

Set

$$S = \{ - \}$$

موجود  $x \in S$  ,  $x \notin S$  موجود

موجود

$$S_1 \cup S_2 = \{ x | x \in S_1 \text{ or } x \in S_2 \}$$

$$S_1 \cap S_2 = \{ x | x \in S_1 \text{ and } x \in S_2 \}$$

$$S_1 - S_2 = \{ x | x \in S_1 \text{ and } x \notin S_2 \}$$

$$\overline{S} = \{ x | x \in U \text{ and } x \notin S \}$$

$$\overline{\overline{S}} = S, \overline{U} = \phi$$

$$|2^S|$$

تعداد اعضا

$$2^S$$

$$|2^S| = 2^{|S|}$$

Power Set

تعداد اعضا

مثلاً مجموعه ۳ عضو  $|2^S| = 2^3 = 8$  زیرمجموعه دارد.

Partition افراز  $(S_1, S_2, \dots, S_n)$

مجموعه های  $S_1, S_2, \dots, S_n$  که هیچ دو عضو مشترکی ندارند.

$$1. S = S_1 \cup \dots \cup S_n$$

اجتماع کل مجموعه S را می دهد

$$2. \forall i, j \quad i \neq j \quad S_i \cap S_j = \emptyset$$

هیچ دو مجموعه دو عضو مشترک ندارند

$$3. \forall i, S_i \neq \emptyset$$

هیچ یک از مجموعه ها خالی نیستند

تعارف روابط Relation در مقابل تابع Function

به هر زیرمجموعه از حاصلضرب دکارتی دو مجموعه یک رابطه می گویند.

۱- بازتابی: هر عضو مجموعه S با خودش رابطه دارد.

$$1. \text{Reflexive } \forall x \in S \quad x R x$$

در هر n، y و x رابطه دارد و هم با رابطه دارد.

$$2. \text{Symmetric: } \forall x, y \in S \quad x R y \Rightarrow y R x$$

$$3. \text{Transitive: } \forall x, y, z \in S \quad x R y \wedge y R z \Rightarrow x R z$$

PAPCO

نسبتی

یک تراو منطقی صحیح است.

$$S - \emptyset = S \cup \emptyset = S$$

Demorgan's law

$$\overline{S_1 \cap S_2} = \overline{S_1} \cup \overline{S_2} \quad , \quad \overline{S_1 \cup S_2} = \overline{S_1} \cap \overline{S_2}$$

Sub set:  $S_1 \subseteq S_2$

Proper

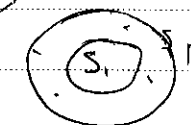
Proper subset

$$S_1 \subseteq S_2 \quad (S_1 \subset S_2 \wedge S_1 \neq S_2)$$

A ⊆ B

زیرمجموعه محض

$$(*) |S| = \text{تعداد اعضا مجموعه} = \text{تعداد اعضا S}$$



$$S_1 \times S_2 = \{(x, y) \mid x \in S_1 \text{ and } y \in S_2\}$$

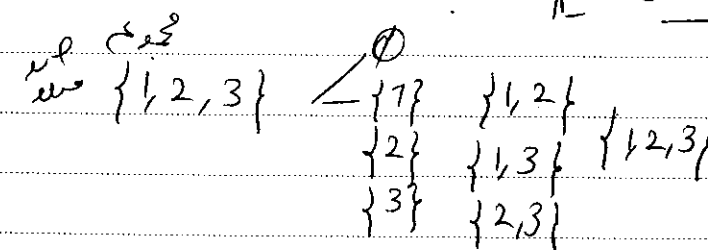
حاصلضرب دکارتی: زوجی مرتب از x و y بطوریکه x عضو مجموعه اول و y عضو مجموعه دوم.

$$S_1 \times \dots \times S_n = \{(x_1, \dots, x_n) \mid x_i \in S_i\}$$

همه متعلق به S و غیره

مجموعه توان Power Set

مجموعه کل زیرمجموعه های S را می گویند (Power Set)



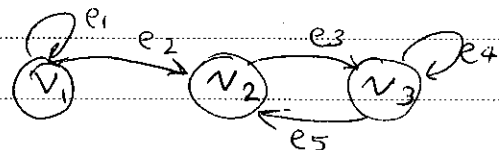
PAPCO

# Graph

لبه ها  $\rightarrow$  Edges  
 Vertices  $\rightarrow$  E  
 $G = \langle V, E \rangle$

رأس = گره  
 بگراف ۲ جزء دارد

$E = \{e\}$   
 $e = (v_1, v_2)$



درون جهت  
 $e_2 = \{v_1, v_2\}$   
 زوج مرتب است

## Path

مسیر یعنی از یک گره به گره دیگر با استفاده از یک یا چند لبه. هر گره ای در آن تکرار شده باشد به مسیر بسته می گویند. اگر تکرار نشده باشد به مسیر باز می گویند.

## Cycle

مسیری که رأس ابتدا و انتهای آن یکی باشد به Cycle می گویند. اگر رأس تکرار شده باشد به Cycle بسته می گویند. اگر تکرار نشده باشد به Cycle باز می گویند.

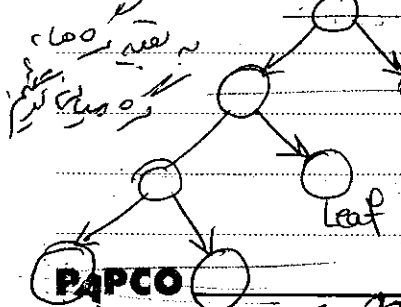
## Loop

Cycle که دارای طول ۱ (یک لبه) باشد

طول مسیر = تعداد یال ها

تقریب درخت به گراف که همبند نباشد و دور نداشته باشد

## Tree: root



۱- گراف جهت داری که راجع آن دور وجود نداشته باشد  
 ۲- همبند نباشد

root گره ای که پدر نداشته باشد  
 Leaf گره ای که فرزند نداشته باشد

ارتفاع: طولانی ترین مسیر از ریشه به برگ

تابع  $f$  از مجموعه  $S_1$  به مجموعه  $S_2$

$\{1, 2\}$

$xRy \wedge yRz \Rightarrow xRz$   
 T F F

بهر هم آید

اگر هر رابطه ۳ خاصیت قبل را داشته باشد می گوئیم هم آری است

تابع  $f$  از مجموعه  $S_1$  به مجموعه  $S_2$  که در آن هیچ دوزوج مرتب متفاوتی در آن وجود نداشته باشد به تابع یک به یک می گویند.

Total:  $D_f = S_1$

Partial:  $D_f \subset S_1$

$f: S_1 \rightarrow S_2$

Total

$D_f \subset S_1$

$R_f \subset S_2$

Partial

به تابعی که در  $S_1$  برابر  $f$  باشد

به تابعی که در  $S_1$  برابر  $f$  باشد و در  $S_2$  برابر  $R_f$  باشد

Proper Subset

طول رشته  $\xrightarrow{P}$  رشته  $\xrightarrow{P}$

طول رشته  $\xrightarrow{P}$  رشته  $\xrightarrow{P}$

String:

Concat ( $\circ$ ) = بهم پیوستن رشته یا ادغام کردن یا الحاق کردن  
 طول رشته

$x = 'a^pbc'$   $|x| = 3$

$x \cdot y = abcde$  رشته به اضافه (یا به پیوستن)

$y = 'c^pde'$   $|y| = 3$

$|x \cdot y| = |x| + |y| = 6$

$|x \cdot y| = |x| + |y|$  (نسبت در کتاب است) و طول Concat شده دو رشته برابر است با

$\lambda = \text{lambd}$   $|\lambda| = 0$  طول 0  
 عضو  $\emptyset$  رشته ای به طول صفر و عضو  $\emptyset$  عمل Concat

$x \cdot \lambda = \lambda \cdot x = x$

$|x \cdot \lambda| = |x| + |\lambda| = |x|$

$L = \{0, 1, 00, 11, 000, 111, \dots\}$  (این زبان 3 رشته دارد با الفبای  $\{0, 1\}$ )

$\emptyset$  مجموعه است اما  $\lambda$  رشته است.

الفبا الفبا است

Reverse ( $R$ ) معکوس

$x = (abc)$   $x^R = cba$

$x^{RR} = (abc)$

$W^{R^2} = W$   
 رشته ای را که 2 بار معکوس کنیم  
 خودی می شود

طول هر رشته  $\xrightarrow{P}$  تعداد الفبا  
 رشته  $\xrightarrow{P}$

Language: A set of strings

مجموعه از رشته ها = زبان

String: A sequence of alphabet

رشته دنباله از الفبا

Alphabet: A set of Symbols

الفبا، مجموعه از علائم یا عضو

Natural = طبیعی  
 Formal \* = رسمی  
 Languages

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

Finite  $L = \{010, 11, 110\}$

Infinite  $L = \{0^n \mid n \text{ is even}\}$

$= \{00, 0000, \dots\}$

الفبا

$\Sigma = \{0\}$

الفبا صفر  $\rightarrow$  (نامشخص)  
 نامشخص اگر تعداد رشته ها نامحدود باشد  
 مشخص اگر تعداد رشته ها محدود باشد (مشخص)

الفبا در اینجا  $\Sigma$  نشان می دهد

الفبا یک زبان با  $\Sigma$  نشان می دهد

$$\emptyset^* = \{\lambda\}$$

Subject:

Year.

Month.

Date.

( )

$$L^R = \{w^R \mid w \in L\}$$

معکوس زبان =  
معکوس رشته ها

$$L = \{a^n b^n \mid n \geq 0\}$$

هم رشته ها هستند از  $ab$  به خصوص  
این هستند که  $a$  اول بعد  $b$  و تعداد  
 $a, b$  برابر است

$$= \{\lambda, ab, aabb, \dots\}$$

$$L^2 = L \cdot L = \{a^n b^n a^m b^m \mid n, m \geq 0\}$$

$$\rightarrow L^R = \{b^n a^n \mid n \geq 0\}$$

$$\bar{L} = \{w \mid w \in \Sigma^* \text{ and } w \notin L\}$$

Complicate  
مفید نیست

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

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

$$L^0 = \{\lambda\}$$

از یک عضو دارد که طول صفر است

$$L^1 = L = \{a, b\}$$

$$L^2 = L \cdot L = \{aa, ab, ba, bb\}$$

$$L^3 = L^2 \cdot L = \{aaa, aab, \dots\}$$

$$L^n = L^{n-1} \cdot L = \{\dots\}$$

$$\bigcup_{i \geq 0} L^i = L^0 \cup L^1 \cup \dots \cup L^n$$

هم رشته ها هستند  
هم رشته ها هستند

PAPCO

$$L_1 \cap L_2 \mid L_1 \cup L_2 \mid L_1 \cdot L_2 \mid L_1 \circ L_2$$

Subject:

Year.

Month.

Date.

( )

$$w^{R(n-1)} = \begin{cases} w^R & \text{if } n \text{ is even} \\ w & \text{if } n \text{ is odd} \end{cases}$$

زوج  
فرد

$$*(x \cdot y)^R = y^R \cdot x^R *$$

$$\text{Reverse}(a_1 \dots a_n) = \begin{cases} a_n & \text{if } n=1 \\ a_n \cdot \text{Reverse}(a_1 \dots a_{n-1}) & \text{otherwise} \end{cases}$$

معکوس  $w$  در  $L$

$$L_1 \cup L_2 = \{w \mid w \in L_1 \text{ or } w \in L_2\}$$

$$L_1 \cap L_2 = \{w \mid w \in L_1 \text{ and } w \in L_2\}$$

$$L_1 - L_2 = \{w \mid w \in L_1 \text{ and } w \notin L_2\}$$

$$L_1 \cdot L_2 = \{w_1 w_2 \mid w_1 \in L_1 \text{ and } w_2 \in L_2\}$$

$$L_1 \cdot L_2 \neq L_2 \cdot L_1$$

Concat  
تکثیر شده های  $L_1, L_2$  به هم  
Max  
تعداد رشته  $L_1$  در تعداد رشته  
 $L_2$  چون تکثیر ها را نمی توانیم

$$L_1 = \{a, ab\}$$

$$L_2 = \{10, 100, 110, 1000, 1010\}$$



$\Sigma^*$  : جملین عناصر الفبا است. مجموع در توصیف  $\Sigma$  می آید.

Subject: Year: Month: Date:  $\Sigma^*$  : جملین عناصر مثبت رشته است.

$$U_{i \geq 0} L^i = L^+$$

star-closure of L

تمام رشته ها حاصل از  $a, b$

$$L = \{a, b\} \quad L = \{aa, b\}$$

$$U_{i \geq 0} L^i = L^+$$

Positive-closure of L

تمام رشته ها با طول  $\geq 1$

تمام رشته ها حاصل از  $a, b$  با طول  $\geq 1$  است.  $\Sigma^+$  : جملین عناصر مثبت رشته است.

$$\Sigma^+$$

(جملین عناصر مثبت رشته است)

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

$$L = \{aa, b\}$$

جملین عناصر مثبت رشته است / جملین عناصر مثبت رشته است  $\Sigma^*$

$$L = \{aa, b\}$$

تمام رشته ها حاصل از  $a, b$  که هر دنباله ای که از  $a$  ها طولش زوج است

$$L^* = \text{موجود}$$

(تمام رشته ها بدون محدودیت)

$$L \neq \Sigma^* \text{ and } L \neq \emptyset$$

تمام رشته ها که در  $\Sigma^*$  با آن الفبا ساخته شده اما داخل  $L$  نیستند.

(زمانی که  $L$  تمام رشته ها با طول  $\geq 1$  را داشته باشد  $\Sigma^+ = L^*$  در مثال فوق)

$$\phi^* = / \phi^+ =$$

Subject: Year: Month: Date:

$$L = \{a^n b^n | n \geq 0\}$$

$$L^+ = \{b^n a^n | n \geq 0\}$$

$$L^2 = L.L = \{a^n b^n a^m b^m\}$$

$$L = \{a^n b^m | n \neq m\} \cup \{w | w \text{ contains at least 'ba'}\}$$

$$\phi^* = \{ \lambda \}$$

$$\phi^+ = \emptyset$$

$$L.\phi = \phi.L = \phi = \{ \}$$

$$L.\{ \lambda \} = \{ \lambda \}.L = L$$

که اینک زود فارسی است

$$1) L^* = L^+ \cup \{ \lambda \} \checkmark$$

$$1) L^+ = L^* - \{ \lambda \}$$

$$3) \Sigma^* = \Sigma^+ \cup \{ \lambda \} \checkmark$$

$$e) \Sigma^+ = \Sigma^* - \{ \lambda \} \checkmark$$

در بعضی زبانها این خاصیت را داریم، یعنی  $L \subseteq L^*$  و  $L^* \subseteq L$  در صورتی درست است که این دو خاصیت را داشته باشند.

$L = \{w \mid w \in \{a,b\}^*, n_a(w) = n_b(w)\}$   
تعدادهای متساوی از  $a, b$  که تعداد  $a, b$  هم برابرند.

مجموعه از رشته ها که تعداد  $a, b$  برابر  $L^1$

زیر  $L^2$  هم مجموعه ای از رشته ها که تعداد  $a, b$  برابر  $L^2 = L$ ?

که این خود زیر مجموعه ای از  $L$  می شود همین ترتیب برای  $L^3, L^4, \dots$

که  $L^*$  هم  $L$  است پس  $L^* \subseteq L$  و  $L \subseteq L^*$

چون  $L$  داریم و کوتاه ترین رشته است رشته های کوتاه هم در  $L^2, L^3, \dots$

$L = \{w \mid w \in \{a,b\}^*, |w| \bmod 8 = 0\}$   
تعدادهای متساوی از  $a, b$  که طول رشته ضربی از ۸ باشد.

زیرا طول رشته های ضربی از ۸ است همیشه  $L^2$  هم رشته های ضربی از ۸ است.

از ۸ است و همیشه را آخر  $L^* \subseteq L$  است.

این چون  $L$  داریم همیشه  $L^2$  یک  $L$  ضربی از ۸ است.

تعداد رشته ها بطوریکه طولش مضرب از ۸ باشد یعنی  $L = \{w \mid w \in \{a,b\}^*, |w| \bmod 8 = 0\}$

$L = \{w \mid w \in \{a,b\}^+, |w| \bmod 8 = 0\}$   
آنها را می توانیم  $L = L^*$  است.

$L^+$  رشته  $\{a\}$  دارد که خود  $L$  را ساخته است.

در این  $L$   $\{a\}$  ندارد پس  $L^+$  هم  $\{a\}$  ندارد.

$L^*$  همه  $\{a\}$  دارد.  $L$   $\{a\}$  ندارد اما  $L^*$  همه  $\{a\}$  دارد.

در این چون  $L$  ندارد پس کوتاه ترین رشته  $L$ ، ۸ تایی است همین ترتیب کوتاه ترین رشته  $L^2$ ، ۱۶ تایی ... پس  $L^n$ .

ترتیب کوتاه ترین رشته  $L^2$ ، ۱۶ تایی ... پس  $L^n$ .  
درست است  $L^2 \subseteq L$ ؟  $L^2 = L$ ؟

\* وقتی  $L^*$  برابر  $L$  است که تمام رشته ها طول یکسانی دارند و وجود داشته باشد.

