# نظریه زبان ها و ماشین ها

**اتوماتا یک مدل انتزاعی (کلی) است از یک کامپیوتر**
**که بر اساس کاراکتر ورودی**
**وضعیت فعلی و وضعیت حافظه**
**تصمیم میگیرد به چه وضعیتی برود**

مدرس: مهندس مهری رجایی

تهیه و تنظیم : مرتضی سرگلزائی جوان

گروه فن آوری اطلاعات

زمستان ۸۵

**اصطلاحات و تعاریف:**

**کلمه تهی:** $\lambda$ **یا** $\varepsilon$

**مجموعه تمام کلماتی که با a ساخته می شوند:**

$\{a\}^* = \{\varepsilon , a , aa, aaa, \ldots\}$

**مجموعه تمام کلماتی که با a و b ساخته می شوند:**

$\{a,b\}^* = \{\varepsilon, a, aa, aaa, \ldots b, bb, \ldots , ab, abb, \ldots , bbb , \ldots\}$

**در حالتی که تهی عضو مجموعه نباشد:**

$\{a\}^+ = \{ a , aa, aaa, \ldots\}$

**الفبا:** $\Sigma=\{0,1\}$

بنابراین همه کلمات زبان معادل $\Sigma^*$ است.

**مثال۱) تمام کلماتی که به یک ختم می شوند و ما قبل آنها صفر است:**

$L=\{0^*1\}$

**مثال۲) تمام کلماتی که به یک ختم شوند:**

$L=\{(0,1)^*1\}$

**مثال۳) ۰ و ۱ به تعداد مساوی:ϵ**

L={0n1n | n>=0} = {ε,01,0011,000111}

**مثال۴)کلماتی که به تعداد فرد صفر دارند:**

$L=\{0^{2n+1} \mid n\epsilon N\}$

**مثال۵) کلماتی که به صورت متقارن 01..10 هستند:**

$L+\{(01)^n(10)^n \mid n\epsilon N\} = \{\lambda,0110, 01011010, \ldots\}$

---

**نمونه گرامر الفبا:**

<identifier[1]> →[2] <letter><X>
<letter[3]> → a | b | c | d | e | … | z | A | …
<X> → <letter> | <digit>
<digit> → 0 |1 |2 | 3 | … | 9

---

۱ شناسه

۲ قانون

۳ متغییر

**نمونه گرامر زبان پاسکال:**

<program> → begin <statements> end.
<statements> → <statement> ; <statements>

**تعریف گرامر:**

گرامر با یک چهارتایی معرفی می شود:

**G={V,T,S,P}**

V:متغییر

T:پایانه

S:متغییرشروع

P:مجموعه قوانین

**مثال:**

**S → a S b | λ** **G:{{S},{a,b},S,{S→aSb| λ}}**

---

اگر بتوان یک دنباله اشتقاق پیدا کرد به آنچه که بدست می آید کلمه گویند.

S → aSb → aaSbb → aabb

اگر یک کلمه را بصورت الحاق دو کلمه نشان دهیم اولی پیشوند ودومی پسوند است:

{ab,b, λ } پسوند ← **ab** → پیشوند{ λ ,a,ab}

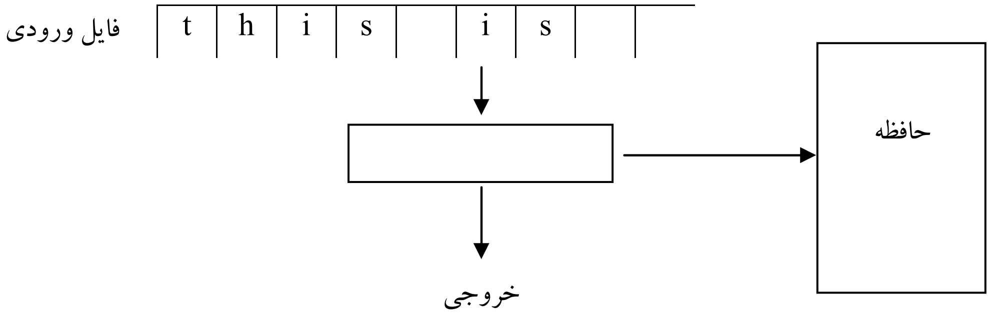
**W=U.V** → {U:Prefix, V:Postfix}

اگر یک کلمه را بصورت الحاق سه کلمه نشان دهیم، وسطی زیر رشته (substring) است.

Pre و post یک نوع خاص از زیر رشته هستند.

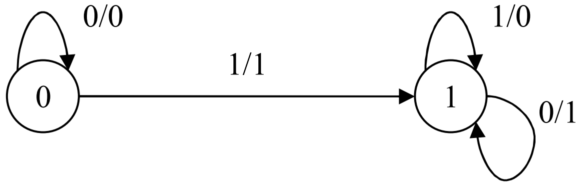
---

**اتوماتا**

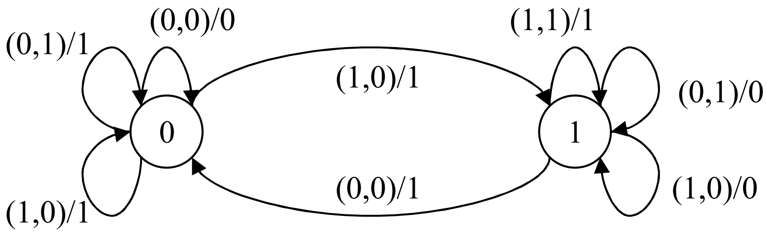


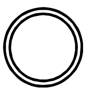
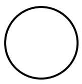
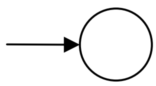

**DFA: براس مسائلی که جواب بله|خیر دارند**

تولید مکمل ۲:

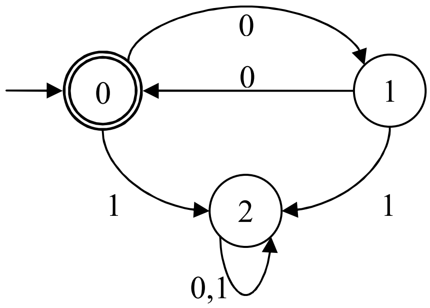



جمع کننده باینری:

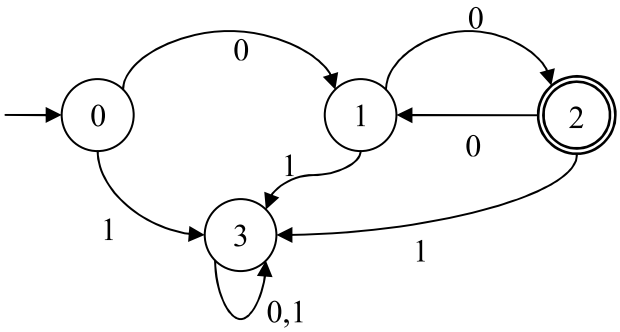



| هدف | عادی | شروع |
|---|---|---|
|  |  |  |

**مثال۱: $(00)^*$**



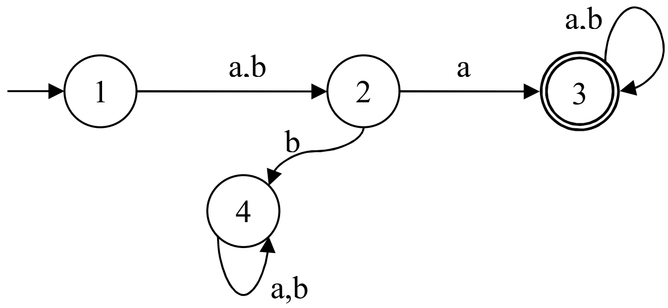
**مثال۲: $(00)^+$**




**مثال۳:**

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

L={w| حرف دوم $a$ باشد }




**نحوه بیان DFA برای مثال فوق :**

**DFA{Q, Σ , $q_0$, F, δ}**

Q: وضعیت ها

Σ:{a,b}

$q_0$: $q_1$

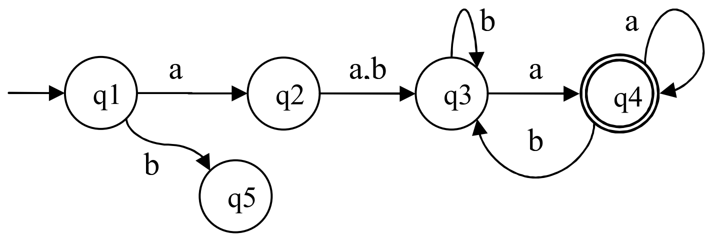
F={$q_3$}

$\delta = Q * \Sigma \rightarrow Q$

**مثال۴)**

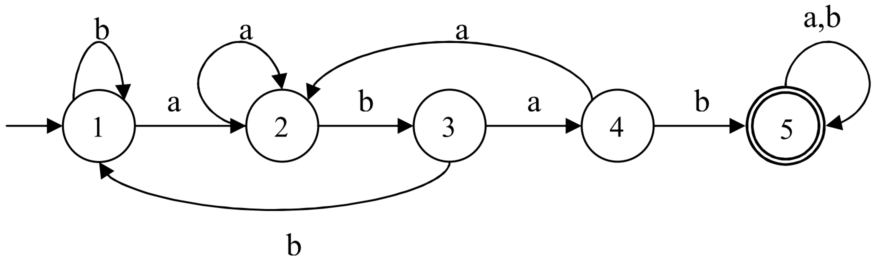
$L = a(a,b)^+a$

$S \rightarrow aAa$
$A \rightarrow aB|bB$
$B \rightarrow \lambda|A$



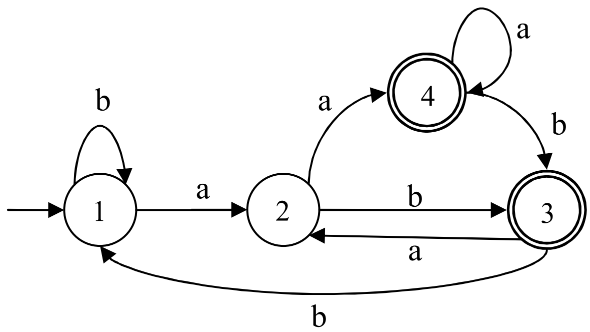

**مثال۵)**

L={w| w رشته هایی که شامل زیر رشته abab باشد}



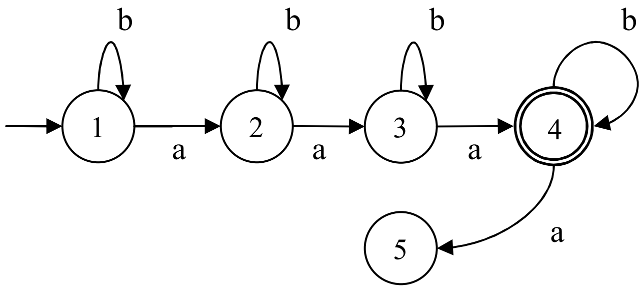
**مثال۶)**

L={w|تمام رشته هایی که حرف دوم از آخر a باشد}



**مثال۷)**

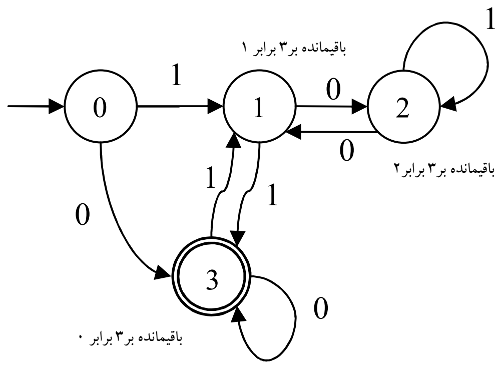
L={w|تمام رشته هایی که حداکثر ۳ تا a داشته باشند}



**مثال۸)**

$\Sigma=\{0,1\}$

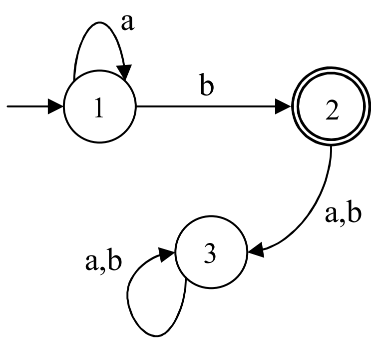
L={w|عدد باینری w بر۳ بخش پذیر باشد}



**زبان منظم (با قاعده):**

اگر DFA ای وجود داشته باشد که تمام رشته های زبان را قبول کند

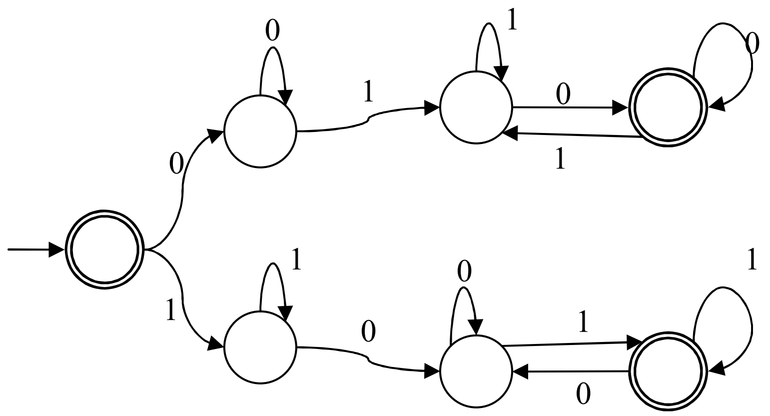
$L=\{a^*b\}$




---

مثال) DFA رسم کنید که زبان زیر را بپذیرد:

