مبانی نظریه محاسبه

مرجع

۱. کتاب مقدمه‌ای بر نظریه زبان‌ها و ماشین‌ها نویسنده پیتر لینز

مترجم دکتر مهدی صادق زاده

۲. کتاب مقدمه‌ای بر نظریه محاسبات مایکل سیپسر

مترجم: جواد وحیدی

حسین رسولی

مقدمات ریاضی

• مجموعه: دسته‌ای از عناصر بدون هیچ ساختاری، صرفاً عضو مجموعه هستند

$A=\{1,2,3\}$

$B=\{\text{book}, \text{bus}, \text{teacher}, \text{student}\}$

می‌نویسیم:

$\text{ship} \notin B \quad , \quad 1 \in A$

• نمایش مجموعه:

۱- نمایش با اعضا:

$A=\{a,b,c,d,e,f\}$

$A=\{a,b,\dots,f\}$ $\longrightarrow$ مجموعه متناهی

$B=\{1,2,3,\dots\}$ $\longrightarrow$ مجموعه نامتناهی

2- نمایش با نماد ریاضی:

$B=\{i \mid i>0\}$

مجموعه جهانی (U): همه عناصر ممکن

مجموعه تهی (ϕ)

$\{\}$

• عملگرهای مجموعه: اجتماع، اشتراک، تفاضل، متمم

$A=\{1,2,3\} \quad , B=\{3,4,5,6\}$

اجتماع $A\cup B=\{1,2,3,4,5,6\}$

اشتراک $A\cap B=\{3\}$

تفاضل $A-B=\{1,2\} \qquad B-A=\{4,5,6\}$

متمم if $U=\{1,2,3,4,5,6,7,8\}$

$\overline{A}=\{4,5,6,7,8\}$

• قوانین دمورگان :

$$\overline{A\cup B} = \bar{A} \cap \bar{B}$$

$$\overline{A\cap B} = \bar{A} \cup \bar{B}$$

• مفاهیم اساسی مجموعه‌ها :

زیرمجموعه بودن :

$A=\{1,2,3\} \quad B=\{1,2,3,4,5\} \quad A\subseteq B$

مجموعه‌های مجزا :

$A=\{1,2,3\} \quad B=\{4,5\} \quad A\cap B=\phi$

کاردینالیتی مجموعه : برای مجموعه‌های متناهی

$A=\{1,2,3\} \quad |A|=3$

مجموعه توانی : مجموعه توانی A یعنی مجموعه تمام زیرمجموعه‌های A

$A=\{1,2,3\}$

$$2^A=\{\phi,\{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\},\{1,2,3\}\}$$

$$|2^A| = 2^{|A|} = 2^3 = 8$$

• حاصلضرب دکارتی :

$$A_1\times A_2=\{(x,y)\mid x\in A_1 , y\in A_2\}$$

$\vdots$

$$A_1\times A_2\times\cdots\times A_n=\{(x_1,x_2,\ldots,x_n)\mid x_i\in A_i \quad i=1,\ldots,n\}$$

$A=\{1,2\} \quad , B=\{2,3,4\}$

$$A\times B=\{(1,2),(1,3),(1,4),(2,2),(2,3),(2,4)\}$$

$$|A\times B| = |A|\,|B|$$

• توابع :

$\underset{\text{دامنه}}{A} \xrightarrow{f} \underset{\text{برد}}{B}$ ، f را تابع گوییم هرگاه :

$\forall x_1, x_2 \in A \quad x_1 = x_2 \longrightarrow f(x_1) = f(x_2)$

if $D_f = A \longrightarrow$ f تابع کلی یا تام   o.w تابع جزئی

• رابطه : رابطه کلی‌تر از تابع هست

$R = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots\}$

$x_i R y_i$ می‌نویسیم

به عنوان مثال if $R = '>'$; $2>1$, $3>2$, $3>1$

در رابطه x می‌تواند تکرار شود

رابطه هم‌ارزی :

خاصیت انعکاسی xRx

خاصیت تقارن $xRy \to yRx$

خاصیت تعدی xRy and $yRz \to xRz$

کلاس هم‌ارزی :

برای رابطه هم‌ارزی R ، کلاس هم‌ارزی X به صورت زیر است

$X = \{Y \mid XRY\}$

مثال

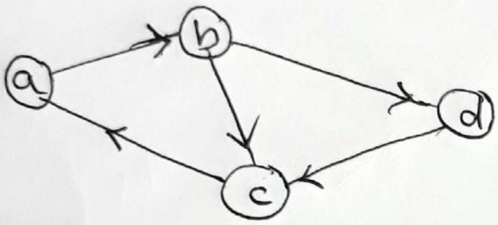
$R = \{(1,1),(2,2),(1,2),(2,1),(3,3),(4,4),(3,4),(4,3)\}$

کلاس هم‌ارزی 1 $= \{1,2\}$

کلاس هم‌ارزی 3 $= \{3,4\}$

گراف‌ها و درخت‌ها :

گراف ساده - گراف جهت‌دار - گراف برچسب‌دار



رأس‌ها : $V = \{a,b,c,d\}$

یال‌ها : $E = \{(a,b),(b,c),(c,a),(b,d),(d,c)\}$

گشت = دنباله‌ای از یال‌ها که متصل به هم

طول یک گشت = تعداد یال‌های گشت از ابتدا (مبدا) تا مقصد

مسیر = راهی که یال تکراری ندارد

مسیر ساده = مسیری که راس تکراری ندارد

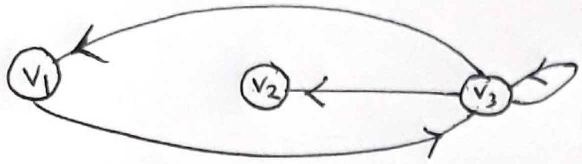
دور (چرخه) = یک گشت از راس v به خودش بدون یال تکراری

دور ساده = دوری که هیچ راس تکراری ندارد (بجز راس اول)

درخت جهت‌دار = گراف جهت‌داری که دور ندارد و دارای یک گره خاص به نام ریشه دارد.

Scanned by CamScanner

مثال : گراف زیر را در نظر می گیریم



یک گشت :

$(v_1,v_3),(v_3,v_3),(v_3,v_1),(v_1,v_3),(v_3,v_2)$

یک مسیر : $(v_1,v_3)(v_3,v_2)$

یک دور : $(v_1,v_3),(v_3,v_3),(v_3,v_1)$

• روش های اثبات

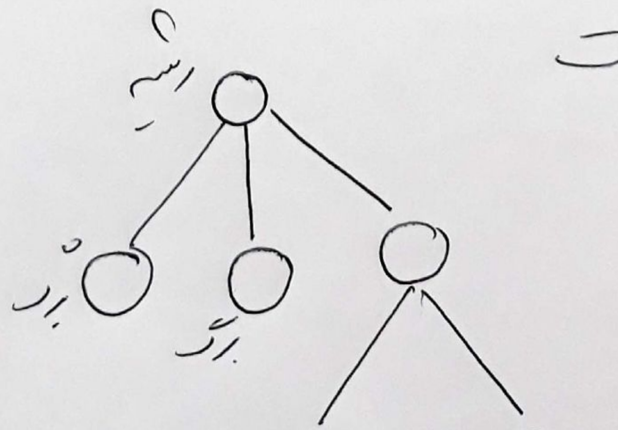
حکم استقرا   فرض استقرا   ابتدای استقرا

۱- استقراء : $P(1)$ درست ، فرض $P(n)$ ، حکم $P(n+1)$

۲- برهان خلف : عکس آنچه که می خواهیم اثبات کنیم

ثابت کنید که $\sqrt{2}$ عدد گویا نیست.

درخت



مفاهیم اساسی در نظریه محاسبات:

زبانها - گرامرها - اتوماتا (ماشین ها)

نماد: کوچک ترین و بنیادی ترین عضو یک زبان است. در برخی مواقع به نمادها حرف هم گفته می شود. نمادها را معمولا با حروف لاتین کوچک مثل a,b و ... نشان می دهند

الفبا: یک مجموعه متناهی از نمادهاست که معمولا با Σ نشان می دهیم

رشته: دنباله ای از نمادهای یک مجموعه الفباست که با عمل الحاق به هم پیوسته اند

رشته ممکن است متناهی یا نامتناهی باشد

مثال: اگر الفبا $\Sigma=\{a,b\}$ باشد آنگاه abab ، aab ، aaabbba رشته هایی روی Σ هستند

ما برای نامگذاری رشته ها از حروف $u,v,w,\dots$ استفاده می کنیم

در مثال بالا می توانیم بنویسیم

$u = aab$

$v = abab$

$w = aaabbba$

که نشان می دهد رشته ای با نام u دارای مقدار خاص aab می باشد.

• الحاق دو رشته: اگر u و v دو رشته باشند، حاصل الحاق u , v رشته ای است که از قرار دادن v به دنبال u به صورت uv بدست می آید.

مثال. فرض کنید $u = abba$, $v = bbbaaa$ آنگاه

$$uv = \underbrace{abba}_{u}\underbrace{bbbaaa}_{v}$$

- معکوس یک رشته بوسیله نوشتن نمادها به ترتیب معکوس حاصل می‌شود

فرض کنید $u = a_1a_2 \cdots a_n$ آنگاه $u^R = a_n \cdots a_2a_1$

- طول رشته u، تعداد نمادها در رشته می‌باشد و با $|u|$ نشان می‌دهیم.