\* آیا هر زبانی که  $L = L^*$  بدین قاعده می باشد؟ به جز وقتی که  $L = \{a\}$

در غیر اینصورت اگر  $L = \{a\}$  باشد  $L^* = \{a\}$  و  $L = \{a\}$

نمونه ها

گرافها = یامنها / بزرگ غیر یامنها / λ چند یامنها است

Subject: \_\_\_\_\_  
Year. \_\_\_\_\_ Month. \_\_\_\_\_ Date. \_\_\_\_\_

$$G = \langle V, T, S, P \rangle$$

$$V = \{S, A, B\}$$

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

$$P = \begin{cases} S \rightarrow AB \\ A \rightarrow aAb | \lambda \\ B \rightarrow Bb | b \end{cases}$$

$$\Rightarrow^* \text{استق در صفرا یامنها}$$

$$\Rightarrow^+ \text{استق در یک یا بیشتر یامنها}$$

$$\Rightarrow^p \text{استق}$$

Derivation

قرار دارد پس

حروف نزدیک غیر یامنها  
حروف دور یامنها

$$a^2b^3 \text{ تولید}$$

$$S \Rightarrow AB \Rightarrow aAbB \Rightarrow aaAbbB \Rightarrow aabbBb \Rightarrow aabbBb$$

$$aabbBb \Rightarrow^* a^2b^3$$

$$S \Rightarrow^* aAbB$$

$$S \Rightarrow^+ aAbB$$

$$S \Rightarrow^* aabb$$

$$S \Rightarrow^+ aabb$$

$$S \Rightarrow^* S$$

$$S \Rightarrow^+ S$$

$$L = \{a^n b^m \mid n \leq m\}$$

$$L(G) = \{w \in T^* \mid S \Rightarrow^* w\}$$

PAPCO

(تولید)

که بتوان در صفرا یا بیشتر یامنها S به آن رسید

گرافها؟

Subject: \_\_\_\_\_  
Year. \_\_\_\_\_ Month. \_\_\_\_\_ Date. \_\_\_\_\_

گرامرها = مجموعه قوانین هستند که از طریق آنها رشته‌های زبان را می‌توان نوشت

$$G = \langle V, T, S, P \rangle$$

$V$ : Variables = غیر یامنها / Non Terminal

Non-Terminals = غیر یامنها

Terminals = یامنها

$S$ : Start Symbol = علامت شروع  $S \in V$

$P$ : Set of Production rule = مجموعه قوانین تولید

$$P = \{ \langle \text{text} \rangle ::= \langle \text{sentence} \rangle \mid \langle \text{sentence} \rangle \langle \text{text} \rangle \}$$

$S \leftarrow$

$$\langle \text{sentence} \rangle ::= \langle \text{subject} \rangle \langle \text{verb} \rangle \langle \text{object} \rangle$$

$\rightarrow T$

$$\langle \text{subject} \rangle ::= I / you / \dots / they / \langle \text{Name} \rangle$$

$$\langle \text{Name} \rangle ::= A \langle \text{rest} \rangle \mid \dots \mid Z \langle \text{rest} \rangle$$

$$\langle \text{rest} \rangle ::= a \mid \dots \mid z \mid a \langle \text{rest} \rangle \mid \dots \mid z \langle \text{rest} \rangle$$



سپار از خطی بعد می گیریم و به صورت می نویسیم

$$\begin{aligned} A &\rightarrow \lambda \\ A &\rightarrow a_1 a_2 \dots a_n \rightarrow A \rightarrow a_1 x_1 \\ A &\rightarrow B \quad x_1 \rightarrow a_2 x_2 \\ A &\rightarrow a_1 \dots a_n B \quad \vdots \\ &\quad x_{n-1} \rightarrow a_n B \end{aligned}$$

2) مستقل از متن: Context Sensitive  
 + 1 سمت راست نمی تواند از متن جدا شود

$$A \rightarrow \alpha, \alpha \in (\cup V \cup T)^*$$

می تواند هر رشته ای با طول معین را آویزیم  
 نمی آویزیم

$$L = \{a^n b^n \mid n \geq 0\}$$

$$S \rightarrow aSb \mid \lambda$$

$$L = \{w \mid n_a(w) = n_b(w)\}$$

مستوی که طول a و b برابر باشد

$$S \rightarrow aSb \mid bSa \mid \lambda \mid SS$$

abba  
 به abba می توان نوشت

$$S \rightarrow SS \rightarrow aSb bSa \rightarrow abba$$

انواع گرامرها: از ساده ترین تا پیچیده ترین

- 1- گرامرهای نوع منظم (منظم، Regular)
- 2- (مستقل از متن، Content-free)
- 3- اول (حساس به متن، Content-sensitive)
- 4- صفر (بدون محدودیت، Unrestricted)

اگر ادعا می کنیم که گرامر مستقل از متن این چنین باشد:  $N \rightarrow T \mid NT$   
 { مستقل از متن }  $N \rightarrow T \mid NT$

منظم از متن

$$\begin{cases} A \rightarrow a \\ A \rightarrow aB \\ a \in T \\ A, B \in V \end{cases}$$

این گرامر مستقل از متن است

منظم از متن

$$\begin{cases} A \rightarrow a \\ A \rightarrow Ba \\ a \in T \\ A, B \in V \end{cases}$$

این گرامر مستقل از متن است

این گرامر مستقل از متن است

$$L = \{a^n \mid n \geq 2\}$$

$$\begin{cases} S \rightarrow aA \\ A \rightarrow a \mid aA \end{cases}$$

منظم از متن

$$\begin{cases} S \rightarrow Aa \\ A \rightarrow a \mid Aa \end{cases}$$

منظم از متن

$$L = \{a^n \mid n \geq 0\}$$

$$S \rightarrow aS \mid \lambda$$

منظم از متن

$$\begin{cases} A \rightarrow x \\ A \rightarrow xB \\ x \in T^* \\ A, B \in V \end{cases}$$

خط از متن

$$\begin{cases} A \rightarrow x \\ A \rightarrow Bx \\ x \in T^* \\ A, B \in V \end{cases}$$

خط از متن

خطی که می توان نوشت

4) Unrestricted

$$\alpha \rightarrow \beta$$

$$\alpha \in (VUT)^+$$

$$\beta \in (VUT)^*$$

برای هر بدول محدودیت  
 در این بین حالت کاهش طول هم می توانیم داشته باشیم

برای سدهای شور قوانین را با قوانین هم می توان کرد آن قانون  
 غیر قابل حل است.

✓ 7- 11- 13- 14- 15- 16- 18

(در این بین از یکسوم رقم فعلی)

نشان دهم

③ Content - Sensitive

"حساس به محتوا"

$$\alpha \rightarrow \beta$$

سدهای طول ایستاده اند و غیره

$$\beta \in (VUT)^*$$

طول قاعده نمی تواند کاهش پیدا کند به استثنای اولیای  
 باین اولیای

$$L = \{a^n b^n c^n / n \geq 1\}$$

برای این می توان مستقیماً از متن نوشت

$$S \rightarrow S / \alpha$$

برای آنکه  $\alpha$  را تولید کنیم

$$S \rightarrow abc$$

$$S \rightarrow axbc$$

$$xb \rightarrow bx$$

$$xc \rightarrow ybcc$$

$$by \rightarrow yb$$

$$ay \rightarrow aa$$

$$ay \rightarrow aax$$

یک و اضافه می شود  
 ① در متن می کنیم درست است و غیره  
 ② یک و اضافه می کنیم که  $a$  به دور  
 اماره همدگر به آن بر می گردیم

$$S \Rightarrow a x b c \Rightarrow a b x c \Rightarrow a y b c c \Rightarrow a y b b c c \Rightarrow$$

$$a a x b b c c \Rightarrow^* a a b b x c c \Rightarrow a a b b y b c c c \Rightarrow^+$$

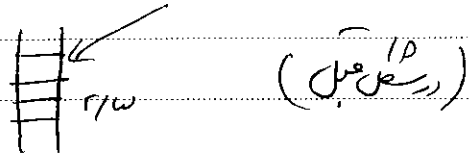
$$a a y b b b c c c \Rightarrow a a a b b b c c c$$

کاهش طول مجاز نیست اما فقط حالتی که  $\alpha$  را می خواهم آنرا می توانیم به این شکل  
 ردی به Start به سردی \*\*\*

# Transducer

خوبی م. ۴۸ و No محدود نیست و هر خوبی general را تولید می کند.

که خوبی را روی حافظه (r/w) نشان می دهد



ماشین کامل  
 با قابلیت های Turning Machine (نوع محدود) : (حافظه و طرف نامحدود)

ماشین که انداخته  
 در محدودیت حافظه ماشین بین باید می شود. LBA  
 ماشین PDA  
 در این حافظه تبدیل به Stack

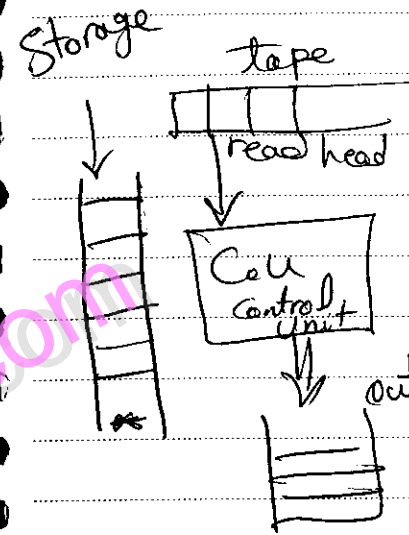
قدرت ماشین که برای محدود نوع ۲ محدود می کند که بدان محدود به است و PDA  
 متعلق از متن

ماشین FSA  
 اگر ماشین بدون حافظه شود. ماشین حالت ساده می شود. FSA  
 (گرامر محدود به نوع ۲) گرامر منظم

تفاوت این ماشین = روی میل حافظه است

تفاوت این ماشین = روی مقدار حافظه مورد نیاز

خوبی م. ۴۸ و No محدود نیست و هر خوبی general را تولید می کند.



- انواع Automata
1. Finite-state Automata (FSA)  $L_3$
  2. Push-Down Automata (PDA)  $L_2$
  3. Linear Bounded Automata (LBA)  $L_1$
  4. Turning Machine  $L_0$

از روی tape می خواند  
 خوانده می شود که در این حالت  
 حافظه (Storage) است.  
 حافظه از طرف نامشخص است

Turning Machine : که قابلیت های ماشین

(امم مفر Automation)

پذیرنده (مصرف کننده)  
 Acceptor  
 تولید کننده  
 Transducer

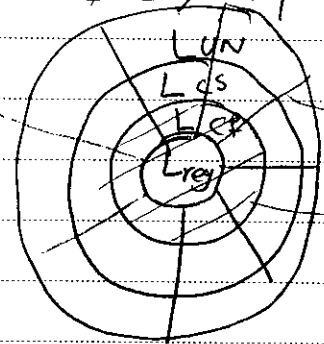
Acceptor  
 یعنی اگر برای زبان خاصی رشته از روی tape می خواند آن رشته

yes if we  
 No if not

گزاره ۲: زبان منظم  $\Sigma^*$  است. هیچ مجموعه‌ای نیست  $\Sigma^*$  که این گزاره غلط است.

گزاره ۳: خانواده زبانهای منظم زیر مجموعه محفل خانواده زبانهای مستقل از محفل است. (صحیح است و دلیل فوق)

گزاره ۴: خانواده زبانهای منظم زیر مجموعه محفل خانواده زبانهای نامنظم است. (غلط است زیرا زبان نامنظم با منظم اشتراک  $\emptyset$  است).



که  $L_{reg}$  منظم و  $L_{cs}$  نامنظم است.

بسیار بسیار مهم

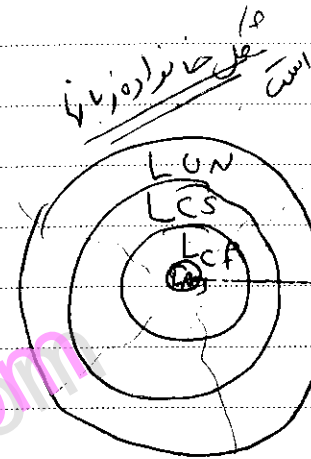
خانواده زبانهای منظم یعنی همه زبانهای که با گرامر منظم قابل توصیف هستند.

خانواده زبانهای مستقل از محفل

ممکنه زبانی باشد که با گرامر مستقل از محفل قابل توصیف باشد اما با گرامر منظم نه. اما از زبانی که با گرامر منظم توصیف می‌شود با محفل مستقل است.

همه زبانها خانواده از زبانها داریم.

انواع زبان (خانواده)



۱- زبانهای منظم  $L_3 = L_{reg}$   
 ۲- مستقل از محفل  $L_{cp}$   
 ۳- حساس به محفل  $L_{cs}$   
 ۴- بدون محدودیت  $L_{un}$  ( $L_{un} = L_4$ )

گزاره ۱: برای هر زبان منظم A یک زبان نامنظم B وجود دارد که  $A \subseteq B$ .

گزاره ۲: برای هر زبان منظم A یک زبان B وجود دارد که  $A \subseteq B$ .

خانواده زبان منظم: مجموعه زبانهای که گرامر منظم وجود دارند.

خانواده زبانهای مستقل از محفل: گرامر مستقل از محفل برای آن وجود دارد.

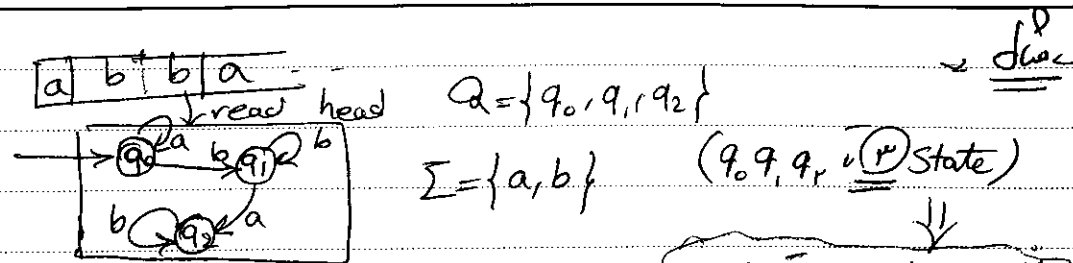
خانواده زبان = مجموعه‌ای از زبانها که با گرامر منظم قابل توصیف هستند.

گزاره اول غلط است. زیرا اگر  $\Sigma^*$  است.  $\Sigma^*$  زبان منظم است.

باب ۲: رگر منظم وجود ندارد. زیرا  $\Sigma^* \subseteq \Sigma^*$ .

$\Sigma^*$  منظم است و نمی‌تواند نامنظم باشد.





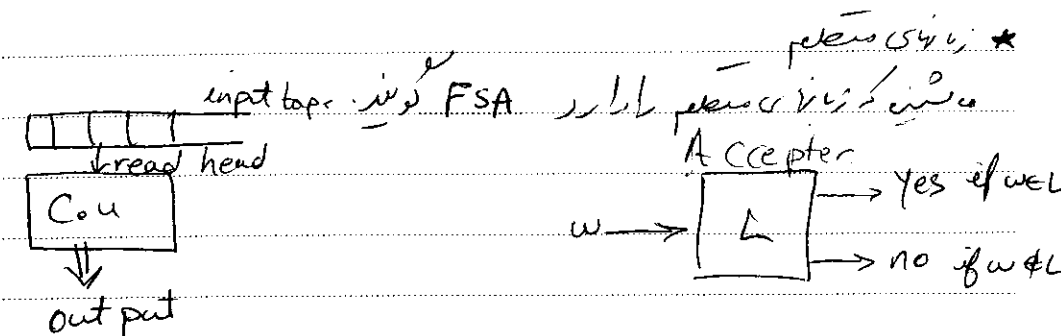
Transition Table

$\delta$	a	b
$q_0$	$q_1$	$q_2$
$q_1$	$q_2$	$q_0$
$q_2$	$q_2$	$q_2$

$\delta^*(q_0, a b b a) = ?$   $q_2$   
 $\delta^*(q_0, b a a)$

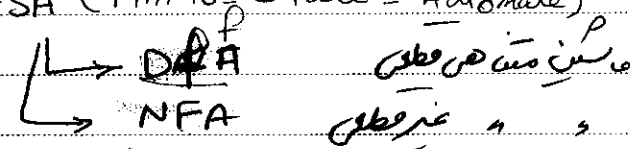
« توقف پذیرش توسط DFA »

اگر در پایان یک رشته ماشین در یکی از حالات نهایی قرار بگیرد پس این رشته پذیرفته می شود. و توقف پذیرش توسط DFA را می بینیم.  
 اگر در پایان یک رشته ماشین در یکی از حالات غیر نهایی قرار بگیرد پس این رشته نپذیرفته می شود.



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

FSA (Finite State Automate)



$\delta: Q \times \Sigma \rightarrow Q$   
 $\delta: Q \times (\Sigma \cup \{\epsilon\}) \rightarrow 2^Q$

State  $Q$   
 مجموعه  $Q$  را به اول State  
 حرف  $\Sigma$  را به حرف



چون حالت نهایی است پس این  
 رشته پذیرفته می شود.

$\delta^*(q_0, abba) = ?$

$\delta(\delta^*(q_0, abb), a) = q_2$

$\delta(\delta^*(q_0, ab), b) = q_1$

$\delta(\delta^*(q_0, a), b) = q_0$

زیر ماشین دفا «

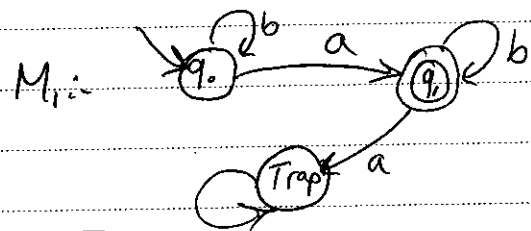
$L(M) = \{w \mid w \in \Sigma^*, \delta^*(q_0, w) \in F\}$

رشته های که نمی شوند

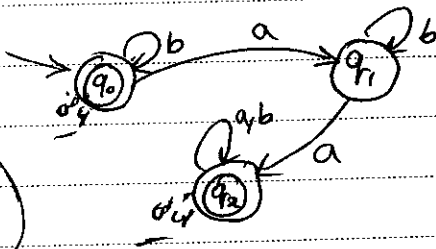
$\overline{L(M)} = \{w \mid w \in \Sigma^*, \delta^*(q_0, w) \notin F\}$

Compliment

\* یک State, Trap را می توان تبدیل به یک حالت نهایی و غیره یا به هم وصل  
 می شود.



$L(M_2) = \overline{L(M_1)}$



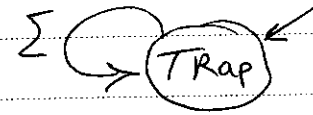
Compliment یک DFA کامل!

