نظریه زبانها و ماشین ها

دکتر سید جواد حاج سید جوادی

جلسه‌ی اول

مرجع: کتاب لینز

زبان صوری Formal Language

زبان های طبیعی ← با این ها گویش می کنیم — با دیگری دارنده‌ی کلمات ← ناکردد

رسم الخط یک زبان ← تحولات زبانی به وجود آورد

صحبت در ارتباط با نوع گویش

نقطه‌ی تحول نهایی ← وقتی بود که گرامر زبان شکل گرفت

زبان های صوری

✓ زمانی که نیاز می خواهیم با سخت افزاری ارتباط برقرار کنیم

✓ همه چیز قانونمند است

✓ از همان ابتدا سخت مجموعه‌ی حروف الفبایی تشریح می شود

✓ در زبان های طبیعی ساخت کلمات براساس الگو نیست ولی زبان Formal از همان ابتدا براساس الگو است.

✓ روی سخت افزار باید اطلاعات در DB آن باشد پس دامنه‌ی کلمات نباید نامحدود باشد.

زبان های Formal

۱- دیدگاه سخت افزاری

سخت افزار آن را بفهمد توسط سخت افزاری پذیرفته شد پس همان است.

مثلاً ماشین های DFA یا NFA ← منظم

ماشین تورینگ ← بدون قید

۲- دیدگاه گرامری

ساختار جمله‌بندی ها مشخص است ← گرامر زبان فارسی / انگلیسی / ...

دسته‌بندی زبان ها از دیدگاه گرامری است.

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

پشته ای ماشین ← مستقل از متن

حساس به متن ← گرامرهای حساس به متن

✓ زبان های منظم دیدگاه دیگری هم دارد

← دیدگاه عبارت های منظم

۱,

آتاماتا ← ماشین ها

هدف ← طبقه بندی زبان های formal

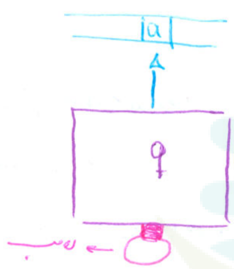
شروع درس:

ماشین قطعی متناهی DFA

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

✓ یک نوار و یک هد خواندن دارد 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 دوگانه 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 $w = a_1a_2a_3\cdots a_n$ رشته‌ای به طول n است، (روی حروف الفبای Σ) هرگاه برای هر i ،
$a_i \in \Sigma$ باشد.

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

رشته‌ای به طول صفر را رشته پوچ نامیده و با λ یا ε نمایش می‌دهیم.
← رشته‌ای که کنار هر رشته 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 رشته w_1 به دست می‌آید.

$w_1 = Ali$

$w_2 = Ahmadi$

$w_1.w_2 = AliAhmadi$

تعریف Σ^n

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

مثال: $\Sigma=\{a,b\}$ آنگاه

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

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

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

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

تعداد $3^3=27$

نکته: هرگاه Σ مجموعه حروف الفبا باشد:

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

۲۱

تعریف Σ^* و Σ^+

هرگاه Σ مجموعه حروف الفبای باشند،

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

n را از ابتدا انتخاب کرده ام ∞ نیست.

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

توجه

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

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

$(\Sigma^*, \cdot)$

Semigroup

۱- عمل الحاق روی Σ^* بسته است.

$\forall w_1, w_2 \in \Sigma^* \quad w_1 \cdot w_2 \in \Sigma^*$

✓ اگر مجموعه ای نسبت به عملی بسته باشند یک سیستم جبری است.

۲- شرکت پذیری

$\forall w_1, w_2, w_3 \in \Sigma^* \quad w_1 \cdot (w_2 \cdot w_3) = (w_1 \cdot w_2) \cdot w_3$

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

نیم گروه

۳- عضو خنثی

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

✓ اگر هر سه خصوصیت را داشته باشند به آن تکوار، تک لون یا monoid گفته می شود.

تکوار

۴- اگر هر عضوی وارون داشته باشند گروه می شود که در اینجا نداریم.

✓ تنها عضوی که عضو وارون دارد λ است.

تعریف طول رشته نسبت به حرف

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

مثال: اگر $\Sigma=\{a,b,c\}$ آنگاه برای $\omega = abacba$ داریم:

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

نکته: هرگاه (ω_1, ω_2 متعلق به Σ^*) $\omega_1, \omega_2 \in \Sigma^*$ آنگاه:

i) $|\omega_1 . \omega_2| = |\omega_1| + |\omega_2|$

ii) $|\omega_1 . \omega_2|_a = |\omega_1|_a + |\omega_2|_a$ برای $a \in \Sigma$

iii) $\sum_{a\in\Sigma} |\omega|_a = |\omega|$ $\quad \sum \leftarrow$ مجموع

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

مثال: $|\omega|_a + |\omega|_b + |\omega|_c = |\omega|$

قرارداد: هرگاه $a \in \Sigma$ باشد، آنگاه:

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

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

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

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

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

تعریف معکوس یک رشته

هرگاه ω یک رشته روی Σ (حروف الفبای) باشد، آنگاه معکوس رشته‌ی ω را با ω^R یا ω^r نمایش می‌دهیم. به شکل زیر تعریف می‌کنیم:

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

نتیجه: هرگاه $a \in \Sigma$ آنگاه

$a^r = a$

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

۳،

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

تعریف رشته ی خود معکوس (پالیندروم)

رشته ی w متعلق به Σ^* را خود معکوس می نامیم هرگاه

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

* تعریف زبان

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

نتیجه: مجموعه ی تمام زبان ها روی حروف الفبای را با 2^{Σ^*} نمایش می دهیم.

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

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 می دهیم و به شکل زیر تعریف می کنیم:

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

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:

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

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) $\forall\, \omega_1, \omega_2 \in L \quad (\omega_1 . \omega_2)^r = \omega_2^r . \omega_1^r$, $(\omega_1^r)^r = \omega_1$

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 الفبا باشد، تعداد زبان‌های 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 می دهیم. در این صورت:

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

نکته: مجموعه ی مرجع به مفهوم عام وجود ندارد.

تست: تعداد زبان های L که $\overline{L^*} = \bar{L}^*$ صدق می کند چیست؟

۱) ۰ ✓ ۲) ۱ ۳) ۲ ۴) بی نهایت (بی شمار)

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

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

تست: $\varphi\ 2^{\Sigma^*} \to 2^{\Sigma^*}$

L_2 هر زبان

$\varphi(L) = \bar{L}$ در این صورت:

الف) φ تابعی ۱-۱ و پوشا است. ✓

ب) φ ۱-۱ است ولی پوشا نیست.

ج) φ پوشا است ولی ۱-۱ نیست.

د) φ پوشا و ۱-۱ نیست.

۱-۱ ✓

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

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

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 ✗

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

تست: اگر $\Sigma = \{a,b,c\}$ و $L - \Sigma^* = \phi$ آن گاه L کدام یک از زبان های زیری تواند باشد؟

i) Σ^* — ۱) فقط I

ii) $a^n b^{n^2} c^n$ — ۲) فقط iv

iii) ϕ — ۳) فقط i و iii

iv) ε — ۴) همه ✓

نکته

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

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

(۱)

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

(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)

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

$$\Rightarrow$$

$$L^* = L$$

(۵)

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

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

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

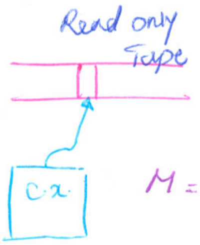
$$\{a,b\}^* \subseteq \{a^n b^n \omega \mid \omega \in \{a,b\}^*\}$$

(۶)

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

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

۵۱



* پذیرنده متناهی معین/قطعی DFA

Deterministic Finite Automaton/Acceptance

یک پذیرنده ی متناهی قطعی dfa یک پنج تایی مرتب است که در آن

$M = \langle Q, \Sigma, \delta, q_0, F \rangle$

الف: Q ← مجموعه ای متناهی و ناتهی از وضعیت های داخلی است

ب: Σ مجموعه ی حروف الفبایی است

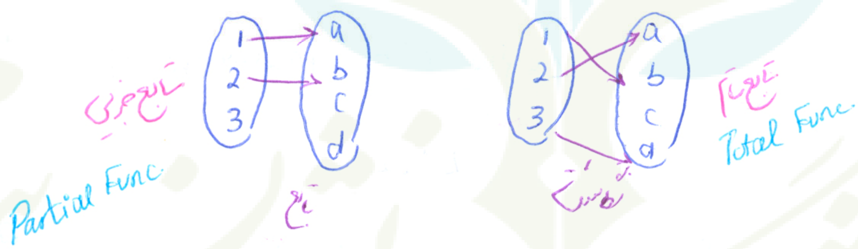
ج: δ ← تابع گذر یا انتقال است Transition Func

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

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

ه: $q_0 \in Q$ ← وضعیت شروع است

فرق نگاشت و تابع



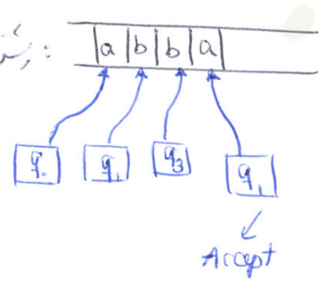
مثال: DFA ←

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

که تابع δ به شکل زیر توصیف 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

رشته ی ورودی:



✓ می‌خواهیم تابع δ را به تابع δ^* چنان 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 input ی نباید تغییر کند.

اگر در وضعیتی باشد حالتی که هیچ کاراکتری وارد نشده نباید به وضعیت دیگری برود

۲)

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

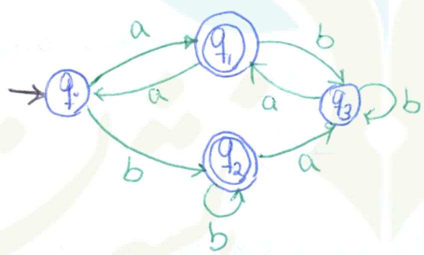
δ^* نسخه‌ی جدیدی از δ است. پس عملکرد تابع δ^* در دریافت یک حرف باید همان رفتار تابع δ باشد

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

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



جلسه ی دهم

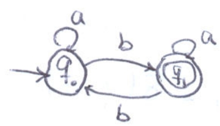
تعریف : زبان پذیرفته شده توسط DFA

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

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

مثال: زبان پذیرفته شده توسط dfa مقابل چیست؟

M:



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

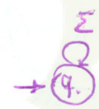
← باقی مانده ی تعداد b بر 2 یک است ← یعنی تعدادش فرد است.

تعریف زبان منظم

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

مثال: نشان دهید زبان های زیر منظم هستند:

1) $L=\emptyset$

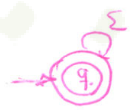


2) $L=\lambda$

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

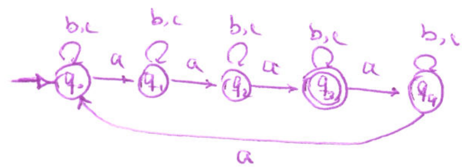
توجه: اگر زبان یک DFA بخواهد شامل رشته ی پوچ باشد، حتماً می بایست وضعیت شروع پایانی باشد.

3) $L=\Sigma^*$

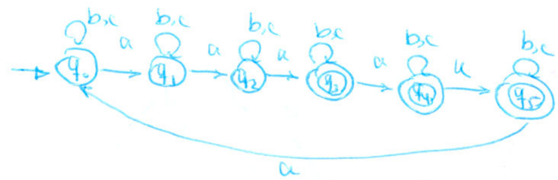


4) $L=\{w\in\{a,b,c\}^* \mid n_a(w) \bmod 5 = 3\}$ 3, 8, 13, ⋮

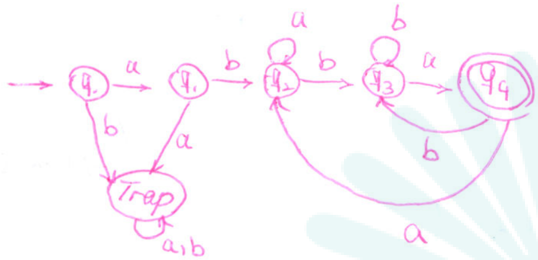
$\{0,1,2,3,4\}$



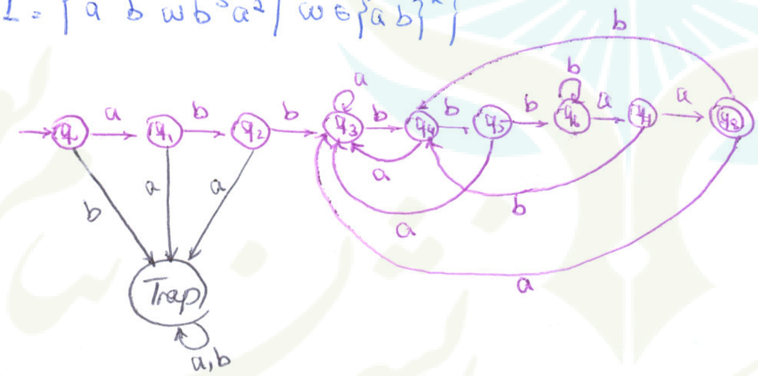
5) $L=\{w\in\{a,b,c\}^* \mid n_a(w) \bmod 6 > 2\}$ 3 4 5



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

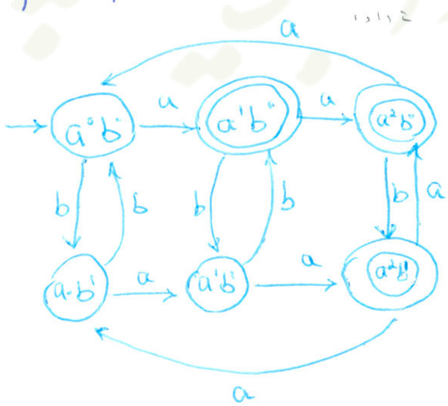


7) $L=\{ab^2wb^3a^2 \mid w\in\{a,b\}^*\}$

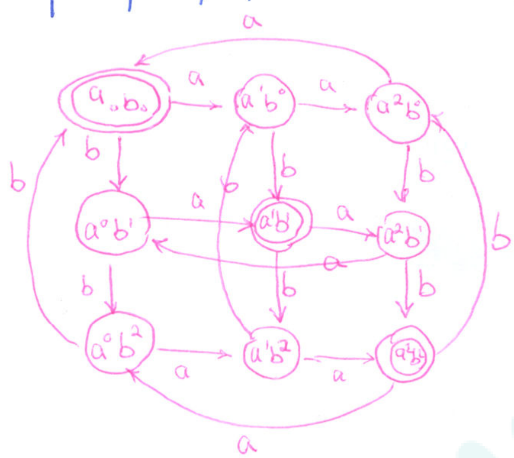


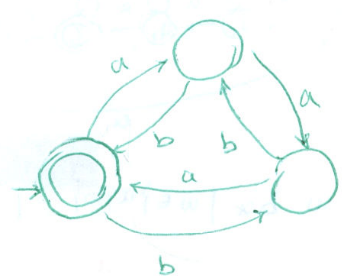
8) $L=\{w\in\{a,b\}^* \mid n_a(w) \bmod 3 > n_b(w) \bmod 2\}$

0, 1, 2 0, 1



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





تمرین : DFA ها را رسم کنید.