$L=\{w|\ n(01)=n(10)\}$ → 101 010 0110 1001




---

**قضیه:** اگر L1 منظم و L2 نیز یک زبان منظم باشد ، آنگاه اجتماع و اشتراک آنها نیز منظم است.

$M_1=\{Q_1,\Sigma,q_{01}, F_1, \delta_1\}$ $\delta_1=Q_1*\Sigma \rightarrow Q_1$

$M_2=\{Q_2,\Sigma,q_{02}, F_2, \delta_2\}$ $\delta_2=Q_2*\Sigma \rightarrow Q_2$

$\mathbf{M_1UM_2 \Rightarrow M=\{Q,\Sigma,q_0, F, \delta\}}$

$Q=Q_1*Q_2$

$\Sigma=\Sigma$

$q_0=(q_{01},q_{02})$

$\delta=Q*\Sigma\rightarrow Q$

$\delta((q_{1i},q_{2j}),a)=(\delta_1(q_{i1},a),\delta_2(q_{j2},a))$

Union: $F=\{(q_i,q_j)|q_i\epsilon F_1 \text{ or } q_j\epsilon F_2\}$

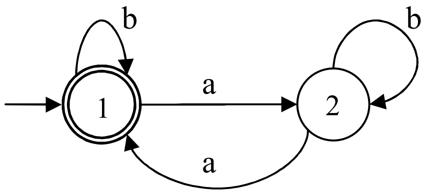
Intersects: $F=\{(q_i,q_j)|q_i\epsilon F_1 \text{ and } q_j\epsilon F_2\}$

---

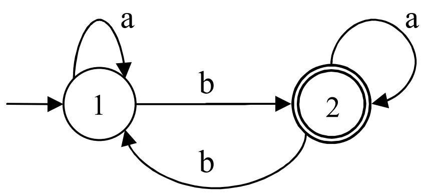
**مثال)**

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

$L_1=\{w\epsilon\Sigma^*$ | دارد a شامل رشته هایی که تعداد زوج w}

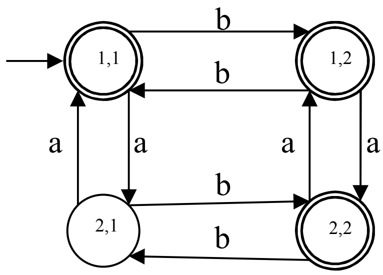



$L_1=\{w\epsilon\Sigma^*$ | w شامل رشته هایی که تعداد b فرد دارد}



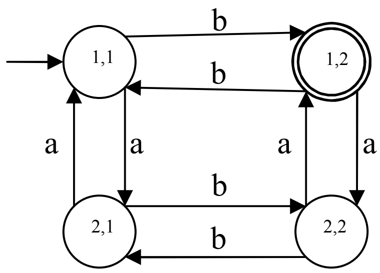

$L_1 \cup L_2$

W شامل رشته هایی که تعداد a زوج یا تعداد b فرد دارد:




$L_1 L_2$

هم تعداد زوج a و هم تعداد فرد b دارد:

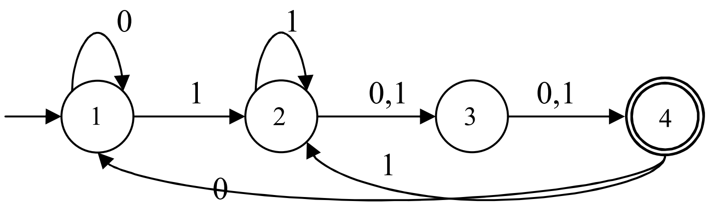



---

**NFA : اتوماتای متناهی نامعین**

**زمانی رشته را قبول می کند که حداقل یک مسیر برای رسیدن به هدف وجود داشته باشد.**

**مثال)** اگر حرف سوم از آخر یک باشد:



**نکته:اگر از λ استفاده شود ، می توان هر یک از وضعیت های دلخواه را انتخاب کرد!**

**نحوه بیان NFA:**

$\{Q, \Sigma, \delta, q_0, F\}$ $\delta(q_i,a)=\{\text{مجموعه}\}$

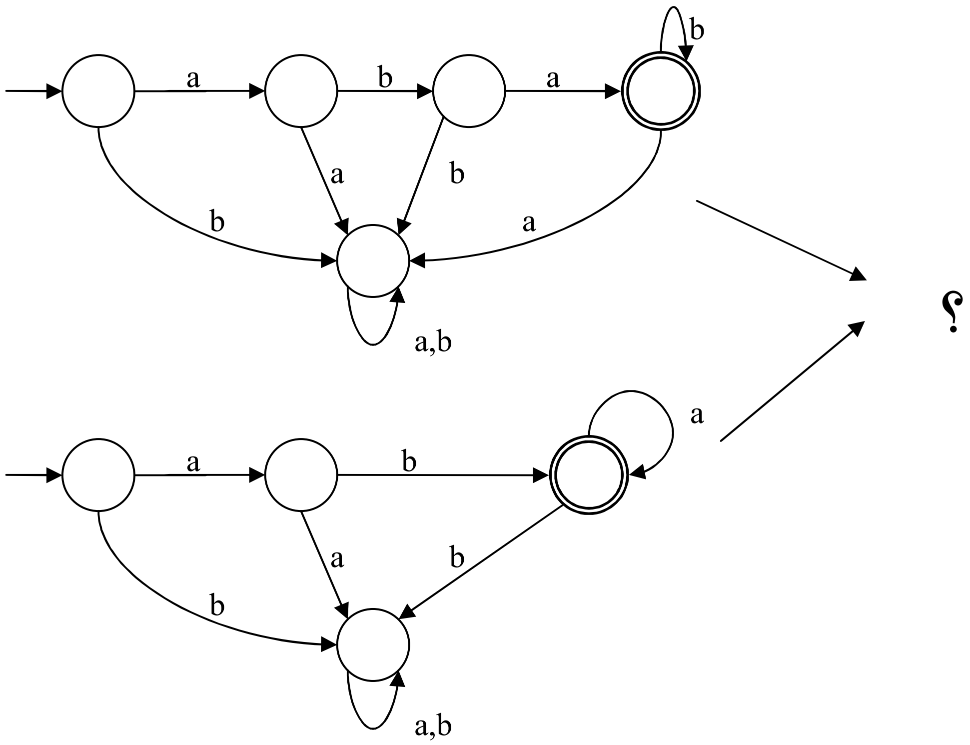
$P=2^Q$

$\delta: Q*(\Sigma U\{\lambda\}) \rightarrow P$

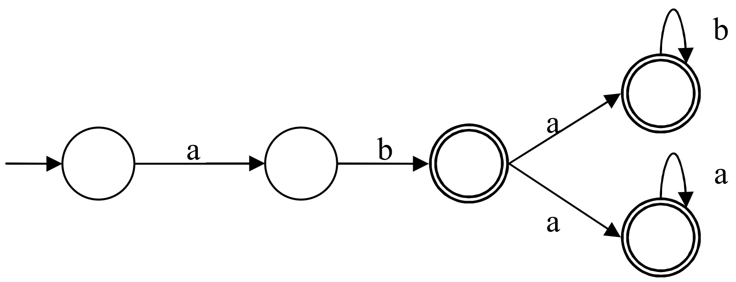
**مثال)**

$\{abab^n \mid n>=0\}$ U $\{aba^n \mid n>=0\}$

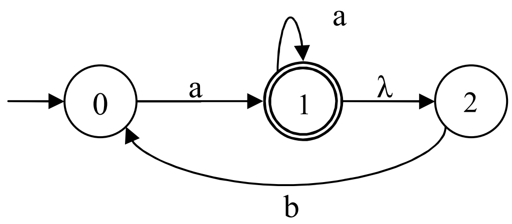
DFA:

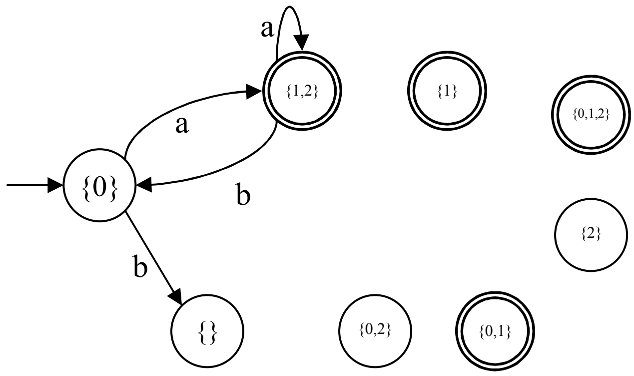



NFA:



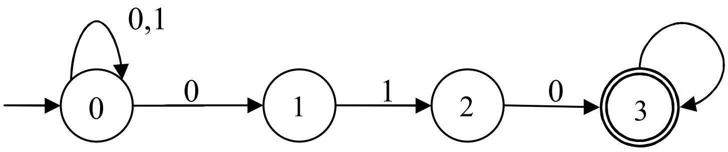
**تبدیل NFA به DFA:**

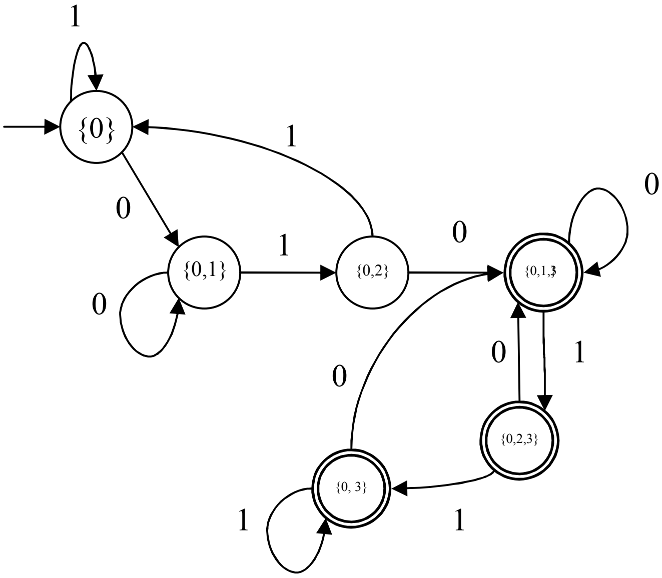



---

**مثال۲)**




**نکته: پس اگر بتوان یک NFA هم برای یک زبان طراحی کرد ، اثبات می شود که منظم است.**

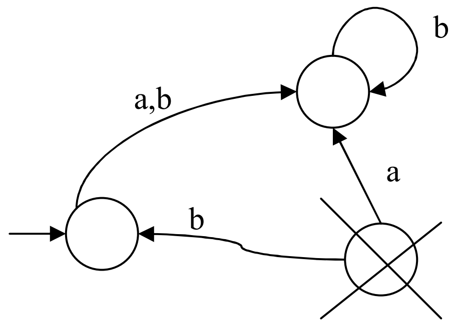
**تمرین)ثابت کنید اگر A منظم باشد $A^R$ نیز منظم است.**

$Q' = Q$
$\Sigma' = \Sigma$
$\delta' = \delta^{-1} \cup \{q_0'\} * \{\lambda\} * F$
$q_0' = q_0$
$F' = \{q_0\}$

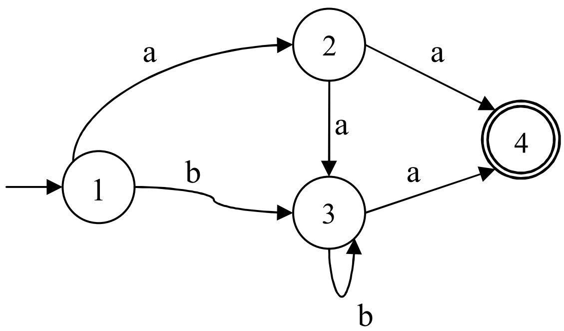
> **نکته:**
> **یک DFA فقط یک زبان را قبول می کند ولی یک زبان ممکن است چند DFA را قبول کند.**

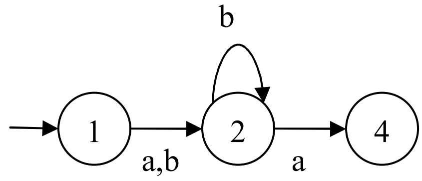
**ساده سازی DFA:**

۱- تمام گره هایی که برای آنها مسیری از گره‌ی شروع وجود ندارد را حذف می کنیم.




۲- ادغام وضعیت های تکراری

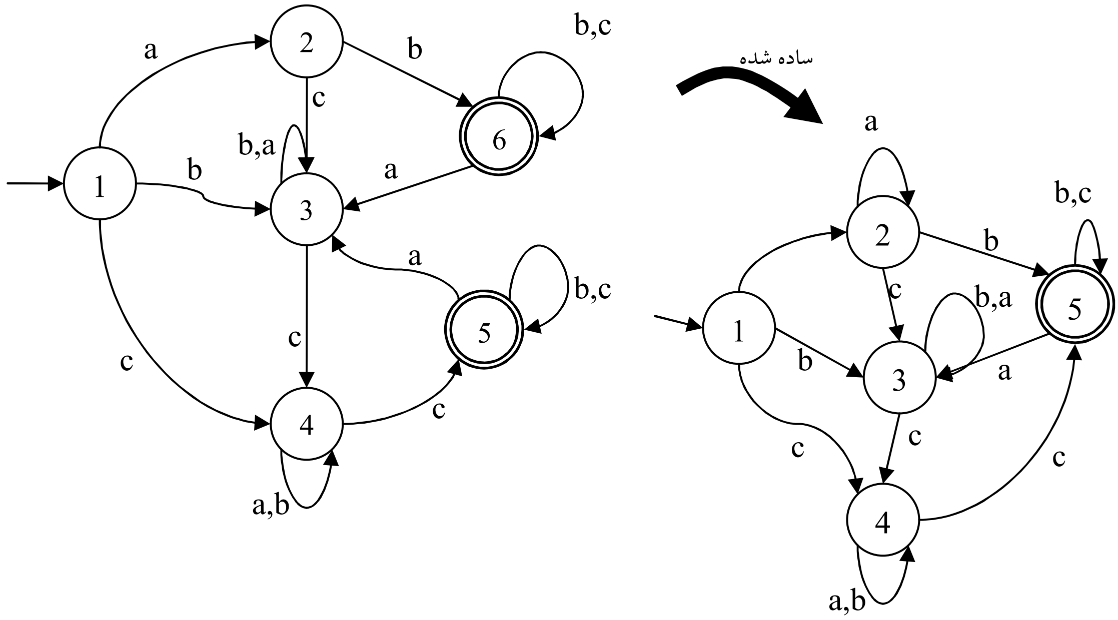



**روش ادغام:(گره های ۲ و ۳ در مثال قبل ادغام شده اند)**

| | | | | |
|---|---|---|---|---|
| **4** | | | | √ |
| **3** | | | √ | |
| **2** | | √ | √ | |
| **1** | √ | 2,4 | 2,4 | |
| | **1** | **2** | **3** | **4** |