Com Trap

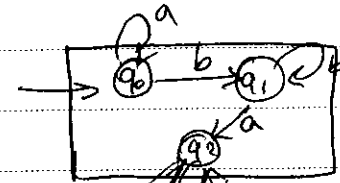
P4PCO

توان State, Trap

یک State نهایی است که در هر پروف می تواند بسته می شود



هر State می تواند هر حرفی که ندارد را می بیند



برای رشته های زبان را تولید می کند  
 ماشین = رشته های زبان را تشخیص می دهد

$a^n b^m a$   
 $n \geq 0, m \geq 1$

تابع  $\delta^*$   
 Extended  $\delta, \delta^*$

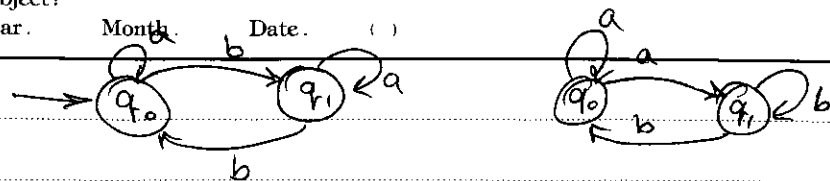
$\delta^*: Q \times \Sigma^* \rightarrow Q$

$\delta^*(q, \lambda) = q$

$\delta^*(q, wa) = \delta(\delta^*(q, w), a)$

\* از یک State به یک State

در یک dfa استناد به Trap را می بیند؛ اما برای State  
 نمی تواند در Trap سرگردان



\*DFA:

$$\begin{aligned}\delta(q_0, a) &= q_0 \\ \delta(q_0, b) &= q_1 \\ \delta(q_1, a) &= q_0 \\ \delta(q_1, b) &= q_1\end{aligned}$$

\*NFA:

$$\begin{aligned}\delta(q_0, a) &= \{q_0, q_1\} \\ \delta(q_0, b) &= \{q_1\} \\ \delta(q_1, a) &= \{q_0\} \\ \delta(q_1, b) &= \{q_0, q_1\}\end{aligned}$$

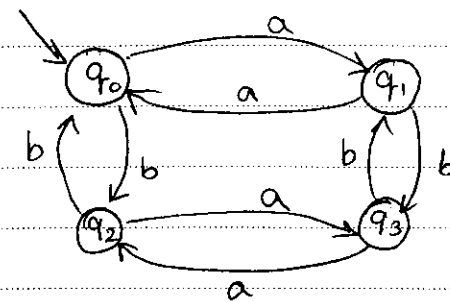
NFA  $\delta = Q \times (\Sigma \rightarrow 2^Q)$   
 Power set (مجموع توانی های Q)

تعریف NFA

اگر حداقل یک مسیر وجود داشته باشد یا چند مسیر باشد یا هیچ یکی از این مسیرها در یک حالت نهایی متوقف نشود.

تعریف NFA

اگر هیچ مسیری وجود نداشته باشد یا هیچ یکی از این مسیرها در یک حالت نهایی متوقف نشود.



(Final)  $F \leftarrow q_3$   
 آنگاه تعداد a, b فرد هستند

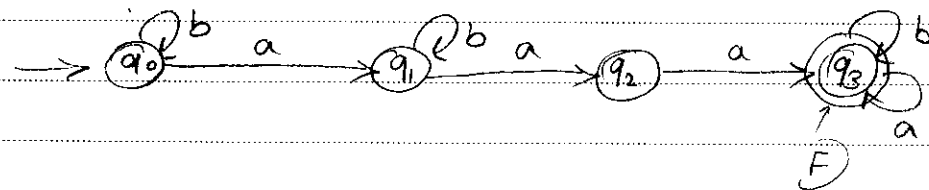
$F \leftarrow q_0$  آنگاه تعداد a, b زوج هستند.

$F \leftarrow q_2$  آنگاه تعداد a زوج، b فرد هستند.

$F \leftarrow q_1$  آنگاه تعداد a فرد، b زوج هستند.

مسئله: یک DFA طراحی کنید که پذیرای زیر باشد.

$$L = \{w \mid n_a(w) \geq 3\}, \Sigma = \{a, b\}$$



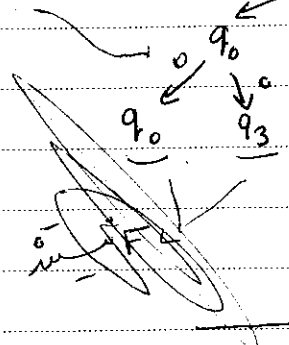
تعریف DFA (FSA) به یک DFA گفته می شود اگر در هر کدام از State

آن یک یا از هر کدام از الفبا فقط یک حرکت وجود داشته باشد.

تعریف NFA

به یک (FSA) NFA گفته می شود اگر حداقل یکی از این

مسیرها از الفبا بیش از یک حرکت داشته باشد.



$$\delta(P, a) = \bigcup_{P \in P} \delta(P, a)$$

شوری سے قطعاً نہ

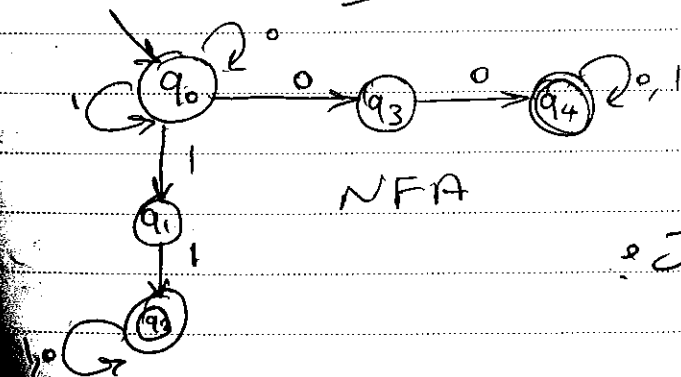
چون در NFA می‌توانیم از هر حالت به هر حالت دیگری برویم.

تولید رسمی NFA

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

مکمل NFA (Compliment) / تبدیل NFA میں DFA /

قسم ۲ و ۳ NFA یا DFA میں تبدیل کرنا (وہاں جس)



پیش از سید (۲) منشی صدر

کایرہ ۱۵ نومبر ۱۹۵۷ء

نکته: در ماشین قبل [q] افزودن می بینیم زیرا می توانیم وارد شدن کرده و می توانیم حرف

مورد:

نکته: هم در ماشین قبل [ ] نقش Trap را بازی می کند که وی دورش loop می زند

چون در ماشین اول  $q_0$  با حرکت نداشته که حالتی این ماشین از  $q_0$  با

$b$  به آن رفته است. \* [ ] اند و چون در ماشین قبل  $p$  و فقط  $p$  می باشد

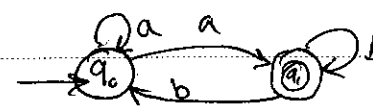
$$\begin{aligned} \text{DFA} \quad \delta'([q_0 \dots q_i], a) &= [p_0 \dots p_j] \\ \text{iff} \rightarrow & \text{ (if only if)} \\ \text{NFA} \quad \delta(\{q_0 \dots q_i\}, a) &= \{p_0 \dots p_j\} \end{aligned}$$

Final های ماشین جدید هر State که در State می تواند State  
 فاینال ماشین قبل را داشته باشد

تبدیل  $NFA \rightarrow DFA$  (تبدیل  $NFA$  به  $DFA$ )

در ماشین  $NFA$  امر واضح است به  $DFA$  تبدیل کنیم بطوریکه زبان تغییر نکرده باشد

$$\begin{aligned} M: \{Q, \Sigma, \delta, q_0, F\} &\longrightarrow M': \{Q', \Sigma', \delta', F'\} \\ \downarrow \text{NFA} & \quad \downarrow \text{DFA} \\ L(M) &= L(M') \end{aligned}$$



$$\begin{aligned} \delta(q_0, a) &= \{q_0, q_1\} \\ \delta(q_0, b) &= \{ \} \\ \delta(q_1, a) &= \{ \} \\ \delta(q_1, b) &= \{q_0, q_1\} \end{aligned}$$

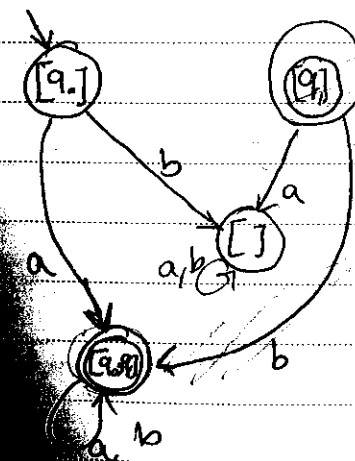
$$Q' = 2^Q = \{[ ], [q_0], [q_1], [q_0, q_1]\}$$

$$\Sigma' = \Sigma$$

$$\delta' = ?$$

$$q'_0 = [q_0]$$

$$F' = ?$$



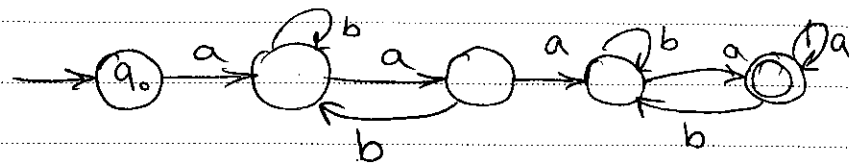
$\delta'$	a	b
$[q_0]$	$[q_0, q_1]$	$[ ]$
$[q_1]$	$[ ]$	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_1]$	$[q_0, q_1]$
$[ ]$	$[ ]$	$[ ]$

$$\begin{aligned} \delta(\{q_0, q_1\}, a) &= \delta(q_0, a) \cup \delta(q_1, a) \\ &= \{q_0, q_1\} \cup \{ \} = \{q_0, q_1\} \end{aligned}$$

که امر ماشین را در P

Subject: \_\_\_\_\_  
Year. \_\_\_\_\_ Month. \_\_\_\_\_ Date. \_\_\_\_\_

$$L^2 = \{awaaw'a \mid w, w' \in \{a, b\}^*\}$$



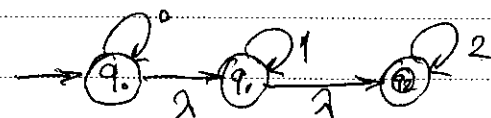
(2, 5, 6\*, 7, 8, 9, 10, 11, 14, 17, 23)

(آخرین غیبی ۲-۱)

$M = NFA_n$

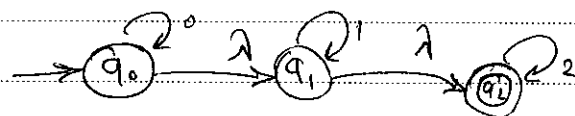
در اینجا

ما به دریم روی یک حالت به حسب  $\lambda$  داشته باشیم یعنی از  $q_0$  بدون توجه به ورودی



بهتر  $q_1$

آرک ماشین این دو حالت را داشته باشد پس می توانیم  $M = NFA_n$



0012

002

1

0021

$\delta$	0	1	2	$\lambda$
$q_0$	$\{q_0\}$	$\{1\}$	$\{1\}$	$\{q_1\}$
$q_1$	$\{1\}$	$\{q_1\}$	$\{1\}$	$\{q_2\}$
$q_2$	$\{1\}$	$\{1\}$	$\{q_2\}$	$\{1\}$

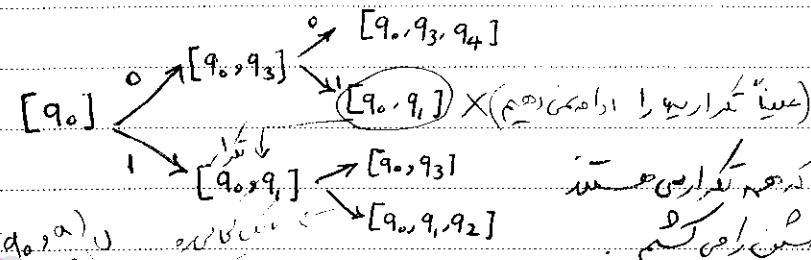
$$L = \{0^m 1^n 2^p \mid n, m, p \geq 0\}$$

PAPCO

Subject: \_\_\_\_\_  
Year. \_\_\_\_\_ Month. \_\_\_\_\_ Date. \_\_\_\_\_

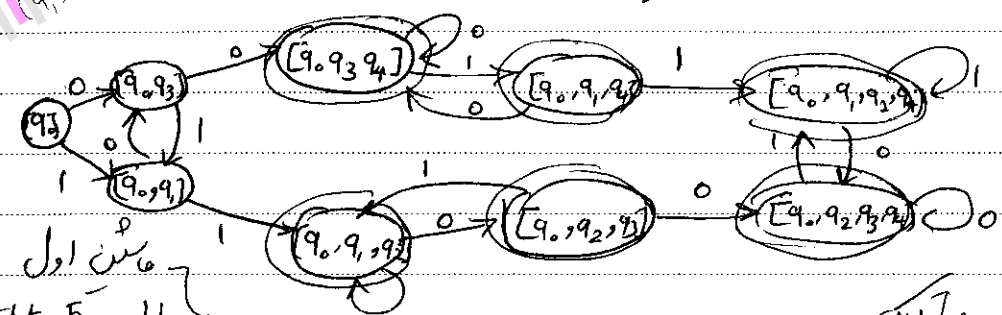
(2)

مثال تبدیل کننده  $P$   
مان ماشین قبل به روش دیگر  $P$



(q0, q0) و (q1, q0)

حالت آخری می بینیم که همه تقارن هستند  
حال همان طور ماشین را می کشیم



ماشین اول NFA

دارای 5 state بود

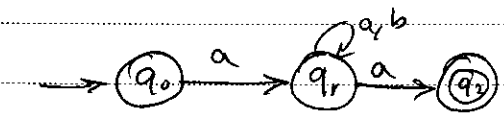
ماشین DFA دارای

5 state

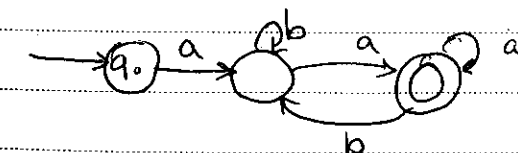
در وقت ساده کردیم

مقطار  $q$

حالت



DNF



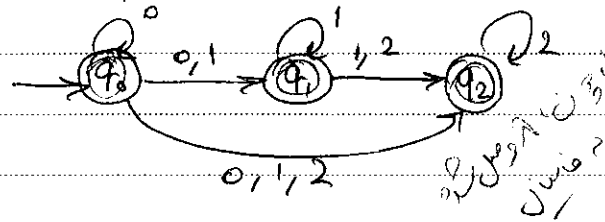
NFA



$\delta'$	0	1	2
$q_0$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
$q_1$	$\{\}$	$\{q_1, q_2\}$	$\{q_2\}$
$q_2$	$\{\}$	$\{\}$	$\{q_2\}$

تمام فاینالهای قبلی که فاینال  
 بر ریزد این هم فاینال هستند

State های فاینال هستند  
 Closure های آن به  $q_0$  فاینال



حالت نهایی State فاینال  
 $F' = F \cup \{q_i \mid \lambda\text{-closure}(q_i) \cap F \neq \emptyset\}$

فاینالهای قبلی، فاینال هستند علاوه بر آن  $\lambda\text{-closure}$  آن State  
 به  $F$  تعلق داشته باشد

همه State های که صرفاً با حرکت از  $q_0$  به Final رسیده باشند

همه Final هستند

به این حساب  $q_0, q_1, q_2$  همه فاینال هستند

در  $q_0$  و  $q_1$  می توانیم به  $q_2$  Final برسیم



البته  $q_1$  را می توان از حالت فاینال خارج کرد زیرا می توان به  $q_2$

رفته اما  $q_0$  را نمی توان حذف کرد چون از می خواهیم داشته باشیم

وقتی فاینال را داریم nfa انت زیرا dfa ندارد

فنی  $\lambda$  وجود عدم قطعیت است

$\lambda\text{-closure}$  بسته  $\lambda$

$\lambda\text{-closure}(q)$  مجموعه حالتی است که از  $q$  بدون مصرف ورودی می توان در آنجا  
 حضور داشت

$$\lambda\text{-closure}(q_0) = \{q_0, q_1, q_2\}$$

$$\lambda\text{-closure}(q_1) = \{q_1, q_2\}$$

$$\lambda\text{-closure}(q_2) = \{q_2\}$$

$\lambda\text{-NFA} \rightarrow \text{NFA without } \lambda\text{-moves}$

$$M = (Q, \Sigma, \delta, q_0, F) \xrightarrow{\lambda\text{-NFA}} M' = (Q', \Sigma', \delta', q', F')$$

$\lambda\text{-NFA}$

NFA: without  $\lambda\text{-moves}$

$$L(M) = L(M')$$

بنابراین همانند  $q_0$  می هستند چون می توانیم به  $q_2$  برسیم

چون تکرار است اگر در  $q_1$  هستیم می توانیم به  $q_2$  برسیم

که اگر هم به  $q_0$  برسیم می توانیم به  $q_2$  برسیم

تجس ۲-۶ از زبان صلا الوکس

Subject: Year: Month: Date: ( )

۷-۸-۱۰-۱۱-۱۴-۲۰-۲۱  
توب incomplete

تجس ۲-۲

۵-۶-۷-۹-۱۱-۱۲-۱۴

تجس ۲-۳

۲-۴-۵-۶-۷

فصل ستم زبانهای منظم و گرامرهای منظم

Regular Expressions (RES)

عکس زوری عبارات منظم است

Definition:

گرامر برای تولید رشته‌های یک زبان \* گرامر برای صسته  
عبارت منظم برای توصیف زبان منظم. عبارت منظم برای صسته  
میشود برای پذیرش یا تشخیص یک زبان. ماسین و د و ؟

این عبارتهای پایه‌ای زبانهای منظم هستند

1,  $\emptyset$  is a RE.

$L(\emptyset) = \{ \}$

2,  $\lambda$  " "

$L(\lambda) = \{ \lambda \}$

3,  $a \in \Sigma$  " " هر الفبای صلا الوکس

$L(a) = \{ a \}$

4, if  $r_1$  and  $r_2$  are 2 RES

Then

$r_1 + r_2$

$L(r_1 + r_2) = L(r_1) \cup L(r_2)$

$r_1 \cdot r_2$

$L(r_1 \cdot r_2) = L(r_1) \cdot L(r_2)$

$(r_1)^*$

$L(r_1^*) = L(r_1)^*$

PAPCO

are RE, too.

