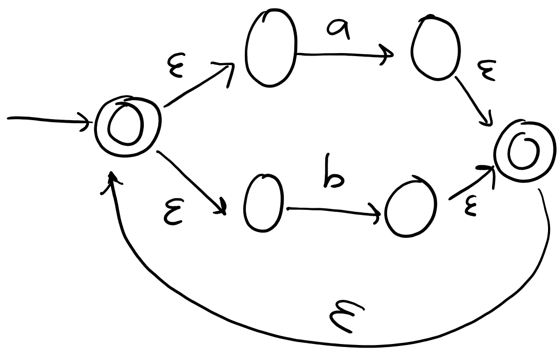
$\left[(aa \cup b)^* (aba)^* bab\right]^* \Rightarrow$



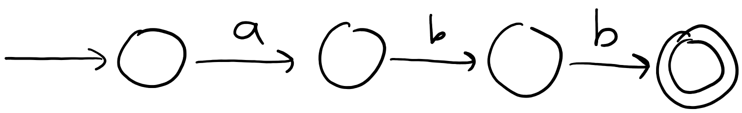


(b)

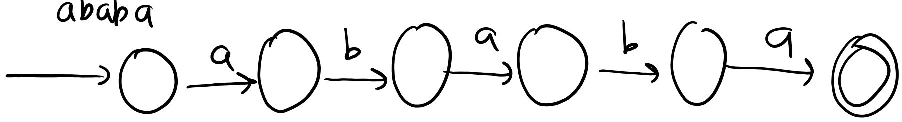
$(a \cup b)^*$



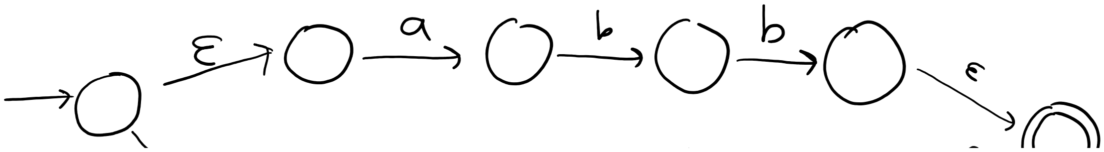
abb

