جزوه نظریه زبانها و ماشینها

مقدمه: زمینه‌های مشترک در زبانها:

یک زبان کلکسیونی است از جملات(sentences)، یک جمله دنباله‌ای است ازکلمات(words) و یک کلمه ترکیبی است از نمادها. در مبحث نظریه زبانها و ماشین‌ها سه موضوع اساسی شامل **زبانها**(Languages) **, گرامرها** (Grammars) **و ماشین‌ها یا اوتوماتا** (Automata) مورد توجه قرار می‌گیرند.

زبانها

آموزش یک زبان مستلزم ۳مرحله به شرح زیر است:

۱. یادگیری الفباء زبان یعنی نمادهائی که در زبان بکار می‌روند.

۲. یادگیری کلمات یعنی دنباله‌های متعدد از نمادهای متعلق به الفباء زبان.

۳. یادگیری تشکیل جملات: دنباله‌هائی از کلمات که ضابطه و قانون ویژه‌ای از زبان را دنبال می‌کنند.

تعاریف.

✓**.الفباء** (alphabet) : مجموعه متناهی از نمادها یا کاراکترها. الفباء را معمولا با نماد Σ نشان می‌دهند.

✓**.رشته بر روی** Σ (string): دنباله متناهی از کاراکترهارا یک رشته گویند.

✓**.طول رشته**(length) **:** تعداد کاراکترهای تشکیل دهنده رشته. طول رشته w را با $|w|$ نشان می‌دهند.

✓**.رشته تهی** (empty string) **:** رشته‌ی فاقد نماد و آن را با نمادهای ε و λ نشان می‌دهند. و $|\lambda| = 0$

✓**.وارون یک رشته**(Reverse) **:** رشته‌ای که از نوشتن نمادهای آن به ترتیب عکس بدست می‌آید.

✓**. زیر رشته** (substring). هر رشته‌ی بدست آمده از تعدادی از کاراکترهای متوالی درون رشته مفروض w را یک زیر رشته‌ی w گویند. اگر $w = vu$ آنگاه v و u زیر رشته‌های w هستند.در رشته $w = vu$, v را **پیشوند**(prefix) **و** u **را پسوند**(suffix) رشته w گویند.

✓**. الحاق**(concatenation) **:** فرض کنید w و vدو رشته است. wv یعنی الحاق w و v و عبارت از رشته ای است که از الحاق نمادهای vبه انتهای راست نمادهای w بدست می‌آید.

✓. اگر w رشته‌ی مفروض باشد آنگاه $w^0 = \lambda$ و $w^n = \underbrace{ww\cdots w}_{n\ \ times}$.

توجه:

- رشته λ عضو همانی نسبت به عمل الحاق است یعنی به ازاء هر رشته w , $\lambda w = w\lambda = w$
- فرض کنید $a \in \Sigma$و w یک رشته بر روی Σاست. تعداد تکرار a در رشته w را با نماد $|w|_a$ نشان می‌دهیم. مثلا اگر $w = ababaaa$ آنگاه $|w|_a = 5$ و $|w|_b = 2$

مثال۱. $\Sigma = \{a,b\}$ الفباء و $w = ababaaa$ یک رشته است. در این رشته $|w| = 7$, $w^R = aaababa$

مثال۲. اگر $w = a_1a_2\cdots a_n$ و $v = b_1b_2\cdots b_m$ آنگاه $wv = a_1a_2\cdots a_nb_1b_2\ldots b_m$ و در این مثال $|w| = n$, $|v| = m$ و $|wv| = |w| + |v| = n + m$

مثال۳. اگر $w = \{abbab\}$ آنگاه $\{\lambda, a, ab, abb, abba, abbab\}$ مجموعه پیشوندها و $\{\lambda, b, ab, bab, bbab, abbab\}$ مجموعه پسوندهای w هستند.

مثال۴. با فرض $w = (abba)$ نتیجه می‌گیریم: $(abba)^0 = \lambda$ و $w^2 = abbaabba = ab^2a^2b^2a$

تعاریف. اگر مجموعه‌ی ناتهی و متناهی الفباء را با Σ نشان دهیم آنگاه

- Σ^* : مجموعه کلیه رشته‌های ممکن و متناهی از الفباء Σ که رشته‌ی تهی λ را نیز شامل می‌شود.
- Σ^+: یعنی مجموعه کلیه رشته‌های ممکن و متناهی و غیر تهی از الفباء Σ. یعنی $\Sigma^+ = \Sigma^* - \{\lambda\}$
- Σ^n یعنی مجموعه کلیه رشته‌های متناهی به طول n .

مثال۵.

$$\Sigma = \{a,b\} \Rightarrow \begin{cases} \Sigma^* = \{\lambda, a, b, aa, ba, bb, aaa, aab, \cdots\} \\ \Sigma^+ = \{a, b, aa, ba, bb, aaa, aab, \cdots\} \\ \Sigma^2 = \{aa, ab, ba, bb\} \end{cases}$$

توجه: Σ^* هیچ وقت تهی نیست زیرا همیشه $\lambda \in \Sigma^*$. و اگر $\Sigma = \varnothing$(مجوعه تهی) آنگاه $\Sigma^* = \{\lambda\}$

مثال۶. $\Sigma = \{a,b\}$الفباء مفروض

الف. $\Sigma^0, \Sigma^1, \Sigma^2, \Sigma^3$ را بدست آورید.

ب. با فرض $A = \Sigma^0 \cup \Sigma^1$ و $B = \Sigma^2 \cup \Sigma^3$ به صورت کلامی A و B و $A \cup B$ را توضیح دهید.

جواب.

الف.

$$\Sigma^0 = \{\lambda\}, \quad \Sigma^1 = \{a,b\}, \quad \Sigma^2 = \{aa, ab, ba, bb\},$$
$$\Sigma^3 = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$$

ب. A مجموعه‌ای است از کلیه رشته‌ها بر روی Σ حداکثر بطول ۱, و B مجموعه‌ای است از کلیه رشته‌ها بر روی Σ به طول ۲ یا ۳ و $A \cup B$ مجموعه‌ای است از کلیه رشته‌ها بر روی Σ حداکثر بطول ۳.

زبان (Language)

هر مجموعه‌ای از رشته‌های متناهی از عناصر Σ به عبارت دیگر هر زیرمجموعه‌ای از Σ^*رایک زبان گویند.

مثال۷. اگر $\Sigma=\{a,b\}$ الفباء مفروض و $\Sigma^*=\{\lambda,a,b,aa,ba,bb,aaa,aab,\cdots\}$ آنگاه هر یک از $L_1=\{a,aa,aab\}$, $L_2=\{\lambda,abba,baba,aa,ab\}$ و $L_3=\{a^nb^n \mid n\geq 0\}$ چند نمونه از زبانهای مختلف بر روی $\Sigma=\{a,b\}$است. تعدادی از رشته‌های متعلق به زبان L_3 در زیر ارائه شده است.

$$L_3=\{a^nb^n \mid n\geq 0\}\Rightarrow \left.\begin{matrix}\lambda\\ ab\\ aabb\\ aaabbb\end{matrix}\right\}\in L_4\,,\qquad abb\notin L_4$$

مثال۸. با فرض $\Sigma=\{a,b\}$ زبانهای L_1 و L_2 تعریف شده به شرح زیر را مشخص کنید.

الف. L_1 شامل رشته‌هائی است که با کاراکتر a آغاز و طول هر رشته حداکثر ۳ باشد.

ب. L_2 شامل رشته‌هائی است که از هر دو طرف یک جور خوانده می‌شوند(palindrome) .مانند aa , aba, baab , madamimadam و نظایر آن.

جواب. الف. $L_1=\{a,aa,ab,aaa,aab,aba,abb\}$

ب. $L_2=\{\lambda,a,b,aa,bb,aaa,bab,abba,babaabab,\cdots\}$

توجه:

- $\Sigma=\{a,b\}$یعنی یک الفبا که a و b کاراکترهای آن است. و لی $\Sigma^1=\{a,b\}$ یعنی مجموعه‌ای از رشته‌های a و b و هر رشته به طول یک.
- مجموعه تهی $\varnothing$ و $\{\lambda\}$ زبان بر روی هر الفباء هستند و $\varnothing\neq\{\lambda\}$چون زبان $\varnothing$ فاقد رشته است ولی $\{\lambda\}$ دارای یک رشته بنام λاست.