Subject: Year: Month: Date: ( )

نیا این می‌توانیم تدریس  $F$  را به این صورت تفسیر دهیم

$F' = F \cup \{ q_0 \mid \lambda \in L(M) \}$  این را به این صورت می‌توانیم تفسیر دهیم که  $F$  را به  $F'$  اضافه می‌کنیم و  $q_0$  را به  $F'$  اضافه می‌کنیم.  $q_0$  هم فایده‌ای دارد.

به سطر که در زبان (مکسین) قبل از پذیرفته می‌شود

$\delta'(q, a) = \epsilon\text{-closure}(\delta(\delta^*(q, \epsilon), a))$

در بعضی مواقع  $\epsilon\text{-closure}$  /  $\lambda\text{-closure}$  استفاده می‌شود

$\delta'$	0	1	2
$q_0$	$\{q_0, q_2\}$	$\{q_1, q_2\}$	
$q_1$		$\{q_1, q_2\}$	
$q_2$			$\{q_2\}$

$\{q_1, q_2\}$   
 $\{q_1, q_2\}$   
 $\{q_2\}$

①  $(ab)^*(a+bb)$   $\rightarrow$  ترکیب  $a, b$  به  $a$  و  $bb$  می شوند.

②  $(aa)^*b(bb)^*$  زوج  $a$  شروع شوند به دنبال آن  $b$  و  $bb$ .

همیشه  $a, b, c$  که به هم پیوسته می شوند به دنبال  $c$  می آیند.  
 همیشه  $abc$  که به دنبال آن  $c$  می آید به  $(c+\phi)$  می شود.  
 $(a+bc)^*(c+\phi)$  اگر  $\phi$  به  $\phi$  می آید و  $\phi = \phi$  است.

①  $(a^*)^* = a^*$

$a^* = \{w \mid w = a_1 \dots a_i, i \geq 0, a_i = a\}$

$(a^*)^* = \{w \mid w = a_1 \dots a_i, i \geq 0, a_i = a_1 \dots a_j, j \geq 0, a_j = a\}$

همیشه به هم می پیوندند.

②  $\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma$

③  $\alpha\alpha^* = \alpha^*\alpha$

④  $\alpha\alpha^* + \lambda = \alpha^* + \lambda = \alpha^*$

⑤  $(\alpha + \beta)^* = (\alpha^* + \beta^*)^* = (\alpha^*\beta^*)^* = (\alpha^*\beta^*)^*$   $\alpha^* = \alpha + \lambda$   
 $\beta^* = \beta + \lambda$

$a^*(a+b) = ?$   $L(a) = \{a\}$   $L(b) = \{b\}$   
 $L(a+b) = L(a) \cup L(b)$   
 $L(a+b) = \{a, b\}$

$L(a)^* = \{a\}^*$

$\Rightarrow L(a^*(a+b)) = \{a\}^* \cdot \{a, b\}$

$\lambda \cdot a = a$   $\Rightarrow \{a, aa, \dots\} \cdot \{a, b\}$   
 $(\text{نوع اول}) = \{a, b, aa, ab, aaa, \dots\}$

①  $(a+b)^*$

②  $(a+b)^*bb(a+b)^*$   $\rightarrow$  همیشه  $a, b$  می آیند.  
 همیشه  $bb$  می آیند.

③  $(bb)^*$

همیشه  $bb$  می آیند.

④  $(bb)^*b$

همیشه  $bb$  می آیند و در آخر  $b$ .

⑤  $(bb)^* + b(bb)^* = b^*$

نکته: عبارت منظم زیر مندرج شده است

$$(a+b)^* b (a+ab)^*$$

1) 10

2) 11

3) 12

4) 13

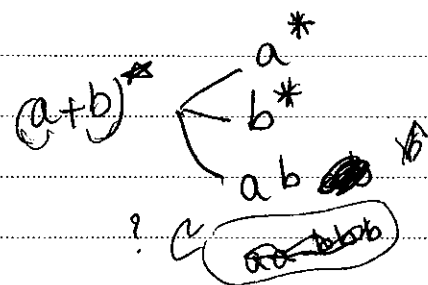
حل:

نادر: 0 بطول

3.

1: b	2 1 0	1 1 1	0 1 2
2: ab	aa b b	a b a	b a a
bb	abb	b ba	bab
ba	ba b		
	bb b		

بسیار  
 ۶-۷-۱۱-۱۳-۱۴-۱۶-۱۷-۱۸-۲۱  
 ۲۴-۲۷



$$\textcircled{6} \alpha(\beta\alpha)^* = (\alpha\beta)^*\alpha$$

$$\textcircled{7} \alpha(\beta\alpha + \alpha)^* = (\alpha\beta + \alpha)^*\alpha$$

مثال: دو عبارت منظم داریم که معادلند یا خیر؟

$$R_1 = (1^* 0 1 1^*)^* (0 + 1) + 1^* (0 + 1) \quad (1^* 0 1 1^* + 1^*) (0 + 1)$$

$$R_2 = \frac{(1 + 0 1)^*}{y} (0 + 1)$$

تمام رشته‌های ۰ و ۱ که هیچ دو صفر متوالی ندارند

$$\overline{R_1} \cup R_1 = \underbrace{((1^* 0 1 1^*)^* + 1^*)}_{x} (0 + 1)$$

$$y = \{ w \mid w = \alpha_1 \dots \alpha_i, i \geq 0, \alpha_i = 1 \text{ or } 0 \}$$

$$x = \{ w \mid w = \alpha_1 \dots \alpha_i, i \geq 0, \alpha_i = 1 \text{ or } 0 \}$$

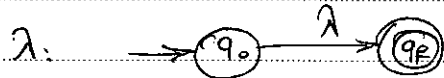
و

$$\{ w \mid w = \alpha_1 \dots \alpha_i, i \geq 0, \alpha_i = 1 \text{ or } 0 \}$$

$$= y$$

سپارین دو زبان معادلند

RE  $\rightarrow$   $\lambda$ -NFA



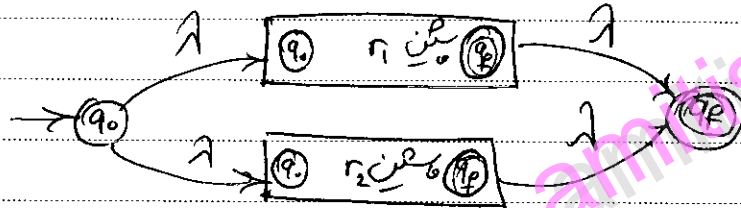
مسارهای ممکن حالت  
 $\lambda$ -NFA تبدیل سیستم عبارت

بین باریک شدن به ابتدای و انتهای

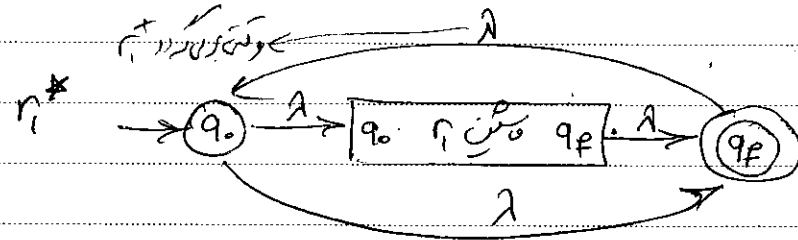
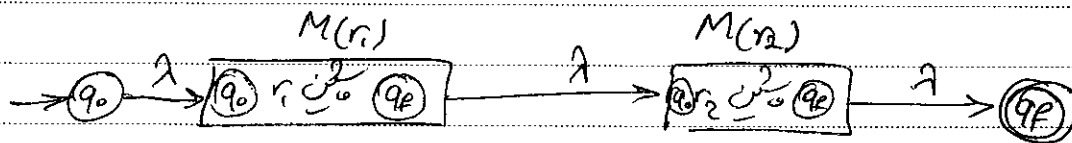
State ها یک حالت

اضافه کردن حتماً به هم ربط  
 در هم

$r_1 + r_2$ :



$r_1 \cdot r_2$ :



سوال: عبارت منظم بنویسید که ناانتهای زبان زیر را  
 $L = \{a^n b^m \mid n \geq 1, m \geq 1, nm \geq 3\}$

نظ:

$n \geq 1, m \geq 3$   $aa^* bbb^*$   
 $n \geq 3, m \geq 1$   $aaaa^* bb^*$   
 $n=2, m=2$   $aa bb$

چون رشته های کوتاه را  
 تولید می کند و زبان  
 محدود تر می شود.

$n=1, 2, m \geq 3 (a+aa)^* bbb^*$

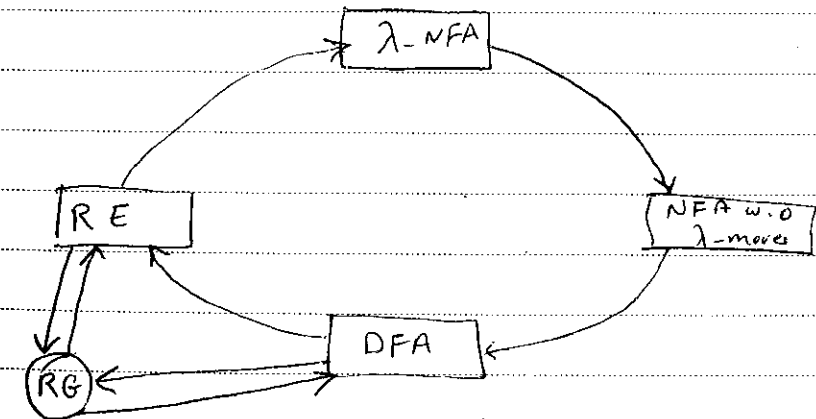
$L = \{a^n b^m \mid n+m \geq 3\}$

$(aa)^* (bb)^*$   
 $+$

یا هر دو زوج

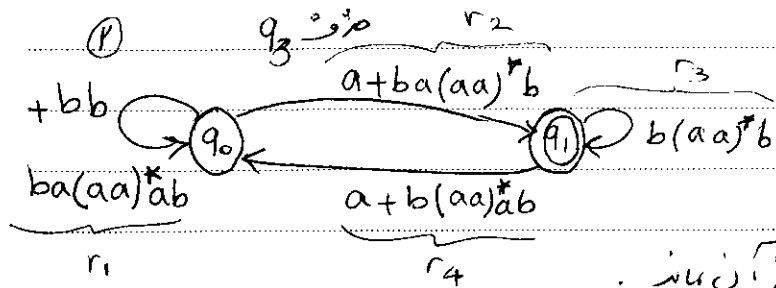
$(aa)^* a (bb)^* b$

یا هر دو فرد



هم قابل تبدیل به یکدیگر هستند.





وقتی حالتی حذف می شود باید از آن بماند.

وقتی تعداد فاینالها بیشتر از یک باشد مراحل با جلا خطای این مورد نیز را می بینیم.

$$RG \leftrightarrow RE$$

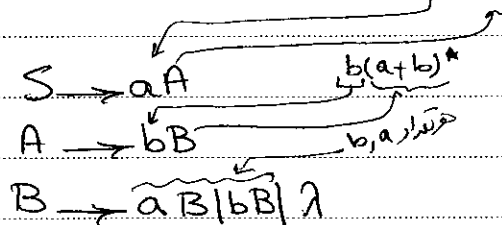
عبارت منظم و گرامر منظم

$$ab^*b(a+b)^* \equiv abb^*(a+b)^*$$

ساده کردن عبارت

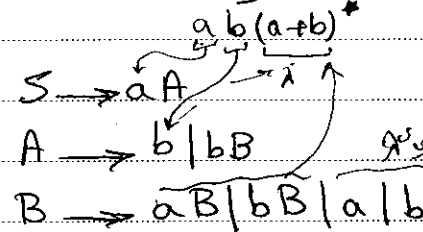
$$\equiv ab(a+b)^*$$

بهر صورت منظم



بگذاریم منظم از راست به چپ

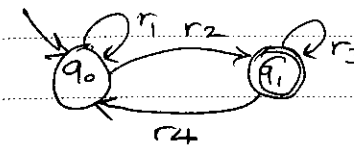
اگر بتوانیم از راست به چپ (یعنی توی گرامر نقض نمی آید)



از اینجا می بینیم که a و b از سمت چپ می آید چون در اینجا می بینیم که b و a از سمت چپ می آید.

سبب چیست

DFA or NFA  $\rightarrow$  RE



۱. اول سعی کنید به فاینال برسید.

۲. سعی کنید روی فاینال بمانید.

$$r_1^* r_2 (r_3 + r_4 r_1^* r_2)^*$$

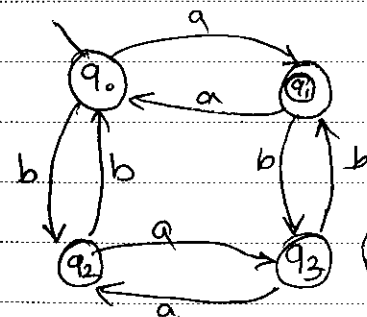
روی فاینال بمانیم. یا فاینال را بگذرانیم.

یعنی هر گویی از اینجا مارا به فاینال می ریزد.

گرامر منظم تعمیم یافته GTG: Generalized Transition Graph

اینها می آید. روها هر یال گراف دارای تعمیم یافته ی عبارت منظم قرار می گیرند.

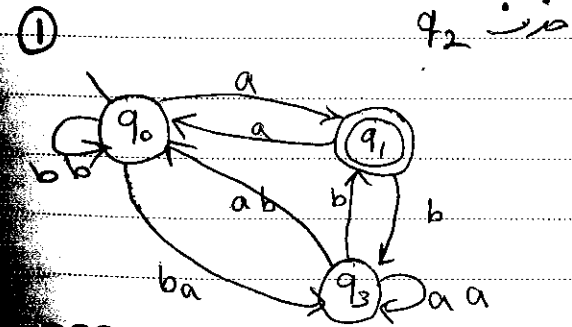
آزاد تئوری تعمیم نکرده فقط نحوه نمایش تعمیم می کند.



هدف ساده سازی این ماشین

از اینجا که a ها و b ها در اینجا می آید.

توسط GTG



Subject:  
Year:

Month:

Date:

( )

برای نوشتن گرامر از راست به چپ شروع می‌کنیم.

گرامر از راست از اول رشته را می‌سازد.

گرامر از چپ از آخر رشته را می‌سازد.

منظم

به ازای شروع می‌کنیم.

چون گرامر منظم خطی از چپ به راست را می‌سازد.

رشته شروع می‌کنیم.

در آخر هر مرحله اولش را می‌سازیم.

مثال: گرامر منظم زبان زیر را بنویسید.

$abba(aba)^*bb$

$S \rightarrow Abb$

$A \rightarrow Aaba | abba$

$L((aab^*ab)^*)$

$S \rightarrow aaA | \lambda$

$A \rightarrow bA | ab$

برای این که \* را تعریف کنیم (\* در فرم از ( ) )

به چپ S را می‌سازیم.

دو وقتی تمیز می‌دهیم که می‌توانیم S را بنویسیم.

حالت تکمیل را می‌نویسیم.

(برای \* باید یک بار حلقه را بنویسیم)

Subject:  
Year:

Month:

Date:

( )

برای نوشتن DPA از راست به چپ شروع می‌کنیم.

برای این که گرامر زبان منظم بنویسیم باید از راست به چپ خطی را می‌سازیم.

از راست همان منظم از راست است و گرامر خطی از چپ به راست منظم از چپ به راست است.

Right-Linear vs. Right-Regular & Left-linear vs.

Left-Regular

$$\frac{1^*00 + 11^*0}{1^*00 + 11^*0} \equiv 1^*00 + 1^*10 = 1^*(0+1)0$$

$S \rightarrow A00 | 1A0$

$A \rightarrow 1A | 1$

$$1^*(0+1)0$$

$S \rightarrow 1S | 0A | 1A$

$A \rightarrow 0$

مثال: برای گرامر منظم از راست زیر یک عبارت منظم و یک گرامر معادل منظم از چپ بنویسید.

$S \rightarrow abA$

$A \rightarrow baB$

$B \rightarrow aA | bb$

کوچکترین

$abba bb$

$abba abba$

$abba(aba)^*ba bb$

$abba(aba)^*abb$

$abba(aba)^*bb$

در هر مرحله یک معادله

# خانواده زبانهای منظم

Subject: Year: Month: Date: ( )

توضیح چهارم

خصوصیات زبانهای منظم

(خصوصیات دیگری زبانهای منظم)

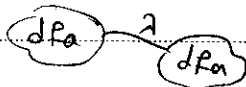
تغییر بسته بودن خانواده زبانهای منظم تحت عملگرهای

~~L1 ∪ L2 : dfa~~

خانواده زبانهای منظم تحت عملگرهای اجتماع، کات، \*، +، و مکمل بسته است.

$L_1 \cup L_2 : dfa$   
grammar  $\rightarrow S \rightarrow S_1 | S_2$

RE



$L_1 \cdot L_2 : dfa$

دو ماشین را به هم وصل

grammar  $\rightarrow S_1 \rightarrow S_2 \rightarrow aS_2 | abS_2$

در گرامر زبان اول S به یک a وصل می شود و به S2 می نهد که می نهد  
S2 را برایش می نهد و هر زبان دوم قرار می نهد که می نهد

$L^* : dfa$

از روی dfa صحت آن را nfa درست کرد

grammar  $\rightarrow S_1 \rightarrow \lambda$

$aS_1 | abS_1$

آنها که غیر ترسیدند با تمام می شوند که S1 صحت آن را هم اضافه کنید

PAPCO

RE نیست \* بسته

Subject: Year: Month: Date: ( )

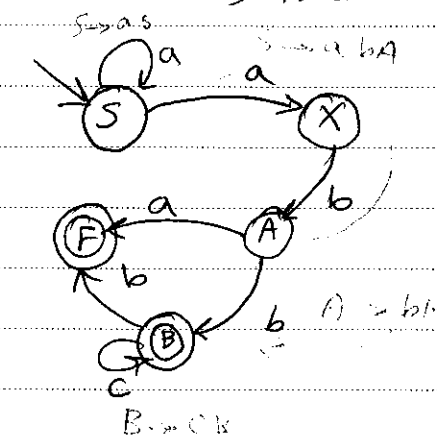
تغییر

تبدیل گرامر منظم به NFA و DFA

$S \rightarrow as | abA \rightarrow \begin{cases} S \rightarrow as | aX \\ X \rightarrow bA \end{cases}$   
 $A \rightarrow bB | a$   
 $B \rightarrow b | CB | \lambda$

برای ساده شدن حالت فوق را باز کرده

\* می داند هرگز نباید در واقع نیست  
state داریم



چگونه می توان یک گرامر منظم از یک NFA تبدیل کرد یا بالعکس؟ باید گفت  
۳۵ کتاب را بخوانید

۵-۷-۸-۱۰-۱۳-۱۶-۱۸

کتاب ۲

۶-۷-۹-۱۲-۱۳-۱۷

کتاب ۳

به ۱۷ اضافه کنید آیا G منظم است؟

PAPCO

قضیه: خنواره زبانهای منظم گشت Reverse و تقابل معکوس است.