فرض کنید $u = a_1a_2 \cdots a_n$ آنگاه $|u| = n$

$v = aabbb$ آنگاه. $|v| = 5$

- رشته تهی، رشته ای بدون هیچ نماد است آن را با λ نمایش می‌دهیم

$$|\lambda| = 0$$

رشته تهی (λ) عضو خنثای عمل الحاق است

$$\lambda u = u\lambda = u \quad \forall u$$

- زیر رشته: بخشی از حروف متوالی رشته u را زیر رشته u گویند اگر

$$u = vw$$

آنگاه زیر رشته های v و w را به ترتیب پیشوند و پسوند u گویند

مثال اگر $u = abbab$ آنگاه $\{\lambda, a, ab, abb, abba, abbab\}$

~~همچنین~~ b، مجموعه همه پیشوندهای u است. b، ab، bab برخی از پسوندهای آن است

۱

اگر u یک رشته باشد آنگاه u^n نشان دهنده رشته‌ای است که به وسیله عمل الحاق u به تکرار n بار حاصل می‌شود

تعریف:

$$\frac{u^0 = \lambda \quad \forall u}{\text{یک مورد خاص}}$$

$$u^n = \underbrace{uu\ldots u}_{n\ \text{بار}}$$

مثال

$$(abba)^2 = abbaabba$$

$$(abba)^0 = \lambda$$

* الفبای Σ را در نظر بگیرید، مجموعه تمام رشته‌های ممکن از الفبای Σ را با Σ^* نشان می‌دهیم

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

$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$

همچنین

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

اگرچه Σ متناهی است، اما Σ^* و Σ^+ همیشه نامتناهی هستند

• زبان

یک زبان مجموعه‌ای از رشته‌های یک الفبا است یعنی $L \subseteq \Sigma^*$

اگر مجموعه L متناهی باشد، L یک زبان متناهی است در غیر این صورت نامتناهی است

مثال : L_1 , L_2 هر دو زبان‌هایی بر روی الفبای $\Sigma=\{a,b\}$ هستند

$L_1=\{a, aa, aab\}$ زبان متناهی

$L_2=\{a^nb^n : n\geq 0\}$ زبان نامتناهی

عملیات روی زبان‌ها :

فرض می‌کنیم $L, L_1, L_2 \in \Sigma^*$

- اجتماع دو زبان : $L_1 \cup L_2=\{x \mid x\in L_1 \vee x\in L_2\}$
- اشتراک دو زبان $L_1 \cap L_2=\{x \mid x\in L_1 \wedge x\in L_2\}$
- تفاضل دو زبان $L_1 - L_2=\{x \mid x\in L_1 \wedge x\notin L_2\}$
- متمم یک زبان $\overline{L}=\Sigma^* - L$
- معکوس یک زبان $L^R=\{x^R \mid x\in L\}$
- الحاق دو زبان $L_1L_2=\{x_1x_2 \mid x\in L_1 \wedge x_2\in L_2\}$
- توان یک زبان

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

$L^{n+1}=L^nL=LL^n$

• بستار

$$L^* = L^0 \cup L^1 \cup L^2 \cup \dots = \bigcup_{i=0}^{\infty} L^i$$

$$L^+ = L^1 \cup L^2 \cup \dots = \bigcup_{i=1}^{\infty} L^i$$

مثال . اگر $L = \{a^n b^n \mid n \geq 0\}$ آنگاه .

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

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

مثال

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

$\Sigma^* = \{\lambda, a, b, aa, ab, abb, aabb, baa, bab, \dots\}$

و مجموعه $\{a, ab, bab\}$ یک زبان روی الفبای Σ است (زبان متناهی)

یک زبان روی Σ، یک زیرمجموعه ای از Σ^* است

$L_1 = \{a, aa, aab\}$

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

$L_1 \cup L_2 = \{a, aa, aab, ab, a^2b^2\}$

$L_1 \cap L_2 = \cancel{\phi}\ \{a\}$

$L_1 - L_2 = \{aa, aab\}$

$L_1^R = \{a, aa, baa\}$

$L_2^R = \{ba, b^2a^2, a\}$

$L_1L_2 = \{\underline{a}ab, \underline{a}a^2b^2, \underline{a}a, \underline{aa}ab, \underline{aa}a^2b^2, \underline{aa}a,$

$\underline{aab}ab, \underline{aab}a^2b^2, \underline{aab}a\}$

$L_2L_1 = \{\underline{ab}a, \underline{ab}aa, \underline{ab}aab, \underline{a^2b^2}a, \underline{a^2b^2}aa, \underline{a^2b^2}aab$

$\underline{a}a, \underline{a}aa, \underline{a}aab\}$

$L_1L_1 = \{\underline{a}a, \underline{a}aa, \underline{a}aab, \underline{aa}a, \underline{aa}aa, \underline{aa}aab,$

$\underline{aab}a, \underline{aab}aa, \underline{aab}aab\} =$

$\{a^2, a^3, a^4, a^3b, \cancel{a^3}a^4b, a^2ba, a^2ba^2, a^2ba^2b\}$

lk

مثال

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

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

$L_1 \cup L_2 = \{a^n b^n \mid n \geq 0\} \cup \{ab^2\}$

$L_1 \cap L_2 = \{a^2b^2\}$

$L_1 - L_2 = \{a^n b^n \mid n \geq 0 \,\&\, n \neq 2\}$

$L_2 - L_1 = \{ab^2\}$

$L_1^R = \{b^n a^n \mid n \geq 0\}$

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

$L_2L_2 = \{\underline{ab^2}ab^2, \underline{ab^2}a^2b^2, \underline{a^2b^2}ab^2, \underline{a^2b^2}a^2b^2\}$

$L_1L_2 = \{\underbrace{\lambda}_{\in L_1}\overbrace{ab^2}^{\in L_2}, \underbrace{\lambda}_{\in L_1}\overbrace{a^2b^2}^{\in L_2}, \underbrace{ab}_{\in L_1}\overbrace{ab^2}^{\in L_2}, \underbrace{ab}_{\in L_1}\overbrace{a^2b^2}^{\in L_2}, \underbrace{a^2b^2}_{\in L_1}\overbrace{ab^2}^{\in L_2}, \underbrace{a^2b^2}_{\in L_1}\overbrace{a^2b^2}^{\in L_2}, \dots\}$

$= \{a^n b^n ab^2, a^n b^n a^2 b \mid n \geq 0\}$

$L_1L_1 = \{\lambda, \lambda ab, \lambda a^2b^2, \lambda a^3b^3, \dots, ab\lambda, abab, aba^2b^2,$

$aba^3b^3, \dots, a^2b^2\lambda, a^2b^2ab, a^2b^2a^2b^2, a^2b^2a^3b^3,$

$\dots\} = \{a^n b^n a^m b^m \mid n \geq 0, m \geq 0\}.$

n, m هیچ ارتباطی به هم ندارند.

$\underbrace{aabb}_{\in L_1}\underbrace{aaabbb}_{\in L_2}$ در L_1^2 است

گرامرها:

یک گرامر G، چهارتایی به صورت $G = (N, T, S, P)$ است که در آن

N: یک مجموعه متناهی است که نماد غیرانتهایی (nonterminal symbols) نامیده می‌شوند

T: یک مجموعه متناهی است که نماد انتهایی نامیده می‌شود

$S \in N$: نماد آغازین یا نماد شروع کننده می‌باشد.

$P \subseteq ((N \cup T)^* \setminus T^*) \times (N \cup T)^*$: مجموعه محصولات می‌باشد (productions)

توجه شود که $N \cap T = \phi$

قرارداد: اگر زوج $(x, z) \in P$، آنگاه آنرا به صورت $z \to x$ نشان می‌دهیم و این یک محصول است.

مثال: فرض کنید $N = \{s, S\}$ (نماد آغازین)، $T = \{a, b\}$،

$$P = \{S \to bS,\ S \to a s,\ s \to b s,\ s \to b\}$$

آنگاه $G = (N, T, S, P)$ یک گرامر است.

تعریف زبان یک گرامر:

فرض کنید $G = (N, T, S, P)$ یک گرامر باشد زبان G که با $L(G)$ نشان می‌دهیم عبارتست از مجموعه تمام کلمات در T^*، که با بکارگیری یک سری از محصولات بتوان به آن کلمه رسید. مشروط بر آنکه در اولین -

مشروط بر اینکه در این تمام از محصولی به صورت $S \to x$ استفاده شود که S همان نماد آغازین است.

توضیح اینکه اگر محصول $z \to w$ را داشته باشیم می‌توانیم به

$$xzy \Rightarrow xwy \quad (*)$$

برسیم که به $(*)$ یک اشتقاق گوییم (derivation).

فرض کنید که $z_{i-1} \Rightarrow z_i$ یک اشتقاق باشد برای $i=1,2,\dots,n$. یعنی داشته باشیم

$$z_0 \Rightarrow z_1 \Rightarrow z_2 \Rightarrow \cdots \Rightarrow z_n$$

در این حالت گوییم z_n اشتقاق از z_0 است و می‌نویسیم

$$z_0 \overset{*}{\Rightarrow} z_n$$

حال می‌توانیم تعریف زبان گرامر را حسب بیان کنیم:

گوییم $w \in L(G)$ اگر و فقط اگر $w \in T^*$ و w از S قابل اشتقاق باشد (مشتق شود)

زبان تولید شده توسط G:

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

مثال : فرض کنید $N = \{S\}$ ، $T = \{a, b\}$ ،

$P = \{S \rightarrow aSb \,,\, S \rightarrow \lambda\}$

آنگاه:

$S \Rightarrow \lambda$

$S \Rightarrow aSb \Rightarrow ab$

$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow a^2b^2$

$S \Rightarrow aSb \Rightarrow a^2Sb^2 \Rightarrow a^3Sb^3 \Rightarrow a^3b^3$

$\vdots$

$S \Rightarrow aSb \Rightarrow a^2Sb \Rightarrow a^3Sb \Rightarrow \cdots \Rightarrow a^nSb^n \Rightarrow a^nb^n$

اگرچه همیشه ممکن نیست که از روی دستور زبان را تشخیص دهیم اما در این مثال روشن است که

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

مثال : فرض کنید $N = \{S, S'\}$ ، $T = \{a, b\}$

$P = \{S' \xrightarrow[1]{} bS' \,,\, S' \xrightarrow[2]{} aS \,,\, S \xrightarrow[3]{} bS \,,\, S \xrightarrow[4]{} b\}$

$S' \overset{2}{\Rightarrow} aS \overset{3}{\Rightarrow} abS \overset{4}{\Rightarrow} abb$

$S' \underset{1}{\Rightarrow} bS' \underset{2}{\Rightarrow} baS \underset{4}{\Rightarrow} bab$

$S' \underset{2}{\Rightarrow} aS \underset{4}{\Rightarrow} ab$

$S' \underset{1}{\Rightarrow} bS' \underset{1}{\Rightarrow} bbS' \Rightarrow \cdots \underset{1}{\Rightarrow} b^nS' \underset{2}{\Rightarrow} b^naS \underset{3}{\Rightarrow} b^nabS$

$\underset{③}{\Longrightarrow} \cdots \underset{③}{\Longrightarrow} b^n a b^m S \underset{④}{\Longrightarrow} b^n a b^m b$

از مطالب فوق در می یابیم که:

$L(G) = \{ b^n a b^m \mid n \geq 0 , m \geq 1 \}$

مثال: دستور زبانی بسازید که $L = \{ a^n b^{n+1} \mid n \geq 0 \}$ را تولید کند.

