نظریه زبانها و ماشین ها

دکتر سید جواد حاج سید جوادی

جلسه‌ی اول

مرجع: کتاب لینز

زبان صوری Formal Language

زبان‌های طبیعی ← با آن‌ها گویش می‌کنیم — با دیگری دایره‌ی کلمات ← نامحدود

رسم‌الخط یک زبان ← تحولات زیادی به وجود آورد

صحبت در رابطه با نوع گویش

نقطه‌ی تحول بزرگ ← وقتی بود که گرامر زبان مشخص گردید

زبان‌های صوری

✓ ما به نیاز می‌خواهیم با سخت‌افزاری ارتباط برقرار کنیم

✓ همه چیز قانون‌مند است

✓ از همان ابتدا مشخص مجموعه‌ی حروف الفبایی شرح می‌شود

✓ در زبان‌های طبیعی مشخص کلمات براساس الگو نیست ولی زبان Formal از همان ابتدا براساس الگو است.

✓ روی سخت‌افزار باید اطلاعات در DB آن باشد پس دایره‌ی کلمات نباید نامحدود باشد.

زبان‌های Formal

۱- دیدگاه سخت‌افزاری

سخت‌افزار آن را بفهمد توسط سخت‌افزاری پذیرفته شد پس همان است.

مثلاً ماشین‌های DFA و NFA ← منظم

ماشین تورینگ ← بدون قید

۲- دیدگاه گرامری

ساختار جمله‌بندی‌ها مشخص است ← گرامر زبان فارسی / انگلیسی / ...

دسته‌بندی زبان‌ها از دیدگاه گرامری است.

NFA و DFA ← گرامرهای خطی چپ و راست، منظم، باقاعده

پشته‌ای است ← مستقل از متن

حساس به متن ← گرامرهای حساس به متن

✓ زبان‌های منظم دیدگاه دیگری هم دارد

└ دیدگاه عبارت‌های منظم

۱,

آتاماتا ← ماشین ها

هدف ← طبقه بندی زبان های formal

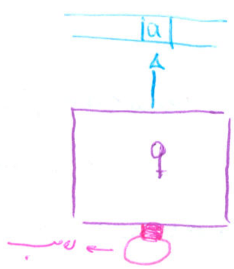
شروع درس:

ماشین قطعی متناهی DFA

✓ چون ماشین ساده ای است از هیچ حافظه ای استفاده نمی کند (حافظه ی کمکی ندارد)

✓ یک نوار ورودی هد خواندنی که به سمت راست حرکت می کند (کاراکتر به کاراکتر)

✓ وقتی به آخر رشته رسید لامپ یا خاموش می ماند یا روشن می شود



accept reject

مفاهیم اولیه

تعریف حروف الفبا

هر مجموعه ی متناهی و ناتهی از علائم خوش تعریف می تواند یک مجموعه ی حروف الفبا باشد.
معمولاً برای نمایش مجموعه ی حروف الفبا از نمادهای Σ ، Γ ، Δ و ... استفاده می شود.
سیگما ، گاما ، دلتا

خوش تعریف

برداشت های گوناگون یا متناقض از یک مفهوم نداشته باشیم.

$\Sigma = \{a, b\}$ ✓

$\Sigma = \mathbb{N} = \{0, 1, 2, \dots\}$ ✗ نامتناهی است

$\Sigma = \emptyset$ ✗ ناتهی باید باشد

$\Sigma = \{a, aa, b\}$ ✗ aa ← نمی توان فهمید aa یا دو a است

$\Sigma = \{bc, ad, ab\}$ ✓

تعریف رشته

گیریم $w = a_1 a_2 a_3 \cdots a_n$ رشته‌ای به طول n است، روی حروف الفبای Σ) هرگاه برای هر i،

$a_i \in \Sigma$ است.

طول رشته w را با $|w|$ یا $n(w)$ نمایش می‌دهیم.

رشته‌ای به طول صفر را رشته‌ی پوچ نامیده و با λ یا ε نمایش می‌دهیم.

رشته‌ای که کنار هر رشته‌ی دیگری بنشیند، همان رشته را نتیجه می‌دهد.

مثال: رشته abcab رشته‌ای به طول ۵ روی حروف الفبای $\Sigma=\{a,b,c\}$.

تعریف الحاق دو رشته (Concatenation)

هرگاه w_1 و w_2 دو رشته روی حروف الفبای Σ باشند، آن‌گاه منظور از $w_1 . w_2$ (می‌خوانیم w_1 concat w_2) رشته‌ای است که از چسباندن w_2 به انتهای رشته‌ی w_1 به دست می‌آید.

$w_1 = Ali$ $\quad w_1 w_2 = AliAhmadi$

$w_2 = Ahmadi$

تعریف Σ^n

هرگاه Σ مجموعه حروف الفبای باشد منظور از Σ^n مجموعه‌ی همه‌ی رشته‌ها به طول n روی حروف الفبای Σ است.

مثال: $\Sigma=\{a,b\}$ آن‌گاه

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

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

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

$\{a,b,c\}^3 = \{aaa, \ldots, ccc\}$

تعداد $3^3 = 27$

نکته: هرگاه Σ مجموعه حروف الفبای باشد:

$$|\Sigma^n| = |\Sigma|^n$$

تعریف Σ^* و Σ^+

هرگاه Σ مجموعه حروف الفبای باشد،

$$\Sigma^* : \bigcup_{n=0}^{\infty} \Sigma^n = \{\omega \mid \exists n \in \mathbb{N} \text{ s.t } \omega \in \Sigma^n\}$$

n را از ابتدا انتخاب کرده‌ام ∞ نیست.

$$\Sigma^+ : \bigcup_{n=1}^{\infty} \Sigma^n = \{\omega \mid \exists n \in \mathbb{N}^+ \text{ s.t } \omega \in \Sigma^n\}$$

توجه:

$$\Sigma^+ = \Sigma^* \setminus \{\lambda\}$$

$$\Sigma^* = \Sigma^+ \cup \{\lambda\}$$

$(\Sigma^*, \cdot)$

Semi group

۱- عمل الحاق روی Σ^* بسته است.

$$\forall \omega_1, \omega_2 \in \Sigma^* \quad \omega_1 \cdot \omega_2 \in \Sigma^*$$

✓ اگر مجموعه‌ای با عملی بسته باشد یک سیستم جبری است.

۲- شرکت پذیری

$$\forall \omega_1, \omega_2, \omega_3 \in \Sigma^* \quad \omega_1 \cdot (\omega_2 \cdot \omega_3) = (\omega_1 \cdot \omega_2) \cdot \omega_3$$

نیم گروه

✓ اگر بسته باشد و شرکت پذیر باشد به آن نیم گروه یا Semi Group گوییم.

۳- عضو خنثی

$$\forall \omega \in \Sigma^* \quad \omega \cdot \lambda = \lambda \cdot \omega = \omega$$

تکوار

✓ اگر هر سه خصوصیت را داشته باشد به آن تکوار، تک گون یا monoid گفته می‌شود.

۴- اگر هر عضوی وارون داشته باشد گروه می‌شود که در اینجا نداریم.

✓ تنها عضوی که عضو وارون دارد λ است.

تعریف طول رشته نسبت به حرف

هرگاه Σ مجموعه حروف الفبای باشد، ω یک رشته روی حروف الفبای Σ باشد. برای $a \in \Sigma$ طول رشته ω نسبت به حرف a را با $n_a(\omega)$ یا $|\omega|_a$ نمایش می‌دهیم عبارت است از تعداد a های ظاهر شده در ω.

مثال: اگر $\Sigma=\{a,b,c\}$ آن گاه برای $w = abacba$ داریم:

$n_a(w)=3 \quad n_b(w)=2 \quad n_c(w)=1$

نکته: هرگاه w_1 و w_2 متعلق به Σ^*) $w_1, w_2 \in \Sigma^*$ آن گاه:

i) $|w_1 . w_2| = |w_1|+|w_2|$

ii) $|w_1 . w_2|_a = |w_1|_a + |w_2|_a$ برای $a\in\Sigma$

iii) $\sum_{a\in\Sigma} |w|_a = |w|$ ← مجموع

که مجموع طول رشته‌ها نسبت به حرف = طول رشته
کل حروف الفبا

مثال: $|w|_a + |w|_b + |w|_c = |w|$

قرارداد: هرگاه $a\in\Sigma$ باشد، آن گاه:

$$a^n = \begin{cases} \lambda & n=0 \\ a.a^{n-1} & o.w \end{cases}$$

$$a^n = \underbrace{a\cdots a}_{n\text{تا}}$$

$a^n a^m = a^{n+m}$

$(a^n)^m = a^{nm}$

$a^n b^n = \underbrace{a\cdots a}_{n\text{تا}}\underbrace{b\cdots b}_{n\text{تا}}$

تعریف معکوس یک رشته

هرگاه w یک رشته روی Σ باشد، آن گاه معکوس رشته‌ی w را با w^R یا w^r نمایش می‌دهیم و به شکل زیر تعریف می‌کنیم.
حروف الفبای

$$w^r = \begin{cases} \lambda & w=\lambda \\ w'^r a & w=aw', a\in\Sigma \end{cases}$$

نتیجه: هرگاه $a\in\Sigma$ آن گاه

$a^r = a$

$a^r = (a\lambda)^r = \lambda^r a = a$

۳،

مثال: $(abb)^r = (bb)^r a = b^r ba = bba$

تعریف رشته‌ی خودمعکوس (پالیندروم)

رشته‌ی w متعلق به Σ^* را خودمعکوس می‌نامیم هرگاه

$$\boxed{w = w^r}$$

* تعریف زبان

گوییم L یک زبان روی حروف الفبای Σ است هرگاه $L \subseteq \Sigma^*$.

نتیجه: مجموعه‌ی تمام زبان‌ها روی حروف الفبای Σ را با 2^{Σ^*} نمایش می‌دهیم.

$$2^{\Sigma^*} = \{L \mid L \subseteq \Sigma^*\}$$

مثال: $2^{\{1,2\}} = \{\{\},\{1\},\{2\},\{1,2\}\}$

معکوس زبان

هرگاه L یک زبان روی حروف الفبای Σ باشد، آنگاه معکوس زبان L را L^r نمایش می‌دهیم و به شکل زیر تعریف می‌کنیم:

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

مثال: هرگاه $L = \{ab, abb, abc\}$ آنگاه:

$$L^r = \{ba, bba, cba\}$$

نتیجه: اگر L زبان باشد: $(L^r)^r = L$

الحاق دو زبان

هرگاه L_1 و L_2 دو زبان روی حروف الفبای Σ باشند:

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

مثال:

$L_2=\{a,b\}$, $L_1=\{ab,a,\lambda,a^2\}$

$L_1.L_2=\{aba, a^2, a, a^3, ab^2, ab, b, a^2b\}$

تعمیم: $\boxed{|L_1.L_2|\leq|L_1||L_2|}$

مثال: $L_1=\{a,\lambda\}$, $L_2=\{a,\lambda\}$, $L_1.L_2=\{a,\lambda,a^2\}$

نکته: هرگاه L_1 و L_2 دو زبان روی حروف الفبای Σ باشند آنگاه:

i) $(L_1.L_2)^r=L_2^r.L_1^r$, $(L_1^r)^r=L_1$

ii) $\forall\, w_1, w_2 \in L$ $(w_1.w_2)^r=w_2^r.w_1^r$, $(w_1^r)^r=w_1$

iii) $|L_1.L_2|\leq|L_1||L_2|$

تعریف L^n

هرگاه L یک زبان روی حروف الفبای Σ باشد:

$$L^n=\begin{cases}\{\lambda\} & n=0\\ L.L^{n-1} & n\geq 1\end{cases}$$

تعریف L^+ و L^*

$$L^*=\bigcup_{n=0}^{\infty}L^n \quad , \quad L^+=\bigcup_{n=1}^{\infty}L^n$$

نکته:

$$L^+=\begin{cases}L^* & \lambda\in L\\ L^*\setminus\{\lambda\} & \lambda\notin L\end{cases}$$

تست: هرگاه Σ مجموعه حروف الفبای باشد، تعداد زبان های L روی Σ که $|L^*|<\infty$ کدام است؟

۱) ۰ ۲) ۱ ۳) ۲ ✓ ۴) بی‌شمار

$L=\lambda \Rightarrow L^*=\{\lambda\}$

$L=\{\} \Rightarrow L^*=\{\lambda\}$

ϕ^*

دو زبان متناهی

۴۱

تعریف متمم زبان

متمم زبان L را با $\bar{L}$ یا L^c یا L' نمایش می‌دهیم. و برابر است با :

$$\bar{L} = \Sigma^* \setminus L$$

نکته: مجموعه‌ی مرجع به مفهوم عام وجود ندارد.

تست: تعداد زبان‌های L که $\overline{L^*} = \bar{L}^*$ صدق می‌کند چیست؟

۱) ۰ ✓ ۲) ۱ ۳) ۲ ۴) بی‌نهایت (بی‌شمار)

هر زبانی L^* است λ دارد. $\bar{L}^*$ ← λ را دارد

L^* ← λ را دارد ← $\overline{L^*}$ ← λ را ندارد

تست: $\varphi : \underset{\text{هر } L}{2^{\Sigma^*}} \to \underset{L_2}{2^{\Sigma^*}}$

$\varphi(L) = \bar{L}$ در این صورت:

الف) φ تابعی ۱-۱ و پوشاست. ✓

ب) φ ۱-۱ است ولی پوشا نیست.

ج) φ پوشاست ولی ۱-۱ نیست.

د) φ پوشا و ۱-۱ نیست.

1-1 ✓

$\varphi(L_1) = \varphi(L_2) \Rightarrow L_1 = L_2$

$\bar{L_1} = \bar{L_2} \Rightarrow \bar{\bar{L_1}} = \bar{\bar{L_2}} \Rightarrow L_1 = L_2$

پوشا بودن ✓

$\forall b \in B \quad \exists a \in A \quad f(a) = b$

$\forall L_2 \in 2^{\Sigma^*}$

$\varphi(\bar{L_2}) = \bar{\bar{L_2}} = L_2$

تست: $\varphi(L) = L^*$

د) نه پوشا، نه یک به یک

$\varphi(\{\emptyset\}) = \varphi(\lambda) \Rightarrow \{\emptyset\}^* = \{\lambda\}^*$

$\Rightarrow \emptyset \neq \lambda$

۱-۱ نیست ✗

$a \in \Sigma \quad \{a\} \in 2^{\Sigma^*} \quad \nexists L \ \varphi(L) = L^* = \{a\}$

تست: اگر $\Sigma=\{a,b,c\}$ و $L-\Sigma^*=\emptyset$ آنگاه L کدام یک از زبان‌های زیر می‌تواند باشد؟

i) Σ^* — 1) فقط I

ii) $a^n b^{n^2} c^n$ — 2) فقط iv

iii) $\emptyset$ — 3) فقط ii و iii

iv) ε — 4) همه ✓

نکته

$$A-B=\emptyset \Rightarrow A\subseteq B \Leftrightarrow A\cap B=A \Leftrightarrow A\cup B=B \Leftrightarrow \bar{B}\subseteq\bar{A}$$

$$\Leftrightarrow \bar{B}-\bar{A}=\emptyset$$

۱)

$$\overline{L}^R \neq \overline{L^R}$$

۲)

$$(L^*)^* = (L^*)^+ = (L^+)^* = L^*$$

۳)

$$\overline{\{a^n b^m \mid n\neq m\}}_{n,m\geq 0} = \Sigma^* - \text{خودش}$$

روی $\Sigma=\{a,b\}$

۴)

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

$$\Rightarrow$$

$$L^*=L$$

۵)

$$\{a^n b^n w \mid w\in\{a,b\}^*\}_{n\geq 0} = \{a,b\}^*$$

$$\Rightarrow \{a^n b^n w \mid n\geq 0,\ w\in\{a,b\}^*\} \subseteq \{a,b\}^*$$

$$x\in\{a,b\}^* \Rightarrow x=a^0 b^0 x \in \{a^n b^n w \mid n\geq 0,\ w\in\{a,b\}^*\}$$

$$\{a,b\}^* \subseteq \{a^n b^n w \mid w\in\{a,b\}^*\}$$

۶)

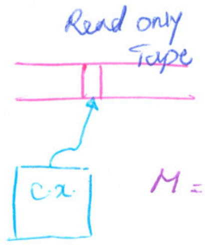
$$\underbrace{\{x^n y^n \mid n\geq 0,\ x,y\in\{a,b\}^*\}}_{L} = \{a,b\}^*$$

$$x\in\{a,b\}^* \Rightarrow x=x^1\lambda^1\in L$$

۵۱

* پذیرنده متناهی معین/قطعی DFA

Deterministic Finite Automaton/Acceptance



$M = \langle Q, \Sigma, \delta, q_0, F \rangle$ یک پذیرنده‌ی متناهی قطعی dfa یک پنج تایی مرتب است که در آن

الف: Q ← مجموعه‌ای متناهی و ناتهی از وضعیت‌های داخلی است

ب: Σ ← مجموعه‌ی حروف الفبایی است

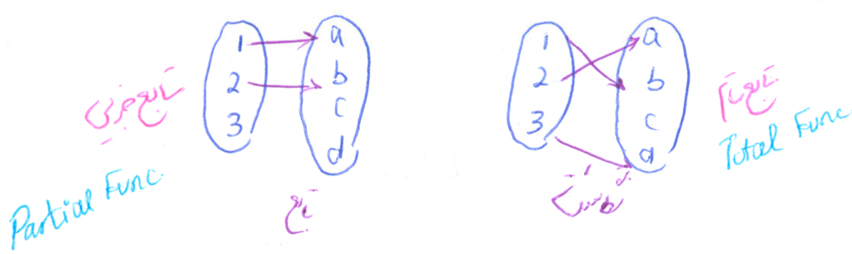
ج: δ ← تابع گذر یا انتقال است Transition Func

$\delta: Q \times \Sigma \to Q$ ← یک تابع تام است.

د: $F \subseteq Q$ ← مجموعه وضعیت های نهایی

هـ: $q_0 \in Q$ ← وضعیت شروع است

فرق نگاشت و تابع

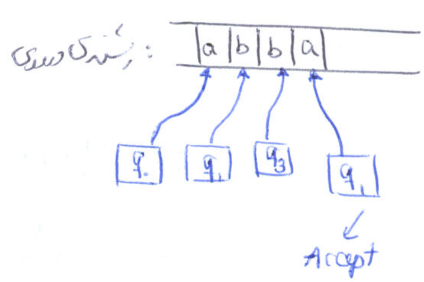


مثال: DFA ←

$M = \langle \{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \{q_1, q_2\} \rangle$

که تابع δ به شکل زیر تعریف شده است:

Q \ Σ	a	b
q_0	q_1	q_2
q_1	q_0	q_3
q_2	q_3	q_2
q_3	q_1	q_3



۷ می‌خواهم تابع δ را به تابع δ^* چنین توسعه دهیم که تابع δ^* با دریافت یک وضعیت و یک رشته، وضعیتی را که به آن خواهد رسید را مشخص نماید.

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

۱) $$\forall q \in Q \quad \delta^*(q, \lambda) = q$$

بدون هیچ move ی نباید تغییر کند.
اگر در وضعیتی باشد مادامی که رشته‌ها (حرفی) وارد نشده نباید به وضعیت دیگری برود

۲) $$\forall a \in \Sigma, w \in \Sigma^*, q \in Q \quad \delta^*(q, aw) = \delta^*(\delta(q,a), w)$$

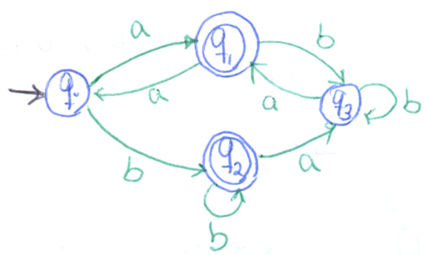
δ^* نسخه‌ی جدیدی از δ است. پس عملکرد تابع δ^* در دریافت یک حرف باید همان رفتار تابع δ باشد

$$\delta^*(q,a) = \delta^*(q,a,\lambda) = \delta^*(\overset{q}{\delta(q,a)}, \lambda) = \delta(q,a)$$

مطلب بالا نشان می‌دهد که δ^* یک ورژن جدید از تابع δ است.

مثال: $\delta^*(q_0, ab) = \delta^*(\delta(q_0, a), b) = \delta^*(q_1, b) = \delta(q_1, b) = q_3$

دیاگرام انتقال



جلسه‌ی دهم

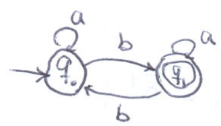
تعریف : زبان پذیرفته شده توسط DFA

هرگاه $M=\langle Q,\Sigma,\delta,q_0,F\rangle$ یک dfa باشد زبان پذیرفته شده توسط dfa ، M را با L(M) نمایش می‌دهیم.

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

مثال: زبان پذیرفته شده توسط dfa مقابل چیست؟

M:



$$L(M)=\{w\in\{a,b\}^* \mid n_b(w) \bmod 2 = 1\}$$

← باقی ماندن تعداد b بر 2 یک است ← یعنی تعدادش فرد است.

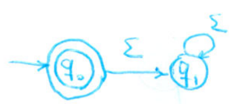
تعریف زبان منظم

زبان L روی حروف الفبای Σ را منظم یا Regular می‌نامیم، هرگاه dfa ای چون $M\langle Q,\Sigma,\delta,q_0,F\rangle$ چنان موجود باشد که : $L = L(M)$.

مثال: نشان دهید زبان‌های زیر منظم هستند:

1) $L=\emptyset$

2) $L=\lambda$



$$\delta^*(q_0,\lambda)=\lambda\in F$$

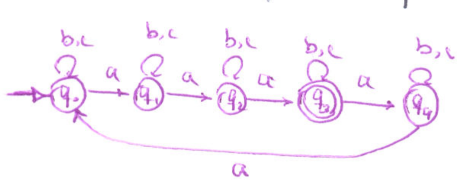
توجه: اگر زبان یک DFA بخواهد شامل رشته‌ی پوچ باشد، حتماً می‌بایست وضعیت شروع پایانی باشد.

3) $L=\Sigma^*$

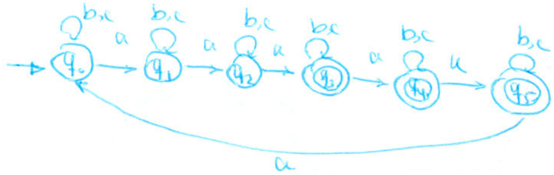
4) $L=\{w\in\{a,b,c\}^* \mid n_a(w) \bmod 5 = 3\}$

3
8
13
⋮

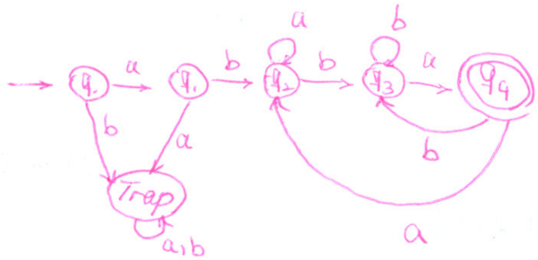
$\{0,1,2,3,4\}$



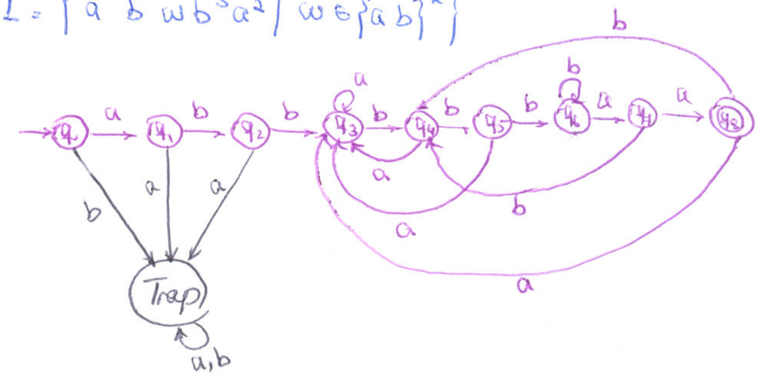
5) $L = \{ w \in \{a,b,c\}^* \mid n_a(w) \bmod 6 > 2 \}$ 3 4 5



6) $L = \{ ab\,w\,ba \mid w \in \{a,b\}^* \}$

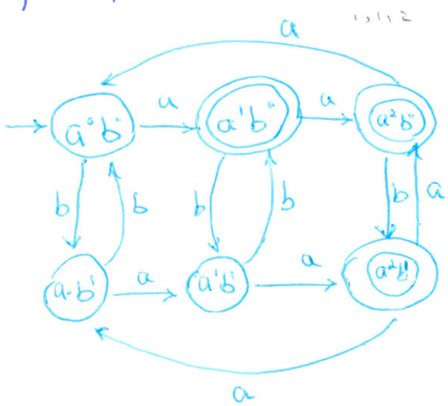


7) $L = \{ a\,b^2 w\,b^3 a^2 \mid w \in \{a\,b\}^* \}$



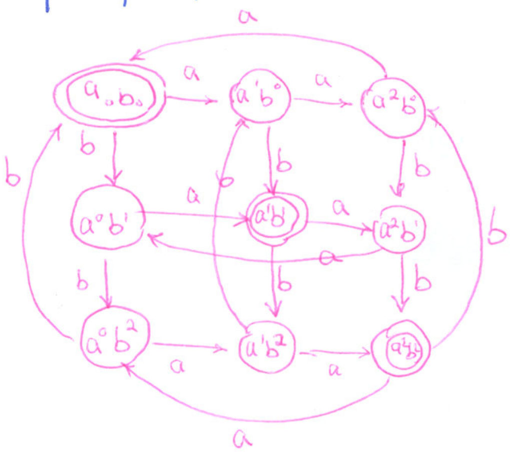
8) $L = \{ w \in \{a,b\}^* \mid n_a(w) \bmod 3 > n_b(w) \bmod 2 \}$

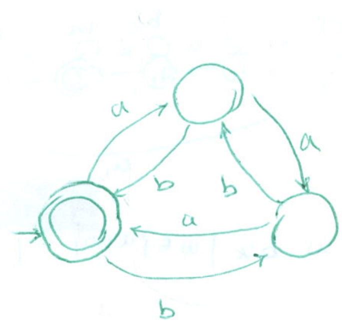
0, 1, 2 0, 1



۷۱

9) $L = \{ w \in \{a,b\}^* \mid n_a(w) \bmod 3 = n_b(w) \bmod 3 \}$





تمرین: DFA ها را رسم کنید.

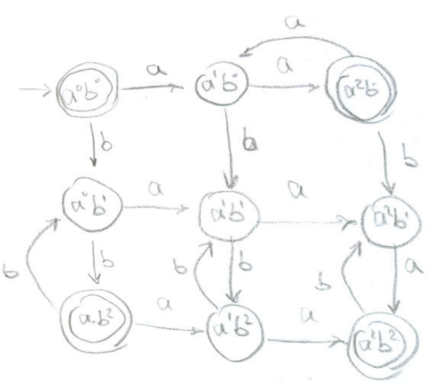
① $L = \{ a^n b^m \mid (n+m) \bmod 2 = 0 \}$ → n+m عدد زوج

زوج n,m

فرد n,m

② $L = \{ w \in \{a,b\}^* \mid n_a(w) \overset{4}{\equiv} n_b(w) \}$

①



مثال: نشان دهید $L=\{a^nb^n \mid n\geq 0\}$ منظم نیست.

این زبان منظم نیست چرا که باید تعداد a را جایی ذخیره کند تا هم به همین تعداد b ساخته شود، نمی شود. فضای ذخیره سازی نداریم.

اثبات: (برهان خلف)

$$p \to q \equiv p \wedge \neg q \to F$$

فرض کنید L منظم باشد در این صورت DFA ای چون $M\langle Q,\Sigma,\delta,q_0,F\rangle$ چنان وجود دارد که: $L=L(M)$. در نتیجه تعریف می کنیم $S_i=\{\delta^*(q_0,a^i) \mid i\in W\}$ پس $\subseteq Q$ که چون $|Q|<\infty$ پس $|S|<\infty$

(نتیجه ی δ^* همیشه یک وضعیت است) ✓

پس طبق اصل لانه ی کبوتری برای دو عدد طبیعی متمایز k_1 و k_2 داریم: $(k_1\neq k_2)$

$$(1)\ \delta^*(q_0,a^{k_1})=\delta^*(q_0,a^{k_2})$$

$$\delta^*(q_0,a^{k_1}b^{k_2})=\delta^*(\delta^*(q_0,a^{k_1}),b^{k_2})$$

$$\text{طبق اصل } (1) = \delta^*(\delta^*(q_0,a^{k_2}),b^{k_2})$$

$$=\delta^*(q_0,a^{k_2}b^{k_2})\in F$$

در نتیجه $\delta^*(q_0,a^{k_1}b^{k_2})\in F$ پس $a^{k_1}b^{k_2}\in L$ و $k_1\neq k_2$ که این یک تناقض است.

✓ هرگاه L یک زبان منظم باشد، آن گاه متمم L نیز یک زبان منظم خواهد بود.

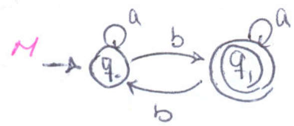
اثبات: چون L یک زبان منظم است DFA ای چون $M=\langle Q,\Sigma,\delta,q_0,F\rangle$ چنان وجود دارد که $L=L(M)$ حال اگر این DFA را به گونه ای تغییر دهیم که هر وضعیت نهایی آن به وضعیت غیر نهایی و هر وضعیت غیر نهایی به وضعیت نهایی تبدیل شود، DFA به دست آمده یعنی $\overline{M}=\langle Q,\Sigma,\delta,q_0,Q\setminus F\rangle$ DFA ای خواهد بود که متمم L را می پذیرد. $L(\overline{M})=\overline{L(M)}=\overline{L}$

نتیجه: متمم هر زبان نامنظم، نامنظم است.

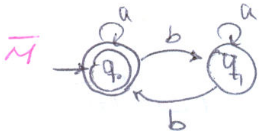
زیرا اگر متمم یک زبان نامنظم، منظم می بود، متمم آن نیز که خود زبان اصلی است باید منظم بود که طبق فرض زبان نامنظم بوده است.