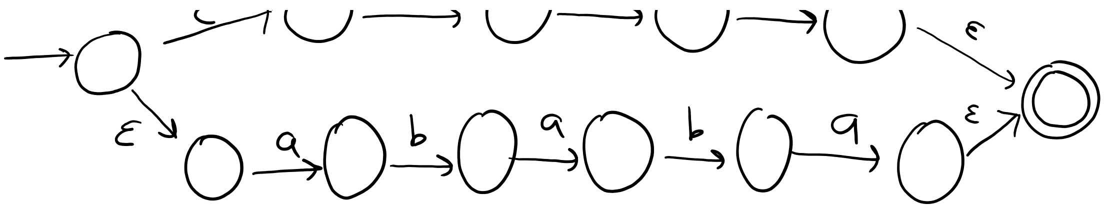


ababa

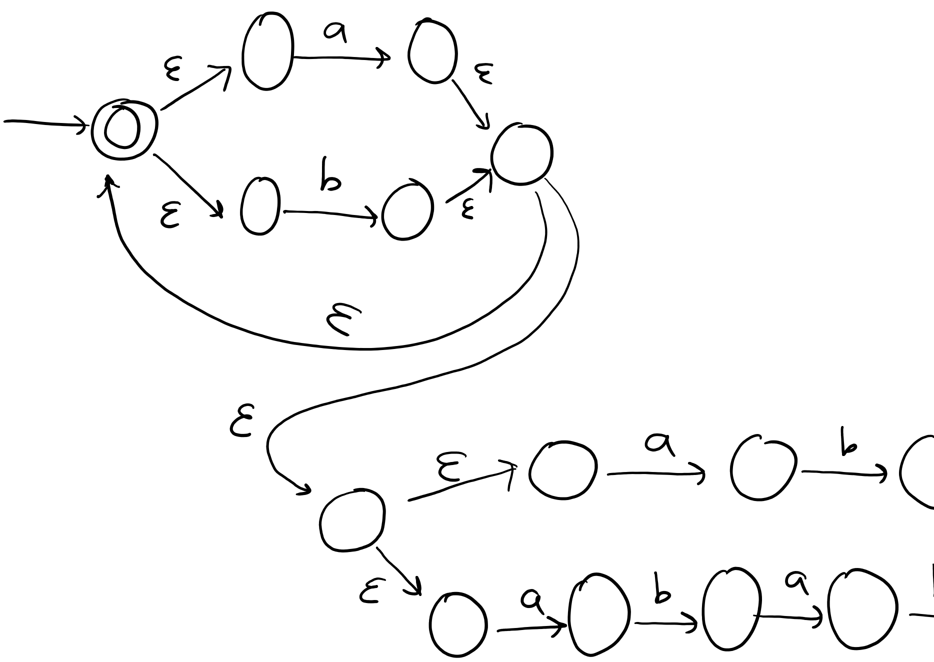


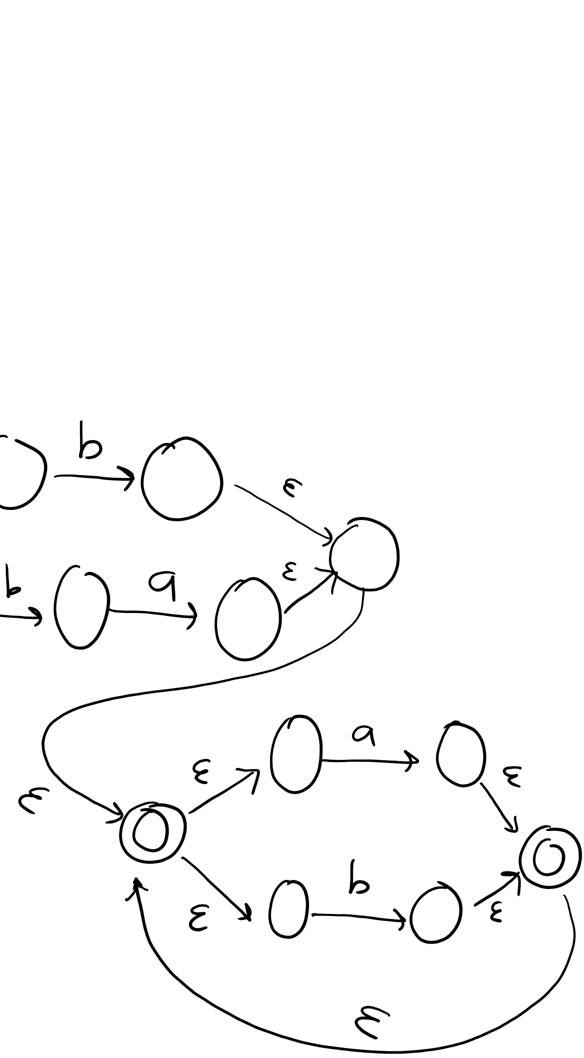
$abb \cup ababa$





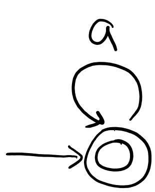
$(a \cup b)^*(abb \cup ababa)(a \cup b)^*$



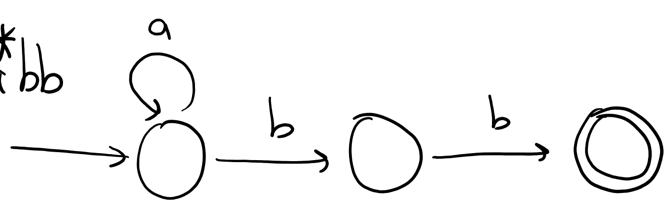


(c)