عملیات بر روی زبان‌ها(operations on languages)

چون زبان یک مجموعه است بنابراین عملیات متعدد بر روی مجموعه‌ها بر روی زبانها نیز صادق است. فرض کنید x و y رشته‌هائی بر روی Σ هستند. به ازاء هر زبان L و L' بر روی Σ و استفاده از عملیاتی به شرح زیر می‌توانیم به زبان‌های جدید به صورت زیر دسترسی داشته باشیم:

- ✓ عمل **اجتماع**: $L\cup L'=\{x \mid x\in L\vee x\in L'\}$
- ✓ عمل **اشتراک**: $L\cap L'=\{x \mid x\in L\wedge x\in L'\}$
- ✓ عمل **تفاضل**: $L-L'=\{x \mid x\in L\wedge x\notin L'\}$
- ✓ **عمل متمم‌گیری**: $\overline{L}=\Sigma^*-L$
- ✓ **عمل وارون**: $L^R=\{w^R : w\in L\}$
- ✓ **عمل الحاق**: $LL'=\{xy : x\in L\wedge y\in L'\}$

مثال۹. اگر $L=\{a,ab,aaaa\}$ و $L'=\{bb,ab\}$ آنگاه

$$L\cup L'=\{a,ab,bb,aaaa\},\quad L\cap L'=\{ab\}\ \text{و}\ L-L'=\{a,aaaa\}$$

مثال۱۰. اگر $L=\{a,ba\}$ آنگاه $\overline{L}=\Sigma^*-L=\overline{\{a,ba\}}=\{\lambda,b,aa,bb,aaa,\cdots\}$

مثال۱۱. اگر $L=\{ab,aab,baba\}$ آنگاه $L^R=\{ba,baa,abab\}$

مثال۱۲. اگر $L=\{a,ab,ba\}$ و $L'=\{b,aa\}$ آنگاه

$$LL'=\{a,ab,ba\}\{b,aa\}=\{ab,aaa,abb,abaa,bab,baaa\}$$

تعریف L^n : $L^n=\underbrace{LL\cdots L}_{n\ times}$ یعنی الحاق L به تعداد n بار با خودش. $L^0=\{\lambda\}$ و $L^1=L$

مثال۱۳. اگر $L=\{a,b\}$ آنگاه

$$L^3=\{a,b\}\{a,b\}\{a,b\}=\{aaa,aab,aba,abb,baa,bab,bba,bbb\}$$

✓ L^* (star-closure) : $L^*=L^0\cup L^1\cup L^2\ldots=\bigcup_{n\geq 0}L^n$

✓ L^+ (positive-closure) : $L^+=L^1\cup L^2\cup L^3\ldots=\bigcup_{n\geq 1}L^n$

مثال۱۴. با فرض $L=\{a^nb^n\mid n\geq 0\}$ مطلوب است L^R و L^2

جواب. $L^R=\{b^na^n\mid n\geq 0\},\quad L^2=\{a^nb^na^mb^m\mid n\geq 0,m\geq 0\}$

توجه داشته باشید که در تعریف L^2 , m و n ربطی بهم ندارند. مثلا $aabbaaabbb\in L^2$

توجه: فرض کنید w یک رشته یا یک کاراکتر و L یک زبان است. $L^0=\{\lambda\}$ ولی $w^0=\lambda$

مثال۱۵. با توجه به تعریف عمل* حاصل be^*t , $(1100)^*$, $(00+11)^*$ و $(0+1)^*(00+11)$ را بدست آورید.

جواب.

$$be^*t=\{bt,bet,beet,beeet,\ldots\}$$

$$(1100)^*=\{\lambda,1100,11001100,1100110011\ \ 00,\ldots\}$$

$$(00+11)^*=\{\lambda,00,11,0000,0011,1100,1111,000000,000011,001100,\ldots\}$$

$$(0+1)^*(00+11)=is\ \ all\ \ strings\ \ of\ \ zeros\ \ and\ \ ones\ \ that\ \ end\ \ with\ \ either\ \ 00\ \ or\ \ 11$$

معرفی یک زبان با استفاده از خواص رشته‌های تشکیل دهنده آن

اغلب می‌توانیم انواع زبانهای رسمی را به سادگی از طریق بیان خواص رشته‌های تشکیل دهنده آنها توضیح دهیم. با استفاده از این روش نه تنها توضیح مناسب یک زبان بلکه تفهیم خواص آن به نحو مطلوبی امکان پذیر می‌گردد. (به مثالهای زیر توجه کنید)

مثال۱۶. زبانهای زیر را با توجه به خواص رشته‌های تشکیل دهنده آنها به صورت مجموعه نشان دهید.

الف. مجموعه‌ی کلیه رشته‌ها برروی الفباء $\Sigma = \{a,b,c\}$ به قسمی‌که ac زیر رشته آنها می‌باشد.

ب. مجموعه کلیه رشته‌ها برروی Σ به قسمی‌که کاراکتر a به تعداد زوج در هر رشته ظاهر شده باشد

ج. مجموعه کلیه رشته‌ها برروی الفباء Σ به قسمی‌که رشته‌ها از هر دو طرف یک جور خوانده شوند.

د. مجموعه کلیه رشته‌ها برروی الفباء مفروض به قسمی‌که کاراکتر a در هر رشته در محل پنجم از سمت راست آن رشته ظاهر شده باشد.

ه مجموعه کلیه رشته‌هائی بر روی Σ که تعداد aها برابر تعداد bها باشد

جواب.

الف. $L = \{xacy \mid x,y \in \{a,b,c\}^*\}$ یا $L = \{x \in \{a,b,c\}^* \mid |x|_{ac} \geq 1\}$

ب. $L = \{x \in \Sigma^* \mid |x|_a = 2n, n \in N\} = \{x \in \Sigma^* \mid |x|_a \equiv 0 \mod(2)\}$

ج. $L = \{x \in \Sigma^* \mid x = x^R\}$

د. $L = \{xay \mid x,y \in \Sigma^*, |y| = 4\}$

ه $L = \{x \in \Sigma^* \mid |x|_a = |x|_b\}$

عبارات منظم(با قاعده)(regular expressions)

عبارت‌های منظم بر روی الفباء مفروض Σ به صورت زیر تعریف می‌شود.

- ✓ رشته λ و زبان تهی $\varnothing$ و هر یک از کاراکترهای الفباء Σ عبارتهای منظم Σ هستند.
- ✓ اگر r و s عبارتهای منظم باشند آنگاه عبارتهای rs (الحاق r و s), $(r)^*$, $(s)^*$ و $r \mid s$ (هر یک از r یا s) نیز منظم هستند. ($r \mid s$ را به صورت $r \cup s$ و $r+s$ نیز نشان می‌دهند.)
- ✓ هر عبارت منظم با استفاده از تعداد متناهی و به صورت بازگشتی از قوانین فوق بدست می‌آید.

یک عبارت منظم یک رشته متنی است که به عنوان یک الگوی جستجو مورد استفاده قرار می‌گیرد.

تعریف. اگر r یک عبارت منظم باشد, آنگاه زبانی که توسط r ارائه می‌شود را با L(r) نشان می‌دهیم.

به عنوان مثال زبان منظم $L(a) = \{a^n \mid n \geq 0\}$ را می‌توان بوسیله عبارت منظم a^* نشان داد.

مثال۱۷.مجموعه‌ای از کلیه رشته‌ها بر روی $\{a,b\}$ به طوری که ab زیر رشته کلیه رشته‌ها باشد یک زبان منظم است. می‌توانیم مجموعه‌ی مورد نظر را به صورت زیر نشان دهیم:

$$L = \{x \in \{a,b\}^* \mid ab \quad is \quad substring \quad of \quad x\}$$
$$= \{yabz \mid y,z \in \{a,b\}^*\} = \{\{a,b\}^* ab \{a,b\}^*\}$$

بنابراین $(a+b)^* ab (a+b)^*$ یک عبارت منظم است که زبان L را نشان می‌دهد.

مثال۱۸. عبارتهای منظم $(a+b)^*(a+bb),(ab)^+,a^*b^*,(a+b)^2$ را بر روی $\Sigma=\{a,b\}$ تعریف کنید.

جواب.

$(a+b)^2$: یعنی مجموعه کلیه رشته‌ها به طول۲از الفباء : $\{aa,ab,ba,bb\}$.

a^*b^*: مجموعه رشته‌ها شامل صفر یابیشتر از aها که توسط صفر یا بیشتر از هاb دنبال شده است.

$(ab)^+$: مجموعه رشته‌ها شامل یک یا بشتر از abها: $\{ab,abab,ababab,\cdots\}$.

$(a+b)^*(a+bb)$: مجموعه‌ای از رشته‌هائی بر روی $\Sigma=\{a,b\}$ که به a یا bb ختم شده باشد.

مثال۱۹. عبارت $(aa)^*(bb)^*b$ یعنی مجموعه‌ای از کلیه رشته‌ها بر روی $\Sigma=\{a,b\}$ شامل تعداد زوج از aها که با تعداد فرد از bها دنبال شده باشد. به عبارت دیگر $L(r)=\{a^{2n}b^{2m+1}\mid n\geq 0,m\geq 0\}$

مثال۲۰. عبارتهای منظم زیر را که بر روی $\Sigma=\{0,1\}$ تعریف شده‌اند به صورت کلامی بیان کنید.