۸/

مثال:



$L(M) = \{\omega \mid n_b(\omega) \bmod 2 = 1\}$



$L(\overline{M}) = \{\omega \mid n_b(\omega) \overset{2}{\equiv} 0\}$

نکته: اشتراک دو زبان منظم، منظم است. به عبارت دیگر خانواده‌ی زبان‌های منظم تحت اشتراک بسته هستند.

فرض کنید L_1 ، L_2 دو زبان منظم باشند که به ترتیب توسط DFA های $M_1 = \langle Q, \Sigma, \delta_1, q_0, F_1 \rangle$ و $M_2 = \langle Q, \Sigma, \delta_2, P_0, F_2 \rangle$ پذیرفته شوند.

یعنی

$$L(M_1) = L_1 \quad , \quad L(M_2) = L_2$$

حال DFA $\hat{M} = \langle \hat{Q}, \Sigma, \hat{\delta}, \hat{q}_0, \hat{F} \rangle$ را به طوری زیر تعریف می‌کنیم:

$$\hat{Q} = Q \times P$$

$$\hat{F} = F_1 \times F_2$$

$$\hat{\delta} : \hat{Q} \times \Sigma \to \hat{Q} \qquad \hat{\delta}(\langle q, p \rangle, a) = \langle \delta_1(q, a), \delta_2(p, a) \rangle$$

$$\hat{q}_0 = \langle q_0, P_0 \rangle$$

با این تعریف:

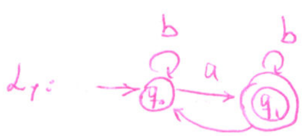
$$L(\hat{M}) = L(M_1) \cap L(M_2) = L_1 \cap L_2$$

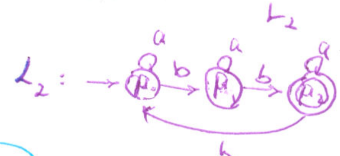
مثال:

$$L = \{\omega \in \{a,b\}^* \mid n_a(\omega) \bmod 2 = 1 \ , \ n_b(\omega) \bmod 3 = 2\}$$

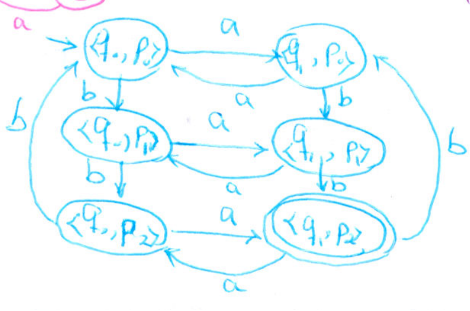
نشان دهید L منظم است.

$$= \underbrace{\{\omega \in \{a,b\}^* \mid n_a(\omega) \bmod 2 = 1\}}_{L_1} \cap \underbrace{\{\omega \in \{a,b\}^* \mid n_b(\omega) \bmod 3 = 2\}}_{L_2}$$





تعداد وضعیت‌ها = $2 \times 3 = 6$



وضعیت شروع، همیشه یکی

Final ها، ضرب همدیگر

نکته: اجتماع دو زبان منظم، منظم خواهد بود. به عبارت دیگر خانواده‌ی زبان های منظم روی حروف الفبای Σ تحت عمل اجتماع بسته هستند.

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

نکته: زیرمجموعه های زبان‌های منظم، لزوماً منظم نیستند.

$\underbrace{\{a^n b^n \mid n \geq 0\}}_{\text{منظم نیست}} \subseteq \underbrace{\{a,b\}^*}_{\text{منظم}}$

آیا زیرمجموعه های زبان های نامنظم، منظم هستند؟

$\underbrace{\phi}_{\text{منظم}} \subseteq \underbrace{\{a^n b^n \mid n \geq 0\}}_{\text{منظم نیست}}$ لزوماً منظم نیست

نکته: $A_1 \subseteq B \subseteq A_2$ ، A_1 و A_2 منظم، B لزوماً منظم نیست

$\phi \subseteq \{a^n b^n \mid n \geq 0\} \subseteq \Sigma^*$

نکته: خانواده های نامنظم تحت اجتماع بسته نیستند
یعنی لزوماً اجتماع دو زبان نامنظم، نامنظم نمی باشد.

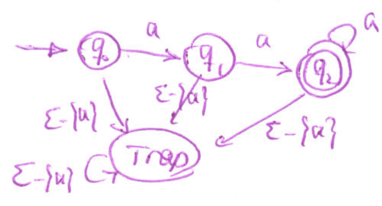
نکته: خانواده‌ی زبان های نامنظم تحت اشتراک نیز بسته نیستند
یعنی لزوماً اشتراک دو زبان نامنظم نامنظم نمی شود.

نکته: در مورد منظم یا نامنظم بودن اجتماع و اشتراک یک زبان منظم با یک زبان نامنظم، نمی توان اظهار نظری کرد.

نکته: در مورد الحاق یک زبان منظم و یک زبان نامنظم نیز نمی توان اظهار نظری کرد.

$\underbrace{L_2 = \{a^k \mid k \geq 0\}}_{\text{منظم}}$ $\quad$ $\underbrace{L_1 = \{a^p \mid p \text{ عدد اول}\}}_{\text{نامنظم}}$

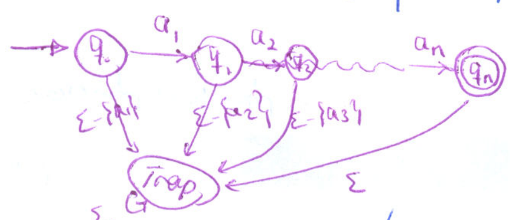
$L_2 \cdot L_1 = \underbrace{\{a^2, a^3, a^4, a^5, \ldots\}}_{\text{منظم}}$



q_1

نکته: هرگاه Σ مجموعه‌ای حروف الفبای $w=a_1a_2\cdots a_n$ رشته‌ای روی Σ باشد،

$L=\{w\}$ منظم است.



نتیجه: هرگاه L یک زبان روی حروف الفبای Σ باشد که $|L|<\infty$ آنگاه L منظم است.

$L=\{a^n b^{2^n} \mid n<m!\ ,\ m\le 1388\}$ متناهی $\Leftarrow$ منظم

نکته: اجتماع نامتناهی از زبان‌های منظم، لزوماً منظم نمی‌باشد.

هم منظم

$L_0=\{a^0b^0\}$

$L_1=\{a^1b^1\}$

$L_2=\{a^2b^2\}$

$\vdots$

$L_k=\{a^kb^k\}$

$L=\bigcup_{k=0}^{\infty} L_k=\underbrace{\{a^nb^n \mid n\ge 0\}}_{\text{نامنظم}}$

توجه: مجموعه‌ای را شمارا می‌نامیم که یا متناهی باشد و یا با اعداد طبیعی تناظر یک به یک باشد.

یعنی عناصر آن مجموعه را می‌توان در یک آرایه بنشانیم.

1	2	3	4	5	6
0	1	2	3	4	5

← Q^+ و Q^- شمارا . Q^+ ، Q^- باهم اجتماعشان شمارا (یکی از این یکی از آن)

پس Q شماراست.

نکته: اجتماع شمارای از زبان‌های منظم لزوماً منظم نیست.

نکته: اشتراک نامتناهی از زبان‌های منظم، لزوماً منظم نمی‌باشد.

$L_0=\overline{\{a^0b^0\}}$

$L_1=\overline{\{a^1b^1\}}$

$\vdots$

$L_k=\overline{\{a^kb^k\}}$

$\bigcap_{k=0}^{\infty} L_k=\bigcap_{k=0}^{\infty}\overline{\{a^kb^k\}}=\overline{\bigcup_{k=0}^{\infty}\{a^kb^k\}}$ $\Rightarrow$ نامنظم

* NFA (NDFA)

Non-deterministic Finite Automaton

یک NFA یک پذیرنده‌ی متناهی غیرقطعی (نامعین) یک پنج تایی $M=(Q,\Sigma,\delta,q_0,F)$ که در آن:

1) Q مجموعه‌ای متناهی و ناتهی از وضعیت‌های داخلی است

2) Σ مجموعه حروف الفبا

3) δ $\delta: Q\times\Sigma\cup\{\lambda\}\to 2^Q$ تابع انتقال (گذر) نامیده می‌شود.

4) $q_0\in Q$ وضعیت شروع

5) $F\subseteq Q$ مجموعه وضعیت‌های نهایی است.

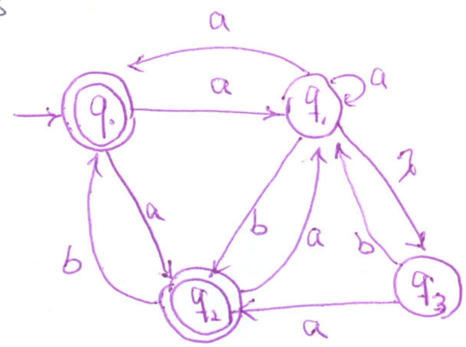
مثالی از یک NFA

$$M=\langle\{q_0,q_1,q_2,q_3\},\{a,b\},\delta,q_0,\{q_0,q_2\}\rangle$$

δ تابع تعریف شده به شکل زیر nfa است.

$Q \backslash \Sigma\cup\{\lambda\}$	a	b	λ
q_0	$\{q_1,q_2\}$	$\emptyset$	$\emptyset$
q_1	$\{q_0,q_1\}$	$\{q_2\}$	$\{q_3\}$
q_2	$\{q_1\}$	$\{q_0\}$	$\emptyset$
q_3	$\{q_2\}$	$\{q_1\}$	$\emptyset$

مطابق آن چه که گفته شد، برای NFA نیز می‌توان دیاگرام انتقال رسم کرد. که دیاگرام انتقال یک NFA را دیاگرام انتقال غیرقطعی می‌نامیم.



λ Transition ← بدون این که به سیستم انرژی وارد کنیم، از یک وضعیت به وضعیت دیگری می‌رود.

✓ می‌خواهیم تابع δ را به تابع δ^* مجدداً تعمیم دهیم که با دریافت هر وضعیت و هر رشته تمام وضعیت‌هایی را که می‌توان با خوانده شدن تمام رشته بدان دست یافت را بیابیم.

توجه می‌کنیم که رشته‌ی w دارای تعدادی نشانه‌های برابر است. تمام رشته‌هایی که از اضافه کردن تعدادی رشته‌ی پوچ (λ) به دو طرف رشته‌ی w به دست آمده باشند، همان رشته‌ی w است (از نگاه ما) بنابراین $\delta^*(q,w)$ نتیجه‌ی پیمایش همه‌ی رشته‌های w از وضعیت q است.

۱۰۱

$$\delta^*(q_1, b) = \{ \underset{\lambda b}{q_2}, \ \underset{\lambda b \lambda}{q_1}, q_3 \}$$

$\checkmark$ $\forall a \in \Sigma \quad \forall q \in Q \quad \delta(q,a) \subseteq \delta^*(q,a)$

جلسه ی بعد:
- زبان پذیرفته شده توسط NFA
- معادل بودن NFA و DFA
- کمینه کردن DFA ها

جلسه‌ی سوم

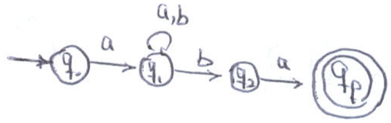
زبان پذیرفته شده توسط پذیرنده متناهی غیرقطعی nfa:

هرگاه $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ یک NFA باشد، زبان پذیرفته شده توسط M را با $L(M)$ نمایش می‌دهیم.

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

یعنی حداقل به یک وضعیت پایانی برسد

مثال: NFA طراحی کنید که $L = \{ awba \mid w \in \{a,b\}^* \}$ را بپذیرد.



فقط کافی است از یک راه به رشته برسیم آن گاه رشته پذیرفته شده است.

۱ برخلاف DFA ها در یک NFA می‌توان از یک وضعیت با یک حرف الفبای به هیچ یا یک یا چند وضعیت دست یافت.

۲ در یک NFA می‌توانیم λ Transition داشته باشیم. از یک وضعیت با یک برچسب پوچ به وضعیت دیگری برویم.

← به این دو دلیل است NFA ها را اتوماتاهای غیرقطعی می‌نامند.

نکته: از آن جایی که تابع هر DFA را می‌توان تابع یک NFA در نظر گرفت، پس هر زبان پذیرفته شده توسط یک DFA، توسط یک NFA نیز پذیرفته می‌شود.

تعریف دو اتوماتای معادل (هم‌ارز):

دو اتوماتای M_1 و M_2 را معادل با هم ارز می‌نامیم هرگاه:

$$L(M_1) = L(M_2)$$

زبان‌هایی که می‌پذیرند با هم یکسان باشند.

۱۱/

نکته: به ازای هر زبان پذیرفته شده توسط یک NFA یک DFA معادل با آن وجود دارد که آن زبان را می پذیرد.

به عبارت دیگر الگوریتمی وجود دارد که هر NFA $M_N = \langle Q_N, \Sigma, \delta_N, q_{0N}, F_N \rangle$ یک DFA

$$M_D = \langle Q_D, \Sigma, \delta_D, q_{0D}, F_D \rangle$$

می توان ایجاد کرد که:

$$L(M_N) = L(M_D)$$

الگوریتم ایجاد DFA MD

هرگاه NFA M_N داده شده باشد، در این صورت:

۱) قرار می دهیم

$$q_{0D} = \{q_{0N}\}$$

۲) هرگاه وضعیت $\{q_i, q_{i+1}, \ldots, q_j\}$ از DFA، M_D ایجاد شده باشد، برای $a \in \Sigma$ قرار می دهیم:

$$\delta_D(\{q_i, q_{i+1}, \ldots, q_j\}, a) := \bigcup_{k=i}^{j} \delta_N^*(q_k, a)$$

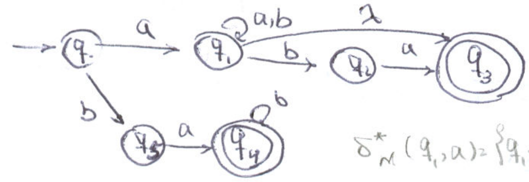
اگر حاصل اجتماع مذکور $(q_k, q_{k+1}, \ldots, q_M)$ باشد، در این صورت این وضعیت را به عنوان یک وضعیت DFA، M_D در نظر می گیریم.

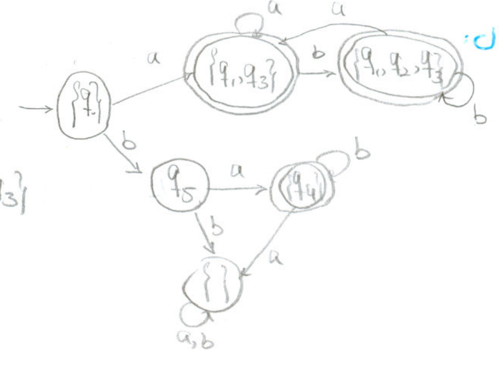
۳) گام ۲ را آن قدر تکرار می کنیم تا نتوان وضعیت جدیدی ایجاد کرد.

۴) برای معین کردن وضعیت های نهایی هر وضعیتی که شامل یک وضعیت نهایی از NFA، M_N باشد، آن وضعیت را به عنوان وضعیت نهایی در نظر می گیریم.

۵) اگر $\lambda \in L(M_N)$ باشد، (متعلق به زبان NFA باشد) وضعیت شروع DFA را پایانی می کنیم.

مثال:





$\delta_M^*(q_0, a) = \{q_1, q_3\}$

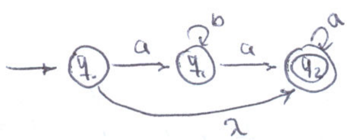
$\delta_N^*(q_1, a) = \{q_1, q_3\}$

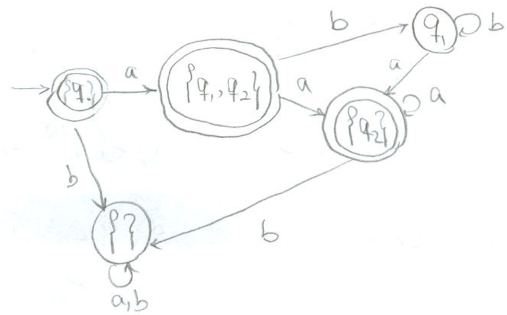
$\delta_N^*(q_1, a) = \{q_1, q_2, q_3\}$

$\delta_N^*(q_3, a) = \{\}$

$\delta_N^*\{q_3, b\} = \Phi$

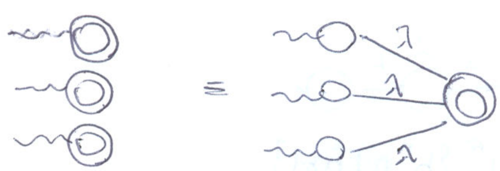
مثال: NFA شکل مقابل را به DFA معادلش تبدیل کنید.





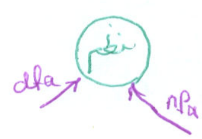
نکته: به ازای هر NFA داده شده، با N ، state (وضعیت) DFA معادل با آن، حداکثر دارای 2^N وضعیت خواهد بود.

نکته: به ازای هر NFA یک NFA معادل با آن چنین موجود است که صرفاً یک وضعیت پایانی دارد.



نکته: در برخی از کتب که تعریف متفاوت از NFA است، به NFA این امکان را می‌دهند که چندین وضعیت شروع داشته باشد. این تعریف دقیقاً معادل با تعریف ماست.

نکته: خانواده‌ی زبان‌های منظم دقیقاً خانواده‌ی زبان‌های پذیرفته شده توسط NFA هاست.



$$* \; N_\Sigma = D_\Sigma \; *$$

$N_\Sigma = \{ L \mid$ L توسط یک nfa روی Σ پذیرفته می‌شود. $\}$

$D_\Sigma = \{ L \mid$ L توسط یک dfa روی Σ پذیرفته می‌شود. $\}$

۱۲۱

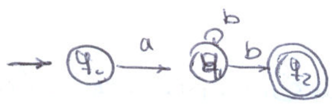
در پذیرش زبان‌ها DFA و NFA یکسان عمل می‌کنند

$$\begin{cases} D_\Sigma \subseteq N_\Sigma \\ N_\Sigma \subseteq D_\Sigma \end{cases}$$

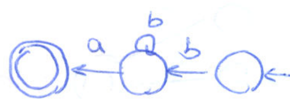
نکته: هرگاه L یک زبان منظم باشد، آنگاه L^R یک زبان منظم است.

چون L منظم است، پس یک NFA وجود دارد که L را می‌پذیرد. این NFA را می‌توان به NFA ای تبدیل کرد که فقط یک وضعیت پایانی داشته باشد، حال اگر وضعیت نهایی را به وضعیت شروع و وضعیت شروع را به نهایی تبدیل کنیم و به علاوه جهت فلش‌ها را معکوس کنیم، NFA به دست آمده L^R را می‌پذیرد.

مثال:



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

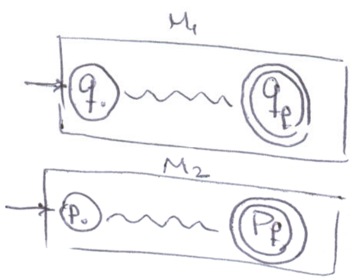


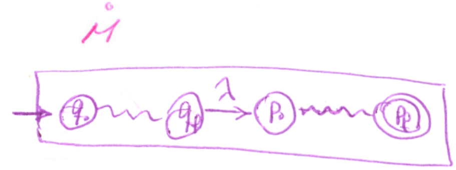
$L^R = \{b^n a \mid n \geq 1\}$

نکته: الحاق دو زبان منظم، منظم است. به عبارت دیگر خانواده‌های زبان‌های منظم تحت عمل الحاق بسته هستند.

هرگاه L_1 و L_2 دو زبان منظم پذیرفته شده توسط NFAهای M_1 و M_2 باشند، می‌خواهیم NFA ای بسازیم که $L_1 \cdot L_2$ را بپذیرد. برای این منظور می‌توان فرض کرد که NFA M_1 فقط یک وضعیت پایانی داشته باشد.

$$\begin{cases} L(M_1) = M_1 \\ L(M_2) = M_2 \end{cases}$$

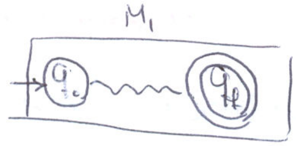


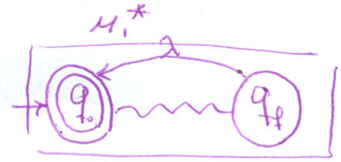


$L(\hat{M}) = L(M_1) \cdot L(M_2) = L_1 \cdot L_2$

نکته: اگر L یک زبان منظم باشد، بستار ستاره‌ای (Kleene Closure) L^* نیز یک زبان منظم خواهد بود.

هرگاه L، یک زبان منظم پذیرفته شده توسط NFA، M_1 با تنها یک وضعیت پایانی باشد، در این صورت NFA، M_1^* را مطابق زیر می‌توان در نظر گرفت:





نکته: زبان‌های منظم تحت عمل بر زدن بسته هستند. Shuffle

$$S(L_1,L_2)=\{(uv)\mid u\in L_1, v\in L_2, n\geq 0\}=(L_1\cdot L_2)^*$$

نکته: خارج قسمت راست

$$L_1 /^{r} L_2=\{x\mid \exists y\in L_2 \ \text{s.t.}\ xy\in L_1\}$$

$$\frac{xy}{y}$$

خارج قسمت چپ

$$l\,\frac{L_1}{L_2}=\{x\mid \exists y\in L_2 \ \text{s.t.}\ yx\in L_1\}$$

$$\frac{yx}{y}$$

$$* \quad \left(\frac{L_1}{L_2}\right)^R = l\,\frac{L_1^R}{L_2^R} \quad *$$

$$* \quad l\,\frac{L_1}{L_2} = \left(L_1^R /^{r} L_2^R\right)^R \quad *$$

مثال:

$$L_1=\{a^nb^m\mid n\geq 0, m\geq 1\}\cup\{ba\}=\{b^1,b^2,b^3,b^4,\dots,a^1b^1,a^1b^2,a^1b^3,\dots,a^2b^1,a^2b^2,a^2b^3,\dots\}\cup\{ba\}$$

$$L_2=\{b^m\mid m\geq 1\}=\{b^1,b^2,b^3,\dots\}$$

$$L_1 /^{r} L_2=\{\lambda,b,b^2,b^3,\dots,a^1,a^1b,a^1b^2,\dots,a^2,a^2b^1,a^2b^2,\dots\}=\{a^nb^m\mid n,m\geq 0\}$$

$$L_1 /_{l} L_2=\{\lambda,b^1,b^2,b^3,\dots\}\cup\{a\}=\{b^n\mid n\geq 0\}\cup\{a\}$$

۱۴۱

نکته: اگر L_1 یک زبان منظم و L_2 یک زبان دلخواه باشد، آن گاه L_1/L_2 یک زبان منظم است.

خانواده‌ی زبان‌های منظم تحت زبان خارج قسمت بر هر زبان دلخواه بسته هستند.

الگوریتم شناسه‌ی زبان منظم خارج قسمتی به صورتی که L_1 یک زبان منظم باشد

به عبارت دیگر اگر L_1 یک زبان منظم باشد در این صورت dfa ای وجود دارد که L_1 را می‌پذیرد.

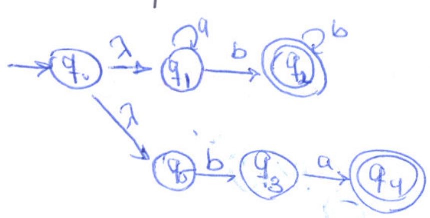
حال یک dfa دقیقاً مانند dfa زبان L_1 ترسیم می‌کنیم که برای آن هیچ وضعیت نهایی فعلاً در نظر گرفته نمی‌شود.

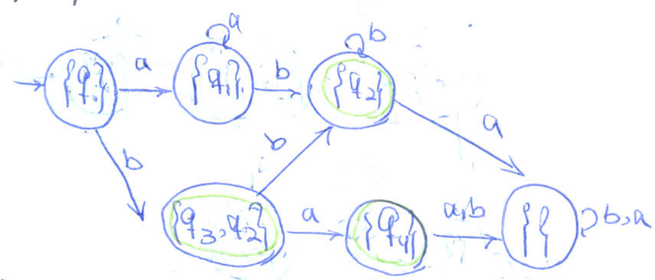
حال برای مشخص کردن وضعیت‌های نهایی DFA هم به شیوه‌ی زیر عمل می‌کنیم. اگر رشته‌ای از وضعیت q_0 از DFA زبان L_1 مشاهده می‌کنیم آیا با رشته‌ای از L_2 با وضعیت شروع از وضعیت q_0 در DFA زبان L_1 اگر به وضعیت نهایی دست یابیم، وضعیت q_0 DFA خارج قسمتی را نهایی می‌کنیم.

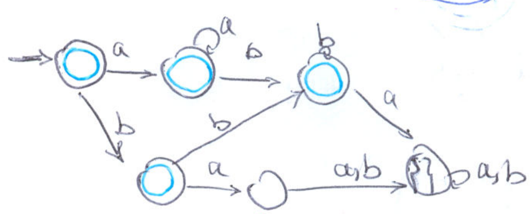
مجدداً همین کار را با سایر وضعیت‌های DFA دنبال می‌کنیم و روند مذکور را تکرار می‌کنیم.

مثال:

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







dfa خارج قسمت

هیچ رشته‌ای از L_2 نمی‌تواند به وضعیت نهایی برسد.

نکته: همین قضیه برای خارج قسمت چپ هم برقرار است.

نکته: اگر $\lambda \in L_2$

$$\lambda \in L_2 \Rightarrow L_1 \subseteq \frac{L_1}{L_2} r$$

نتیجه: اگر خارج قسمت دو زبان نامنظم است، در آن صورت زبان صورت نامنظم است.

$$\frac{\{\lambda, a\}}{a} r = \{\lambda\}$$

مثال:

$$L_1 = \{a^n b^n c^3 \mid n \geq 0\} \Rightarrow \text{نامنظم}$$

$$\frac{a^n b^n c^3}{\{c^3\}} = \{a^n b^n \mid n \geq 0\}$$

تعریف هم‌ریختی

هرگاه $(G_1, *_1)$ و $(G_2, *_2)$ دو مجموعه با عملگر باشند، گوییم

$\varphi : G_1 \to G_2$ یک هم‌ریختی یا هومومورفیسم است هرگاه:

φ تابعی تام از G_1 به G_2 باشد و به علاوه برای هر $g_1, g_2 \in G_1$ داشته باشیم:

$$\varphi(g_1 *_1 g_2) = \varphi(g_1) *_2 \varphi(g_2)$$

هم‌ریختی زبان‌ها

هرگاه Σ و Γ دو مجموعه حروف الفبایی باشند، آنگاه $(\Sigma^*, .)$ و $(\Gamma^*, .)$ دو عملگر هستند.

در این صورت $\varphi : \Sigma^* \to \Gamma^*$ یک هم‌ریختی است هرگاه φ یک تابع تام بوده و به علاوه برای هر $w_1, w_2 \in \Sigma^*$

$$\varphi(w_1 . w_2) = \varphi(w_1) \circ \varphi(w_2)$$

۱٤۱

نتیجه: برای هر ریختی φ همواره داریم: $\varphi(\lambda) = \lambda$

(تصویر رشته‌ی پوچ خودش است)

اثبات: $\varphi(\lambda) = \varphi(\lambda . \lambda) = \varphi(\lambda) . \varphi(\lambda) \Rightarrow$

$$|\varphi(\lambda)| = |\varphi(\lambda) . \varphi(\lambda)| = |\varphi(\lambda)| + |\varphi(\lambda)| \Rightarrow |\varphi(\lambda)| = 0$$

$$\Rightarrow \varphi(\lambda) = \lambda$$

مفهوم پایه

مجموعه‌ای که مولد کل فضاست و به نحو منحصر به فردی می‌توان هر یک از اعضا را با آن نشان داد.

مجموعه‌ای است که هر عضو از فضا را به شکل منحصر به فرد نمایش می‌دهد.

Σ^* دارای پایه‌ی Σ است. یعنی هر عضو از Σ^* را می‌توان به شکل منحصر به فردی از الحاق حروف Σ به دست آورد. در نتیجه Σ^* یک مونوئید (تکوار) آزاد نامیده می‌شود.

پایه دارد

بنابراین هر هم‌ریختی از Σ^* به Γ^* را می‌توان روی عناصر پایه تعریف کرد.

مثال:

$\Sigma = \{ \{, \} \}$ $\qquad$ $\Gamma = \{a, \ldots, z\}$

$\varphi : \Sigma^* \to \Gamma^*$

$\varphi(\{) = \text{begin}$

$\varphi(\}) = \text{end}$

$\varphi(\{\{\}\}) =$

$\varphi(\{)\varphi(\{\}\}) =$

$\text{begin } \varphi(\{).\varphi(\}\}) =$

$\text{begin begin } \varphi(\})\varphi(\}) =$

$\text{begin begin end end}$

نکته: تصویر همریختی زبان

هرگاه φ یک هم‌ریختی از Σ^* به Γ^* باشد آنگاه، هرگاه L یک زبان روی حروف الفبای Σ باشد. در این صورت تصویر هم‌ریختی L را با $\varphi(L)$ نمایش می‌دهیم و به شکل زیر تعریف می‌شود:

$$\varphi(L) = \{\varphi(w) \mid w \in \Sigma^*\}$$

مثال: $L = \{a^n b^n \mid n \geq 0\}$

$\varphi: \{a,b\}^* \to \{c,d,e,f\}^*$ همریختی صورت $\varphi(L) = \{(cd)^n (ef)^n \mid n \geq 0\}$

$\varphi(a) = cd$

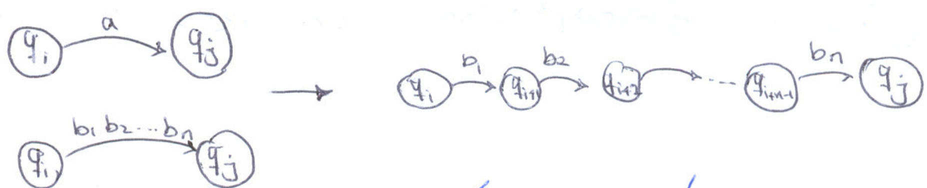
$\varphi(b) = ef$

نکته: هرگاه L یک زبان منظم باشد، آن گاه $\varphi(L)$ نیز منظم است که φ یک همریختی از Σ^* به Γ^* است و L یک زبان روی Σ^* است.