**مثال۲)**




| | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| **5** | | | | | **6,5** |
| **4** | | | | | |
| **3** | | | 4,5 | | |
| **2** | | 6,3 | 6,4 | | |
| **1** | 6,3 | 2,3 | 4,5 | | |

**عبارت منظم → زبان منظم**

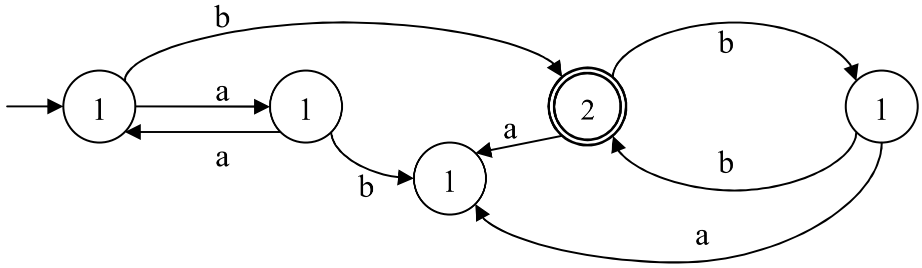
**انواع مختلف عبارت منظم:**

۱- λ

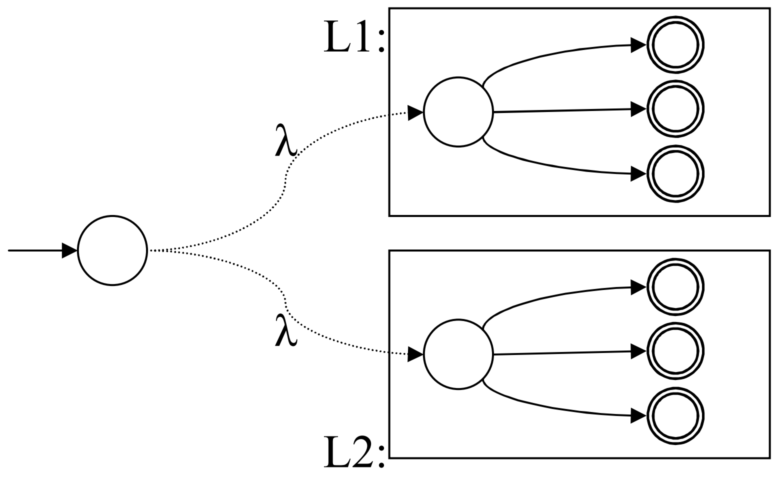
۲- **A**

۳- $r_2 , r_1$ ← **r1*** **(r1)** **r1.r2** **r1Ur2**

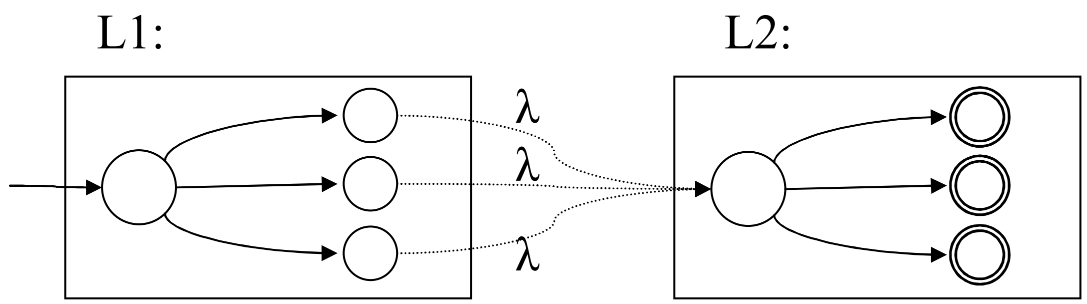
**مثال)** $(a.a)^*(bb)^*b$



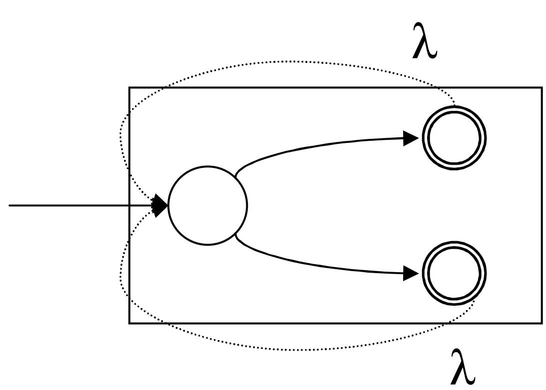
**اگر L1 یک زبان منظم و L2 یک زبان منظم باشد L1UL2 نیز منظم است.**



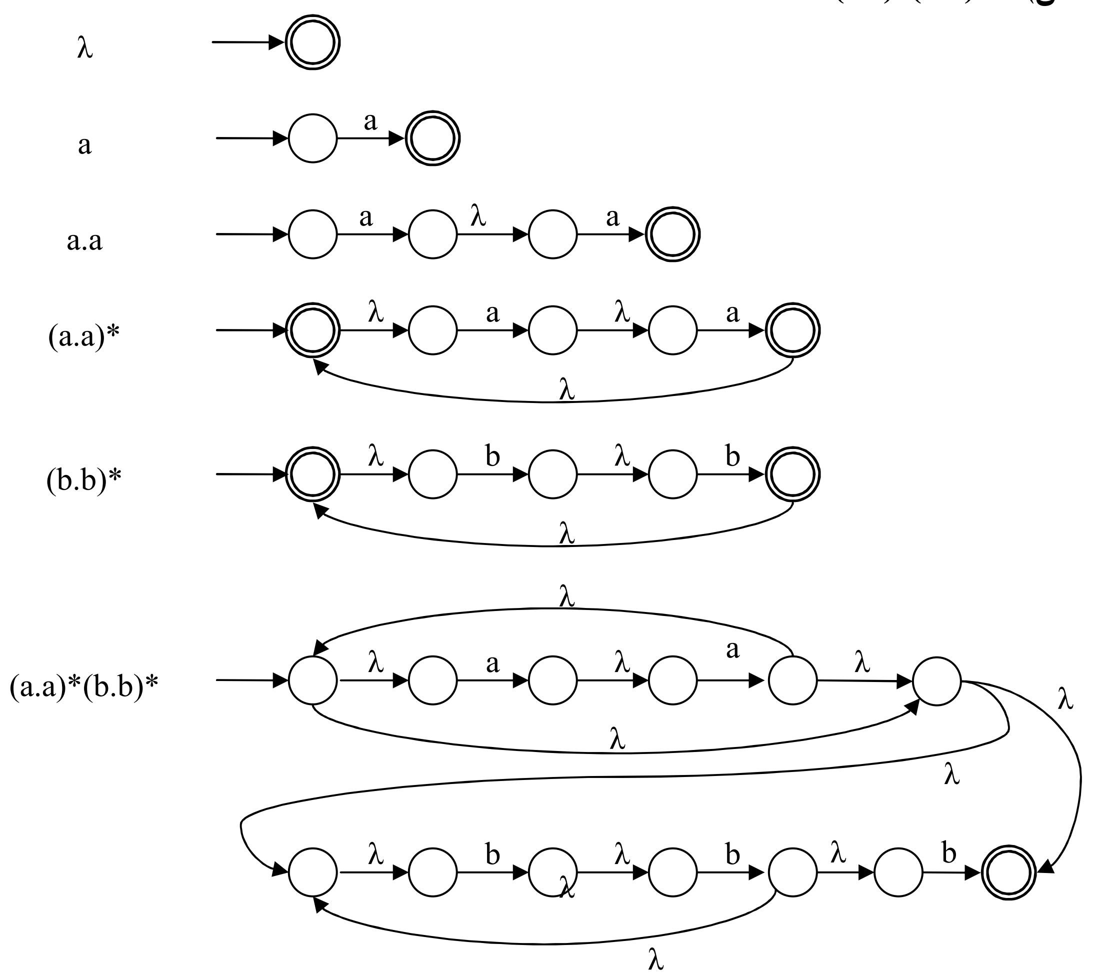

**اگر L1 یک زبان منظم و L2 یک زبان منظم باشد L1.L2 نیز منظم است.**




**اگر L1 یک زبان منظم و L2 یک زبان منظم باشد *L1 نیز منظم است.**

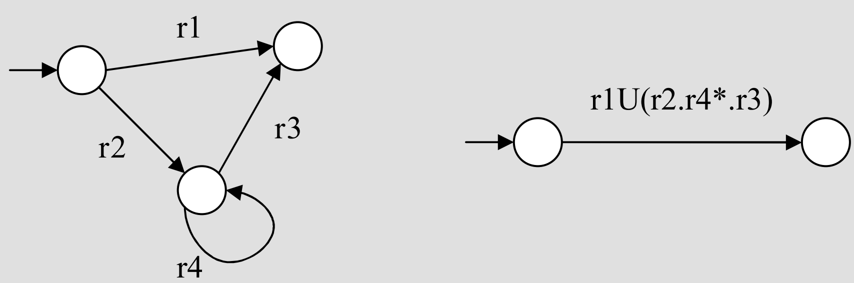


**مثال) $(a.a)^*(b.b)^*b$**



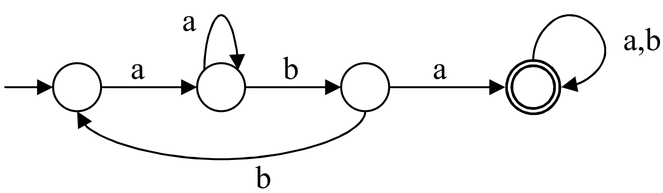

**بدست آوردن عبارت منظم یک زبان منظم (با توجه به DFA):**

**DFA → GNFA (برچسب یالها عبارت منظم است)**

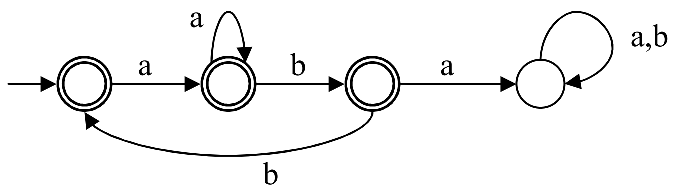



**مثال)زبانی که شامل زیر رشته aba نباشد: (معادل معکوس زبان aba)**

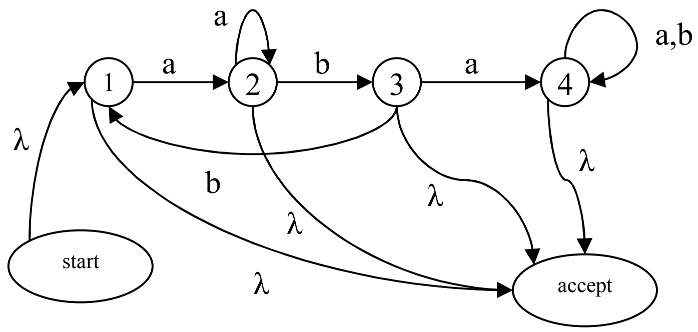
- زبان پذیرنده aba

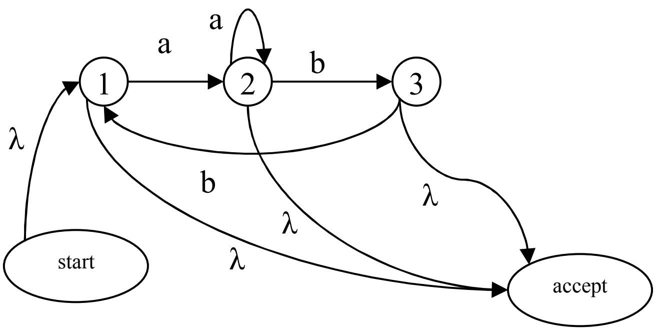


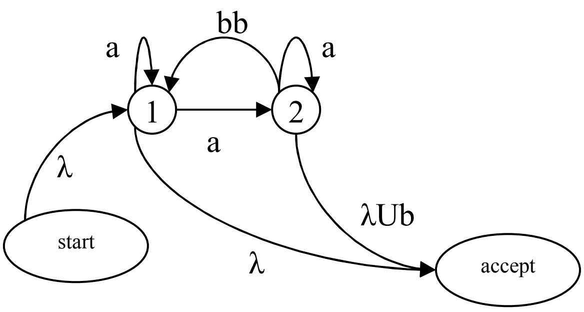
- معکوس زبان aba

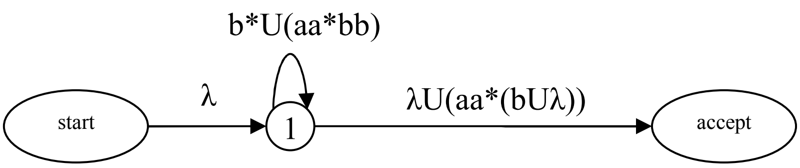


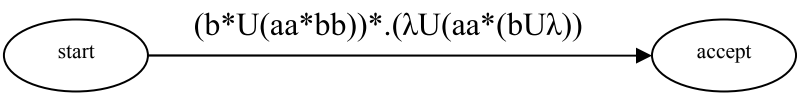

**بدست آوردن عبارت منظم:**



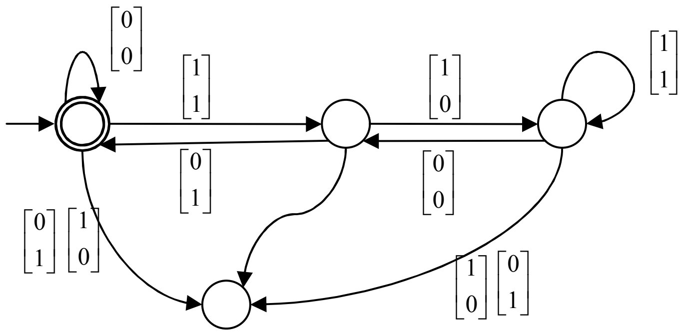



**تمرین) رشته هایی که ردیف بالایی ۳ برابر ردیف پایینی باشد (با توجه به الفبای زیر):**

$$\Sigma_2 = \{\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1\\0\end{bmatrix},\begin{bmatrix}0\\1\end{bmatrix},\begin{bmatrix}1\\1\end{bmatrix}\}$$