- ✓ $(0\mid 1)^*111$: مجموعه رشته‌های مختوم به ۳تا 1 متوالی.
- ✓ $0^*1(0\mid 1)^*$: مجموعه از رشته‌ها که حداقل دارای یک نماد 1 می‌باشد.
- ✓ $0^*\mid 0^*10^*$: مجموعه از رشته‌ها که حداکثر دارای یک نماد 1 می‌باشد.

مثال۲۱. عبارتهای منظم تعریف شده بر روی $\Sigma=\{a,b,c\}$ به قسمی‌که برای هر رشته شرایط زیر منظور شده باشد را نشان دهید.

الف. تعداد aها در هر رشته زوج باشد **ب.** تعداد bها در هر رشته برابر $4i+1,i\geq 0$ باشد.

جواب الف. $(b+c)^*((a(b+c)^*)^2)^*$ **جواب ب.** $(b+c)^*b(a+c)^*((b(a+c)^*)^4)^*$

خواص عبارت‌های منظم

خواص زیر برای زبانهای منظم و گرامرهای منظم حاصل از آنها صدق می‌کند. (نماد $\approx$ نماد معادل است)

1. $r\lambda\approx\lambda r\approx r$
2. $r_1r_2\neq r_2r_1 \quad in \quad general$
3. $r_1(r_2r_3)\approx(r_1r_2)r_3$
4. $r\varnothing\approx\varnothing r\approx\varnothing$
5. $\varnothing^*\approx\lambda$
6. $\lambda^*\approx\lambda$
7. $if \quad \lambda\in L(r) \quad then \quad r^*\approx r^+$
8. $rr^*\approx r^*r\approx r^+$
9. $(r_1+r_2)r_3\approx r_1r_3+r_2r_3$
10. $r_1(r_2+r_3)\approx r_1r_2+r_1r_3$
11. $(r^*)^*\approx r^*$
12. $(r_1r_2)^*r_1\approx r_1(r_2r_1)^*$
13. $(r_1+r_2)^*\approx(r_1^*r_2^*)^*$

مثال. ثابت کنید $b^+(a^*b^*+\lambda)b\approx b^+a^*b^+$

اثبات.

$$b^+(a^*b^*+\lambda)b\approx(b^+a^*b^*+b^+\lambda)b\approx b^+a^*b^*b+b^+b\approx b^+a^*b^++b^+b$$

$$L(b^+b)\subseteq L(b^+a^*b^+)\ \Rightarrow\ b^+a^*b^++b^+b\approx b^+a^*b^+$$

گرامرها (Grammars)

طبق تعریف, زبان رسمی بر روی Σ مجموعه‌ای است از رشته‌های متناهی از عناصر Σ. ولی در حالت کلی یک زبان L روی یک الفبا Σ کلیه‌ی رشته‌های ممکن از اعضاء Σ را شامل نبوده، بلکه تعدادی از رشته‌ها را شامل می‌شود. بنابراین به قواعد مشخصی نیاز داریم که بوسیله آن بتوانیم رشته‌های موردنیاز را ایجاد و هم‌چنین تعلق یا عدم تعلق یک رشته به یک زبان را تشخیص دهیم. البته در زبان‌های برنامه‌سازی مانند پاسکال، C و نظایر آن، ایجاد قواعد معین و مشخص تا اندازه‌ای مشکل است. زیرا این قواعد باید به اندازه‌ی کافی جامع و بدون اشکال بوده و با ارائه یک برنامه, شامل یک رشته خیلی طولانی درست یا نادرست بودن برنامه را تشخیص دهد. یکی از روشهای کلی و مفید در تشریح زبان‌ها گرامرها هستند.

تعریف:گرامر عبارت است از مجموعه‌ای از قواعد متناهی که با دنبال کردن آن می‌توان جملات معتبر از یک زبان را ایجاد و تعلق یا عدم تعلق یک رشته موردنظر به آن زبان را تشخیص داد.

مشخصات یک گرامر

گرامر G که آن را با نماد $G=<V,T,S,P>$ نشان می‌دهیم دارای چهار مشخصه به شرح زیر است:

الف. یک مجموعه متناهی و غیرتهی مانند **T** از نمادها بنام الفبا. هرعنصر **T** را یک پایانه یا یک ترمینال (Terminal)گویند. پایانه‌ها را با حروف کوچک مانند a و b و c و یا بعضی نمادهای ویژه نشان می‌دهند.

ب.یک مجموعه‌ی متناهی مانند **V** از نمادها که هر عنصر آن را یک متغیر (Variable) یا یک غیرپایانه (Nonterminal) گویند. متغیرهارا با حروف بزرگ مانند A و B و C و S و نظایر آن نشان می‌دهند. V و T دو مجموعه مجزا هستند $(V \cap T = \varnothing)$.

پ. عضو مشخصی مانند S متعلق به V که آنرا نماد شروع (Start symbol) گویند.

ت. مجموعه‌ی متناهی مانند **P** از تولیدات(Productions) بنام قوانین تولید یا قوانین جای‌گزینی. مجموعه**P** از عباراتی به صورت $A \to X_1 X_2 \dots X_n$ تشکیل شده است که در آن A یک غیر پایانه و $X_1 X_2 \dots X_n$ یک رشته متناهی از پایانه‌ها و یا غیر پایانه‌ها است$(X_i \in T \cup V)$. رشته‌ی $X_1 X_2 \dots X_n$ می‌تواند یک رشته تهی هم باشد که در این‌صورت $A \to \lambda$ و آنرا یک محصول تهی گویند. گرامر تعریف شده به صورت بالا را یک گرامر مستقل از متن (**C**ontext **F**ree **G**rammar) گویند که در بخش مربوط به انواع گرمرها به تفصیل توضیح داده شده است.

مثال۲۲. گرامر $G = \langle V,T,S,P \rangle$با مشخصات {a,b,c}=T، {S,A,B,C}=V، نماد شروعS و مجموعه قوانین تولیدP به صورت زیر تعریف شده است.

$$P = \{S \overset{1}{\to} AaB, S \overset{2}{\to} B, A \overset{3}{\to} aB, A \overset{4}{\to} a, B \overset{5}{\to} bC, C \overset{6}{\to} ac, C \overset{7}{\to} \lambda\}$$

چگونگی ایجاد یک زبان توسط یک گرامر

تعریف :گرامر $G=\langle V,T,S,P\rangle$ مفروض است. هر رشته‌ای از نمادها مانند $X_1X_2\dots X_n$ که در آن هر X_i یک پایانه یا یک غیر پایانه می‌باشد را یک فرم جمله‌ای (Sentential Form) از گرامر Gگویند. یک فرم جمله‌ای ممکن است رشته تهی λ هم باشد.

تعریف:گرامر $G=\langle V,T,S,P\rangle$ و فرم جمله‌ای $\sigma=\alpha A\beta$ که در آن α و β نیز فرم‌های جمله-ای، A یک غیر پایانه ، و $A\to x_1x_2x_3\dots x_n$ یک تولید درP باشد داده شده است. اگر τ یک فرم جمله‌ای نتیجه‌گیری شده از σ با جایگزینیA با $x_1x_2x_3\dots x_n$ باشد, یعنی $\tau=\alpha x_1x_2x_3\dots x_n\beta$, آنگاه گفته می‌شود که فرم جمله‌ای τ قابل نتیجه‌گیری بلادرنگ از σ با استفاده از تولید $A\to x_1x_2,x_3\dots x_n$ است و آن را با نماد $\sigma\Rightarrow\tau$ نشان می‌دهند.

دنباله w_n یعنی $w_1\Rightarrow w_2\Rightarrow w_3\Rightarrow\dots\Rightarrow w_n$ قابل اشتقاق از w_1 بوده و آنرا با نماد $w_1\overset{*}{\Rightarrow}w_n$ نشان می‌دهند. به عنوان مثال در گرامر $G=\langle V,T,S,P\rangle$، T={a,b }، V={S,B}، نماد شروع S و مجموعه تولیدات $P=\{S\to AS,S\to B,A\to a,B\to bB,B\to\lambda\}$ اشتقاق رشته abb به صورت $S\Rightarrow AS\Rightarrow aB\Rightarrow abB\Rightarrow abbB\Rightarrow abb$ است. در این عبارت هریک از رشته‌های $abb,abbB,abB,aB,AS,S$ یک فرم جمله‌ای است و $S\overset{*}{\Rightarrow}abb$

تعریف: گرامر $G=<V,T,S,P>$ مفروض است. زبانL(G) ایجاد شده توسط گرامر G مجموعه‌ای است از رشته‌ها بصورت $L(G)\{w\mid w\in T^*,S\overset{*}{\Rightarrow}w\}$. اعضاءL(G) را جمله‌های زبان گویند

مثال۲۳. با استفاده از گرامر تعریف شده در مثال ۲۲ ثابت کنید رشته $aabac$ متعلق به L(G) است.