به عبارت دیگر زبان‌های منظم تحت همریختی‌ها بسته هستند.

چون L منظم است پس یک NFA وجود دارد که L را می‌پذیرد. حال یک NFA برای $\varphi(L)$ می‌سازیم که $\varphi(L)$ را بپذیرد. برای این منظور به ازای هر حرف $a \in \Sigma$ به عنوان برچسب از NFA زبان L، $\varphi(a)$ را جایگزین می‌کنیم به عنوان برچسب. اما ممکن است $\varphi(a)$ یک رشته مثل $\varphi(a) = b_1 b_2 \cdots b_n$ باشد.

درنتیجه اگر $\varphi(a)$ با طولی بزرگتر از یک باشد بین q_i و q_j به اندازه‌ی طول آن وضعیت جدید درج می‌کنیم.



بدین ترتیب برای زبان $\varphi(L)$ یک NFA به دست می‌آید. درنتیجه $\varphi(L)$ یک زبان منظم خواهد بود.

نکته: اگر $\varphi(L)$ نامنظم باشد، حتماً L نامنظم است.

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

$\varphi(a) = a$

$\varphi(b) = b$ $\qquad \varphi(L) = \{a^n b^n \mid n \geq 0\}$

$\varphi(c) = \lambda$

کمینه کردن DFA ها

اگر $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ یک DFA باشد رابطه‌ی $\sim$ را روی وضعیت‌ها به شکل زیر تعریف می‌کنیم

$$\forall p,q \in Q \ [p \sim q \Leftrightarrow (\forall w \in \Sigma^* \ \ \delta^*(p,w) \in F \Leftrightarrow \delta^*(q,w) \in F)]$$

۷ رابطه‌ی مذکور یک رابطه‌ی هم‌ارزی است که آن را رابطه‌ی ادغام‌پذیری می‌نامیم.

$$[p] = \{q \in Q \mid q \sim p\}$$

برای این که تلاقی‌ها به وجود نیایند، با کلاس‌های هم‌ارزی حوزه‌ها را حذف می‌کنیم سپس کار را آنقدر تکرار تا پایانی شود.
تمام عناصری که در یک کلاس هم‌ارزی می‌باشند را با هم ادغام می‌کنیم و برای آن‌ها یک وضعیت در نظر می‌گیریم.
به این ترتیب می‌نیمم DFA به دست آمده حتماً یک می‌نیمم منحصر است. (کلاس‌ها با هم اشتراک ندارند و اجتماع کلشان، کل مجموعه است)

روال علامت‌گذاری برای یافتن زوج وضعیت‌های ادغام‌ناپذیر

دو وضعیت p و q را ادغام‌ناپذیر می‌نامیم (Distinguishable) هرگاه برای رشته‌ای چون $w \in \Sigma^*$ از وضعیت p بتوان با این رشته به یک وضعیت پایانی رفت و از وضعیت q با همین رشته نتوان به وضعیت پایانی دست یافت و یا به عکس.

مراحل انجام روال علامت‌گذاری

۱ تمام وضعیت‌هایی که غیرقابل دسترس هستند را حذف می‌کنیم یعنی وضعیتی که از وضعیت شروع نتوانیم به آن برسیم.

۲ اگر $p \in F$ و $q \notin F$ باشند یا برعکس، زوج (p, q) را به عنوان زوج ادغام‌ناپذیر علامت می‌زنیم.

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

۳ هرگاه برای $a \in \Sigma$، $\delta(p, a) = p_a$ ، $\delta(q, a) = q_a$ باشد، به طوری که زوج (p_a, q_a) ادغام‌ناپذیر باشد، در این صورت زوج (p, q) را به عنوان زوج ادغام‌ناپذیر علامت می‌زنیم.

$$\delta^*(p_a, w) \in F \qquad \delta^*(p, aw) = \delta^*(\overbrace{\delta(p, a)}^{p_a}, w) \in F$$
$$\delta^*(q_a, w) \in F \qquad \delta^*(q, aw) = \delta^*(\overbrace{\delta(q, a)}^{q_a}, w) \notin F$$

۴ مرحله‌ی سه را آن‌قدر تکرار می‌کنیم تا هیچ زوج ادغام‌ناپذیری باقی نماند.

جلسه ی چهارم

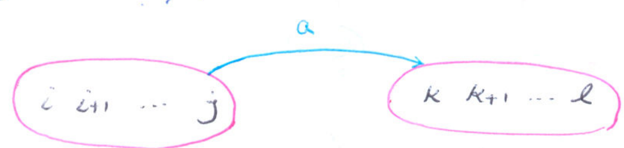
روال کمینه کردن DFA ها

هدف ← حداقل کردن تعداد وضعیت ها

مراحل این پروسه به شرح زیر است:

۱- روال علامت گذاری را فراخوانی می کنیم تا تمام جفت وضعیت های ادغام ناپذیر مشخص شوند. سپس وضعیت هایی که قابل ادغام هستند را با هم ادغام می کنیم. پس اگر وضعیت های q_i، q_{i+1}، ...، q_j ادغام پذیر باشند، در این صورت یک وضعیت $i\ i{+}1 \dots j$ را ایجاد می کنیم.

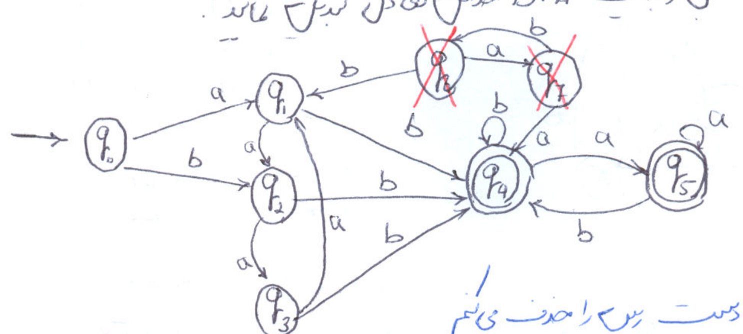
۲- اگر برای وضعیت q_i و حرف $a \in \Sigma$، $\delta(q_i, a) = q_k$ به طوری که q_k با وضعیت های $q_k \dots q_l$ ادغام پذیر باشد، آن گاه از وضعیت $i\ i{+}1 \dots j$ یالی با برچسب a به وضعیت $k\ k{+}1 \dots l$ ترسیم می کنیم. مطابق زیر:



۳- وضعیتی وضعیت شروع است، که اندیس صفر را داشته باشد.

وضعیتی وضعیت پایانی است که شامل اندیس یکی از وضعیت های نهایی باشد.

مثال: DFA شکل مقابل را به یک DFA حداقل معادل تبدیل نمائید.



۱- وضعیت های غیرقابل دسترس را حذف می کنیم

۲- وضعیت های ادغام ناپذیر (یکی final باشد یکی نه) را مشخص می کنیم

(q_0, q_4) (q_0, q_5)

(q_1, q_4) (q_1, q_5)

(q_2, q_4) (q_2, q_5)

(q_3, q_4) (q_3, q_5)

١٦١

$\delta(q_0, b) = q_2$

$\delta(q_1, b) = q_4$

$\delta(q_2, b) = q_4$

$\delta(q_3, b) = q_4$

$\delta(q_4, a) = q_5$

$\delta(q_5, a) = q_5$

$\delta(q_4, b) = q_4$

$\delta(q_5, b) = q_4$

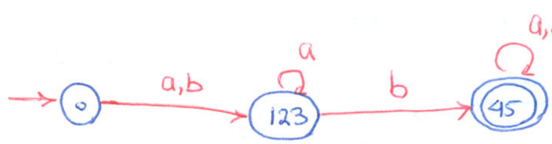
۳- در ستون‌ها سایر ادغام‌ناپذیرها را پیدا می‌کنیم.

(q_0, q_1)

(q_0, q_2)

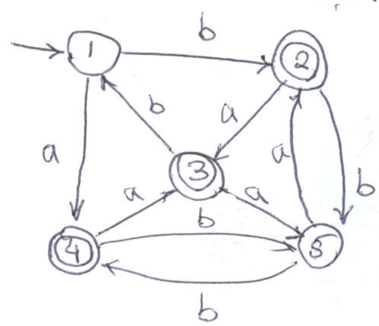
(q_0, q_3)

۴- این ماشین یک ماشین مینیمال است.



مثال: آتامات متناهی زیر را درنظر بگیرید. آتامات کمینه (مینیمال) مربوطه دارای چند حالت خواهد بود؟

۱) سه تا ✓ ۲) دو تا ۳) پنج تا ۴) چهار تا



(2,3) (5,2) (1,2)

(4,3) (5,3) (1,3)

(5,4) (1,4)

$5, a \to 2$ $5, b \to 4$

$1, a \to 4$ $1, b \to 2$

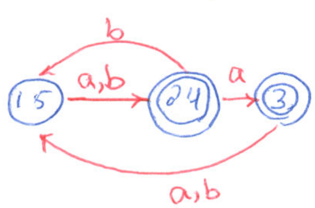
$2, b \to 5$ $3, b \to 1$

$2, a \to 3$ $3, a \to 5$ ×

$4, a \to 3$ $3, a \to 5$

1,5 ادغام می‌شوند

2,4 ادغام می‌شوند



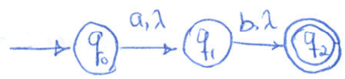
تمرین: هرگاه L یک زبان منظم باشد، درمورد منظم بودن زبان های زیر اظهار نظر نمائید.

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

① $L_1 = \{ w \in L \mid$ w از حذف هیچ یا یک یا چند حرف از رشته ای از L به دست می آید $\}$

منظم است

$L_1 = \{\lambda, a, b, c, ab, ac, bc, \dots\}$



L_1 منظم خواهد بود. زیرا چون L منظم است DFA ی وجود دارد که L را می پذیرد. اگر به برچسب هر یال آن λ را اضافه کنیم، NFA ای به دست می آید که زبان L_1 را می پذیرد.

② $L_2 = \{ w \in L \mid n_a(w) = 0 \}$

کافی است در DFA پذیرنده ی L هر یال با برچسب a را حذف کنیم شکل حاصل یک NFA خواهد شد پس L_2 منظم است.

یا هرچه a داریم را به Trap بفرستیم.

③ $L_3 = \{ w \in L \mid n_a(w) = 1 \}$

این زبان هم منظم است.

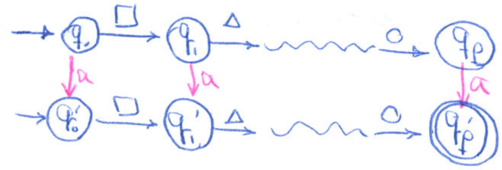
به ازای هر یال a یک copy تهیه و یال های a را حذف می کنیم

نکته: $n_a(w) = k$ ← منظم خواهد بود.

$L_{k_0} = \{ w \in L \mid n_a(w) = k_0 \}$

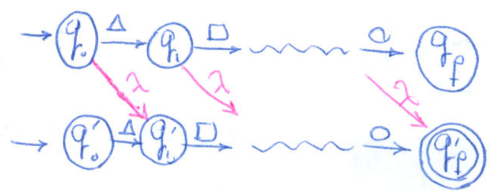
④ $L_4 = \{ uav \mid u, v \in \Sigma^*, uv \in L \}$

منظم است.



17,

⑤ $L_5 = \{uv \mid u,v \in \Sigma^*, \exists a \in \Sigma \text{ s.t } uav \in L\}$



سؤال: کدام یک از زبان های زیر منظم است؟

$L_1 = \{x^n y^n \mid x \in \{0,1\}^*, y \in \{0,1\}^*\}$ Σ^*

$L_2 = \{w \in L(A) \mid$ A یک dfa است و در مسیر پذیرش کلمه از هیچ حالتی بیش از یکبار عبور نمی شود $\}$ ؟A شکل حاصل NFA

$L_3 = \{w \in \{0,1\}^* \mid$ تعداد 0ها و 1ها برابر مقدار ثابت $n \geq 0$ است $\}$

$L_1 \subseteq \{0,1\}^*$

$\forall x \quad x \in \{0,1\}^* \Rightarrow x = x'\lambda' \in L_1$

$\{0,1\}^* \subseteq L_1$

$L_1 = \{0,1\}^*$ منظم

1) L_1 و L_3

2) L_2 و L_3

3) L_1 و L_2 و L_3 ✓

4) هیچ کدام منظم نیستند.

$L_3 = \{w \in \{0,1\}^* \mid n_1(w)=5\} \cap \{w \in \{0,1\}^* \mid n_0(w)=5\}$

هر یک منظم است $\Leftarrow$ L_3 منظم است

نکته: تعداد وضعیت های DFA کمینه ی L (زبان منظم L) با تعداد وضعیت های DFA کمینه $\bar{L}$ با هم برابرند.

نکته: هنگامی که NFA زبان L را داشته باشیم (منظم) نمی توان با complement کردن وضعیت ها (یعنی نهایی به غیر نهایی و بالعکس) به NFA زبان نهایی برسیم. ولی اگر DFA را داشته باشیم، می توانیم.

مثال: اگر $M = (Q, Q_0, \Sigma, F, \delta)$ یک اتومات متناهی باشد، تعریف می‌کنیم $\bar{M} = (Q, Q_0, \Sigma, Q-F, \delta)$

همچنین $d(M)$ اتومات قطعی معادل M خواهد بود. اگر M_1 و M_2 دو اتومات متناهی باشند. M_1+M_2

اتومات متناهی است که زبان آن اجتماع زبان‌های M_1، M_2 است. فرض کنید G_1 و G_2 دو گرامر منظم

باشند که زبان آن‌ها به ترتیب معادل زبان‌های M_1 و M_2 هستند. کدام عبارت صحیح است؟

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

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

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

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

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

هربار، complement می‌کنیم، می‌بایستی d آن را بگیریم که معین یا قطعی شود

عبارت های منظم

تعریف عبارت منظم

هرگاه Σ مجموعه حروف الفبایی باشد، گوییم R، عبارت منظم روی Σ است هرگاه

الف) $r=\phi$ ، $r=a$ برای $a\in\Sigma$ ، $r=\lambda$ ، را یک عبارت منظم پایه می نامیم

ب) اگر r_1 و r_2 دو عبارت منظم باشند، آن گاه r_1+r_2 (که گاهی با $r_1\cup r_2$ یا $r_1|r_2$ نمایش داده می شود) ، (r_1) ، r_1r_2 و r_1^* عبارات منظم خواهند.

ج) هر عبارتی که از ترکیب متناهی بار از گام های الف و ب به دست آمده باشد، یک عبارت منظم است.

مثال: هرگاه $\Sigma=\{a,b,c\}$ باشد، آن گاه:

$$r=(\underline{ab}+\underline{ac})^*(\underline{a+\lambda})+\phi+\underline{b}$$

یک عبارت منظم است

✓ برای هر عبارت منظم r روی حروف الفبای Σ می توان یک زبان به آن نسبت داد، که آن را با $L(r)$ نمایش می دهیم و به شکل زیر تعریف می شود:

۱) برای $r=a$ ، $L(r)=L(a)=a$ ، برای $r=\lambda$ داریم $L(\lambda)=\{\lambda\}$

و برای $r=\phi$ داریم $L(\phi)=\{\}$

۲) هرگاه r_1 و r_2 دو عبارت منظم باشند، آن گاه

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

$$L(r_1.r_2)=L(r_1).L(r_2)$$

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

مثال: هرگاه $r=(a+b)^*(a+\lambda)$ آن گاه $L(r)$ چیست؟

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

$$\underbrace{(L(a)\cup L(b))^*}_{\{a\}\cup\{b\}}\underbrace{(L(a)\cup\{\lambda\})}_{\{a,\lambda\}}=\{a,b\}^*.\{a,\lambda\}=\{a,b\}^*$$

تعریف دو عبارت منظم r_1 و r_2 معادل

دو عبارت منظم r_1 و r_2 را معادل می‌نامیم هرگاه $L(r_1)=L(r_2)$ و

در این صورت می‌نویسیم $r_1 \equiv r_2$ یا $r_1 = r_2$

✓ هرگاه r_1 و r_2 دو عبارت منظم روی Σ باشند، آنگاه:

1) $r_1 + r_2 = r_2 + r_1$

2) $r_1 r_2 \neq r_2 r_1$

3) $(r_1^*)^* = r_1^*$

4) $(r_1+r_2)^* = (r_1^* r_2^*)^* = (r_1+r_2^*)^* = (r_1^*+r_2^*)^*$

5) $r_1 + \emptyset = \emptyset + r_1 = r_1$, $r_1 . \emptyset = \emptyset . r_1 = \emptyset$

6) $\emptyset^* = \lambda^* = \lambda$

7) $r_1(r_2+r_3) = r_1r_2 + r_1r_3$

$(r_2+r_3)r_1 = r_2r_1 + r_3r_1$

8) $r_1(r_2r_3) = (r_1r_2)r_3$

مثال: برای زبان‌های زیر عبارات منظم را چنان بیابید که $L_i = L(r_i)$.

1) $L_1 = \{w \in \{0,1\}^* \mid$ w حداقل دارای یک زوج صفر متوالی باشد $\}$

$r_1 = (0+1)^* 00 (0+1)^*$

2) $L_2 = \{w \in \{0,1\}^* \mid$ w دقیقاً دارای یک زوج صفر متوالی باشد $\}$

$r_2 = (1+01)^* 00 (1+10)^*$

3) $L_3 = \{w \in \{0,1\}^* \mid$ w فاقد هر زوج صفر متوالی باشد $\}$

$r_3 = (0+\lambda)(1+10)^* = (1+01)^*(0+\lambda)$

$=(1+01)^*(1+10)^*$

4) $L_4 = \{a^n b^m \mid (n+m) \bmod 2 = 0\}$

$r_4 = (aa)^*(bb)^* + a(aa)^*(bb)^*b$

191

$(aa)^*(\lambda+ab)(bb)^* = (aa)^*(bb)^* + (aa)^*ab(bb)^*$

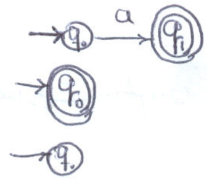
قضیه: زبان هر عبارت منظم، منظم است.

نکته: ابتدا نشان می‌دهیم زبان هر عبارت منظم، روی مجموعه حروف الفبای Σ یک زبان منظم است.

$r=a$

$r=\lambda$

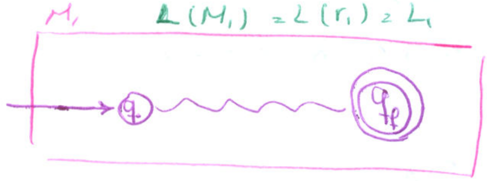
$r=\phi$



حال فرض کنید که r_1 و r_2 دو عبارت منظم باشند، که به ترتیب $L(r_1)$ و $L(r_2)$ توسط NFAهای M_1 و M_2 که تنها یک وضعیت نهایی دارند و به علاوه هیچ دو وضعیت هم‌نهایی ندارند، پذیرفته می‌شوند.

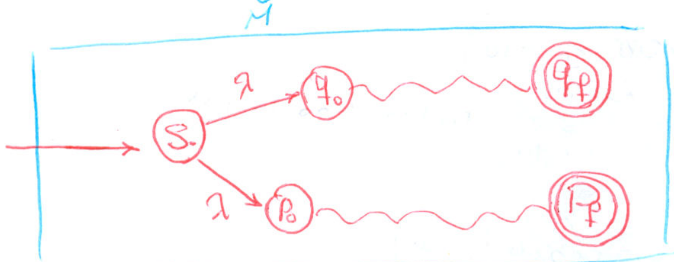
در این صورت:

M_1 $\quad L(M_1)=L(r_1)=L_1$



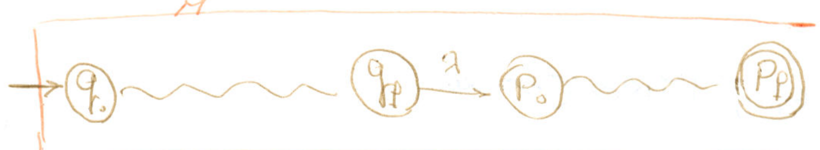
M_2 $\quad L(M_2)=L(r_2)=L_2$

$\overset{\cup}{M}$

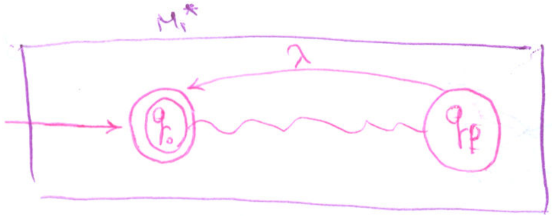


$L(\overset{\cup}{M})=L(M_1)\cup L(M_2)=L_1\cup L_2=L(r_1)\cup L(r_2)=L(r_1+r_2)$

$\overset{\cdot}{M}$



$L(\overset{\cdot}{M})=L(M_1)\cdot L(M_2)=L(r_1)\cdot L(r_2)=L(r_1r_2)$



$L(M_1^*) = (L(M_1))^* = (L(r_1))^* = L(r_1^*)$

نتیجه: زبان هر عبارت منظم، منظم است.

تعریف GTG Generalized Transition Graph

گراف انتقال تعمیم یافته

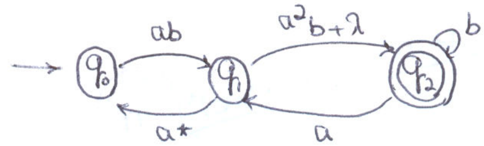
Generalized Transition Diagram GTD

دیاگرام انتقال تعمیم یافته

یک GTG یک NFA است که برچسب هر یال آن می تواند هر عبارت منظمی باشد.

درنتیجه هر NFA و هر DFA خودش یک GTG می باشد.

مثال:

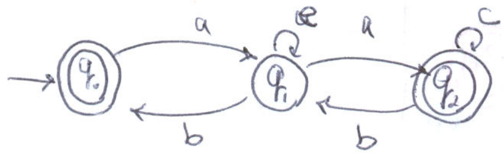


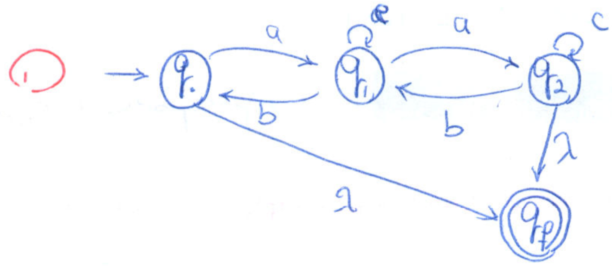
نکته: برای هر دیاگرام انتقال غیرقطعی یا قطعی می توان یک GTG با تنها دو وضعیت ایجاد کرد که یکی وضعیت نهایی و دیگری وضعیت شروع است به طوری که این دو وضعیت متمایز از یکدیگرند.

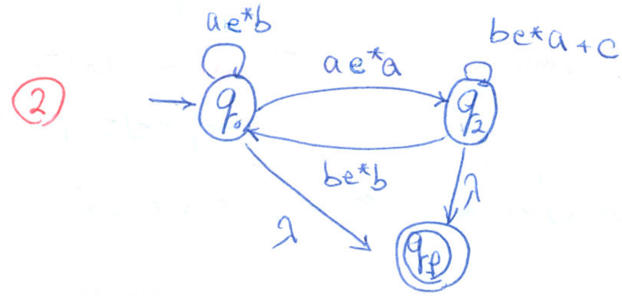
اثبات: ابتدا NFA یا DFA داده شده را به NFA ی تبدیل می کنیم که فقط یک وضعیت نهایی داشته باشد و به علاوه این وضعیت نهایی متمایز از وضعیت شروع باشد.

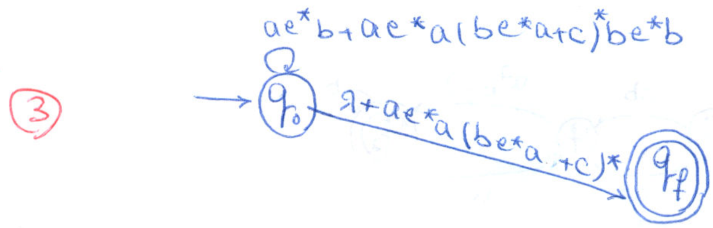
سپس وضعیت های میانی را به طور متوالی یکی یکی حذف می کنیم به طوری که با حذف هر وضعیت عملکرد آن را روی وضعیت های مجاورش اعمال می کنیم. تا این که فقط دو وضعیت شروع و پایانی باقی بمانند.

مثال:



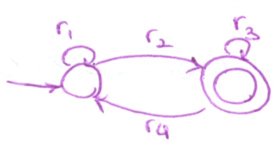






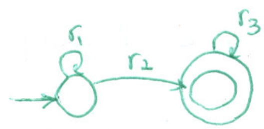
نکته: به ازای هر زبان منظم مانند L روی حروف الفبای Σ یک عبارت منظم r چنان موجود است که $L(r)=L$ باشد. به عبارت دیگر همه‌ی زبان‌های منظم را می‌توان با عبارات منظم نمایش داد.

اثبات: از آن جایی که L منظم است، می‌توان برای آن یک GTG با دو وضعیت مطابق کننده‌ی نگهداری کرد. کلی‌ترین حالت این GTG به شکل زیر است.



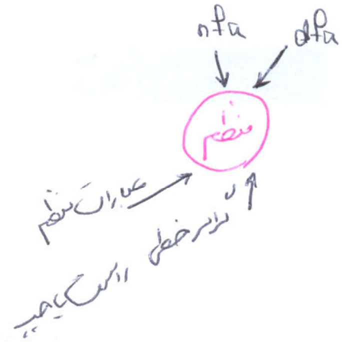
در نتیجه عبارت منظم معادل با GTG مذکور عبارت است از:

$$r_1^*(r_3+r_4r_1^*r_2)^*$$



$r_1^* r_2 r_3^*$

نتیجه: خانواده ی زبان های منظم دقیقاً خانواده ی زبان های پذیرفته شده توسط عبارت های منظم هستند.



* کدام یک از زبان های زیر نامنظم است.

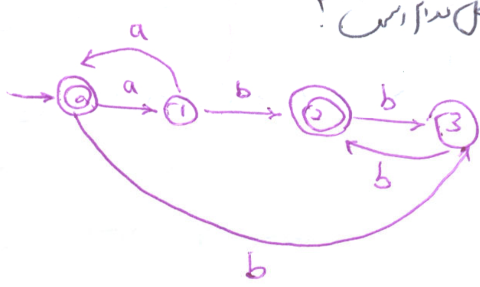
1) $\{a,b\}^* \subseteq \{a^n b^n (a+b)^* \mid n \geq 0\} \subseteq \{a,b\}^*$

2) ✓ $\{b^* a^n b^n a^* \mid n \geq 0\}$ چون باید n تا a و n تا b خوانده شود.

3) $L(a^* b^*) \subseteq \{a^* a^n b^n b^* \mid n \geq 0\} \subseteq \{a^* b^*\}$

4) هر سه نامنظم هستند.

* زبان ماشین حالت متناهی نشان داده شده در شکل کدام است؟



1) $(aa)^*(ab+\lambda)(bb)^*$

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

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

4) ✓ هر سه مورد

$\{a^n b^m \mid (n+m) \bmod 2 = 0\}$

۲۶

* اگر $\Sigma = \{a,b,c\}$ و $\Sigma^* - L = \emptyset$ باشد آنگاه L کدامیک از زبان‌های زیر می‌تواند باشد؟

I. Σ^* II. $a^n b^{n^2} c^n$ III. $\emptyset$ IV. ε

1) فقط I

$L \subseteq \Sigma^*$

2) فقط IV

3) فقط I و III

4) I ، II ، III و IV ✓

* زبان $\{a^{2^n} b^{2^n} \mid n \leq 2^{100}\}$ از چه نوعی است؟

1) منظم ✓

2) مستقل از متن ولی منظم نیست.

3) حساس به متن ولی مستقل از متن نیست.

4) بدون محدودیت ولی حساس به متن نیست.

* زبان زیر با $\alpha, \gamma \subseteq \Sigma^*$ و $\beta \in \Sigma^*$ مفروضند. کدام گزینه صحیح است؟

$L_1 = \{\alpha^i (\alpha\beta)^j (\gamma\alpha)^i \quad j \geq 0, i \geq 0\} \subseteq \Sigma^*$ $\quad \gamma \in \Sigma^* \Rightarrow \gamma = \lambda^1 (\lambda\beta)^0 (\gamma\lambda)^1 \in L_1$

$\Sigma^* \subseteq L_1 \Rightarrow \Sigma^* = L_1$

$L_2 = \{\alpha^i (\alpha\beta)^j (\gamma\alpha)^{i+j} \quad j \geq 0, i \geq 1\}$

$\gamma = \lambda^1 (\lambda\beta)^0 (\gamma\lambda)^{1+0} \subseteq L_2$

$L_3 = \{\alpha^i (\alpha\beta)^j (\gamma\alpha)^i \quad j \geq 1, i \geq 1\}$

$\Sigma^* \subseteq L_2 \quad L_2 = \Sigma^*$

1) L_1، L_2 و L_3 هر دو منظم هستند. ✓

2) L_1 منظم و L_3 نامنظم است.

3) L_1 منظم و L_2 نامنظم است.

4) L_1 ، L_2 و L_3 همگی نامنظم هستند.

$\beta \subseteq \Sigma^+ \Rightarrow$

$\beta = \lambda^1 (\lambda\beta)^1 (\lambda\lambda)^1 = \beta \in L_3$

پس $\Sigma^+ \subseteq L_3$

چون λ عضو این زبان نیست پس Σ^* نمی‌تواند باشد

$L_3 = \Sigma^+ = \Sigma^* . \Sigma$

جلسه ی پنجم

تعریف گرامر

برای زبان های منظم دو دیدگاه بیان شده است [ماشینی (nfa, dfa) / عبارت های منظم

دیدگاه دیگری که مطرح شده است، گرامر است.

تعریف

گوییم چهارتایی مرتب $G = \langle V, T, S, P \rangle$ یک گرامر است، هرگاه:

۱- V مجموعه ی متناهی و ناتهی از علائمی باشد که آن مجموعه غیرپایانه ها (متغیرها) Nonterminal = Variable

معمولاً هر متغیر را با حرف بزرگ الفبای لاتین نشان می دهند.