\*  $L_1^R$

\*  $L_1 \circ L_2 = L_1 \cap L_2$

(تایید هم میسر می شود طریقی بزرگ)

homomorphism

تایید است که از  $\Sigma^*$  است.  $\Gamma$  مجموعه الفبا است.  $\Gamma^*$  مجموعه رشته های متعلق از الفبا.

$h: \Sigma \rightarrow \Gamma^*$

تصویر  $a$  از الفبا  $\{a, b\}$  روی  $\Gamma$ .  $\Sigma = \{a, b\}$   $\Gamma = \{x, y, z\}$

$h(a) = xxy$   $h(b) = xz$

تصویر homomorphism یک زبان = تصویر یک رشته های آن زبان.

$h(L) = \{h(w) | w \in L\}$

مثال:  $L = \{a^2b, bb\} \Rightarrow h(L) = \{xxyxxyxz, xzxz\}$

$h(a) = xxy$   
 $h(b) = xz$

خنواره زبانهای منظم گشت

$h(L_1 \cup L_2) = h(L_1) \cup h(L_2)$  homomorphism معکوس است.

$h(L_1 \circ L_2) = h(L_1) \circ h(L_2)$

$h(L_1 \cap L_2) \neq h(L_1) \cap h(L_2)$

نوعی نمی باشد

$L_1: dfa$  جاری تر منی و غیر تر منی، احوال کرده پس اینهم  $dfa$  است.

$L_1 \cap L_2: dfa \rightarrow dfa(L_1) = (Q_1, \Sigma, \delta, q_1, F_1)$

$(L_1 \cup L_2)$

$dfa(L_2) = (Q_2, \Sigma, \delta, q_2, F_2)$

منظم

منظم  $\cup$  منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

منظم

$dfa(L_1 \cap L_2) = (Q_1 \times Q_2, \Sigma, \delta', q_1', F_1 \times F_2)$

از خروجی تر منی شروع کرد

از خروجی تر منی شروع کرد

قضیه ۱: کتاب را بخوانید و نتیجه را بنویسید.

$A(q_1, a) = q_2$

$\Rightarrow (q_1, p_1, a, q_2, p_2) \checkmark$

$B(q_2, a) = p_3$

دسته ماشین تمام شد اگر جایی بودیم که فانیل شروع بود در آن بود

فانیل ماشین بنویسید.  $\frac{4.1}{4.1}$

استاد (۲)  $dfa$  هم

مثال ۳)  $L_1 = \{a^n b^m \mid n \geq 1, m \geq 0\} \cup \{ba\}$

$L_2 = \{b^m \mid m \geq 0\}$

$m=0$  پس از آن به باقی می ماند

در اینجا  $a$  می تواند به  $a$  یا به  $b$  و وقتی  $a$  را از  $a$   $ba$  برابریم  $ba$  می ماند

پس  $ba$  را برمی گردانند درستی  $ba$  برقرار نیست (جواب نوز).

$L_1/L_2 = \{a^n b^m \mid n \geq 1, m \geq 0\} \cup \{ba\}$

مثال ۴)  $L_1 = \{a^n b^m \mid n \geq 1, m \geq 0\} \cup \{ba\}$  (مثال ۳)  $L_2 = \{a^n b^m \mid n, m \geq 0\}$

بر اساس  $a$  و  $a$  می تواند به  $a$  یا به  $b$  و وقتی  $a$  را از  $a$   $ba$  برابریم  $ba$  می ماند

$L_1/L_2 = \{a^n b^m \mid n \geq 1, m \geq 0\} \cup \{ba, b, \lambda\}$

تقریباً: خانواده زبان های منظم تحت عمل Right quotient است. (البته هم)

در اینجا وقتی  $a$  را از  $a$   $ba$  برابریم  $ba$  می ماند

پس وقتی  $a$  را از  $a$   $ba$  برابریم  $ba$  می ماند

تقریباً: خانواده زبان های منظم تحت عمل Right quotient است.

روی این تغییر

مثال ۵)  $L_1 = \{a^n b^m \mid n \geq 1, m \geq 0\} \cup \{ba\}$  (مثال ۳)  $L_2 = \{b^m \mid m \geq 0\}$

$L_1/L_2 = \{a^n b^m \mid n \geq 1, m \geq 0\} \cup \{ba\}$

مثال ۶)  $L_1 = \{a^n b^m \mid n \geq 1, m \geq 0\} \cup \{ba\}$  (مثال ۳)  $L_2 = \{b^m \mid m \geq 0\}$

تقریباً: خانواده زبان های منظم تحت عمل Right quotient است.

$L_1/L_2 = \{x \mid xy \in L_1 \text{ for some } y \in L_2\}$

مثال ۷)  $L_1 = \{a^n b^m \mid n \geq 1, m \geq 0\} \cup \{ba\}$  (مثال ۳)  $L_2 = \{b^m \mid m \geq 0\}$

مثال ۸)  $L_1 = \{a^n b^m \mid n \geq 1, m \geq 0\} \cup \{ba\}$  (مثال ۳)  $L_2 = \{b^m \mid m \geq 0\}$

مثال ۹)  $L_1 = \{a^n b^m \mid n \geq 1, m \geq 0\} \cup \{ba\}$  (مثال ۳)  $L_2 = \{b^m \mid m \geq 0\}$

$L_1 = \{ab, aba, bba\}$

$L_2 = \{a, aa\}$

$L_1/L_2 = \{ab, bb\}$

مثال ۱۰)  $L_1 = \{a^n b^m \mid n \geq 1, m \geq 0\} \cup \{ba\}$  (مثال ۳)  $L_2 = \{b^m \mid m \geq 0\}$

$L_1 = \{a^n b^m \mid n \geq 1, m \geq 0\} \cup \{ba\}$

$L_2 = \{b^m \mid m \geq 1\}$

$L_1/L_2 = ? \Rightarrow \{a^n b^m \mid n \geq 1, m \geq 0\}$

$L_1 = a^n b^m$   
 $L_2 = b$   
 $L_1/L_2 = a^n b^{m-1}$

$L_1 = \{a^n b^m \mid n \geq 1, m \geq 0\} \cup \{ba\}$



Subject :  
Year .      Month .      Date .      ( )

آیا خانواده زبان‌های منظم است؟  
 آهسته آهسته؟ نه درست است؟

$\chi^2 \leftarrow \text{reol}$ ,  $\text{Reol}$   $\leftarrow$   $\text{reol}$   
 $\text{reol} \leftarrow \text{reol}$

$$\text{head}(L) = L/5^*$$

موجود  $L$  روی  $l$  به است  $head(L)$  هم به است

\*  $dp[i][j]$  :  $i$  تا  $j$  رانگ را با یک رنگ می‌توان رنگ کرد یا نه

حکمہ State جہی سر راہ State فاسل راہ فاسل سبیل مرکب

Chức năng của đơn vị này là (phân công và điều hành)

the head

Start of new state no, head(h), dfa

Final 2 Final

Subject: \_\_\_\_\_  
 Year.      Month.      Date.      ( )

(Symmetric difference)  $A \oplus B$

$$S_1 \cap S_2 = \{x \mid x \in S_1, \text{ or } x \in S_2, \text{ but } x \notin S_1 \cap S_2\}$$

$$= (S_1 \cup S_2) - (S_1 \cap S_2)$$

$$= (S_1 - S_2) \cup (S_2 - S_1)$$

این بیکردن قبیله است.

Head(L) = {x | xy ∈ L for some y ∈ Σ\*}      Head(L) = {x | ∃ y, xy ∈ L}

$\mathcal{D}_w \therefore L_1 = \{101, 11, 0\}$

→ اترتے ہیں اور بار

از سید احمد

از تہ ش ۱۰۱ پر بار...

$$\text{Head}(L) = \{1, 0, 1, 1, 0, 1, 1, 1, 1\}$$

(۱) رسته نطوله بر دار نطول ایرار نطوله آه . . .

۴۶ این عمل درست است  $p \in \text{head}(L)$  و  $L \subseteq \text{head}(L)$  است

استخوانی head کپڑا لباس دھو دھو دھو

$$\text{head}(\Sigma^*) = \Sigma^*$$

$$\text{head}(A) = A$$

$$p \text{ (sw)} L = \text{head}_p \quad \text{sub } p \text{ (sw)}$$
[illegible]

$L = \text{head}(L)$ ; ~~... ..~~

آه ایست؟ فرمات که در  $\phi$  می باشد یعنی است. زیرا  $\phi$ !

قضیه تورینگ وجود دارد که می تواند برای هر زبان منظم  $L_1$  و  $L_2$  که به هم افزا باشند

ارائه شده اند تعیین کند آیا  $L_1 = L_2$  هست یا نه؟

دفا را می توانیم به کمک ارباب ن برداریم به هم میزنیم (dfa منیال، Uniq هست)

$$\begin{cases} L_3 = L_1 \cup L_2 \\ L_1 = L_2 \Rightarrow L_3 = \emptyset \end{cases}$$

(از تفاضل تفاوت دوزمان می توان آرا یافت)

نکته این جمله مستحق از متن صادر نیست. (طبق قضیه قبل حال می شود  $L_1 = L_2$  یا نه؟)

مثال: تورینگ ارائه دهد که تعیین کند آیا یک زبان منظم  $L$  یا پالیندروم Palindrome

می باشد یا خیر؟

تولید: زبان Palindrome نوشتار  $L = L^R$ ، زبان  $L$  و  $L^R$  یکسان است

که هر رشته که دارد Reverse برعکس داشته باشد.

۱- دفا  $L$  را داریم  $L_1$

۲- دفا  $L^R$  را داریم  $L_2$

۳- حال با قضیه قبل چک می کنیم  $L_1 = L_2$  است یا نه؟

کمترین ۱- ۵، ۷، ۹، ۱۰، ۱۱، ۱۲، ۱۵، ۱۶، ۱۹، ۲۱، ۲۳، ۲۵

۲- ۲، ۴، ۶، ۸، ۱۰، ۱۲، ۱۴، ۱۶، ۱۸، ۲۰، ۲۲، ۲۴، ۲۶، ۲۸، ۳۰

۴.۲ برخی پرسش های رایج در مورد زبان های منظم

تعیین تورینگ وجود دارد که می تواند برای هر زبان منظم  $L$  به هم افزا باشد ارائه شده

است تعیین کند که آیا  $L \in \Sigma^*$  عضو زبان  $L$  هست یا نه؟ جوابش

فرا بسته ندارد زبان های منظم چیست؟

۱. گرامر منظم

۲. عبارت منظم

۳. ماشین دفا

جواب: نمایش dfa مربوطه.

قضیه تورینگ وجود دارد که می تواند تعیین کند آیا یک زبان منظم  $L$  که به هم افزا باشد

ارائه شده است، متناهی است یا نامتناهی هست یا نه؟

در مورد dfa

اگر start و final، cycle داشته باشد زبان نامتناهی

\* اگر نه، صریح نداشته باشد زبان نامتناهی

این در مورد nfa این چه وقتی معتبر است که cycle غیر  $\epsilon$  وجود داشته باشد

(start و final)

آنها زبان نامتناهی است.

Subject: \_\_\_\_\_  
Year .      Month .      Date .      ( )

$$a^m b^m$$

رفق كنيم اين به معلم (m) و

$$x = a^{k_1}$$

$$k_1 + k_2 \leq m$$

$$y = \mathbb{A}^{K^2}$$

$$k_2 > 1$$

$$Z = a^{m-k_1-k_2} b^m$$

$$\omega_i^0 = a^{k_1} (a^{k_2})^i a^{m-k_1-k_2} b^m$$

$$\sum_{j=0}^n a_j x^j = 0 \quad \omega_0 = a \quad a^{m-k_1-k_2} \quad b^m$$

$$w_b = a^{m-k_2} b^m$$

کہ این در اسیست کہ L معظم نیست

بیکل کردن داخل است <sup>p</sup> نامیستم بودن یک زمان تر سرطالم Pumping به رقیابت در عرف ے

۲. با توجه به  $m$  اشیای بهرین،  $\omega \in L$  را بنویسید اشیای  $m$  را  $\omega$  را  $m \geq \omega$  را  $\omega$  را

عادرانی به این رشته کز اهریم

۳- حرف بی بگوئی <sup>۱۵</sup>  $\omega = xyz$  ،  $z$  هم نشانه

۱.  $|x| \leq m$  و  $|y| \leq n$  فرض این است که  $m$  و  $n$  به نحوی عمل گشتن را انجام

۴- ما سعی می‌کنیم عدد  $\omega_i$  را به قدری افزایش دهیم که  $\omega_i = \omega_j$  عضو  $\omega$  نباشد. اگر نتوانستیم بازی را بدو ادامه دهیم.

**P4PCO**

Subject : \_\_\_\_\_  
 Year .      Month .      Date .      (

طبيعہ

فرض کنیم زمان نامتناهی است یعنی متناهی نیست.

لم Pumping فرض کنید که زبان نامتناهی باشد، و عدد صحیح مثبت

$m$  و  $n$  را در  $W$  قرار دهیم.  $m$  و  $n$  را در  $W$  قرار دهیم.  $m$  و  $n$  را در  $W$  قرار دهیم.

رشته ها بصورت زیر به زیر رشته  $x, y, z$  شکافتند:  
 ۱. بتوانیم به زیر رشته  $y$  بگنیم