جواب. $S\overset{1}{\Rightarrow}AaB\overset{4}{\Rightarrow}aaB\overset{5}{\Rightarrow}aabC\overset{6}{\Rightarrow}aabac$

مثال۲۴. با فرض $T=\{a,+,*,(,)\}$ مجموعه نمادها , $V=\{E,T,F\}$ مجموعه‌ی متغیرها، E نماد شروع وP مجموعه‌ی تولیدات به شرح زیر گرامری را نشان می‌دهد که با استفاده از آن می‌توان کلیه عبارات جبری درست را تولیدکرد.

1. $E\to E+T$, 2. $E\to T$, 3. $T\to T*F$, 4. $T\to F$, 5. $F\to(E)$, 6. $F\to a$

در این مثال اثبات می‌کنیم که عبارت جبری $(a+a)*a$ متعلق به L(G) است.

$$E\overset{2}{\Rightarrow}T\overset{3}{\Rightarrow}T*F\overset{4}{\Rightarrow}F*F\overset{5}{\Rightarrow}(E)*F\overset{1}{\Rightarrow}(E+T)*F\overset{2}{\Rightarrow}(T+T)*F$$
$$\overset{4}{\Rightarrow}(F+T)*F\overset{6}{\Rightarrow}(a+T)*F\overset{4}{\Rightarrow}(a+F)*F\overset{6}{\Rightarrow}(a+a)*F\overset{6}{\Rightarrow}(a+a)*a$$

قرارداد: تولیدات $A \to \alpha_1, A \to \alpha_2, ..., A \to \alpha_k$ را که نماد A در سمت چپ همه آنها بکار رفته است می‌توان به صورت $A \to \alpha_1 | \alpha_2 | .. | \alpha_{k-1} | \alpha_k$ نیز نشان می‌دهند.

به عنوان مثال گرامر مثالهای۲۲ و ۲۴ را می‌توان به ترتیب به صورت‌های زیر نشان داد

$$P = \{S \to AaB | B, A \to a | aB, B \to bC, C \to aC | \lambda\}$$

$$P = \{E \to E + T | T, T \to T * F | F, F \to (E) | a$$

انواع گرامرها (types of grammars)

گرامرها را با توجه به نوع تولیدات مجاز در آن‌ها به انواع مختلف به شرح زیر تقسیم‌بندی می‌کنند:

الف.گرامر نوع 0 (type - 0)

گرامری است که هیچ محدودیتی بر روی تولیدات آن منظور نشده است. در این گرامر یک تولید به صورت کلی $u \to v$ بیان می‌شود که u و v رشته‌های اختیاری از ترمینال‌ها و غیرترمینال‌ها بوده و $u \neq \lambda$. در این نوع گرامر جایگزینی رشته‌ی u بوسیله‌ی رشته‌ی v کاملاً اختیاری است.

مثال۲۵. گرامرG با الفبا {a,c,d}=T، مجموعه متغیرهای{S,X,Y}=V،نماد شروعS و مجموعه تولیدات P به شرح زیر یک گرامر از نوع 0 است.

$P = \{S \overset{1}{\to} aXYc, aX \overset{2}{\to} cad, Xc \overset{3}{\to} aXa, XYc \overset{4}{\to} Xcc, Yc \overset{5}{\to} \lambda\}$ اشتقاق رشته $acadac$

در این گرمر به صورت $S \Rightarrow a\underbrace{XYc}_{4} \Rightarrow a\underbrace{Xcc}_{3} \Rightarrow a\underbrace{aX}_{2}ac \Rightarrow acadac$ است:

ب. گرامر نوع 1 (type-1)

گرامر نوع 1 که آن‌را گرامر حساس نسبت به متن(context-sensitive) نیز گویند گرامری است شامل تولیدات به صورت کلی $uAv \to uwv$. در این تولید A یک غیرترمینال وw یک رشته غیر تهی $(w \neq \lambda)$ از ترمینال‌ها یا غیر ترمینال‌ها وu و vرشته‌هایی از ترمینال‌ها یا غیرترمینال‌ها می‌باشند. نکته‌ای که در این گرامر وجود دارد این است که در هر تولید $w_1 \to w_2$, $length(w_1) \leq leng(w_2)$. به عبارت دیگر $|w_1| \leq |w_2|$. باید توجه داشت که گرامر نوع ۱ گرامر نوع صفر هم هست ولی برعکس آن درست نیست

مثال۲۶. گرامر نوع ۱ $G = \langle V, T, S, P \rangle$ با {a,b,c}=T، {S,A,B,C}=V، نماد شروع S و مجموعه قوانین تولید به صورت زیر چه زبانی را تولید می‌کند. و اشتقاق رشته aabbcc در گرامر را نشان دهید.

$$1.S \to aSBC | aBC, \quad 2.CB \to BC, \quad 3.aB \to ab, \quad 4.bB \to bb,$$
$$5.bC \to bc, \quad 6.cC \to cc$$

جواب. $L(G) = \{a^n b^n c^n | n \geq 1\}$ زبان تولید شده و اشتقاق رشته aabbcc به صورت زیر است.

$$S \overset{1}{\Rightarrow} aSBC \overset{1}{\Rightarrow} aaBCBC \overset{3}{\Rightarrow} aabCBC \overset{2}{\Rightarrow} aabBCC \overset{4}{\Rightarrow} aabbCC$$
$$\overset{5}{\Rightarrow} aabbcC \overset{6}{\Rightarrow} aabbcc$$

گرامرنوع۲ (types- 2)

گرامر نوع ۲ که آنرا گرامر مستقل از متن (**C**ontext-**F**ree-**G**rammar) یا به اختصار CFG نیز گویند گرامری است شامل تولیدات به صورت کلی $A \to W$. در این تولیدA یک متغیر وW عبارت است از هر رشته‌ای از ترمینال‌ها یا غیرترمینال‌ها. رشته تهی λ نیز ممکن است در سمت راست یک تولید ظاهر شده باشد. *گرامرهای متن آزاد به راحتی قابل تجزیه بوده و عملاً کلیه زبان‌های کامپیوتری مانند پاسکال و C و ... از نوع متن آزاد هستند*

توجه: در یک گرامر متن آزاد استنتاج یک رشته ممکن است به طرق متفاوتی انجام گیرد.

مثال۲۷. گرامر $G = \langle V, T, S, P \rangle$ با مشخصات $V = \{S, A\}$, $T = \{a, b\}$ و قوانین تولید P به شرح زیر داده شده است. $P = \{S \to AA, A \to AAA | bA | Ab | a\}$. استنتاج رشته ababaa را به صورت‌های مختلف بدست آورید.

جواب.

$1. S \Rightarrow \underline{A}A \Rightarrow a\underline{A} \Rightarrow a\underline{A}AA \Rightarrow ab\underline{A}AA \Rightarrow aba\underline{A}A \Rightarrow abab\underline{A}A \Rightarrow ababa\underline{A} \Rightarrow ababaa$

$2. S \Rightarrow A\underline{A} \Rightarrow \underline{A}a \Rightarrow AA\underline{A}a \Rightarrow AAb\underline{A}a \Rightarrow A\underline{A}baa \Rightarrow Ab\underline{A}baa \Rightarrow \underline{A}babaa \Rightarrow ababaa$

$3. S \Rightarrow \underline{A}A \Rightarrow a\underline{A} \Rightarrow aAA\underline{A} \Rightarrow aAAa \Rightarrow abAAa \Rightarrow ab\underline{A}bAa \Rightarrow abab\underline{A}a \Rightarrow ababaa$

جایگزینی در بند 1 از سمت چپ, در بند 2 از راست و در بند 3 از هر دو طرف رشته انجام گرفته است.

گرامر نوع ۳ (types- 3)

گرامر متن آزاد $G = <V, T, S, P>$ را گرامر نوع ۳ یا گرامر منظم **(Regular Grammar)** یا گرامر خطی **(Linear Grammar)** گویند اگر هر تولید در آن به صورت یکی از دو حالت زیر باشد:

الف. $A \to WB | W$ که در آن $A, B \in V$ و $W \in T^*$

ب. $A \to BW | W$ که در آن $A, B \in V$ و $W \in T^*$

در حالت الف که متغیرB در سمت راست رشته W قرار گرفته است, گرامر را راست خطی(right linear) ودر حالت ب که B در سمت چپ رشته W قرار گرفته است, گرامر را چپ خطی(left linear) گویند.