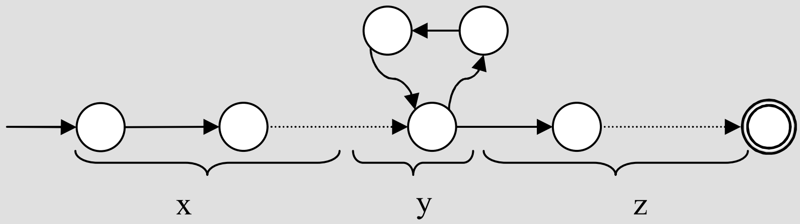
توضیح: اعداد را از بایت کم ارزش وارد می کنیم(بصورت معکوس)



**اگر زبانی قابل شمارش باشد منظم است.**

**اثبات از طریق لم تزریق(pumping):**

**فرض می کنیم زبان منظم است و می توان برای آن یک DFA رسم کرد. پس می توان آن را به سه قسمت xyz تقسیم کرد.**




**مثال۱)**

یک DFA وجود دارد که این زبان را بپذیرد → فرض خلف منظم → **$L=\{WW^R\}$**

تعداد وضعیت های این DFA m است: $a^mb^mb^ma^m \in L$

$$|xy|<=m$$
$$|y|>=1$$

$$x=a^{m-k}$$
$$y=a^k \quad m>k>=1$$
$$z=b^mb^ma^m$$

$xy^0z = w_0 = xz = a^{m-k}b^mm^ma^k$ → متقارن نیست → L منظم نیست

---

**مثال۲)** **$L=\{w \mid n_a(w) <> n_b(w)\}$**

برای اثبات منظم نبودن این مثال ، ثابت می کنیم که عکس زبان فوق منظم نیست ، یعنی حالتی که تعداد a و b برابر ستند (مانند مثال قبل)

**مثال۳)** ثابت کنید که L منظم نیست:

$$\Sigma_2 = \{\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1\\0\end{bmatrix},\begin{bmatrix}0\\1\end{bmatrix},\begin{bmatrix}1\\1\end{bmatrix}\}$$

L={Σ2| رشته بالایی معکوس رشته پایینی است}

$$\begin{bmatrix}0\\0\end{bmatrix}^p\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix}^p$$

**|xy|<=p**
**|y|>=1**

$$\mathbf{x}=\begin{bmatrix}0\\0\end{bmatrix}^{p-k}$$

$$y=\begin{bmatrix}0\\0\end{bmatrix}^k$$

$$z=\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix}^p$$

$$i=0 \rightarrow xz \rightarrow \begin{bmatrix}0\\0\end{bmatrix}^{p-k}\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix}^p \notin L$$

**برای هر زبان منظم یک گرامر وجود دارد.**

**(V,T,S,P)**
**a* → S → aS|λ**

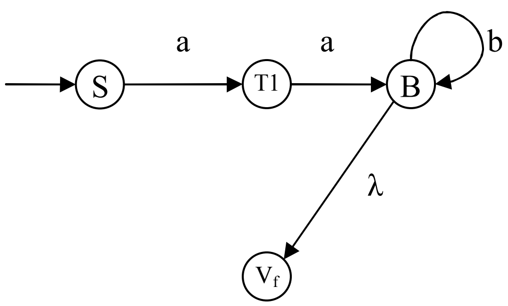
**گرامر خطی : اگر سمت راست قوانین حداکثر یک متغیر بکار رود**

- **خطی سمت راست S→abcA**
- **خطی سمت چپ S→Aabc**

**قضیه ۱: زبان هر گرامر خطی راست یک زبان منظم است.**

**مثال)**

**S→ aaB**
**B → bB|λ**



**قضیه۲: برای هر زبان منظم ، یک گرامز خطی راست وجود دارد.**

اثبات← رسم DFA و نوشتن گرامر از روی آن

$M=\{Q,\Sigma,q_0,F,\delta\}$

$G=\{V,S,P,T\}$
$T=\Sigma U\{\lambda\}$
$V=Q$
$S=q_0$
$P: \delta(q_i, a)=q_j \Rightarrow \quad Q_i \to aQ_j$

به ازای هر $\delta$ یک قانون در P تعریف می کنیم ، وضیت های نهایی را هم به قوانین اضافه می کنیم

If $q_i \epsilon F \Rightarrow Q_i \to \lambda$

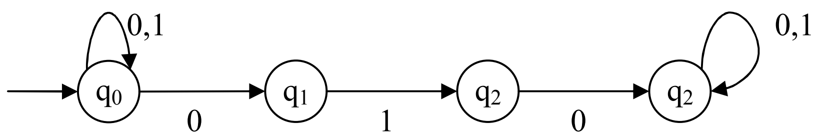
**نکته: این قضیه برای NFA نیز صادق است:**

$\delta(q_i , a) = \{q_i \ldots q_k\}$

$Q_i \to aQ_L$
$Q_i \to aQ_m$
$\ldots$
$Q_i \to aQ_k$

---

**مثال) زبانی که شامل زیر رشته ۰۱۰ باشد:**




$Q_0 \to 0Q_0 \mid 1Q_0 \mid 0Q_1$
$Q_1 \to 1Q_2$
$Q_2 \to 0Q_3$
$Q_3 \to 0Q_3 \mid 1Q_3 \mid \lambda$

$1101 \to Q_0 \to 1Q_0 \to 11Q_0$
$\to 110\ Q_0 \to 1101Q_0 \ldots$
$\to 110\ Q_1 \to 1101Q_2$

---

**گرامر خطی چپ معکوس یک زبان را می پذیرد.**

$L(G) \qquad\qquad L(G')$
$V_i \to V_j\ t_1 \ldots t_n \quad \Rightarrow \quad V_i' \to t_n \ldots t_1 V_j'$

$L(G) = L^R(G')$

---

**مثال:**

| گرامر | | معکوس |
|---|---|---|
| $S \to A1$ | $\Rightarrow$ | $S \to 1A$ |
| $A \to A0 \mid \lambda$ | $\Rightarrow$ | $A \to 0A \mid \lambda$ |

---

**تشخیص چگونگی تساوی دو زبان منظم:**

زبان منظم نسبت به اجتماع ، اشتراک ، الحاق ، star ، متمم و معکوس بسته است

اگر برابر باشند $(L1-L2) \cup (L2-L1) = \lambda$

$\rightarrow (L1 \cap L2') \cup (L2 \cap L1')$

اگر نتیجه ، وضعیت نهایی نداشته باشد ، یا اگر داشته باشد ولی مسیری به آن نباشد ، معادل تهی خواهد بود

**بنابراین زبانهای منظم نسبت به عمل تفاضل نیز بسته هستند**

---

می توان یک رشته را توسط یک تابع $h(\sigma)=w$ و $(h:\Sigma \rightarrow \Sigma')$ از یک زبان به زبان دیگر نگاشت کرد.

| σ | h(σ) |
|---|---|
| a | cdd |
| b | eee |

$L=\{w\epsilon\Sigma$ | تعداد a زوج باشدو aهاقبل از bبیاید } $\rightarrow$ $(aa)^*b^*$ $-h(\sigma)\rightarrow$ $(cddcdd)^*(eee)^*$

---

**عملگر تقسیم:**

**$L1/L2 \rightarrow \{x|xy\epsilon L1 , y\epsilon L2\}$**

**مثال)**

**1) L1= a*b*  L2=b  L1/L2 = a*b***

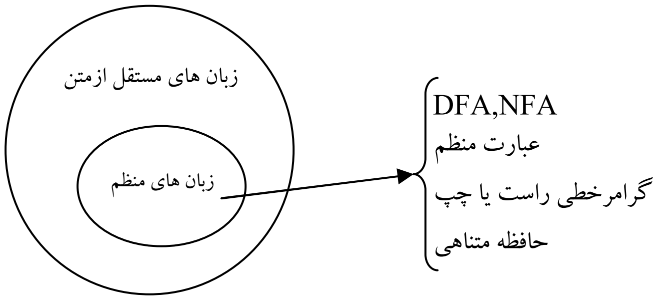
**2) L1= a*b  L2=b  L1/L2 = a***

**3) L1= a*  L2=b  L1/L2 = {}**

**4) L1= b*  L2=b*  L1/L2 = b***

**5) L1 = b  L2=b  L1/L2 = λ**

## زبان های مستقل از متن (C.F.L)
## Context-Free Languages




زبانی مستقل از متن است که برای آن یک گرامر مستقل از متن (CFG) وجود داشته باشد:

**{V,T,S,P}**
$A \in V$
$A \rightarrow T$
$A \rightarrow (V \cup T)^*$

---

مثال ۱)

1) $a^n b^n$ $\quad\quad$ $S \rightarrow aSb \mid \lambda$

2) $WW^R$ $\quad\quad$ $S \rightarrow aSa \mid bSb \mid \lambda$

---

مثال۲)گرامر زیر تمام عبارات ریاضی را تولید می کند

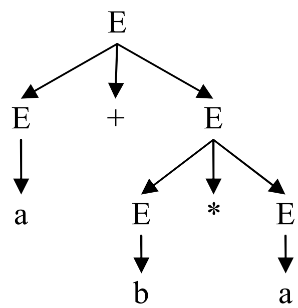
**E ➔ E+E | E*E | (E) | a | b**

مثلا برای تولید a * b + a می توان یکی از دنباله های اشتقاق زیر را طی کرد:

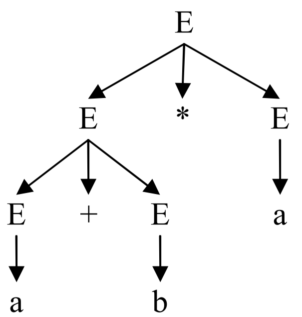
1) E ➔ E + E ➔ a + E ➔ a + E * E ➔ a + b * E ➔ a + b * a

2) E ➔ E * E ➔ E + E * E ➔ a + E * E ➔ a + b * E ➔ a + b * a

تفاوت دو دنباله فوق در این است که در حالت ۱ ، ابتدا ضرب و سپس جمع انجام میشود ، اما در حالت ۲ ابتدا جمع و سپس ضرب انجام میشود.




1




2

**نکته: به گرامری که دو درخت اشتقاق متفاوت (یا دو دنباله اشتقاق چپ) برای یک رشته داشته باشد ، گرامر مبهم گویند. (اگر برای زبانی نتوان گرامر غیرمبهم تعریف کرد ، آنگاه ذاتاً مبهم است)**
**برای رفع ابهام در گرامر قبل ، می تواند اولویت بندی نمود:**

$$E \rightarrow E+T \mid T$$
$$T \rightarrow T*T \mid a \mid b$$

مثلا دنباله اشتقاق برای a+b*c بصورت زیر است:

$$E \rightarrow E + T \rightarrow T + T \rightarrow a + T \rightarrow a + T * T \rightarrow a + b * T \rightarrow a + b * a$$

---

تمرین) برای عبارات زیر گرامر مستقل از متن بنویسید

1) $a^nb^n$

$S \rightarrow aSb \mid \lambda$

2) $a^nb^m$ $n<>m$

$S \rightarrow AS_1 \mid S_1B$
$A \rightarrow aA \mid a$
$B \rightarrow bB \mid b$
$S_1 \rightarrow aS_1b \mid \lambda$

3) $a^nb^mc^{n+m}$

$S \rightarrow aSc \mid T$
$T \rightarrow bTc \mid \lambda$

4) $a^nb^na^mb^m$

$S \rightarrow AA$
$A \rightarrow aAb \mid \lambda$

5) $a^nb^na^mb^m$ $n=m$ زبان مستقل از متن نیست(متمم آن نیز مستقل از متن نیست)

6) $a^nb^ma^mb^n$

$S \rightarrow aSb \mid A$
$A \rightarrow bAa \mid \lambda$

7) $a^nb^m$ $n<=m+3$

روش یک

$S \rightarrow a \mid aa \mid aaa \mid \lambda \mid aSb \mid SB$
$B \rightarrow bB \mid b$

روش دو

$S \rightarrow aS_1M \mid S_3$
$S_1 \rightarrow aS_2M \mid S_3$
$S_2 \rightarrow aS_3M \mid S_3$
$S_3 \rightarrow aS_3b \mid M$
$M \rightarrow Mb \mid \lambda$

8) $a^n b^m c^k$ $k<>n+m$ متمم قسمت ۳ $(k>n+m \text{ or } k<n+m)$

$S \rightarrow S_1 \mid S_2 \mid S_3$

$k = n+m$

$S_0 \rightarrow aS_0c \mid T$

$T \rightarrow bTc \mid \lambda$

$k>n+m$

$S_1 \rightarrow S_0C$

$C \rightarrow cC|c$

$k<n+m$ با a بیشتر

$S_2 \rightarrow AS_2 \mid AS_2C \mid T_1$

$A \rightarrow aA \mid a$

$k<n+m$ با b بیشتر

$S_3 \rightarrow aS_3c \mid AT_1$

$T_1 \rightarrow bT_1c \mid bT_1 \mid b$

9) $a^n b^m$ $2n<=m<=3n$

$S \rightarrow aSbb \mid aSbbb \mid \lambda$

**آیا رشته جزء زبان است؟**

$A \rightarrow B$ قوانین یکه

$B \rightarrow A$

$S \rightarrow \lambda$ قوانین $\lambda$

**با توجه به وجود قوانین فوق به الگوریتم با پیچیدگی زیر نیاز است (بدترین حالت)**

$|p| + |p|^2 + \dots + |p|^{2|w|}$

**برای کاهش پیچیدگی ، نیاز به بهینه سازی است:**

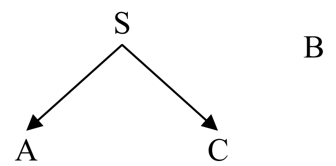
- **حذف قوانین بلااستفاده**
- **حذف متغییرهای غیرقابل تولید**

مثال ۱)

**$T = \{S \rightarrow aS \mid A \mid C \text{ , } A \rightarrow a, B \rightarrow aa, C \rightarrow aCb\}$**

متغییرهای Bو C در قوانین فوق بلااستفاده هستند:

(در بررسی از اول به آخر هیچگاه به B نمی رسیم. در بررسی از آخر به اول ، هیچگاه از C خارج نمیشویم!)