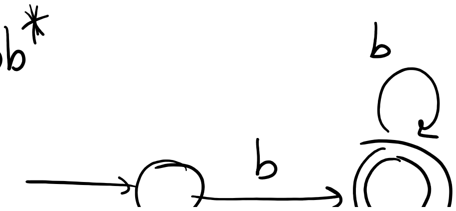
a^*

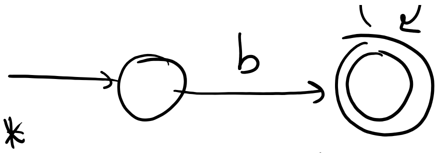


a^*bb

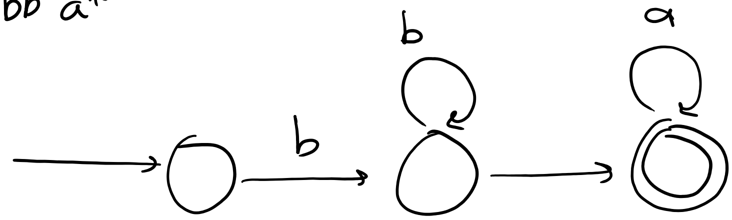


bb^*

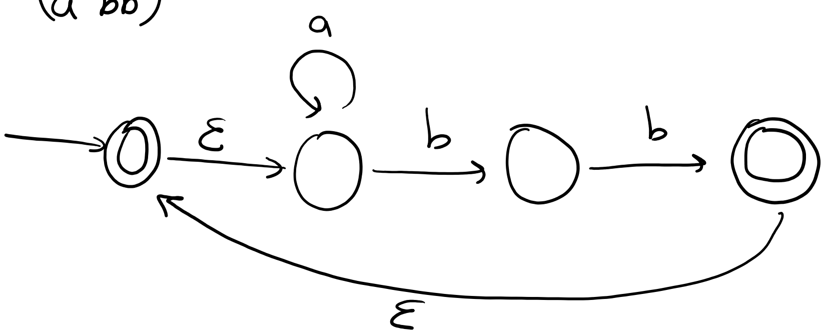




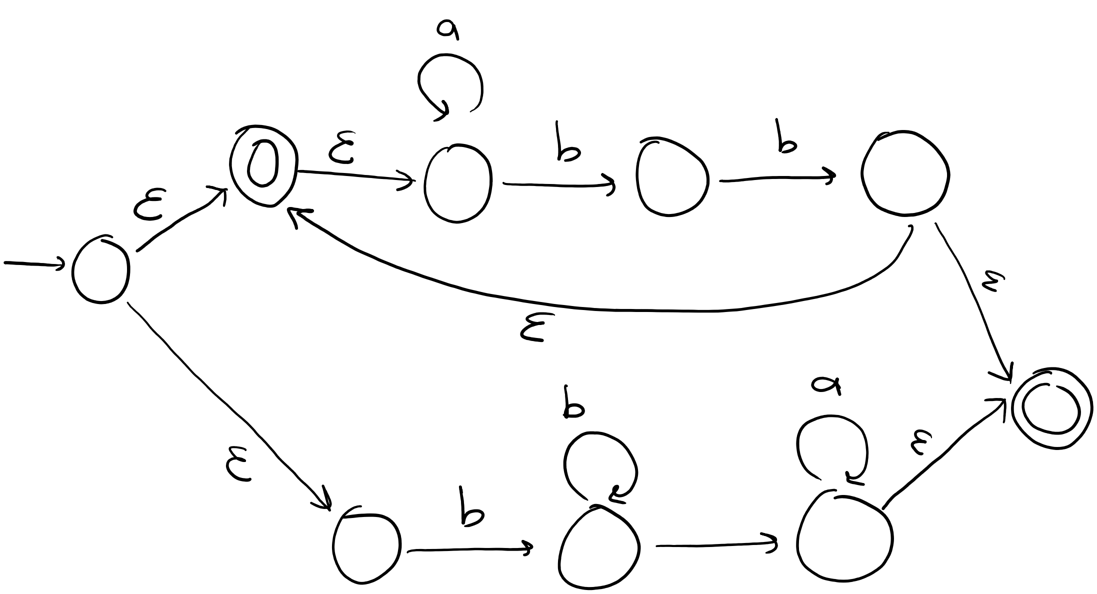
bb^*a^*



$(a^*bb)^*$



$(a^*bb)^* \cup bb^*a^*$



۴.۲ _ قوانین استفاده شده، با شماره ی آن ها در فایل فرستاده شده، ارجاع شده اند.