مثال۲۸. گرامر $G = \langle V, T, S, P \rangle$ با مشخصات $V = \{S, A\}, T = \{x, y, z\}$ و قوانین تولید به شرح زیر یک گرامر راست خطی است: $P = \{S \to xS,\ S \to yA,\ A \to yA,\ A \to z\}$

و با مشخصات $T = \{a, b\}$, $V = \{S, A, B\}$ و قوانین تولید به شرح زیر یک گرامر چپ خطی است:

$$P = \{S \to Aab,\ A \to Aab | B\ , A \to a\}$$

بطور پیش‌فرض هر گرامر منظم یک گرامر راست خطی است, و چون $W \in T^*$ پس W یک ترمینال تکی هم می‌تواند باشد و در این صورت $|W|=1$. اگر $|W|>1$ آنگاه تولیدات شامل W را می‌توان به تعداد بیشتری از قوانین تولید تبدیل کرد به قسمی‌که هر یک از قوانین تولید فقط شامل یک ترمینال و یا شامل یک ترمینال و یک غیر ترمینال در سمت راست آن باشد.

مثال۲۹. گرامر راست‌خطی G با مجموعه تولیدات به صورت زیر داده شده است:

$$P=\{S \to aaB \mid aa,\ \ B \to bB \mid bb\}$$

گرامر داده شده را به گرامر راست‌خطی معادل تبدیل کنید به قسمی که هر یک از قوانین تولید فقط شامل یک ترمینال و یا شامل یک ترمینال و یک غیر ترمینال در سمت راست آن باشد.

جواب. $S \to aaB \Rightarrow \begin{cases} S \to aC \\ C \to aB \end{cases},\ \ S \to aa \Rightarrow \begin{cases} S \to aD \\ D \to a \end{cases},\ \ B \to bb \Rightarrow \begin{cases} B \to bE \\ E \to b \end{cases}$

بنابراین گرامر داده شده را می‌توانیم به صورت زیر بازنویسی کنیم.

$$P=\{S \to aC, C \to aB, S \to aD, D \to a, B \to bB, B \to bE, E \to b\}$$

مثال۳۰. گرامر منظم با قوانین تولید به شرح زیر چه زبانی را تولید می‌کند.

$$S \to aA \mid bS \mid \lambda \ \ , \ \ A \to aS \mid bA$$

جواب. گرامر داده شده رشته‌هائی را بر روی $\{a,b\}$ تولید می‌کند که تعداد aها در آن زوج باشد.

مثال۳۱. زبان ارائه شده توسط عبارت منظم a^*b^+ به صورت $\{a^m b^n \in \{a,b\}^* \mid m \geq 0, n \geq 1\}$ است.

گرامر منظم تولید کننده زبان مورد نظر را نشان دهید.

جواب. $S \to aS \mid B \ \ , \ \ B \to bB \mid b$

مثال۳۲. گرامری با قوانین تولید زیر یک گرامر منظم است. زبان تولید شده توسط آن را مشخص کنید.

$$S \to bS \mid aA$$
$$A \to bA \mid aB$$
$$B \to bB \mid aS \mid \lambda$$

جواب. زبان تولید شده توط گرامر فوق: $\{x \in \{a,b\}^* \mid |x|_a \equiv 2 \mod(3)\}$

توجه:گرامرهای نوع0 و1 و2 و 3 را ارجهیت چامسکی (Chomsky hierarchy) گویند و از انواع گرمرهای یاد شده ، گرمر نوع 2 و 3 از اهمیت بیشتری برخوردار هستند

قضیه. هر زبان منظم توسط یک گرامر متن آزاد (CFG) تولید می‌شود.

مثال۳۳. گرامر متن آزاد مربوط به عبارت منظم $11^*(00+11)$ را تشکیل دهید.

جواب. $S \to AB, B \to 11, A \to AC \mid \lambda, C \to 00 \mid 11$

اشتقاق و نشان دادن زبان‌ها توسط گرامر آن‌ها

از نقطه نظر علوم کامپیوتر دو پرسش اساسی به شرح زیر که وارون یکدیگرند حائز اهمیت است:

پرسش اول. رشته‌ی α از ترمینال‌ها داده شده است. آیا این رشته متعلق به زبان موردنظر است ؟

پرسش دوم. استنتاج رشته‌ی α متعلق به یک زبان با توجه به گرامر موردنظر چگونه است؟

اهمیت پرسش اول واضح است. برنامه‌ای به یکی از زبان‌های برنامه‌سازی تهیه شده است، می‌خواهیم درستی برنامه را ارزیابی کنیم. منظور از درستی برنامه‌ی تهیه شده در این قسمت معتبر بودن آن از نقطه نظر دستور زبان مربوطه است، نه چگونگی عملکرد و خروجی برنامه. به عبارت دیگر هدف پی بردن به این است که آیا برنامه‌ی نوشته شده از نظر گرامری درست است؟ آیا دستورالعملها به صورت صحیح نوشته شده است؟ *عمل ارزیابی این نوع پرسش‌ها را تحلیل لغوی (lexical Analysis) گویند.*

در پرسش دوم که در مقایسه با پرسش اول مهم‌تر است تفسیر معنی و مفهوم برنامه و چگونگی اجرای آن موردنظر است. *مطالعه و بررسی پرسش دوم را عمل تجزیه (parsing) یا تحلیل نحو (syntax analysi) گویند*

مثال۳۴. گرامر $G=<V,T,S,P>$ مربوط به عبارات محاسباتی معتبر متشکل از اعداد صحیح بدون علامت به صورت زیر است:

$V=\{E,T,F\}$ ، $T=\{a,+,*,(,)\}$ ، E نماد شروع و P مجموعه تولیدات به صورت زیر داده شده است

1. $E \to E+T,$ 2. $E \to T,$ 3. $T \to T*F,$ 4. $T \to F,$ 5. $F \to (E),$ 6. $F \to a$

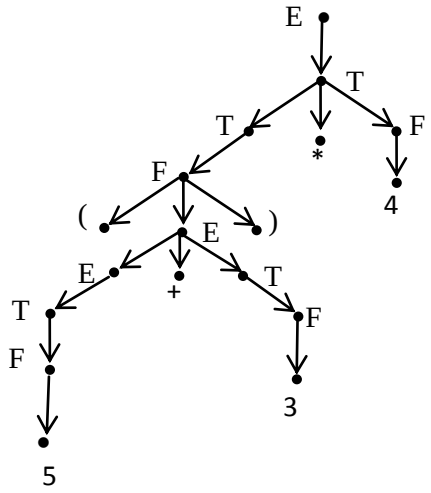
در این مثال نماد a جهت نشان دادن هر عدد صحیح مثبت و فاقد علامت بکار رفته است. با در دست داشتن یک جمله در زبان L(G) مثلاً 4*(5+3) می‌خواهیم بدانیم:

الف: آیا این عبارت معتبر است؟

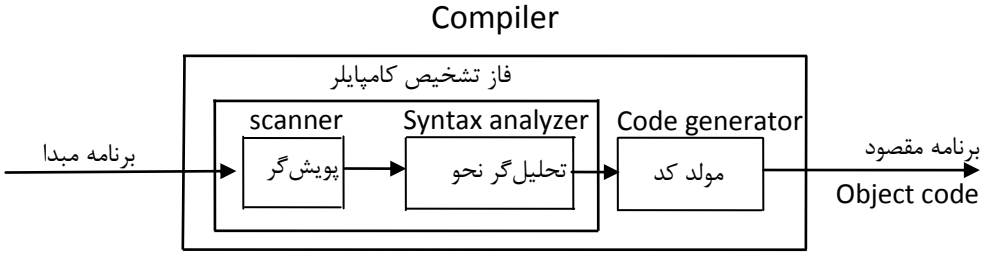
ب: مفهوم این عبارت چیست؟

جواب. هدف از سئوال فوق در این مثال، تفسیر یا ارزیابی رشته‌ی 4*(5+3) است. وقتی این رشته به ماشین تغذیه گردید ماشین باید اعداد 3 و4 و5 را در ثبات(register) یا محل‌های مناسب حافظه ذخیره ، عمل جمع انجام ، حاصل‌جمع در محل مناسب حافظه نگهداری، و در نهایت عمل ضرب انجام گیرد. کلیه‌ی این اعمال باید کاملاً بصورت خودکار انجام شود. بنابراین یک *ماشین* باید به نحوی برنامه‌ریزی گردد که وقتی رشته‌ی4*(5+3) به آن داده شد، کلیه‌ی موارد خواسته شده به ترتیب درست اجرا گردد. اکنون می‌خواهیم نشان دهیم که اگر ماشین بتواند درخت اشتقاق(derivation tree) جمله‌ی موردنظر را تولید کند می‌تواند چگونگی انجام کلیه موارد گفته شده را تشخیص و مفهوم یا ارزش جمله را استنتاج کند. اکنون فرض کنید که درخت اشتقاق عبارت4*(5+3) به شکل زیر داده شده است.