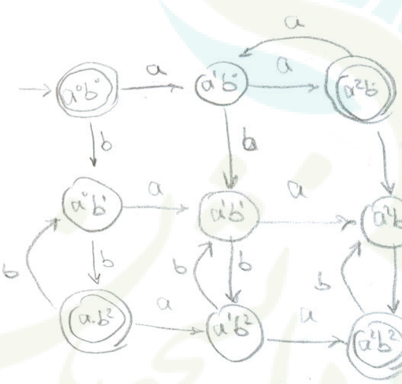
① $L = \{ a^n b^m \mid (n+m) \bmod 2 = 0 \}$ → n+m عدد زوج

n,m زوج

n,m فرد

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

①



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

این زبان منظم نیست چراکه باید تعداد a را جایی ذخیره کند، طی هر بار چون تعداد بار شمرده نمی‌شود. جایی ذخیره‌سازی 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: $L=L(M)$. درنتیجه تعریف می‌کنیم $S:=\{\delta^*(q_0,a^i) \mid i\in W\}$ پس $S\subseteq Q$ که چون $|Q|<\infty$ است پس $|S|<\infty$

(✓ نتیجه‌ی δ^* همیشه یک وضعیت است)

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

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

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

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

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

درنتیجه $\delta^*(q_0,a^{k_1}b^{k_2})\in F$ 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، آن‌گاه متمم L نیز یک زبان منظم خواهد بود.

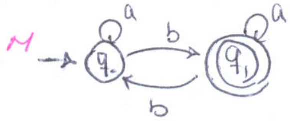
اثبات: چون L یک زبان منظم است DFA ای چون $M=\langle Q,\Sigma,\delta,q_0,F\rangle$ چنان موجود است که $L=L(M)$ حال اگر این DFA را به گونه‌ای تغییر دهیم که هر وضعیت نهایی آن به وضعیت غیرنهایی و هر وضعیت غیرنهایی به وضعیت نهایی تبدیل شود، DFA به دست آمده یعنی $\bar{M}=\langle Q,\Sigma,\delta,q_0,Q\setminus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

نتیجه: متمم هر زبان نامنظم، نامنظم است.

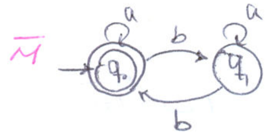
زیرا اگر متمم یک زبان نامنظم، منظم می‌بود، متمم آن نیز که خود زبان اصلی است باید منظم بود که طبق فرض زبان نامنظم بوده است.

۸/

مثال:



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



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

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 $M = \langle \hat{Q}, \Sigma, \hat{\delta}, \hat{q}_0, \hat{F} \rangle$ را به طور 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:

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

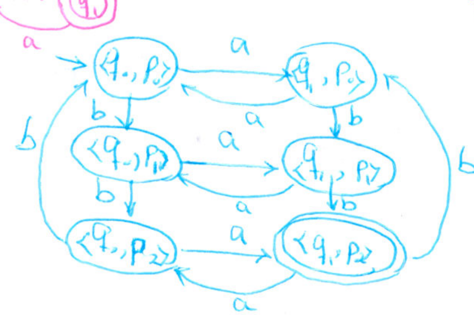
مثال:

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

نشان دهید L منظم است.

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

تعداد وضعیت‌ها = 2x3 = 6



وضعیت شروع همیشه یکی

Final ها ضرب دکارتی

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

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

نکته: زیرمجموعه‌های زبان‌های منظم، الزاماً منظم نیستند.

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

آیا ابرمجموعه‌های زبان‌های نامنظم، منظم هستند؟

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

لزوماً منظم نیستند

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: خانواده‌های نامنظم تحت اجتماع بسته نیستند. یعنی لزوماً اجتماع دو زبان نامنظم، نامنظم نمی‌باشد.

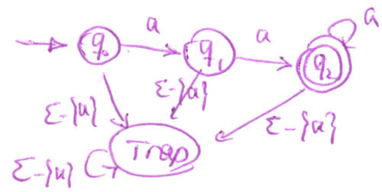
نکته: خانواده‌ی زبان‌های نامنظم تحت اشتراک نیز بسته نیستند. یعنی لزوماً اشتراک دو زبان نامنظم، نامنظم نمی‌شود.

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

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

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

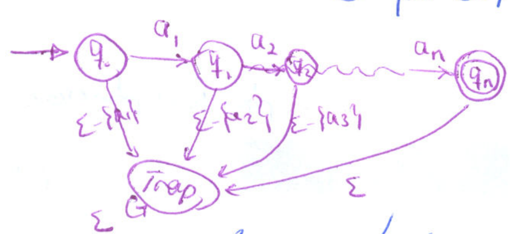
$$L_2 \cdot L_1 = \underbrace{\{a^2, a^3, a^4, a^5, \dots\}}_{\text{منظم}}$$



q_1

نکته: هرگاه Σ مجموعه‌ی 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 ⇐ منظم

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

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

$L_1=\{a^1b^1\}$ همه منظم

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

$\vdots$

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

$L=\bigcup_{k=0}^{\infty} L_k=\{a^nb^n \mid n\ge 0\}$ 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 نشاند.

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

← Q^+ و Q^- شمارا. Q^+ ، Q^- با هم 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.

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

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

$\vdots$

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

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

NFA* (NDFA)

Non-deterministic Finite Automaton

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

1) Q مجموعه‌ای متناهی و ناتهی از وضعیت‌ها که داخل آن است.

2) Σ مجموعه حروف الفبای

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 پایانی است.

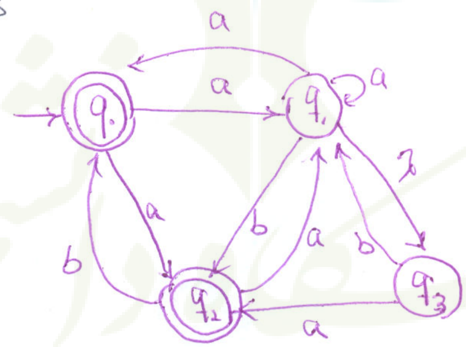
مثالی از یک NFA

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

Q \ $\Sigma\cup\{\lambda\}$	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	$\{q_2\}$	$\{q_1\}$	$\emptyset$

• تابع تعریف شده به شکل زیر nfa است.

مطابق آن چه 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 داده شود، از یک وضعیت به وضعیت دیگری برود.

✓ می‌خواهیم تابع δ را به تابع δ^* چنان 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 نشته‌های برابر است. تمام رشته‌هایی که از اضافه کردن تعدادی رشته‌ی پوچ (λ) به هر یک رشته‌ی w به دست آمده باشند، همان رشته‌ی w است (از نظر ما) بنابراین $\delta^*(q,w)$ نتیجه‌ی پیمایش همه‌ی رشته‌های w از وضعیت q است.

۱۰۱

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

λb $\lambda b\lambda$

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

جلسه ی بعد

- زبان پذیرفته شده توسط NFA
- معادل بودن NFA و DFA
- کمینه کردن DFA ها

جلسه ی سوم

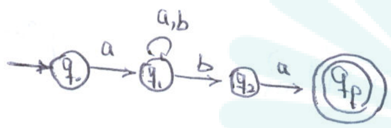
زبان پذیرفته شده توسط پذیرنده متناهی غیرقطعی nfa:

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

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

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 می توان از یک وضعیت با یک حرف الفبای به هیچ یا یک یا چند وضعیت دست یافت.

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

به این دلیل است NFA ها را آتاماتهای غیرقطعی نامند.

نکته: از آن جایی که ثابت هر DFA را می توان برای یک NFA در نظر گرفت، پس هر زبان پذیرفته شده توسط یک DFA، توسط یک NFA نیز پذیرفته می شود.

تعریف دو آتاماتای معادل (هم ارز):

دو آتاماتای M_1، M_2 را معادل با هم ارز می نامیم هرگاه،

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

زبان هایی که می پذیرند با هم یکسان باشند.

۱۷

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

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

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

معادل آن ایجاد کرد که:

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

الگوریتم ایجاد DFA M_D

هرگاه NFA، M_N داده شده باشد، در این صورت:

1) قرار می‌دهیم

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

2) هرگاه وضعیت $\{q_i, q_{i+1}, \dots, q_j\}$ از DFA، M_D ایجاد شده باشد، برای $a \in \Sigma$ قرار می‌دهیم:

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

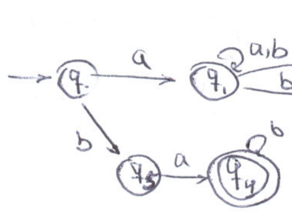
اگر حاصل اجتماع مذکور $(q_K, q_{K+1}, \dots, q_M)$ باشد، در این صورت این وضعیت را به عنوان یک وضعیت DFA، M_D در نظر می‌گیریم.

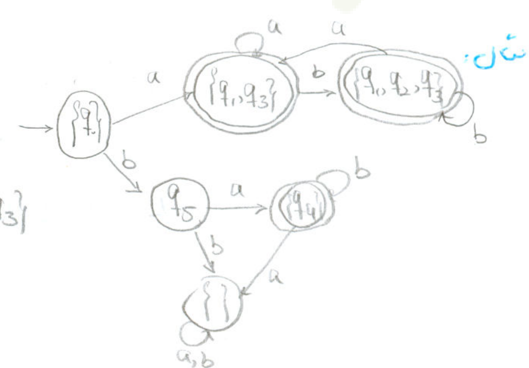
3) گام 2 را آن قدر تکرار می‌کنیم تا نتوان وضعیت جدیدی ایجاد کرد.

4) برای تعیین کردن وضعیت‌های نهایی هر وضعیتی که شامل یک وضعیت نهایی از NFA، M_N باشد، آن وضعیت را به عنوان وضعیت نهایی در نظر می‌گیریم.

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

مثال:





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

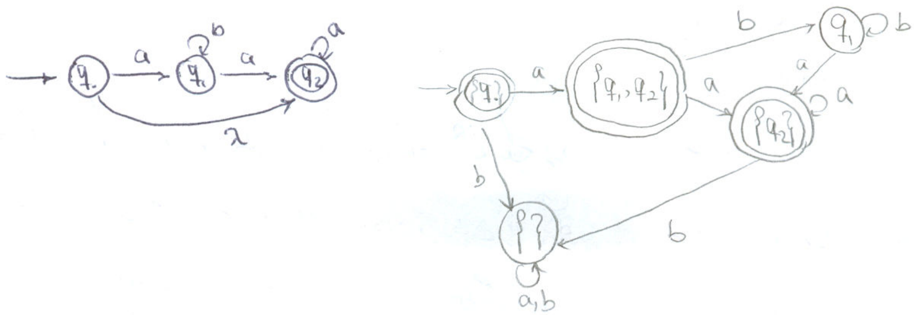
$\delta_N^*(q_3, a) = \{\}$

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

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

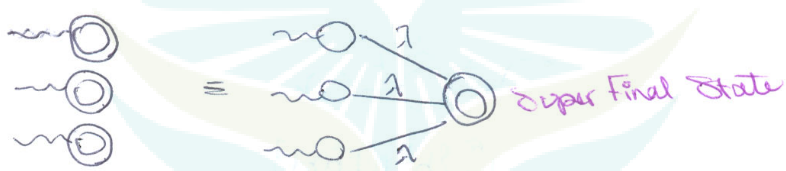
$\delta_N^*\{q_3, b\} = \Phi$

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



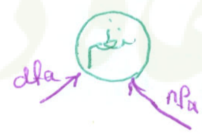
نکته: به ازای هر NFA داده شده با N حالت (State)، DFA معادل با آن، حداکثر دارای 2^N حالت خواهد بود.

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



نکته: در برخی از کتب وقتی صحبت از NFA است، به این NFA امکان را می‌دهند که چندین حالت شروع داشته باشد. این تعریف دقیقاً معادل با تعریف سابق است.

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

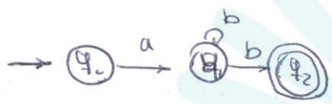
در پذیرش زبان‌ها، DFA و NFA یکسان عمل می‌کنند

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

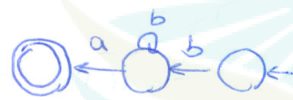
نکته: هرگاه L یک زبان منظم باشد، آن گاه L^R یک زبان منظم است.

چون L منظم است، پس یک NFA وجود دارد که L را می‌پذیرد. این NFA را می‌توان به NFA ی تبدیل کرد که صرفاً 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:



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



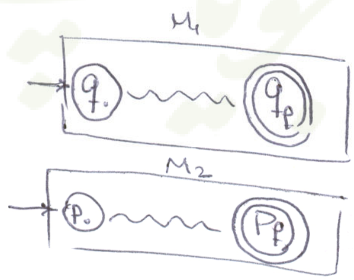
$L^R = \{b^n a \mid n \geq 1\}$

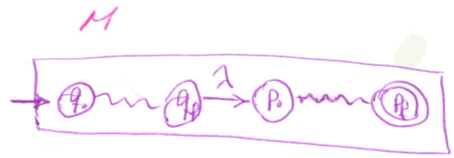
نکته: الحاق دو زبان منظم، منظم است. به عبارت دیگر خانواده‌های زبان‌های منظم تحت عمل الحاق بسته هستند.

هرگاه L_1 و L_2 دو زبان منظم پذیرفته شده توسط NFA های M_1 و M_2 باشند، یعنی

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

می‌خواهیم NFA ی بسازیم که $L_1 . L_2$ را پذیرش نماید. برای این منظور می‌توان فرض کرد که NFA M_1 صرفاً یک وضعیت پایانی داشته باشد:

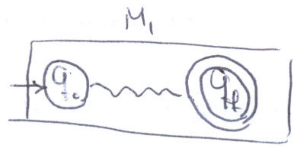


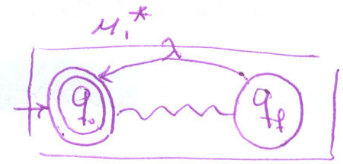


$L(\hat{M}) = L(M_1) . L(M_2) = L_1 . L_2$

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

هرگاه L_1 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

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

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

$$L_1/_l L_2=\{x \mid \exists y\in L_2 \text{ s.t } yx\in L_1\}$$

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

$$* \left(\frac{L_1}{L_2}\right)_r^R = {}_l\, L_1^R / L_2^R \; *$$

$$* \; {}_l\frac{L_1}{L_2} = \left(L_1^R /_r L_2^R\right)^R \; *$$

مثال:

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

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

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

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

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 زبان های منظم تحت زبان خارج قسمت بر هر زبان دلخواه بسته هستند.

الگوریتم محاسبه‌ی زبان منظم خارج قسمتی به صورتی که L_2 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.

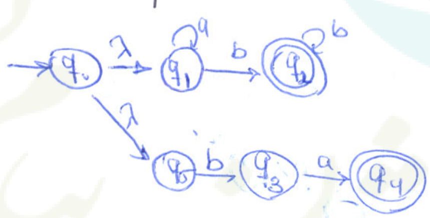
حال یک dfa دقیقاً مانند dfa زبان L_1 ترسیم می‌کنیم که برای آن هیچ وضعیت نهایی ابتدا در نظر گرفته نمی‌شود.

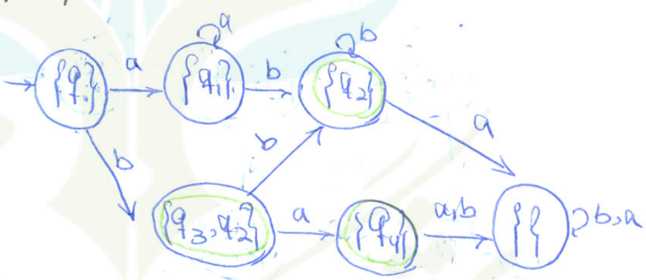
حال برای مشخص کردن وضعیت های نهایی DFA هم به شیوه‌ی زیر عمل می‌کنیم. اگر رشته‌ای از وضعیت q_0 از DFA زبان L_1 شروع می‌کنیم آیا رشته‌ای از L_2 با وضعیت شروع از وضعیت q_0 در DFA زبان L_1 الزاماً به وضعیت نهایی دست یابیم، وضعیت q_0 DFA خارج قسمتی را نهایی می‌کنیم.