**الگوریتم تشخیص متغییرهای غیرقابل تولید (بررسی از آخر به اول)**

$V_1=\{\}$
Repeat next step until not change V1
-for each $A\epsilon V$ , $A\rightarrow X_1X_2\ldots X_n$ $X_i\epsilon(V_1UT)^*$
Add A to $V_1$
- $P_1$ Include rules of P that every of Symbol rules exist in $(V_1UT)^*$

**مثال)**

V={A,B,S,C}
T{a,b,c}
$S \rightarrow aSb \mid A \mid BB \mid C$
$A \rightarrow aA \mid B$
$B \rightarrow bB$
$C \rightarrow Sc|c$

$P_1=\{c\rightarrow c , S\rightarrow aSb , S\rightarrow C , C\rightarrow Sc\}$

---

**مثال) حذف قوانین λ:**

$S\rightarrow AbaC$
$A\rightarrow BC$
$B\rightarrow b|\lambda$
$C\rightarrow D|\lambda$
$D\rightarrow d$

بعد از حذف بصورت زیر در می آید:

$S\rightarrow AbaC \mid Aba \mid AaC \mid BaC \mid Aa \mid Ba \mid ac \mid a$
$A\rightarrow BC \mid B \mid C$
$B\rightarrow b$
$C\rightarrow D$
$D\rightarrow d$

---

**مثال)حذف قوانین یکه:**

$S\rightarrow AA \mid B$
$B \rightarrow A$
$A\rightarrow a$

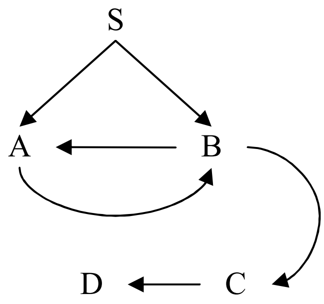
بعد از حذف بصورت زیر در می آید:

$S\rightarrow AA|a$
$B\rightarrow a$
$A\rightarrow a$

---

**مثال)**

$S \rightarrow a|aA|B|C$
$A \rightarrow aB|\lambda$
$B \rightarrow aA$
$C \rightarrow cCD$
$D \rightarrow ddd$




۱)حذف قوانین بلااستفاده:

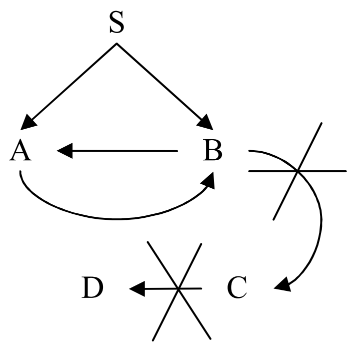
$S \rightarrow a|aA|B$
$A \rightarrow aB|\lambda$
$B \rightarrow aA$
$C \rightarrow cCD$
$D \rightarrow ddd$

۲)حذف قوانین غیرقابل تولید:

$V_1=\{S,D,A,B\}$
$T=\{a,c,d\}$
$P_1=\{S \rightarrow a,\ D \rightarrow ddd,\ A \rightarrow \lambda,\ S \rightarrow aA,\ B \rightarrow aA,\ A \rightarrow aB\ ,\ S \rightarrow B\}$

۳)

$S \rightarrow a|aA|B$
$A \rightarrow aB|\lambda$
$B \rightarrow aA$




۴)حذف قوانین یکه:

$S \rightarrow a \mid aA$
$A \rightarrow aB \mid \lambda$
$B \rightarrow aA$

---

**مثال)**

$S \rightarrow AaB \mid aaB$
$A \rightarrow \lambda$
$B \rightarrow bbA|\lambda$

**حذف قوانین $\lambda$:**

1) $S \rightarrow AaB \mid Aa \mid aB \mid a \mid aaB \mid aa$
$B \rightarrow bbA \mid bb$

2) $S \rightarrow aB \mid a \mid aaB \mid aa$
$B \rightarrow bb$

**شکل نرمال چامسکی (Chomsky)**

**قالب کلی:**

**$A \rightarrow BC$**
**$A \rightarrow a$**

**ابتدا می بایست قوانین بلااستفاده ، یکه و $\lambda$ حذف شود.**

مثال۱) بصورت فرم چامسکی بنویسید:

$S \rightarrow aSaA \mid A$
$A \rightarrow abA \mid b$

۱) حذف قوانین یکه:

$S \rightarrow aSaA \mid abA \mid b$
$A \rightarrow abA \mid b$

۲) فرم نرمال چامسکی

$S \rightarrow B_aSB_aA \mid B_aB_bA \mid b$
$A \rightarrow B_aB_bA \mid b$
$S \rightarrow B_aD_1 \mid B_aD_2 \mid b$
$D_1 \rightarrow SD_3$
$D_2 \rightarrow B_bA$
$D_3 \rightarrow B_aA$
$A \rightarrow B_aD_2 \mid b$

---

مثال۲) به فرم نرمال چامسکی بنویسید:

$A \rightarrow BAB \mid B \mid \lambda$
$B \rightarrow 00 \mid \lambda$

1) $S \rightarrow A \mid \lambda$
$A \rightarrow BAB\lambda BB\lambda B$
$B \rightarrow 00 \mid \lambda$

2) $S \rightarrow A \mid \lambda$
$A \rightarrow BAB \mid BA \mid AB \mid A \mid BB \mid B$
$B \rightarrow 00$

3) $S \rightarrow BAB \mid BA \mid AB \mid BB \mid 00 \mid \lambda$
$A \rightarrow BAB \mid BA \mid AB \mid BB \mid 00$
$B \rightarrow 00$

4) $S \rightarrow BD_1 \mid BA \mid AB \mid BB \mid B_0B_0 \mid \lambda$
$D_1 \rightarrow AB$
$B_0 \rightarrow 0$
$A \rightarrow BD_1 \mid BA \mid AB \mid BB \mid B_0B_0$
$B \rightarrow B_0B_0$

**الگوریتم با پیچیدگی $w^3$ برای پیمایش (فرم چامسکی)**

$W=a_1 \ldots a_n$

$V_{ij} = a_i \ldots a_j$ $\quad V_{11}=a_1$ $\quad V_{ii}=a_i$

If $A \rightarrow a_i \quad \Rightarrow A \in V_{ii}$

$V_{ij} = \{A \mid \exists A \rightarrow BC_{\,(k=i..j-1)},\ B \in V_{i,k}\ ,\ C \in V_{k=1,j}\}$

**در نهایت اگر $S \in V_{1n}$ باشد ، پس جزء زبان است**

**مثال)**

بررسی کنید که آیا رشته aab توسط گرامر زیر پذیرفته میشود یا خیر؟

$S \rightarrow AB$
$A \rightarrow AD_1 | D_2D_2$
$D_1 \rightarrow D_2D_2$
$D_2 \rightarrow a$
$B \rightarrow BD_3 | b$
$D_3 \rightarrow b$

$V_{13}$=aab

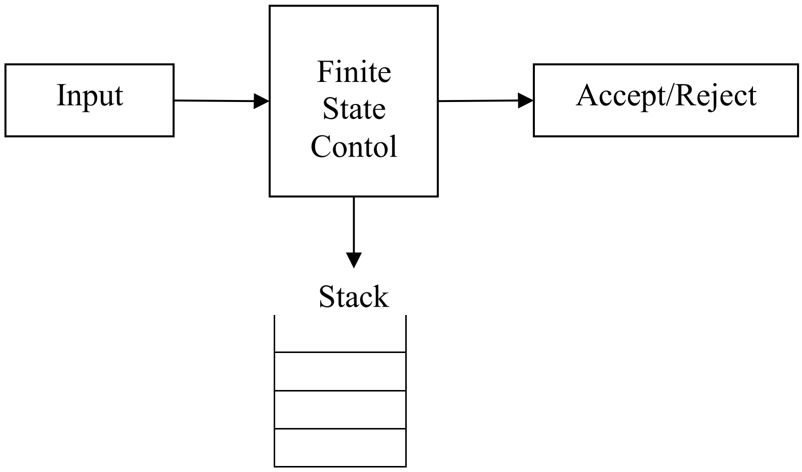
1) $V_{11}$ $\{D2\}$ $\quad V_{23}$ $\{D2\}$ $\quad V_{33}$ $\{D3, B\}$

2) $V_{12}$ $k=1: V_{11}V_{22}$ $\{D1,A\}$ $\quad V_{23}$ $k=2: V_{22}V_{33}$ $\{\}$

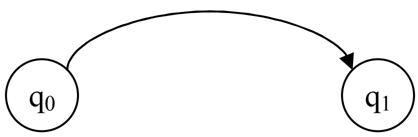
3) $V_{13}$
$k=1: V_{11}V_{23}$ $\quad D_2 \{\}$
$k=2: V_{12}V_{33}$ $\quad$ D1 D3
A B
$\{S\}$

پس رشته داده شده جزء زبان است.

## اتوماتای پشته ای (PDA)
## Push Down Automaton




**وضعیت اولیه: $q_0$**
**ورودی: U**
**پشته: X**
**جایگزینی: W**
**وضعیت بعدی $q_1$**




**PDA M=(K,Σ,Γ,Δ,S,F)**

Kوضعیت ها

Σالفبا

Γ:الفبای پشته

Δ:قوانین

S:شروع

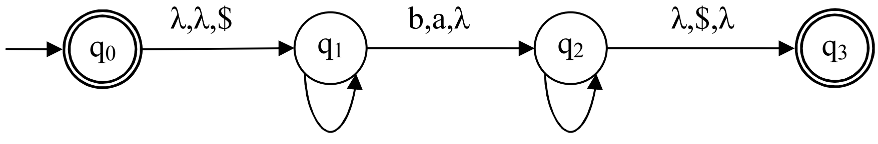
F:پایان

**$\Delta \subseteq (K*\Sigma*\Gamma)*(k*\Gamma)$**

---

**مثال ۱)**

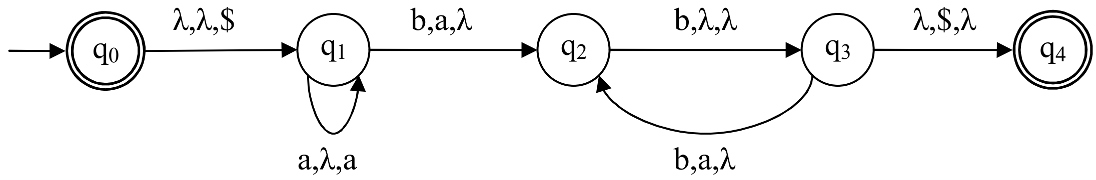
**PDA L={$a^nb^n$ | n>=0}**




---

**مثال۲)**

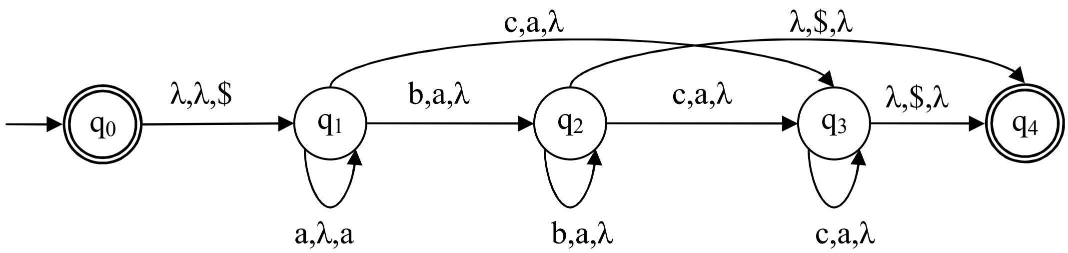
$a^nb^{2n}$




---

**مثال ۳)**

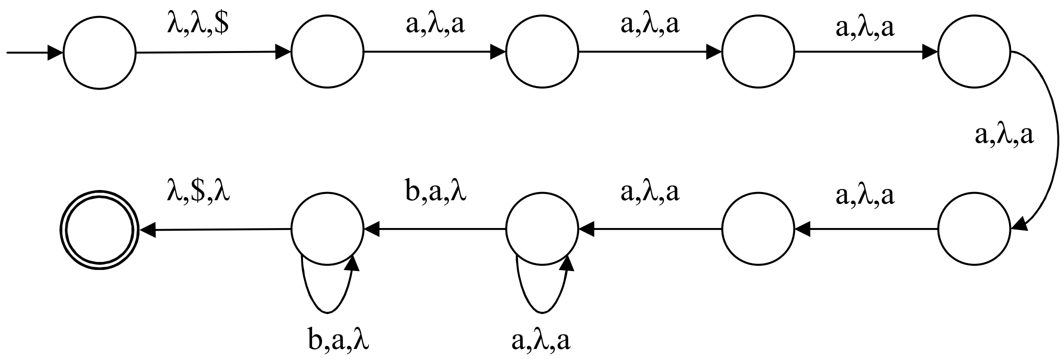
$a^nb^mc^k$ $n=m+k$




---

**مثال۴)**

$a^nb^n$ $n>5$




$S \rightarrow aaaaaaS_1bbbbbb$
$S_1 \rightarrow aS_1b \mid \lambda$

---

**مثال۵)**

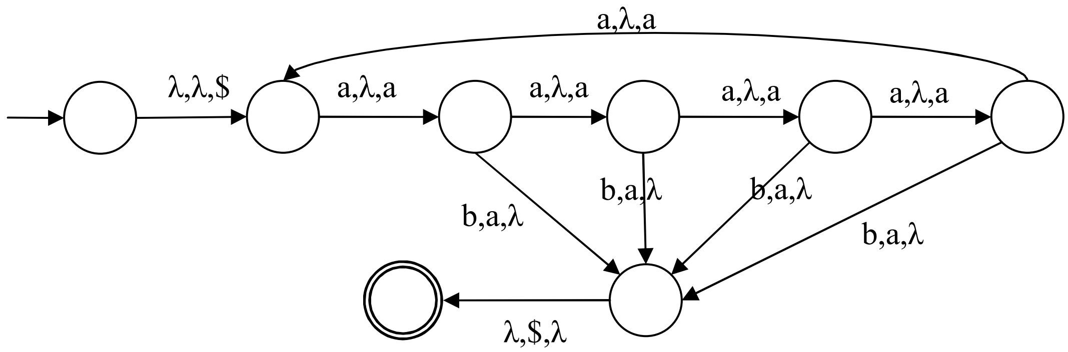
$a^nb^n$ **n مضربی از ۵ نباشد**

روش ۱)