درخت اشتقاق عبارت4*(3+5)



با الحاق برگ‌های درخت از بالا به پائین و از چپ به راست و تطبیق نماد به نماد رشته بدست آمده با رشته اصلی، تشخیص تعلق رشته موردنظر به زبان مربوطه به سادگی امکان‌پذیر است. جهت آگاهی از چگونگی ذخیره چنین درختی در حافظه و نحوه دسترسی به گره‌های درخت، به توضیح خیلی کلی چگونگی ترجمه یک زبان می‌پردازیم. یک هم‌گردان(compiler) در یک زبان وظائف مختلفی را از قبیل تشخیص تعلق یک جمله به زبان موردنظر ، ساختن درخت نحو(syntax tree) برای جمله‌ی داده شده ودر صورت معتبر بودن آن از نقطه نظر نحوی و معنایی، ایجاد دستورالعملهای مقصود (object code) برای جمله‌ی داده شده به عهده دارد .این پردازش در شکل زیر ارائه شده است.



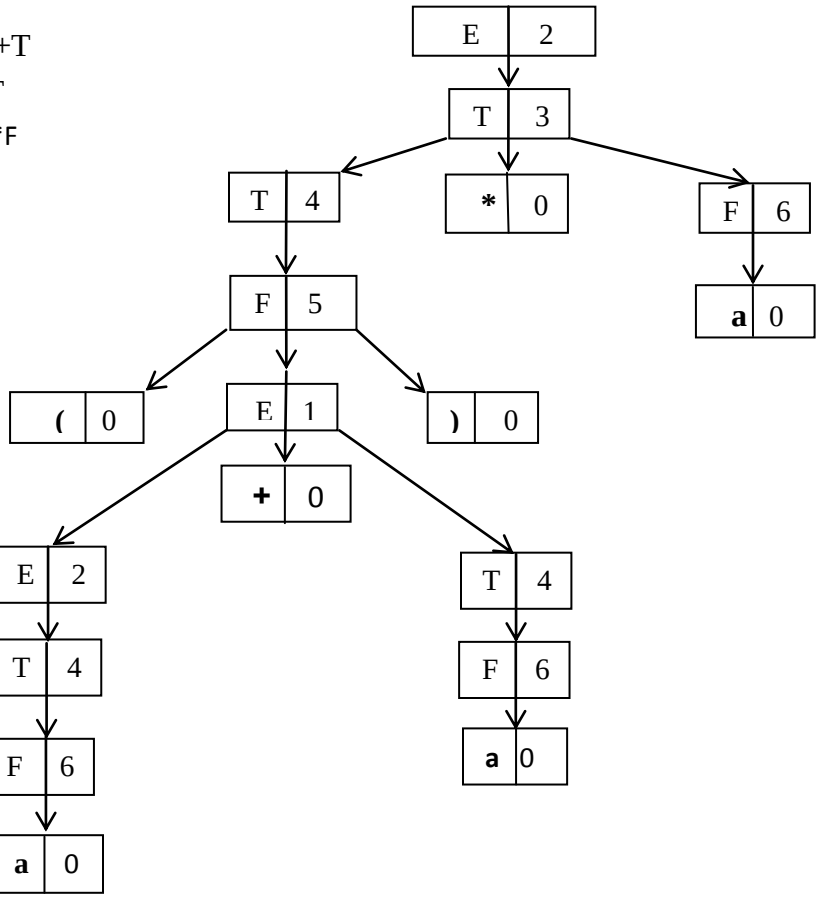
برنامه مبداء یک ورودی برای پویش‌گر است که هدف آن تفکیک متن دریافت شده به قطعاتی مانند ثابتها ، اسامی متغیرها، کلمات کلیدی (مانند forوif ... و عمل‌گرها) می‌باشد. معمولاً این اعمال با ساختن جداولی توسط پویشگر انجام می‌گیرد. پویش‌گر تحلیل‌گر نحو را که وظیفه آن ساختن درخت نحو برای برنامه مبدأ می‌باشد تغذیه می‌کند. این عمل در مقایسه با کار پویش‌گر بسیار پیچیده است. خروجی تحلیلگر نحو , به قسمت مولد کد تغذیه می‌گردد.

مثال۳۵. درخت اشتقاق عبارت جبری $a*(a+a)$ را با استفاده از گرمر تعریف شده رسم کنید.

توجه. برای رسم درخت اشتقاق از روش جدیدی برخلاف روشهای معمول استفاده شده است.

جواب. هر گرهی از درخت بصورت | X | n | نشان داده می‌شود. X یک ترمینال یا غیر ترمینال و n شماره تولیدی هست که جهت دسترسی به فرزندان گره مورد استفاده قرار می‌گیرد. n=0 یعنی برگ درخت. با الحاق برگ‌های درخت از بالا به پائین و چپ به راست و تطبیق نماد به نماد رشته بدست آمده با رشته اصلی، تشخیص تعلق رشته موردنظر به زبان مربوطه به سادگی امکان‌پذیر است. در شکل زیردرخت اشتقاق عبارت جبری $a*(a+a)$ ارائه شده است.

1. E→E+T
2. E→ T
3. T→T*F
4. T→ F
5. F→(E)
6. F→a

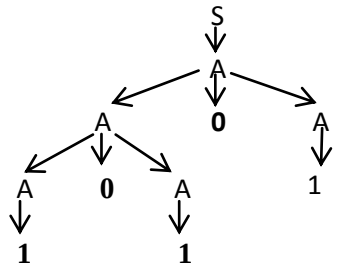


مثال۳۶. با استفاده ازگرامر $P=\{S\xrightarrow{1}A, A\xrightarrow{2}A0A, A\xrightarrow{3}1\}$ درخت اشتقاق رشته $\sigma=10101$ را نشان دهید.

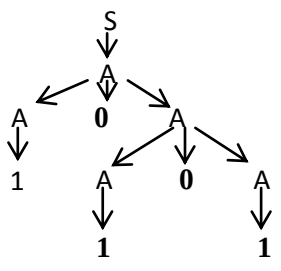
جواب. رشته داده شده دارای دو اشتقاق و در نتیجه دو درخت اشتقاق متمایز به صورت زیر است.

الف. $S\xRightarrow{1}\underline{A}\xRightarrow{2}\underline{A}0A\xRightarrow{2}\underline{A}0A0A\xRightarrow{3}10\underline{A}0A\xRightarrow{3}1010\underline{A}\xRightarrow{3}10101$ (در هر فرم جمله‌ای متغیر زیر خطدار جایگزین شده)

ب. $S\xRightarrow{1}\underline{A}\xRightarrow{2}A0\underline{A}\xRightarrow{2}A0A0\underline{A}\xRightarrow{3}A0\underline{A}01\xRightarrow{3}\underline{A}0101\xRightarrow{3}10101$



درخت اشتقاق الف



درخت اشتقاق ب

تعریف. فرض کنید G یک گرامر و σ یک جمله در زبان $L(G)$ است. فرض کنید اشتقاق σ با شروع از S به صورت $S \Rightarrow \sigma_1 \Rightarrow \sigma_2 \Rightarrow \cdots \Rightarrow \sigma_n = \sigma$ است. گفته می‌شود که اشتقاق مورد نظر

الف. سمت چپ‌ترین اشتقاق (leftmost derivation) σ است اگردر هر مرحله‌ی جایگزینی $\sigma_i \Rightarrow \sigma_{i+1}$ غیر ترمینال جایگزین شده σ_i سمت چپ‌ترین غیر ترمینال باشد.

ب. سمت راست‌ترین اشتقاق (rightmost derivation) σ است اگر در هر مرحله‌ی جایگزینی $\sigma_i \Rightarrow \sigma_{i+1}$ غیر ترمینال جایگزین شده σ_i سمت راست‌ترین غیر ترمینال باشد.

مثال۳۷. گرامر زیر جهت تشکیل عبارت محاسباتی معتبر متشکل از اعداد صحیح بدون علامت داده شده است. سمت چپ‌ترین $(\underset{lm}{\Rightarrow})$ و سپس سمت راست‌ترین $(\underset{rm}{\Rightarrow})$ اشتقاق عبارت $(a+a)*a$ را نشان دهید.