۲- T مجموعه متناهی و ناتهی از پایانه ها Terminal است که $V \cap T = \emptyset$

به علاوه هر پایانه را معمولاً با حرف کوچک الفبای لاتین نمایش می دهند.

۳- $S \in V$ غیرپایانه ی شروع گرامر است

۴- P مجموعه قواعد ها تولید است و هر قاعده به شکل $\alpha \to \beta$ که در آن $\alpha, \beta \in (V \cup T)^*$

با توجه به محدودیتی که روی گرامر می گذاریم، نوع آن مشخص می شود.

انواع گرامر را می توان به شکل زیر نام برد:

۱- نوع صفر (گرامر بدون قید)

گرامر $G \langle V, T, S, P \rangle$ نوع صفر نامیده می شود، هرگاه هر قاعده ی آن $\alpha \to \beta$ باشد به طوری که

$\alpha \in (V \cup T)^+$

$\beta \in (V \cup T)^*$

قرارداد: هرگاه برای α داشته باشیم
$$\alpha \to \beta_1,\quad \alpha \to \beta_2,\quad \vdots\quad \alpha \to \beta_n$$
عناصری از P باشند، برای سادگی می نویسیم:

$\alpha \to \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$

گرامر نوع صفر را گرامر بدون محدودیت (بدون قید) نیز می نامند.

۲۲۱

مثال:

$G(\{S,A,B\},\{a,b\},S,P)$

$$P:\{S \rightarrow AB \mid bB$$
$$AB \rightarrow BA \mid aA \mid b \mid \lambda$$
$$bB \rightarrow Bb \mid a \mid bBa$$
$$a \rightarrow Ab \mid bB\}$$

✓ سمت چپ هر رشته ای می تواند باشد، حتی ترمینال

۲- گرامر نوع ۱ (حساس به متن) وابسته به متن   CSG   Context Sensitive

گرامر $G(V,T,S,P)$ را یک گرامر حساس به متن می نامیم هرگاه هر قاعده آن به شکل $\alpha \rightarrow \beta$ باشد به طوری که

$\beta, \alpha \in (V \cup T)^+$ $|\alpha| \leq |\beta|$

سمت چپ هم ترمینال هم غیرپایانی می تواند باشد

۳- گرامر نوع 2 (مستقل از متن) آزاد از متن   CFG   Context Free

هرگاه هر قاعده آن به شکل $A \rightarrow \beta$ به طوری که $A \in V$

$\beta \in (V \cup T)^*$ (حتی λ)

سمت چپ فقط غیرپایانه باشد.

۴- گرامر خطی (Linear)

هرگاه هر قاعده آن به شکل $A \rightarrow \beta$ به طوری که $A \in V$ به علاوه در رشته β

$\beta \in (V \cup T)^*$

حداکثر یک غیرپایانه وجود داشته باشد.

$S \rightarrow aAb$ (یک متغیر)   خطی

$C \rightarrow aDCb$ (دو متغیر)   غیرخطی

۵- گرامر خطی راست (Right linear)

هرگاه هر قاعده ای آن به شکل $A \rightarrow xB$ یا $A \rightarrow x$ (متغیر حتما باید سمت راست ظاهر شود) به طوری که

$A,B \in V$

$x \in T^*$

۶- گرامر خطی چپ left linear

هرگاه هر قاعده آن به شکل $A \to Bx$ یا $A \to x$ که در آن $A, B \in V$ و $x \in T^*$

۷- گرامر نوع 3 (گرامر منظم)

گرامری که صرفاً خطی راست یا خطی چپ باشد.

مثال: نوع هر یک از گرامرهای زیر را معین کنید.

قرارداد: برای نمایش گرامرها کافی است صرفاً قواعد آن را مشخص کنیم. غیرپایانه اولین قاعده را از سمت چپ با غیرپایانه شروع، شروع می‌کنیم.

1) $S \to AB$
$A \to aAb \mid \lambda$
$Ab \to bA \mid Bb \mid b$
$B \to aA \mid bB \mid \lambda$

طول سمت راست از سمت چپ ← حساس به متن نیست

سمت چپ هر رشته‌ای می‌تواند باشد

⇐ نوع صفر

2) $S \to AB$ ← خطی نیست
$A \to aAb \mid \lambda$
$B \to bBc \mid \lambda$

مستقل از متن (نوع ۲)

3) $S \to aSb \mid aAb$
$A \to cAd \mid \lambda$

مستقل از متن خطی

4) $S \to aS \mid bS \mid aA$
$A \to Ab \mid Ac \mid \lambda$

نه خطی راست نه خطی چپ ولی خطی است

5) $S \to aSb \mid SS \mid a \mid b$

مستقل از متن ← خطی

حساس به متن ← سمت راست از سمت چپ

بدون محدودیت ← همه گرامرها بدون محدودیت هستند

6) $S \to aS \mid bS \mid \lambda$

خطی راست ← خطی ← مستقل از متن

۲۲۱

7) $S \to AaB$

$aB \to bb \mid bB$

$Aa \to aAa \mid bb$

$B \to a$

سمت راست ≥ سمت چپ

حساس به متن

اشتقاق (Derivation)

هرگاه $G(V, T, S, P)$ یک گرامر باشد و $\alpha = uvw \in (V \cup T)^*$ باشد، به طوری که $v \to \beta$ قاعده‌ای از P باشد گوییم $u\beta w$ یک اشتقاق (مشتق مرتبه‌ی اول) از رشته‌ی α است و با نماد $uvw \Rightarrow u\beta w$ نمایش می‌دهیم.
هرگاه این جایگذاری هیچ یا یک یا چند بار باشد از نماد $\overset{*}{\Rightarrow}$ استفاده می‌کنیم.

$$\alpha \overset{0}{\Rightarrow} \alpha \quad , \quad \alpha \overset{1}{\Rightarrow} u\beta w \quad , \quad \alpha \overset{*}{\Rightarrow} \gamma$$

تعریف زبان پذیرفته شده توسط یک گرامر

هرگاه $G = (V, T, S, P)$ باشد. زبان پذیرفته شده توسط G را با $L(G)$ نمایش می‌دهیم:

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

مثال: زبان پذیرفته شده توسط G چیست؟

* $G: S \to aSb \mid \lambda$

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow \dots \Rightarrow a^n S b^n$$

λ $\quad ab \quad a^2b^2 \quad a^nb^n$

$L(G) = \{\lambda, ab, a^2b^2, \dots\}$

$L(G) = \{a^nb^n \mid n \geq 0\}$

* $L(G) = \{a^nb^{2n} \mid n \geq 0\}$

$S \to aSb^2 \mid \lambda$

* $L = \{a^nb^m \mid n \leq m \leq 2n\}$

$S = aSb \mid aSb^2 \mid \lambda$

$n = m$ یا $m = 2n$

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

$S \to AB$ | یا | $S \to aSb \mid Sb \mid b$

$a^n b^n b^+$ $\quad A \to aAb \mid \lambda$

$a^n b^+ b^n$ $\quad B \to bB \mid \lambda$

* $L = \{a^n b^m \mid n > m\}$ $\qquad S \to aSb \mid aS \mid a$

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

وقتی می‌خواهیم اجتماع بگیریم باید چند حالت کنیم

$S \to aSb \mid A \mid B$

$A \to aA \mid a$

$B \to Bb \mid b$

* $L = \{a^n b^{n+1} \mid n \geq 0\}$

$S \to aSb \mid b$

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

$a^n b^m c^n c^m$

$S \to aAc \mid A$ $\qquad$ تعداد مساوی a,c

$A \to bAc \mid \lambda$ $\qquad$ تعداد مساوی b,c

* $L = \{a^n b^m c^{|n-m|} \mid n, m \geq 0\}$ $\qquad S_1 \mid S_2$

$n > m \Rightarrow |n-m| = n-m = k \Rightarrow n = m+k$

$n < m \Rightarrow |n-m| = m-n = k \Rightarrow m = n+k$

S_1 $\quad L_1 = \{a^{m+k} b^m c^k \mid m, k \geq 0\}$ $\quad a^k a^n b^m c^k$ $\qquad \Rightarrow L = L_1 \cup L_2$

S_2 $\quad L_2 = \{a^n b^{n+k} c^k \mid n, k \geq 0\}$ $\quad a^n b^n b^k c^k$

$S \to S_1 \mid S_2$

$S_1 \to aS_1c \mid A$		$S_2 \to AD$
$A \to aAb \mid \lambda$	و	$A \to aAb \mid \lambda$
		$D \to bDc \mid \lambda$

۲۴۱

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

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

که همه ترکیبات تولید می‌شود

مثلاً عبارتی که به a ختم می‌شود : $S \Rightarrow SS \Rightarrow aSbS \Rightarrow abS \Rightarrow abbSa \Rightarrow abba$

* $L = \{ w \in \{a,b\}^* \mid n_a(w) = n_b(w)+1 \}$

$S \to AaA$

$A \to aAb \mid bAa \mid \lambda$

از ابتدا یک a اضافه می‌کنیم

(اول / وسط / آخر)

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

زبان تصویر آینه‌ای abb|bba

$S \to aSa \mid bSb \mid \lambda$

مثل تک تک

* $L = \{ x \in \{a,b\}^* \mid x = x^r \}$

$S \to aSa \mid bSb \mid b \mid a \mid \lambda$

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

۲ تا c ، ۲ تا b تولید می‌شود

$aAbc \to ab\,AC \to abBbcc$

$aaAb^2c^2 \leftarrow aBb^2c^2 \leftarrow$

می‌خواهیم A را پس حرف b بذاریم تا به تعداد

مساوی a ، b و c تولید کنیم

$S \to abc \mid aAbc$

$Ab \to bA$

$Ac \to Bbcc$

$bB \to Bb$

$aB \to aa \mid a^2A$ برای اینکه دوباره به حالت اول برگردد

دسته‌بندی زبان‌ها بر حسب گرامر

۱- تعریف زبان بدون محدودیت (نوع صفر)

زبان L را نوع صفر می‌نامیم هرگاه گرامر بدون محدودیت چون G چنان موجود باشد که $L = L(G)$.

۲- تعریف زبان نوع یک (حساس به متن) CSL

هرگاه گرامر حساس به متنی چون G چنان موجود باشد که $L = L(G)$ اما زبان حساس متن می‌تواند λ را داشته باشد ولی گرامر حساس به متن نمی‌تواند به طور معادل زبان L حساس به متن است اگر $L - \{\lambda\}$ توسط یک گرامر حساس به متن پذیرفته شود.

مثال: یک زبان حساس به متن است زیرا $L = \{a^n b^n c^n \mid n \geq 0\}$ زبان $L - \{\lambda\} = \{a^n b^n c^n \mid n \geq 1\}$ توسط گرامر پذیرفته می‌شود. *

بدون λ

۳- تعریف زبان مستقل از متن CFL نوع 2

هرگاه گرامر مستقل از متنی چون G چنان موجود باشد که $L = L(G)$ باشد.

۴- زبان خطی

هرگاه گرامر خطی چون G چنان موجود باشد که $L = L(G)$ باشد.

۵- زبان منظم

هرگاه توسط یک گرامر خطی چپ یا راست پذیرفته شود. به عبارت دیگر آن زبان خطی راست یا چپ باشد.

نکته: به ازای هر گرامر خطی راست مانند G، $L(G)$ یک زبان منظم است یعنی یک nfa مثل M وجود دارد که $L(M) = L(G)$

فرض کنید گرامر خطی راست G داده شده است. واضح است که هر قاعده‌ی آن به یکی از سه شکل زیر است:

$$A \to \lambda \quad \text{یا} \quad A \to B \quad \text{یا} \quad A \to a_1 a_2 \cdots a_n B$$

برای قواعدی چون ↑ می‌توان به شکل بازگشتی به قواعدی تبدیل کرد که سمت راست آن قواعد فقط یک ترمینال داشته باشیم:

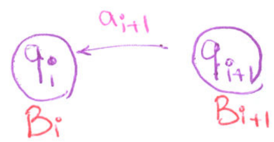
$$A \to a_1 a_2 \cdots a_n B$$

$$A \to a_1 B_1$$
$$B_1 \to a_2 B_2$$
$$B_2 \to a_3 B_3$$
$$\vdots$$
$$B_{n-1} \to a_n B$$

۲۵۱

- با این تغییرات در سطح گرامر به یک گرامر خطی جدیدی می‌رسیم حال به ازای هر متغیر پایانه در این گرامر جدید یک وضعیت از nfa در حال ساخت ایجاد می‌کنیم

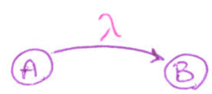
- حال برای قاعده‌ای مانند $B_i \to a_{i+1} B_{i+1}$ از وضعیت متناظر با B_i به وضعیت متناظر با B_{i+1} یالی با برچسب a_{i+1} رسم می‌کنیم مطابق شکل زیر



این وضعیت را ادامه می‌دهیم تا تمام وضعیت‌ها و برچسب‌های حال تکمیل شوند.

- برای مشخص کردن وضعیت‌های پایانی تمام وضعیت‌های متناظر با متغیرهای پایانه‌ای که به λ رفته‌اند، پایانی می‌کنیم یعنی اگر $A \to \lambda$

- برای قواعدی که الزاماً قاعده‌ای مانند $A \to B$ از وضعیت متناظر با متغیر پایانه سمت چپ به وضعیت متناظر با متغیر پایانه سمت راست یالی با برچسب λ رسم می‌کنیم.



در نتیجه: برای nfa ساخته شده که M نامگذاری می‌کنیم $L(M)=L(G)$

مثال:

$S \to abS \mid bbA \qquad S \to aS_1 \to abS \to abbA_1 \to abbbA \to abbb$

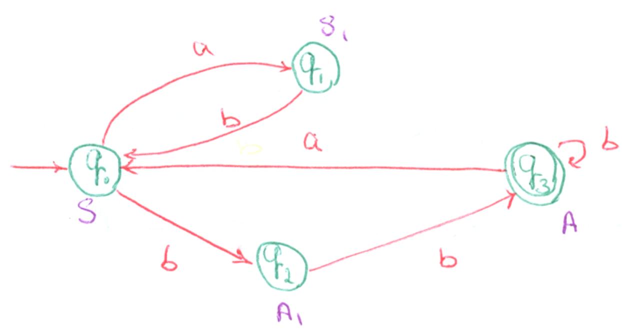
$S \to aS \mid bA \mid \lambda$

$S \to aS_1 \mid bA_1$

$S_1 \to bS$

$A_1 \to bA$

$A \to aS \mid bA \mid \lambda$



نکته: به ازای هر زبان منظم L یک گرامر منظم وجود دارد.

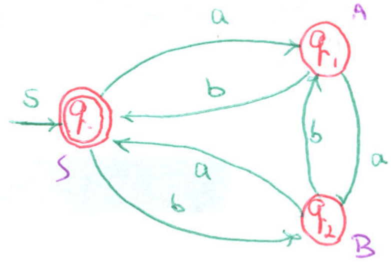
به ترتیب عکس آن چه که در اثبات نکته قبل گفته شد می‌توانیم یک گرامر خطی راست که برای زبان منظم L است بیابیم.

* $L = \{ w \in \{a,b\}^* \mid n_a(w) \overset{3}{\equiv} n_b(w) \}$

$S \to aA \mid bB \mid \lambda$

$A \to bS \mid aB$

$B \to bA \mid aS$



تعریف: دو گرامر G_1 و G_2 را هم ارز می‌نامیم هرگاه زبان پذیرفته شده توسط هر دو گرامر پذیرفته شده باشد:

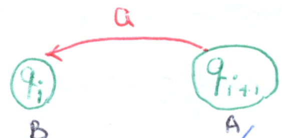
$$L(G_1) = L(G_2) \Rightarrow G_1 \equiv G_2$$

نکته: به ازای هر گرامر خطی راست یک گرامر خطی چپ معادل با آن وجود دارد و بالعکس.

اثبات: چون G یک گرامر خطی راست است، می‌توان یک NFA با تنها یک وضعیت پایانی برای آن ترسیم کرد. (Super Final) حال این NFA را به شرح زیر تغییر می‌دهیم:

- وضعیت شروع به پایانی و بالعکس تبدیل و جهت یال‌ها را هم معکوس می‌کنیم.
- حال به ازای هر State از این NFA یک متغیر پایانه در نظر می‌گیریم.

حال اگر داشته باشیم:



اکنون قاعده‌ی زیر را جایگزین می‌کنیم:

$A \to Ba$

بدین ترتیب یک گرامر خطی چپ ایجاد می‌شود که معادل با گرامر خطی راست داده شده است.

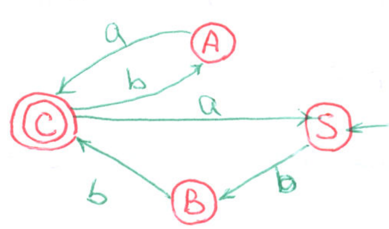
مثال: برای گرامر مثال قبل یک گرامر خطی چپ بیابید.

$S \to Bb \mid Sb$

$B \to Cb$

$C \to Sa \mid Ab \mid \lambda$

$A \to Ca$



$S \to Bb \to Cbb \to Abbb \to Cabbb \to abbb$

۲۶۱

✓ در کامپایلرها چون در استک‌ها ورودی‌ها از راست به چپ وارد می‌شوند، از گرامر خطی راست استفاده می‌کنند

نکته: گرامرهایی که هم خطی راست هم خطی چپ باشند فقط $A \to B \mid A \to x, \varepsilon, T^*$ است.

خواص تصمیم‌پذیری زبان‌های منظم

نمایش استاندارد از زبان منظم

شکل استاندارد از یک زبان منظم نمایشی است برای زبان منظم که در آن زبان منظم با یک DFA یا یک NFA یا یک عبارت منظم و یا یک گرامر خطی چپ یا راست نمایش داده می‌شود.

نکته: هرگاه L یک زبان منظم با نمایش استاندارد باشد، مسئله تهی بودن یا نبودن L یک مسئله تصمیم‌پذیر است. به عبارت دیگر الگوریتمی وجود دارد که تعیین می‌کند L تهی است یا خیر.

اثبات: چون L دارای نمایش استاندارد است از این نمایش می‌توان به نمایش DFA از زبان L دست یافت. حال این DFA را به یک DFA کمینه تبدیل می‌کنیم. اگر این DFA کمینه فاقد وضعیت نهایی باشد یعنی $L(G)$ تهی است. در غیر این صورت $L(G)$ ناتهی است.

نکته: هرگاه L یک زبان منظم با نمایش استاندارد باشد، مسئله متناهی بودن L یک مسئله تصمیم‌پذیر است چون L دارای نمایش استاندارد است. می‌توان فرض کرد این نمایش استاندارد همان DFA کمینه L باشد. حال اگر در این DFA کمینه یک حلقه از یک وضعیت به خودش یا به اجدادش داشته باشیم حتماً L نامتناهی است. اگر هیچ loop ای نداشته باشیم، متناهی است.

نکته: مسئله تساوی دو زبان منظم که هر دو دارای نمایش استاندارد هستند یک مسئله تصمیم‌پذیر است

$$A \hat{\oplus} B = (A \setminus B) \cup (B \setminus A) = (A \cap \bar{B}) \cup (B \cap \bar{A})$$

$$A \oplus B = \emptyset \Leftrightarrow A = B$$

چون L_1 و L_2 دارای نمایش استاندارد هستند $L_1 \oplus L_2$ نیز دارای نمایش استاندارد است. در نتیجه مسئله تهی شدن $L_1 \oplus L_2$ یک مسئله تصمیم‌پذیر است اگر $L_1 \oplus L_2 = \emptyset$ آنگاه $L_1 = L_2$ و اگر تهی نشد $L_1 \neq L_2$

مثال: به طور نشان به مسئله زیرمجموعه بودن $l_1 \subseteq l_2$ وقتی l_1 و l_2 دو زبان منظم با نمایش استاندارد هستند نیز یک مسئله تصمیم پذیر است.

چون l_1 و l_2 دارای نمایش استاندارد هستند پس $l_1 \setminus l_2$ نیز دارای نمایش استاندارد است. مسئله تهی شدن $l_1 \setminus l_2$ مسئله تصمیم پذیر است در نتیجه اگر $l_1 \setminus l_2 = \phi \Rightarrow l_1 \subseteq l_2$

نکته: فرض کنید l یک زبان منظم با نمایش استاندارد است. مسئله وجود یا عدم وجود رشته ای به طول n در l آیا یک مسئله تصمیم پذیر است؟

تمام رشته های به طول n متناهی (منظم) هستند در نتیجه می توان برای آن ها NFA ترسیم کرد. پس شکلی یک زبان منظم با نمایش استاندارد می دهند. اسم این زبان را l_n می نامیم. حال برای زبان داده شده l با نمایش استاندارد $l \cap l_n$ دارای نمایش استاندارد است. مسئله تهی بودن $l \cap l_n$ یک مسئله تصمیم پذیر است اگر $l \cap l_n = \phi$ یعنی رشته ای به طول n ندارد.

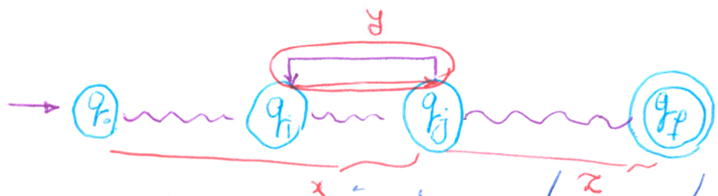
جلسه‌ی ششم

لم پامپینگ زبان‌های منظم

فرض کنید L یک زبان روی حروف الفبای Σ باشد. هرگاه برای هر n متعلق به اعداد طبیعی بتوان رشته‌ای چون ω متعلق به L را چنان یافت که $|\omega| \geq n$ باشد و برای هر تجزیه‌ی دلخواه حریف مانند $\omega = xyz$ که $|xy| \leq n$ ، $|y| \geq 1$ باشد، بتوان $i \in \mathbb{N}$ را چنان یافت که $xy^iz \notin L$
آن گاه L نامنظم است.

اثبات: فرض کنید L منظم باشد. در این صورت DFA ای چون M چنان موجود است که L را بپذیرد یعنی $L(M) = L$ ، $M = \langle Q, \Sigma, \delta, q_0, F \rangle$

هرگاه L توسط این DFA پذیرفته شود، طبق فرض سئوال بایستی برای $n = |Q| + 1$ رشته‌ای چون $\omega \in L$ چنان موجود باشد که $|\omega| \geq n$. بنابراین در پذیرش رشته‌ی ω توسط DFA ، اما بایستی حالت زیر رخ دهد:



چون $|\omega| \geq n$ طبق اصل لانه کبوتری بایستی یک حلقه داشته باشیم. (زیرا طول رشته از تعداد استیت‌ها بیشتر است)

حال برای تجزیه‌ی دلخواه $\omega = xyz$ به طوری که برای هر $i \in \mathbb{N}$ داریم $xy^iz \in L$

و این با فرض سئوال در تناقض است. پس فرض باطل می‌شود و در نتیجه L منظم نیست.

مثال: نشان دهید زبان $L = \{\omega\omega^R \mid \omega \in \{a,b\}^*\}$ منظم نیست.

حتما نیاز به حافظه داریم برای ذخیره‌ی این.

فرض کنید حریف عدد طبیعی n را داده باشد. قرار می‌دهیم

$\omega = (a^n b^n)(a^n b^n)^R$

$= a^n b^{2n} a^n$

در این صورت $|\omega| = 4n \geq n$

۲۸۱ حال برای تجزیه‌ی دلخواه حریف $a^n b^{2n} a^n = xyz$ که $|xy| \leq n$ ، $|y| \geq 1$.

درنتیجه $y=a^k$ که $k\ge 1$, $xz=a^{n-k}b^{2n}a^n$ درنتیجه $xy^iz=y^ixz=$

$$a^{ik}a^{n-k}b^{2n}a^n = a^{n+k(i-1)}b^{2n}a^n \xRightarrow{i=0} a^{n-k}b^{2n}a^n \notin L$$

پس L منظم نیست. ✓ در لم پامپینگ i هیچ گاه نباید یک باشد.

مثال: نشان دهید $L=\{a^p \mid p \text{ عدد اول}\}$ منظم نیست.

انواع سئوال ها از لم پامپینگ:
① صورت تست
② n را به دست آورید

فرض کنید حریف عدد n را داده باشد. قرار می‌دهیم $w=a^p$ که p اولین عدد اول بزرگتر از n است.

به وضوح $w\in L$ است و $|w|=p\ge n$

حال برای تجزیه دلخواه حریف $a^p=w=xyz$ که $|xy|\le n$, $|y|\ge 1$.

پس $y=a^k$ $xz=a^{p-k}$ درنتیجه

$$xy^iz=a^{ik}a^{p-k}=a^{p+k(i-1)} \xRightarrow{i=p+1} a^{p+k(p+1-1)}=a^{p+kp}=a^{p(k+1)}\notin L$$

روش دیگر برای اثبات نامنظم بودن زبان ها استفاده از خواص بستاری است

با استفاده از خواص بستاری زبان ها می‌توان ثابت کرد که یک زبان منظم نمی‌باشد

مثال: نشان دهید زبان $L=\{a^nb^m \mid n\ne m\}$ یک زبان منظم نیست.

فرض کنید که L منظم باشد. در این صورت $\bar{L}$ نیز منظم است. از طرفی $L(a^*b^*)$ نیز منظم است.

درنتیجه $\bar{L}\cap L(a^*b^*)$ $=\{a^nb^n \mid n\ge 0\}$ نیز باید منظم باشد. درصورتی که چنین نیست. این تناقض نشان می‌دهد که L نامنظم است.

مثال: نشان دهید زبان $L=\{w\in\{a,b\}^* \mid n_a(w)=n_b(w)\}$ منظم نیست.

فرض کنید L منظم باشد. در این صورت $L\cap L(a^*b^*)=\{a^nb^n \mid n\ge 0\}$ باید منظم باشد.
درحالی که منظم نیست.

زبان‌ها و گرامرهای مستقل از متن

یادآوری: گرامر $G = \langle V, T, S, P \rangle$ یک گرامر مستقل از متن است، هرگاه هر قاعده‌ی آن به شکل $A \to \alpha$ باشد که $A \in V$ ، $\alpha \in (V \cup T)^*$.

زبان پذیرفته شده توسط این گرامر را زبان مستقل از متن می‌نامند.

تعریف اشتقاق چپ Left Most Derivation

هرگاه $G = \langle V, T, S, P \rangle$ یک گرامر مستقل از متن باشد به طوری که $xA\alpha \in (V \cup T)^*$ به طوری که $x \in T^*$ ، $A \in V$ در این صورت یک اشتقاق چپ از $xA\alpha$ آن است که چپ‌ترین غیرپایانه موجود در عبارت $xA\alpha$ یعنی A را انتخاب و طبق قاعده‌ای مانند $A \to \beta$ به جای آن β را جایگزین کنیم.

$$xA\alpha \Rightarrow x\beta\alpha$$

مثال: $E \to E+E \mid E*E \mid (E) \mid id$ را در نظر بگیرید. یک اشتقاق چپ نشان دهید گرامر مذکور رشته $id+id*id$ را می‌پذیرد.

$$E \overset{lm}{\Rightarrow} E+E \Rightarrow id+E \Rightarrow id+E*E \Rightarrow id+id*E \Rightarrow id+id*id$$

اشتقاق راست $\Rightarrow E \overset{rm}{\Rightarrow} E+E \Rightarrow E+E*E \Rightarrow E+E*id \Rightarrow E+id*id \Rightarrow id+id*id$

تعریف درخت اشتقاق جزئی Partial Derivation Tree

یک درخت اشتقاق جزئی درختی است که ریشه‌ی آن هر یک از غیرپایانه‌های گرامر می‌تواند باشد. به علاوه هر گره‌ی داخلی آن نیز تنها یک غیرپایانه است. برگ‌های آن یا پایانه است یا λ و یا یک غیرپایانه. گره‌ای که λ است هیچ همزادی ندارد. به علاوه درخت‌های اشتقاق جزئی درخت‌های مرتب هستند.

✓ درخت مرتب درختی است که جایگاه فرزندان مهم باشد.

هم‌چنین برای هر رشته از پایانه‌ها و غیرپایانه‌ها، از گرامر مستقل از متن G که خود از یک غیرپایانه به دست آمده باشد، می‌توان یک درخت اشتقاق جزئی ایجاد کرد که برگ‌های آن از چپ به راست همان رشته‌ی داده شده باشند.

۲۹۱

زبان‌ها و گرامرهای مستقل از متن

یادآوری: گرامر $G=\langle V,T,S,P\rangle$ یک گرامر مستقل از متن است، هرگاه هر قاعده‌ی آن به شکل $A\to\alpha$ باشد که $A\in V$ ، $\alpha\in(V\cup T)^*$.

زبان پذیرفته شده توسط این گرامر را زبان مستقل از متن می‌نامند.

تعریف اشتقاق چپ Left Most Derivation

هرگاه $G=\langle V,T,S,P\rangle$ یک گرامر مستقل از متن باشد به طوری که $xA\alpha\in(V\cup T)^*$ به طوری که $x\in T^*$ ، $A\in V$ در این صورت یک اشتقاق چپ از $xA\alpha$ آن است که چپ‌ترین غیرپایانه موجود در عبارت $xA\alpha$ یعنی A را برحسب قاعده‌ای مانند $A\to\beta$ به جای آن β را جایگزین کنیم.

$$xA\alpha \Rightarrow x\beta\alpha$$

مثال: $E\to E+E \mid E*E \mid (E) \mid id$ را در نظر بگیرید. با یک اشتقاق چپ نشان دهید گرامر مذکور رشته $id+id*id$ را می‌پذیرد.

$$E \overset{lm}{\Rightarrow} E+E \Rightarrow id+E \Rightarrow id+E*E \Rightarrow id+id*E \Rightarrow id+id*id$$

اشتقاق راست $\Rightarrow E \overset{rm}{\Rightarrow} E+E \Rightarrow E+E*E \Rightarrow E+E*id \Rightarrow E+id*id \Rightarrow id+id*id$

تعریف درخت اشتقاق جزئی Partial Derivation Tree

یک درخت اشتقاق جزئی درختی است که ریشه‌ی آن هر یک از غیرپایانه‌های گرامر می‌تواند باشد. به علاوه هر گره‌ی داخلی آن نیز دقیقاً یک غیرپایانه است. برگ‌های آن یا پایانه است یا λ و یا یک غیرپایانه. گره‌ای که λ است هیچ همزادی ندارد. به علاوه درخت‌های اشتقاق جزئی درخت‌های مرتب هستند.