$\{ b, ab^2, a^2b^3, \dots \}$

$S \xrightarrow{①} Ab \quad , \quad A \xrightarrow{②} \lambda$

$A \xrightarrow{③} aAb$

پس گرامر G را به صورت زیر داریم

$G = (\overset{N}{\{S, A\}}, \overset{T}{\{a, b\}}, S, P)$

$S \overset{①}{\Rightarrow} Ab \overset{②}{\Rightarrow} b$ ✓

$S \underset{①}{\Rightarrow} Ab \underset{③}{\Rightarrow} aAbb \underset{②}{\Rightarrow} ab^2$ ✓

$S \underset{①}{\Rightarrow} Ab \underset{③}{\Rightarrow} aAbb \underset{③}{\Rightarrow} aaAbbb \underset{②}{\Rightarrow} a^2b^3$ ✓

$\vdots$

$S \underset{①}{\Rightarrow} Ab \underset{③}{\Rightarrow} aAbb \underset{③}{\Rightarrow} \cdots \underset{③}{\Rightarrow} a^n A b^n b \Rightarrow a^n b^{n+1}$

فرض کنید $L = \{ab, aa, baa\}$

1) $baaaaabaa \in L^*$

2) $aaaabaaaaa \in L^*$

3) $abaabaaabaa \in L^*$

4) $baaaaabaaaab \notin L^*$

زبان شامل گرامر زیر را در نظر بگیرید. آیا $L = L^*$

$S \to Ab$ ①

$A \to aAb$ ②

$A \to \lambda$ ③

$S \xRightarrow{①} Ab \xRightarrow{③} b$ ✓

$S \xRightarrow{①} Ab \xRightarrow{②} aAbb \Rightarrow \cdots \xRightarrow{②} a a^n A b^n bb$

$a^n b^{n+1}$ ✓ $n \geq 0$

چون $\lambda \notin L$ پس $L \neq L^*$

آیا زبانی وجود دارد به طوری که $\overline{L^*} = \overline{L}^*$

چون برای هر زبان L داریم $\lambda \in L^*$ پس $\lambda \notin \overline{L^*}$

$\lambda \in \overline{L}^*$ ولی $\lambda \notin \overline{L^*}$

زبانی که توسط گرامر زیر تولید می‌شود

$S \to aA$, 1

$A \to bS$ 2

$S \to \lambda$ 3

$S \underset{3}{\Rightarrow} \lambda$ ✓

$S \underset{1}{\Rightarrow} aA \underset{2}{\Rightarrow} abS \underset{3}{\Rightarrow} ab$ ✓

$S \underset{1}{\Rightarrow} aA \underset{2}{\Rightarrow} abS \underset{1}{\Rightarrow} abaA \underset{2}{\Rightarrow} ababS \underset{3}{\Rightarrow} (ab)^2$

$\vdots$

$S \underset{1}{\Rightarrow} aA \underset{2}{\Rightarrow} abS \underset{1}{\Rightarrow} abaA \underset{2}{\Rightarrow} ababS \underset{1}{\Rightarrow} ababaA$

$\Rightarrow abababS \Rightarrow \dots \Rightarrow (ab)^n S \Rightarrow (ab)^n$ ✓

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

برای هر یک از زبان‌های زیر گرامری که آن را تولید کند، ارائه بدهید.

$L_1 = \{ a^n b^m \mid n \geq 0 , m > n \}$

$S \to aSb$

$S \to B$

$B \to bB$

$B \to b$

11

$L_2 = \{ a^n b^{2n} \mid n \geq 0 \}$

$S \to aSbb$

$S \to \lambda$

$a^3 b^1$

$L_3 = \{ a^{n+2} b^n \mid n \geq 1 \}$

$S \to aSb$

$S \to aaab$

اشتق

$S \Rightarrow aaab$ ✓ $n=1$

$S \Rightarrow aSb \Rightarrow \cdots \Rightarrow a^n S b^n \Rightarrow a^{n+3} b^{n+1}$ ✓

$L_4 = \{ a^n b^{n-3} \mid n \geq 3 \}$

if $n=3 \rightsquigarrow a^3 \in L_4$

if $n=4$ $a^4 b \in L_4$

$S \to aaa$

$S \to aSb$

$S \Rightarrow a^3$ ✓

$S \Rightarrow aSb \Rightarrow \cdots \Rightarrow a^n S b^n \Rightarrow \underbrace{a^n aaa b^n}$

$a^{n+3} b^n$ ✓

$\equiv a^n b^{n-3}$ ✓

$L = \{ a^{2n} b^{n+3} \mid n \geq 0 \}$

$S \rightarrow aaSb$

$S \rightarrow bbb$

$L = \{ b^n a b^m \mid n \geq 0 , m \geq 1 \}$

$S \rightarrow AaB$

$A \rightarrow bA$

$A \rightarrow \lambda$

$B \rightarrow bB$

$B \rightarrow b$

$L = \{ a^{2n} b^{3n} \mid n > 0 \}$

$S \rightarrow aaSbbb$

$S \rightarrow aabbb$

$L = \{ a^n b^{2n} \mid n \geq 1 \}$

$S \rightarrow aSbb$

$S \rightarrow abb$

$L = \{ a^n b^{2n} \quad n \geq 0 \}$

$S \rightarrow aSbb$

$S \rightarrow \lambda$

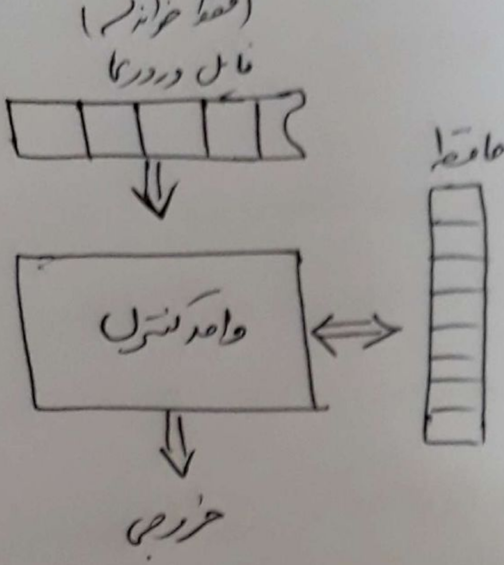
١٠

Scanned by CamScanner

اتوماتا

اتوماتا: مدل انتزاعی از کامپیوتر که دارای مکانیزم زیر است:

۱) ورودی: یک رشته الفبایی داده شده است که روی یک فایل ورودی نوشته شده است. فایل ورودی به سلول تقسیم شده و هر سلول یک نماد از الفبا است. فایل ورودی از چپ به راست خوانده می‌شود. مکانیزم خواندن انتهای ورودی را می‌تواند تشخیص دهد.



* اتوماتا می‌تواند فایل ورودی را بخواند اما نمی‌تواند آن را تغییر دهد

۲) حافظه موقت: تعداد نامتناهی سلول است و در داخل هر سلول یک نماد از الفبا می‌تواند قرار بگیرد. (قابل خواندن و نوشتن)

۳) واحد کنترل: اتوماتا هر زمان می‌تواند در یکی از حالت‌های داخلی خود باشد. تعداد حالت‌ها متناهی است. اتوماتا در هر زمان در یک حالت خاص است و حرف خاصی از ورودی را می‌خواند. حالت بعدی توسط تابع تبدیل حالت تعیین می‌گردد و باعث خروجی و یا تغییر در حافظه موقت می‌شود.

اتوماتای متناهی معین و نامعین (اتوماتای متناهی قطعی و غیرقطعی)
(Deterministic & Nondeterministic finite automata

DFA NFA

در اتوماتای معین اگر حالت فعلی، ورودی و محتوای حافظه را بدانیم، می‌توان رفتار بعدی اتوماتا را تعیین کرد ولی در اتوماتای نامعین اینگونه نیست.

اتوماتای نامعین در هر زمان می‌تواند چند حرکت مختلف انجام دهد.

اتوماتا

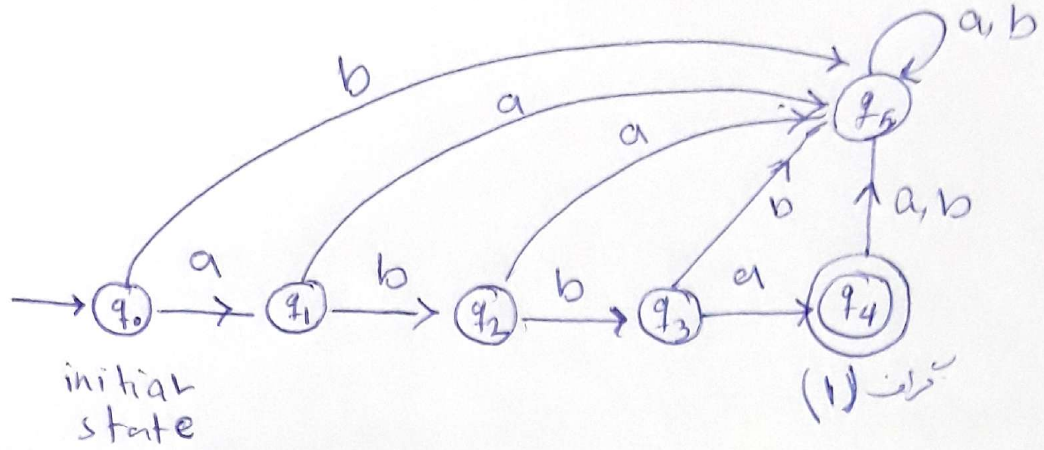
پذیرنده (Acceptor): اتوماتایی که خروجی آن قبول یا عدم قبول است (ورودی خود را قبول یا رد می‌کند)

تراگذار (Transducer): اتوماتایی که خروجی آن بصورت رشته است

کاربرد: برای زبان‌های برنامه‌نویسی، کامپایلر نوشته می‌شود. کامپایلر نیاز به توصیف دقیق و رسمی آن زبان برنامه‌نویسی دارد. لذا می‌توان در زبان‌های برنامه‌نویسی، از گرامر یا اتوماتا برای پذیرش یا عدم پذیرش (قبول یا رد) یک قطعه کد توسط آن زبان برنامه‌نویسی استفاده نمود.