(a)

$$\text{LHS} \longrightarrow (a+b)^* \overset{14.f}{=\!=} a^*(ba^*)^*$$

$$\overset{a^* = a^*a^*}{=\!=} a^*\left(b(a^*a^*)\right)^*$$

$$\overset{13}{=\!=} a^*\left((ba^*)a^*\right)^*$$

$$\overset{14.f}{=\!=} (a^* + ba^*)^* = \text{RHS} \; . \square$$

(b)

$$\text{LHS} \longrightarrow bb^*(a^*b^* + \varepsilon)b$$

$$\overset{3}{=\!=} bb^*\left[(a^*b^* + \varepsilon)b\right]$$

$$\overset{8}{=\!=} bb^*\left[a^*b^*b + b\right]$$

$$\overset{7}{=\!=} bb^*a^*b^*b + bb^*b$$

$$\overset{b^+ = bb^* = b^*b}{=\!=} bb^*a^*bb^* + bbb^*$$

$$\overset{8}{=\!=} (bb^*a^* + b)\, bb^*$$

5 ↘

$$\overset{b\varepsilon = b \;,\, 7}{=\!=} b(b^*a^* + \varepsilon)\, bb^* = \text{RHS} \; . \square$$

(c)

$$\text{RHS} = (ba)^* ba\, a^* (b^* + \varepsilon)$$

$$= (ba)^* ba \left[a^* (b^* + \varepsilon) \right]$$

$$\overset{8}{=} (ba)^* ba \left[a^* b^* + a^* \right]$$

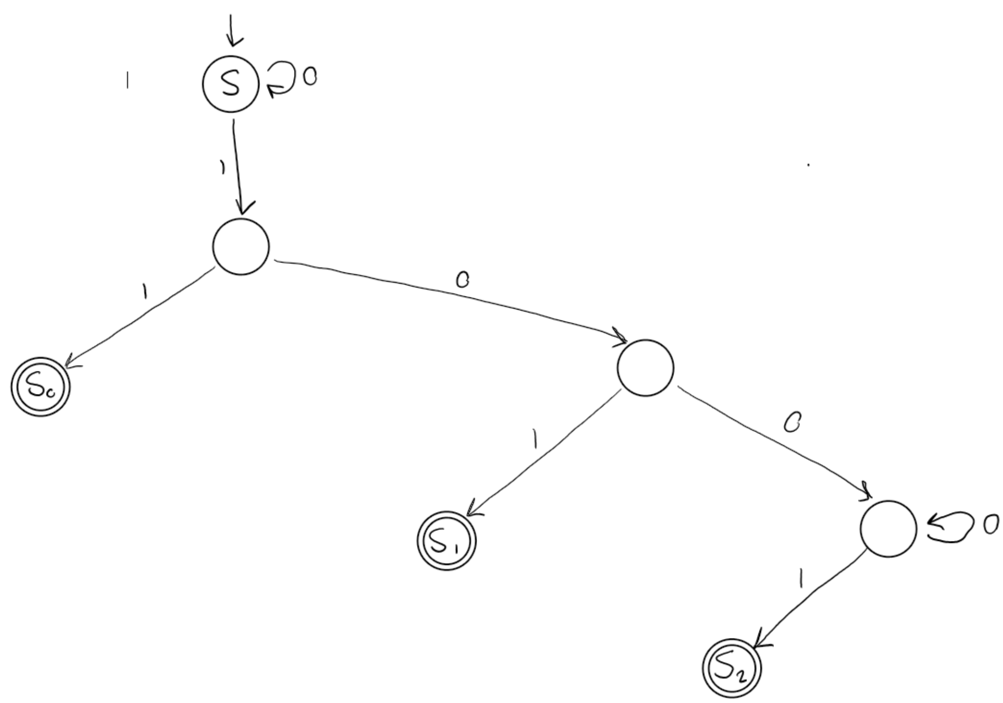
$$\overset{x^* x = x x^*}{=} (ba)(ba)^* (a^* b^* + a^*) = \text{LHS}$$