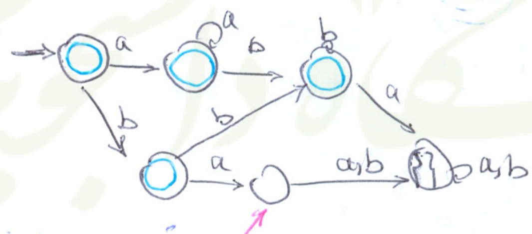
مجدداً همین کار را با سایر وضعیت های DFA دنبال می‌کنیم و روند مذکور را تکرار می‌کنیم.

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 حالت 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 هر 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 هم ریختی

هرگاه $(G_1, *_1)$, $(G_2, *_2)$ دو تیگروه یا تکوار باشند، گوییم

$\varphi : G_1 \to G_2$ یک هم ریختی 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 :

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

هم ریختی زبان ها

هرگاه Σ و Γ دو مجموعه حروف الفبای باشند: آنگاه $(\Sigma^*, \cdot)$, $(\Gamma^*, \cdot)$ دو تکوار هستند.

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

$$\varphi(w_1 \cdot w_2) = \varphi(w_1) \cdot \varphi(w_2)$$

۱٤۱

نتیجه: برای هر ریخت 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 عنصر از فضا را به شکل منحصر به فرد نمایش می‌دهد.

Σ حلقه‌ای پایه‌ی Σ^* است. یعنی هر عنصر 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:

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

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

$\varphi(\{)=begin$

$\varphi(\})=end$

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

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

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

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

$begin\ begin\ end\ end$

نکته: تصویر هم‌ریخت زبان

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

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

مثال:

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

$\varphi : \{a,b\}^* \longrightarrow \{c,d,e,f\}^*$ همانی صورت $\Phi(L) = \{(cd)^n (ef)^n \;\; n \geq 0\}$

$\varphi(a) = cd$

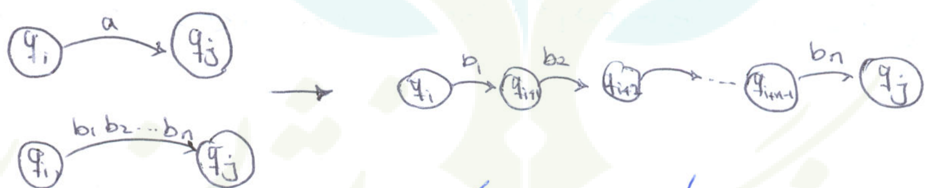
$\varphi(b) = ef$

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 تصویر همریختی ها بسته هستند.

چون L منظم است پس یک NFA وجود دارد که L را می‌پذیرد. حال یک NFA ارائه می‌کنیم که $\varphi(L)$ را بپذیرد. برای این منظور به ازای هر حرف $a \in \Sigma$ به عنوان برچسبی 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 طول بزرگتر از یک باشد بین q_i و q_j به اندازه کافی 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:

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

$\varphi(a) = a$

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

$\varphi(c) = \lambda$

همارز 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 $P(Q)$.

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

151

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 موازی‌ها را حذف می‌کنیم سپس کار راحت‌تر و پایا‌تر می‌شود.
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 بدست آید، 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 داد 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 آن قدر تکرار می‌کنیم تا هیچ زوج ادغام‌ناپذیری باقی نماند.

جلسه‌ی چهارم

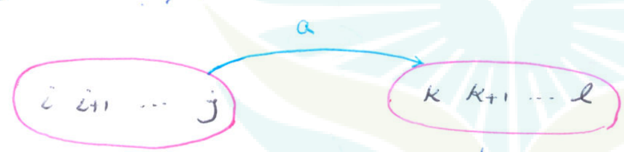
روال کمینه کردن DFA ها

هدف ← حداقل کردن تعداد وضعیت ها

مراحل این پردازه به شرح زیر است:

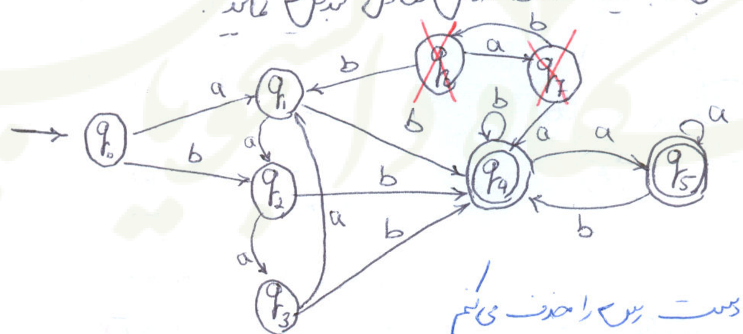
۱- روال علامت‌گذاری را فراخوانی می‌کنیم تا تمام جفت وضعیت های ادغام ناپذیر معین شوند. سپس وضعیت هایی که قابل ادغام هستند را با هم ادغام می‌کنیم. یعنی اگر وضعیت های q_i، q_{i+1}، ... ، q_j ادغام پذیر باشند، در این صورت یک وضعیت $i\ i{+}1 \dots j$ را ایجاد می‌کنیم.

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 می‌کنیم. مطابق زیر:



۳- وضعیتی وضعیت شروع است، که اندیس صفر را داشته باشد.
وضعیتی وضعیت پایانی است که تشکیل اندیس یکی از وضعیت های نهایی باشد.

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



۱- وضعیت های غیر قابل دسترس را حذف می‌کنیم

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

(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)

۱۶۱

۳- در این گام سایر ادغام ناپذیرها را پیدا می‌کنیم.

(q_0, q_1)

(q_0, q_2)

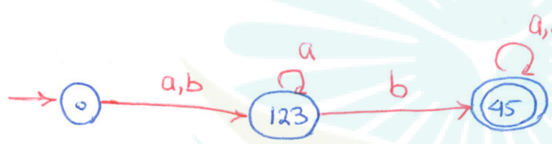
(q_0, q_3)

$\delta(q_0, b) = q_2$
$\delta(q_1, b) = q_4$
$\delta(q_2, b) = q_4$
$\delta(q_3, b) = q_4$

$\delta(q_4, a) = q_5$
$\delta(q_5, a) = q_5$

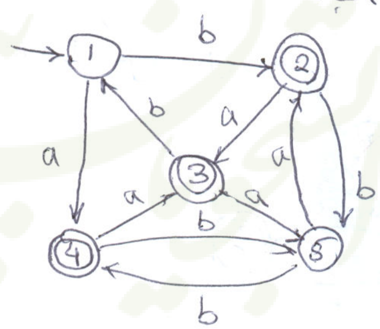
$\delta(q_4, b) = q_4$
$\delta(q_5, b) = q_4$

۴- این ماشین یک ماشین مینیمال است.



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

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



(1,2) (1,3) (1,4)

(5,2) (5,3) (5,4)

(2,3) (4,3)

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

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

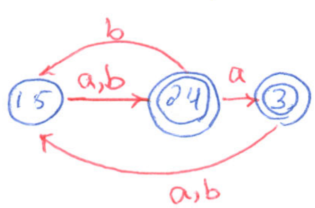
$2,b \to 5$ $3,b \to 1$

$2,a \to 3$ $3,a \to \cancel{5}$ ×

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

1,5 ادغام می‌شوند

2,4 ادغام می‌شوند



تمرین: هرگاه L یک زبان منظم باشد، در مورد منظم یا نبودن زبان های زیر اظهار نظر نمائید.

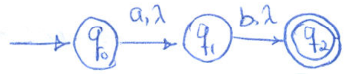
$\Sigma = \{a, b, c\}$

① $L_1 = \{w \in L \mid$ که 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 هر جا a داریم را به Trap بفرستیم.

③ $L_3 = \{w \in L \mid n_a(w) = 1\}$

این زبان هم منظم است.

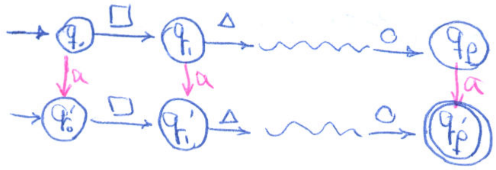
به ازای هر یال a یک Copy تهیه می کنیم و یال های a را حذف می کنیم

نکته: $n_a(w) = k$ ← منظم خواهد بود.

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

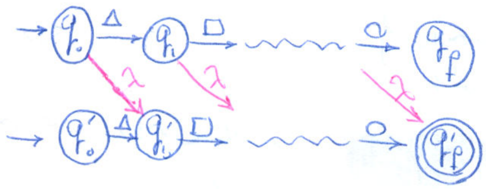
④ $L_4 = \{uav \mid u, v \in \Sigma^*, uv \in L\}$

منظم است.



171

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



مثال: کدامیک 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 هر حالت بیش از یک بار عبور نمی شود $A\}$ شکل حاصل NFA

$L_3 = \{w \in \{0,1\}^* \mid$ تعداد 0 ها و 1 ها برابر و 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) هیچکدام 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 می کنیم $\overline{M} = (Q, Q_0, \Sigma, Q-F, \delta)$ 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 زبان های M_1 و M_2 است. فرض کنید G_1 و G_2 دو گرامر منظم باشند که زبان آن ها به ترتیب معادل زبان های M_1 و M_2 هستند. کدام عبارت صحیح است؟

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

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

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

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

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

هربار complement می کنیم می بایستی 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 $a\in\Sigma$، $r=\lambda$، r را یک عبارت منظم پایه می نامیم

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

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

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

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

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، آنگاه

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

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

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

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

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

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

تعریف دو عبارت منظم r_1 و r_2 معادل

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

در این صورت می‌نویسیم

$r_1 \equiv r_2$ یا $r_1 = r_2$

✓ هرگاه r_1 و r_2 دو عبارت منظم دلخواه 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) $r_1 + \phi = \phi + r_1 = r_1$, $r_1 . \phi = \phi . r_1 = \phi$

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

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

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

8) $r_1 (r_2 r_3) = (r_1 r_2) r_3$

مثال: برای زبان‌های زیر عبارات منظم را طوری 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) $L_4 = \{ a^n b^m \mid (n+m) \bmod 2 = 0 \}$

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

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

191

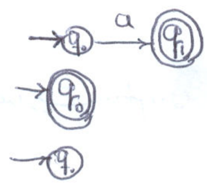
قضیه: زبان هر عبارت منظم، منظم است.

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

$r=a$

$r=\lambda$

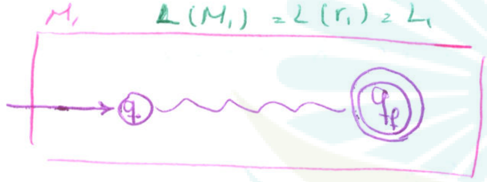
$r=\phi$



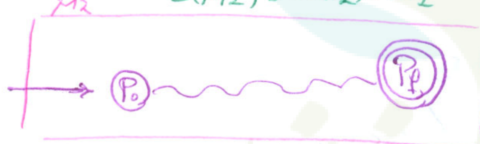
حال فرض کنید که r_1 و r_2 دو عبارت منظم باشند، که به ترتیب $L(r_1)$ و $L(r_2)$ توسط NFA هایی M_1 و M_2 که تنها یک وضعیت نهایی دارند و به علاوه هیچ دو وضعیت هم‌نامی ندارند، پذیرفته می‌شوند.

در این صورت:

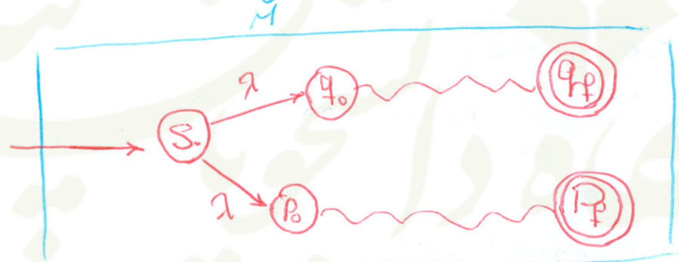
M_1 $\quad L(M_1)=L(r_1)=L_1$



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

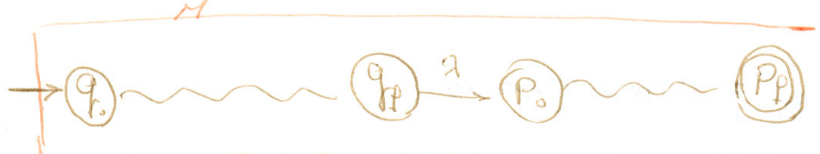


$\breve{M}$



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

$\hat{M}$



$L(\hat{M})=L(M_1).L(M_2)=L(r_1).L(r_2)=L(r_1r_2)$

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

نتیجه: زبان هر عبارت منظم، منظم است.

تعریف GTG Generalized Transition Graph

گراف انتقال تعمیم یافته

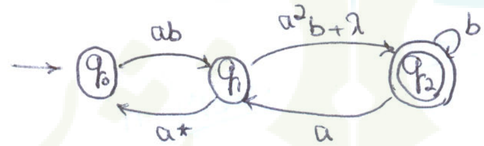
Generalized Transition Diagram GTD

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:



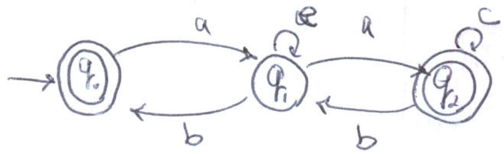
نکته: برای هر دیاگرام انتقال غیرقطعی را قطعاً می توان یک GTG با تنها دو وضعیت ایجاد کرد که یکی وضعیت نهایی و دیگری وضعیت شروع است به طوری که این دو وضعیت متمایز از یکدیگرند.

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.

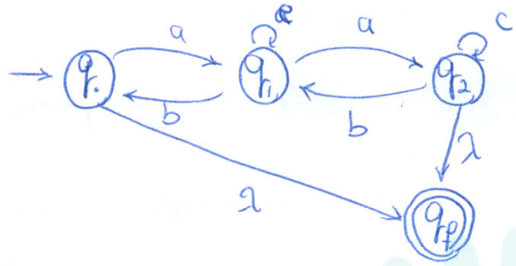
سپس وضعیت های میانی را به طور سریال یکی یکی حذف می کنیم به طوری که با حذف هر وضعیت عملکرد آن را روی وضعیت های مجاورش اعمال می کنیم. تا این که فقط دو وضعیت شروع و پایانی باقی بمانند.

۲۰۱

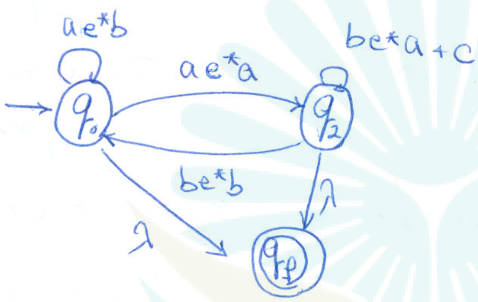
مثال:



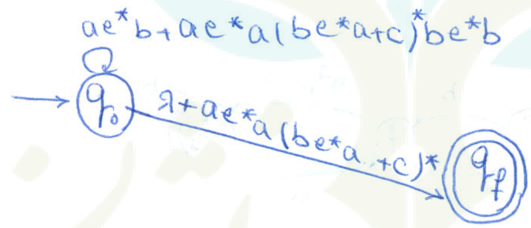
1



2

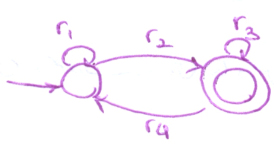


3



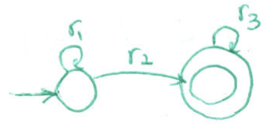
نکته: به ازای هر زبان منظم مانند L روی حروف الفبای Σ یک عبارت منظم r چنان موجود است که L(r) = L باشد. به عبارت دیگر همه ی زبان های منظم را می توان با عبارات منظم نمایش داد.