1. $E \to E+T$
2. $E \to T$
3. $T \to T*F$
4. $T \to F$
5. $F \to (E)$
6. $F \to a$

جواب.

$$E \underset{lm}{\overset{2}{\Rightarrow}} T \overset{3}{\Rightarrow} T*F \overset{4}{\Rightarrow} F*F \overset{5}{\Rightarrow} (E)*F \overset{1}{\Rightarrow} (E+T)*F \overset{2}{\Rightarrow} (T+T)*F$$
$$\overset{4}{\Rightarrow} (F+T)*F \overset{6}{\Rightarrow} (a+T)*F \overset{4}{\Rightarrow} (a+F)*F \overset{6}{\Rightarrow} (a+a)*F \overset{6}{\Rightarrow} (a+a)*a$$

بنابراین $E \underset{lm}{\overset{*}{\Rightarrow}} (a+a)*a$

$$E \underset{rm}{\overset{2}{\Rightarrow}} T \overset{3}{\Rightarrow} T*F \overset{6}{\Rightarrow} T*a \overset{4}{\Rightarrow} F*a \overset{5}{\Rightarrow} (E)*a \overset{1}{\Rightarrow} (E+T)*a \overset{4}{\Rightarrow} (E+F)*a$$
$$\overset{4}{\Rightarrow} (E+a)*a \overset{6}{\Rightarrow} (T+a)*a \overset{4}{\Rightarrow} (F+a)*a \overset{1}{\Rightarrow} (a+a)*a$$

بنابراین $E \underset{rm}{\overset{*}{\Rightarrow}} (a+a)*a$

۳۸. گرامر متن آزاد $G=<V,T,S,P>$ با مشخصات $V=\{E,I\}$, $T=\{+,*,a,b,0,1\}$ و قوانین تولید به شرح زیر داده شده است: $P=\{E \to I \mid E+E \mid E*E \mid (E), I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1\}$ با استفاده از اشتقاق چپ و اشتقاق راست تعلق رشته $a*(a+b00)$ به زبان مورد نظر را اثبات کنید.

جواب.

$$E \underset{lm}{\Rightarrow} E*E \Rightarrow I*E \Rightarrow a*E \Rightarrow a*(E) \Rightarrow a*(E+E) \Rightarrow a*(I+E)$$
$$\Rightarrow a*(a+E) \Rightarrow a*(a+I) \Rightarrow a*(a+I0) \Rightarrow a*(a+I00) \Rightarrow a*(a+b00)$$

$$E \underset{rm}{\Rightarrow} E*E \Rightarrow E*(E) \Rightarrow E*(E+E) \Rightarrow E*(E+I) \Rightarrow E*(E+I0) \Rightarrow E*(E+I00)$$
$$\Rightarrow E*(E+b00) \Rightarrow E*(I+b00) \Rightarrow E*(a+b00) \Rightarrow I*(a+b00) \Rightarrow a*(a+b00)$$

مثال۳۹. با استفاده از گرامر تعریف شده به شرح زیر ابتدا سمت چپ‌ترین و سپس سمت راست‌ترین اشتقاق عبارت $abbbb$ را نشان دهید.

$$S \to aAB, A \to bBb, B \to A \mid \lambda$$

ابتدا سمت چپ‌ترین و سپس سمت راست‌ترین اشتقاق عبارت $abbbb$ را نشان دهید.

جواب.

$$S \underset{lm}{\Rightarrow} a\underline{A}B \Rightarrow ab\underline{B}bB \Rightarrow ab\underline{A}bB \Rightarrow abb\underline{B}bbB \Rightarrow abbbb\underline{B} \Rightarrow abbbb$$

$$S \underset{rm}{\Rightarrow} aA\underline{B} \Rightarrow a\underline{A} \Rightarrow ab\underline{B}b \Rightarrow ab\underline{A}b \Rightarrow abb\underline{B}bb \Rightarrow abbbb$$

گرامر مبهم(ambiguous grammar)

فرض ضمنی در رابطه با کاربرد یک گرامر متن-آزاد وجود ساختار منحصر بفرد برای هر یک از رشته‌های متعلق به زبان آن گرامر است. ولی ممکن است گرامری وجود داشته باشد که فاقد چنین خاصیت باشد. به عنوان مثال در گرامر G با تولیدات $S \to a \mid aAb \mid abSb, \quad A \to aAAb \mid bS$ اشتقاق رشته $w = abab$ به دو صورت متفاوت زیر امکان پذیر است. و گفته می‌شود که گرامر G یک گرامر **مبهم** است

(۱). $S \Rightarrow abSb \Rightarrow abab$ (۲). $S \Rightarrow aAb \Rightarrow abSb \Rightarrow abab$

تعریف. گرامر متن-آزاد G را گرامر مبهم گویند اگر برای رشته‌ای مانند $w \in L(G)$ حداقل دو درخت اشتقاق- چپ متمایز وجود داشته باشد. همین وضعیت در مورد درخت اشتقاق- راست نیز صادق است.

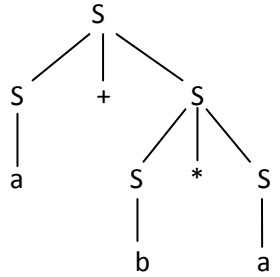
۴۰. قوانین تولید یک گرامر متن آزاد در زیر ارائه شده است. اشتقاق چپ عبارت $a+b*a$ و درخت اشتقاق آن را نشان دهید. $S \to S+S \mid S*S \mid (S) \mid a \mid b$

جواب. برای عبارت داده شده دو اشتقاق چپ و در نتیجه دو درخت اشتقاق چپ متمایز وجود دارد.

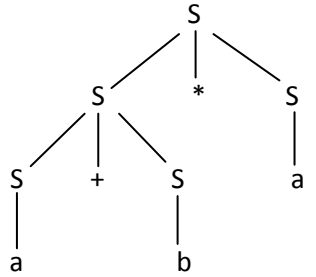
الف. $S \underset{lm}{\Rightarrow} S+S \underset{lm}{\Rightarrow} a+S \underset{lm}{\Rightarrow} a+S*S \underset{lm}{\Rightarrow} a+b*S \underset{lm}{\Rightarrow} a+b*a$

ب. $S \underset{lm}{\Rightarrow} S*S \underset{lm}{\Rightarrow} S+S*S \underset{lm}{\Rightarrow} a+S*S \underset{lm}{\Rightarrow} a+b*S \underset{lm}{\Rightarrow} a+b*a$

درخت اشتقاق چپ الف



درخت اشتقاق چپ ب

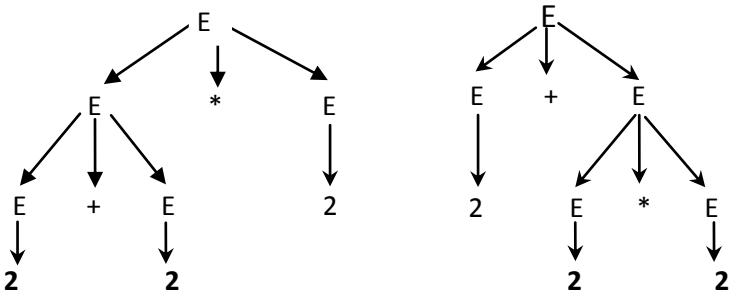


توجه. مواردی وجود دارد که دو گرامر متمایز G و G' زبان یکسانی را تولید کنند . یعنی $L(G) = L(G')$.
به عنوان مثال گرامرهای $G: S \to 0A0, A \to S \mid \lambda$ و $G': S \to 0B, B \to 0S \mid 0$ زبان یک‌سان یعنی مجموعه کلیه رشته‌ها که تعداد 0ها زوج است را تولید می‌کنند. گرامر G' یک گرامر منظم و بدون تولید تهی است. در صورتیکه گرامر G فاقد چنین خواص است.

چرا مبهم بودن گرامر بایستی مورد توجه قرار گیرد؟

جهت پاسخ به این سوال به دو درخت اشتقاق مربوط به عبارت محاسباتی **a+a*a** در صورتیکه **a = 2** باشد توجه کنید.

شکل ۱ شکل۲



شکل ۱ درخت اشتقاق عبارت $a+(a*a)=6$ و لی شکل ۲ درخت اشتقاق $(a+a)*a=8$ است. بنابراین عدم تعیین ارجهیت بین عملگرها مثلا * و + , هم‌چنین عدم وجود دسته‌بندی در دنباله‌ای از عملگرها مثلا مشخص نبودن اینکه آیا منظور از عبارت جبری $E+E+E$ عبارت $(E+E)+E$ است یا عبارت $E+(E+E)$ موجب بروز ابهام می‌گردد. باید توجه داشت که الگوریتم معین و مشخصی جهت رفع ابهام (removing ambiguity) وجود ندارد ولی با اعمال تمهیداتی می‌توان یک گرامر مبهم را به غیر مبهم تبدیل کرد.

مثال ۴۱. در زیر دو مورد از گرمرهای متن آزاد (CFG) , یک یا چند اشتقاق و زبان تولید شده توسط آن نشان داده شده است:

۱. $G: \quad S \to aSb \mid \lambda \quad \Rightarrow \quad L(G) = \{a^n b^n \mid n \geq 0\}$

اشتقاق رشته aabb : $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$

۲. $G: \quad S \to aSa \mid bSb \mid \lambda \quad \Rightarrow \quad L(G) = \{WW^R \mid W \in \{a,b\}^*\}$

اشتقاق رشته‌های abba و abaaba به شرح زیر است.