$$\omega = xyz$$

$$|xy| \leq m$$

1917

بدونای نه رسته  $z = x + jy$   $\omega = 1$   $\omega = 0, 1, \dots$   $\omega = 0$   $\omega = 1$   $\omega = 2$   $\omega = 3$   $\omega = 4$   $\omega = 5$   $\omega = 6$   $\omega = 7$   $\omega = 8$   $\omega = 9$   $\omega = 10$   $\omega = 11$   $\omega = 12$   $\omega = 13$   $\omega = 14$   $\omega = 15$   $\omega = 16$   $\omega = 17$   $\omega = 18$   $\omega = 19$   $\omega = 20$   $\omega = 21$   $\omega = 22$   $\omega = 23$   $\omega = 24$   $\omega = 25$   $\omega = 26$   $\omega = 27$   $\omega = 28$   $\omega = 29$   $\omega = 30$   $\omega = 31$   $\omega = 32$   $\omega = 33$   $\omega = 34$   $\omega = 35$   $\omega = 36$   $\omega = 37$   $\omega = 38$   $\omega = 39$   $\omega = 40$   $\omega = 41$   $\omega = 42$   $\omega = 43$   $\omega = 44$   $\omega = 45$   $\omega = 46$   $\omega = 47$   $\omega = 48$   $\omega = 49$   $\omega = 50$   $\omega = 51$   $\omega = 52$   $\omega = 53$   $\omega = 54$   $\omega = 55$   $\omega = 56$   $\omega = 57$   $\omega = 58$   $\omega = 59$   $\omega = 60$   $\omega = 61$   $\omega = 62$   $\omega = 63$   $\omega = 64$   $\omega = 65$   $\omega = 66$   $\omega = 67$   $\omega = 68$   $\omega = 69$   $\omega = 70$   $\omega = 71$   $\omega = 72$   $\omega = 73$   $\omega = 74$   $\omega = 75$   $\omega = 76$   $\omega = 77$   $\omega = 78$   $\omega = 79$   $\omega = 80$   $\omega = 81$   $\omega = 82$   $\omega = 83$   $\omega = 84$   $\omega = 85$   $\omega = 86$   $\omega = 87$   $\omega = 88$   $\omega = 89$   $\omega = 90$   $\omega = 91$   $\omega = 92$   $\omega = 93$   $\omega = 94$   $\omega = 95$   $\omega = 96$   $\omega = 97$   $\omega = 98$   $\omega = 99$   $\omega = 100$   $\omega = 101$   $\omega = 102$   $\omega = 103$   $\omega = 104$   $\omega = 105$   $\omega = 106$   $\omega = 107$   $\omega = 108$   $\omega = 109$   $\omega = 110$   $\omega = 111$   $\omega = 112$   $\omega = 113$   $\omega = 114$   $\omega = 115$   $\omega = 116$   $\omega = 117$   $\omega = 118$   $\omega = 119$   $\omega = 120$   $\omega = 121$   $\omega = 122$   $\omega = 123$   $\omega = 124$   $\omega = 125$   $\omega = 126$   $\omega = 127$   $\omega = 128$   $\omega = 129$   $\omega = 130$   $\omega = 131$   $\omega = 132$   $\omega = 133$   $\omega = 134$   $\omega = 135$   $\omega = 136$   $\omega = 137$   $\omega = 138$   $\omega = 139$   $\omega = 140$   $\omega = 141$   $\omega = 142$   $\omega = 143$   $\omega = 144$   $\omega = 145$   $\omega = 146$   $\omega = 147$   $\omega = 148$   $\omega = 149$   $\omega = 150$   $\omega = 151$   $\omega = 152$   $\omega = 153$   $\omega = 154$   $\omega = 155$   $\omega = 156$   $\omega = 157$   $\omega = 158$   $\omega = 159$   $\omega = 160$   $\omega = 161$   $\omega = 162$   $\omega = 163$   $\omega = 164$   $\omega = 165$   $\omega = 166$   $\omega = 167$   $\omega = 168$   $\omega = 169$   $\omega = 170$   $\omega = 171$   $\omega = 172$   $\omega = 173$   $\omega = 174$   $\omega = 175$   $\omega = 176$   $\omega = 177$   $\omega = 178$   $\omega = 179$   $\omega = 180$   $\omega = 181$   $\omega = 182$   $\omega = 183$   $\omega = 184$   $\omega = 185$   $\omega = 186$   $\omega = 187$   $\omega = 188$   $\omega = 189$   $\omega = 190$   $\omega = 191$   $\omega = 192$   $\omega = 193$   $\omega = 194$   $\omega = 195$   $\omega = 196$   $\omega = 197$   $\omega = 198$   $\omega = 199$   $\omega = 200$   $\omega = 201$   $\omega = 202$   $\omega = 203$   $\omega = 204$   $\omega = 205$   $\omega = 206$   $\omega = 207$   $\omega = 208$   $\omega = 209$   $\omega = 210$   $\omega = 211$   $\omega = 212$   $\omega = 213$   $\omega = 214$   $\omega = 215$   $\omega = 216$   $\omega = 217$   $\omega = 218$   $\omega = 219$   $\omega = 220$   $\omega = 221$   $\omega = 222$   $\omega = 223$   $\omega = 224$   $\omega = 225$   $\omega = 226$   $\omega = 227$   $\omega = 228$   $\omega = 229$   $\omega = 230$   $\omega = 231$   $\omega = 232$   $\omega = 233$   $\omega = 234$   $\omega = 235$   $\omega = 236$   $\omega = 237$   $\omega = 238$   $\omega = 239$   $\omega = 240$   $\omega = 241$   $\omega = 242$   $\omega = 243$   $\omega = 244$   $\omega = 245$   $\omega = 246$   $\omega = 247$   $\omega = 248$   $\omega = 249$   $\omega = 250$   $\omega = 251$   $\omega = 252$   $\omega = 253$   $\omega = 254$   $\omega = 255$   $\omega = 256$   $\omega = 257$   $\omega = 258$   $\omega = 259$   $\omega = 260$   $\omega = 261$   $\omega = 262$   $\omega = 263$   $\omega = 264$   $\omega = 265$   $\omega = 266$   $\omega = 267$   $\omega = 268$   $\omega = 269$   $\omega = 270$   $\omega = 271$   $\omega = 272$   $\omega = 273$   $\omega = 274$   $\omega = 275$   $\omega = 276$   $\omega = 277$   $\omega = 278$   $\omega = 279$   $\omega = 280$   $\omega = 281$   $\omega = 282$   $\omega = 283$   $\omega = 284$   $\omega = 285$   $\omega = 286$   $\omega = 287$   $\omega = 288$   $\omega = 289$   $\omega = 290$   $\omega = 291$   $\omega = 292$   $\omega = 293$   $\omega = 294$   $\omega = 295$   $\omega = 296$   $\omega = 297$   $\omega = 298$   $\omega = 299$   $\omega = 300$   $\omega = 301$   $\omega = 302$   $\omega = 303$   $\omega = 304$   $\omega = 305$   $\omega = 306$   $\omega = 307$   $\omega = 308$   $\omega = 309$   $\omega = 310$   $\omega = 311$   $\omega = 312$   $\omega = 313$   $\omega = 314$   $\omega = 315$   $\omega = 316$   $\omega = 317$   $\omega = 318$   $\omega = 319$   $\omega = 320$   $\omega = 321$   $\omega = 322$   $\omega = 323$   $\omega = 324$   $\omega = 325$   $\omega = 326$   $\omega = 327$   $\omega = 328$   $\omega = 329$   $\omega = 330$   $\omega = 331$   $\omega = 332$   $\omega = 333$   $\omega = 334$   $\omega = 335$   $\omega = 336$   $\omega = 337$   $\omega = 338$   $\omega = 339$   $\omega = 340$   $\omega = 341$   $\omega = 342$   $\omega = 343$   $\omega = 344$   $\omega = 345$   $\omega = 346$

مهم: پمپ Pumping فقط می‌توان آب را بردارد در زمان خاموشی است یعنی آب را می‌توان از پمپ بردارد در زمان خاموشی است

پمپ آب به کم Pumping  $\Rightarrow$  مایع را از  $L$  دارای سطح  $m$  به سطح  $m + \Delta h$  می‌برد  
و در آنجا می‌ریزد.

اسم و مشعر ← این  $\frac{d}{dt}$  محقق است ← و فراین  $\frac{d}{dt}$  محقق است.

$$L: dfa(m) \left\{ \begin{array}{l} \xRightarrow{p} \\ \xRightarrow{q} \\ \xRightarrow{r} \\ \xRightarrow{s} \end{array} \right. \xRightarrow{w} \underbrace{q_1 \dots q_r \dots q_r \dots q_p}_{\substack{x \quad y \quad z}}$$

9. در این حقیقه کدام مورد  $\frac{p}{q}$  حقیقه است و توان از  $\frac{p}{q}$  بیرون بیرون

$$\Rightarrow x y^2 \in L_{\bar{u}_1} \Rightarrow \bar{u}_1 \bar{x} \bar{y} \bar{y} \bar{u}_1$$

**PAPCO**

امانت حسن نقیض لم Pumping  
 حرفی می گوید من برای این حتمی نه زبان منظم است و DFA دارد

State - m

و من می گویم که برای m اثبات کنید تا من حلقه آن DFA  
 را پیدا کنم. و رسته را اثبات کردیم بعد حرف شروع می کنیم پس  
 رسته تا ی رسیدن به اما نه نقدن ی رسیدن کند ما رسیدن  
 می دهیم که خود آن DFA سبب بر زبان ما DFA ندارد پس  
 منظم نیست و حرف نیست من خود

Subject: Year: Month: Date: ( )

$$L = \{ ww^R \mid w \in \{a, b\}^* \} \quad \text{مسئله ۲}$$

$$\begin{array}{c} b \\ a^m b^m a^m \end{array} \quad \begin{array}{c} \text{حرف} \\ m \end{array} \Rightarrow \begin{array}{l} x = a^{k_1} \\ y = a^{k_2} \\ z = a^{m-k_1-k_2} b^{2m} a^m \end{array} \quad \begin{array}{l} k_1 + k_2 \leq m \\ k_2 \geq 1 \end{array}$$

$$w_i = a^{k_1} (a^{k_2})^i a^{m-k_1-k_2} b^{2m} a^m$$

$$i=0 \rightarrow a^{m-k_2} b^{2m} a^m \notin L \quad \text{پس } L \text{ نامنظم است}$$

$$L = \{ a^n b^p \mid n \geq p \} \quad \text{مسئله ۳}$$

$$\begin{array}{c} b \\ a^m b^m \end{array} \quad \begin{array}{c} \text{حرف} \\ m \end{array}$$

$$\begin{array}{l} x = a^{k_1} \\ y = a^{k_2} \\ z = a^{m-k_1-k_2} b^m \end{array} \quad w_i = a^{k_1} (a^{k_2})^i a^{m-k_1-k_2} b^m$$

$$i=0 \rightarrow a^{k_1} a^{m-k_1-k_2} b^m$$

$$a^{m-k_2} b^m \quad \text{درست} \quad m-k_2 \geq m \quad \text{است و درست نیست} \quad n \geq p$$

خوبه نه Final می رسم. حالا می بینیم تا m حلقه داریم.  
 و حداقل حلقه طریقی می بینیم اینجا.

نشان بده که Pumping Lemma (پمپینگ لیم) را نقض می‌کند.

نشان بده که Pumping Lemma (پمپینگ لیم) را نقض می‌کند.

فرض کن  $L$  یک زبان منظم است. چون  $L$  منظم است dfa دارد چون

نشان بده که  $L$  یک زبان منظم است. dfa با حالت  $m$  دارد در هر  $i$  state.

فرض کن  $L$  یک زبان منظم است. dfa با حالت  $m$  دارد در هر  $i$  state.  $L$  یک زبان منظم است. dfa با حالت  $m$  دارد در هر  $i$  state.

نشان بده که  $L$  یک زبان منظم است. dfa با حالت  $m$  دارد در هر  $i$  state.  $L$  یک زبان منظم است. dfa با حالت  $m$  دارد در هر  $i$  state.

نشان بده که  $L$  یک زبان منظم است. dfa با حالت  $m$  دارد در هر  $i$  state.  $L$  یک زبان منظم است. dfa با حالت  $m$  دارد در هر  $i$  state.

نشان بده که  $L$  یک زبان منظم است. dfa با حالت  $m$  دارد در هر  $i$  state.  $L$  یک زبان منظم است. dfa با حالت  $m$  دارد در هر  $i$  state.

نشان بده که  $L$  یک زبان منظم است. dfa با حالت  $m$  دارد در هر  $i$  state.  $L$  یک زبان منظم است. dfa با حالت  $m$  دارد در هر  $i$  state.

نشان بده که  $L$  یک زبان منظم است. dfa با حالت  $m$  دارد در هر  $i$  state.  $L$  یک زبان منظم است. dfa با حالت  $m$  دارد در هر  $i$  state.

نشان بده که  $L$  یک زبان منظم است. dfa با حالت  $m$  دارد در هر  $i$  state.  $L$  یک زبان منظم است. dfa با حالت  $m$  دارد در هر  $i$  state.

$$L = \{ ww^R \mid w \in \{a,b\}^* \}$$

$$\begin{aligned} x &= a^{k_1} \\ y &= a^{k_2} \\ z &= a^{m-k_1-k_2} b^{2m} a^m \end{aligned} \Rightarrow \begin{aligned} x &= a^{k_1} \\ y &= a^{k_2} \\ z &= a^{m-k_1-k_2} b^{2m} a^m \end{aligned}$$

$$w_i = a^{k_1} (a^{k_2})^i a^{m-k_1-k_2} b^{2m} a^m$$

$$i=0 \rightarrow a^{m-k_2} b^{2m} a^m \notin L$$

$$L = \{ a^n b^p \mid n \geq p \}$$

$$\begin{aligned} x &= a^{k_1} \\ y &= a^{k_2} \\ z &= a^{m-k_1-k_2} b^m \end{aligned} \Rightarrow w_i = a^{k_1} (a^{k_2})^i a^{m-k_1-k_2} b^m$$

$$i=0 \rightarrow a^{k_1} a^{m-k_1-k_2} b^m$$

$$i=1 \rightarrow a^{k_1+k_2} a^{m-k_1-k_2} b^m = a^m b^m$$



آنچه در نیمه اول منظم است.

$$L' = L - \{a^1\}$$

چون فرض کردیم  $L$  منظم و  $\{a^1\}$  منظم است پس  $L - \{a^1\}$  نیز

منظم است. در صورتیکه ثابت کردیم  $L'$  نامنظم است پس  $L$  باید نامنظم

باشد. الله فوق هیچ باب.

نکته دوم: وقتی دیدیم که تعداد محدود  $m$  حالت ایجاد می کند یعنی  $dfa$  نمی توان برابر  $P$

بی منظم است؛ زیرا برای  $dfa$  زدن نگاه یکی زبان  $\infty$

$dfa$  می توانیم داشته باشیم و این برای  $m=3$  به بعد تناقض دارد.

پس چون در حالت که  $dfa$  نمی توان زد پس  $L'$  نامنظم است.

هزاران منظم تعداد محدودی  $Dfa$  دارد.

$$L = \{a^n \mid n \geq 1\}$$

$$\frac{b}{a^m!} \quad \frac{m}{m}$$

$$x = a^{k_1}$$

$$k_1 + k_2 \leq m$$

$$y = a^{k_2}$$

$$k_2 \geq 1$$

$$z = a^{m! - k_1 - k_2}$$

$$i=0 \rightarrow a^{m! - k_2}$$

برای ثابت کنیم  $m! - k_2$  کی عدد فاکتوریل است

برای ثابت کنیم بین دو عدد متوالی فاکتوریل است

چون  $m=3$  به بعد هیچ حالتی نیست پس باید  $P$  و  $P$  می تواند فاکتوریل باشد.

$$m=1 \rightarrow 1 < 1 - \frac{k_2}{0!}$$

$$m=1 \rightarrow 1 < 1 - \frac{1}{0!} \quad \times$$

$$m=2 \rightarrow 1 < 2 - \frac{\log 2}{1!} \quad \times \quad \frac{k_2}{2} < \frac{1}{2}$$

$$m=3 \rightarrow 1 < 3 - \frac{\log 2 \text{ or } 3}{2!} \quad \checkmark \quad \frac{k_2}{1-2-3}$$

که از این به بعد هیچ است. و این از ۳ به بعد هیچ است.

برای  $m=1$  هم هیچ زیر حرف نمی تواند داشته ای به طرف ۱ را باید  $dfa$  ۱

State بی رسم کند صفا ۲ به State دارد.

$$L' = \{a^n \mid n \geq 2\}$$

حالا در فرض گفتیم  $L'$

$L = \{a^n b^m \mid n, m \geq 0\}$  <sup>منظم و برآیند  $a, b$  مستقل است</sup>

$L = \{a^n b^{2n} \mid n \geq 0\}$  <sup>منظم اما فقط زمانی که  $n$  زوج باشد  $2n$  را می توانیم</sup>

$L = \{a^n b^m \mid n > m \geq 0\}$  <sup>برای  $n$  و  $m$  هر دو  $\geq 0$  و  $n > m$</sup>

$L = \{a^n b^m \mid n, m \geq 50\}$  <sup>منظم</sup>

$L = \{a^n b^m \mid n = 2k, m = 5k', k, k' \geq 0\}$  <sup>منظم چون زوج از  $a$  و مضرب 5 از  $b$   $(a^2)^* (b^5)^*$</sup>

$L = \{a^n b^m \mid n = 2k, m = 5k, k \geq 0\}$  <sup>ک  $k$  و  $5k$  برابرند و منظم</sup>

$L = \{a^n b^m \mid n = 2^k, m = 5k', k, k' \geq 0\}$   <sup>$2^k$  محدودیت منظم</sup>

$L = \{a^n b^m \mid n = 2^m, m = 2^k, 0 \leq k \leq 16\}$   <sup>$2^k$  محدودیت منظم</sup>

$L = \{w w^R\}$  <sup>منظم نیست چون اگر  $a, b$  باشد  $a^p b^q$  و  $b^q a^p$  متفاوتند</sup>

$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24\}$  <sup>پمپینگ</sup>

(۱۵) ضرب حجم - اثبات این خواهد بود که

چون منظم است اما منظم بر آن زبان

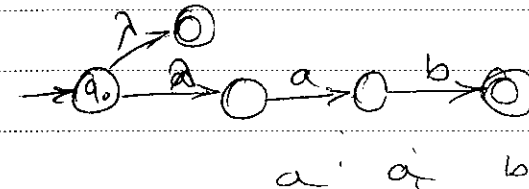
باشند زبان منظم که ماشین بدون حافظه است

نکات: ۱- هر زبان منظم است.

۲- در برای تشخیص تعلق یا عدم تعلق رشته  $a^n b^m$  به زبان نیاز به حافظه نامتناهی را نداریم آن زبان منظم نیست.

$L = \{a^n b^n \mid n \geq 0\}$

این  $n$  و  $n$  برابرند و حافظه نداریم و فقط این حالت



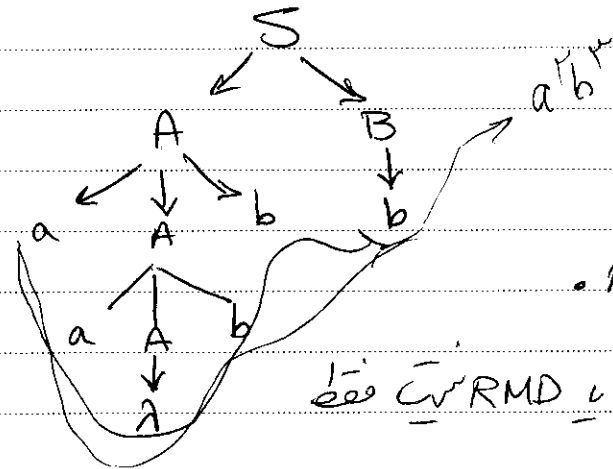
این نیز  $a^n b^n$  است چون به صورت  $a^n b^n$  و  $a^n b^n$  و این حالت State های فایده محدودیت دارند

در حافظه منتهی مورد نیاز است پس تعداد محدود State نداریم

پس منظم نیست چون State ها محدود است

# Derivation Tree / درخت استنتاج

آرگومان را نسبت به صورت درخت می‌توانیم در درخت استنتاج



درخت را می‌توان گفت  
 LMD است یا RMD

درخت استنتاج را می‌توانیم LMD یا RMD بنویسیم فقط  
 می‌توانیم از میانه می‌کنیم

↓ \* Top-Down Parsing (از Start به پایین می‌رویم به ترتیب)  
 → Bottom-Up Parsing (تجزیه از پایین به بالا می‌کنیم و از شروع شروع می‌کنیم به Stand به پایین)

5.1)  $a^2b^3$   
 $L = \{a^n b^m c^k \mid n=m \text{ or } m \neq k\}$   
 اگر مستقل از متن

$$S \rightarrow S_1 S_2$$

$$S_2 \rightarrow BD$$

$$S_1 \rightarrow AC$$

$$B \rightarrow aB \mid \lambda$$

$$A \rightarrow aAb \mid \lambda$$

$$D \rightarrow E \mid F$$

$$C \rightarrow cC \mid \lambda$$

$$E \rightarrow bEC \mid bE \mid b$$

$$F \rightarrow bFC \mid FC \mid c$$

زبان مستقل از متن  
 $A \rightarrow a$   
 $a \in (V \cup T)^*$

$$L = \{a^n b^m \mid n \neq m\}$$

$$S \rightarrow S_1 S_2$$

$$S_1 \rightarrow aS_1 \mid b \mid aS_1 \mid a$$

$$S_2 \rightarrow aS_2 b \mid S_2 b \mid b$$

$$L = \{a^n b^m \mid n < m\}$$

$$S \rightarrow AB$$

$$A \rightarrow aAb \mid \lambda$$

$$B \rightarrow bB \mid b$$

LMD = Left-Most Derivation

در درخت استنتاج همیشه Nonterminal چپ‌ترین شروع می‌شود LMD

$$w = a^2 b^3$$

$$S \rightarrow AB \rightarrow aAbB \rightarrow aaAbbB \rightarrow aabbbB \rightarrow aabbbb$$

RMD = Right-Most Derivation

در درخت استنتاج همیشه Nonterminal راست‌ترین شروع می‌شود RMD

$$S \rightarrow AB \rightarrow Ab \rightarrow aAbb \rightarrow aaAbbb \rightarrow aabbbb$$

تت ۱۸ سوال ۵۱ (محدود مسیور)

۴۱- اگر  $M = (Q, Q_0, \Sigma, F, \delta)$  اتومات منتهی باشد توپ می کنیم  $\bar{M} = (Q, Q_0, \Sigma, Q-F, \delta)$

همچنین  $d(M)$  اتومات قطعی معادل  $M$  خواهد بود. اگر  $M_1, M_2$  دو اتومات منتهی

باشند  $M_1 + M_2$  اتومات منتهی است که زبان آن اجتماع زبان  $M_1, M_2$  است

فرض کنید  $A$  و  $B$  دو زبان منظم باشند که زبان  $A \cup B$  به ترتیب معادل زبان

$M_1$  و  $M_2$  هستند که عبارت زیر صحیح است؟

حل: فرض ۲ است.

$$L(G_1) - L(G_2) = L(d(d(M_1) + M_2)) \quad \checkmark$$

$$L(G_1) - L(G_2) = L(\overline{M_1 + M_2})$$

گزینه ۲ صحیح است.  $M_1 + M_2$  منتهی نیست پس  $\overline{M_1 + M_2}$  منتهی نیست.