اثبات: از آن جایی که L منظم است، می توان برای آن یک GTG با دو وضعیت مطابق شکل زیر در نظر گرفت. کلی ترین حالت این GTG به شکل زیر است.



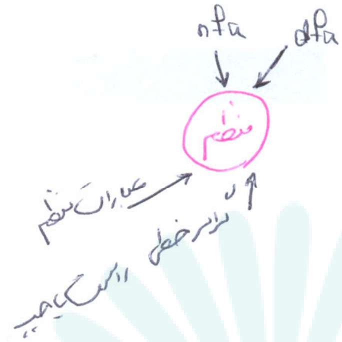
درنتیجه عبارت منظم معادل با GTG مذکور عبارت است از:

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



$r_1^* r_2 r_3^*$

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.

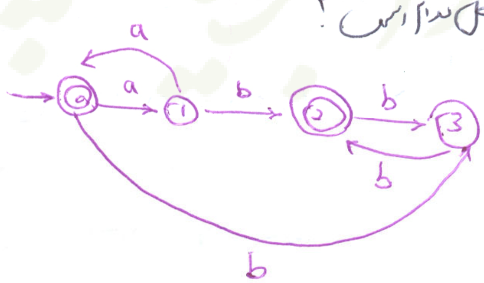
1) $\{a,b\}^* \subseteq \{a^n b^n (a+b)^* \mid n \geq 0\} \subseteq \{a,b\}^*$

2) ✓ $\{b^* a^n b^n a^* \mid n \geq 0\}$ چون باید a ها n و b ها n تا بخواند نمی 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 $L\cdot\Sigma^*=\phi$ باشد آن گاه L کدامیک از زبان های زیر می تواند باشد؟

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

۱) فقط I
۲) فقط IV
۳) فقط I و III
۴) I، II، III و IV ✓

$L\subseteq\Sigma^*$

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

۱) منظم ✓
۲) مستقل از متن ولی منظم نیست.
۳) حساس به متن ولی مستقل از متن نیست.
۴) بدون محدودیت ولی حساس به متن نیست.

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

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

$\gamma\in\Sigma^*\Rightarrow\gamma=\lambda^1(\lambda\beta)^0(\gamma\lambda)^1\in L_1$

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

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

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

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

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

۱) L_2 و L_3 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، L_2 و L_3 همگی نامنظم هستند.

$\beta\in\Sigma^+\Rightarrow$

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

پس $\Sigma^+\subseteq L_3$

چون λ عضو این زبان نیست پس Σ^* نمی تواند باشد

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

جلسه ی پنجم

تعریف گرامر

برای زبان های منظم دو دیدگاه بیان شده است: ماشین (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 ای متناهی و ناتهی از علائمی باشد که به آن مجموعه غیرپایانه ها (متغیرها) Nonterminal = Variable

معمولاً هر متغیر را با حروف بزرگ الفبایی 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 حروف کوچک الفبایی نمایش می دهند.

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

۴- P مجموعه قواعد جانشینی 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 گرامرها 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

$\alpha\to\beta_1$
$\alpha\to\beta_2$
$\vdots$
$\alpha\to\beta_n$

عناصری از P باشند برای سادگی می نویسیم:

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

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

۲۲

مثال:

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

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

سمت چپ هر رشته ای می تواند باشد، حتی تهی

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

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

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

سمت چپ هم رشته ای هم غیرپایانی می تواند باشد

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

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

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

حداکثر یک غیرپایانه وجود داشته باشد.

$S \to aAb$ خطی
یک متغیر

$C \to aDCb$ غیرخطی
دو متغیر

۵- گرامر خطی راست (Right linear)

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

$A, B \in V$
$x \in T^*$

۶- گرامر خطی چپ left linear

هرگاه همه قواعد آن به شکل $A \to Bx$ یا $A \to x$ که در آن $A, B \in V$، $x \in T^*$

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 گرامرهای است صرفاً قواعد آن را مشخص کنیم غیرپایانه اولین قاعده را از سمت چپ با غیرپایانه شروع، شروع می کنیم.

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 < سمت چپ ⇐ حساس به متن نیست

سمت چپ هر رشته ای می تواند 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 — حساس به متن — بدون محدودیت

خطی — سمت راست < سمت چپ — 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 با 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؟

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

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow \cdots \Rightarrow a^nSb^n$$

$$\lambda \qquad ab \qquad a^2b^2 \qquad a^nb^n$$

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

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

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

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

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

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

$n = m$ یا $m = 2n$

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

$S \to AB$

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

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

یا $\quad S \to aSb \mid Sb \mid b$

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

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

دقت می‌کنیم هر گاه احتیاج داشتیم باید چند حالت کنیم

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

$A \to aA \mid a$

$B \to Bb \mid b$

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

$S \to aSb \mid b$

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

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

$S \to aAC \mid A$ $\quad$ تعداد مساوی a, c

$A \to bAC \mid \lambda$ $\quad$ تعداد مساوی b, c

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

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

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

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

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

$S \to S_1 \mid S_2$

$S_1 \to aS_1c \mid A$ $\quad$ و $\quad S_2 \to AD$

$A \to aAb \mid \lambda$ $\quad A \to aAb \mid \lambda$

$D \to bDc \mid \lambda$

۲۴۱

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

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

که همین ترتیب تولید می‌شود

مثلاً جوابی 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 $abb \mid bba$

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

مثل قبل است

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

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

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

b تا ۲ و c تا ۲ تولید می‌شود

$aAbc \to abAc \to abBbcc$

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

می‌خواهیم A را پشت حروف بگذاریم تا به تعداد مساوی a و 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 باز گردد

دسته‌بندی زبان‌ها برحسب 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 به 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 مانند 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:

$$A\rightarrow\lambda \quad \text{یا} \quad A\rightarrow B \quad \text{یا} \quad A\rightarrow a_1a_2\cdots a_nB$$

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

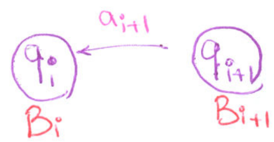
$$A\rightarrow a_1a_2\cdots a_nB$$

$$A\rightarrow a_1B_1$$
$$B_1\rightarrow a_2B_2$$
$$B_2\rightarrow a_3B_3$$
$$\vdots$$
$$B_{n-1}\rightarrow a_nB$$

۲۵

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

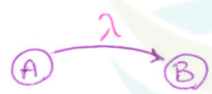
- حال برای قاعده ای مانند $B_i \to a_{i+1} B_{i+1}$ از وضعیت متناظر با B_i به وضعیت متناظر با B_{i+1} یالی با برچسب a_{i+1} رسم می کنیم مطابق شکل زیر



این وضعیت را ادامه می دهیم تا تمام وضعیت ها و برچسب های شکل ایجاد شوند.

- برای تعیین کردن وضعیت های پایانی تمام وضعیت های متناظر با متغیرهای پایانه هایی که به λ رفته اند، پایانی می کنیم یعنی اگر $A \to \lambda$

- برای قاعده ای که الگوی قاعده ای 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 می کنیم.



درنتیجه: برای nfa ساخته شده که M نامگذاری می کنیم $L(M)=L(G)$

مثال:

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

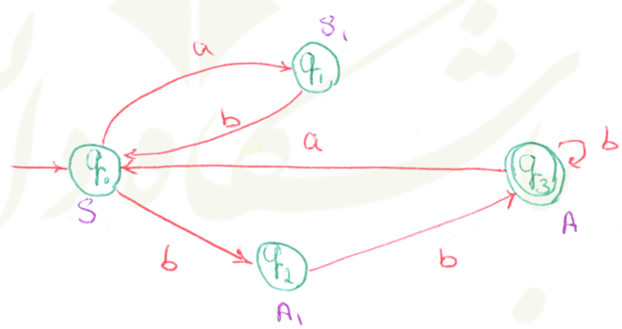
$S \to aS \mid bA \mid \lambda$

$S \to aS_1 \mid bA_1$

$S_1 \to bS$

$A_1 \to bA$

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



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

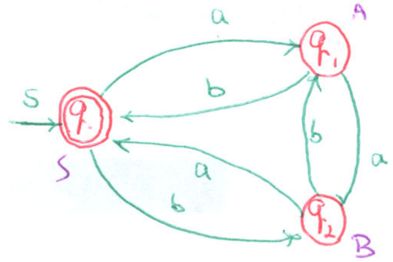
به ترتیب عکس آن چه که در اثبات نکته قبل گفته شد می توانیم یک گرامر خطی راست که برای زبان منظم L درست باشیم.

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

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

$A \to bS \mid aB$

$B \to bA \mid aS$



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

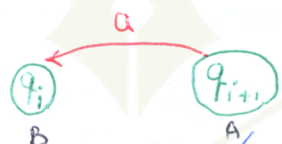
$$L(G_1) = L(G_2) \Rightarrow G_1 \equiv G_2$$

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، می توان یک NFA با تنها یک وضعیت پایانی برای آن ترسیم کرد. (Super Final) حال این NFA را به شرح زیر تغییر می دهیم:

- وضعیت شروع به پایانی و بالعکس تبدیل و جهت یال ها را هم معکوس می کنیم.
- حال به ازای هر State از این NFA یک متغیر پایانه متناظر می کنیم.

حال اگر داشته باشیم:



اکنون قاعده ی زیر را جایگزین می کنیم:

$A \to Ba$

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

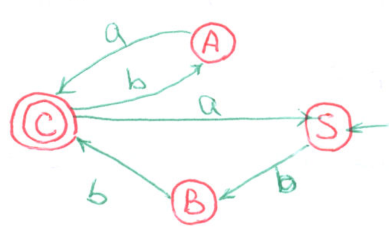
مثال: برای گرامر مثال قبل یک گرامر خطی چپ بیابید.

$S \to Bb \mid Sb$

$B \to Cb$

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

$A \to Ca$



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

۲۶۱

✓ در کامپایلرها چون که اسکنرها ورودی ها را از راست به چپ وارد می‌کنند، از گرامر خطی راست استفاده می‌کنند.

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

خواص تصمیم پذیری زبان های 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 تصمیم پذیر است. به عبارت دیگر الگوریتمی وجود دارد که تعیین می‌کند L تهی است یا خیر.

اثبات: چون L دارای نمایش استاندارد است لذا 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 تصمیم پذیر 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 loop از یک وضعیت به خودش یا به اجدادش داشته باشیم حتماً L نامتناهی است. الرهیچ loop ی نداشته باشیم، متناهی است.

نکته: مسئله تساوی دو زبان منظم که هر دو دارای نمایش استاندارد هستند یک مساله تصمیم پذیر است.

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

$$A \oplus B = \phi \Leftrightarrow A = B$$

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 تصمیم پذیر است اگر $l_1 \oplus l_2 = \phi$ آنگاه $l_1 = l_2$ و اگر تهی نباشد $l_1 \neq l_2$

مثال: به طور نشان به مسئله زیرمجموعه بودن $L_1 \subseteq L_2$ وقتی L_1، 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 $L_1 \subseteq L_2 \Leftarrow \phi = L_1 \setminus L_2$

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

تمام رشته‌ها به طول n متناهی (منظم) هستند در نتیجه می‌توان برای آن‌ها NFA ترسیم کرد. پس شکل یک زبان منظم با نمایش استاندارد می‌دهند. اسم این زبان را L_n می‌نامیم. 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 رشته‌ای به طول n ندارد.

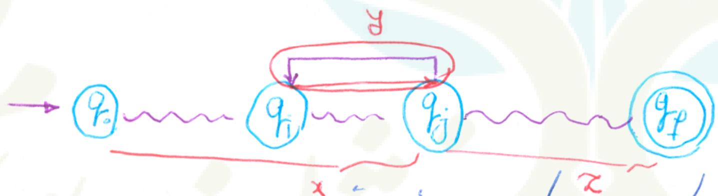
جلسه‌ی ششم

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 w متعلق به L را چنان یافت که $|w|\geq n$ باشد و برای هر تجزیه‌ی دلخواه حریف مانند $w=xyz$ که $|xy|\leq n$ ، $|y|\geq 1$ باشد، بتوان $i\in\mathbb{N}$ را چنان یافت که $xy^iz\notin L$ آن‌گاه L نامنظم است.

اثبات: فرض کنید L منظم باشد. در این صورت DFA ای چون M چنان وجود دارد 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 مسئله بایستی برای $n=|Q|+1$ رشته‌ای چون $w\in L$ چنان وجود داشته باشد که $|w|\geq n$. بنابراین در پذیرش رشته‌ی w توسط DFA ، M بایستی حالت زیر رخ دهد:



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

حال برای تجزیه‌ی مذکور $w=xyz$ به وضوح برای هر $i\in\mathbb{N}$ داریم $xy^iz\in L$

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

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

حتماً نیاز به حافظه داریم برای ذخیره‌ی این.

فرض کنید حریف عدد طبیعی n را داده باشد. قرار می‌دهیم $w=(a^nb^n)(a^nb^n)^R$

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

در نتیجه $|w|=4n\geq n$

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

۲۸۱

در نتیجه $y = a^k$ که $k \geq 1$ ، $xz = a^{n-k} b^{2n} a^n$ در نتیجه $xy^iz = y^ixz =$

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

پس L منظم نیست. دیدیم با پمپینگ نه هیچ گاه نباید یک باشد.

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

انواع سئوال های لم پمپینگ:
① صورت تهیه
② به را به دست آورید

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

به وضوح $w \in L$ است و $|w| = p \geq n$

حال برای تجزیه دلخواه حریف $a^p = w = xyz$ که $|xy| \leq n$ ، $|y| \geq 1$.

پس $y = a^k$ در نتیجه $xz = a^{p-k}$

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

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

با استفاده از خواص بستاری زبان ها می توان ثابت کرد یک زبان منظم نمی باشد

مثال: نشان دهید زبان $L = \{a^n b^m \mid n \neq m\}$ 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.

در نتیجه $\bar{L} \cap L(a^*b^*) = \{a^n b^n \mid n \geq 0\}$ نیز باید منظم باشد. در صورتی که چنین نیست. این تناقض نشان

می دهد 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 $L \cap L(a^*b^*) = \{a^n b^n \mid n \geq 0\}$ باید منظم باشد.

در حالی 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 $A\in V$ ، $x\in T^*$ در این صورت یک اشتقاق چپ از $xA\alpha$ آن است که چپ‌ترین غیرپایانه موجود در عبارت $xA\alpha$ یعنی A را با استفاده 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. 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 یک غیرپایانه است. برگ‌های آن یا پایانه است یا λ و یا 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 از طریق 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 دارای 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

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

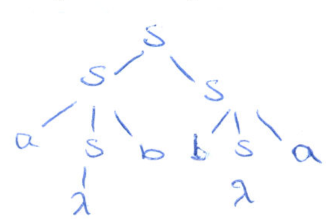
درخت اشتقاق درخت مرتبی است که

① اولاً درخت اشتقاق تجزیه بوده

② دوماً 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. $S \to SS \mid aSb \mid bSa \mid \lambda$

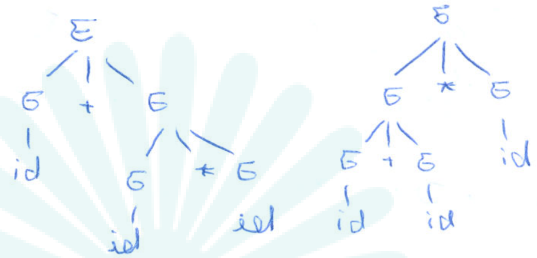


تعریف گرامر گنگ

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.

الف + الف * الف



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 در حقیقت 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 تنها زبان ذاتاً گنگ است که در کتاب معرفی شده است.