مثال.

abba - Finite Accepter



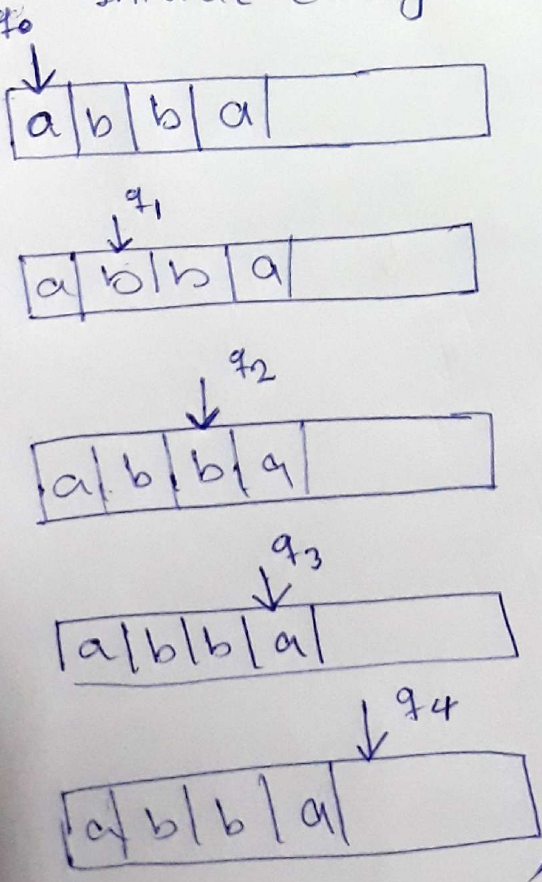
گراف (۱)

states: q_i ، $i = 0, 1, 2, 3, 4, 5$

transition $\longrightarrow$

final state : q_4

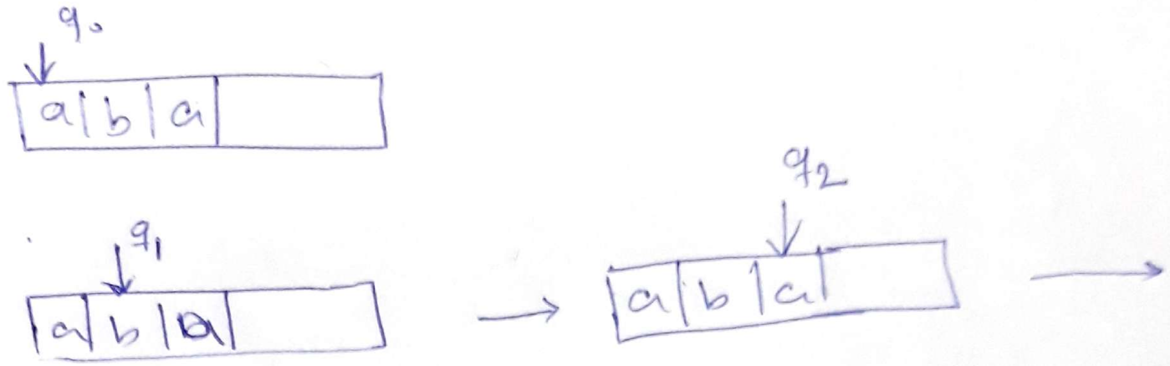
Initial configuration : $(q_0, abba)$

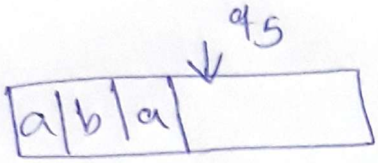


Input string

$\Longrightarrow$ output : "accept"

چون q_4 حالت پایانی است پس
رشته abba توسط این گراف پذیرفته می‌شود.





output : "reject"

یعنی این رشته aba توسط گراف (۸) پذیرفته نمی‌شود. چون q_5 حالت نهایی نیست

مثال (پذیرنده)

گرامر و اتوماتای مربوط به مجموعه متغیرهای زبان پاسکال را بررسی نمایید :

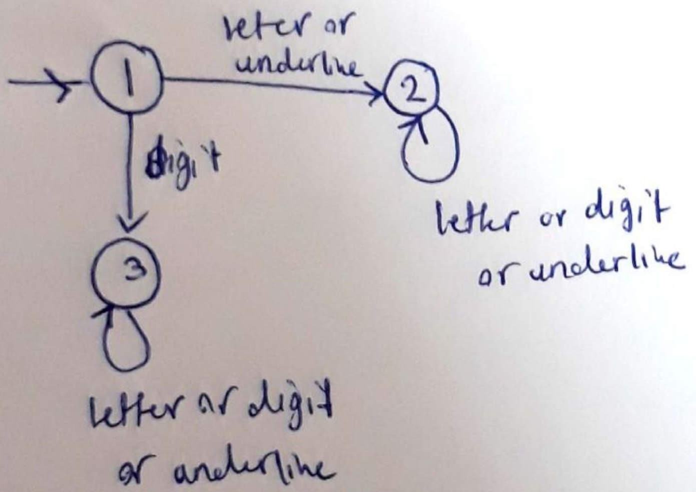
S → <letter><A> | <underline><A>

<A> → <letter> <A> | <Digit><A> | <underline> <A> | λ

<letter> → A|B|...|Z|a|b—|z

<Digit> → ۰ | ۱ | -- | ۹

<underline> → _



states :

1) start

2) Acceptor

3) Not Acceptor

فصل دوم

اتوماتای متناهی (Finite Automata) (FA)

اتوماتای متناهی دارای حافظه موقت نیست و ساده‌ترین نوع اتوماتا است.

تعریف اتوماتای متناهی معین - DFA (Deterministic Finite Automata)

یک اتوماتای حالت متناهی معین (قطعی) پنج تایی $A=(Q,\Sigma,\delta,q_0,F)$ است که در آن

Q: مجموعه متناهی از حالت‌ها است

Σ: مجموعه متناهی از ورودی‌ها است (الفبای ورودی)

$\delta: Q\times\Sigma \to Q$: تابع تغییر حالت

q_0: حالت ابتدایی است

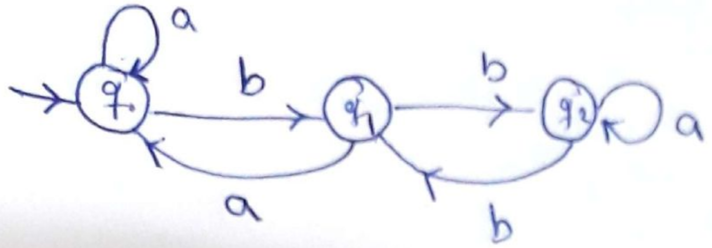
F: مجموعه حالت‌های نهایی است.

مثال: $A=(\{q_0,q_1,q_2\},\{a,b\},\delta,q_0,\{q_1\})$

$$\delta: \begin{array}{ll} \delta(q_0,a)=q_0 , & \delta(q_0,b)=q_1 \\ \delta(q_1,a)=q_0 , & \delta(q_1,b)=q_2 \\ \delta(q_2,a)=q_2 & \delta(q_2,b)=q_1 \end{array}$$

	a	b
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_2	q_1

نمایش DFA به صورت یک گراف تغییر حالت است که راس‌های آن حالت‌ها و یال‌های آن نمادهای ورودی هستند و دارای $|Q|$ راس است.



اتوماتا در ابتدا در حالت اولیه (ابتدایی) q_0 است. مکانیزم ورودی آن روی اولین حرف ورودی (سمت چپ) است. با هر حرکت اتوماتا، مکانیزم ورودی یک حرف به جلو می‌رود (به راست) وقتی که به انتهای رشته برسیم اگر اتوماتا در حالت نهایی باشد می‌گوییم که رشته پذیرفته شده است.

تعریف: فرض کنید $A=(Q,\Sigma,\delta,q_0,F)$ یک اتوماتای حالت متناهی باشد. فرض کنید $x=x_1x_2\cdots x_n\in\Sigma^*$. اگر وجود داشته باشد حالت‌های $q_0,q_1,\ldots,q_n$ به‌طوری که

1) $\delta(q_{i-1},x_i)=q_i \quad i=1,\ldots,n$

2) $q_n\in F$

آنگاه می‌گوییم x یک کلمه پذیرش در اوتوماتا است.

مثال قبل را در نظر بگیرید:

توجه: به نمایش حالت اتوماتا در هر لحظه به صورت (ادامه ورودی، حالت فعلی) یک پیکربندی گوییم. حرکت بین پیکربندی‌ها با نماد ⊢ توصیف می‌شود.

$$(q_0,abbb)\vdash(q_0,bbb)\vdash(q_1,bb)\vdash(q_2,b)\vdash(q_1,\lambda)$$

چون با نمایش پیکربندی هر لحظه به صورت (ادامه ورودی و حالت فعلی)، اتوماتای فوق ورودی‌اش را که رشته $w=abbb$ است، به‌طور کامل مصرف می‌کند. چون رشته ورودی تمام شد در حالت q_1 یک حالت نهایی برای این اتوماتا است، رشته w توسط این اتوماتا پذیرفته می‌شود.

اما اگر رشته $w = baa$ را به عنوان ورودی به اتوماتون می‌دهیم :

$$(q_0, baa) \vdash (q_1, aa) \vdash (q_0, a) \vdash (q_0, \lambda)$$

رشته ورودی تمام شد اما اتوماتون در یک حالت غیرنهایی توقف کرد، پس w توسط این اتوماتون پذیرفته نمی‌شود.

تعریف تابع تغییر حالت توسعه یافته

$$\delta^* : Q \times \underbrace{\Sigma^*}_{\text{رشته}} \to Q$$

تعریف بازگشتی :

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

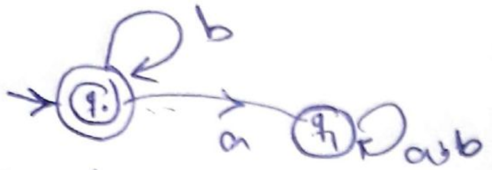
مثلاً :

$$\begin{cases} \delta(q_0, a) = q_1 \\ \delta(q_1, b) = q_2 \\ \delta(q_0, ab) = q_2 \end{cases} \xRightarrow[\text{بازگشتی}]{\text{با توجه به تعریف}} \begin{cases} \delta^*(q_0, ab) = \delta(\delta^*(q_0, a), b) \\ \delta^*(q_0, a) = \delta(\delta^*(q_0, \lambda), a) = \\ \qquad \delta(q_0, a) = q_1 \\ \delta^*(q_0, ab) = \delta(q_1, b) = q_2 \end{cases}$$

زبان‌های پذیرفته شده توسط اتوماتای متناهی قطعی (معین)

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

L(A) شامل تمام رشته‌های ورودی پذیرفته شده توسط A هست.