Com. فرض ۱ را میزنیم و میبینیم که گزینه اول هم درست است.

Com

5.1  $\rightarrow 8 \rightarrow e$

$$L = \{a^n b^m c^k \mid k = |n-m|\}$$

$$\begin{cases} k = n-m \Rightarrow k+m=n \Rightarrow \{a^k a^m b^m c^k\} S_1 \\ k = m-n \Rightarrow k+m=m \Rightarrow \{a^n b^n b^k c^k\} S_2 \end{cases}$$

که مقدار  $a, b, c$  مساوی و مقدار  $a, c$  مساوی و  $b$  مساوی  
 $S \rightarrow S_1 / S_2$   
 تولید  $a, c$  برابر  
 $S_1 \rightarrow a S_1 c \mid A$   
 $A \rightarrow a A b \mid \lambda$  تولید  $a, b$  برابر  
 $S_2 \rightarrow B C$   
 $B \rightarrow a B b \mid \lambda$  تولید  $a, b$  برابر  
 $C \rightarrow b C c \mid \lambda$  تولید  $b, c$  برابر

سوال ۵۱- ۵.1  $(7(b, c, d) - 8(d, f, g) - 9 - 12 - 13)$

$$18 - 21 - 23 - 24 - 27$$

سوال ۵۲- 5.2  $(3 - 4 - 5 - 6 - 9 - 10 - 11 - 13 - 19)$

(صفحه خالی جهت یادداشت)

۲۳ کدام صبیح است؟

آوردن یک شیء به بیرون از یک مجموعه

۱- در هر زبان منظم، رابطه ای مثل  $Z$  وجود دارد به قسمی که  $Z = uvw$  و  $v \neq \epsilon$

برای هر مقدار صحیح  $n$  رابطه  $Z^n = uv^n w^n$  نیز متعلق به زبان ما است.

غ - متده - نامیده می شود

۲- برای هر زبان منظم، عدد  $n$  صبیح مثل  $k$  وجود دارد به قسمی که اگر رابطه ای از ما مثل

حرکت طول بزرگتر از  $k$  داشته باشیم آن وقت رابطه ای از ما طول کوچکتر از  $k$  نیز خواهیم داشت.

۳- اگر رابطه ای از زبان ما مثل  $Z = uvw$  وجود داشته باشد به قسمی که برای هر  $n \geq 0$  رابطه  $Z^n = uv^n w^n$  متعلق به ما باشد آن وقت ما زبان منظم است.

غ چون  $v \neq \epsilon$  را به یاد داشته باشید.

۴-  $L$  زبان است؛ الفبای  $\{a, b\}$  به قسمی که طبق رابطه خاص ما دارای حداقل

یک زیررشته اوف قد برر شده هستند کوچکترین آتاقاتی که این زبان را

تلقه قطعی یا غیر قطعی

سند سازی کند دارد خیر و صفت (احتمالاً) است؟

پروفا  $dfa$  داریم

در  $dfa$  منوال  $a$  به  $Trap$  که  $a$  به  $Trap$  +  $a$  می شود

این سبک



ادامه جلسه ششم فصل پنجم

تجربش 5-2 خصوصیت در زبان مستقل از متن

فرض ①  
 قضیه فریون کسب می کند برای مستقل از متن باشد که فایده قولی بدو فرم  $A \rightarrow A$  (A-prob)  $A \rightarrow B$  (B-prob)  
 Exhaustive Search Parsing (Unit Production) باشد که نگاه دیگر

قادر خواهد بود تعیین کند که آیا رشته  $w \in \Sigma^*$  توسط گرامر می تولید هست یا نه

(این روش موازی حالت را مختلف را بسطید همه حالتها را می کشید)

مر ۱۲۹ توضیح در همه order زبان محدود است

وقتی دو شرط فرض ایجاب دارد هم چگاه طول رشته کاملی است یا نه می تواند به دست

مدتها طول اوراق می آید از start به هر چندتا حاکم می توانی

درست موازی آن را کسب حال هر یک از آن ترسیل رسیدیم دوباره به ازاد

می کشید موازی کسب حال آن قرار بدو سیر از رشته باشد نگاه

هنگامی که ماند اگر آن را ترسیل بردن به سیر نگار است که می کشید

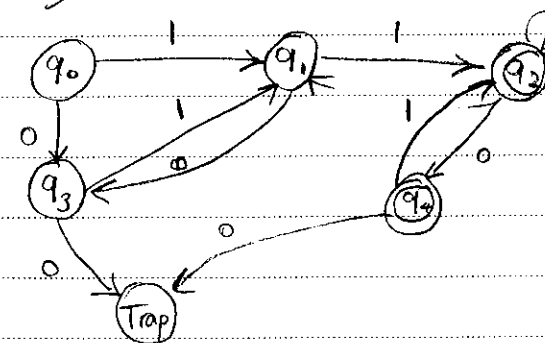
مهم تر از این (بلاش درخت)

$$L(G_1) - L(G_2) = L(\overline{d(d(M_1) + M_2)})$$

$$L_1 - L_2 = L_1 \cap \overline{L_2} = \overline{L_1} \cup L_2$$

قضیه ۱: به توصیف می کنیم  $M = (Q, Q_0, \Sigma, Q_F, \delta)$  همچنین  $d(M)$  اتوماتون قطعی  
 معادل  $M$  خواهد بود. اگر  $M_1$  و  $M_2$  دو اتوماتون قطعی باشند  $M_1 + M_2$  اتوماتون قطعی  
 است که زبان آن اجتماع زبانهای  $M_1$  و  $M_2$  است فرض کنید  $M_1$  و  $M_2$  دو گرامر مستقل  
 باشند که زبان آنها به ترتیب معادل زبانهای  $M_1$  و  $M_2$  هستند که این عبارت زیر صحت است!

$L$  زبان است با الفبای  $\Sigma = \{a, b\}$  متنی که طبع شده ای سار لای صاف یک زیر رشته او  
 فاقد زیر رشته  $00$  هستند که می تواند این آما متنی که این زبان را می کشد این



State 5  
 Final 2  
 Trap 1

در  $d(M)$  Trap را می کشد که در  $M$  Trap را می کشد

ادامه حبس فعل بنیم

کلاس 2-5 خصوصیت در زبان مستقل از متن

فرض ①  
 قضیه فرقی کنید یک گرامر مستقل از متن با یک فاکتور تولیدی بدون (A → A) (A-Produced)

نویسنده ②  
 و (Unit Production) A → B با یک نگاه دیگر Exhaustive Search Parsing

قادر خواهد بود تعیین کند که آیا رشته  $w \in \Sigma^*$  توسط گرامر می تولید هست یا نه

(این روش موازی حالتی را میسازد همه حالتها را می کند)

مر 119 توضیح دهید order زمان حقیقت است

وقتی دو شرط فرض باشد از این هم می تواند استفاده کرد

مدیریت اصل اوراق می آید از start به هر چند تا حاکم می توانی بری

درخت موازی آن را میسازد هر حالتی که از ترنیل رسیدیم دوباره به ازاد می

می بیند موازی می کند حال اگر قرار بود سیر از رشته باشد

هنگامی که می تواند ترنیل بردارد به یک باره می تواند

حالت موازی (بلافاصله)

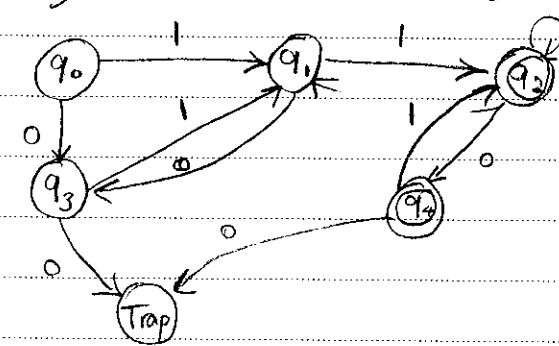
$$L(G_1) \cup L(G_2) = L(\sqrt{d(M_1) + M_2})$$

$$L_1 \cap L_2 = L_1 \bar{L}_2 = \bar{L}_1 L_2$$

نمودار  $M = (Q, Q_0, \Sigma, F, \delta)$  یک اتومات

قضیه: اگر  $M_1$  و  $M_2$  دو اتومات باشند به سبب  $M_1 + M_2$  اتومات تمام  
 است که زبان آن اجتماع زبانهای  $M_1$  و  $M_2$  است فرض کنید  $M_1$  و  $M_2$  دو گرامر مستقل  
 باشند که زبان آن به ترتیب معادل زبانهای  $M_1$  و  $M_2$  هستند که این عبارت زیر صحت است!

زبان است  $\Sigma = \{a, b\}$  متنی که طوری شده که برای هر یک زیر رشته او  
 فاکتور زیر رشته ه ه هستند که میگویند آن متنی که این زبان را میسازد می کنند برای



State 5  
 Final 2  
 Trap 1

در DFA Trap را میسازد  
 در nfa Trap را میسازد

ایم در گرامر مستقل از متن

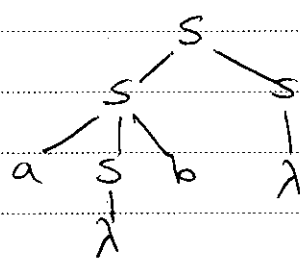
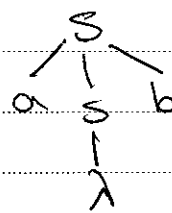
توقیف یک گرامر مهم به یک گرامر مستقل از متن مهم گفته می شود اگر زبان آن گرامر

مداخل یک رشته وجود داشته باشد برای آن حداقل دو LMD یا حداقل ۲ RMD

یا حداقل دو درخت اشتقاق داشته باشیم

$$S \rightarrow aSb | SS | \lambda$$

مثال



برای بسته ab ضد درخت داریم به هم نیست

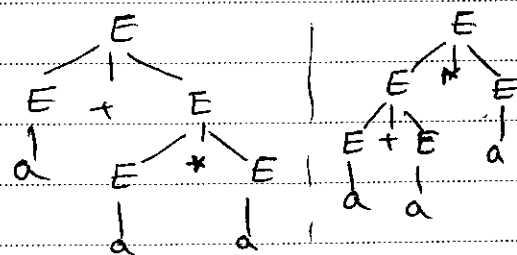
$$E \rightarrow E + E$$

$$E \rightarrow E * E$$

$$E \rightarrow (E)$$

$$E \rightarrow a$$

$$a + a * a$$



مثال

رشته

قدیم الگوریتمی وجود دارد که می تواند هر رشته  $w \in L(G)$  را به یک گرامر مستقل از متن  $G$

را در تعداد مراحل متناسب با  $|w|^3$  تجزیه کند. در  $O(n^3)$  می توان مشخص کرد رشته  $w$

گرامر زبان نیست.  $O(n^3)$  (بدون نرم افزار چسبک -)

تعریف یک گرامر ساده یا Simple Grammar: S-Grammar

گرامر مستقل از متن  $G$  S-Grammer گفته می شود اگر تمام قواعد آن به فرم زیر باشند

و  $A$  یک متغیر است

$$\begin{cases} A \rightarrow ax \\ a \in T \\ x \in V^* \end{cases}$$

در سمت راست هیچ دو  $A$  - قاعده ای با یکدیگر شروع شوند

$$S \rightarrow ax | ay$$

قاعده

این یک S-Grammar است

مثال

$$S \rightarrow aSb | ab$$

$$\textcircled{1} S \rightarrow aSB | aB$$

$$B \rightarrow b$$

$$\textcircled{2} S \rightarrow aX$$

$$X \rightarrow SB | B$$

$$B \rightarrow b$$

$$\textcircled{3} S \rightarrow aX$$

$$X \rightarrow aXB | b$$

$$B \rightarrow b$$

تاکثر از ۳ مرحله

محصول زبان ترسینال ندارند

گرامر مهم: گرامری که بتواند یک رشته را به روشی تولید کند می شود گرامر مهم.

Subject: \_\_\_\_\_  
Year: \_\_\_\_\_ Month: \_\_\_\_\_ Date: \_\_\_\_\_

"جلسه هفتم"

BNF = Backus - Naur Form

(این فرم)

یک گرامر مستقل از متن را می توان به دو  $\langle \text{Nonterm} \rangle$  و  $\langle \text{term} \rangle$  تقسیم کرد.

هر طبعی که بین  $\langle \rangle$  باشد معرف  $\text{Nonterm}$  است.

یا از  $::=$  استفاده می کنیم.

در این روش محدودیت نداریم برای  $\text{Nonterm}$  و پیرامون  $\text{N.T.}$  نخواهیم داریم.

معمولاً در برنامه نویسی از BNF استفاده می شود. در اینجا می بینیم که  $\langle \text{stat} \rangle$  و  $\langle \text{cond} \rangle$   $\text{Nonterm}$  هستند و  $\langle \text{term} \rangle$   $\text{term}$  است.

$\langle \text{stat} \rangle ::= \text{if} \langle \text{cond} \rangle \text{ then} \langle \text{stat} \rangle$

$\text{if} \langle \text{cond} \rangle \text{ then} \langle \text{stat} \rangle \text{ else} \langle \text{stat} \rangle$

$\langle \text{other kinds of stat} \rangle$

اگرچه در این روش گرامر زبان منطبق است. چون می توانیم به روشی دیگر  $\langle \text{stat} \rangle$  را تعریف کنیم.

$\text{if} \langle \text{cond} \rangle \text{ then}$   
 $\quad \text{if} \langle \text{cond} \rangle \text{ then}$   
 $\quad \quad \langle \text{stat} \rangle$   
 $\quad \text{else}$   
 $\quad \quad \langle \text{stat} \rangle$

PAPCO

Subject: \_\_\_\_\_  
Year: \_\_\_\_\_ Month: \_\_\_\_\_ Date: \_\_\_\_\_

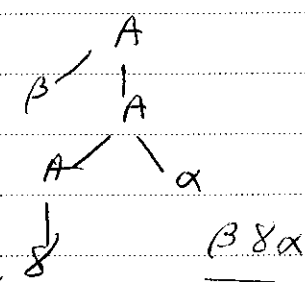
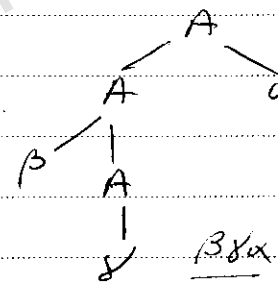
این روش می تواند یک گرامر مهم باشد یا نه؟ هیچ اهمیتی ندارد برای اینکه می توانیم.

گرامر مهم است یا نه؟ / این گرامر مهمی هم نیست و هم راستی آن تعریف می شود بر مبنای این.

$A \rightarrow A\alpha \mid B\alpha \mid \gamma$

مثال

$A \rightarrow A\alpha \mid \beta A \mid \gamma$



این گرامر مهم نیست  
مهم

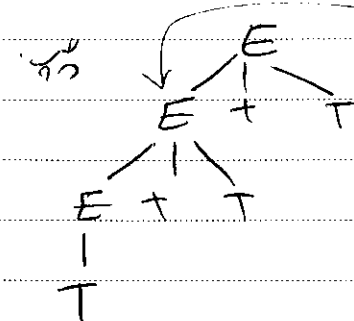
$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid a$

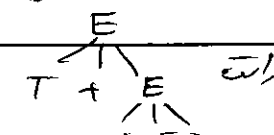
پرسش از شما: این گرامر می تواند یک گرامر مهم باشد یا نه؟  
(چون می توانیم به روشی دیگر  $\langle \text{stat} \rangle$  را تعریف کنیم)

این گرامر مهم نیست.



این گرامر مهم نیست. می توانیم به روشی دیگر  $\langle \text{stat} \rangle$  را تعریف کنیم.