$S \to S_1 \mid S_2$

$S_1 \to AB$

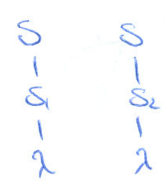
$B \to bBc \mid \lambda$

$A \to aA \mid \lambda$

$S_2 \to CD$

$C \to aCb \mid \lambda$

$D \to cD \mid \lambda$



۲۰۱

پوشش رشته‌ها (تجزیه رشته‌ها)

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

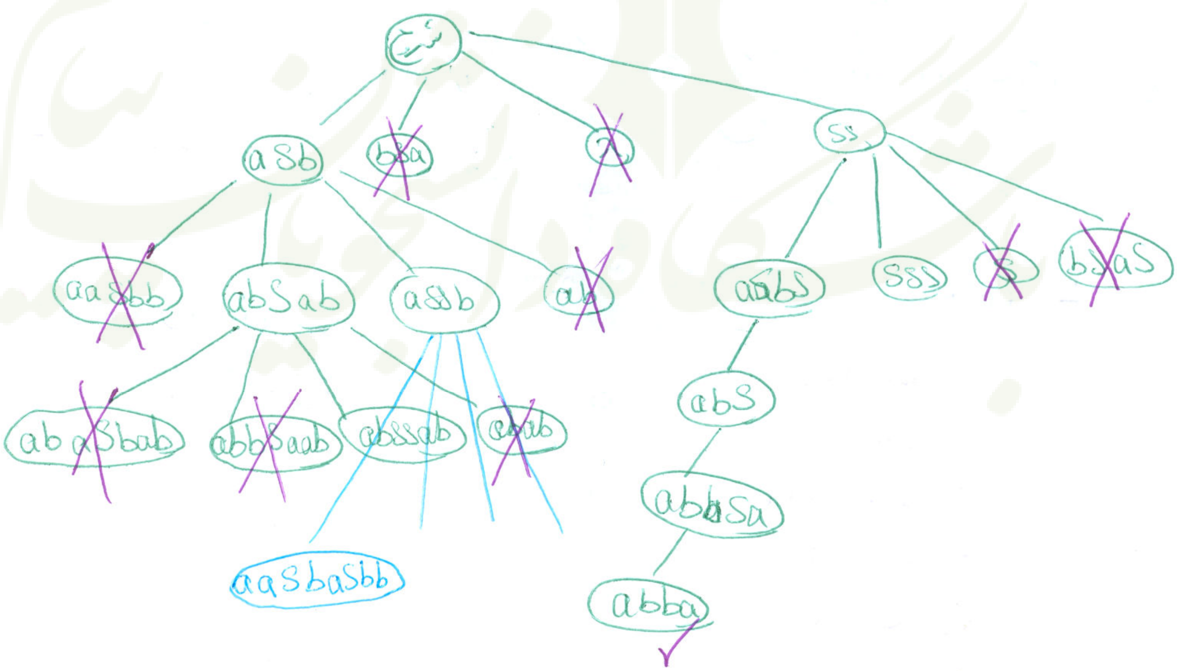
یکی از روش‌ها برای آن که نشان دهیم $w \in L(G)$ است روش Exhaustive Search یا جستجوی کامل است. (Brute Force)

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. سپس به ازای هر غیرپایانه‌ای به کار رفته در چپ‌ترین موقعیت هر رشته روند قبل 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، پیدا می‌کند



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 λ و فاقد قاعده‌ی unit 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: $|P|^0+|P|^1+\cdots+|P|^{2|w|-1}$
$= O(|P|^{2|w|-1})$

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

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 خوبی اند زیرا در پذیرش رشته‌ها نوع خاصی را پیمایش می‌کنند.
مسئله‌ی تعلق یا عدم تعلق یک رشته به گرامر ساده در زمان $O(|w|)$ امکان پذیر است.

تصمیم 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 جانشینی B قواعد نظیر $B \to y_1 \mid y_2 \mid \cdots \mid y_n$ باشد. گرامری که $\hat{G}$ از حذف قاعده‌ی $A \to \alpha B\beta$ و جانشینی قواعد $A \to \alpha y_1\beta \mid \alpha y_2\beta \mid \cdots \mid \alpha y_n\beta$ به دست می‌آید، معادل با گرامر G است.

۲۱۱

شکل : $S \to aAB \mid bAa$ $\quad A \to aAb \mid \lambda$

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

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

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

$A \to aAb \mid \lambda$

تعریف متغیر مفید — ابلعدی

متغیر پایانی $A \in V$ از گرامر مستقل از متن $G=\langle V,T,S,P \rangle$ را مفیدی نامیم هرگاه آن متغیر بایند دست کم مصداق 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 — useless

متغیر پایانی A را نامفید نامیم، هرگاه مفید نباشد. به عبارت دیگر یا از متغیر پایانی شروع نتوان به آن رسید یا از آن نتوان به رشته ای از زبان رسید.

تعریف قاعده ی نامفید

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

هم چنین قاعده ای 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، متغیر نامفیدی ندارد.

نکته : کافی است تمام متغیرها و قواعد نامفید را از گرامر حذف کنیم. گرامر به دست آمده، فاقد هر متغیر و قاعده ی 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) ✓ $a^{2}a^{2k}b^{k}b^{2} \cup b^{L}a^{L+1} \quad k \geq 0 , L \geq 1$

$S \to aAb \mid bBa$

$A \to aaAb \mid ab$

$B \to bBa \mid a$

$A \overset{*}{\Rightarrow} a^{2k}Ab^{k} \overset{*}{\Rightarrow} a^{2k}aAbb^{k} \Rightarrow aa^{2k}Ab^{k}b$

$A \Rightarrow aa^{2k}Ab^{k}b$

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

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

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

حذف قوانین λ (λ-Free)

هرگاه $G = \langle V, T, S, P \rangle$ یک گرامر مستقل از متن باشد. در این صورت می‌توان با جایگذاری قوانین λ گرامر G را به شکل λ-Free درآورد که حداکثر یا یا به شرط تعدی λ داشته باشد.

مثال:

$S \to AB$

$A \to aAb \mid \lambda$

$B \to cBd \mid \lambda$

$$\begin{tikzcd} S \arrow[r] \arrow[d] & A \\ B & \end{tikzcd}$$

با حذف λ باید $S \to \lambda$ می‌شود.

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

$A \to aAb \mid ab$

$B \to cBd \mid cd$

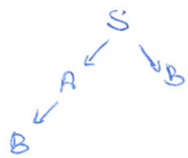
۲۲۱

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، چرا که زبان λ را نمی‌پذیرد.

تعریف قوانین یکه

یک NT به یک NT دیگر برود.

حذف قوانین یکه

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

1. قواعدی که یکه نیستند
2. قواعد یکه

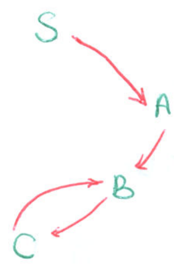
مثال:

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

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

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

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



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 طرف قواعد پایین

دقت loop زده دیگر تکرار نمی کنیم

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

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

$B \to aA \mid b$

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

در نتیجه

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

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

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

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

۲۰۷

قضیه

به ازای هر گرامر مستقل از متن که $\lambda \notin L(G)$ آنگاه گرامر مستقل از متن وجود دارد که فاقد قوانین λ، یکه و نامفید است و به علاوه این گرامر، معادل با گرامر اولی است. ابتدا λ را حذف می‌کنیم، سپس یکه را حذف می‌کنیم. سپس نامفیدها را حذف می‌کنیم.

جلسه‌ی هفتم

فرم نرمال گریباخ

گوییم گرامر مستقل از متن $G=\langle V,T,S,P\rangle$ 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 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.

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

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

۳۴۱

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 می تواند در زمان $O(|w|^3)$ معین کند که رشته ی w متعلق به $L(G)$ هست یا نه!

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

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

به وضوح $w_{ii}=a_i$ ، $w_{1n}=w$

هم چنین قرار می دهیم

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

مطابق الف مذکور واضح است که:

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

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

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

به زبان دیگر

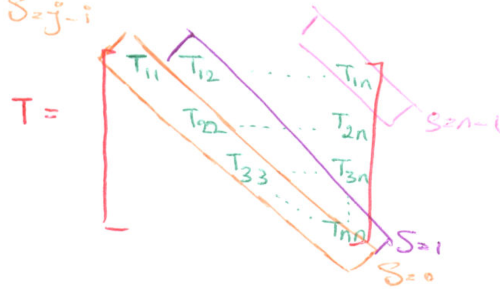
$$T[i,j]=\begin{cases}\{A\in V \mid A\to a_i\} & (i=j) \quad S=0,\ i=1,2,\ldots,n\\ \bigcup_{k=i}^{j-1}\{X\in V \mid X\to YZ,\ Y\in T[i,k],\ Z\in T[k+1,j]\} & (j>i) \quad 1\le S\le n-1,\ i=1,2,\ldots,n-S\end{cases}$$

تعریف: $S=j-i$

$j-i=S$

$j=S+i\le n$

$\Rightarrow i\le n-S$



for $i = 1$ to n do

$\Theta(n)$ $\quad T[i,i] \leftarrow \{X \in V \mid X \to a_i\}$

for $s = 1$ to $n-1$ do

$\Theta\left(\sum_{s=1}^{n-1} \sum_{i=1}^{n-s} s\right)$ $\quad$ for $i \leftarrow 1$ to $n-s$ do

$\quad\quad j = i + s$

$\quad\quad T[i,j] = \bigcup_{k=i}^{j-1} \{X \mid X \to YZ,\ Y \in T[i,k],\ Z \in T[k+1,j]\}$

return $(S \in T[1,n])$

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

$= \Theta(n^3)$

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

✓ CLR و ... از رتبه‌ی خطی هستند ولی پیچیدگی‌شان بیشتر است

اما زبان‌های بیشتری را پوشش می‌دهد.

زبان‌های مستقل از متن

خواص بستاری

لم پمپینگ زبان‌های 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:

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

آن گاه 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 بپذیرد. طبق مطالب خوانده شده می‌دانیم که اگر L مستقل از متن باشد $L - \{\lambda\}$ نیز مستقل از متن است و بالعکس.

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

هم‌چنین می‌توان فرض کرد که گرامر مستقل از متن G فاقد تولید λ، یکه و بی‌فایده باشد. یعنی گرامر غیرانقباضی است گرامر مذکور.

طبق فرض اگر n را به گونه‌ای انتخاب کنیم که n بزرگتر از تعداد غیرپایانه‌های گرامر باشد، در این صورت بایستی رشته‌ای چون $w \in L(G)$ چنان وجود داشته باشد که $|w| \geq n$ و به علاوه $S \stackrel{*}{\Rightarrow} w$. اما اگر در مسیر پذیرش w از یک غیرپایانه حداکثر 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.

$S \overset{*}{\Rightarrow} uAz \overset{*}{\Rightarrow} uvAyz \overset{*}{\Rightarrow} w$

ثابت پامپینگ = تعداد Non Terminal ها (طول بلندترین رشته)

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

در نتیجه : $S \overset{*}{\Rightarrow} uv^ixy^iz \quad \forall i \in \mathbb{N}$

در نتیجه تناقض با فرض است.

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

1. تعداد واژه های زبان در T
2. ✓ تعداد واژه های نحوی در V، نسبت به تعداد متغیر پایانه ها
3. تعداد قواعد تولید در P
4. هیچ کدام

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

فرض کنید حریف عدد طبیعی n را داد، با ثابت قراردادی هم $w=a^nb^nc^n \in L$ به طوری که $|w|=3n\geq n$

حال برای تجزیه دلخواه حریف $a^nb^nc^n=uvxyz$ که $|vxy|\leq n$، $|vy|\geq 1$ داریم:

در نتیجه ممکن است یکی از حالات زیر رخ دهد:

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

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

با قرار دادن $i=0$ نتیجه می شود که

$uv^0xy^0z \notin L$

۳۶۱

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

لم پامپینگ زبان های 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 $|uvyz|\leq n$ ، $|vy|\geq 1$ بتوان یک $i\in N$ 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 $|uvyz|\leq n$ ، $|vy|\geq 1$ داریم:

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

با قرار دادن $i=0$ در رشته uv^ixy^iz حداقل یکی از تعداد a ها کاسته می‌شود درحالی 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، بستار ستاره ای بسته هستند

در حالی که تحت اشتراک و عمل متمم 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:

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

هم چنین 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

$\dot{S} \notin V_1 \cup V_2$ ، $\dot{T} = T_1 \cup T_2$ ، $\dot{P} = P_1 \cup P_2 \cup \{\dot{S} \to S_1 S_2\}$

در این صورت:

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

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

به طوری که $S^* \notin V_1$ ، $\overset{*}{P} = P_1 \cup \{\overset{*}{S} \to S_1 \overset{*}{S} | \lambda\}$

در این حالت:

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

۳۷۱

حال نشان می دهیم که خانواده های زبان های مستقل از متن نسبت به اشتراک بسته نیستند:

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

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

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

نشان می دهیم که متمم زبان های مستقل از متن، لزوماً مستقل از متن نیستند

فرض کنید چنین نباشد. یعنی مکمل هر زبان مستقل از متن، مستقل از متن باشد. درنتیجه برای L_1 و L_2 زبان های تعریف شده در فوق. آن گاه بایستی $\bar{L_1}$ و $\bar{L_2}$ مستقل از متن باشند چون اجتماع دو زبان مستقل از متن، مستقل از متن است پس بایستی $\bar{L_1} \cup \bar{L_2}$ مستقل از متن باشد.

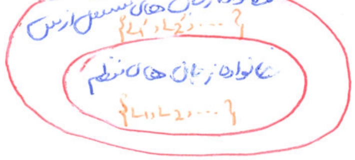
دراین صورت مکمل زبان مذکور باید مستقل از متن باشد. پس $L_1 \cap L_2$ مستقل از متن است درحالی که مستقل از متن نیست.

کاربرد خواص بستاری در مشخص کردن مستقل از متن بودن یا نبودن زبان های داده شده

نکته: زبان های مستقل از متن تحت اشتراک با زبان های 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. آن گاه $\underbrace{L}_{\text{مستقل از متن}} \cap \underbrace{L(a^*b^*c^*)}_{\text{منظم}} = \{a^n b^n c^n \mid n \geq 0\}$ بایستی مستقل از متن باشد که این یک تناقض است.



نکته ی ۲۶۷ : 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 $A\to\alpha\in P^r \Longleftrightarrow A\to\alpha\in P$

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

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

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:

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

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

خواص بستاری زبان های خطی

- زبان های خطی تحت اجتماع بسته هستند.
- تصویر هم ریختی زبان های خطی، خطی است.
- معکوس زبان های خطی، خطی است.
- الحاق دو زبان خطی، خطی نیست.

مثال:

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

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

- بستار ستاره ای زبان های خطی، لزوماً خطی نیست.
- زبان های خطی تحت اشتراک و تمم پذیری بسته نیستند.

مثال:

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

$$\begin{cases} S \to Sc \mid A \\ A \to aAb \mid \lambda \end{cases}$$

$$S \overset{*}{\Rightarrow} Sc \overset{*}{\Rightarrow} Sc^m \overset{*}{\Rightarrow} Ac^m \overset{*}{\Rightarrow} a^n b^n c^m$$

مستقل از متن

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

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

ذاتاً مبهم

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

- اشتراک دو زبان خطی ممکن است حتی ذاتا نیز 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 آن گاه $L_1 . L_2$ 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 می پذیرد. فرض کنید غیرپایانه ی شروع این گرامر A باشد. هم چنین چون L_2 یک زبان منظم است، پس برای آن یک گرامر خطی چپ وجود دارد که آن را می پذیرد.