✓ درخت مرتب درختی است که جایگاه فرزندان مهم باشد.

هم‌چنین برای هر رشته از پایانه‌ها و غیرپایانه‌ها، از گرامر مستقل از متن G که خود از یک غیرپایانه به دست آمده باشد، می‌توان یک درخت اشتقاق جزئی ایجاد کرد که برگ‌های آن از چپ به راست همان رشته‌ی داده شده باشند.

مثال:

$S \to aSb \mid A$

$A \to cAd \mid \lambda$

$a^2c^2Ad^2b^2$

تعریف فرم جمله‌ای

دنباله‌ای از پایانه‌ها و غیرپایانه‌ها است که از غیرپایانه‌ی شروع گرامر به دست می‌آید.

مثال: $S \to aSb \mid bA$

$A \to cAd \mid \lambda$

c^3Ad^3

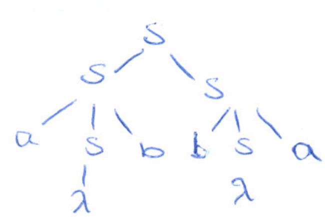
* فرم جمله‌ای نیست
* از غیرپایانه‌ی شروع نمی‌توان به آن دست یافت

تعریف درخت اشتقاق

درخت اشتقاق درخت مرتبی است که

1. اولاً درخت اشتقاق تجزیه بوده
2. به علاوه ریشه‌ی آن درخت غیرپایانه‌ی شروع گرامر باشد
3. برگ‌های آن صرفاً پایانه یا λ باشد.

مثال: برای رشته‌ی ab^2a یک درخت اشتقاق به دست آورید. $S \to SS \mid aSb \mid bSa \mid \lambda$

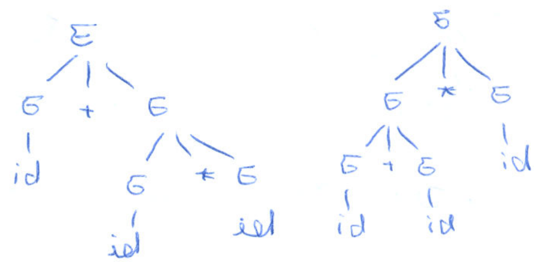


تعریف گرامر گنگ

گرامر مستقل از متن G را مبهم یا گنگ می نامیم هرگاه برای رشته ای از زبان این گرامر حداقل دو درخت اشتقاق متمایز داشته باشیم.

مثال: نشان دهید گرامر $E \to E+E \mid E*E \mid (E) \mid id$ گنگ است.

$id + id * id$



سوال اصلی این است که مسئله ی گنگ بودن صرفاً برای گرامرها قابل تعریف است یا ممکن است یک زبان هم گنگ باشد.

به عبارت دیگر آیا برای هر گرامر گنگ یک گرامر غیر گنگ معادل با آن وجود دارد یا خیر.

پاسخ منفی است. هرسیون ثابت کرد زبان مستقل از متنی وجود دارد که برای آن هیچ گرامر غیر گنگی وجود ندارد. به عبارت دیگر هر گرامری که آن زبان را بپذیرد گنگ است.

این گونه زبان ها را زبان های گنگ یا درجی گنگ ذاتاً گنگ می نامند.

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

✓ این مثال زبان ذاتاً گنگی است که در کتاب معرفی شده است.

$S \to S_1 \mid S_2$

$S_1 \to AB$

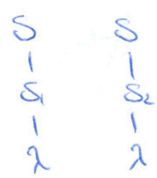
$B \to bBc \mid \lambda$

$A \to aA \mid \lambda$

$S_2 \to CD$

$C \to aCb \mid \lambda$

$D \to cD \mid \lambda$



۲۰۱

پوشش رشته‌ها (تجزیه رشته‌ها)

هرگاه G یک گرامر مستقل از متن باشد و w یک رشته‌ای دلخواه روی حروف الفبای گرامر G باشد، الگوریتمی که بررسی می‌کند $w \in L(G)$ هست یا خیر را الگوریتم پوشش رشته‌ی w می‌گوییم.

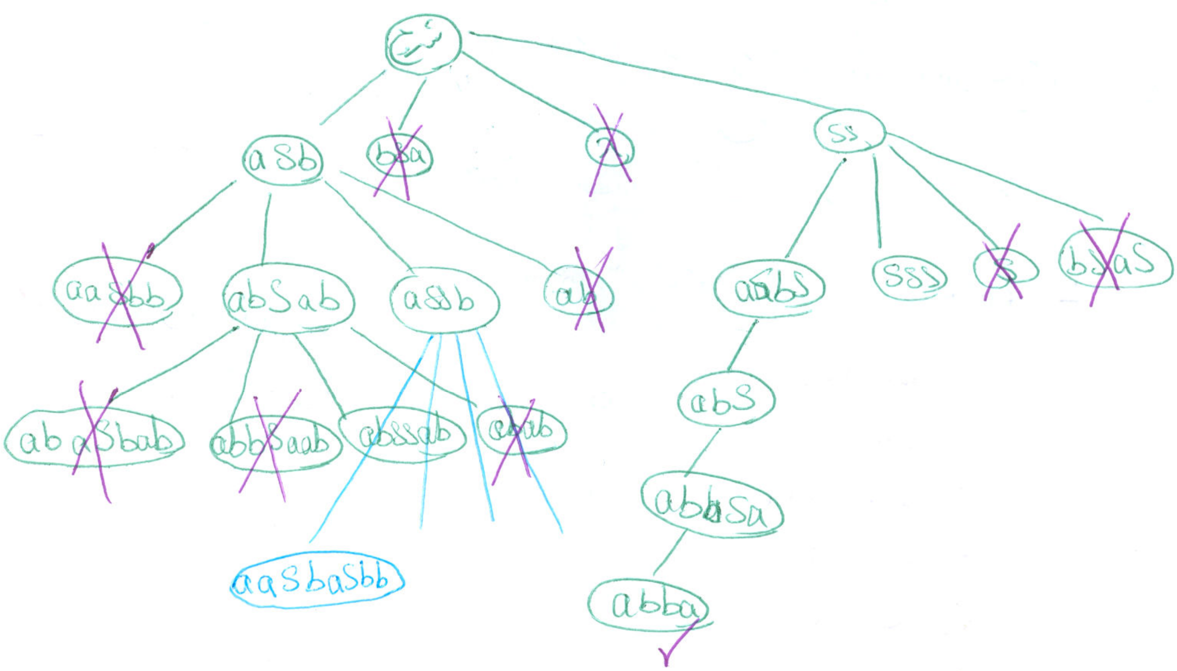
یکی از روش‌ها برای آن که نشان دهیم $w \in L(G)$ است روش Exhaustive Search یا جستجوی کامل است. (Brute Force)

در این الگوریتم که برگرفته از پیمایش BFS است ابتدا در ریشه غیرپایانه‌ی شروع گرامر قرار می‌دهیم و سپس به عنوان فرزندان این غیرپایانه تمام سمت راست‌های S را به عنوان فرزندان S در نظر می‌گیریم. از چپ به راست در یک درخت قرار می‌دهیم. بعد از آن به ازای هر غیرپایانه‌ای به کار رفته چپ‌ترین در سمت هر رشته روند بسط را تکرار می‌کنیم تا این که مطمئن شویم ادامه‌ی آن شاخه حل مسئله را به بیراهه می‌برد. در این صورت آن شاخه را هرس می‌کنیم.

مثال:

$S \to aSb \mid bSa \mid SS \mid \lambda$ abba

اگر رشته جزء زبان باشد، به جواب می‌رسد



نکته: چنانچه رشته‌ی داده شده عضو زبان $L(G)$ باشد، بالاخره این الگوریتم روی رشته توقف می‌کند، اما اگر متعلق به زبان نباشد، ممکن است unit production داشته باشیم. ولی اگر گرامر فاقد قاعده‌ی یکه و فاقد قاعده‌ی λ باشد، در این صورت گرامر یک گرامر غیر انقباضی است.

$A \to B$

$B \to A$

در این صورت مسئله‌ی توقف برای هر رشته‌ی متعلق به حروف الفبای گرامر رخ خواهد داد و می‌توان گفت که آن رشته متعلق به زبان گرامر مستقل از متن هست یا خیر.

ارتفاع درخت: $2|w|-1$

حداکثر: $|P|^0+|P|^1+\dots+|P|^{2|w|-1}$

$= O(|P|^{2|w|-1})$

نکته: $1+a+\dots+a^{n-1} = \dfrac{1-a^n}{1-a} = O(a^{n-1})$

ساده سازی گرامرهای مستقل از متن

گرامر ساده

گرامر مستقل از متن G را ساده می‌نامیم هرگاه هر قاعده‌ی آن به شکل $A \to av$ باشد که در آن $A \in V$ ، $v \in V^*$ ، $a \in T$ به طوری که زوج (A, a) حداکثر یکبار ظاهر شود.

مثال:

$S \to aAB$

$A \to bAA \mid cB$

$B \to dA \mid b$

اگر $A \to bAB$ را داشتیم دیگر ساده نبود.

نکته: گرامرهای ساده همواره گرامرهای غیرمبهم اند زیرا در پذیرش رشته‌ها فقط یک راه را نمایش می‌کند. مسئله‌ی تعلق یا عدم تعلق یک رشته به گرامر ساده در زمان $O(|w|)$ امکان پذیر است.

قضیه ساده سازی گرامرهای مستقل از متن

اگر $G = \langle V, T, S, P \rangle$ یک گرامر مستقل از متن باشد، به طوری که برای قاعده‌ی $A \to \alpha B \beta$ تنها قواعد $B \to y_1 \mid y_2 \mid \dots \mid y_n$ قواعد شامل B باشد. گرامری که $\hat{G}$ از حذف قاعده‌ی $A \to \alpha B \beta$ و جایگزینی قواعد $A \to \alpha y_1 \beta \mid \alpha y_2 \beta \mid \dots \mid \alpha y_n \beta$ به دست می‌آید معادل با گرامر G است.

۲۱۱

شکل : $S \to aAB \mid bAa$

$A \to aAb \mid \lambda$

$B \to aAb \mid bAc \mid a$

$S \to aAaAb \mid aAbAc \mid aAa$

$\hat{G}$ $\quad B \to aAb \mid bAc \mid a$ ⟵ نامفید

$A \to aAb \mid \lambda$

تعریف متغیر مفید $\quad$ useful

متغیر پایانی $A \in V$ از گرامر مستقل از متن $G = \langle V, T, S, P \rangle$ را مفید می‌نامیم هرگاه آن متغیر پایانی در حین مراحل تولید یک رشته از زبان $L(G)$ به کار رفته باشد. به عبارت دیگر داشته باشیم.

$$\exists\, w \in L(G):\ S \overset{*}{\Rightarrow} \alpha A \beta \overset{+}{\Rightarrow} w$$

تعریف متغیر نامفید $\quad$ useless

متغیر پایانی A را نامفید می‌نامیم، هرگاه مفید نباشد. به عبارت دیگر یا از متغیر پایانی شروع نتوان به آن رسید و یا از آن نتوان به رشته‌ای از زبان رسید.

تعریف قاعده‌ی نامفید

قاعده‌ای نامفید است که در آن حداقل یک متغیر نامفید به کار رفته باشد.

هم‌چنین قاعده‌ای مفید است که نامفید نباشد.

قضیه : هرگاه G یک گرامر مستقل از متن باشد، در این صورت یک گرامر مستقل از متن مانند $\hat{G}$ چنان موجود است که معادل با G است و به علاوه هیچ قاعده و متغیر نامفیدی ندارد.

نکته: کافی است تمام متغیرها و قواعد نامفید را از گرامر حذف کنیم. گرامر به دست آمده، فاقد هر متغیر و قاعده‌ی نامفید است.

مثال: زبان گرامر G کدام است؟

86

$G: S \to aAb \mid bBa \mid b\cancel{C}a$

$A \to aaAb \mid ab$

$B \to bBa \mid a$

$C \to a\cancel{C} \mid b\cancel{C}$

1) $a^{2K+2}b^{K+1} \cup b^{+}a^{+} \quad K \geq 0$

2) $a^{2K}b^{K} \cup (ba)^{*}a \quad K \geq 1$

3) $a^{K+1}b^{K} \cup b^{L}a^{L} \quad L \geq 1, K \geq 2$

4) $a^{2}a^{2K}b^{K}b^{2} \cup b^{L}a^{L+1} \quad K \geq 0, L \geq 1$ ✓

$S \to aAb \mid bBa$

$A \to aaAb \mid ab$

$B \to bBa \mid a$

$A \overset{*}{\Rightarrow} a^{2K}Ab^{K} \overset{*}{\Rightarrow} a^{2K}aAb\,b^{K} \Rightarrow aa^{2K}A\,b^{K}b$

$A \Rightarrow aa^{2K}A\,b^{K}b$

$S \Rightarrow aAb \Rightarrow a^{2}a^{2K}Ab^{K}b^{2}$

$B \Rightarrow bBa \overset{*}{\Rightarrow} b^{t}B\,a^{t} \Rightarrow b^{t}a^{t+1} \quad t \geq 0$

$S \Rightarrow bBa \to b^{\overset{l}{(t+1)}}a^{\overset{l+1}{(t+2)}} \quad t \geq 0, l \geq 1$

حذف تولید λ (λ-Free)

هرگاه $G = \langle V, T, S, P \rangle$ یک گرامر مستقل از متن باشد. در این صورت می‌توان با جایگذاری تولید λ گرامر G را به نحوی λ-Free کرد که حداکثر یک قاعده‌ی λ داشته باشد.

مثال:

$S \to AB$

$A \to aAb \mid \lambda$

$B \to cBd \mid \lambda$

$S \to A$, $S \to B$

با حذف λ باید $S \to \lambda$ می‌بود.

$S \to AB \mid B \mid A \mid \lambda$

$A \to aAb \mid ab$

$B \to cBd \mid cd$

۲۲۱

مثال :

$S \rightarrow AaB$

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

$B \rightarrow bB \mid \lambda$

$A \rightarrow AaB \mid Aa$

$A \rightarrow aAb \mid bB \mid \lambda \mid b$

$B \rightarrow bB \mid b$

B، λ-Free شد

$A \rightarrow AaB \mid Aa \mid aB \mid a$

$A \rightarrow aAb \mid bB \mid b \mid ab$

$B \rightarrow bB \mid b$

A، λ-Free شد

$S \rightarrow \lambda$ نداریم، چراکه زبان λ را نمی‌پذیرد.

تعریف قوانین تکی

یک NT به یک NT دیگر برود.

حذف قوانین تکی

هرگاه $G=\langle V,T,S,P \rangle$ یک گرامر مستقل از متن باشد که فاقد λ است، می‌توان این گرامر را از هر قاعده تکی‌ای پاکیزه کرد و به گرامر $\hat{G}$ معادل با G که فاقد هر قاعده‌ی تکی‌ای است، تبدیل کرد. ابتدا قواعد گرامر را به دو دسته تقسیم می‌کنیم :

① قواعدی که تکی نیستند

② قواعد تکی

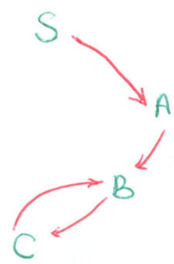
مثال:

$S \to aSb \mid aA \mid A$

$A \to B \mid bA \mid cA \mid a$

$B \to C \mid aA \mid bB \mid a$

$C \to B \mid aA \mid b$



قواعد غیر یکه را جدا می کنیم

$S \to aSb \mid aA$

$A \to bA \mid cA \mid a$

$B \to aA \mid bB \mid a$

$C \to aA \mid b$

از طریق قواعد جایگزین

دقت loop زده دیگر تکرار نمی کنیم

$S \to bA \mid cA \mid a \mid aA \mid bB \mid a \mid aA \mid b$

$A \to aA \mid bB \mid a \mid aA \mid b$

$B \to aA \mid b$

$C \to aA \mid bB \mid a$

درنتیجه

$S \to aSb \mid aA \mid bA \mid cA \mid a \mid bB \mid b$

$A \to bA \mid cA \mid a \mid aA \mid bB \mid b$

$B \to aA \mid bB \mid a \mid b$

$C \to aA \mid b \mid bB \mid a$

قضیه

به ازای هر گراف مستقل ارزش که $\lambda \notin L(G)$ آنگاه گراف مستقل ارزشی وجود دارد که فاقد قوانین λ، یکه و ناسفید است و به علاوه این گراف، معادل با گراف اولیه است. ابتدا λ را حذف می‌کنیم، سپس یکه را حذف می‌کنیم. سپس ناسفیدها را حذف می‌کنیم.

جلسه‌ی هفتم

فرم نرمال گریباخ

گوییم گرامر مستقل از متن $G=<V,T,S,P>$ در فرم نرمال گریباخ صدق می‌کند، هرگاه هر قاعده‌ی آن به شکل $A \to ax$ باشد به طوری که $a \in T$ ، $x \in V^*$ ، $A \in V$.

مثال: گرامر

$$S \to aAB \mid aS$$
$$A \to bA \mid bB$$
$$B \to aA \mid c$$

یک گرامر است که در فرم نرمال گریباخ صدق می‌کند.

قضیه: برای هر گرامر مستقل از متن G که $\lambda \notin L(G)$ یک گرامر $\hat{G}$ با فرم نرمال گریباخ معادل وجود دارد که $L(\hat{G}) = L(G)$

مثال: برای

$$G: S \to AB$$
$$A \to aAb \mid b$$
$$B \to cBd \mid c$$

یک گرامر $\hat{G}$ معادل با آن بیابید که در فرم نرمال گریباخ صدق کند.

$$S \to aAbB \mid bB$$
$$A \to aAb \mid b$$
$$B \to cBd \mid c$$

$$S \to aAB_bB \mid bB$$
$$A \to aAB_b \mid b$$
$$B \to cBB_d \mid c$$
$$B_b \to b$$
$$B_d \to d$$

۳۴

الگوریتم CYK:

هرگاه G یک گرامر مستقل از متن باشد که در فرم نرمال چامسکی صدق نماید، برای هر $w\in T^+$ الگوریتمی وجود دارد که می‌تواند در زمان $O(|w|^3)$ تعیین کند که رشته‌ی w متعلق به $L(G)$ هست یا نه!

فرض کنید $G=\langle V, T, S, P\rangle$ یک گرامر با فرم نرمال چامسکی باشد، $w=a_1a_2\cdots a_n\in T^+$ داده شده باشد. تعریف می‌کنیم:

$$w_{ij}:=a_ia_{i+1}\cdots a_{k-1}a_ka_{k+1}\cdots a_j$$

به وضوح $w_{ii}=a_i$ ، $w_{1n}=w$

هم چنین تعریف می‌کنیم

$$T[i,j]=\{A\in V \mid A\overset{*}{\Rightarrow} w_{ij}\}$$

با تعاریف مذکور واضح است که:

$$w\in L(G) \iff S\in T[1,n]$$

$$T[i,j]=\begin{cases}\{A\in V \mid A\overset{*}{\Rightarrow} \underbrace{w_{ii}}_{a_i}\} & i=j\\[2ex] \displaystyle\bigcup_{k=i}^{j-1}\{X\in V \mid X\to YZ,\ Y\in T[i,k],\ Z\in T[k+1,j]\} & j>i\end{cases}$$

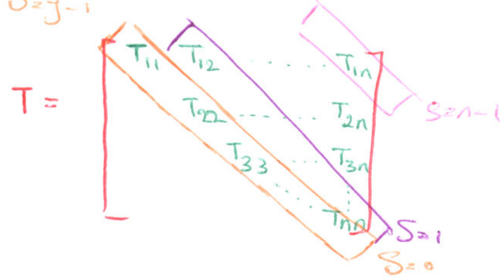
$$X\overset{*}{\Rightarrow} a_ia_{i+1}\cdots a_{k-1}a_ka_{k+1}\cdots a_j$$

به زبان دیگر:

$$T[i,j]=\begin{cases}\{A\in V \mid A\to a_i\} & (i=j) \quad s=0,\ i=1,2,\dots,n\\[2ex] \displaystyle\bigcup_{k=i}^{j-1}\{X\in V \mid X\to YZ,\ Y\in T[i,k],\ Z\in T[k+1,j]\} & (j>i)\end{cases}$$

$1\leq s\leq n-1$ ، $i=1,2,\dots,n-s$

تعریف: $s=j-i$



$j-i=s$

$j=s+i\leq n$

$\Rightarrow i\leq n-s$

```
for i = 1 to n do
        T[i, i] ← {x ∈ V | x → a_i}
for s = 1 to n-1 do
        for i ← 1 to n-s do
                j = i + s
                T[i, j] = ∪_{k=i}^{j-1} {x | x → YZ, Y ∈ T[i, k], Z ∈ T[k+1, j]}
return (S ∈ T[1, n])
```

$\Theta(n)$

$\Theta\left(\sum_{s=1}^{n-1} \sum_{i=1}^{n-s} s\right)$

$\Theta(n) + \Theta(n^3)$

$= \Theta(n^3)$

$= \Theta(|w|^3)$

✓ CLR و ... از رتبه‌ی خطی هستند پس هزینه‌شان بیشتر است

اما زبان‌های بیشتری را پوشش می‌دهد.

زبان های مستقل از متن

خواص بستاری

لم پمپینگ زبان های مستقل از متن

فرض کنید L یک زبان روی حروف الفبای Σ باشد، به طوری که برای هر عدد طبیعی n رشته‌ای چون $w \in L$ چنان موجود باشد که $|w| \geq n$ و برای هر تجزیه $w = uvxyz$ که طول $|vxy| \leq n$, $|vy| \geq 1$ بتوان $i \geq 0$ چنان یافت که:

$uv^ixy^iz \notin L$

آنگاه L مستقل از متن نیست.

اثبات: فرض کنید L زبانی باشد مستقل از متن که در شرایط مذکور صدق کند

⇐ گرامر مستقل از متنی وجود دارد که L را تولید کند. طبق مطالب خوانده شده می‌دانیم که اگر L مستقل از متن باشد $L - \{\lambda\}$ نیز مستقل از متن است و بالعکس.

پس برای می‌توان فرض کرد که از ابتدا زبان L فاقد رشته‌ی پوچ باشد.

هم‌چنین می‌توان فرض کرد که گرامر مستقل از متن G فاقد قواعد λ، یکه و نامفید باشد. یعنی گرامر غیر انقباضی است گرامر مذکور.

حال فرض اگر n را به گونه‌ای انتخاب کنیم که n بزرگتر از تعداد غیرپایانه های گرامر باشد،

در این صورت بایستی رشته‌ای چون $w \in L(G)$ چنان موجود باشد که $|w| \geq n$ و به علاوه $S \overset{*}{\Rightarrow} w$. اما اگر در مسیر پذیرش w از یک غیرپایانه حداقل یک بار استفاده شده باشد قاعدتاً طول w بایستی کمتر از n می‌بود پس بایستی در مسیر پذیرش w از یک غیرپایانه بیش از یک بار استفاده شده باشد.

$$S \overset{*}{\Rightarrow} uAz \overset{*}{\Rightarrow} u v \underset{\Downarrow * \; x}{A} y z \overset{*}{\Rightarrow} w$$

ثابت پامپینگ = تعداد Non Terminal ها

(لونده‌ی گوهر)

پس

$$A \overset{*}{\Rightarrow} vAy$$

$$A \overset{*}{\Rightarrow} x$$

در نتیجه: $S \overset{*}{\Rightarrow} uv^i x y^i z \quad \forall i \in \mathbb{N}$

در نتیجه تناقض با فرض است.

مثال: ثابت pumping lemma برای زبان‌های مستقل از متن با گرامر $G = \langle S, V, T, P \rangle$ کدام است؟

1. تعداد واژه‌های زبان در T
2. ✓ تعداد واژه‌های نحوی در V — نتیجه: تعداد غیرپایانه‌ها
3. تعداد قواعد تولید در P
4. هیچکدام

مثال: زبان $L = \{a^n b^n c^n \mid n \geq 0\}$ مستقل از متن نیست.

فرض کنید حریف عدد طبیعی n را داد، با شرط قرار می‌دهیم $w = a^n b^n c^n \in L$ به طوری که $|w| = 3n \geq n$

حال برای تجزیه دلخواه حریف $a^n b^n c^n = uvxyz$ که $|vxy| \leq n$ و $|vy| \geq 1$ داریم:

در نتیجه ممکن است یکی از حالات زیر رخ دهد:

1) $vxy = a^k b^{k'}$ که $k, k' \geq 0$ و $k + k' \geq 1$

2) $vxy = b^k c^{k'}$ ، $k, k' \geq 0$ ، $k + k' \geq 1$

با قرار دادن $i=0$ نتیجه می‌شود که

$$uv^0 x y^0 z \notin L$$

٣٦١

- در حالت اول با قرار دادن $i=0$ از تعداد a ها یا b ها حداقل یکی کم می‌شود در حالی که از تعداد c ها کاسته نمی‌شود پس عنصر به دست آمده متعلق به L نخواهد بود.

- در حالت دوم با قرار دادن $i=0$ از تعداد b ها یا c ها حداقل یکی کم می‌شود در حالی که از تعداد a ها کم نمی‌شود پس عنصر به دست آمده متعلق به L نخواهد بود.

لم پامپینگ زبان‌های خطی

فرض کنید L یک زبان روی حروف الفبای Σ باشد به طوری که برای هر عدد طبیعی n رشته‌ای چون $w \in L$ چنان موجود باشد که $|w| \geq n$ و برای هر تجزیه $w=uvxyz$ که $|uvyz| \leq n$ و $|vy| \geq 1$ بتوان یک $i \geq 0$ چنان یافت که $uv^ixy^iz \notin L$ آن گاه L خطی نیست.

مثال: نشان دهید زبان $L=\{w \in \{a,b\}^* \mid n_a(w)=n_b(w)\}$ خطی نیست.

فرض کنید حریف عدد طبیعی n را داده باشد، قرار می‌دهیم $w=a^nb^{2n}a^n$

به وضوح $|w|=4n \geq n$ ، $n_a(w)=n_b(w)=2n$

حال برای تجزیه دلخواه حریف $w=uvxyz$ که $|uvyz| \leq n$ و $|vy| \geq 1$ داریم:

$x=a^kb^{2n}a^{k'}$ در نتیجه $uvyz=a^{2n-(k+k')}$

با قرار دادن $i=0$ در رشته uv^ixy^iz حداقل یکی از تعداد a ها کاسته می‌شود در حالی که از تعداد b ها چیزی کم نمی‌شود پس $uv^0xy^0z \notin L$

خواص بستاری زبان های مستقل از متن

زبان های مستقل از متن تحت اجتماع ، concat ، ستار بستاره ای بسته هستند

درحالی که تحت اشتراک و مکمل گیری بسته نیستند.

* فرض کنید $G_1=\langle V_1,T_1,S_1,P_1\rangle$ و $G_2=\langle V_2,T_2,S_2,P_2\rangle$ دو گرامر مستقل از متن باشند که

$L(G_1)=L_1$, $L(G_2)=L_2$ بدون آن که از کلیت موضوع کم شود می توان فرض کرد که: $V_1\cap V_2=\emptyset$

حال تعریف می کنیم: $G^{\cup}=\langle V^{\cup},T^{\cup},S^{\cup},P^{\cup}\rangle$ که در آن

$V^{\cup}=V_1\cup V_2\cup\{S^{\cup}\}$ که $S^{\cup}\notin V_1\cup V_2$

$T^{\cup}=T_1\cup T_2$ ، $P^{\cup}=P_1\cup P_2\cup\{S^{\cup}\to S_1|S_2\}$

آن گاه:

$$L(G^{\cup})=L(G_1)\cup L(G_2)$$

همچنین تعریف می کنیم $\dot G=\langle \dot V,\dot T,\dot S,\dot P\rangle$ که در آن $\dot V=V_1\cup V_2\cup\{\dot S\}$ به طوری که

$\dot S\notin V_1\cup V_2$ ، $\dot T=T_1\cup T_2$ ، $\dot P=P_1\cup P_2\cup\{\dot S\to S_1S_2\}$

در این صورت:

$$L(\dot G)=L(G_1).L(G_2)=L_1.L_2$$

حال تعریف می کنیم $G_1^{*}=\langle V^{*},T_1,S^{*},P^{*}\rangle$ که در آن $V^{*}=V_1\cup\{S^{*}\}$

به طوری که $S^{*}\notin V_1$ ، $P^{*}=P_1\cup\{S^{*}\to S_1S^{*}|\lambda\}$

در این حالت:

$$L(G_1^{*})=(L(G_1))^{*}=L_1^{*}$$

۳۷۱

حال نشان می‌دهیم که خانواده‌ی زبان‌های مستقل از متن نسبت به اشتراک بسته نیستند.

برای این منظور تعریف می‌کنیم: $L_1=\{a^n b^n c^m \mid n,m\geq 0\}$, $L_2=\{a^m b^m c^n \mid n,m\geq 0\}$

L_1 و L_2 مستقل از متن هستند درحالی که اشتراک آن‌ها مستقل از متن نیست.

$$L_1 \cap L_2 = \{a^n b^n c^n \mid n\geq 0\}$$

نشان می‌دهیم که متمم زبان‌های مستقل از متن، لزوماً مستقل از متن نیستند.

فرض کنید چنین نباشد یعنی مکمل هر زبان مستقل از متن، مستقل از متن باشد. در نتیجه برای L_1 و L_2 زبان‌های تعریف شده در فوق آن‌گاه بایستی $\overline{L_1}$ و $\overline{L_2}$ مستقل از متن باشد و چون اجتماع دو زبان مستقل از متن، مستقل از متن است پس بایستی $\overline{L_1}\cup\overline{L_2}$ مستقل از متن باشد.