$E \rightarrow T + E \mid T$



PAPCO

$\langle \text{stat} \rangle ::= \text{if } \langle \text{cond} \rangle \text{ then } \langle \text{stat} \rangle$

$\text{if } \langle \text{cond} \rangle \text{ then } \langle \text{stat}' \rangle \text{ else } \langle \text{stat} \rangle$

$\langle \text{other kinds of stat} \rangle$

$\langle \text{stat}' \rangle ::= \text{if } \langle \text{cond} \rangle \text{ then } \langle \text{stat}' \rangle \text{ else } \langle \text{stat}' \rangle$

$\langle \text{other kinds of stat} \rangle$

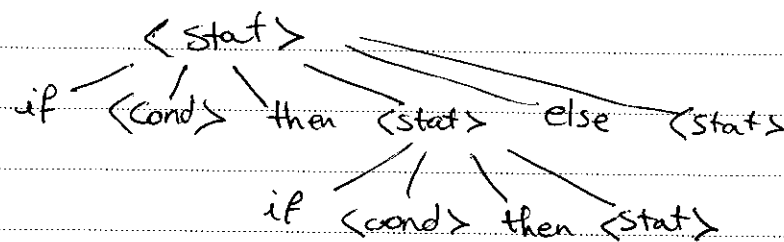
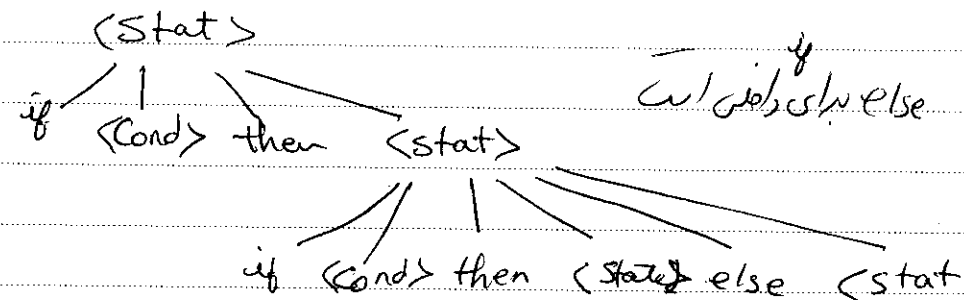
بی‌شک یک زبان ذاتاً مبهم (Inherently Ambiguous)

در یک زبان مستقل از متن، هر جمله‌ای که در آن هیچ گزاره‌ای وجود نداشته باشد، به یک زبان مستقل از متن گفته می‌شود. اما در این زبان، هر گزاره‌ای که در آن هیچ گزاره‌ای وجود نداشته باشد، به یک زبان مستقل از متن گفته می‌شود. این زبان، به یک زبان مستقل از متن گفته می‌شود.

حداکثر زبان مستقل از متن

else برای if دافتر است و در صورت هم‌انگاری اما در صورت می‌توان else را هم

برای بیرون کردن هم دافتر است.



else برای بیرون کردن است. (که او را می‌تواند else برای if دافتر است)

«رفع ابهام»  
 اگرچه else را برای if بیرون در نظر می‌گیریم، if دافتر می‌تواند

بیرون else را بیرون از if بیرون else را بیرون



$$3 \begin{bmatrix} 4 & 9 \\ 7 & i \end{bmatrix} - 6 \begin{bmatrix} 8 & 10 \\ 16 & 7 \end{bmatrix} - 1 \begin{bmatrix} 7 & 1 \\ 7 & 7 \end{bmatrix}$$

مفید (فصل ششم نظریه زبانها) (جزوه کلاس راهنمایی)

ساده سازی گرامرهای مستقل از متن و فرم های نرمال  
 مفید جابجایی

$$\begin{cases} A \rightarrow x_1 B x_2 \\ B \rightarrow y_1 \dots y_n \end{cases} \Rightarrow A \rightarrow x_1 y_1 x_2 \dots x_1 y_n x_2$$

این قوانین نسبت به رداستیم، برای B قوانین را گذاشته و به سمت راست می رویم.  
 به این معنی که تولید جابجایی  
 روابط ساده سازی

اولین محدودیت به سازی  
 حذف قوانین فایده  
 Removing Useless Productions

$$S \rightarrow a S b \mid \lambda \mid A$$

$$A \rightarrow a A$$

این قاعده به رشته نامتناهی می رسد  
 می کنیم

$$S \rightarrow a S b \mid \lambda$$

این قاعده به رشته نامتناهی می رسد  
 مفید (Useful) گفته می شود اگر وقتی این ضابطه را  
 تعریف می کنیم به معنی می آید A مفید

رشته w متعلق به L(G) وجود داشته باشد که برای آن داشته باشیم

$$S \Rightarrow^* x A y \Rightarrow^* w$$

نقشه از آن به ترتیب می رسم پس وقتی می رسم از این به بعد  
 PAPCO

$$L = \{ a^n b^n c^m \mid n, m \geq 0 \} \cup \{ a^n b^m c^n \mid n, m \geq 0 \}$$

این زبان را می توان مستقل از متن در بر داشت  
 $S \rightarrow S_1 \mid S_2$

$$S_1 \rightarrow AC$$

$$A \rightarrow aAb \mid \lambda$$

$$C \rightarrow cC \mid \lambda$$

$$S_2 \rightarrow BD$$

$$B \rightarrow aB \mid \lambda$$

$$D \rightarrow bDc \mid \lambda$$

این گرامر می تواند برای رشته ها  $a^n b^n c^n$  ( $n=m$ ) به روشی خاص  $S_1, S_2$  باشد

تولید می شوند. این زبان ذاتاً مهم است، برای رفع ابهام تنها این است که نتوانیم در یک

نقشه 3-3 نشان ندادیم از اینها  $S_1$  یا  $S_2$  نتوان رشته های از  $c^m a^n$

تولید می کنند.  $a, b, c$  به هم برابر باشند

اما برای این زبان ایده دیگری نمی توان داد این گرامر مهم است و

این زبان ذاتاً مهم است. هیچگاه نتوانیم این را رفع ابهام کنیم

مسئله 1)  $S \rightarrow aS, b$   
 $S_1 \rightarrow aS, b | \lambda$  فقط  $S_1$  قابل است  
 به سبب این  $S_1$  عضو  $\lambda$  است  
 حل در صورت باره 1-  $S_1$  به  $\lambda$

$S \rightarrow ab$

$S \rightarrow aS, b$   
 $S_1 \rightarrow aS, b$  2-  $S_1$  به  $\lambda$

حرف  $\lambda$   
 شده و اگر  $\lambda$  را از سیستم

$S \rightarrow aS, b | ab$   
 $S_1 \rightarrow aS, b | ab$  حرف  $\lambda$

مسئله 2)  $S \rightarrow ABac$   
 $A \rightarrow BC$   
 $B \rightarrow b | \lambda$   
 $C \rightarrow D | \lambda$   
 $D \rightarrow d$   
 عضو  $B, C$  به  $\lambda$  می شود  
 هم به واسطه  $B$  و  $C$  به  $\lambda$   
 به  $A, B, C$  عضو می شود

چون می خواهیم  $\lambda$  را حذف کنیم باید از این ابرجای بزاریم

$S \rightarrow ABac / BaC / AaC / ABA / aC / Ba / Aa / a$   
 $\downarrow$   
 $\lambda$  به  $A$  /  $\lambda$  به  $B$  /  $\lambda$  به  $C$  /  $\lambda$  به  $B, A$  /  $\lambda$  به  $SA$  /  $\lambda$  به  $B, C$  /  $\lambda$  به  $A$

P4PCO  $A \rightarrow BC / C / B$   $B \rightarrow b$   
 $\lambda$  به  $B$   $\lambda$  به  $C$   $C \rightarrow D$   $D \rightarrow d$

غیر پایانه  $A$  مفید است اگر در اول یک استق ناقص باشد  
 در غیر این صورت به پایانه  $A$  می خاند Useless لغت می شود

مسئله 3

1)  $S \rightarrow A$   
 $A \rightarrow aA | \lambda$   
 $B \rightarrow bA$  به  $\lambda$

2)  $S \rightarrow aS | A | C$   
 $A \rightarrow a$   
 $B \rightarrow aa$  حرف  
 $C \rightarrow aCb$  حرف  
 چون  $S$  می شود  
 چون از  $S$  می شود

حذف قواعد  $\lambda$  Removing  $\lambda$ -productions

$A \rightarrow \lambda$  :  $\lambda$ -production  
 $A \rightarrow^* \lambda$  : nullable (موجود)

به  $A \rightarrow \lambda$  می گویند قواعد  $\lambda$

و به آنجا که در یک استق به  $\lambda$  می خاند می گویند عضو

مرحله اول ساده سازی یک مرحله (تسک حذف ضعیف می باشد)

(۱) حذف قواعد  $A$  : چون  $A$  حذف می شود قاعده  $Unit$  (ایجاد می شود)

(۲) حذف قواعد  $Unit$  : چون حذف  $Unit$  معادل قاعده  $Useless$  می باشد.

(۳) حذف قواعد  $Useless$  :

$S \rightarrow AaB | aaB$  مثال

$A \rightarrow \lambda$  حذف  $A$  (عند حذف)  
 $B \rightarrow bbA | \lambda$   $A, B$   $B \rightarrow bbA | bb$

(۲) حذف  $Unit$  داریم

(۳) حذف  $Useless$   $S \rightarrow aB | a | aaB | aa$   
 $B \rightarrow bb$

هرچه  $A$  داشته باشد از بین می رود چون  $A$  وجود ندارد.

اینجا (انجام این ۳ مرحله) فقط باعث ساده سازی می شود

(و باعث نمی شود که گرامر را بهینه تر کند)

این گرامر ساده می باشد اما قواعد آن طولانی و زیاده است.

\* اگر گرامری که زبان آن فاقد رشته  $\lambda$  باشد، می توان بدون تغییر زبان قواعد

آن را حذف نمود.

مرحله دوم: حذف قواعد  $Unit$  (اولی می باشد) Removing Unit-Productions

این یک قاعده واحد می باشد.  $A \rightarrow B$   
(یک زبان رسمی که یک زبان رسمی می باشد قاعده واحد)

مثال  $S \rightarrow Aa | B$   
 $B \rightarrow A | bb$   
 $A \rightarrow a | bc | B$

$S \rightarrow Aa$   $S \rightarrow bb | a | bc$   
 $B \rightarrow bb$   $B \rightarrow a | bc$   
 $A \rightarrow a | bc$   $A \rightarrow bb$

نتیجه ۲ می باشد

$S \rightarrow Aa | bb | a | bc$   
 $B \rightarrow bb | a | bc$   
 $A \rightarrow a | bc | bb$

این گرامر را می توان بدون قواعد یک (حذف)

\* زبان متعلق از متن دارای فرم CNF دارد.

هذه زبان متعلق از متن که فاقد رشته  $\epsilon$  باشد دارای فرم چامسکی

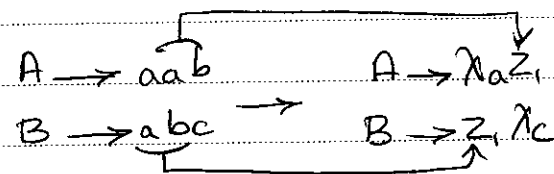
است. به آن زبان  $\epsilon$  را شده نام CNF ندارد.

مراحل تبدیل:

۱- حذف قواعد  $\epsilon$ .

۲- حذف قواعد Unit.

۳- دنبال کردن روند  $\epsilon$  به مثل قبل. و انجام کارهای مشابه دیگر.



\* تعداد مراحل لازم برای تولید رشته ای بطول  $n$  چند است؟

$$S \Rightarrow^{n-1} A_1 \dots A_n \Rightarrow^n a_1 \dots a_n$$

$$n-1 + n = 2n-1 \quad O(n)$$

در یک مرحله  $S$  را می کنیم به  $2$  نان ترمنال و در  $n-1$  مرحله می کنیم

آن را به  $n$  نان ترمنال و در  $n$  مرحله می کنیم به  $n$  ترمنال.

(چون مرحله اول به  $2$  ترمنال می رسد)

فرم های نرمال

۱- نرمال چامسکی CNF: Chomsky Normal Form

CNF:  $A \rightarrow BC$

$A \rightarrow a$

$A, B, C \in V$

$a \in T$

۱- قاعده  $\epsilon$  را اینجا وجود ندارد.

۲- سبقت راسته دو نان ترمنال

یا یک یک ترمنال

مثال  $S \rightarrow ABa$   $S \rightarrow AB\lambda a$

$A \rightarrow aab$   $\rightarrow \lambda a \rightarrow a$

$B \rightarrow Ac$   $A \rightarrow \lambda a \lambda a \lambda b$

$\lambda b \rightarrow b$

$B \rightarrow A\lambda c$

$\lambda c \rightarrow c$

مثال  $S \rightarrow AZ_1$   
 $Z_1 \rightarrow B\lambda a$

$\lambda a \rightarrow a$

$A \rightarrow \lambda a Z_2$

$Z_2 \rightarrow \lambda a \lambda b$

$\lambda b \rightarrow b$

$B \rightarrow A\lambda c$

$\lambda c \rightarrow c$

تبدیل به CNF

## \* مراحل تبدیل به فرم GNF

- ۱- حذف صفری
- ۲- حذف A
- ۳- حذف Unit
- ۴- جایگذاری

در برنامه‌های مستقل از متن ضمیمه می‌توان صفری را حذف نمود.

$$A \rightarrow A\alpha_1 | \dots | A\alpha_n$$

$$A \rightarrow \beta_1 | \dots | \beta_m$$

$$(\beta_1 + \dots + \beta_m)(\alpha_1 + \dots + \alpha_n)$$

$$A \rightarrow \beta_1 A' | \dots | \beta_m A'$$

$$A' \rightarrow \alpha_1 A' | \dots | \alpha_n A' | \lambda$$

\* مقدار مراحل مورد نیاز برای تولید رشته‌ای طول  $n$  چقدر است؟ GNF مورد

در  $n$  مرحله زیرا در هر مرحله ۱ تبدیل تولید می‌کنیم.

## فرم نرمال گریباخ GNF: Geribach Normal Form

یک برنامه مستقل از متن دارای فرم نرمال گریباخ است اگر:  

$$\begin{cases} A \rightarrow aX \\ a \in \tau \\ X \in V^* \end{cases}$$
  
 صفری است - یک تبدیل  
 ترسیمی

مثال) 
$$\begin{aligned} S &\rightarrow AB \\ A &\rightarrow aA | bB | b \\ B &\rightarrow b \end{aligned}$$

$$S \rightarrow aAB | bBB | bB \rightarrow \text{GNF}$$

مثال) 
$$S \rightarrow a b S b | a a$$

$$\begin{aligned} S &\rightarrow aX_b S X_b | aX_a \\ X_a &\rightarrow a \\ X_b &\rightarrow b \end{aligned}$$

آیا هر زبانی می‌تواند به فرم GNF داشته باشد؟

آره یک زبان مستقل از متن فاقد رشته  $\lambda$  است برای آن تراوی

GNF وجود دارد.

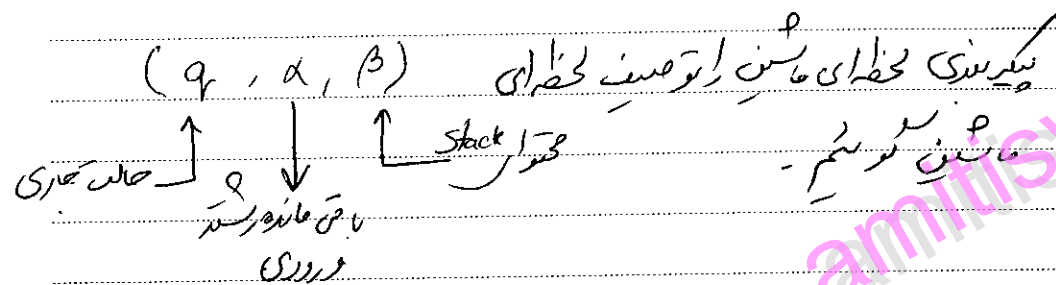


حرکت چهارم: ماشین توی حالت  $q_3$  و آبه است  $C$  و بدون تویم به ورودی

(بدون معرف ورودی) ماشین صفر و توی  $q_4$  و کای  $C$  که  $pop$  کرد

$C$  ۱، پش من کنه

### توصیف لحظه‌ای Instantaneous Description



با تویم به اصل اول حروفه (ماشین)

$(q_0, abb a, cbaz) \rightarrow (q_1, bba, xcba z)$

$\delta(q_0, a, C) = (q_1, xC)$   
 از ورودی بی معرف

$\rightarrow (q_1, bba, xcba z)$

چون متن قطعی را بستیم من تویمتیم کار بالایی، ماشین را بریم.

انواع حرکات قابل انجام توسط PDA

$$1) \delta(q_0, a, C) = \{(q_1, xC), (q'_1, xxC)\}$$

$$2) \delta(q_1, b, x) = \{(q_2, y)\}$$

$$3) \delta(q_1, b, y) = \{(q_3, \lambda)\}$$

$$4) \delta(q_3, \lambda, C) = \{(q_4, C)\}$$

حرکت اول: اگر ماشین تر State  $q$  است و head روی  $a$  است و Top Stack  $C$  است که نگاه

State، آبن  $q_1$  و  $xC$  را push کن.

حرکت دوم: اگر ماشین توی State  $q_1$  است و زیر head  $b$ ، آبه است

$x$  نگاه: State  $q_2$  من کنه  $y$  را push من کنه.

حرکت سیم: اگر ماشین توی State  $q_2$  است و زیر head  $b$  و آبه است

$y$  نگاه: ماشین من State  $q_3$  و آن که  $pop$  کردیم

همین push من کنه head من کنه

اوپر دیے پذیرش: پذیرش با Stack خالی ہے

$$L(M) = \{ w \in \Sigma^* \mid (q_0, w, z) \xrightarrow{*} (p, \lambda, \lambda), p \in Q \}$$

تقریب پذیرش توسط PDA:

۱. دوسری پذیرش صرفی می کنند

۲. پذیرش Final state با حالت نهایی

$$L(M) = \{ w \in \Sigma^* \mid (q_0, w, z) \xrightarrow{*} (q_p, \lambda, \lambda), q_p \in F, z \in \Gamma^* \}$$

در تمام رشته‌های از  $\Sigma^*$  پذیرفته می شوند و وقتی از حالت  $q$  و رشته ورودی

کل رشته و Stack خالی هستیم حال دیگر کاملاً Satisfy می کنند

برای در این حالت یک State فانی از ورودی می خواند یعنی

با این رشته هستیم و همچنین نوی Stack می خواند. یکی از کاملاً

در  $\Sigma^*$  (عامل تعیین کننده  $(q_p, \lambda, \lambda)$ )  
 پذیرفته می شود  
 در صورتی

⇐ زبان PDA پذیرفته می شود

رشته مورد قبول است