$L(A) = b^* = \{\lambda, b, b^2, b^3, \dots\}$

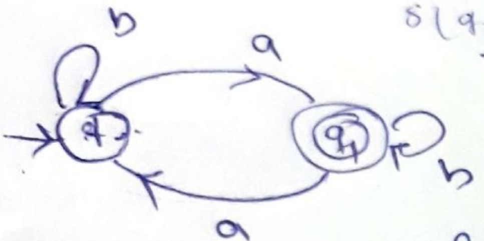
$= \{ b^n \mid n \geq 0 \}$

$Q = \{q_0, q_1\}$ $\qquad \delta(q_0, a) = q_1$

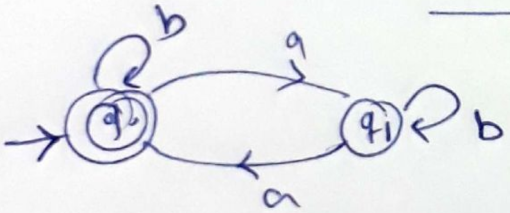
$\Sigma = \{a, b\}$ $\qquad \delta(q_0, b) = q_0$

$F = \{q_0\}$ $\qquad \delta(q_1, a) = q_1$

$\delta(q_1, b) = q_1$



L(A) زیرمجموعه‌ای از $\{a,b\}^*$ است که شامل کلماتی است که تعداد aها ی ظاهر شده در آن فرد باشد.

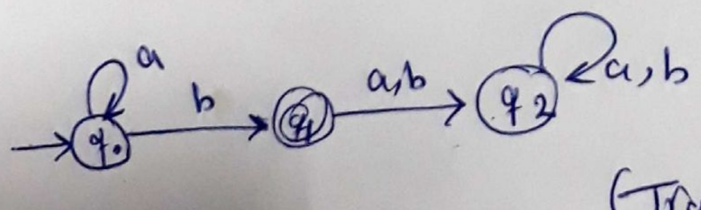


زبان زیرمجموعه‌ای از $\{a,b\}^*$ است که شامل کلماتی است که در آن

a تعداد $= 2K \quad , K \geq 0$

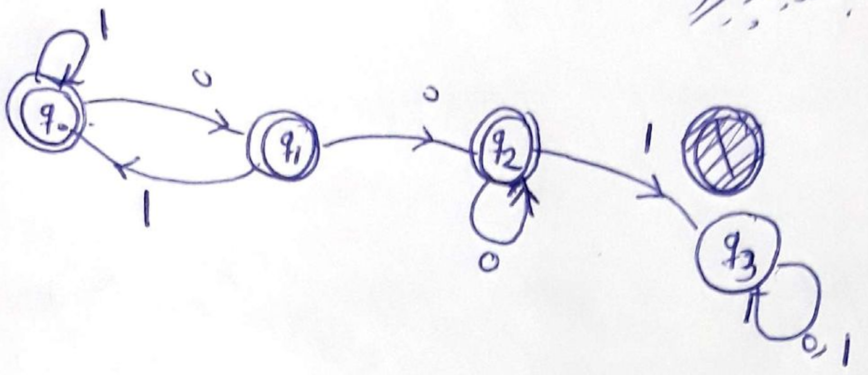
زبان پذیرفته شده توسط اتوماتای زیر عبارت است از

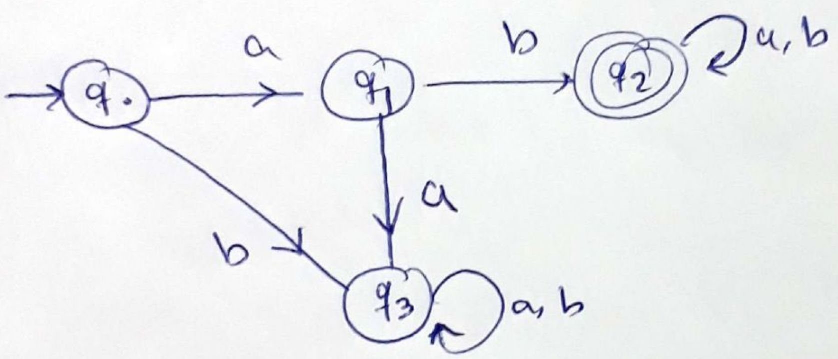
$L = \{ a^n b \mid n \geq 0 \}$



q_2: حالت تله (Trap state)

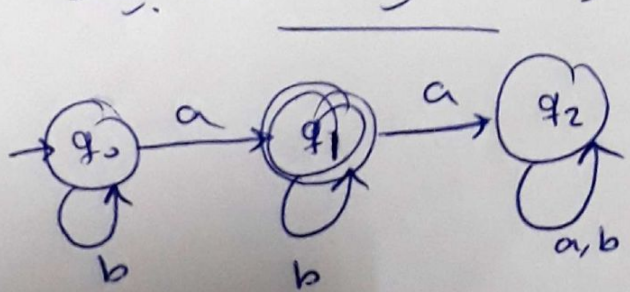
مثال: یک DFA طراحی کنید که تمام رشته‌های روی الفبای {0,1} را بجز رشته‌هایی که حاوی رشته "001" باشند بپذیرد.



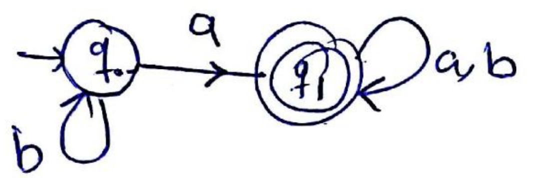


$L = \{ab\{a,b\}^*\}$ یعنی ab تمام رشته‌های با پیشوند }

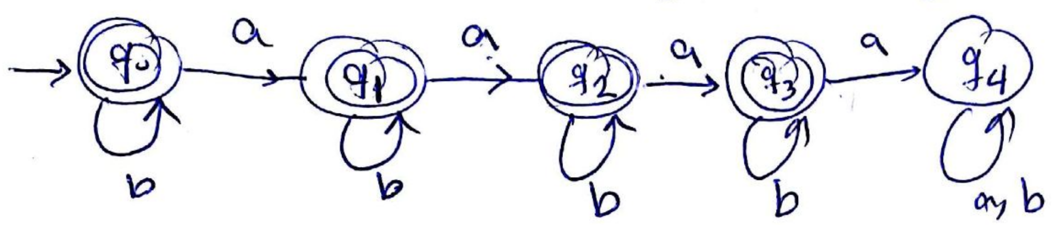
یک DFA طراحی کنید که تمام رشته‌هایی که فقط یک a دارند را بپذیرد.



یک DFA طراحی کنید که تمام رشته هایی که حداقل یک a دارند را بپذیرد.



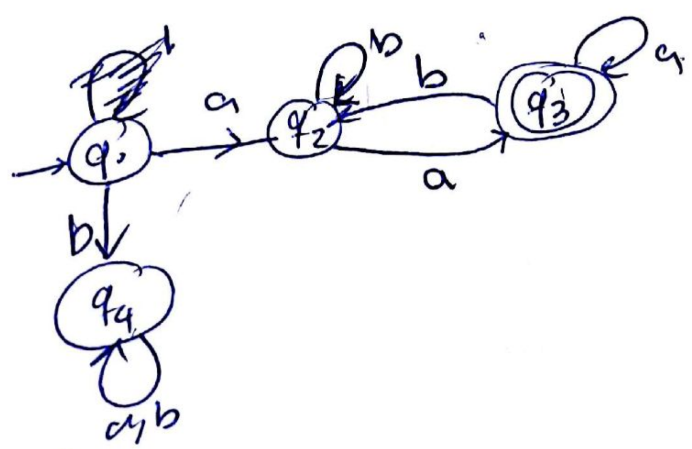
یک DFA طراحی کنید که تمام رشته هایی که حداکثر سه a دارند



زبان منظم: زبان L منظم است اگر یک DFA A موجود باشد بطوریکه $L = L(A)$

به عبارت دیگر زبان L منظم است هرگاه اتوماتای متناهی قطعی A وجود داشته باشد که L را بپذیرد.

مثال. زبان $L = \{awa : w \in \{a,b\}^*\}$ منظم است زیرا

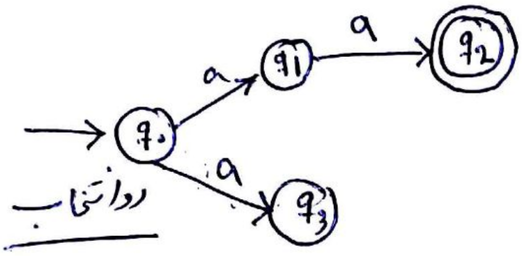


۳۰

تمرین: نشان دهید که $L^2 = \{awaawa \mid w \in \{a,b\}^*\}$ منظم است.