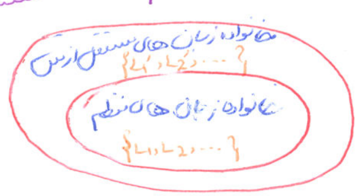
در این صورت مکمل زبان مذکور باید مستقل از متن باشد. پس $L_1 \cap L_2$ مستقل از متن است درحالی که مستقل از متن نیست.

کاربرد خواص بستاری در تعیین کردن مستقل از متن بودن یا نبودن زبان‌های داده شده

نکته: زبان‌های مستقل از متن تحت اشتراک با زبان‌های منظم بسته هستند یعنی اشتراک یک زبان مستقل از متن با یک زبان منظم مستقل از متن است.

مثال: نشان دهید زبان $L=\{w\in\{a,b,c\}^* \mid n_a(w)=n_b(w)=n_c(w)\}$ مستقل از متن نیست.

فرض کنید مستقل از متن باشد. آن‌گاه $\underbrace{L}_{\text{مستقل از متن}} \cap \underbrace{L(a^*b^*c^*)}_{\text{منظم}}=\{a^nb^nc^n \mid n\geq 0\}$ بایستی مستقل از متن باشد که این یک تناقض است.



نکته ۹۲: reverse یک زبان مستقل از متن، مستقل از متن است.

فرض کنید L مستقل از متن باشد درنتیجه گرامر مستقل از متنی چون $G=\langle V,T,S,P\rangle$ چنان موجود است که $L=L(G)$

تعریف می‌کنیم $G^r=\langle V,T,S,P^r\rangle$ که در آن برای هر قاعده $A\to\alpha\in P^r \Leftrightarrow A\to\alpha\in P$

در این صورت $L(G^r)=(L(G))^r=L^r$

$\{a^nb^n \mid n\geq 0\}$ $\quad S\to aSb\mid\lambda$

$\{b^na^n \mid n\geq 0\}$ $\quad S\to bSa\mid\lambda$

نکته: تصویر همریختی هر زبان مستقل از متن، مستقل از متن است.

چون L مستقل از متن است درنتیجه گرامر مستقل از متنی چون $G=\langle V,T,S,P\rangle$ چنان موجود است که $L=L(G)$

تعریف می‌کنیم $G^\phi=\langle V,T,S,P^\phi\rangle$ که در آن $\phi: T^*\to\Gamma^*$ یک همریختی است به طوری که

$$P^\phi=\{A\to\beta \mid A\to\alpha\in P\ ,\ \beta \text{ از جایگذینی هر a از رشته } \alpha \text{ به } \phi(a) \text{ به دست آمده است}\}$$

در این صورت $L(G^\phi)=\phi(L)$.

$\varphi(a)=c$

$\varphi(b)=ab$

$S\to aSb\mid\lambda \qquad L=\{a^nb^n \mid n\geq 0\}$

$$\varphi(L)=\{c^n(ab)^n \mid n\geq 0\}$$

$S\to\varphi(a)S\varphi(b)\mid\varphi(\lambda)$

$S\to cSab\mid\lambda$

۲۸۱

مثال:

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

$$\underbrace{\{w \in \{a,b\}^* \mid n_a(w) = n_b(w)\}}_{\text{مستقل از متن}} \cap \underbrace{L(a^*b^*)}_{\text{منظم}} = \{a^n b^n \mid n \geq 0\}$$

خواص بستاری زبان‌های خطی

- زبان‌های خطی تحت اجتماع بسته هستند.
- تصویر همریختی زبان‌های خطی، خطی است.
- معکوس زبان‌های خطی، خطی است.
- الحاق دو زبان خطی، خطی نیست.

مثال:

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

$L.L = \{a^n b^n a^m b^m \mid n,m \geq 0\} \Rightarrow$ خطی نیست

- بستار ستاره‌ای زبان‌های خطی، لزوماً خطی نیست.
- زبان‌های خطی تحت اشتراک و متمم‌گیری بسته نیستند.

مثال:

مستقل از متن

$L_1 = \{a^n b^n c^m \mid n,m \geq 0\} \Rightarrow$ خطی است

$$\begin{cases} S \to Sc \mid A \\ A \to aAb \mid \lambda \end{cases} \qquad S \stackrel{*}{\Rightarrow} Sc \stackrel{*}{\Rightarrow} Sc^m \stackrel{*}{\Rightarrow} Ac^m \stackrel{*}{\Rightarrow} a^n b^n c^m$$

$L_2 = \{a^m b^n c^n \mid n,m \geq 0\} \Rightarrow$ خطی است

$$\begin{cases} S_2 \to aS_2 \mid B \\ B \to bBc \mid \lambda \end{cases}$$

ذاتاً مبهم

$L_1 \cap L_2 = \{a^n b^n c^n \mid n,m \geq 0\} \Rightarrow$ مستقل از متن نیست

- اشتراک دو زبان خطی ممکن است حتی ذاتاً مبهم باشد.

- هر خاصیتی که برای اشتراک یا اجتماع برقرار نشد، برای متمم هم برقرار نیست.

نکته: الحاق یک زبان خطی به یک زبان منظم خطی است.

* هرگاه L_1 یک زبان خطی، L_2 یک زبان منظم باشد آن‌گاه $L_1 . L_2$ خطی است. *

مثال:

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

اثبات: چون L_1 یک زبان خطی است پس برای L_1 یک گرامر خطی وجود دارد که آن را می‌پذیرد. فرض کنید متغیر پایانه‌ی شروع این گرامر A باشد. هم‌چنین چون L_2 یک زبان منظم است، پس برای آن یک گرامر خطی چپ وجود دارد که آن را می‌پذیرد.

$L_1 = \{a^n b^n \mid n \geq 1\}$ $L_2 = \{c^m \mid m \geq 1\}$

$A \rightarrow aAb \mid ab$ $S \rightarrow Sc \mid c$

$S \rightarrow Sc \mid Ac$

$A \rightarrow aAb \mid ab$

$B \rightarrow \alpha^{\in T^*} \Rightarrow B \rightarrow A\alpha$

نکته: concat یک زبان منظم در یک زبان خطی، زبانی خطی است.

خواص تصمیم‌پذیری زبان‌های مستقل از متن

نکته: هرگاه G یک گرامر مستقل از متن باشد، الگوریتمی وجود دارد که بررسی می‌کند $L(G)$ تهی است یا خیر.

اثبات: گرامر G را ساده، حذف قواعد λ و نامفید می‌کنیم. اگر متغیر پایانه‌ی شروع (S) نامفید باشد یعنی $L(G)$ تهی است.

۲۹۱

نکته: اگر G یک گرامر مستقل از متن باشد، الگوریتمی وجود دارد که تعیین می‌کند $L(G)$ نامتناهی است یا خیر.

بدون آن که از کلیت موضوع کم شود می‌توان فرض کرد که گرامر G فاقد تولید λ، یکه و نامفید است.
یعنی یک گرامر غیر انقباضی است.

در این صورت برای هر متغیر پایانه مانند A با مقدار متناهی جایگذاری مشخص می‌شود که آیا A دارای مشتقات زیر هست یا خیر.

$A \xRightarrow{*} uAv$

اگر چنین اتفاقی رخ دهد، $L(G)$ نامتناهی است. در غیر این صورت متناهی است.

نکته: اگر G یک گرامر مستقل از متن باشد و $w \in T^*$ آنگاه می‌توان گفت که $w \in L(G)$ یا $w \notin L(G)$

چون G مستقل از متن است پس می‌توان برای آن یک گرامر چامسکی ارائه کرد. مسئله‌ی تعلق یک رشته به گرامر G یک مسئله‌ی تصمیم‌پذیر است.

نکته: فرض کنید G یک گرامر مستقل از متن باشد. مسئله‌ی تعلق یا عدم تعلق حداقل یک رشته به طول n به زبان $L(G)$ یک مسئله تصمیم‌پذیر است. (Reduction)

تمام رشته‌های به طول n روی حروف الفبای آن برابر است با T^n (چون متناهی است پس منظم است)

$$\underbrace{L(G_1)}_{\text{مستقل از متن}} = \underbrace{L(G)}_{\text{مستقل از متن}} \cap \underbrace{T^n}_{\text{منظم}}$$

اگر $L(G_1)=\emptyset$ باشد پس ندارد.

اگر $L(G_1)\neq\emptyset$ پس وجود دارد.

نکته: مسئله‌ی تساوی دو زبان مستقل از متن undecidable است.

نکته: فرض کنید L_1 منظم، L_2 مستقل از متن که به ترتیب توسط گرامرهای منظم G_1 و G_2 پذیرفته می‌شوند. مسئله‌ی $L(G_2) \subseteq L(G_1)$ یک مسئله‌ی تصمیم‌پذیر است.

$L(G_2) - L(G_1) = \emptyset$ $\qquad \underbrace{L(G_2)}_{\text{مستقل از متن}} \cap \underbrace{\overline{L(G_1)}}_{\text{منظم}}$

* اما مسئله $L(G_1) \subseteq L(G_2)$ تصمیم پذیر نیست.

نکته: هرگاه مجموعه حروف الفبای یک تک حرفی باشد، مفهوم مستقل از متن بودن با منظم بودن یکی است.

جلسه‌ی هشتم

ماشین‌های پشته‌ای NPDA

ماشین پشته‌ای نامعین (غیرقطعی) Nondeterministic Push Down Automata

یک NPDA عبارت است از هفت تایی مرتب $M = \langle Q, \Sigma, \Gamma, \delta, q_0, z, F \rangle$ که در آن

الف. Q مجموعه‌ای متناهی و ناتهی از وضعیت‌های واحد کنترل است.

ب. Σ مجموعه‌ای از حروف الفبای نوار است (متناهی و ناتهی)

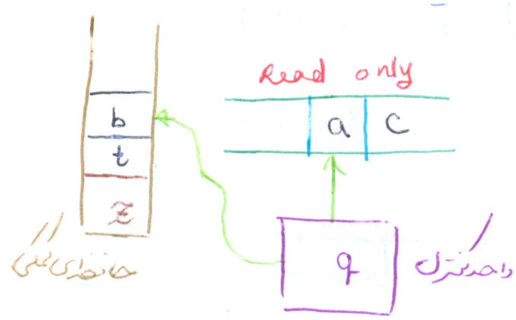
ج. Γ مجموعه‌ای متناهی و ناتهی از علائم پشته است که به آن گاهی الفبای پشته نیز گفته می‌شود

د. تابع δ از $\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to \{A \subseteq 2^{Q \times \Gamma^*} \mid |A| < \infty\}$ تابع انتقال است.

ه. $q_0 \in Q$ وضعیت آغازین است

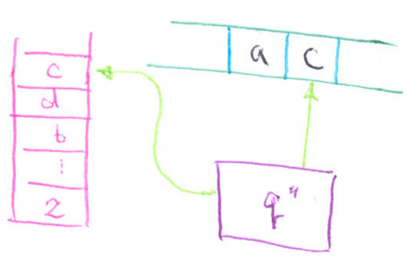
و. $z \in \Gamma$ علامت شروع پشته است

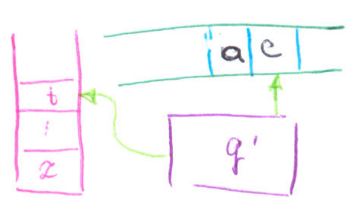
ز. $F \subseteq Q$ مجموعه وضعیت‌های نهایی است.



$$\delta(q, a, b) = \{(q', \lambda), (q'', cd)\}$$

هد همیشه به سمت راست یا نوار به سمت چپ حرکت می‌کند.

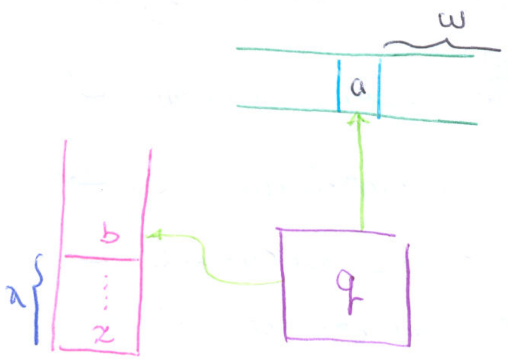




* اگر بخواهیم رشته‌ی $L = \{a^n b^n \mid n \geq 0\}$ را بپذیریم.

وقتی a می‌خوانیم، a را در پشته بریزیم، به ازای هر a خوانده شده هر جا b می‌خوانیم یک a بر می‌داریم از پشته.

configration پیکربندی



(q, aw, bx)

✓ خروجی δ دو تا است زیرا تغییراتی که می‌تواند اعمال کند فقط روی پشته و q است. روی نوار نمی‌تواند تغییری ایجاد کند.

تعریف حرکت

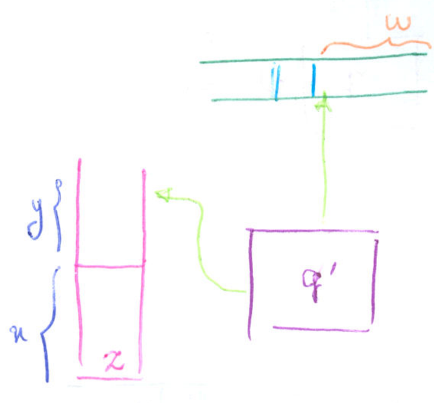
ماشین پشته‌ای غیرقطعی M از پیکربندی (q, aw, bx) به پیکربندی (q', w, yx) حرکت می‌کند و می‌نویسیم:

$$(q, aw, bx) \vdash (q', w, yx)$$

اگر و فقط اگر

$$(q', y) \in \delta(q, a, b)$$

باشد.



✓ همیشه یک حرف را برداشته و به جای آن رشته‌ای را جایگزین می‌کند.

$\vdash^*$

هیچ یا چند قدم

تعریف زبان پذیرفته شده توسط یک NPDA

* هرگاه $M=\langle Q,\Sigma,\Gamma,\delta,q_0,Z,F\rangle$ یک NPDA باشد زبان پذیرفته شده توسط M را با L(M) نمایش می‌دهیم و به شکل زیر تعریف می‌شود.

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

نکته: رشته‌ی پوچ توسط NPDA ای پذیرفته می‌شود که وضعیت شروع آن نهایی باشد.

$$(q_0,\lambda,z)\vdash^* (q_0,\lambda,z)$$

مثال: یک NPDA طراحی کنید که زبان $L=\{a^nb^n \mid n\geq 0\}$ را بپذیرد.

$\delta(q_0,a,z)=\{(q_0,az)\}$

$\delta(q_0,a,a)=\{(q_0,aa)\}$

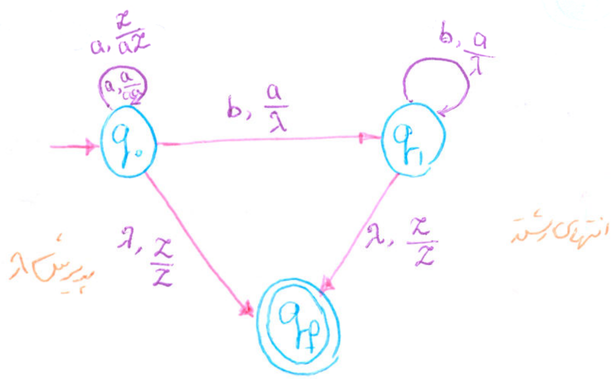
$\delta(q_0,b,a)=\{(q_1,\lambda)\}$ ⇝ وضعیت را عوض می‌کنیم تا دیگر اجازه ندهیم a دیگری بیاید.

$\delta(q_1,b,a)=\{(q_1,\lambda)\}$

$\delta(q_1,\lambda,z)=\{(q_f,z)\}$ ⇝ کلیه‌ی a ها push شده accept.

$\delta(q_0,\lambda,z)=\{(q_f,z)\}$ ⇝ λ را می‌پذیرد

ترسیم دیاگرام



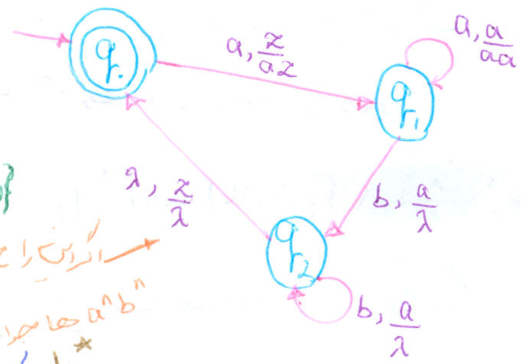
غیر قطعی

$\delta(q_0,\lambda,z)=\{(q_f,z)\}$

$\delta(q_0,a,z)=\{(q_0,az)\}$

۴۲۱

* در مثال قبل می‌خواهیم وضعیت q_0 را پایانی کنیم.



$\left\{\prod_{i=1}^{k}(a^n b^n) \mid n\in N\right\}$

اگر دیگر z را نگذاریم حاصلضرب $a^n b^n$ ها جواب خواهد بود.

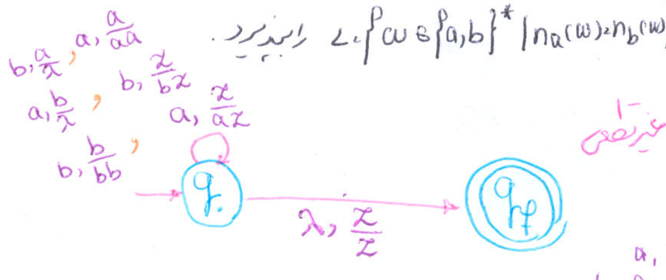
L^*

نکته تئوری: برای هر زبان مستقل از متن یک NPDA وجود دارد که آن را می‌پذیرد و بالعکس و به ازای هر زبان NPDA (پذیرفته شده توسط NPDA) یک گرامر مستقل از متن وجود دارد که L را می‌پذیرد. به عبارت دیگر خانواده‌ی زبان‌های پذیرفته شده توسط گرامرهای مستقل از متن دقیقاً خانواده‌ی زبان‌های پذیرفته شده توسط NPDA ها هستند.

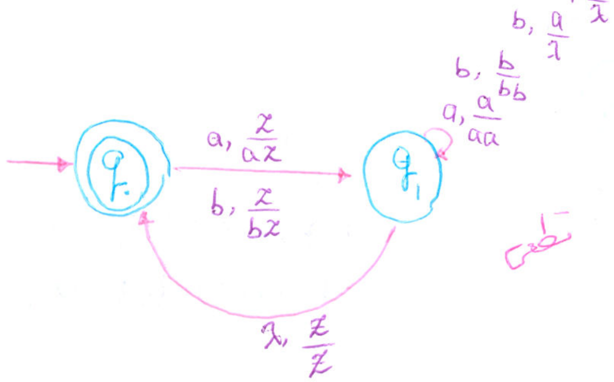
نکته: برای هر NPDA یک NPDA با سه وضعیت وجود دارد که زبان آن NPDA را بپذیرد.

نکته: اگر λ متعلق به زبان آن NPDA باشد، می‌توان این NPDA را با دو وضعیت ترسیم کرد.

مثال: یک NPDA ترسیم کنید که زبان $L=\{w\in\{a,b\}^* \mid n_a(w)=n_b(w)\}$ را بپذیرد.



✓ اگر λ متعلق به L بود می‌توان وضعیت شروع را پایانی کرد.



✓ زبان‌هایی که $L^*=L$ باشد

در این صورت باید بتواند دور بزند.

در این صورت می‌توان $\lambda, \frac{z}{z}$ گفت نه $\lambda, \frac{z}{\lambda}$. نباید پشته را خالی کنم.

مثال:

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

این زبان تصویر آینه‌ای است.

مبحث Back Tracking مربوط به ماشین است.

یکی از ویژگی‌های ماشین‌های غیرقطعی توانایی Back Tracking است.

نمی‌توان برای ww^r ماشین قطعی رسم کرد. حتماً غیرقطعی است.

زیرا بایستی نقطه‌ی وسط رشته را پیدا کنیم پس Back Tracking لازم دارد.

$$\delta(q_0, a, z) = \{(q_0, az)\}$$
$$\delta(q_0, b, z) = \{(q_0, bz)\}$$
$$\delta(q_0, a, a) = \{(q_0, aa)\}$$
$$\delta(q_0, b, b) = \{(q_0, bb)\}$$
$$\delta(q_0, a, b) = \{(q_0, ab)\}$$
$$\delta(q_0, b, a) = \{(q_0, ba)\}$$
$$\delta(q_0, \lambda, a) = \{(q_1, a)\}$$
$$\delta(q_0, \lambda, b) = \{(q_1, b)\}$$
$$\delta(q_1, b, b) = \{(q_1, \lambda)\}$$
$$\delta(q_1, a, a) = \{(q_1, \lambda)\}$$
$$\delta(q_1, \lambda, z) = \{(q_f, \lambda)\}$$
$$\delta(q_0, \lambda, z) = \{(q_f, z)\}$$

یک پیاده‌سازی بهتر

$$\delta(q_0, a, z) = \{(q_0, az)\}$$
$$\delta(q_0, b, z) = \{(q_0, bz)\}$$
$$\delta(q_0, a, a) = \{(q_0, aa), (q_1, \lambda)\}$$
$$\delta(q_0, b, b) = \{(q_0, bb), (q_1, \lambda)\}$$
$$\delta(q_0, a, b) = \{(q_0, ab)\}$$
$$\delta(q_0, b, a) = \{(q_0, ba)\}$$

$$\delta(q_1, a, a) = \{(q_1, \lambda)\}$$
$$\delta(q_1, b, b) = \{(q_1, \lambda)\}$$
$$\delta(q_1, \lambda, z) = \{(q_f, z)\}$$
$$\delta(q_0, \lambda, z) = \{(q_f, z)\}$$

۴۳۱

تعریف DPDA
Deterministic Push Down Automata

گوییم $M=\langle Q,\Sigma,\Gamma,\delta,q_0,z,F\rangle$ یک ماشین پشته‌ای معین (DPDA) است هرگاه

۱- M محدود یک NPDA باشد و به علاوه

۲- برای هر $a\in\Sigma$ ، $b\in\Gamma$ داشته باشیم

الف. $|\delta(q,a,b)|\leq 1$ ، $|\delta(q,\lambda,b)|\leq 1$

ب. $\delta(q,\lambda,b)\neq\emptyset \Rightarrow \forall c\in\Sigma \quad \delta(q,c,b)=\emptyset$

جلوی غیرقطعی شدن را می‌گیریم

تعریف زبان مستقل از متن معین

زبان L را مستقل از متن معین می‌نامیم هرگاه یک DPDA مانند M چنان وجود داشته باشد که $L=L(M)$

نتیجه: هر زبان مستقل از متن معین خود یک زبان مستقل از متن می‌باشد

زبان‌های مستقل از متنی وجود دارد که برای آن‌ها نمی‌توان هیچ DPDA ای ترسیم کرد.

مثال:

$$L=\{a^nb^n \mid n\geq 0\}\cup\{a^nb^{2n}\mid n\geq 0\}$$

$S\rightarrow S_1\mid S_2$

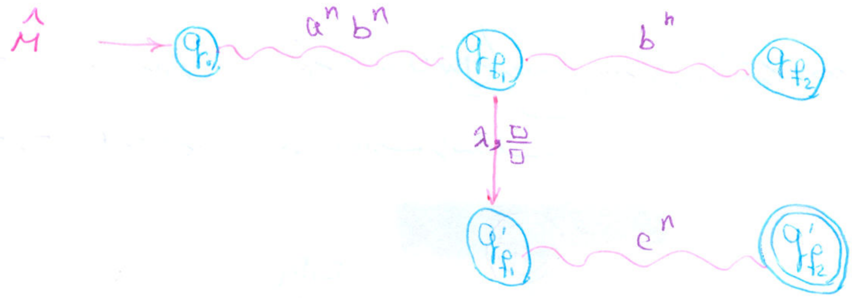
$S_1\rightarrow aS_1b\mid\lambda$

$S_2\rightarrow aS_2b^2\mid\lambda$

زبان L یک زبان مستقل از متن است پس یک NPDA وجود دارد که L را بپذیرد.

می‌خواهیم ثابت کنیم که هیچ DPDA ای وجود ندارد که L را بپذیرد.

اثبات: فرض کنید یک DPDA وجود داشته باشد که L را بپذیرد، در این صورت این DPDA را به NPDA زیر به شکل زیر تبدیل می‌کنیم.



$$L(\hat{M}) = \{a^n b^n c^n \mid n \geq 0\}$$

این زبان مستقل از متن نیست. پس به تناقض رسیده‌ایم.

در حالی که این زبان مستقل از متن نیست.

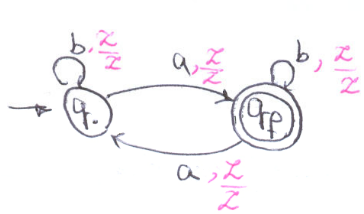
نکته: خانواده زبان‌های مستقل از متن قطعی تحت اجتماع بسته نیستند.

نکته: زبان‌های مستقل از متن ذاتاً مبهم قطعی نیستند زیرا برای پذیرش آن‌ها یک راه بیشتر نداریم.

نکته: هر زبان منظم یک زبان مستقل از متن قطعی می‌باشد زیرا زبان‌های منظم توسط DFAها پذیرفته می‌شوند و ساختار DFA را می‌توان به عنوان یک ساختار ماشین پشته‌ای در نظر گرفت که همواره مقدار Z را در هر انتقال به جای Z می‌برد.

مثال:

$$L = \{w \in \{a,b\}^* \mid n_a(w) \bmod 2 = 1\}$$



تبدیل به ماشین پشته‌ای

نکته: هرگاه Ψ را خانواده‌ی زبان‌های مستقل از متن قطعی در نظر بگیریم که توسط ماشین پشته‌ای قطعی با خالی شدن پشته رشته‌ها را می‌پذیرند و $\daleth$ را خانواده‌ی زبان‌هایی در نظر بگیرید که صرفاً با یک ماشین پشته‌ای قطعی پذیرفته می‌شوند.

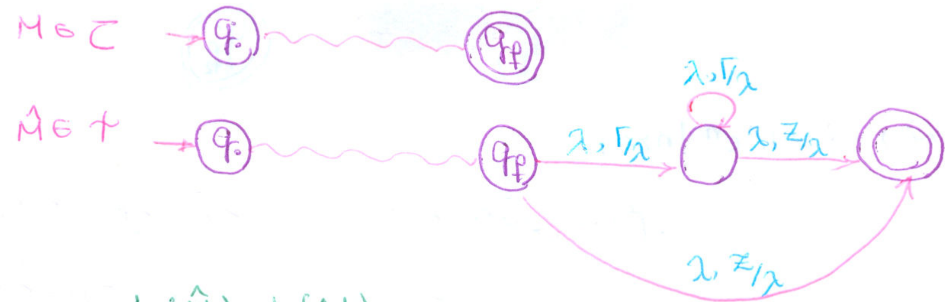
$$\Psi \subseteq \daleth$$

۴۴۱

نکته: حال فرض کنید ماشین پشته‌ای وجود دارد که زبان را می‌پذیرد، این ماشین متعلق به خانواده C است.

این ماشین را می‌توان به ماشین پشته‌ای تبدیل کرد که با خالی کردن پشته رشته‌ها را بپذیرد.

مطابق شکل زیر می‌توان نتیجه‌ی مطلوب را به دست آورد.



$L(\hat{M}) = L(M)$

نکته:

ماشین پشته‌ای معینی که با خالی شدن پشته رشته‌ها را می‌پذیرد زبان پذیرفته شده توسط این ماشین این ویژگی را دارد که هیچ رشته‌ای از آن پیشوند رشته‌ی دیگر از زبان آن ماشین نیست.

نتیجه: ماشین‌های پشته‌ای قطعی با خانواده‌ی زبان‌های ماشین‌های پشته‌ای قطعی با خانواده زبان‌های ماشین‌های پشته‌ای قطعی که با خالی شدن پشته رشته‌ها را می‌پذیرند یکسان نیستند.

مثال: $L(a^*) = \{\lambda, a, a^2, a^3, \dots\}$ یعنی اگر پذیرفت دیگر بقیه را نمی‌پذیرد.

نکته: زبان یک آتوماتا معین حتماً غیرمبهم است ولی ممکن است برای یک زبان غیرمبهم هیچ آتوماتای معینی موجود نباشد.

$\{a^n b^n \mid n \geq 0\} \cup \{a^n b^{2n} \mid n \geq 0\}$

جلسه‌ی نهم

ماشین های تورینگ

تعریف: گوییم $M = \langle Q, \Sigma, \Gamma, \delta, q_0, \square, F \rangle$ هفت تایی یک ماشین تورینگ (قطعی) است هرگاه

الف) Q مجموعه‌ای ناتهی و متناهی از وضعیت های واحد کنترل است.

ب) Σ مجموعه ای متناهی و ناتهی از حروف الفبا است.

ج) Γ مجموعه حروف الفبای نوار است که یک مجموعه متناهی و ناتهی است، به علاوه

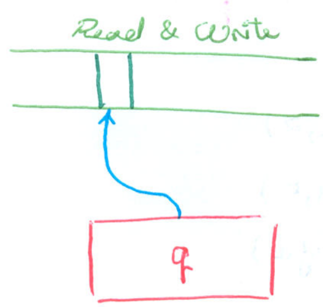
$$\Sigma \subseteq \Gamma - \{\square\}$$

د) تابع $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ یک تابع انتقال نامیده می‌شود.

ه) $q_0 \in Q$ وضعیت آغازین واحد کنترل است.

و) $\square \in \Gamma$ علامت خاصی است که نقش کننده‌ی جای خالی است.

ز) $F \subseteq Q$ مجموعه از وضعیت های پایانی است.



یک ماشین تورینگ استاندارد از عناصر زیر تشکیل شده است:

۱- یک نوار که از دو طرف نامحدود است و این نوار قابلیت خواندن و نوشته شدن بر روی آن وجود دارد.

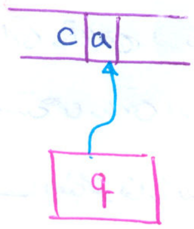
۲- یک هد خواندن و نوشتن که به هر میزان که بخواهد می‌تواند به چپ یا راست حرکت نماید

۳- واحد کنترل که با توجه به وضعیتی که آن واحد در آن قرار گرفته است و علامتی که از نوار توسط هد خوانده می‌شود، تصمیم می‌گیرد که هد به چه سمتی حرکت نماید و چه علامتی در نوار درج شود و وضعیت واحد کنترل

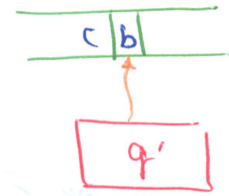
421

به چه وضعیت جدیدی انتقال یابد.