$S \rightarrow aS_1b$
$S_1 \rightarrow aS_2b \mid \lambda$
$S_2 \rightarrow aS_3b \mid \lambda$
$S_3 \rightarrow aS_4b \mid \lambda$
$S_4 \rightarrow aSb \mid \lambda$

روش ۲)

**$S \rightarrow aS_1b \mid aaS_1bb \mid aaaS_1bbb \mid aaaaS_1bbbb$**
**$S_1 \rightarrow aaaaaS_1bbbbb \mid \lambda$**

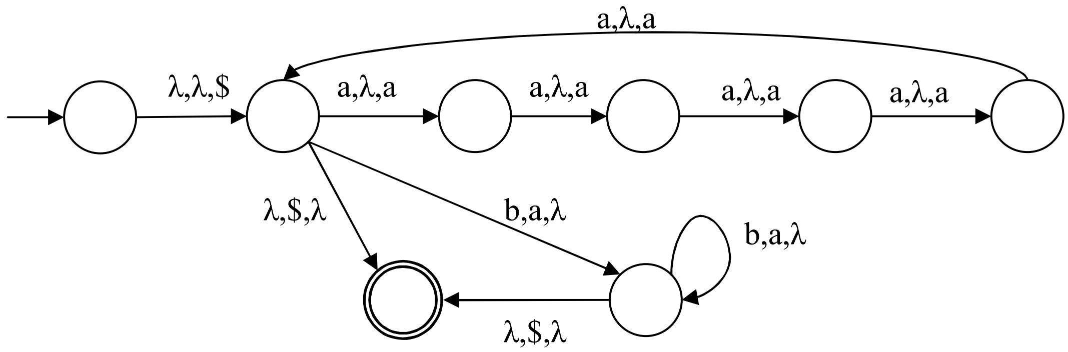



---

**مثال۶)**

**$a^nb^n$ مضربی از ۵ باشد n**

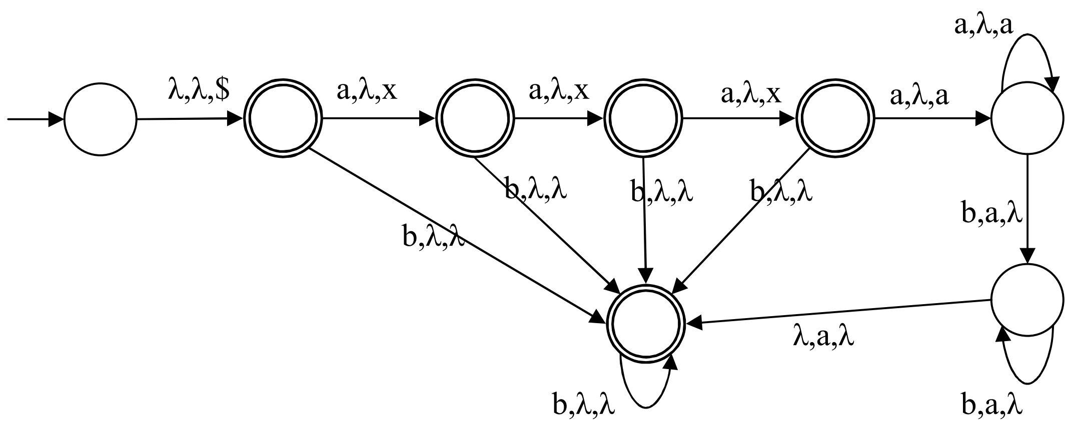
**$S \rightarrow aaaaaSbbbbb \mid \lambda$**




---

**مثال ۷)**

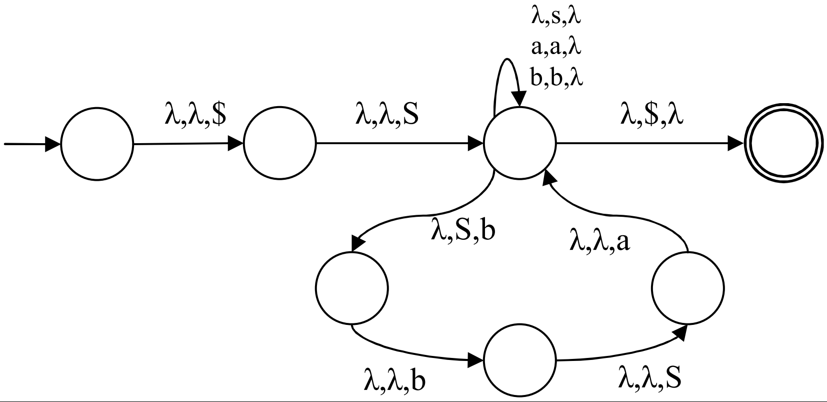
**$a^nb^m$ $n <= m+3$**



**مثال۸)**

$\mathbf{a^n b^{2n}}$

$S \rightarrow aSbb \mid \lambda$

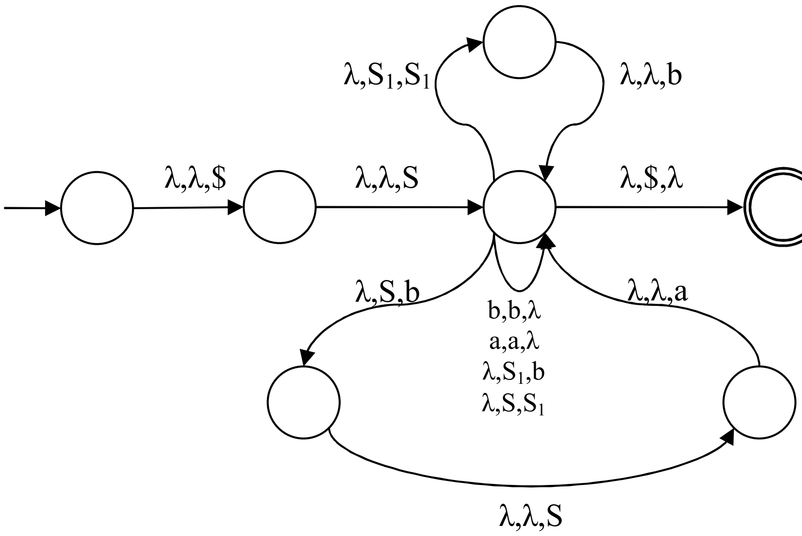



---

**مثال۹)**

$\mathbf{a^n b^m}$ **n<m**

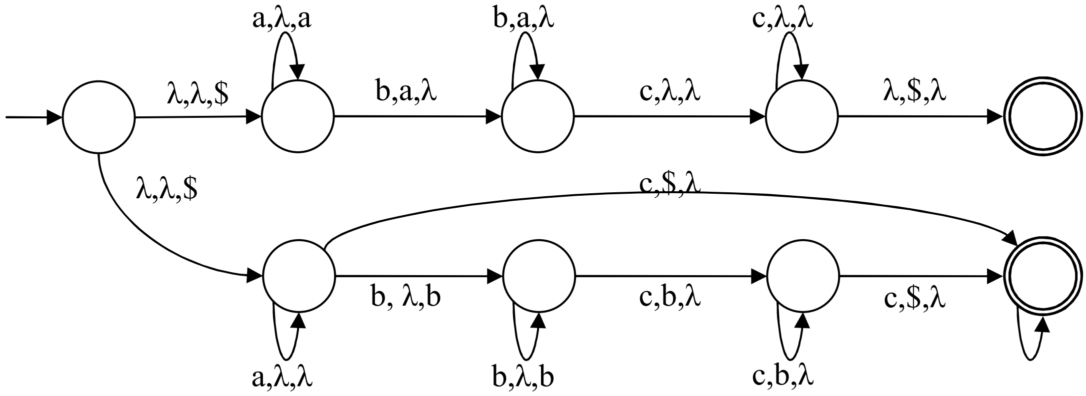
$S \rightarrow aSb \mid S_1$
$S_1 \rightarrow bS_1 \mid b$




---

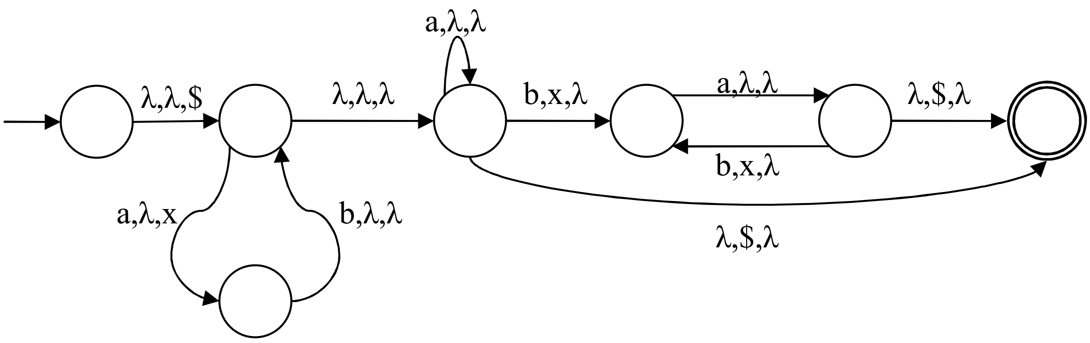
**مثال۱۰)**

$\mathbf{a^n b^m c^k}$ n=m or m<k



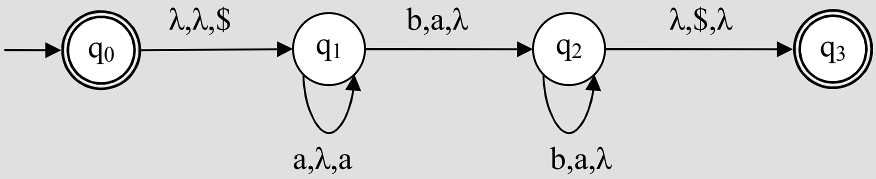
**مثال۱۱)**

$(ab)^n a^* (ba)^n$

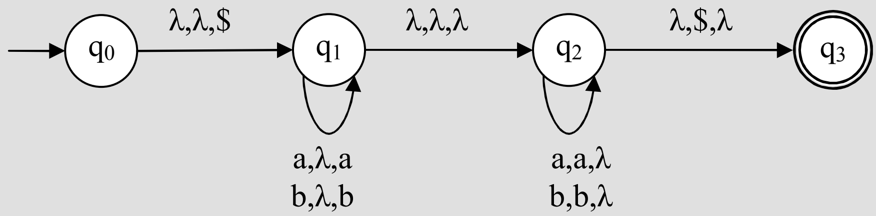



**انواع PDA:**

**قطعی:** $a^n b^n$




**غیرقطعی:** $ww^R$




## <u>لم تزریق :</u>

برای تشخیص اینکه یک عبارت مستقل از متن نیست بکار می رود.

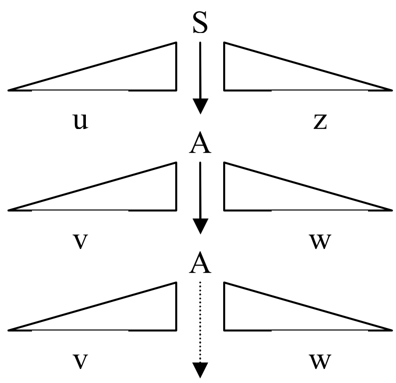
$|w|>=m$

$w=uvxyz$

$|vy|>=1$

$|vxy|<=m$

$uv^ixy^iz \in L$



**مثال ۱) ثابت کنید $a^nb^nc^n$ مستقل از متن نیست:**

$|vxy| <= m \rightarrow vxy = a^m$, $v=a^{k1}$, $y=a^{k2}$ $\rightarrow k1+k2>=1 \rightarrow a^{m-k1-k2}b^mc^m \notin L$

$\rightarrow vxy$, $v=a^k$, $y=a^k$ $\rightarrow k>1 \rightarrow a^{m-k}b^{m-k}c^m \notin L$

---

**مثال ۲) ثابت کنید L:WW مستقل از متن نیست:**

$L:WW \rightarrow a^mb^ma^mb^m \in L$

1) $\rightarrow v=a^{k1}$, $y=a^{k2}$ $\quad k1+k2>=1 , i=0 , a^{m-k1-k2}b^ma^mb^m \notin L$

2) $\rightarrow v=a^{k1}$, $y=b^{k2}$ $\quad k1+k2>=1 , i=0 , a^{m-k1}b^{m-k2}a^mb^m \notin L$

3) $\rightarrow v=b^{k1}$, $y=a^{k2}$ $\quad k1+k2>=1 , i=0 , a^mb^{m-k1}a^{m-k2}b^m \notin L$