$$\therefore \square$$

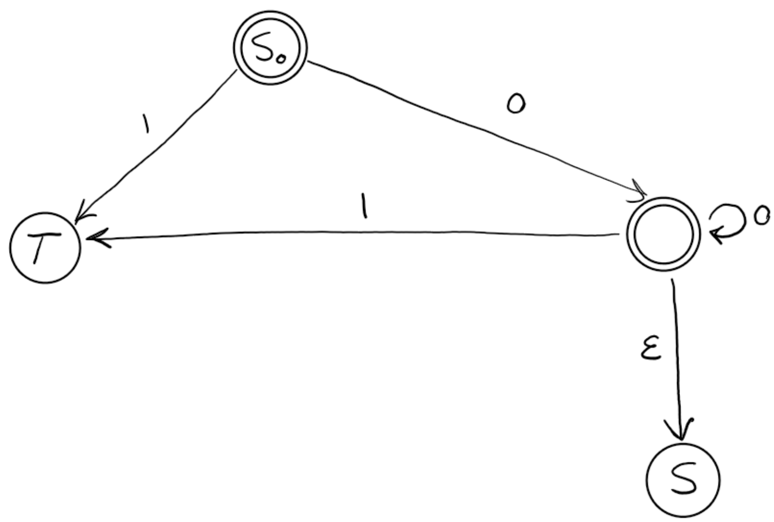
۳. ماشین‌های متناهی توسعه‌یافته (امتیازی)

۱.۳ ماشین متناهی دوجهته

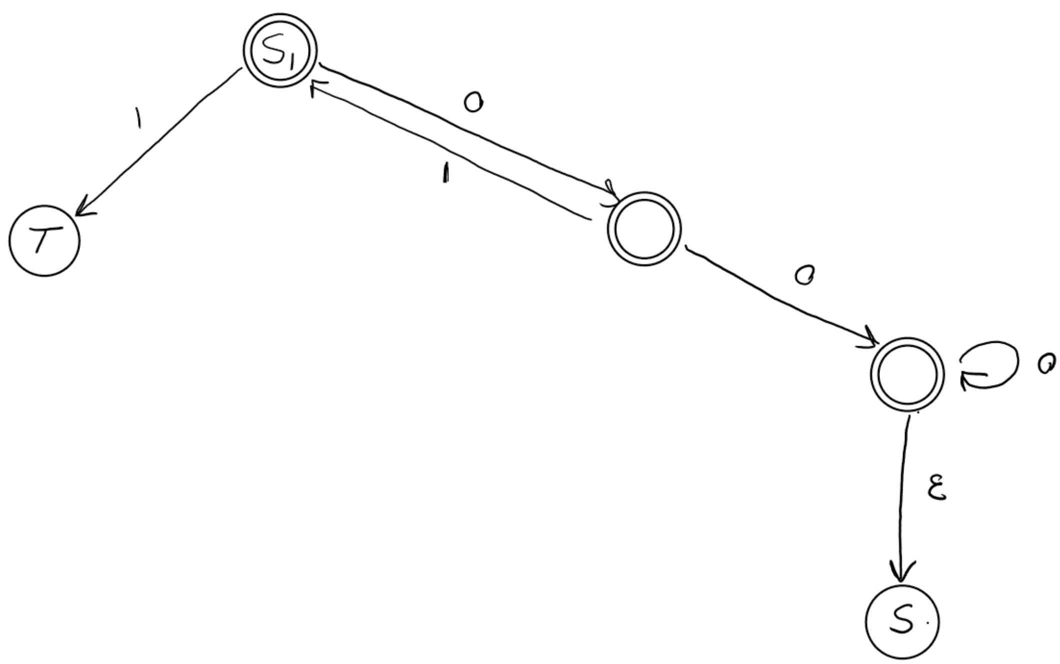
طراحی NFA ما به صورت زیر است:



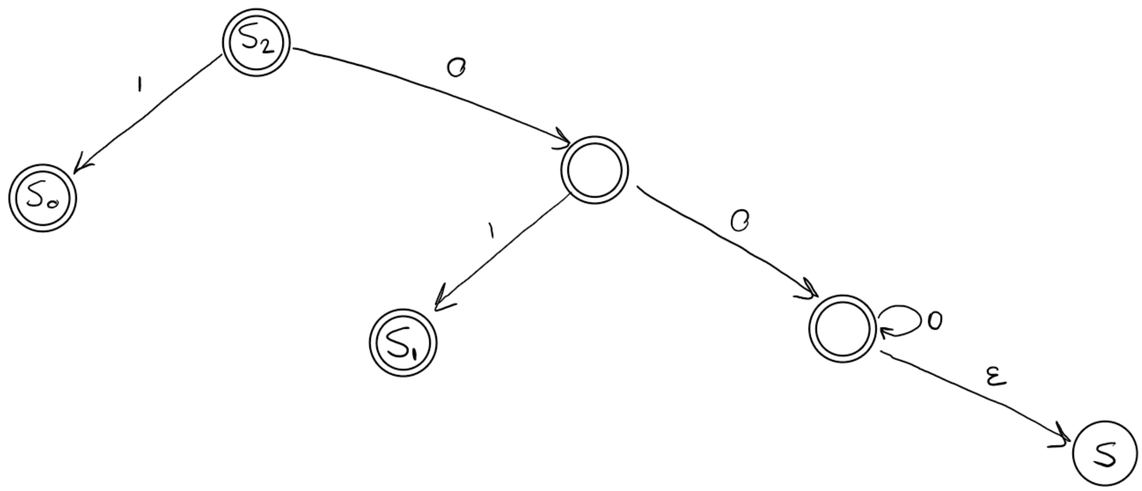
که در آن ادامه استیت S_0 به صورت زیر است:



و ادامه استیت S_1 به صورت زیر است:



و ادامه استیت S_2 به صورت زیر است:



دقت میکنیم که استیت T استیت ریجکت است و با هر ورودی از T به T میرویم. همچنین روش به دست آمدن آن از طریق زبانی که میپذیرد است و برای درک بهتر آن زبان به شکل زیر دقت میکنیم:

که رشته ورودی را بر حسب ۱ های آن جداسازی میکنیم و بین هر دو ۱ متوالی تعداد ۰ های بینشان را x_i میگیریم و چند تا شرط ساده برای پذیرفتن رشته وجود دارد به این صورت که ما در ماشین هر مرتبه ۲ تا از x_i و x_{i+1} را در نظر میگیریم و بنا به حالت های مختلف این ۲ (به فرض ۰ بودن x_{i+1} یا ۱ بودن x_{i+1} و بیشتر مساوی ۲ بودن x_{i+1}) تصمیم میگیریم که آیا ادامه رشته میتواند مورد پذیرش باشد یا نه و یا اینکه استیت بعدی مان چی است را تصمیم میگیریم و اگر دقت شود استیت های S_0 برای حالتی است که $x_i = 0$ و استیت S_1 برای حالتی است که $x_i = 1$ و استیت S_2 برای حالتی است که $x_i \geq 2$ و در ادامه هر استیت حالت های مختلف برای x_{i+1} بررسی شده است. در نتیجه از درستی طراحی مان نیز مطمئن هستیم.

۲.۳ ماشین متناهی دوجهته‌ی علامت‌گذار

در ابتدا برای سادگی نمایش فرض میکنیم $\vdash$ برای شروع رشته استفاده میشود و نماد $\dashv$ براپایان رشته.

همچنین ۲ نماد جدید تعریف میکنیم $\triangleleft$ و $\triangleright$ نماد هایی برای نشان دادن سر و ته یک عملیات در بخشی از یک رشته است.

ابتدا تابع $P : \Sigma \to \Sigma \cup \{\triangleleft, \triangleright\} \cup \Sigma^{\diamond}$ را به صورت زیر تعریف میکنیم:

$$P(a_1a_2...a_n) = a_1...a_n \triangleleft \triangleright a_1^{\diamond}a_2...a_n \triangleleft \triangleright a_1a_2^{\diamond}...a_n \triangleleft ... \triangleright a_1a_2...a_n^{\diamond} \triangleleft \triangleright a_1a_2...a_n$$

که در اینجا داریم که $\Sigma^{\diamond}$ همان مجموعه $a_i^{\diamond}$ ها هستند که $a_i^{\diamond}$ نماد خانه علامت‌گذاشته شده است.

اگر ماشین مان را M بنامیم ما در ابتدا یک ماشین $2DFA$ به نام M' میسازیم که معادل با M است و به دلیل اینکه میدانیم $2DFA$ ها فقط زبان های منظم را میپذیرند آنگاه مسئله حل میشود.

حال اگر ماشین M به صورت زیر تعریف شود:

$$M = \{Q, \Sigma, \delta, q_1, F\}$$

آنگاه M' را به صورت زیر تعریف میکنیم:

$$M' = \{Q', \Sigma', \delta', q_1', F'\}$$

به طوری که $\Sigma' = \Sigma \cup \{\triangleleft, \triangleright\} \cup \Sigma^{\diamond}$ و رشته ω در M پذیرفته است اگر و تنها اگر $P(\omega)$ در M' پذیرفته باشد.