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

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

$S \to Sc \mid Ac$
$A \to aAb \mid ab$

$B \to \alpha^{\epsilon T^*} \Rightarrow B \to A\alpha$

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

خواص تصمیم پذیری زبان های مستقل از متن

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

اثبات: گرامر G را ساده، حذف قواعد لاندا و ناهدفمند می کنیم. اگر غیرپایانه ی شروع (S) ناهدفمند باشد، یعنی $L(G)$ تهی است.

۳۹

نکته: اگر G یک گرامر مستقل از متن باشد، الگوریتمی وجود دارد که تعیین می کند L(G) تهی است یا خیر.

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

در این صورت برای هر غیر پایانه مانند A با تشکیل درخت های جایگذاری مشخص می شود که آیا A دارای ساختار 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 می توان گفت که $w \in L(G)$ یا $w \notin L(G)$

چون G مستقل از متن است پس می توان برای آن یک گرامر چامسکی ارائه کرد. مسئله ی تعلق یک رشته به گرامر G یک مسئله ی تصمیم پذیر است.

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

تمام رشته های به طول n روی حروف الفبای T 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: مسئله ی 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 می شوند مسئله ی $L(G_2) \subseteq L(G_1)$ یک مسئله ی تصمیم پذیر 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.

جلسه ی هشتم

ماشین های پشته ای NPDA

ماشین پشته ای 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 مجموعه ای متناهی و ناتهی از وضعیت های واحد کنترل است

ب. Σ مجموعه ای از حروف الفبای نوار است (متناهی و ناتهی)

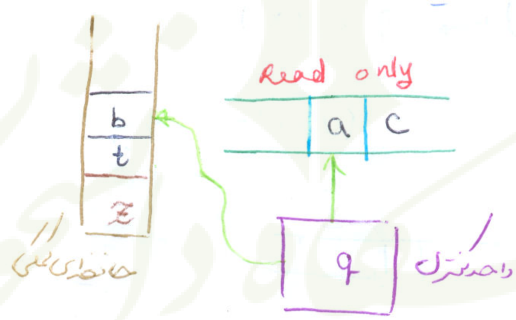
ج. Γ مجموعه ای متناهی و ناتهی از علائم پشته است که به آن گاهی الفبای پشته نیز گفته می شود

د. تابع δ از $\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to \{A \subseteq Q \times \Gamma^* \mid |A| < \infty\}$ تابع انتقال است

ه. $q_0 \in Q$ وضعیت آغازین است

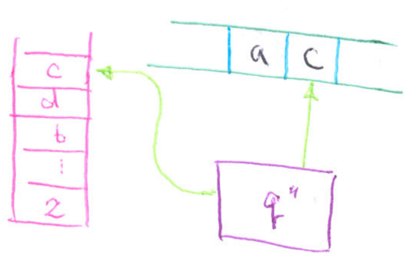
و. $z \in \Gamma$ علامت شروع پشته است

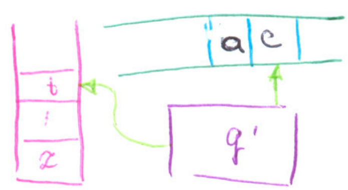
ز. $F \subseteq Q$ مجموعه وضعیت های پایانی است.



$$\delta(q, a, b) = \{(q', \lambda), (q'', cd)\}$$

هد همیشه به سمت راست یا نوار به سمت چپ حرکت می کند.



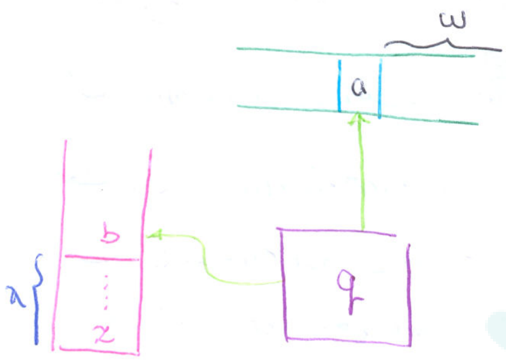


۴۱

* اگر بخواهیم رشته ی $L=\{a^nb^n \mid n\geq 0\}$ را بپذیریم .

وقتی a می خوانیم ، a را در پشته بریزیم به ازای هر a خوانده شده هرجه b می خوانیم یک a برمی داریم از پشته .

configration پیکربندی



(q, aw, bx)

✓ خروجی δ زوج است زیرا تغییراتی که می تواند اعمال کند فقط روی پشته و q است . روی نوار نمی تواند تغییری ایجاد کند.

تعریف حرکت

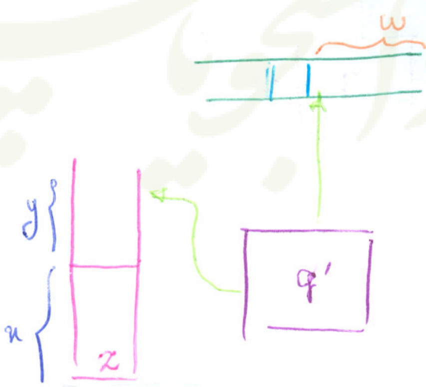
ماشین پشته ای غیرقطعی M از پیکربندی (q, aw, bx) به پیکربندی (q', w, yx) حرکت می کند و می نویسیم :

$$(q, aw, bx) \mapsto (q', w, yx)$$

اگر و تنها اگر :

$$(q', y) \in \delta(q, a, b)$$

باشد .



✓ همیشه یک حرف را برداشته و به جای آن رشته ای را جایگزین می کند.

$\overset{*}{\longmapsto}$

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.

$$L(M)=\{\omega\in\Sigma^* \mid (q_0,\omega,z)\overset{*}{\longmapsto}(q_f,\lambda,u), u\in\Gamma^*, q_f\in F\}$$

نکته: رشته‌ی λ توسط NPDA ی پذیرفته می‌شود که وضعیت شروع آن نهایی باشد.

$$(q_0,\lambda,z)\overset{*}{\longmapsto}(q_0,\lambda,z)$$

مثال: یک NPDA طراحی کنید که زبان $L=\{a^nb^n \mid n\geq 0\}$ را بپذیرد.

$\delta(q_0,a,z)=\{(q_0,az)\}$

$\delta(q_0,a,a)=\{(q_0,aa)\}$

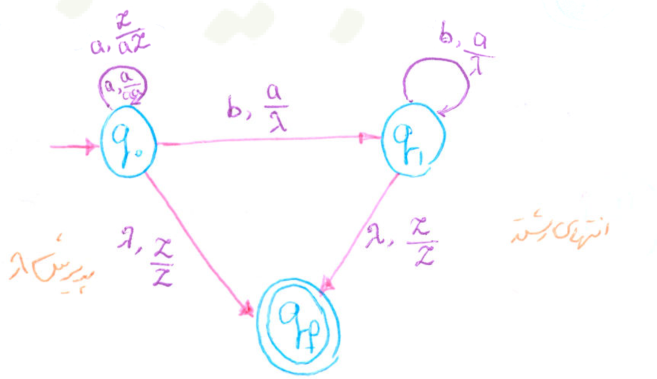
$\delta(q_0,b,a)=\{(q_1,\lambda)\}$ ⇝ وضعیت را عوض می‌کنیم تا دیگر اجازه ندهیم a دیگری بیاید.

$\delta(q_1,b,a)=\{(q_1,\lambda)\}$

$\delta(q_1,\lambda,z)=\{(q_f,z)\}$ ⇝ کلیه‌ی a ها که push شده accept.

$\delta(q_0,\lambda,z)=\{(q_f,z)\}$ ⇝ λ را می‌پذیرد

ترسیم دیاگرام



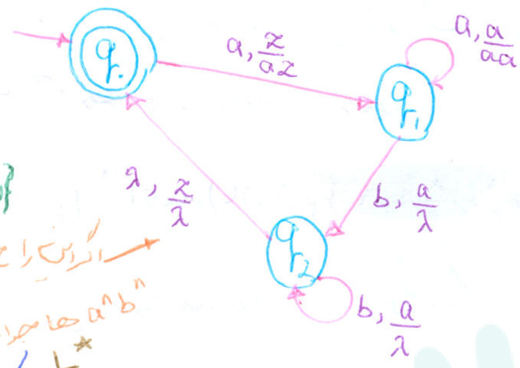
غیر قطعی

$\delta(q_0,\lambda,z)=\{(q_f,z)\}$

$\delta(q_0,a,z)=\{(q_0,az)\}$

۴۲،

* در حالت قبل می‌خواهیم وضعیت q_0 را پایانی کنیم.



$\left\{\prod_{i=1}^{k}(a^n b^n) \mid n \in N\right\}$

اگر این را z بنامیم حاصلضرب $a^n b^n$ ها جواب خواهد بود.

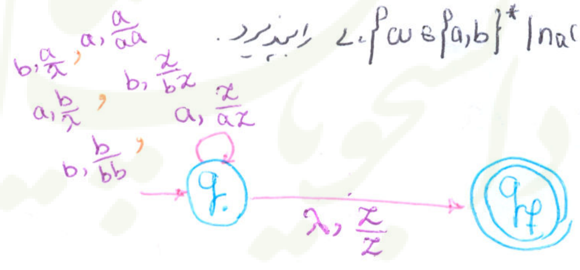
L^*

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 خانواده زبان‌های پذیرفته شده توسط گرامرهای مستقل از متن دقیقاً خانواده زبان‌های پذیرفته شده توسط NPDA ها هستند.

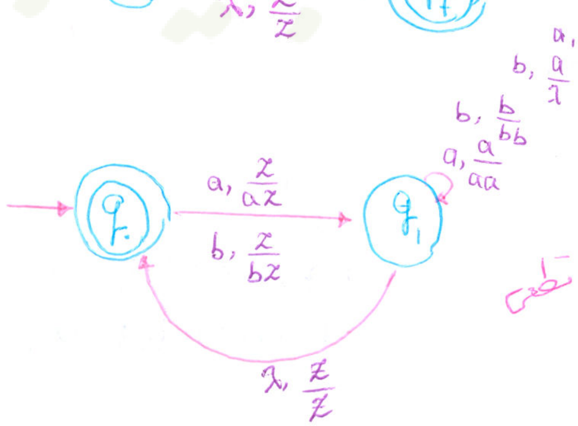
نکته: برای هر NPDA یک NPDA با سه وضعیت وجود دارد که زبان آن NPDA را می‌پذیرد.

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.

مسئله حاوی Back Tracking به این شکل 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 کند 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

$\delta(q_0, a, z) = \{(q_0, az)\}$

$\delta(q_0, b, z) = \{(q_0, bz)\}$

$\delta(q_0, a, a) = \{(q_0, aa), (q_1, \lambda)\}$

$\delta(q_0, b, b) = \{(q_0, bb), (q_1, \lambda)\}$

$\delta(q_0, a, b) = \{(q_0, ab)\}$

$\delta(q_0, b, a) = \{(q_0, ba)\}$

$\delta(q_1, a, a) = \{(q_1, \lambda)\}$

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

$\delta(q_1, \lambda, z) = \{(q_f, z)\}$

$\delta(q_0, \lambda, z) = \{(q_f, z)\}$

۴۲۱

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 حتماً یک NPDA باشد و به علاوه

۲- برای هر $a\in\Sigma$ ، $b\in\Gamma$ داشته باشیم

الف. $|\delta(q,a,b)|\le 1$, $|\delta(q,\lambda,b)|\le 1$

ب. $\delta(q,\lambda,b)\neq\emptyset \Rightarrow \forall c\in\Sigma \quad \delta(q,c,b)=\emptyset$

جلوی غیر قطعی شدن را گرفتیم

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 حتماً یک زبان مستقل از متن می‌باشد

زبان‌های مستقل از متنی وجود دارد که برای آن‌ها نمی‌توان هیچ DPDA ترسیم کرد.

مثال:

$L=\{a^nb^n \mid n\ge 0\}\cup\{a^nb^{2n} \mid n\ge 0\}$

$S\to S_1|S_2$

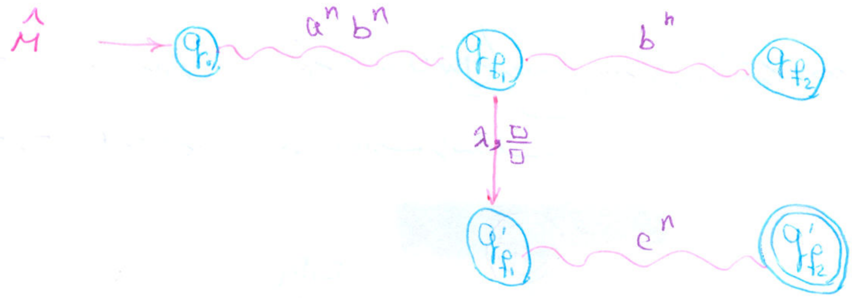
$S_1\to aS_1b|\lambda$

$S_2\to aS_2b^2|\lambda$

زبان L یک زبان مستقل از متن است پس یک NPDA وجود دارد که L را می‌پذیرد.

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 رسیده ایم.

درحالی که این زبان مستقل از متن نیست.

نکته: خانواده زبان های مستقل از متن تحت اجتماع بسته نیستند.

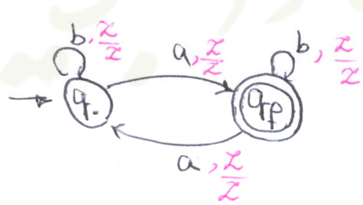
نکته: زبان های مستقل از متن ذاتا لک نیستند. زیرا برای پذیرش آن ها یک راه بیشتر نداریم.

نکته: هر زبان منظم یک زبان مستقل از متن می باشد. زیرا زبان های منظم توسط DFA ها پذیرفته می شوند و ساختار DFA را می توان به عنوان یک ساختار ماشین پشته ای در نظر گرفت که همواره مقدار Z را در هر انتقال به جای Z می برد.

مثال:

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

تبدیل به ماشین پشته ای

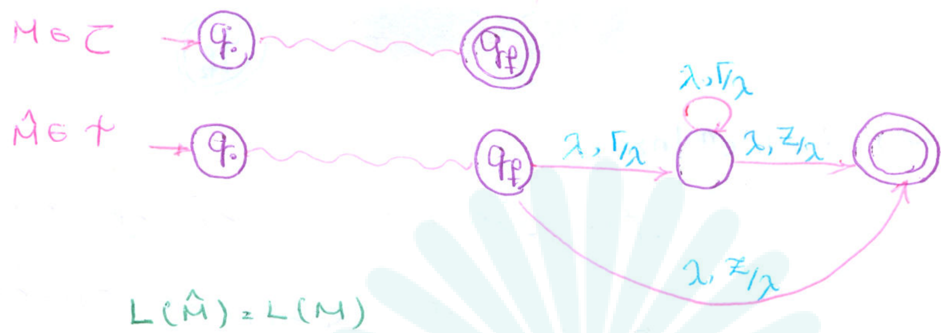


نکته: هرگاه ψ را خانواده ی زبان های مستقل از متنی در نظر بگیریم که توسط ماشین پشته ای ناقطعی با خالی شدن پشته رشته ها را بپذیرند و Γ را خانواده ی زبان هایی در نظر بگیرید که صرفا با یک ماشین پشته ای ناقطعی پذیرفته می شوند.

$$\psi \subseteq \Gamma$$

۴۴۱

نکته: حال فرض کنید ماشین پشته‌ای وجود دارد که زبان را می‌پذیرد و 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 نتیجه مطلوب را به دست آورد.



$L(\hat{M}) = L(M)$

نکته:

ماشین پشته‌ای قطعی 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 یا 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 را که پیشوند دیگر رشته‌ها را نمی‌پذیرد.

نکته: زبان یک آتوماتای غیرقطعی غیرمبهم است ولی ممکن است برای یک زبان غیرمبهم هیچ آتوماتای قطعی موجود نباشد.