---

**مثال ۳)**

$L: a^{n^2} \rightarrow a^{mm} \in L$

$v = a^{k1}$, $y = a^{k2}$ $\quad k1+k2>=1 , i=0 , uxz : a^{m.m-k1-k2} \notin L$

$(m-1)^2 = m^2-2m+1$

$M > k_1+k_2 >= 1 \rightarrow 2m-1 > k1+k2$

---

**مثال ۴)**

$a^mb^{m^2}$

$\rightarrow v=a^{k1}$ $\quad m>k1+k2>=1 , i=0, a^{m-k1}b^{m.m-k2} \notin L$

$\rightarrow y=b^{k2}$ $\quad m^2-k_2$ مربع کامل نیست

$\rightarrow$

**اثبات عدم وجود گرامر خطی با استفاده از لم تزریق:**

**$|uvyz| <= m \rightarrow uxz$**

**مثال) $S \rightarrow S1S1 , S1 \rightarrow aS_1b \mid \lambda$:**

**$L:a^nb^na^pb^p$** $\rightarrow a^mb^ma^mb^m \in L$

$v = a^{k1} , y=b^{k2}$

$uxz \rightarrow a^{m-k1}b^ma^mb^{m-k2} \notin L$

**مجموعه زبان های مستقل از متن نسبت به عمل اجتماع/الحاق/استار بسته هستند:**

**L1 U L2 → $S \to S_1 \mid S_2$**

**L1.L2 → $S \to S_1S_2$**

**L1* → $S \to SS_1 \mid \lambda$**

**اشتراک دو زبان مستقل از متن ، مستقل از متن نیست:**

$L1 \cap L2$

→ L1: $a^nb^nc^m$
→ L2: $a^pb^xc^x$ → $L1 \cap L2$: $a^nb^nc^n$

**زبان های مستقل از متن نسبت به متمم نیز بسته نیستند:**

$L1 \cap L2 = (L1' \cup L2')'$

**قضیه: اگر L1 مستقل از متن و L2 منظم باشد آنگاه $L1 \cap L2$ مستقل از متن است.**

L1 $\{Q_1, \Sigma, \Gamma, \delta_1, q_{01}, F_1\}$ → PDA
L2 $\{Q_2, \Sigma, \delta_2, q_{02}, F_2\}$ → DFA

$\delta_1 = Q_1 * \Sigma * \Gamma \to Q * \Gamma$
$\delta_2 = Q_2 * \Sigma \to Q_2$

$Q=Q_1*Q_2$
$q_0=(q_{01}, q_{02})$
$F=\{(q_i, q_j) \mid q_i \in F_1, q_j \in F_2\}$

$Q*\Sigma*\Gamma \to Q*\Gamma$
$\delta((q_i, q_j), a, b) = (\delta_1(q_i, a, b), \delta_2(q_j, a), c)$

**مثال۱) نشان دهید زبان روبرو مستقل از متن است:**

**$L=\{a^nb^n : n>0, n<>100\}$**

L1: $\{a^{100}, b^{100}\}$

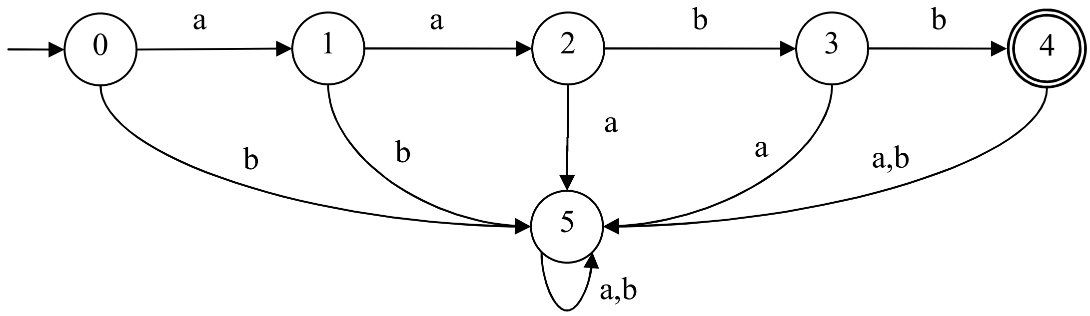
متمم یک زبان منظم ، منظم می باشد ، پس 'L1 منظم است

$L1' \cap \{a^nb^n, n>0\} = L$ مستقل از متن است

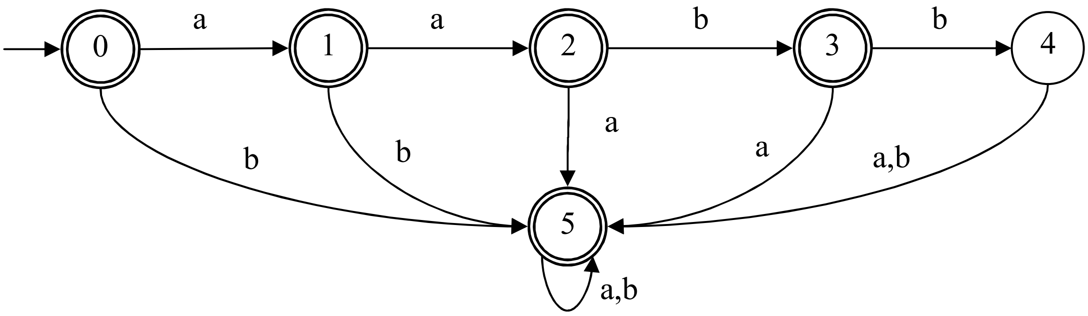
**مثال۲)نشان دهید زبان مقابل مستقل از متن است:**

**$L=\{a^n b^n : n>0 , n<>2\}$**

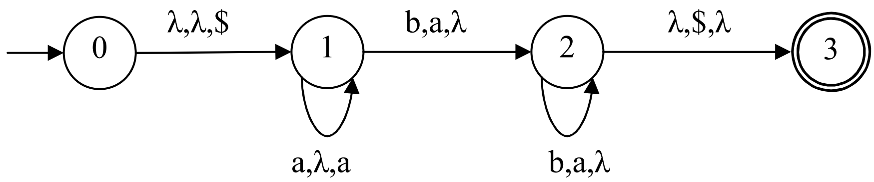
1) $L1=a^2b^2$ →DFA



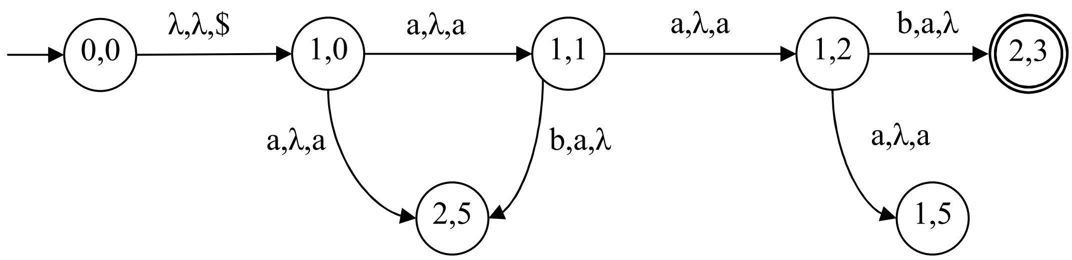

2) L1' →DFA




3) $L2=a^nb^n$ → PDA




4) $L1' \cap L2$




گرامرهایی که به ترمینال ختم نمی شوند را گرامرهای غیرمیرا گویند.

$S \rightarrow S_1a$
$S_1 \rightarrow aS_1b$

اگر یک دور یا حلقه در DFA داشته باشیم ، آنگاه آن زبان نامتناهی خواهد بود.

$S \rightarrow AB$
$A \rightarrow Sa$

گراف مربوط به گرامر را رسم می کنیم ، اگر حلقه وجود داشته باشد نیز نامتناهی خواهد بود.

## **ماشین تورینگ:**

در هر وضعیت ، محتویات نوار (حرفی که هد اشاره می کند) و وضعیت فعلی ، خروجی در همان مکان تولید میشود و به وضعیت بعدی می رویم (می توان به چپ و یا راست حرکت کرد)

$M = (Q,\Sigma,\Gamma,\delta,q_0,B,F)$

$\delta:Q*\Gamma \rightarrow Q*\Gamma*\{L,R\}$

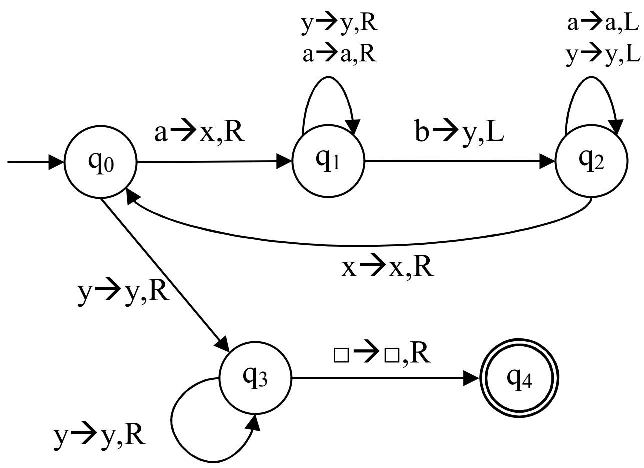
---

**مثال۱)**

$a^n b^n$

رفت: $\delta(q0,a) = (q1,x,R\ ) , \delta(q1,a) = (q1,a,R) , \delta(q1,y)=(q1,y,R) , \delta(q1,b)=(q2,y,L)$

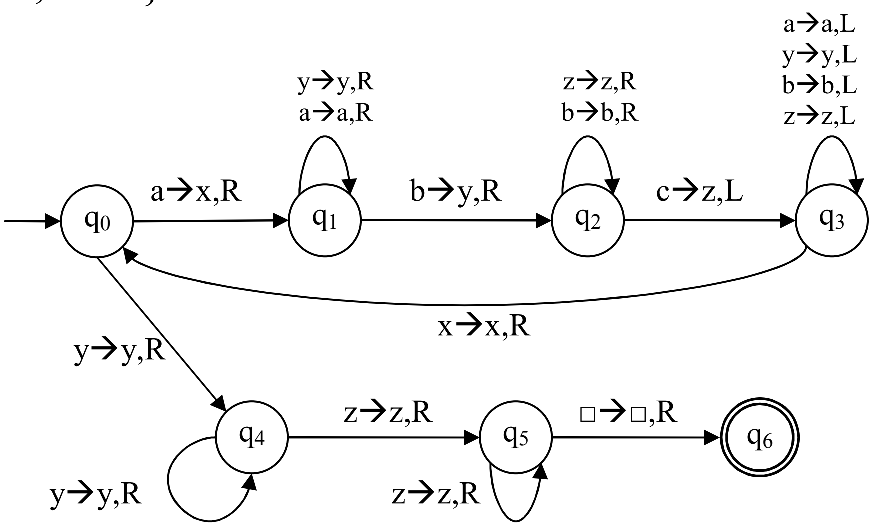
برگشت: $\delta(q2,x) = (q2,y,L) , \delta(q2,a) = (q2,a,L) , \delta(q2,x)=(q2,x,R)$




---

**مثال۲)**

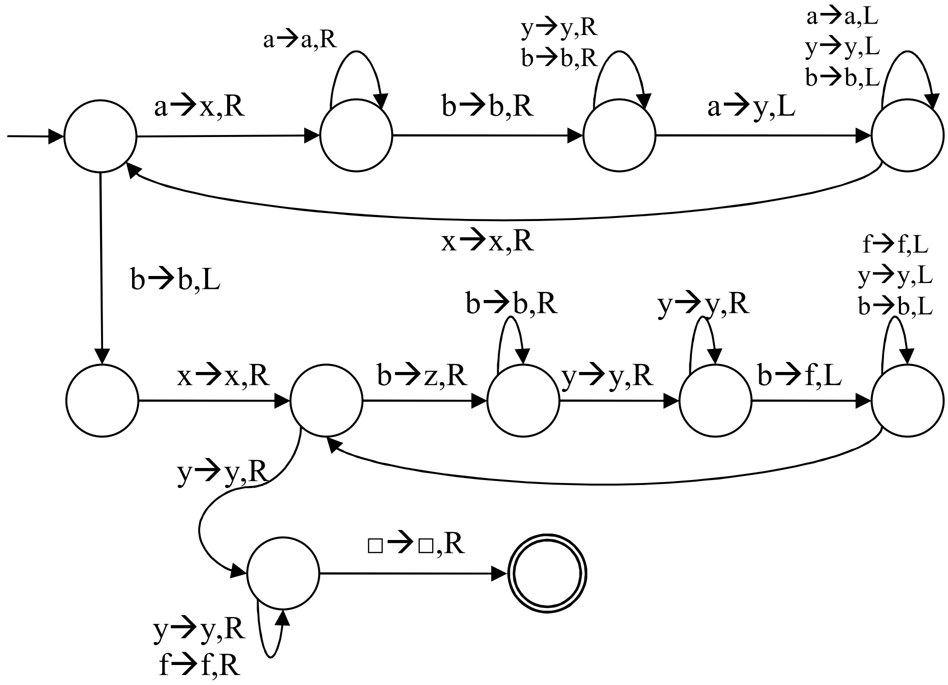
$\{a^n b^n c^n ,n>=1\}$




---

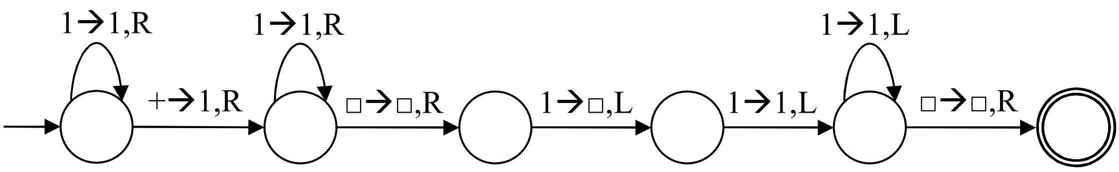
**مثال۳)**

**$a^n b^m a^n b^m$**



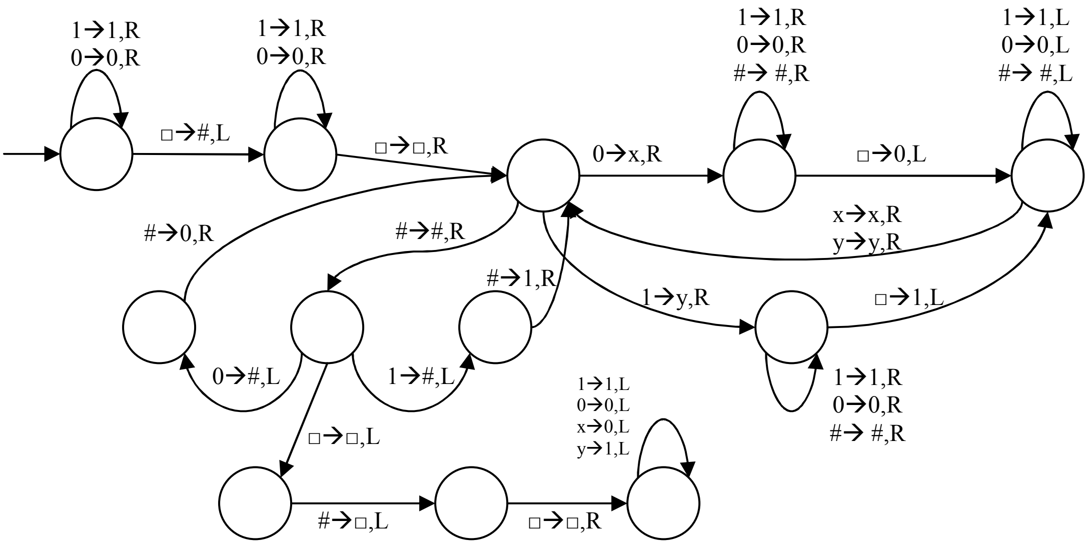

**مثال۴)**

**F(x,y)=x+y**




**مثال۵)**

**$q_0w \rightarrow q_fww$** $\rightarrow$ w=01101 $\rightarrow$ 01101# $\rightarrow$ 01101#01101 $\rightarrow$ 0110101101