ماشین M' به اینگونه کار میکند که هر گاه خانه s علامت زده شود، آنگاه ماشین M' در بازه s خود که همان زیر رشته $\vdash$ $a_1a_2...a_s^{\diamond}...a_n$ $\dashv$ در $P(\omega)$ است فعالیت میکند و تا زمانی که خانه s علامت زده بماند، ماشین M' نیز در همین زیر رشته میماند.

همچنین در هر لحظه اگر خانه مثل $a \in \Sigma \cup \{\vdash, \dashv\}$ دارای علامت باشد آن را با $a^{\bullet}$ نشان میدهیم.

حال کافیست تابع δ' را تعریف کنیم:

$$\delta'(q,a)=\begin{cases}\text{if } \ \delta(q,a,N)=(q',d,U) \text{ for some } q,q'\in Q,\ a\in\Sigma,\ d\in\{L,R\}:\ \delta'(q,a)=(q',d)\\ \text{if } \ \delta(q,\vdash,N)=(q',d,U) \text{ for some } q,q'\in Q,\ d\in\{R\}:\ \delta'(q,\triangleright)=(q',d)\\ \text{if } \ \delta(q,\dashv,N)=(q',d,U) \text{ for some } q,q'\in Q,\ d\in\{L\}:\ \delta'(q,\triangleleft)=(q',d)\\ \text{if } \ \delta(q,a^\bullet,Y)=(q',d,U) \text{ for some } q,q'\in Q,\ a\in\Sigma,\ d\in\{L,R\}:\ \delta'(q,a^\diamond)=(q',d)\\ \text{if } \ \delta(q,\vdash^\bullet,Y)=(q',d,U) \text{ for some } q,q'\in Q,\ d\in\{R\}:\ \delta'(q,\vdash)=(q',d)\\ \text{if } \ \delta(q,\dashv^\bullet,Y)=(q',d,U) \text{ for some } q,q'\in Q,\ d\in\{L\}:\ \delta'(q,\dashv)=(q',d)\end{cases}$$

برای سادگی توضیحات فرض شده است که نماد $L^\bullet$ برای این است که علاوه بر حرکت اشاره گر به چپ است بلکه علامت ما نیز به خانه چپ حرکت میکند و $R^\bullet$ نیز به همین صورت تعریف میشود.

همچنین برای سادگی فرض شده است در M' میتوان نه به چپ و نه به راست رفت در یک مرحله و برای آن به جاب L و R از $NULL$ استفاده میشود.

دقت میکنیم که اگر ما در یک لحظه خانه s علامتدار باشد و این علامت حذف شود و به خانه راست آن برود، در ماشین M' که تا قبل از این جابه‌جایی طبق توضیحات در زیر رشته s بوده حال باید به زیر رشته $s+1$ برود و در آن زیر رشته به دنبال خانه علامت خورده جدید باشد و پس از پیدا کردن آن به روند عادی خود ادامه می‌دهد.

پس برای حرکت به راست داریم:

$$\delta'(q,a) = \begin{cases} \text{if } \ \delta(q,a^{\bullet},Y) = (q', R^{\bullet}, M) \text{ for some } q,q' \in Q, \ a \in \Sigma, \ : & \begin{cases} q'_R \text{ is a passing state and is} \\ \text{a copy of } q' \\ \delta'(q,a^{\diamond}) = (q'_R, R) \\ \delta'(q'_R, a) = (q'_R, R) \\ \delta'(q'_R, \triangleleft) = (q'_R, R) \\ \delta'(q'_R, \triangleright) = (q'_R, R) \\ \delta'(q'_R, a^{\diamond}) = (q', NULL) \\ \delta'(q'_R, \dashv) = (q', NULL) \end{cases} \\ \text{if } \ \delta(q,\vdash^{\bullet},Y) = (q', R^{\bullet}, M) \text{ for some } q,q' \in Q, \ : & \begin{cases} q'_R \text{ is a passing state and is} \\ \text{a copy of } q' \\ \delta'(q,\vdash) = (q'_R, R) \\ \delta'(q'_R, a) = (q'_R, R) \\ \delta'(q'_R, \triangleleft) = (q'_R, R) \\ \delta'(q'_R, \triangleright) = (q'_R, R) \\ \delta'(q'_R, a^{\diamond}) = (q', NULL) \\ \delta'(q'_R, \dashv) = (q', NULL) \end{cases} \end{cases}$$