Nondeterministic Finite Automata (NFA)

$\Sigma = \{a\}$ $\quad Q = \{q_0, q_1, q_2, q_3\}$ $\quad F = \{q_2\}$



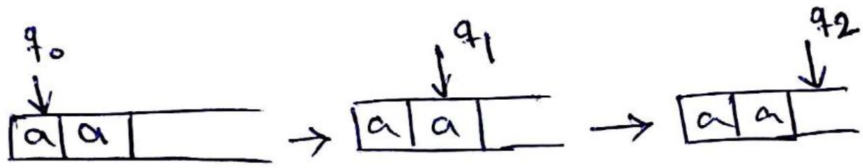
دو انتخاب

δ به صورت زیر است:

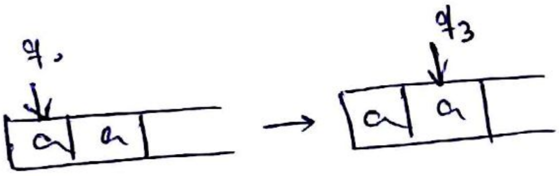
$\delta(q_0, a) = \{q_1, q_3\}$

$\delta(q_1, a) = \{q_2\}$

انتخاب اول:



انتخاب دوم:

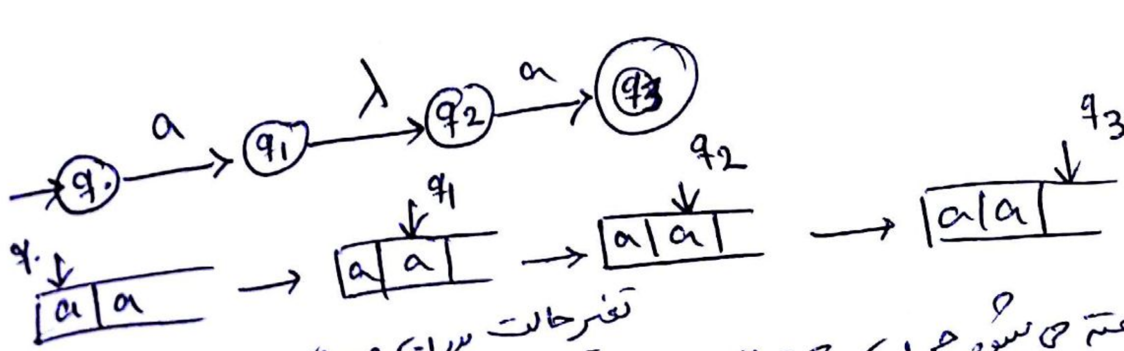


هیچ تبدیلی وجود ندارد.

یک (NFA) یک رشته (ورودی) می پذیرد هرگاه وجود داشته باشد یک محاسبه ای از NFA که رشته را بپذیرد.

در مثال ۱ و ۲ a^2 بوسیله NFA پذیرفته شد. (انتخاب اول)

تبدیل λ



تغییر حالت بدون ورودی

aa پذیرفته می شود چون $q_3 \in F$

Scanned by CamScanner

تعریف: اتوماتون حالت متناهی غیرقطعی A به صورت $A=(Q,\Sigma,\delta,q_0,F)$ هست که در آن:

Q: مجموعه متناهی از حالت‌ها است

Σ: مجموعه متناهی از ورودی‌ها است

$\delta: Q\times(\Sigma\cup\{\lambda\})\to 2^Q$ تابع تغییر حالت است

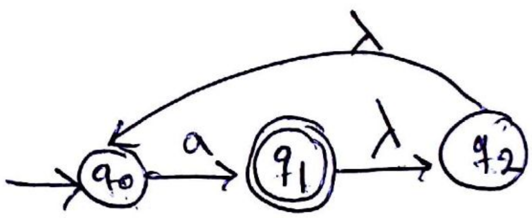
q_0: حالت آغازین

F: مجموعه حالت نهایی (پذیرش) است.

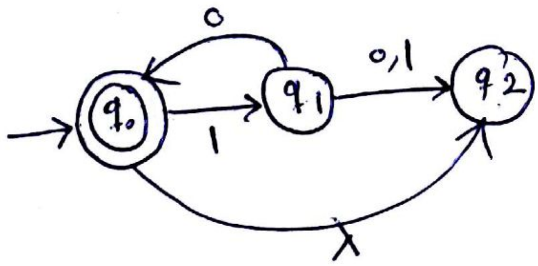
مثال ۱)

$\delta(q_0,a)=\delta(q_1,a)=\delta(q_2,a)=\{q_0,q_1,q_2\}$

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



مثال 2)



$\delta(q_0,1)=\{q_1\}$

$\delta(q_1,01)=\{q_0,q_2\}$

$\delta(q_0,\lambda)=\{q_0,q_2\}$

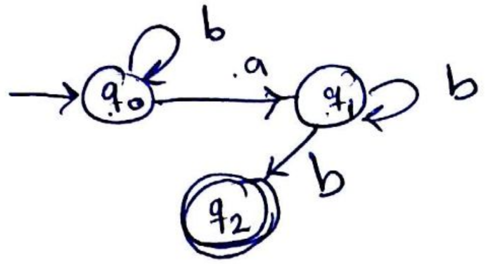
$\delta(q_2,1)=\phi$

تعریف: فرض کنید $A=(Q,\Sigma,\delta,q_0,F)$ یک NFA باشد. $x=x_1\cdots x_n$

را پذیرشی گوییم هرگاه

$$\exists\, q_0,q_1,\dots,q_n\in Q \quad \text{s.t.} \quad q_i\in\delta(q_{i-1},x_i) \quad i=1,\dots,n$$
$$\text{and} \quad q_n\in F$$

مثال 1) اتوماتون زیر را در نظر بگیرید. رشته $x=b^2ab^3$ توسط این اتوماتا پذیرفته می‌شود ولی $x=ba^2b$ پذیرشی نیست.



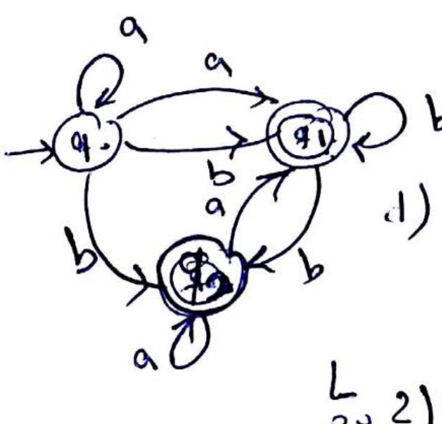
$$(q_0,b^2ab^3)\vdash(q_0,bab^3)\vdash(q_0,ab^3)\vdash(q_1,b^3)\vdash$$
$$(q_1,b^2)\vdash(q_1,b)\vdash(q_2,\lambda)$$

چون $q_2\in F$ پس رشته $x=b^2ab^3$ پذیرشی است.

$$(q_0,ba^2b)\vdash(q_0,a^2b)\vdash(q_1,ab)$$

پس رشته $x=ba^2b$ پذیرشی نیست.

مثال 2) برای $x=aba^2$ دو مسیر پذیرشی داریم:

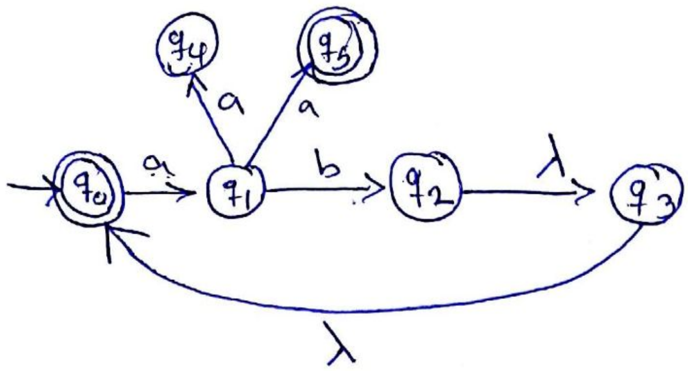


1) $(q_0,aba^2)\vdash(q_0,ba^2)\vdash(q_2,a^2)$
$\vdash(q_2,a)\vdash(q_1,\lambda)$

2) $(q_0,aba^2)\vdash(q_1,ba^2)\vdash(q_2,a^2)$
$\vdash(q_2,a)\vdash(q_1,\lambda)$

برای پذیرشی بودن کلمه در NFA کافیست که یک مسیر وجود داشته باشد که در شرایط پذیرش صدق کند.

تابع تبدیل توسعه یافته:



$\delta^*(q_0, a) = \{q_1\}$, $\delta^*(q_0, aa) = \{q_4, q_5\}$

$\delta^*(q_0, ab) = \{q_2, q_3, q_0\}$

* $q_j \in \delta^*(q_i, w)$ اگر و فقط اگر موجود باشد یک راه از q_i به q_j با برچسب w. ($w \in \Sigma^*$)

معنای عدم قطعیت (نامعین) (Nondeterministic)

* اتوماتای غیرقطعی می‌تواند بصورت غیرقطعی تصمیم صحیح را انجام دهد (بدون عقبگرد) را می‌توان تصور کرد به تعداد نامتناهی پردازنده وجود دارد. هرگاه با یک نماد ورودی دو مسیر وجود داشت یک پردازنده جدید فعال می‌شود و هر مسیر با ادامه‌ی ورودی توسط یک پردازنده دنبال می‌شود به محض اینکه یکی از پردازنده‌ها به حالت نهایی رسید به بقیه‌ی پردازنده‌ها سیگنال توقف می‌دهد.

** عدم قطعیت مکانیزم موثری برای توصیف زبان‌های پیچیده است (مثلاً $\{a^3\} \cup \{a^{2n} \mid n \geq 1\}$)

*** از اتوماتون‌های غیرقطعی می‌توان برای توصیف الگوریتم‌های جستجو با عقبگرد استفاده کرد

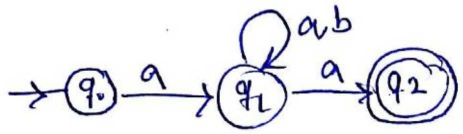
زبان یک NFA

زبان مورد قبول یک NFA، $A=(Q,\Sigma,\delta,q_0,F)$ عبارت است از:

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

یعنی ممکن است چندراه برای پذیرش یک رشته موجود باشد.

مثال ۱) برای زبان $L=\{awa \mid w\in\{a,b\}^*\}$ می توان NFA زیر را ارائه کرد:



هم ارزی DFA و NFA :

دو اتوماتای A_1 و A_2 هم ارزند اگر و فقط اگر $L(A_1)=L(A_2)$

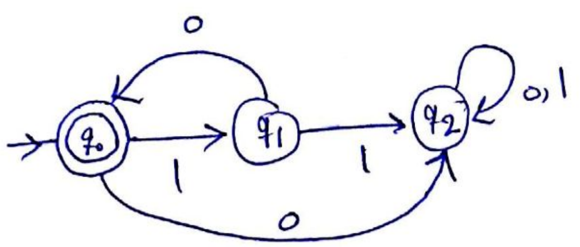
(زبان یکسانی را پذیرش کنند)

مثال 2) دو اتوماتای A_1 و A_2 زیر را در نظر بگیرید.