ماشین تورینگ استاندارد از هیچ حافظه‌ی کمکی برای ذخیره کردن و نگهداری مقادیر خروجی استفاده نمی‌کند و صرفاً همه‌ی عملیات روی نوار انجام می‌شود.

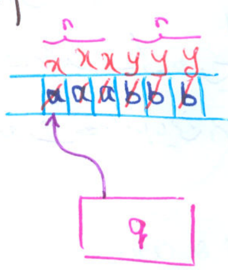


$$\delta(q,a) = (q',b,L)$$



تابع انتقال ماشین تورینگ یک تابع partial است یعنی ممکن است به ازای بعضی از ورودی‌ها تعریف نشده باشد که در این صورت می‌گوییم ماشین تورینگ به حالت توقف رفته است.

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

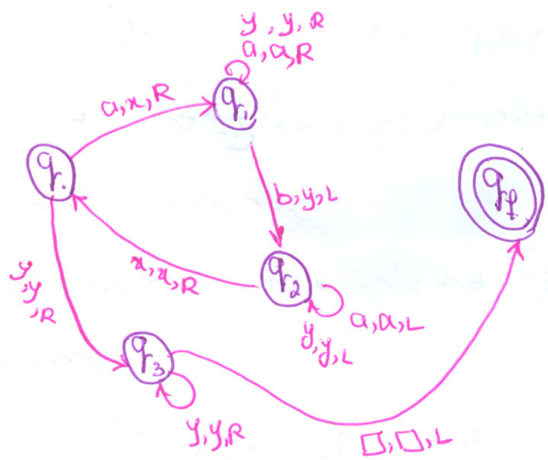


n بار طول n را حرکت می‌کند

$\Leftarrow \Theta(n^2)$

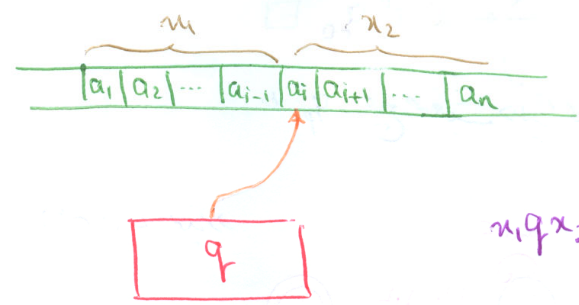
$$\delta(q_0,a) = (q_1,x,R)$$
$$\delta(q_1,a) = (q_1,a,R)$$
$$\delta(q_1,b) = (q_2,y,L)$$
$$\delta(q_2,a) = (q_2,a,L)$$
$$\delta(q_2,x) = (q_0,x,R)$$
$$\delta(q_1,y) = (q_1,y,R)$$
$$\delta(q_2,y) = (q_2,y,L)$$
$$\delta(q_0,y) = (q_3,y,R)$$
$$\delta(q_3,y) = (q_3,y,R)$$
$$\delta(q_3,\square) = (q_f,\square,L)$$

هر ماشین تورینگ را می‌توان با یک دیاگرام انتقال نمایش داد: (دقیقاً مشابه آن چیزی که در مورد NFA و DFA و ماشین پشته ای گفته شد)



تعریف پیکربندی یا توصیف

یک توصیف از ماشین تورینگ به شکل زیر را با $a_1a_2\cdots a_{i-1}\,q\,a_i a_{i+1}\cdots a_n$ نمایش می‌دهند:



$x_1 q x_2$

تعریف حرکت

یک حرکت از ماشین تورینگ می‌تواند حرکت به سمت راست یا چپ روی یک نوار باشد؛ به عبارت دیگر هد نوار به یک خانه‌ی چپ یا راست انتقال یابد.

گوییم ماشین تورینگ M از پیکربندی $a_1a_2\cdots a_{i-1}qa_ia_{i+1}\cdots a_n$ به پیکربندی $a_1a_2\cdots a_{i-1}bq'a_{i+1}\cdots a_n$ انتقال می‌یابد و می‌نویسیم:

$$a_1a_2\cdots a_{i-1}qa_ia_{i+1}\cdots a_n \vdash a_1a_2\cdots a_{i-1}bq'a_{i+1}\cdots a_n$$

هرگاه $\delta(q,a_i)=(q',b,R)$.

همچنین گوییم ماشین تورینگ M از پیکربندی $a_1a_2\cdots a_{i-1}qa_ia_{i+1}\cdots a_n$ به پیکربندی $a_1a_2\cdots a_{i-2}q''a_{i-1}ca_{i+1}\cdots a_n$ در صورتی که یک حرکت چپ بوده، انتقال یافته و می‌نویسیم:

431

$$a_1a_2\cdots a_{i-1}\, q\, a_i\, a_{i+1}\cdots a_n \vdash a_1a_2\cdots a_{i-2}\, q''\, a_{i-1}\, c\, a_{i+1}\cdots a_n$$

هرگاه $\delta(q,a_i)=(q'',c,L)$

✓ همچنین نماد $\overset{*}{\vdash}$ را به معنی هیچ یا یک یا چند حرکت که می‌تواند ترکیبی از حرکت‌های چپ و راست باشد.

تعریف زبان پذیرفته شده توسط ماشین تورینگ

زبان پذیرفته شده توسط ماشین تورینگ M را با $L(M)$ نمایش داده و $M=\langle Q,\Sigma,\Gamma,\delta,q_0,\square,F\rangle$ به شکل زیر تعریف می‌کنیم:

$$L(M)=\{w\in\Sigma^{+} \mid q_0 w \overset{*}{\vdash} x_1 q_f x_2,\ q_f\in F,\ x_1,x_2\in\Gamma^*\}$$

* قرارداد: رشته‌ی پوچ پذیرفته نمی‌شود.

در حالت final باید halt کند و دیگر به جلو نرود.

$$q_0 a^n b^n \overset{*}{\vdash} x^n y^n q_f \square$$

با q_f به هیچ وضعیت دیگری نمی‌رود

مثال: ماشین تورینگی طراحی کنید که $L(00^*)$ بپذیرد.

مثال:

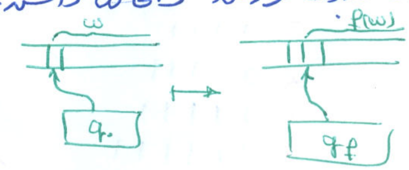
$\{a^nb^nc^n \mid n\geq 1\}$

$\Theta(n^2)$

تعریف محاسبه پذیر

گوییم تابع $f: D \rightarrow R$ محاسبه پذیر یا محاسبه پذیر تورینگ است هرگاه ماشین تورینگی چون M چنان وجود داشته باشد، که برای هر رشته $w \in D$ داشته باشیم:

$$q_0 w \xrightarrow[M]{*} q_f f(w)$$



مثال: نشان دهید جمع اعداد صحیح مثبت محاسبه پذیر تورینگ است.

برای نمایش x از $w(x) \in \{1\}^+$ استفاده می‌کنیم به طوری که $|w(x)| = x$

$$q_0\, w(x)0w(y) \xrightarrow[M]{*} q_f\, w(x+y)0$$

$\delta(q_0, 1) = (q_0, 1, R)$

$\delta(q_0, 0) = (q_1, 1, R)$

$\delta(q_1, 1) = (q_1, 1, R)$

$\delta(q_1, \square) = (q_2, \square, L)$

$\delta(q_2, 1) = (q_3, 0, L)$

$\delta(q_3, 1) = (q_3, 1, L)$

$\delta(q_3, \square) = (q_f, \square, R)$

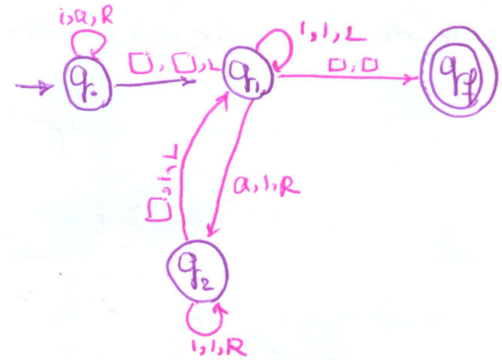
مثال: ماشین تورینگی طراحی کنید که نشان دهد $f: \{1\}^+ \rightarrow \{1\}^+$ که $f(w) = ww$ محاسبه پذیر است.

$$w \in \{1\}^+ \qquad q_0 w \xrightarrow{*} q_f ww$$

ابتدا تمام یک ها را در رشته‌ی w به کاراکتر مانند a تبدیل می‌کنیم، سپس به سمت راست ترین a رفته و یک کاراکتر را با یک جایگزین کرده، و به سمت راست رشته رفته و blank را به یک تبدیل می‌کنیم. این روند را آن قدر تکرار می‌کنیم تا تمام a ها به یک تبدیل شوند.

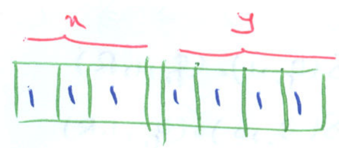
991

1 111
2 aaa□
3 aa1□
4 aa11
5 a111
6 a1111
7 11111
8 111111



طراحی ماشین های تورینگ برای عملیات پیچیده‌تر

$$f(x,y) = \begin{cases} q & x \le y \\ q' & x > y \end{cases}$$



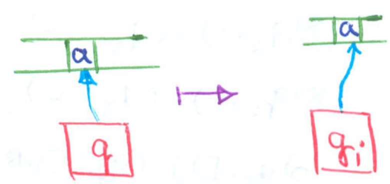
* if a then q_i else q_j

$\forall q \quad \delta(q, a) = (q'_i, a, R)$

$\forall c \in \Gamma \quad \delta(q'_i, c) = (q_i, c, L)$

$\forall b \neq a \quad \delta(q, b) = (q'_j, b, R)$

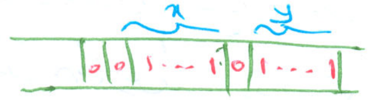
$\forall c \in \Gamma \quad \delta(q'_j, c) = (q_j, c, L)$



✓ در ابتدا به q'_i می‌رویم که اگر روی a بود دوباره جلو نرود و بتواند بازگردد.

مثال: می‌خواهیم نشان دهیم حاصل ضرب دو عدد صحیح مثبت محاسبه‌پذیر تورینگ است.

برای این منظور x و y را به شکل زیر روی نوار قرار دهیم.



سپس به ناحیه‌ی y می‌رویم یک 1 را به a تبدیل می‌کنیم. سپس به سمت چپ‌ترین ناحیه رفته صفر را به ox تبدیل می‌کنیم. این روند را آن قدر تکرار می‌کنیم تا هیچ یکی در ناحیه‌ی y باقی نماند. در این صورت

سمت چپ‌ترین عددی که قرار گرفته است xy است .

$3*2$

|0|0|111|0|11

0111 0|111|0|a1

0111111 0|111|0 aa

$2*3$

00|11|0|111

0 110 11 0 a11

0 1110 11 0 aa1

0111110 110 aaa

مرتبه : $x*y$

مثال :

$$f(x,y) = \begin{cases} x+y & x \le y \\ 0 & x > y \end{cases}$$

ابتدا با ماشین مقایسه، این را تست کرده

اگر $x \le y \Leftarrow$ x را با y جمع

اگر $x > y \Leftarrow$ صفر

با چند تا ماشین می‌توان پیاده‌سازی کرد

تعریف زبان بازگشتی برشمارشی recursively enumerable

زبان پذیرفته شده را بازگشتی برشمارشی گویند هرگاه توسط یک ماشین تورینگ پذیرفته شود.

تعریف زبان بازگشتی

زبان L را بازگشتی می نامیم هرگاه توسط یک ماشین تورینگ L پذیرفته شود، هرگاه این ماشین به ازای هر $w \in \Sigma^+$ توقف نماید.

نتیجه

هر زبان بازگشتی، بازگشتی برشمارشی نیز می باشد، ولی عکس آن لزوماً درست نمی باشد.

نکته

هرگاه L یک زبان بازگشتی باشد، مکمل آن نیز بازگشتی است. یعنی خانواده ی زبان های بازگشتی تحت عمل مکمل گیری بسته هستند.

نکته

خانواده ی زبان های بازگشتی تحت اجتماع، اشتراک، تصویر هم ریختی، عمل خارج قسمت، ستاره کلینی و الحاق بسته هستند.

نکته

خانواده ی زبان های بازگشتی برشمارشی تحت اجتماع، اشتراک، الحاق و ستاره + بسته هستند.

نکته

خانواده ی زبان های بازگشتی برشمارشی تحت تصویر هم ریختی بسته هستند اما تحت عمل مکمل گیری بسته نیستند.

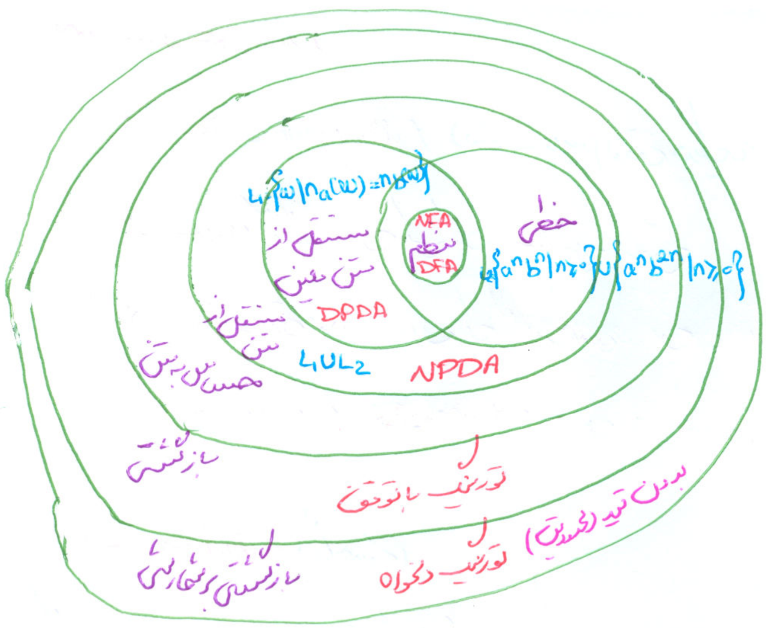
نکته

اگر مکمل یک زبان بازگشتی برشمارشی، بازگشتی برشمارشی باشد، خود آن زبان بازگشتی است.

نکته

زبان های حساس به متن تحت اشتراک، اجتماع، متمم، concat، ستاره، ستاره ای، خارج قسمت حتی تصویر هم ریختی که آن هم ریختی هیچ حرفی را به رشته ی پوچ نگاشت ندهد بسته هستند.

طبقه بندی چامسکی ؟



منظم = خطی چپ یا راست

تعریف مسئله ی تصمیم پذیر

مسئله ای که جواب آن بله یا نه باشد. به عبارت دیگر مسائل تصمیم پذیری که توسط یک ماشین تورینگ پذیرفته می‌شوند، جزء کلاس زبان های بازگشتی قرار می‌گیرند. یعنی الگوریتم قطعی دارند. به عبارت دیگر یک الگوریتم قطعی برای حل آن ها وجود دارد.

نتیجه: تعداد الگوریتم ها برای حل مسائل شمارا است درحالی که تعداد مسائل ناشمارا است.
تعداد ماشین های تورینگ شمارا است.

نکته: هر زبانی که بازگشتی بیان شود، حساس به متن است.

95/

مسئله تصمیم ناپذیر

۱- مسئله ی halting prob (توقف یا عدم توقف در یک ماشین تورینگ)

۲- مسئله ی تهی بودن یک گرامر مستقل از متن

۳- مسئله ی تهی بودن زبان یک گرامر بدون محدودیت یا زبان یک ماشین تورینگ

۴- مسئله ی تساوی دو زبان حساس به متن یا مستقل از متن یا بدون محدودیت

تعریف reduction (کاهش)

گوییم زبان L_1 به زبان L_2 کاهش پیدا می‌کند: $L_1 \leq L_2$

هرگاه: $L_1=\{x \mid \exists f \; f(x) \in L_2\}$

به طوریکه f محاسبه پذیر تورینگ است

به عبارت دیگر محاسبه پذیر تورینگ $x \in L_1 \Leftrightarrow f(x) \in L_2$

نتیجه: اگر L_2 تصمیم پذیر باشد و $L_1 \leq L_2$ باشد، آنگاه L_1 نیز تصمیم پذیر است.

نتیجه: اگر L_1 تصمیم ناپذیر باشد و $L_1 \leq L_2$ باشد، آنگاه L_2 نیز تصمیم ناپذیر است.

* تورینگ غیرقطعی در هر configuration چند عمل می‌تواند انجام دهد.

$\delta: Q \times \Gamma \to 2^{Q \times \Gamma \times \{L,R\}}$

LBA Linear Bounded Automata

ماشین های خطی کران دار است. یک ماشین تورینگ غیرقطعی است که محدوده ی پردازش رشته ها
آن برحسب طول رشته ورودی مشخص است.

$(q, [, R) \in \delta(q, [)$ $\quad (q'',], L) \in \delta(q,])$

LBA ها زبان های حساس به متن را می‌پذیرند.

نکته: هنوز ثابت نشده است که آیا LBAها قطعی و غیرقطعی با هم معادل اند یا نه.

نکته: ماشین های تورینگ قطعی و غیرقطعی با هم معادل هستند.

نکته: تفاوت ماشین های تورینگ با هم در سرعتشان است.

مجموعه سوالات کنکورهای کارشناسی ارشد دولتی

مهندسی کامپیوتر

۱۳۸۸ تا ۱۳۸۱   ۸۱/۸۸

نظریه زبانها و ماشینها

۵۵ـ زبان $L = \{a^n b^m c^n d^m , n, m \ge 0\}$ مفروض است. کدام گزینه صحیح است؟ ($\bar{L}$ زبان مکمل L است)

طول نوار محدود دارد ← حساس به متن ← بازگشتی (مکمل هم بازگشتی)

۱) ✓ L و $\bar{L}$ بازگشتی (Recursive) هستند.

۲) $\bar{L}$ شمارش‌پذیر بازگشتی (Recursively Enumerable) است ولی L بازگشتی نیست.

۳) $\bar{L}$ شمارش‌پذیر بازگشتی نیست ولی L بازگشتی هست.

۴) $\bar{L}$ شمارش‌پذیر بازگشتی نیست و L نیز بازگشتی نیست.

۵۶ـ فرض کنید یک محدودیت در یک ماشین تورینگ ایجاد کنیم، به‌طوری که همواره علامتی را که می‌نویسد با علامتی که می‌خواند متفاوت باشد، یعنی قواعد حرکت ماشین به یکی از دو صورت زیر باشد:

$\delta(q_i, a) = (q_j, b, L)\ , \ a \neq b$

$\delta(q_i, a) = (q_j, b, R)\ , a \neq b$

در این صورت محدودیت فوق چه تأثیری در قدرت ماشین دارد؟

۱) قدرت ماشین را کم می‌کند.

۲) ✓ تأثیری ندارد.   رو به جلو دوباره برگردد

۳) قدرت ماشین را زیاد می‌کند.

۴) اعمال این محدودیت با تعریف ماشین تورینگ مغایرت دارد.

۵۷ـ G_1 و G_2 دو گرامر مستقل از متن و G_3 یک گرامر منظم و رشته ω مفروضند. λ رشته‌ای به طول صفر است. برای کدام سؤال الگوریتم وجود ندارد؟

$\lambda \in G_1, G_2$

۱) آیا $L(G_1) = \phi$ ؟   ۲) آیا $\lambda \in L(G_1) \cap L(G_2)$ ؟

۳) $\omega \in L(G_1) \cap L(G_3)$ ؟   ۴) ✓ آیا $L(G_1) = L(G_2)$ ؟

مستقل از متن

مسئله تساوی دو گرامر مستقل از متن تصمیم ناپذیر است

۵۸ـ در مورد زبان $\{a^n b^n \mid n \ge 0\} \cup \{a^m c^m \mid m \ge 0\}$ کدام گزینه صحیح است؟

۱) زبان ذاتاً مبهم است.

۲) زبان حساس به متن است. ✓

۳) ✓ برای این زبان، گرامر مستقل از متن غیر مبهم وجود دارد. ✓

۴) برای این زبان، اتومات پوش دان (Push Down) قطعی وجود ندارد.

اگر طول نوار محدود ← منظم

طول نوار از یک طرف محدود ← فرقی نمی‌کند (نامحدود)

دو Track در نظر بگیریم

۹۲ـ در یک اتومات پوش‌دان (Push Down) طول پشته (Stack) حداکثر ۵ است. زبان‌هایی که این اتومات می‌تواند بپذیرد در کدام مجموعه زبان قرار می‌گیرد؟

۱) مجموعه تمام زبانهای منظم

۲) مجموعه تمام زبانهای حساس به متن که مستقل از متن نیستند.

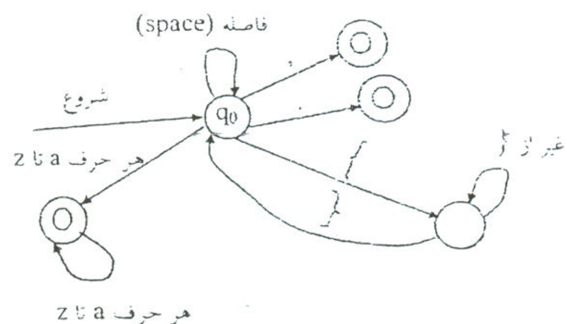
۳) مجموعه تمام زبانهای مستقل از متن که منظم نیستند.

۴) مجموعه تمام زبانهای مستقل از متن که طول سمت راست قواعد تولید گرامر آنها حداکثر ۵ است و منظم نیستند.

۹۳ـ می‌خواهیم قواعد تولید λ (λ - Production)، بی‌فایده (useless) و واحد (Unit) را از یک گرامر مستقل از متن که زبان آن فاقد λ است حذف کنیم. کدام ترتیب برای حذف درست است؟ (۴)

۱) واحد، λ و بی‌فایده ۲) بی‌فایده، λ و واحد ۳) λ، بی‌فایده و واحد ۴) λ، واحد و بی‌فایده

۹۴ـ یک برنامهٔ Scanner براساس اتومات متناهی زیر واژه‌های معتبر یک Word processor را تشخیص می‌دهد. معین کنید Scanner مزبور با دریافت متن زیر چند واژه را تشخیص می‌دهد؟



This is a comment {to be ignored}, in a sample text.

متن :

۱) ۸ ۲) ۱۰ ۳) ۱۱ ۴) ۱۸

۹۵ـ فرض کنید W^R معکوس رشته W و L_4 و L_5 دو زبان منظم دلخواه باشند. زبانهای L_1، L_2 و L_3 به شرح زیر مفروضند:

$L_1 = \{ww^R v \mid v, w \in \{a, b\}^*\}$
$L_2 = \{w_1 c w_2 \mid w_1, w_2 \in \{a,b\}^*, w_1 \neq w_2\}$
$L_3 = \{w \mid w \in L_4, w^R \in L_5\}$

کدام گزینه درست است؟

۱) L_1، L_2 و L_3 نامنظم‌اند. ۲) L_1، L_2 و L_3 هر سه منظم‌اند.

۳) L_1 و L_3 منظم ولی L_2 نامنظم است. ۴) L_2 و L_1 نامنظم‌اند، اما L_3 منظم است.

۹۶ ← گرامرهای بدون محدودیت
منظم
حساس به متن   مستقل از متن

۹۶ـ مجموعه متغیرهای V و پایانه‌های T مفروضند زبان L عبارت است از مجموعه تمام قواعد تولید گرامرهای مستقل از متنی که روی V و T تعریف می‌شوند، آنگاه L زبانی ....

$L = \{A \to \alpha \ (\alpha \in (V \cup T)^*,\ A \in V\}$

۱) منظم است. ✓   ۲) تصمیم‌ناپذیر است.

۳) مستقل از متن است ولی منظم نیست.   ۴) تصمیم‌پذیر است ولی مستقل از متن نیست.

$V \to \{A_1, A_2, \ldots, A_n\}$

$T \to \{a_1, a_2, \ldots, a_n\}$

۹۷ـ نوع زبان $L=\{a^n b^m \mid m \le n^2,\ n \le 1000\}$ کدام است؟

۱) منظم است. ✓   ۲) مستقل از متن است و منظم نیست.

۳) حساس به متن است و مستقل از متن نیست.   ۴) بدون محدودیت است و حساس به متن نیست.

$r = [A_1 \to (A_1 + \cdots + A_n + a_1 + \cdots + a_n)^*]$
$+ [A_2 \to (A_1 + \cdots + A_n + a_1 + \cdots + a_n)^*]$
$+ \cdots +$
$[A_n \to (A_1 + \cdots + A_n + a_1 + \cdots + a_n)^*]$

۹۸ـ فرض کنید G گرامری مستقل از متن باشد که الفبای آن تک نمادی است، آنگاه L(G) ....

۱) ممکن است نامنظم باشد.   ۲) ممکن است ذاتاً مبهم باشد.

۳) همواره زبانی منظم است. ✓   ۴) همواره زبانی نامتناهی است.

۹۹ـ فرض کنید G یک گرامر مستقل از متن به فرم نرمال چامسکی باشد که رشتهٔ مبهم w را تولید می‌کند، در این صورت کدام گزینه درست است؟

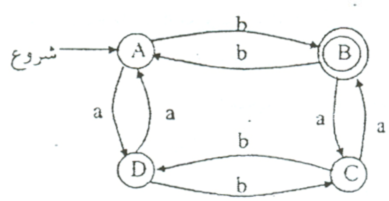
۱) حداکثر تعداد مراحل در اشتقاق‌های چپ برای w برابر |2|w است.

۲) تعداد مراحل در اشتقاق‌های چپ برای w ثابت است.

۳) حداقل تعداد مراحل در اشتقاق‌های چپ برای w برابر |2|w است.

۴) تعداد مراحل در دو اشتقاق چپ متفاوت برای w ممکن است متفاوت باشد.

۱۰۰ـ گرامر زبان اتومات متناهی قطعی (DFA) زیر کدام است؟ (λ رشته‌ای به طول صفر است و A متغیر شروع گرامر است.)



۱) $A \to aAa \mid B$
$B \to A \mid bBb \mid a \mid b$

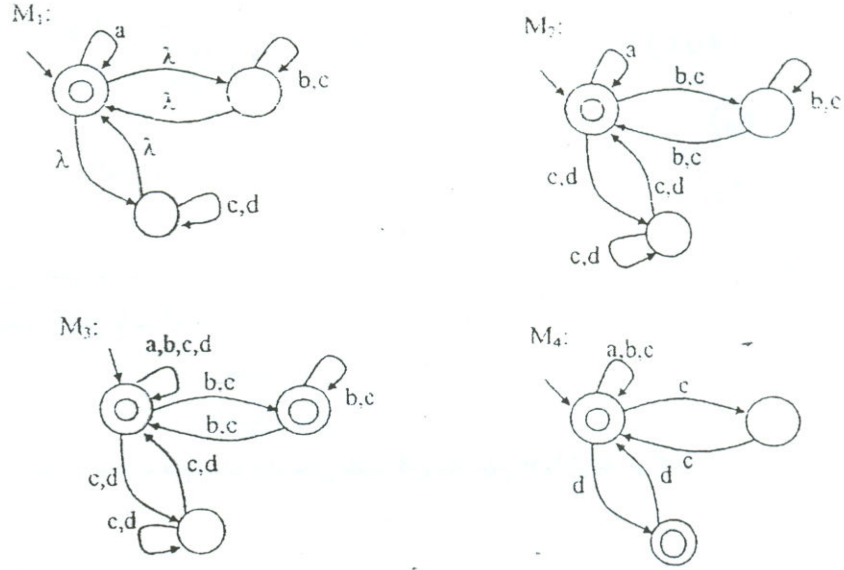
۲) $A \to aD \mid bB \quad B \to aC \mid bA$
$C \to aB \mid bD \quad D \to aA \mid bC$

۳) $a \to Da \mid Bb \quad B \to Ca \mid Ab \mid \lambda$
$C \to Ba \mid Db \quad D \to Aa \mid Cb$

۴) $A \to aD \mid bB \quad B \to aC \mid bA \mid a \mid b$
$C \to aB \mid bD \quad D \to aA \mid bC$

در سؤال‌های ۵٦ تا ٦۱ منظور از λ رشته‌ای به طول صفر است.

۵٦- اتومات‌های متناهی (Finite Automata) زیر را در نظر بگیرید:



کدام گزینه صحیح است؟

۱) $L(M_1) = L(M_3), L_{M4} \subset L(M_1)$　　۲) $L(M_2) = L(M_3), L(M_1) \subset L(M_2)$

۳) $L(M_1) \cap L(M_3) \neq \phi, L(M_4) \subset L(M_2)$　　٤) $L(M_2) \subset L(M_4), L(M_1) \subset L(M_3)$

۵۷- کدام یک از مسائل زیر تصمیم‌پذیر (decidable) است؟

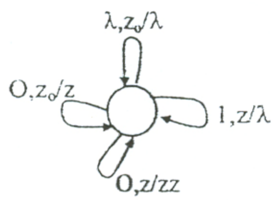
الف - زبان منظم R و عدد صحیح ثابت n مفروض است. آیا R دارای رشته‌ای به طول دقیقاً n است؟

ب - زبان مستقل از متن C و عدد صحیح ثابت n مفروض است. آیا C دارای رشته‌ای به طول دقیقاً n است؟

۱) ب تصمیم پذیر است.　　۲) الف تصمیم پذیر است.

۳) هر دو مسأله تصمیم‌پذیرند.　　٤) هیچیک از دو مسأله تصمیم پذیر نیستند.