$\{a^n b^n \mid n \geq 0\} \cup \{a^n b^{2n} \mid n \geq 0\}$

جلسه‌ی نهم

ماشین‌های 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 وضعیت‌های واحد کنترل است.

ب) Σ مجموعه‌ای متناهی و ناتهی از حروف الفبای است.

ج) Γ مجموعه حروف الفبای نوار است که یک مجموعه متناهی و ناتهی است، به علاوه

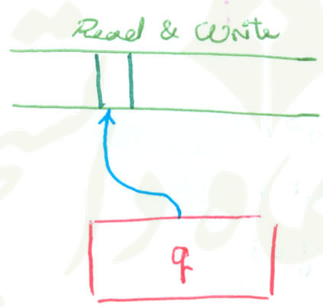
$\Sigma \subseteq \Gamma \setminus \{\square\}$

د) تابع $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ یک تابع انتقال نامیده می‌شود.

هـ) $q_0 \in Q$ وضعیت آغازین واحد کنترل است.

و) $\square \in \Gamma$ علامت خاصی است که نشان‌دهنده‌ی جای خالی است.

ز) $F \subseteq Q$ مجموعه از وضعیت‌های 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.

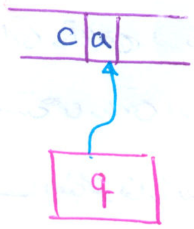
۲- یک هد خواندن و نوشتن که به هر میزان که بخواهد می‌تواند به چپ یا راست حرکت نماید

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

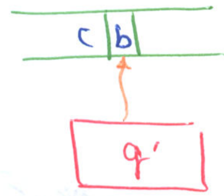
421

به چه وضعیت جدیدی انتقال یابد.

ماشین تورینگ استاندارد از هیچ حافظه‌ی کمکی برای ذخیره کردن، نگهداری مقادیر ورودی خروجی استفاده نمی‌کند و صرفاً همه‌ی عملیات روی نوار انجام می‌شود.

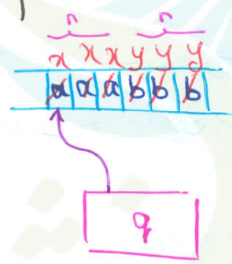


$$\delta(q,a) = (q', b, L)$$



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.

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

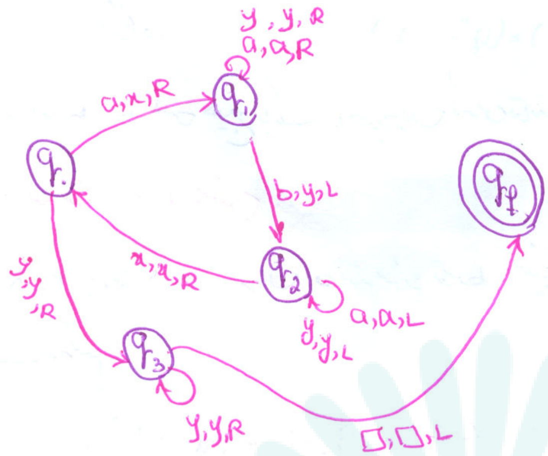


n بار طول n را حرکت می‌کند

$\Leftarrow \Theta(n^2)$

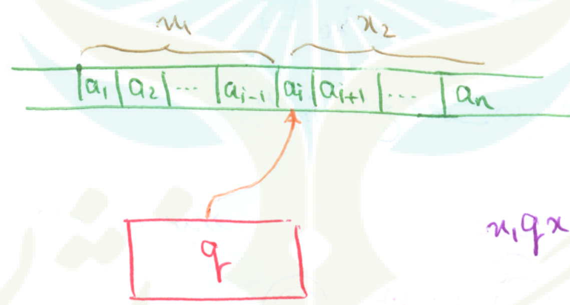
$$\delta(q_0, a) = (q_1, x, R)$$
$$\delta(q_1, a) = (q_1, a, R)$$
$$\delta(q_1, b) = (q_2, y, L)$$
$$\delta(q_2, a) = (q_2, a, L)$$
$$\delta(q_2, x) = (q_0, x, R)$$
$$\delta(q_1, y) = (q_1, y, R)$$
$$\delta(q_2, y) = (q_2, y, L)$$
$$\delta(q_0, y) = (q_3, y, R)$$
$$\delta(q_3, y) = (q_3, y, R)$$
$$\delta(q_3, \square) = (q_f, \square, L)$$

هر ماشین تورینگ را می توان با یک دیاگرام انتقال نمایش داد: (دقیقاً مشابه آنچه که در مورد DFA و NFA و ماشین پشته ای گفته شد)



تعریف پیکربندی یا توصیف

یک توصیف از ماشین تورینگ به شکل زیر را با $a_1a_2\cdots a_{i-1}\,q\,a_ia_{i+1}\cdots a_n$ نمایش می دهند.



x_1qx_2

تعریف حرکت

یک حرکت از ماشین تورینگ می تواند حرکت به سمت راست یا چپ روی نوار باشد؛ به عبارت دیگر هد نوار به یک خانه سمت 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 می یابد و می نویسیم:

$$a_1a_2\cdots a_{i-1}qa_ia_{i+1}\cdots a_n \vdash a_1a_2\cdots a_{i-1}bq'a_{i+1}\cdots a_n$$

$\delta(q,a_i)=(q',b,R)$ هرگاه.

همچنین گوییم ماشین تورینگ M از پیکربندی $a_1a_2\cdots a_{i-1}qa_ia_{i+1}\cdots a_n$ به پیکربندی

431 $a_1a_2\cdots a_{i-2}q''a_{i-1}ca_{i+1}\cdots a_n$ دارای یک حرکت چپ بود، است می نویسیم:

$$a_1a_2\cdots a_{i-1}\,q\,a_i\,a_{i+1}\cdots a_n \vdash a_1a_2\cdots a_{i-2}\,q''\,a_{i-1}\,c\,a_{i+1}\cdots a_n$$

هرگاه $\delta(q,a_i)=(q'',c,L)$

✓ همچنین منظور از $\vdash^*$ یعنی هیچ یا یک یا چند حرکت که می تواند ترکیبی از حرکت های چپ و راست باشد.

تعریف زبان پذیرفته شده توسط ماشین تورینگ

$M=\langle Q,\Sigma,\Gamma,\delta,q_0,\square,F\rangle$ زبان پذیرفته شده توسط ماشین تورینگ M را با L(M) نمایش داده و به شکل زیر تعریف می کنیم:

$$L(M)=\{w\in\Sigma^* \mid q_0w \vdash^* x_1q_fx_2,\ q_f\in F,\ x_1,x_2\in\Gamma^*\}$$

* قرارداد: رشته ی λ پذیرفته نمی شود.

درحالت final باید halt کند و دیگر به جلو نرود.

$$q_0a^nb^n \vdash^* x^ny^nq_f\square$$

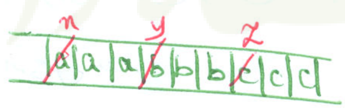
با q_f به هیچ وضعیت دیگری نمی رود.

مثال: ماشین تورینگی طراحی کنید که $L(00^*)$ بپذیرد.

مثال:

$\{a^nb^nc^n \mid n\geq 1\}$

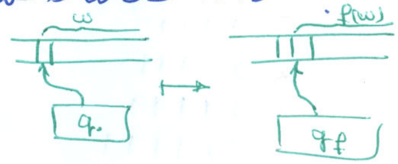
$O(n^2)$



تعریف محاسبه پذیر

گوییم تابع $f: D \to \mathbb{R}$ 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:

$$q_0 w \overset{*}{\vdash}_M q_f f(w)$$



مثال: نشان دهید جمع اعداد صحیح مثبت محاسبه پذیر تورینگ است.

برای نمایش x از $w(x) \in \{1\}^+$ استفاده می کنیم بطوری که $|w(x)| = x$

$$q_0 w(x) 0 w(y) \overset{*}{\vdash}_M q_f w(x+y) 0$$

$\delta(q_0, 1) = (q_0, 1, R)$

$\delta(q_0, 0) = (q_1, 1, R)$

$\delta(q_1, 1) = (q_1, 1, R)$

$\delta(q_1, \square) = (q_2, \square, L)$

$\delta(q_2, 1) = (q_3, 0, L)$

$\delta(q_3, 1) = (q_3, 1, L)$

$\delta(q_3, \square) = (q_f, \square, R)$

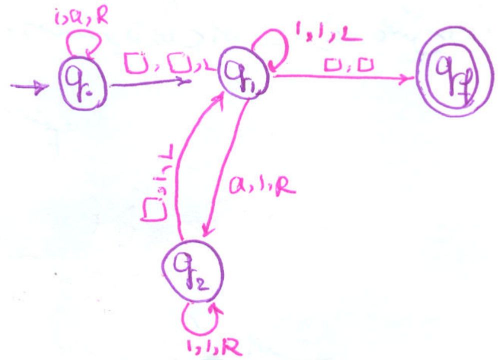
مثال: ماشین تورینگی طراحی کنید که نشان دهد $f(w) = ww$ که $f: \{1\}^+ \to \{1\}^+$ محاسبه پذیر است.

$$w \in \{1\}^+ \quad q_0 w \overset{*}{\vdash} q_f ww$$

ابتدا تمام یک ها را در رشته w به کاراکتری مانند a تبدیل می کنیم سپس به سمت راست ترین موقعیت رفته و یک کاراکتر را با یک جایگزین کرده، و به سمت راست رشته رفته و blank را به یک تبدیل می کنیم. این روند را آن قدر تکرار می کنیم تا تمام a ها به یک تبدیل شوند.

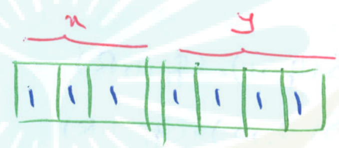
۹۹،

1 111
2 aaa□
3 aa1□
4 aa11
5 a111
6 a1111
7 11111
8 111111



طراحی ماشین های تورینگ برای توابع پیچیده تر

$$f(x,y)=\begin{cases} q & x\le y \\ q' & x>y \end{cases}$$



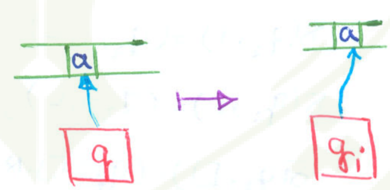
* if a then q_i else q_j

$\forall q \quad \delta(q,a)=(q'_i,a,R)$

$\forall c\in\Gamma \quad \delta(q'_i,c)=(q_i,c,L)$

$\forall b\neq a \quad \delta(q,b)=(q'_j,b,R)$

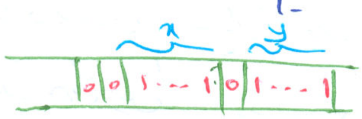
$\forall c\in\Gamma \quad \delta(q'_j,c)=(q_j,c,L)$



✓ در ابتدا به q'_i می رویم که اگر روی a بود دوباره جلو برود و بتواند بازگردد.

مثال: می خواهیم نشان دهیم حاصل ضرب دو عدد صحیح مثبت محاسبه پذیر تورینگ است.

برای این منظور x و y را به شکل زیر روی نوار قرار می دهیم.



سپس به ناحیه ی y می رویم یک ۱ را به a تبدیل می کنیم. سپس به سمت چپ ترین ناحیه رفته صفر را به 0x تبدیل می کنیم. این روند را آن قدر تکرار می کنیم تا هیچ یکی در ناحیه ی y باقی نماند. در این صورت

سمت چپ ترین عددی که قرار گرفته است $x*y$ است.

$3*2$

|0|0|111|0|11

0111 0|111|0|a1

0111111 0|111|0 aa

مرتبه: $x*y$

$2*3$

00|11|0|111

0 110 11 0 a11

0 1111 0 11 0 aa1

0111110 110 aaa

مثال:

$$f(x,y) = \begin{cases} x+y & x \leq y \\ 0 & x > y \end{cases}$$

ابتدا با ماشین مقایسه، این را چک کرده

با چند ماشین دیگر می توان پیاده سازی کرد

اگر $x \leq y$ ⇐ x را با y جمع

اگر $x > y$ ⇐ صفر

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 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 ضرورتاً 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.

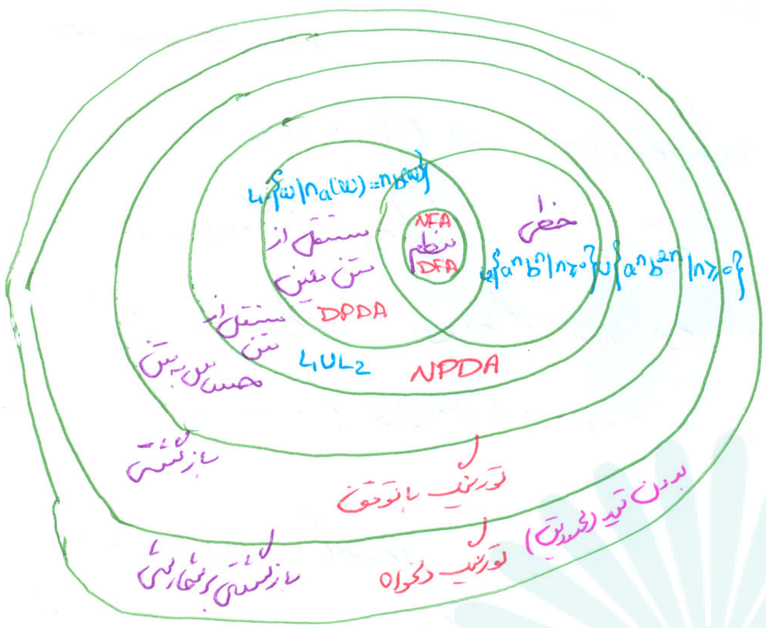
نکته

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 yes یا no باشد. به عبارت دیگر مسائل تصمیم پذیری که توسط یک ماشین تورینگ پذیرفته می شوند، جزء کلاس زبان های بازگشتی قرار می گیرند. 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.

45

چند 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 تهی بودن اشتراک 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 می کند: $L_1 \le L_2$

هرگاه:

$L_1=\{x \mid \exists f \;\; f(x)\in L_2\}$

به طوریکه f محاسبه پذیر تورینگ است

به عبارت دیگر

f محاسبه پذیر تورینگ

$x\in L_1 \Leftrightarrow f(x)\in L_2$

نتیجه: اگر L_2 تصمیم پذیر باشد و $L_1 \le L_2$ باشد، آنگاه L_1 نیز تصمیم پذیر است.

نتیجه: اگر L_1 تصمیم ناپذیر باشد و $L_1 \le L_2$ باشد، آنگاه L_2 نیز تصمیم ناپذیر است.

* تورینگ غیرقطعی در هر configuration چند عمل می تواند انجام دهد.

$\delta: Q\times\Gamma \to 2^{Q\times\Gamma\times\{L,R\}}$

Linear Bounded Automata LBA

ماشین های خطی کران دار است. یک ماشین تورینگ غیرقطعی است که محدوده ی پردازشی رشته ها در آن برحسب طول رشته ورودی مشخص است.

$(q,[,R)\in\delta(q,[)$ $\qquad (q'',],L)\in\delta(q,])$

LBA ها زبان های حساس به متن را می پذیرند.