A_1 اتوماتون قطعی، A_2 اتوماتون غیرقطعی

$$A_1=(\{q_0,q_1,q_2\},\{0,1\},\delta_1,q_0,\{q_0\})$$

$$\begin{cases}\delta_1(q_0,1)=q_1 \\ \delta_1(q_0,0)=q_2\end{cases}\qquad \delta_1(q_1,1)=q_2 \quad \delta_1(q_1,0)=q_0 \qquad \delta_1(q_2,1)=q_2 \quad \delta_1(q_2,0)=q_2$$



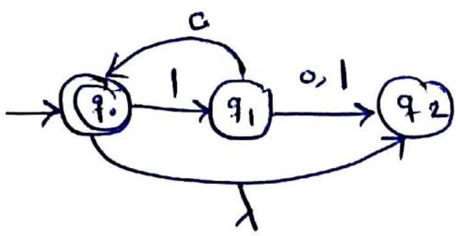
$$L(A_1)=\{(10)^n \mid n\geq 0\}$$

$$A_2=(\{q_0,q_1,q_2\},\{0,1\},\delta_2,q_0,\{q_0\})$$

δ_2 به صورت زیر است

$\delta_2(q_0,1)=\{q_1\}$ $\delta_2(q_0,\lambda)=\{q_2\}$

$\delta_2(q_1,0)=\{q_0,q_2\}$ $\delta_2(q_1,1)=\{q_2\}$



$L(A_2)=\{(10)^n \mid n\geq 0\}$

چون $L(A_1)=L(A_2)=\{(10)^n \mid n\geq 0\}$. پس اتوماتای A_1 و A_2 هم ارزند

سوال: NFAs ≡ DFAs ?

(قضیه) آیا NFAs و DFAs قدرت محاسبه یکسانی دارند؟

آیا زبان یکسانی را می‌پذیرند؟

اثبات: ما باید ثابت کنیم که زبان پذیرفته شده توسط NFAs با زبان پذیرفته شده توسط DFAs یکسان است. به عبارت دیگر

1) زبان پذیرفته شده توسط DFAs $\subseteq$ زبان پذیرفته شده توسط NFAs

2) زبان پذیرفته شده توسط NFAs $\subseteq$ زبان پذیرفته شده توسط DFAs

اثبات 1) هر DFA یک NFA هست. زبان پذیرفته شده توسط یک DFA، توسط یک NFA نیز پذیرفته می‌شود.

اثبات 2) هر NFA می‌تواند به یک DFA هم ارز تبدیل شود. زبان پذیرفته شده توسط یک NFA، توسط یک DFA نیز پذیرفته می‌شود.

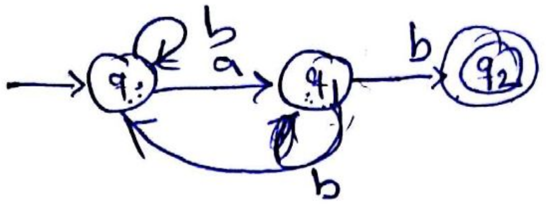
الگوریتم تبدیل NFA به DFA

مثال) اتوماتون زیر را در نظر بگیرید:

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

$Q = \{q_0, q_1, q_2\} \qquad \Sigma = \{a, b\} \qquad F = \{q_2\}$

$$\begin{cases} \delta(q_0, a) = \{q_1\} \\ \delta(q_0, b) = \{q_0\} \\ \delta(q_1, a) = \phi \\ \delta(q_1, b) = \{q_0, q_2\} \\ \delta(q_2, a) = \delta(q_2, b) = \phi \end{cases}$$



واضح است که اتوماتون A غیرقطعی است. حال ما اتوماتون غیرقطعی A' را به صورت زیر در نظر می‌گیریم:

$A' = (\overbrace{P(Q)}^{2^Q}, \Sigma, \delta', \{q_0\}, F')$

حالتها

$F' = \{B \subseteq Q \mid B \cap F \neq \phi\}$

$P(Q) = \{\phi, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0, q_1\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_0, q_1, q_2\}\}$

$F' = \{\{q_2\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_0, q_1, q_2\}\}$

$\delta' : \underline{P(Q)} \times \Sigma \longrightarrow \underline{P(Q)}$

$\delta'(B, x) = \bigcup_{q \in B} \delta(q, x)$

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

$\delta'(\{q_0\}, b) = \{q_0\}$

$\delta'(\{q_1\}, a) = \phi$

$\delta'(\{q_1\}, b) = \{q_0, q_2\}$

$\delta'(\{q_2\}, a) = \delta'(\{q_2\}, b) = \phi$

$\delta'(\{q_0, q_2\}, a) = \delta(q_0, a) \cup \delta(q_2, a) = \{q_1\}$

$\delta'(\{q_0, q_2\}, b) = \delta(q_0, b) \cup \delta(q_2, b) = \{q_0\}$

$\delta'(\{q_0,q_1\},a) = \delta(q_0,a) \cup \delta(q_1,a) = \{q_1\}$

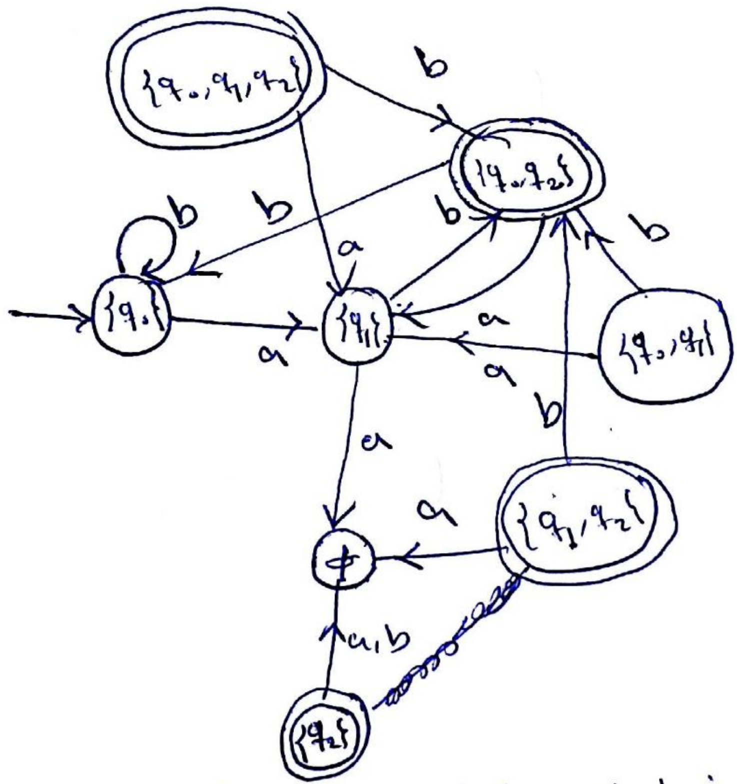
$\delta'(\{q_0,q_1\},b) = \delta(q_0,b) \cup \delta(q_1,b) = \{q_0,q_2\}$

$\delta'(\{q_1,q_2\},a) = \delta(q_1,a) \cup \delta(q_2,a) = \phi$

$\delta'(\{q_1,q_2\},b) = \delta(q_1,b) \cup \delta(q_2,b) = \{q_0,q_2\}$

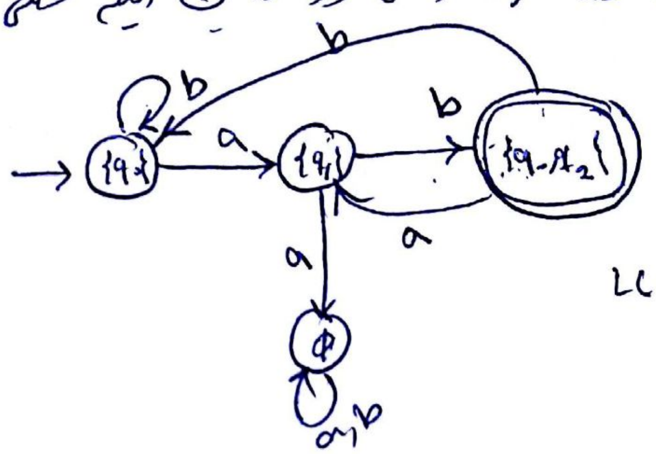
$\delta'(\{q_0,q_1,q_2\},a) = \{q_1\}$

$\delta'(\{q_0,q_1,q_2\},b) = \{q_0,q_2\}$



نکته: گراف (اتوماتا) روبرو حداکثر دارای $2^{|Q|}|\Sigma|$ یال است.

حال با حذف حالت‌هایی که هیچگونه ارجاعی به آن وجود ندارد مانند $\{q_2\}$ و $\{q_1,q_2\}$ و $\{q_0,q_1,q_2\}$ و $\{q_0,q_1\}$ ، اتوماتای مورد نظر ساده می‌شود در عین اینکه خللی به کلمات پذیرش وارد نمی‌شود.



$L(A) = \{ b^n a b (b^m a b)^* \mid n \geq 0, m \geq 0 \}$

w رشته‌ای است دلخواه

	a	b
$\{q_0\}$	$\{q_1\}$	$\{q_0\}$
$\{q_1\}$	ϕ	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_1\}$	$\{q_0\}$

پس الگوریتم تبدیل NFA به DFA به صورت زیر است:

۱) هر حالت DFA به صورت زیرمجموعه‌ای از حالات NFA است

$$Q_D = 2^{Q_N}$$

۲) تابع تبدیل حالت برای هر حالت به صورت زیر تعریف می‌شود

$$\{q_0, q_1, \ldots, q_n\}$$

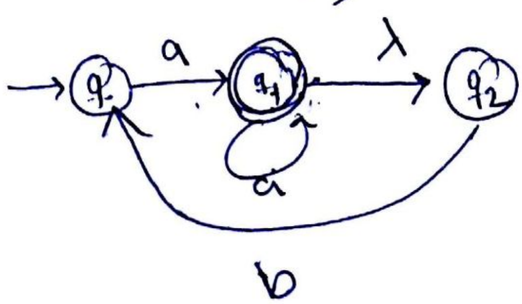
$$\delta_D(\{q_0, q_1, \ldots, q_n\}, a) = \bigcup_{i=0}^{n} \delta_N(q_i, a)$$

۳) مجموعه حالات نهایی DFA شامل حالت‌هایی از Q_D خواهد بود که حداقل یکی از عناصر F_N در آن‌ها باشد.