**آیا چند نوار با یک هد معادل چند نوار با چند هد است؟**

a b c

a a b c

$$\delta(q_i , x_1 , x_2) = (q_j , y_1 , y_2 , d_1 , d_2)$$

$$d_2,d_1 \in \{L,R,S\}$$

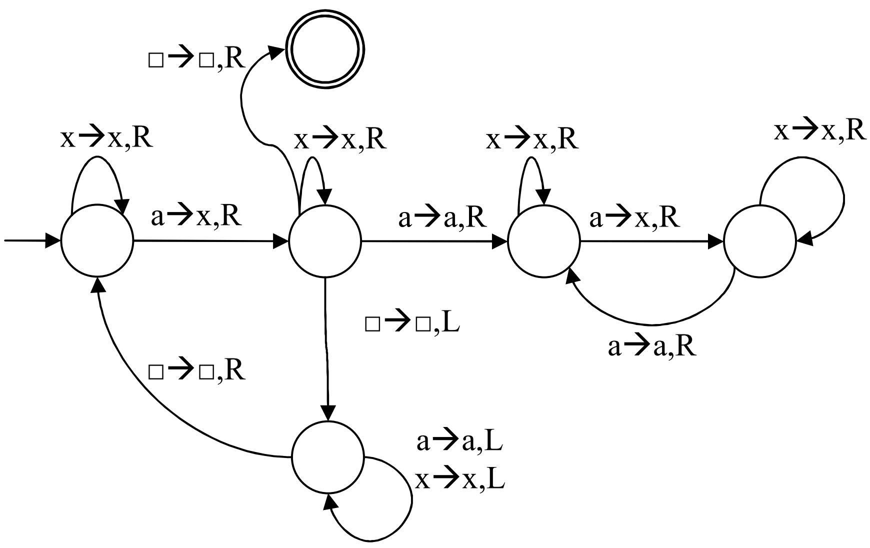
| |
|---|
| # |
| # a b c |
| # X |
| # a a b c |
| # X |

**با توجه به شکل فوق ، نتیجه گرفته میشود که هر k هدی را میتوان با 2k+1 نوار ایجاد کرد!**

**مثال۶) ماشین تورینگی طراحی کنید که زبان $a^{2^n}$ را تشخیص دهد:**

**a** **aa** **aaaa** **aaaaaaaa**

توضیح: هر بار رشته را با تقسیم کردن نصف می کنیم...



**مثال۷)**

$a^i b^j c^k \qquad k=i*j$

به ازای هر a تمام b ها را تبدیل به y می کنیم و به ازای هر b یک c را تبدیل به z می کنیم. پس از اتمام b ها دوباره آنها را به b تبدیل می کنیم و سراغ a بعدی می رویم. در انتها وقتی a ها تمام شد باید بررسی کنیم c باقی نمانده باشد.

aabbcccc→ xaybzccc→ xayyzzcc→ xabbzzcc→ xxybzzzc→ xxyyzzzz

---

**مثال۸)**

$a^{n!}$

هر با باید رشته تقسیم به اعداد از ۲ به بعد شود.

در انتخا باید یک a باقی بماند!

برای شمارش اعداد ۲ به بعد از نوار دوم استفاده می کنیم.

مثلا برای مرحله تقسیم بر ۲ :

| | | | | |
|---|---|---|---|---|
| aaaaaa | → | xaxaxa | → | xaxaxa |
| xa | | xa | | xxa |

**ماشین تورینگ جامع: Enumarator**

باید ماشین تورینگ را کد کنیم. به ماشین تورینگ جامع بدهیم. آنگاه ماشین تورینگ جامع تمام رشته هایی که تورینگ کد شده می پذیرد را تولید می کند:

$\delta(q_1,a) = (q_2,b,R) \quad \rightarrow \quad 0101011011010$

مجموعه اعداد گویا شماراست. زیرا میتوانیم ترتیبی برای شمارش هر یک از آنها بیان کنیم

$$\begin{matrix} \frac{1}{1} & \frac{1}{2} & \frac{1}{3} & \cdots \\ \frac{2}{1} & \frac{2}{2} & \frac{2}{4} & \cdots \\ \frac{3}{1} & \frac{3}{2} & \frac{3}{4} & \cdots \end{matrix}$$

---

مجموعه اعداد حقیقی شمارا نیست:

| | **0 / 2 4 6 …** |
|---|---|
| 1 | 0 / **1** 2 3 9 … |
| 2 | 0 / 2 **3** 4 9 … |
| 3 | 0 / 6 3 **5** 7 . . . |

عدد بدست آمده در مجموعه اعداد شمارش شده وجود ندارد ، زیرا هر عدد k در رقم k ام با این عدد متفاوت است!

---

زیر مجموعه اعداد طبیعی (مجموعه توانی) شمارا نیست.

$2^{|n|}$

{1},{2},{3}…
{1,2},{1,3}…
{1,2,3}…

| | 1 2 3 4 5 6 7 … |
|---|---|
| 1 | **0** 0 1 0 0 0 0 … |
| 2 | 0 **0** 0 0 1 1 0 … |
| 3 | 1 0 **0** 0 1 0 1 … |
| 4 | 1 0 1 **1** 1 0 1 … |
| . | … |
| . | … |

0: عدد جزء مجموعه k ام است

1: عدد جزء زیر مجموعه k ام است

برای بدست آوردن مجموعه جدید (که شمارش نشده است) از هر مجموعه عضوی را که مشخص شده است انتخاب می کنیم. اگر صفر بود یک می گذاریم و بالعکس. بنابراین مجموعه ای بدست می آید که عضو هیچیک از موارد قبل نیست. بنابراین مجموعه توانی شمارا نیست : …1110

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

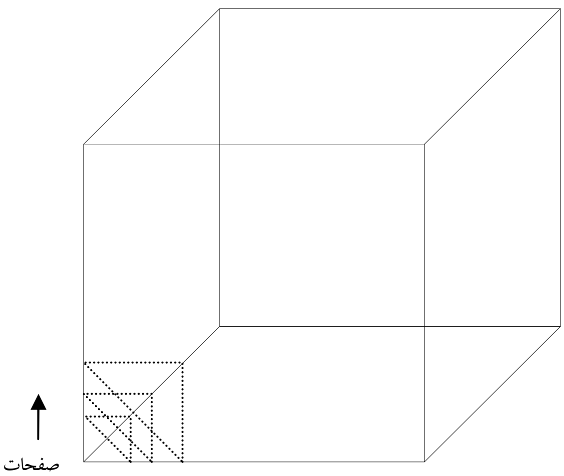
$\Sigma^*$ شماراست ➔ $\lambda$ , a , b , aa , ab , ba , bb , aaa , aab …

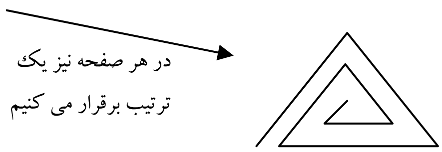
یک حرفی دوحرفی سه حرفی …

1 2 4 8 …

**نتیجه :** الفبایی که توسط زبان $\Sigma$ تولید میشود شماراست اما مجموعه همه زبانهایی که میتوان با الفبای $\Sigma^*$ نوشت ، شمارا نیست! بنابراین مجموعه همه ماشینهای تورینگ شماراست ولی مجموعه همه زبانهای آنها شمارا نیست. پس زبانی وجود دارد که برای آن ماشینی وجود ندارد.

شمارش در فضای سه بعدی (i,j,k)




**انواع ماشین تورینگ«**

بازگشتی برشمردنی (تشخیص پذیر): ممکن است ماشین متوقف نشود و فقط می توانیم بگوییم که رشته جزء زبان است

بازگشتی (تصمیم پذیر): ماشین متوقف می شود و وجود و یا عدم وجود در زبان مشخص می شود

برای تصمیم پذیر میتوان با یک نوار و یک هد ، ماشین جامعی داشت که در نهایت همه حالات را می شمارد و رشته های مختلف را بررسی می کند. پس اگر یک رشته بزرگتر از رشته مورد نظر تولید شد ، آنگاه رشته های کوچکتری که تولید نشده است ، جزء زبان نیست.

اما در مورد تشخیص پذیر باید از نوارهای دو بعدی بینهایت استفاده کرد. پس اگر یک رشته جزء زبان نبود و در loop افتاد مانع تولید رشته های دیگر نمی شود. بنابراین برای بدست آوردن جواب باید صبر کنیم!

ثابت کنید زبان هست که برای آن ماشین تورینگی وجود ندارد:

M1,M2,…
L(M1), L(M2) , …

$L = \{a_i \mid a_i \in L(Mi)\}$ تشخیص پذیر است

$L' = \{a_i \mid a_i \notin L(Mi)\}$

بنابراین برای L' ماشین تورینگی وجود ندارد. زیرا اگر وجود می داشت:

$L(M_k) = L' \rightarrow a_k \notin M(k)$

$A_k \notin L \rightarrow a_k \notin L'$ تناقض

$\rightarrow a_k \in L(M_k)$

پس L' تشخیص پذیر نیست.

---

**– متمم زبان بازگشتی حتما بازگشتی (تصمیم پذیر) است**

**– متمم زبانهای بازگشتی برشمردنی ، لزوما بازگشتی برشمردنی نیست**

**گرامر نامقید (بدون محدودیت):**

این گرامر معادل ماشین تورینگ است:

$\mathbf{X \rightarrow Y}$

$X \in (VUT)^+$
$Y \in (VUT)^*$

**مثال ۱)**

$\mathbf{a^n b^n c^n}$

$S \rightarrow aAbc \mid \lambda$
$A \rightarrow aAbC \mid \lambda$
$Cb \rightarrow bC$
$Cc \rightarrow cc$

برای نمونه:

$S \overset{1}{\rightarrow} aAbc$
$\overset{3}{\rightarrow} aaAbCbc$
$\overset{3}{\rightarrow} aaaAbCbCbc$
$\overset{4}{\rightarrow} aaabCbCbc$
$\overset{5}{\rightarrow} aaabbCCbc$
$\overset{5}{\rightarrow} aaabbCbCc$
$\overset{6}{\rightarrow} aaabbCbcc$
$\overset{5}{\rightarrow} aaabbbCcc$
$\overset{6}{\rightarrow} aaabbbccc$

---

**مثال۲)**

**{ u[u] | u ϵ Σ* }**

روش اول:

S → A[] | []
A →aXA | bYA
Xb → bX
Xa → aX
Yb → bY
Ya → aY
Y[ → [b
X[ → [a

روش دوم:

S → aA[a] | bA[b] | []
A → aAX | bAY | λ
Y[ → [Y
Ya → aY
Yb → bY
Y] → b]
X] → a]
X[ → [X
Xa → aX
Xb → bX

**گرامر حساس به متن**

گرامرهای حساس به متن حتما بازگشتی (تصمیم پذیر) هستند

ولی هر گرامر بازگشتی ، حتما حساس به متن نیست.

**X → Y**
$X \in (V \cup T)^+$
$Y \in (V \cup T)^+$
$|X| <= |Y|$

$\{G_i \ldots\}$ حساس به متن

$\{ w_0 , w_1 ,\ldots\}$ بازگشتی

$L:\{W_i \mid W_i \in G_i\}$
$L = L(G_k)$
$W_k \in L(G_k) \rightarrow W_k \notin L$
$w_k \in L \rightarrow w_k \notin L(G_k)$

بنابراین هر زبان حساس به متن حتما بازگشتی است (وقتی دنباله اشتقاق طولش از رشته بیشتر شود ، دیگر جزء زبان نیست) ولی هر زبان بازگشتی حساس به متن نیست. (گرامرهای حساس به متن با ماشین تورینگ کراندار خطی معادلند)

**مثال۱)**

$\mathbf{a^n b^n c^n}$

S → aAbc | abc
A → aAbc
Cb → bC
Cc → cc

---

**مثال۲)**

$\mathbf{a^n b^m c^n d^m \qquad m,n>=1}$

روش اول:

S→ aSC | $S_1$
$S_1$ → bSd
dC → Cd
bC → bc
cC → cc

روش دوم:

S → S1S2
S1 → aS1c | ac
S2 → BS2d | Bd
cB → Bc
aB → ab
bB → bb

---

**طبقه بندی چامسکی در مورد زبان ها**