همچنین داریم که اگر ما در یک لحظه خانه s علامتدار باشد و این علامت حذف شود و به خانه چپ آن برود، در ماشین M' که تا قبل از این جابه‌جایی طبق توضیحات در زیر رشته s بوده حال باید به زیر رشته $s-1$ برود و در آن زیر رشته به دنبال خانه علامت خورده جدید باشد و پس از پیدا کردن آن به روند عادی خود ادامه می‌دهد.

پس برای حرکت به چپ داریم:

$$\delta'(q,a)=\begin{cases} \text{if } \ \delta(q,a^{\bullet},Y)=(q',L^{\bullet},M) \text{ for some } q,q'\in Q,\ a\in\Sigma,\ : & \begin{cases} q'_L \text{ is a passing state and is} \\ \text{a copy of } q' \\ \delta'(q,a^{\diamond})=(q'_L,L) \\ \delta'(q'_L,a)=(q'_L,L) \\ \delta'(q'_L,\triangleleft)=(q'_L,L) \\ \delta'(q'_L,\triangleright)=(q'_L,L) \\ \delta'(q'_L,a^{\diamond})=(q',NULL) \\ \delta'(q'_L,\dashv)=(q',NULL) \end{cases} \\ \text{if } \ \delta(q,\dashv^{\bullet},Y)=(q',L^{\bullet},M) \text{ for some } q,q'\in Q,\ : & \begin{cases} q'_L \text{ is a passing state and is} \\ \text{a copy of } q' \\ \delta'(q,\dashv)=(q'_L,L) \\ \delta'(q'_L,a)=(q'_L,L) \\ \delta'(q'_L,\triangleleft)=(q'_L,L) \\ \delta'(q'_L,\triangleright)=(q'_L,L) \\ \delta'(q'_L,a^{\diamond})=(q',NULL) \\ \delta'(q'_L,\dashv)=(q',NULL) \end{cases} \end{cases}$$

پس طبق توضیحات بالا میتوان گفت که $Q'=Q\cup Q_L\cup Q_R$ که Q_L و Q_R شامل q'_L و q'_R ها هستند، همچنین داریم که $F'=F$ و $q'_1=q_1$.

حال تعریف ماشین $2DFA$ ما که همان M' است کامل است و کافیست درستی آن را نیز بیان کنیم.

برای اثلات درستی لم زیر را ابتدا اثبات میکنیم:

لم۱. در ماشین M با ورود رشته ω میتوان به استیت q برسد به طوری که اشاره گر به خانه h اشاره میکند و خانه p علامت زده باشد، اگر و تنها اگر با ورودی رشته $P(\omega)$ به ماشین M' به استیت q برسد و اشاره گر به خانه h از زیر رشته p در وردی، اشاره کند.

اثبات:

برای سادگی صرفا کلیت اثبات آمده است.

برای بخش اول اثبات کافیست استقرا بزنیم روی تعداد مراحل حرکت در M و برای بخش دیگر آن نیز استقرا میزنیم روی تعداد دفعاتب که ماشین M' روی یکی از استیت های Q قرار دارد.(طبق تعریفی که از P شده و تابع δ اثبات بدیهی است.)

حال طیق لم داریم که اگر در M با رشته ω با شروع از استیت $q_1 = q_1'$ به یک استیت $q \in F = F'$ برسد آنگاه در ماشین M' با شروع از استیت $q_1 = q_1'$ با رشته $P(\omega)$ به یک استیت $q \in F = F'$ میرسد، همچنین اگر در M' با شروع از $q_1 = q_1'$ با رشته $P(\omega)$ به یک استیت $q_1 = q_1' q \in F = F'$ برسیم به دلیل اینکه طبق تعریف q یک passing through state نمیباشد آنگاه د رماشین M با رشته ω با شروع از استیت $q_1 = q_1'$ به یک استیت $q \in F = F'$ میرسیم، در نتیجه این ۲ ماشین معادل هم هستند و مسئله حل است.