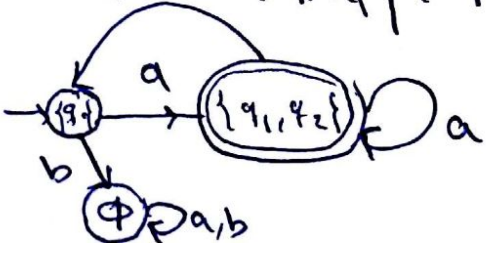
۴) حالت اولیه DFA به صورت $\{q_0\}$ می‌باشد.

نتیجه: زبان پذیرفته شده توسط NFA یک زبان منظم است.

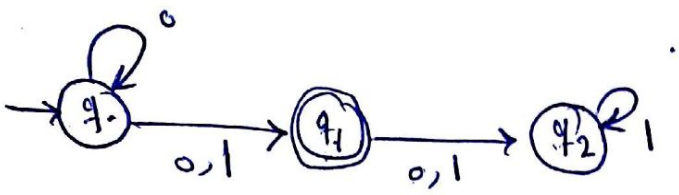
مثال ۲) (تبدیل NFA به DFA). اتوماتون زیر را در نظر بگیرید



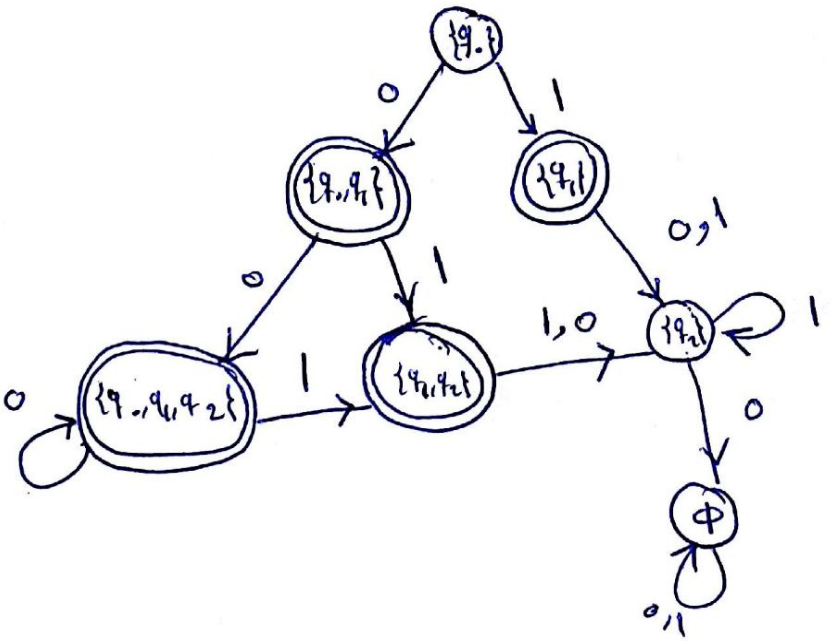
	a	b
$\{q_0\}$	$\{q_1, q_2\}$	ϕ
$\{q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_0\}$



مثال ۳) NFA زیر را به DFA تبدیل کنید.



	0	1
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_1\}$
$\{q_1\}$	$\{q_2\}$	$\{q_2\}$
$\{q_2\}$	ϕ	$\{q_2\}$
$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$
$\{q_1, q_2\}$	$\{q_2\}$	$\{q_2\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$
ϕ	ϕ	ϕ



نکته: تمام زبان‌های متناهی، منظم هستند

نکته دیگر: اگر L منظم باشد، آنگاه L^R هم منظم است. همچنین $\overline{L}$ هم منظم است

کاهش تعداد حالت‌های FA :

زبان یک DFA منحصر به فرد است ولی عکس آن برقرار نیست. یعنی DFAs مختلف می‌توانند با تعداد حالت‌های متفاوت دارای یک زبان باشند. برای استفاده بهینه از حافظه، مطلوب است تا جایی که امکان دارد تعداد حالت‌ها را کاهش دهیم.

ادغام پذیری حالت‌ها (غیرقابل تمایز) (indistinguishable)
تمایزناپذیر

دو حالت p و q از یک DFA درصورتی ادغام پذیر هستند که:

$$\forall w \in \Sigma^* \begin{cases} \delta^*(p,w) \in F \rightarrow \delta^*(q,w) \in F \\ \delta^*(p,w) \notin F \rightarrow \delta^*(q,w) \notin F \end{cases}$$

در غیر این صورت p و q غیرقابل تمایزند (تمایزناپذیرند)

* اگر رشته‌ای مانند $w \in \Sigma^*$ وجود داشته باشد بطوریکه

$$\delta^*(p,w) \in F \ , \ \delta^*(q,w) \notin F$$

یا

$$\delta^*(p,w) \notin F \ , \ \delta^*(q,w) \in F$$

آنگاه حالت‌های p و q را ادغام ناپذیر بواسطه رشته w می‌نامیم.

ادغام پذیر بودن دارای یک رابطه هم‌ارزی است.

روال علامت گذاری:

1) تمام حالت‌های غیرقابل دسترس را حذف نمایید. (حالت‌هایی که هیچگونه ارجاعی به آنها داده نشده) (حالت‌هایی که ورودی ندارند)

2) تمام زوج‌های حالت (p,q) را بررسی نمایید. اگر $p \in F$, $q \notin F$ و یا برعکس باشد آنها را بعنوان ادغام ناپذیر علامت بزنید.

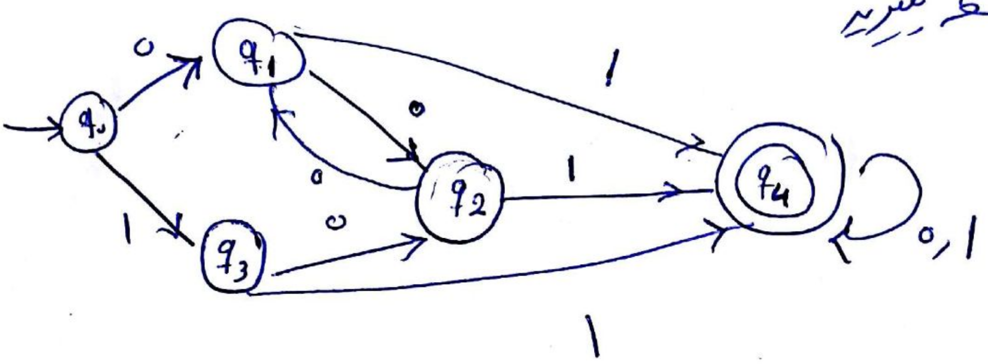
3) کاهش زیر را آنقدر تکرار کنید تا زوجی برای علامت گذاری باقی نماند.

برای همه زوج های (p,q) و هر $a \in \Sigma$ تابع زیر را محاسبه نمایید

$$\delta(q,a) = q_a \quad , \quad \delta(p,a) = p_a$$

اگر زوج (p_a,q_a) بعنوان ادغام ناپذیر علامت گذاری شده باشد، آنگاه (p,q) را بعنوان ادغام ناپذیر علامت بزنید

مثال اتوماتای زیر را در نظر بگیرید



روال علامت گذاری

1- حالت غیر قابل دسترسی وجود ندارد.

2 - زوج های ادغام ناپذیر: (q_3,q_4) ، (q_2,q_4) ، (q_1,q_4) ، (q_0,q_4)

چون $q_4 \in F$ و $q_0, q_1, q_2, q_3 \notin F$

3 -

(q_1,q_4) با 1: ندارد

با 0:
$$\begin{cases} \delta(q_0,0) = q_1 \\ \delta(q_2,0) = q_1 \\ \delta(q_4,0) = q_4 \end{cases}$$

چون (q_1,q_4) ادغام ناپذیرند پس (q_2,q_4) ، (q_0,q_4) نیز ادغام ناپذیرند

(q_2,q_4) با 1: ندارد

با 0:
$$\begin{cases} \delta(q_1,0) = q_2 \\ \delta(q_3,0) = q_2 \\ \delta(q_4,0) = q_4 \end{cases}$$

چون (q_2,q_4) ادغام ناپذیرند پس (q_3,q_4) ، (q_1,q_4) نیز ادغام ناپذیرند

$(q_3, q_4) \xrightarrow{0}$ تولید

$(q_3, q_4) \xrightarrow{1}$
$$\begin{cases} \delta(q_0,1)=q_3 \\ \delta(q_1,1)=q_4 \\ \delta(q_2,1)=q_4 \\ \delta(q_3,1)=q_4 \\ \delta(q_4,1)=q_4 \end{cases}$$

چون (q_3,q_4) ادغام ناپذیرند پس زوج حالت‌های زیر نیز ادغام ناپذیر هستند
$(q_0,q_1)\ (q_0,q_2)\ (q_0,q_3)$
(q_0,q_4)

پس داریم:

زوج‌های ادغام ناپذیر $= \{(q_0,q_4), (q_1,q_4), (q_2,q_4), (q_3,q_4), (q_0,q_1), (q_0,q_2), (q_0,q_3)\}$

مجموعه ادغام پذیر $= \{(q_1,q_2), (q_1,q_3), (q_2,q_3)\}$

$\equiv \Longrightarrow q_1 \equiv q_2,\ q_2 \equiv q_3,\ q_1 \equiv q_3 \Longrightarrow q_1 \equiv q_2 \equiv q_3$

روال کاهش:

یک DFA باشد $A=(Q,\Sigma,\delta,q_0,F)$ داده شده است یک DFA کاهش یافته باشد $A'=(Q',\Sigma',\delta',q'_0,F')$ به صورت زیر می‌سازیم.

$Q' = Q/\equiv$, $\Sigma' = \Sigma$

$q'_0 = [q_0]$, $F' = F/\equiv$

$\delta' : Q' \times \Sigma \rightarrow Q'$

تعریف: $[q] = \{p \in Q \mid q \equiv p\}$

کلاس هم ارزی q

در مثال بالا روال کاهش به صورت زیر است

$Q' = \{[q_0], [q_1], [q_4]\}$

$F' = \{[q_4]\}$, $\delta'([q_0],0)=[q_1]$, $\delta'([q_4],0)=[q_4]$

$\delta'([q_0],1)=[q_1]$, $\delta'([q_4],1)=[q_4]$

$\delta'([q_1],0)=[q_1]$

$\delta'([q_1],1)=[q_4]$