۵۸- اتومات پوش دادن (Push Down Autonata):



$(Q, \Sigma, \Gamma, \delta, q_o, z_o)$

با

$Q = \{q_o\}, \Sigma = \{0,1\}, \Gamma = \{z_o, z\}$

چون final ندارد

مطابق شکل مفروض است. δ دارای ٤ حرکت است که در شکل نشان داده شده است. منظور از برچسب یال‌ها به شکل کلی x/γ و a این است که اتومات در ضمن تغییر حالت از حالت ابتدای پیکان به حالت انتهای پیکان، ورودی a را خوانده (در مورد $a = \lambda$ چیزی نمی‌خواند) و علامت X در بالای Stack را با رشته $\gamma \in \Gamma^*$ عوض می‌کند. زبان اتومات داده شده کدام است؟

۱) $\{\omega \in 0^*1^*$ | تعداد ۱ ها با تعداد صفرها (در ω) برابر است$\}$

۲) $\{\omega \in 0^*1^*$ | تعداد ۱ های ω دوبرابر تعداد صفرهای ω است$\}$

۳) $\{\omega \in (0+1)^*$ | تعداد ۱ های ω با تعداد صفرهای ω برابر است$\}$

٤) هیچکدام

۵۹- گرامر مستقل از متن G به شرح زیر مفروض است:

$S \to \lambda \mid AB$

$A \to S \mid 1A$

$B \to S \mid 0B$

زبان L(G) کدام کدام است؟

۱) $L(G) = (1^*0^*1^*)^*$ ۲) $L(G) = (1^*0^*)^* \cup \{\lambda\}$

۳) $L(G) = (1^*0 + 01^*)^*$ ۴) $L(G) = (1^*0^* + 0^*1^*) \cup \{\lambda\}$

۶۰- برای کدام یک از زبانهای زیر اتومات پوش دان معین (Deterministic) وجود ندارد؟

الف - $\{a^nb^n \mid n \geq 0\} \cup \{a^nb^{2n} \mid n \leq 100\}$

ب - $\{a^nb^{2n}c^n \mid n \geq 0\}$

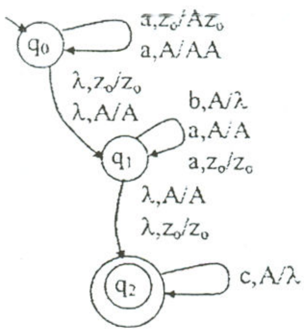
ج - $\{\omega \in (a+b)^* \mid$ طول ω زوج و نیمهٔ اول آن فقط شامل a باشد $\}$

د- $\{a^nb^nb^{m+n}c^m \mid n \geq 0, m \geq 0\} \cup \{a^nb^{2n} \mid n \geq 0\}$

۱) ب ۲) ج

۳) الف و د ۴) برای هر چهار تا زبان اتومات پوش دادن معین وجود دارد.

۶۱- زبان اتومات پوش دان M (مطابق شکل) کدام است؟



۱) $L(M) = a^nb^ka^*c^j \quad n \geq k+j \quad k,j \geq 0$

۲) $L(M) = a^n(b^k+a)^*c^j + a^* \quad n \geq k+j \quad k,j \geq 0$

۳) $L(M) = a^na^*(b+a)^kc^j \quad n \geq k+j \quad k,j \geq 0$

۴) $L(M) = a^n(a^*ba^*)^kc^j \quad n \geq k+j \quad k,j \geq 0$

۵۶ـ $L = \{a^m b^m \mid m \geq 0\}$ مفروض است. کدام گزینه غلط است؟

۱) $\overline{L}$ مستقل از متن است.

۲) $L^3 \cap L^4$ مستقل از متن است. ← λ مستقل از متن

۳) $L^* c$ مستقل از متن معین است.

۴) L^* توسط یک اتومات پوش‌دان معین در حالت خالی شدن Stack پذیرفته می‌شود. ✓

ab abab

۵۷ـ کدام گزینه در مورد زبان‌های مستقل از متن و اتومات‌های پوش‌دان صحیح است؟

زبان مستقل از متن :	Context Free Language
اتومات پوش‌دان :	Push Down Automata
معین :	Deterministic
غیر مبهم :	unambiguous

۱) زبان هر اتومات پوش‌دان معین را با حداقل یک گرامر مستقل از متن غیرمبهم می‌توان توصیف کرد. ✓

۲) برای هر اتومات پوش‌دان معین پذیرنده در حالت نهایی یک اتومات پوش‌دان معین پذیرنده در حالت خالی شدن Stack وجود دارد. غ

آن که خالی می‌کند با خودش معادل نیست

۳) هر زبان مستقل از متن که با حداقل یک گرامر غیرمبهم قابل توصیف باشد، توسط حداقل یک اتومات پوش‌دان معین پذیرفته می‌شود. غ

$\{a^n b^n \mid n \geq 0\} \cup \{a^n b^{2n} \mid n \geq 0\}$ غیرمبهم است ولی اتومات معین ندارد

عکسش درست است

۴) مجموعهٔ زبان‌هایی که برای آنها گرامر مستقل از متن غیرمبهم وجود دارد با مجموعهٔ زبان‌هایی که برای پذیرش آنها اتومات پوش‌دان معین وجود دارد برابر است.

۵۸ـ ماشینی که با دریافت یک گرامر دلخواه به فرم نرمال چامسکی و یک رشته دلخواه w از واژه‌های زبان گرامر، تعیین می‌کند که آیا w به زبان گرامر تعلق دارد یا خیر مفروض است. بهترین عملکرد زمانی ممکن است برای این ماشین بر حسب $|w|$ (طول رشته w) کدام است؟

۱) $O(|w|^2)$ ۲) $O(|w|^3)$ ✓ ۳) $O(2^{|w|})$ ۴) $O(\log|w|)$

غلط ←

۵۹ـ یک اتومات متناهی معین (DFA) که پذیرنده عبارت منظم زیر باشد و تعداد حالات آن حداقل باشد چند حالت دارد؟ ε نشانهٔ رشته‌ای به طول صفر است.

$(0 + 11^*01^*0)^*(\varepsilon + 11^*(\varepsilon + 01^*))$

۱) ۵ ۲) ۴ ۳) ۳ ۴) ۲

۶۰ـ کدام یک از زبان‌های زیر مستقل از متن است؟

۱) $L = \{a^{n^2} : n = 3k\}$ منظم نیست ۲) $L = \{a^{2^n} : n = 3k\}$ منظم نیست

۳) $L = \{a^n : n \geq 100$ یا n عدد اول$\}$ ✓ ۴) هیچ‌کدام

$L = \{a^2, a^3, \dots\} \cup \{a^{101}, a^{102}, a^{103}, \dots\}$

$= \{a^2, a^3, \dots, a^{97}\} \cup \{a^{101}, a^{102}, a^{103}, \dots\}$

۶۱ـ کدام یک از گزاره‌های زیر معادل‌اند؟

الف) ابهام در گرامرهای مستقل از متن یک مسأله تصمیم‌ناپذیر (Undecidable) است.

ب) حداقل یک مسأله تصمیم‌ناپذیر قابل کاهش (Reducable) به مسأله ابهام در گرامرهای مستقل از متن وجود دارد.

$L_1 \leq L_2$ اگر L_2 تصمیم‌ناپذیر باشد

ج) مسأله ابهام در گرامرهای مستقل از متن به حداقل یک مسأله تصمیم‌ناپذیر قابل کاهش است.

د) هیچ گرامر مستقل از متن وجود ندارد که بتوان ثابت کرد که مبهم است یا خیر.

۱) الف و ب ✓ ۲) الف و ج ۳) الف و د ۴) الف، ب و د

۵۶ـ کدام گزینه نادرست است؟

۱) مکمل یک زبان بازگشتی، یک زبان بازگشتی است.

۲) مکمل یک زبان بازگشتی بر شمردنی، بازگشتی است.

۳) مکمل یک زبان بازگشتی، یک زبان بازگشتی برشمردنی است.

۴) مکمل یک زبان بازگشتی برشمردنی، همیشه بازگشتی برشمردنی نیست.

۵۷ـ گرامر زیر چه زبانی را تولید می‌نماید؟ (λ نمایانگر رشته تهی است.)

$$
\begin{aligned}
G: S &\to S_1B \\
S_1 &\to aS_1b \\
bB &\to bbbB \\
aS_1b &\to aa \\
B &\to \lambda
\end{aligned}
$$

۱) $L(G)=\{a^{n+2}b^{3n} \mid n \ge 0\}$

۲) $L(G)=\{a^nb^{n+2k} \mid n \ge 2, k \ge 0\}$

۳) $L(G)=\{a^{n+2}b^{n+2k} \mid n \ge 0, k \ge 0\}$

۴) $L(G)=\{a^{n+1}b^{n+k} \mid n \ge 1, k \ge 0\}$

۵۸ـ کدام یک از زبان‌های زیر منظم است؟

$L_1=\{x^ny^n \mid x \in (0+1)^*, y \in (0+1)^*, n \ge 0\}$

$L_2=\{w \in L(A) \mid$ A یک DFA است و در مسیر پذیرش W از چند حالت معین A عبور نمی‌شود. $\}$

$L_3=\{w \in (0+1)^* \mid$ تعداد 0 ها و 1 ها برابر مقدار ثابت $n \ge 0$ باشد. $\}$

۱) L_3, L_1 ۲) L_3, L_2 ۳) L_3, L_2, L_1 ۴) هیچکدام منظم نیستند.

۵۹ـ در مورد انواع زبان‌های مستقل از متن کدام گزینه صحیح است؟

۱) زبان‌های مستقل از متن قطعی تحت عمل اجتماع بسته نیستند.

۲) زبان‌های مستقل از متن قطعی تحت عمل اشتراک بسته‌اند.

۳) زبان‌های مستقل از متن تحت عمل اشتراک با زبان‌های مستقل از متن قطعی بسته‌اند.

۴) زبان‌هایی که برای آن‌ها گرامر مستقل از متن مبهم وجود دارد تحت عمل اجتماع بسته نیستند.

۶۰ـ عمل بر زدن روی زبان‌های L_1 و L_2 به شرح زیر تعریف می‌شود:

$$S(L_1, L_2)=\{(wv)^* \mid w \in L_1, v \in L_2\}$$

$=(L_1, L_2)^*$

کدام گزاره صحیح است؟

۱) زبان‌های مستقل از متن تحت عمل برزدن (S) بسته نیستند.

۲) زبان‌های مستقل از متن تحت عمل برزدن (S) بسته هستند.

۳) زبان‌های مستقل از متن تحت عمل بر زدن (S) بسته نیستند ولی زبان‌های منظم تحت آن عمل بسته هستند.

۴) زبان‌های منظم تحت عمل برزدن (S) بسته نیستند ولی زبان‌های مستقل از متن تحت آن عمل بسته هستند.

۶۱ـ برای کدام یک از گروه زبان‌های زیر DPA قطعی (Deterministic Push Down Automata) که در حالت خالی شدن Stack می‌پذیرد وجود دارد؟

۱) تمام زبان‌های مستقل از متن قطعی

۲) تمام زبان‌های منظم محدود (یعنی تعداد رشته‌های زبان محدود است.)

۳) تمام زبان‌های مستقل از متنی قطعی که هیچ رشته‌ای از زبان پیشوند رشتهٔ دیگری از زبان نباشد.

۴) تمام زبان‌های منظمی که هیچ رشته‌ای از زبان پیشوند رشتهٔ دیگری از زبان نباشد.

که خود مستقل از متن هستند

۵٦ـ کدام یک از زبان‌های زیر نامنظم است؟

۱) $\{a^n b^n (a+b)^* \mid n \ge 0\}$	۲) $\{b^* a^n b^n a^* \mid n \ge 0\}$

۳) $\{a^* a^n b^n b^* \mid n \ge 0\}$	٤) هر سه نامنظم هستند.

۵۷ـ کدام یک از دلایل زیر برای این‌که نشان دهیم زبان L منظم نیست کافی است؟

۱) عدد ثابت مثل n وجود دارد به طوری که برای هر رشته $|z| \ge 1\ , z \in L$ دداشته باشیم:

$z = uvwxy\ ;\ |vx| \ne 0\ ,\ |vwx| \le n, \forall i \ge 0\, uv^i wx^i y \in L$

۲) عدد ثابت مثل n وجود دارد به طوری که برای هر رشته $|z| \ge n\ , z \in L$ داشته باشیم :

$z = xyw, |y| \ne 0, |xy| \le n\ ,\ \forall i \ge 0\ xy^i w \in L$

۳) هیچ عدد ثابت مثل n وجود ندارد به طوری که برای هر رشته $|z| \ge n, z \in L$ داشته باشیم :

$z = uvwxy\ ,\ |vx| \ne 0\ ,\ |vwx| \le n\ ,\ \forall i \ge 0\ uv^i wx^i \in L$

٤) هیچ‌کدام

۵۸ـ می‌گوییم زبان L ، Definite است اگر عدد k وجود داشته باشد که برای هر رشته w، تعلق آن به زبان تنها وابسته به آخرین k نماد، w باشد. کدام گزینه نادرست است؟

مثال از زبان definite : $(a+b)^*cde$ که در آن $k = 3$ است.

۱) زبان‌های Definite تحت عمل اجتماع بسته هستند.

۲) زبان‌های Definite تحت عمل مکمل‌گیری بسته هستند.

۳) هر زبان Definite با یک ماشین متناهی پذیرفته می‌شود.

٤) زبان‌های Definite تحت عمل *(Kleene star) بسته هستند.

۵۹ـ مجموعه‌های زیر را در نظر بگیرید :

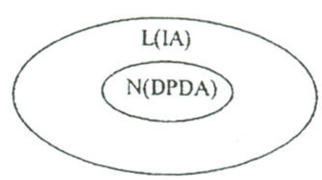
L(PDA) : مجموعه زبان‌هایی که برای آن‌ها PDA (Pushdown Automata) وجود دارد.

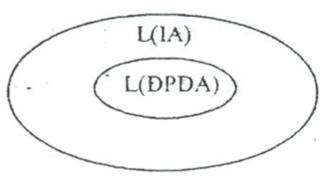
L(DPDA) : مجموعه زبان‌هایی که برای آن‌ها DPDA (Deterministic PDA) وجود دارد.

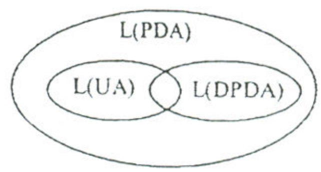
L(UA) : مجموعه زبان‌های مستقل از متن غیر مبهم (unambiguous context free)

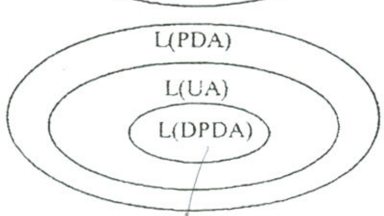
L(IA) : مجموعه زبان‌های مستقل از متن ذاتاً مبهم (Inherently Ambiguous)

کدام یک از نمودارهای زیر مجموعه‌ای زیر درست است؟

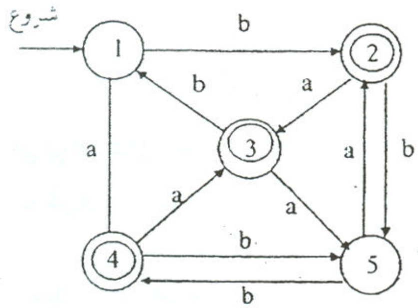
۱)


۲)


۳)


٤)


۶۰ـ اتومات متناهی زیر را درنظر می‌گیریم. اتومات کمینه (minimized) مربوطه دارای چند حالت خواهد بود؟



۱) ۳

۲) ۲

۳) ۵

۴) ٤

۶۱ـ $\delta(q,a)=(q',X,L)$ ، $\delta(q,a)=(q',X,R)$ قواعد نمونهٔ یک ماشین تورینگ می‌باشند که اگر ماشین در حالت q باشد و سر آن حرف a را روی نوار ببیند ماشین به حالت 'q رفته، حرف a با x عوض شده و سر ماشین به ترتیب به راست (R) و یا چپ (L) می‌رود. زبان ماشین تورینگ با قواعد زیر کدام است؟ q_4 حالت نهایی، B علامت جای خالی روی نوار و $\sum=\{a,b\}$ مجموعهٔ واژه‌های زبان است:

$\delta(q_0,a)=(q_1,X,R),\delta(q_0,y)=(q_3,y,R),\delta(q_1,a)=(q_1,a,R)$

$,\delta(q_1,y)=(q_1,y,R),\delta(q_1,b)=(q_2,y,L),$

$\delta(q_2,a)=(q_2,a,L),\delta(q_2,y)=(q_2,y,L),\delta(q_2,x)=(q_0,x,R)$

$,\delta(q_3,y)=(a_3,y,R),\delta(q_3,B)=(q_4,B,R)$

۱) $\{a^nb^n \mid n\geq 0\}$

۲) $\{a^nb^na^n \mid n\geq 1\}$

۳) { تعداد a ها با تعداد b ها برابر است | $w\in(a+b)^+\}$

$\{a^nb^n \mid n\geq 1\}$

۴) هیچکدام ✓

هیچگاه a^0b^0 $n\geq 0$ درست نیست. مساوی صفر توجه [illegible]

۵۶ـ $L=\{a^m cb^n : m \neq n\} \cup \{a^m db^{2m} : m \geq 0\}$ ، کدام گزینه نادرست است؟

۱) هر همومرفیسم L با یک PDA معین شناسایی می‌شود.

۲) یک گرامر غیرمبهم برای زبان L موجود است.

۳) یک PDA نامعین برای شناسایی L موجود است.

۴) همه موارد

۵۷ـ اگر $\sum = \{a,b,c\}$ و $L - \sum^* = \{a,b,c\}$ باشد آن‌گاه L کدام یک از زبان‌های زیر می‌تواند باشد؟

$\varepsilon - IV$, $\varphi - III$, $a^n b^{n^2} c^n - II$, $\sum^* - I$

۱) فقط I   ۲) فقط IV   ۳) فقط I و III   ۴) I، II، III و IV

۵۸ـ ثابت Pumping Lemma برای زبان‌های مستقل از متن با گرامر $G=(S, V, T, P)$ کدام است؟

۱) تعداد واژه‌های زبان در T (Terminals)   ۲) تعداد واژه‌های نحوی در V (Nonterminals)

۳) تعداد قواعد تولید در P (Production rules)   ۴) هیچکدام

۵۹ـ برای تشخیص زبان $L=\{a^n b^n \mid n \geq 0\}$ یک ماشین تورینگ ساخته‌ایم. حداقل هزینه تشخیص $w \in L$ با این ماشین تورینگ در چه حدی است؟

۱) $O(n)$ ۲) $O(n^2)$ ✓   ۳) $O(n^3)$ ۴) $O(2^n)$

S → xSv | A

A → BC

C → wC v | λ

B → Dz²   D → yDz | λ

۶۰ـ زبان L با تعریف زیر مفروض است. کدام یک از گزاره‌ها غلط است؟

$L=\{x^i y^j z^{j+2} w^k v^{i+k} \mid i,j,k \geq 0\}$ DPDA پذیرنده → هست یک راه

۱) یک اتاماتای پشته‌ای غیر قطعی مثل A وجود دارد به قسمتی که $L=L(A)$ ✓

۲) رشته‌های L توسط یک اتامای قطعی کراندار (Linear Bounded Automata) قابل شناسایی هستند. زبان دارد ← هستی برخی ← LBA پذیرنده ✓

۳) ✓ زبان L از نوع مستقل از متن معین (DCFL) نمی باشد.

۴) زبان L از نوع بازگشتی شمارش‌پذیر است. ✓

۶۱ـ زبان گرامر G کدام است؟

$$
\begin{aligned}
G: S &\to aAb \mid bBa \mid bCa \\
A &\to aaAb \mid ab \\
B &\to bBa \mid a \\
C &\to aC \mid bC
\end{aligned}
$$

۱) $a^{2k+2} b^{k+1} \cup b^+ a^+ \quad k \geq 0$ ۲) $a^{2k} b^k \cup (ba)^* a \quad k \geq 1$

۳) $a^{k+1} b^k \cup b^l a^l \quad l \geq 1, k \geq 2$ ۳) $a^2 a^{2k} b^k b^2 \cup b^l a^{l+1} \quad k \geq 0, l \geq 1$

۵۸ـ گرامر G و زبان‌های L_1 و L_2 مفروضند. ارتباط $L(G)$ با L_1 و L_2 کدام است؟ ε نشانه‌ی رشته‌ای به طول صفر است.

$L_1 = \{w \in (a+b)^* \mid$ تعداد a های w با b های w برابر است$\}$

$L_2 = \{w \in (a+b)^* \mid$ تعداد ab های w با ba های w برابر است$\}$

$S \to Sab$
$S \to Sba$
$S \to aSb$
$S \to bSa$
$S \to abS$
$S \to baS$
$S \to \varepsilon$

۱) $L(G) \subset L_1$ ۲) $L(G) = L_1$ ۳) $L(G) = L_1 \cup L_2$ ۴) $L(G) \supset L_2$

۵۹ـ گرامر وابسته به متن G به شرح زیر مفروض است. کدام یک از مجموعه رشته‌های ۱ تا ۴، زیرمجموعه $L(G)$ است؟

$S \to ACab$
$Ca \to aaC$
$CB \to DB$
$CB \to E$
$aD \to Da$
$AD \to AC$
$aE \to Ea$
$AE \to a$

۱) $\{aa, aaaa\}$ ۲) $\{aaa, aaaaa\}$ ۳) $\{a, aaa, aaaaa\}$ ۴) $\{aaaa, aaaaaa\}$

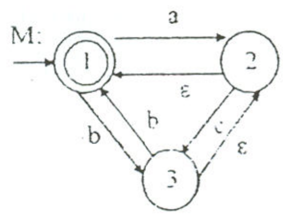
۶۰ـ گرامر G به شرح زیر مفروض است. $L(G)$ کدام است؟ w^R عبارت است از w که از آخر به اول خوانده شود. و ε نشانهٔ رشته های به طول صفر است.

G:
$S \to aA$
$S \to bB$
$S \to \varepsilon$
$A \to Sa$
$A \to \varepsilon$
$B \to Sb$
$B \to \varepsilon$

۱) $(a+b)^*$ ۲) $\{w \in (a+b)^* \mid w = w^R\}$

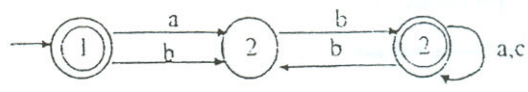
۳) $\{w(a+b)w^R \mid w \in (a+b)^*\}$ ۴) $\{ww^R \mid w \in (a+b)^*\}$

۶۱ـ ماشین متناهی M به شکل زیر مفروض است گزاره صحیح کدام است؟ ε نشانهٔ رشته‌ای به طول صفر است.



۱) $L(M) = \left(a^* \mid (b \mid ac)^* (b \mid \varepsilon)\right)^*$

۲) ماشین قطعی زیر معادل M است.



۳) $L(M) = \{w \in (a \mid b \mid c)^* \mid$ w با c شروع نمی‌شود $\}$

۴) زبان گرامر مقابل همان L(M) است.

$S \to aS \mid bS \mid acS \mid bA \mid acA$

$A \to cA \mid b \mid \varepsilon$

۶۲ـ زبان‌های زیر با $\beta \in \sum^+, \alpha, \gamma \in \sum^*$ مفروضند. کدام گزینه صحیح است؟

$L_1 = \left\{\alpha^i (\alpha\beta)^j (\gamma_\alpha)^i \quad j \geq 0 \ , i \geq 0\right\}$

$L_2 = \left\{\alpha^i (\alpha\beta)^j (\gamma_\alpha)^{i+j} \quad j \geq 0 \ , i \geq 1\right\}$

$L_3 = \left\{\alpha^i (\alpha\beta)^j (\gamma_\alpha)^i \quad j \geq 1 \ , i \geq 1\right\}$

۱) L_3 و L_1 هر دو منظم هستند.　　۲) L_1 منظم و L_3 نامنظم است.

۳) L_1 منظم و L_2 نامنظم است.　　۴) L_1 و L_2 و L_3 همگی نامنظم هستند.

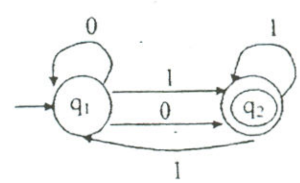
۶۳ـ اتومات متناهی M و زبان‌های L_1 تا L_4 مفروضند. رابطهٔ $L(M)$ با L_1 تا L_4 کدام است؟

$L_1 = (0+1)(0+1)^*$

$L_2 = \left(0+(0+1)1^*1\right)^*(0+1)1^*$

$L_3 = 0^*(0+1)1^*\left(10^*(0+1)1^*\right)^*$

$L_4 = (0+110)(0+1)^*$



۱) $L(M) = L_2 = L_3 = L_4$　　۲) $L(M) = L_1 = L_2 = L_3$

۳) $L(M) = L_2 = L_3$　　۴) $L(M) = L_4$

۵۸- عبارت منظم R و گرامرهای G_1، G_2 و G_3 با تعریف زیر مفروضند. اگر زبان R را L بنامیم و L_1، L_2 و L_3 به ترتیب زبان گرامرهای مذکور باشند، کدام گزاره صحیح است؟

$R = ((aa \mid b)^* b)^* a$

$G_1 : S \to bS \mid aA \mid aC$, $A \to aS$, $C \to \varepsilon$

$G_2 : S \to bS \mid aA \mid aC$, $A \to Sa$, $C \to \varepsilon$

$G_3 : S \to bS \mid Aa \mid C$, $A \to aS$, $C \to a$

۱) $L = L_1 = L_2 = L_3$ ۲) $L_1 \neq L_3 \;,\; L = L_1$ ۳) $L_2 \neq L \;,\; L = L_1 = L_3$ ۴) $L_3 \neq L_2 \;,\; L = L_1 = L_2$

۵۹- زبان‌های منظم L_1، L_2، L_3 و L_4 مفروضند:

$$L_1 = L(a^*)$$
$$L_2 = L((a+b)^*)$$

$L_3 = \{w \in (a+b)^* \mid$ تعداد bهای w زوج باشد.$\}$

$L_4 = \{w \in (a+b)^* \mid$ تعداد bهای w زوج و تعداد aهای آن فرد باشد.$\}$

برای چند زبان از این ۴ زبان می‌توان ماشین پشته‌ای (PDA) با حداکثر ۲ حالت ساخت؟

۱) 1 ۲) 2 ۳) 3 ۴) 4

۶۰- گرامر G را در نظر می‌گیریم و زبان آن را L می‌نامیم. رشته‌های w_1 و w_2 با تعریف زیر را نیز در نظر می‌گیریم. کدام گزاره صحیح است؟

$$G : S \to aSD \mid bB$$
$$D \to dS \mid a$$
$$B \to bB \mid \varepsilon$$
$$w_1 = a^{10} b a^{7} b d b^{10} d$$
$$w_2 = a^{10} b^{9} a^{10} d$$

۱) $w_1, w_2 \in L$ ۲) $w_1, w_2 \notin L$ ۳) $w_2 \notin L \;,\; w_1 \in L$ ۴) $w_1 \notin L \;,\; w_2 \in L$

۶۱- اگر $M = (Q, Q_0, \Sigma, F, \delta)$ یک اتومات متناهی باشد تعریف می‌کنیم: $\overline{M} = (Q, Q_0, \Sigma, Q-F, \delta)$ همچنین $d(M)$ اتومات قطعی معادل M خواهد بود. اگر M_1 و M_2 دو اتومات متناهی باشند $M_1 + M_2$ اتومات متناهی است که زبان آن اجتماع زبان‌های M_1 و M_2 است. فرض کنید G_1 و G_2 دو گرامر منظم باشند که زبان آن‌ها به ترتیب معادل زبان‌های M_1 و M_2 هستند. کدام عبارت زیر صحیح است؟

d(M۲) هم می‌توان نوشت

برای متمم گیری باید حتماً [illegible]

۱) $L(G_1) - L(G_2) = L(\overline{\overline{M_1} + M_2})$ ۲) $L(G_1) - L(G_2) = L(d(\overline{d(\overline{M_1}) + M_2}))$

۳) $L(G_1) - L(G_2) = L(\overline{\overline{d(M_1)} + d(M_2)})$ ۴) $L(G_1) - L(G_2) = L(\overline{d(\overline{M_1}) + d(M_2)})$

۶۲- زبان L مجموعه تمامی زوج‌های مرتب $\langle M, W \rangle$ است که در آن M که یک ماشین تورینگ و w یک رشته است به طوری که ماشین M بر ورودی w متوقف نمی‌شود. کدام یک از جملات زیر صحیح است؟

الف) L بازگشتی است.

ب) L به طور بازگشتی شمار است.

ج) L بازگشتی نیست.

د) L به طور بازگشتی شمارا نیست.

۱) ب ۲) الف و ب ۳) ب و ج ۴) ج و د

۶۳- ماشین تورینگ M با دستورات حرکت زیر مفروض است که در آن q_0 حالت شروع، q_f حالت پایانی و B علامت خانه‌های خالی دو طرف نوار است. منظور از $\delta(q,a)=(P,X,R)$ این است که اگر M در حالت q و سر آن مقابل حرف a روی نوار باشد آنگاه به حالت P رفته، a را با X عوض کرده و سر را به اندازه‌ی یک خانه به راست می‌برد (اگر به جای R، L باشد آنگاه به چپ می‌رود.) اگر در شروع کار M (یعنی حالت q_0 و سر در ابتدای ورودی روی نوار) محتوی نوار برابر رشته‌ی aaabbb باشد پس از دقیقاً 11 حرکت δ محتوی نوار کدام است؟

$$\delta(q_0,a)=(q_1,X,R)$$
$$\delta(q_1,a)=(q_1,a,R)$$
$$\delta(q_1,b)=(q_2,Y,L)$$
$$\delta(q_2,a)=(q_2,a,L)$$
$$\delta(q_2,X)=(q_1,X,R)$$
$$\delta(q_0,B)=(q_f,B,R)$$
$$\delta(q_1,Y)=(q_1,Y,R)$$
$$\delta(q_2,Y)=(q_2,Y,L)$$
$$\delta(q_1,B)=(q_f,B,R)$$

xa yy
aaabbb

۱) XaaYYb ۲) XXaYYb ۳) XXaYbb ۴) XXXYYY

(0) q_0 aaabbb

(1) x$q_1$aabbb

(2) xa$q_1$abbb

(3) xaa$q_1$bbb

(4) xa$q_2$aybb

(5) x$q_2$aaybb

(6) $q_2$xaaybb

(7) x$q_1$aaybb

(8) xa$q_1$aybb

(9) xaa$q_1$ybb

(10) xaay$q_1$bb

(11) xaa$q_2$yyb