نکته: هنوز ثابت نشده است که آیا LBA ها 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
۱۳۸۱ تا ۱۳۸۸

نظریه زبانها و ماشینها

۵۵ـ زبان $L=\{a^n b^m c^n d^m \mid n,m \ge 0\}$ مفروض است. کدام گزینه صحیح است؟ ($\bar{L}$ زبان مکمل L است)

نزول 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. ✓ یک قدم 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 $L(G_1)=L(G_2)$ ؟ ✓

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؟

مستقل از متن

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

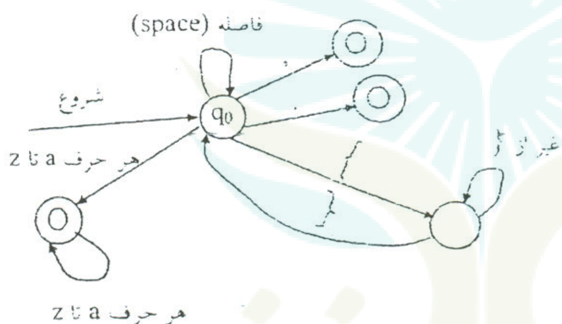
۹۲ـ در یک اتومات پوش‌دان (Push Down) طول پشته (Stack) حداکثر ۵ است. زبانهایی که این اتومات می‌تواند بپذیرد در کدام مجموعه زبان قرار می‌گیرد؟

۱) مجموعه تمام زبانهای منظم
۲) مجموعه تمام زبانهای حساس به متن که مستقل از متن نمی باشند.
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؟



متن: This is a comment {to be ignored}, in a sample text.

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:

$L_1 = \{ww^R v \mid v , w \in \{a , b\}^*\}$
$L_2 = \{w_1 \subset w_2 \mid w_1 , w_2 \in \{a,b\}^* , w_1 \neq w_2\}$
$L_3 = \{w \mid w \in L_4, w^R \in L_5\}$

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 قید   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.

$V \to \{A_1, A_2, \dots, A_n\}$

۳) مستقل از متن است ولی منظم نیست   ۴) تصمیم‌پذیر است ولی مستقل از متن نیست.

$T \to \{a_1, a_2, \dots, a_n\}$

۹۷ـ نوع زبان $L=\{a^n b^m \mid m \le n^2 , n \le 1000\}$ کدام است؟ زیرا متناهی است

۱) منظم است.   ۲) مستقل از متن است و منظم نیست.

$r = [A_1 \to (A_1 + \dots + A_n + a_1 + \dots + a_n)^*]$

۳) حساس به متن است و مستقل از متن نیست.   ۴) بدون محدودیت است و حساس به متن نیست.

$+ [A_2 \to (A_1 + \dots + A_n + a_1 + \dots + a_n)^*]$

$+ \dots +$

$[A_n \to (A_1 + \dots + A_n + a_1 + \dots + a_n)^*]$

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؟

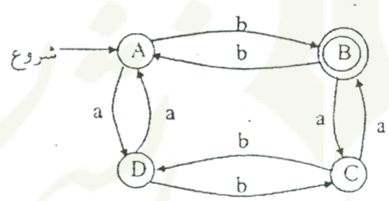
۱) حداکثر تعداد مراحل در اشتقاق‌های چپ برای w برابر $2|w|$ است.

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) $A \to aAa \mid B$
$B \to A \mid bBb \mid a \mid b$

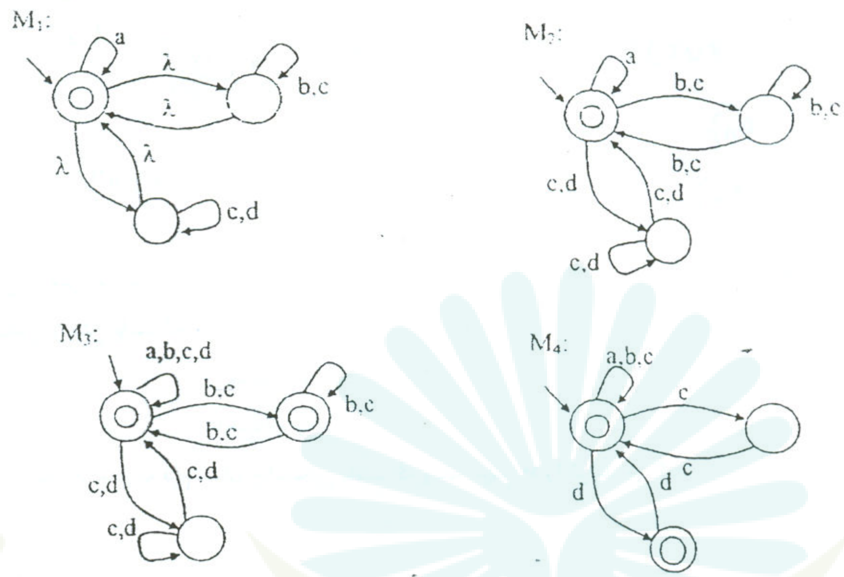
۲) $A \to aD \mid bB \quad B \to aC \mid bA$
$C \to aB \mid bD \quad D \to aA \mid bC$

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 ۵۶ تا ۶۱ منظور از λ رشته‌ای به طول صفر است.

۵۶- 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) $L(M_1)\cap L(M_3)\neq\phi, L(M_4)\subset L(M_2)$

۴) $L(M_2)\subset L(M_4), L(M_1)\subset L(M_3)$

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؟

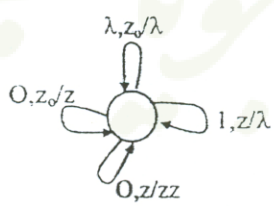
۱) ب تصمیم پذیر است.

۲) الف تصمیم پذیر است.

۳) هر دو مسأله تصمیم‌پذیرند.

۴) 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 ۴ 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 = λ چیزی نمی‌خواند) و علامت X در بالای Stack را با رشته $\gamma\in\Gamma^*$ عوض می‌کند. زبان اتومات داده شده کدام است؟

۱) {تعداد ۱ ها با تعداد صفرها (در ω) برابر است | $\omega\in 0^*1^*$}

۲) {تعداد ۱ های ω دوبرابر تعداد صفرهای ω است | $\omega\in 0^*1^*$}

۳) {تعداد ۱ های ω با تعداد صفرهای ω برابر است | $\omega\in(0+1)^*$}

۴) 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 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 - $\{a^n b^n \mid n \geq 0\} \cup \{a^n b^{2n} \mid n \leq 100\}$

ب - $\{a^n b^{2n} c^n \mid n \geq 0\}$

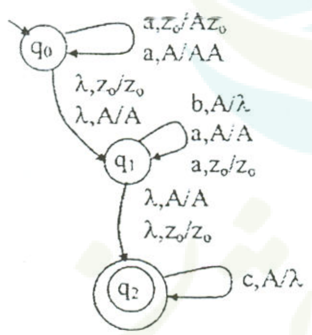
ج - {طول ω زوج و نیمهٔ اول آن فقط شامل a باشد | $\omega \in (a+b)^*$}

د- $\{a^n b^n b^{m+n} c^m \mid n \geq 0, m \geq 0\} \cup \{a^n b^{2n} \mid n \geq 0\}$

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) $L(M) = a^n b^k a^* c^j \quad n \geq k+j \quad k, j \geq 0$

۲) $L(M) = a^n (b^k + a)^* c^j + a^* \quad n \geq k+j \quad k, j \geq 0$

۳) $L(M) = a^n a^* (b+a)^k c^j \quad n \geq k+j \quad k, j \geq 0$

۴) $L(M) = a^n (a^* b a^*)^k c^j \quad n \geq k+j \quad k, j \geq 0$

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 پوش‌دادن معین پذیرنده در حالت نهایی یک اتومات پوش‌دان معین پذیرنده در حالت خالی شدن Stack وجود دارد. آن که خالی می‌شد 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. عکسش درست است

$\{a^n b^n \mid n \geq 0\} \cup \{a^n b^{2n} \mid n \geq 0\}$ مبهم نیست ولی اتومات معین ندارد

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) $O(\log|w|)$

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. ← غلط

$(0 + 11^*01^*0)^* (\varepsilon + 11^*(\varepsilon + 01^*))$

۱) ۵ ۲) ٤ ۳) ۳ ٤) 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) $L = \{a^n : n \geq 100 \text{ یا } n \text{ عدد اول}\}$ ✓ ٤) هیچ‌کدام

$L = \{a^2, a^3, \ldots, \ldots\} \cup \{a^{101}, a^{102}, a^{103}, \ldots\}$
$= \{a^2, a^3, \ldots, a^{97}\} \cup \{a^{101}, a^{102}, a^{103}, \ldots\}$

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_1 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 ٤) الف، ب و د

۵۶ـ کدام گزینه نادرست است؟

۱) مکمل یک زبان بازگشتی، یک زبان بازگشتی است.

۲) مکمل یک زبان بازگشتی برشمردنی، 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.)

$$G: S \to S_1B$$
$$S_1 \to aS_1b$$
$$bB \to bbbB$$
$$aS_1b \to aa$$
$$B \to \lambda$$

۱) $L(G) = \{a^{n+2}b^{3n} \mid n \ge 0\}$

۲) $L(G) = \{a^n b^{n+2k} \mid n \ge 2, k \ge 0\}$

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؟

$L_1 = \{x^n y^n \mid x \in (0+1)^*, y \in (0+1)^*, n \ge 0\}$

$L_2 = \{w \in L(A) \mid$ 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.

خود مستقل از متن هستند

۵۶ـ کدام یک از زبان‌های زیر نامنظم است؟

۱) $\{a^n b^n (a+b)^* \mid n \geq 0\}$ ۲) $\{b^* a^n b^n a^* \mid n \geq 0\}$

۳) $\{a^* a^n b^n b^* \mid n \geq 0\}$ ۴) 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 $|z| \geq 1,\ z \in L$ داشته باشیم:

$z = uvwxy\ ;\ |vx| \neq 0\ ,\ |vwx| \leq n, \forall i \geq 0\ uv^i wx^i y \in L$

۲) عدد ثابت مثل n وجود دارد به طوری که برای هر رشته $|z| \geq n\ ,\ z \in L$ داشته باشیم :

$z = xyw, |y| \neq 0, |xy| \leq n\ ,\ \forall i \geq 0\ xy^i w \in L$

۳) هیچ عدد ثابت مثل n وجود ندارد به طوری که برای هر رشته $|z| \geq n, z \in L$ داشته باشیم :

$z = uvwxy\ ,\ |vx| \neq 0\ ,\ |vwx| \leq n\ ,\ \forall i \geq 0\ uv^i wx^i \in L$

۴) 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 مکمل گیری بسته هستند.

۳) هر زبان Definite با یک ماشین متناهی پذیرفته می‌شود.

۴) زبان‌های Definite تحت عمل *(Kleene star) بسته هستند.

۵۹ـ مجموعه‌های زیر را در نظر بگیرید :

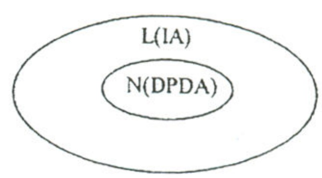
L(PDA) : مجموعه زبان‌هایی که برای آن‌ها PDA (Pushdown Automata) وجود دارد.

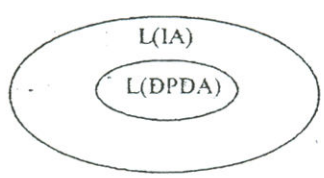
L(DPDA) : مجموعه زبان‌هایی که برای آن‌ها DPDA (Deterministic PDA) وجود دارد.

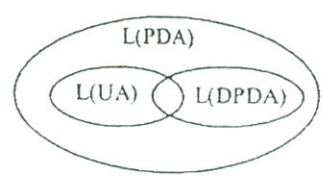
L(UA) : مجموعه زبان‌های مستقل از متن غیر مبهم (unambiguous context free)

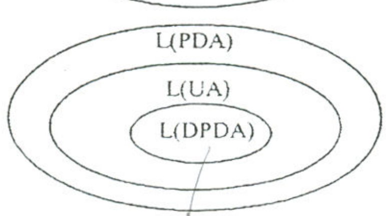
L(IA) : مجموعه زبان‌های مستقل از متن ذاتاً مبهم (Inherently Ambiguous)

کدام یک از نموددارهای مجموعه‌ای زیر درست است؟

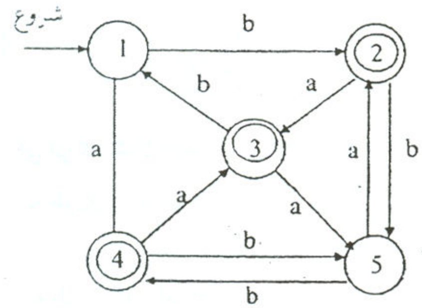
۱)


۲)


۳)


۴)


۶۰ـ اتومات متناهی زیر را درنظر می‌گیریم. اتومات کمینه (minimized) مربوطه دارای چند حالت خواهد بود؟



۱) ۳
۲) ۲
۳) ۵
۴) ۴

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) $\{a^n b^n \mid n \geq 0\}$
۲) $\{a^n b^n a^n \mid n \geq 1\}$
۳) $\{w \in (a+b)^+ \mid$ تعداد a ها با تعداد b ها برابر است $\}$
۴) هیچکدام ✓

$\{a^n b^n \mid n \geq 1\}$

هیچ a^0b^0 $n \geq 0$ درست نیست. تساوی صفر نوع رانی نپذیرد که نباید

۵۶ـ $L = \{a^m cb^n : m \neq n\} \cup \{a^m db^{2m} : m \geq 0\}$ ، کدام گزینه نادرست است؟

۱) هر هومورفیسم 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 $\Sigma = \{a, b, c\}$ و $L - \Sigma^* = \{a, b, c\}$ باشد آن‌گاه L کدام یک از زبان‌های زیر می‌تواند باشد؟

$\varepsilon - IV$, $\varphi - III$, $a^n b^{n^2} c^n - II$, $\Sigma^* - I$

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 → xS v | 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 پذیرند ← هستند 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. نزول دارد ← حساس به متن ← LBA پذیرند ✓

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؟

$$G : S \to aAb \mid bBa \mid bCa$$
$$A \to aaAb \mid ab$$
$$B \to bBa \mid a$$
$$C \to aC \mid bC$$

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) $a^{k+1} b^k \cup b^l a^l \quad l \geq 1, k \geq 2$　　۴) $a^2 a^{2k} b^k b^2 \cup b^l a^{l+1} \quad k \geq 0, l \geq 1$

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) $\{aaaa, aaaaaa\}$

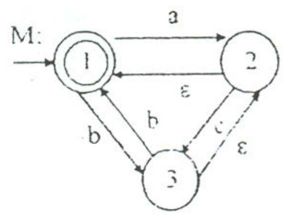
۶۰ـ گرامر G به شرح زیر مفروض است. $L(G)$ کدام است؟ w^R عبارت است از w که از آخر به اول خوانده شود. و ε نشانهٔ رشته های به طول صفر است.

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) $\{w \in (a+b)^* \mid w = w^R\}$

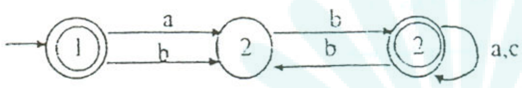
۳) $\{w(a+b)w^R \mid w \in (a+b)^*\}$　۴) $\{ww^R \mid w \in (a+b)^*\}$

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؟

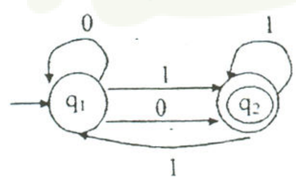
$L_1=\left\{\alpha^i(\alpha\beta)^j(\gamma_\alpha)^i \quad j\ge 0,\ i\ge 0\right\}$

$L_2=\left\{\alpha^i(\alpha\beta)^j(\gamma_\alpha)^{i+j} \quad j\ge 0,\ i\ge 1\right\}$

$L_3=\left\{\alpha^i(\alpha\beta)^j(\gamma_\alpha)^i \quad j\ge 1,\ i\ge 1\right\}$

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؟

$d(M_2)$ هم می‌توان نوشت

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

۶۲- زبان L مجموعه تمامی زوج‌های مرتب <M , W> 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