(1). $S \Rightarrow aSa \Rightarrow abSba \Rightarrow abba$

(2). $S \Rightarrow aSa \Rightarrow abSba \Rightarrow abaSaba \Rightarrow abaaba$

مثال ۴۲. گرمر متن آزاد G با قوانین تولید $S \to aSa \mid aBa,\quad B \to bB \mid b$ چه زبانی را تولید می‌کند.

جواب. $L(G) = \{a^n b^m a^n \mid n > 0, m > 0\}$

مثال ۴۳. گرمر متن آزادی معرفی کنید که زبان $L(G) = \{a^n b^m c^m d^{2n} \mid n \geq 0, m > 0\}$ را تولید کند.

جواب. رابطه بین تعداد a های آغازی و d های انتهائی هم‌چنین تعداد b ها و c های رشته‌های زبان وجود یک قانون بازگشتی را ضروری می‌نماید. در ضمن از قانون تولید $A \to bc$ هم جهت پایان دادن به بازگشتی و هم اطمینان از وجود زیر رشته bc در هر رشته‌ای از زبان استفاده شده است.

$$S \to aSdd \mid A,\quad A \to bAc \mid bc$$

مثال ۴۴. گرمر متن آزادی معرفی کنید که زبان $L(G) = \{a^n b^m \mid 0 \leq n \leq m \leq 2n\}$ را تولید کند.

جواب. جهت تهیه گرامر مورد نظر , یک قانون بازگشتی برای ایجاد یک b انتهائی برای هر a آغازی, و قانون بازگشتی دیگر جهت ایجاد دوتا b برای هر a منظور می‌کنیم. بنابراین برای هر a حداقل یک b و حداکثر دو تا b تولید می‌شود.

$$S \to aSb \mid aSbb \mid \lambda$$

مثال۴۵. گرمر متن آزاد G با قوانین $S \to abScB \mid \lambda,\quad B \to bB \mid b$ چه زبانی را تولید می‌کند. درضمن اشتقاق رشته‌ی $ababcbcbb$ را نشان دهید.

جواب. قانون بازگشتی S تعداد ab ها و cB های یکسان را تولید می‌کند ولی قانون B در جایگزینی-های متفاوت ممکن است تعداد b های متفاوت, در بعضی موارد فقط یک b و موارد دیگر دوتا b یعنی bb را تولید خواهد کرد. بنابراین زبان تولید شده توسط گرامر داده شده بصورت زیر است.

$$L(G) = \{(ab)^n (cb^m)^n \mid n \geq 0, m > 0\}$$

اشتقاق رشته‌ی $ababcbcbb$:

$$S \Rightarrow abScB \Rightarrow ababScBcB \Rightarrow ababcBcB \Rightarrow ababcbcB \Rightarrow ababcbcbB \Rightarrow ababcbcbb$$

مثال ۴۶. گرمر متن آزادی معرفی کنید که بر روی الفباء $\Sigma = \{0,1\}$ زبان‌هائی تولید کند که هر رشته‌ای مانند W متعلق به آنها دارای ویژگیهای زیر باشد.

الف. نمادهای آغاری و پایانی آن یک نوع کاراکتر باشد.

ب. طول رشته‌ی W یعنی $|W|$ فرد باشد.

ج. طول رشته‌ی W یعنی $|W|$ فرد بوده و نماد وسطی آن 0 باشد.

جواب: **الف.** $S \to 0A0 \mid 1A1,\quad A \to 0A \mid 1A \mid \lambda$

ب. $S \to 0A \mid 1,\quad A \to 0S \mid 1S \mid \lambda$

ج. $S \to 0 \mid 0S0 \mid 0S1 \mid 1S0 \mid 1S1$

مثال ۴۷. گرمر متن آزادی معرفی کنید که بر روی الفباء $\Sigma = \{a,b,c\}$ زبان زیر را تولید کند.

$$L(G) = \{a^m b^n c^{m+n} \mid m,n \geq 0\} \quad S \to aSc \mid B, \quad B \to bBc \mid \lambda$$

جواب.

- ✓ از قانون $S \to aSc$ جهت تولید aها و cها به تعداد یکسان در منتهاالیه رشته‌های تولید شده استفاده می‌کنیم. با استفاده از قانون $S \to \lambda$, رشته تولید شده فاقد b و به صورت $a^m c^m$ خواهد بود.
- ✓ جهت تولید b بین a و c متغیر جدیدی مانند A را انتخاب کرده و قانون $S \to A$ را وسیله‌ای جهت تولید bها معرفی می‌کنیم.
- ✓ چون تعداد bها و cها باید برابر باشد از قانون $A \to bAc$ استفاده می‌کنیم.
- ✓ در نهایت از تولید تهی $A \to \lambda$ جهت پایان اشتقاق استفاده می‌کنیم.

بنابراین قوانین تولید برای گرامر متن-آزاد مورد نظر به صورت زیر خواهد بود:

$$S \to aSc, \quad S \to A, \quad S \to \lambda, \quad A \to bAc, \quad A \to \lambda$$

اشتقاق کلی رشته‌ها: $S \Rightarrow^* a^n Sc^n \Rightarrow a^n Ac^n \Rightarrow^* a^n b^m Ac^m c^n \Rightarrow a^n b^m c^{(n+m)}$

توجه: $S \Rightarrow^* a^n Sc^n$ یعنی بازگشتی $S \Rightarrow aSc$ به تعداد n بار و $a^n Ac^n \Rightarrow^* a^n b^m Ac^m c^n$ یعنی بازگشتی $A \to bAc$ به تعداد m بار بازخوانی شده است.

مثال ۴۸. چرا گرامر $S \to 0A \mid 1B, \quad A \to 0AA \mid 1S \mid 1, \quad B \to 1BB \mid 0S \mid 0$ مبهم است؟

جواب. چون رشته‌هائی با اشتقاق چند گانه وجود دارند بنابراین گرامر داده شده مبهم است. یک نمونه از رشته‌ها با اشتقاق چند گانه رشته 001101 است.

$$S \Rightarrow 0A \Rightarrow 00AA \Rightarrow 001S1 \Rightarrow 0011B1 \Rightarrow 001101$$

$$S \Rightarrow 0A \Rightarrow 00AA \Rightarrow 0011S \Rightarrow 00110A \Rightarrow 001101$$

مثال۴۹. گرمر متن آزاد با قوانین زیر چه زبانی را بر روی $\{a,b\}$ تولید می‌کند.

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

جواب. گرامر داده شده رشته‌هائی را بر روی $\{a,b\}$ تولید می‌کند که از هر دو طرف یک جور خوانده می‌شود(palindromes). در ضمن استفاده از قانون $S \to a \mid b$ موجب تولید رشته به طول فرد و استفاده از قانون $S \to \lambda$ موجب تولید رشته به طول زوج خواهد بود. به عنوان مثال اشتقاق رشته $abbabba$ به طول 7 به صورت زیر است:

$$S \Rightarrow aSa \Rightarrow abSba \Rightarrow abbSbba \Rightarrow abbabba$$

فرم‌های نرمال (Normal Forms)

جهت اثبات وجود یک خاصیت ویژه در یک گرامر اغلب مناسب است بدانیم که آیا کلیه تولیدات آن دارای یک فرم ویژه‌ای می‌باشند یا نه. در این قسمت دونوع از این فرم‌های مورد نظر را ارائه می‌دهیم.

الف. فرم نرمال چامسکی (**C**homsky **N**ormal **F**orm)

تعریف. یک گرامر متن_آزاد (CFG) بصورت ***فرم نرمال چامسکی*** (CNF) است اگر کلیه قوانین تولید در آن به صورت یکی از ۲ فرم زیر باشد:

1. $A \to a$
2. $A \to BC$

در این تعریف a یک ترمینال و A, B, C غیر ترمینال(متغیر) هستند. درضمن B و C متغیر آغازی نیستند.

در مثال زیر یک گرامر متن-آزاد به صورت "فرم نرمال چامسکی" تبدیل شده است.

$$\begin{vmatrix} S \to AaBb \\ A \to aB \\ B \to b \end{vmatrix} \overbrace{\Rightarrow}^{transform} \begin{vmatrix} S \to AX_1 \\ A \to A_1B \\ B \to b \\ A_1 \to a \\ B_1 \to b \\ X_1 \to A_1X_2 \\ X_2 \to BB_1 \end{vmatrix}$$

ب. فرم نرمال گریباخ (Greibach Normal Form)

تعریف. یک گرامر متن_آزاد بصورت ***فرم گریباخ*** است اگر سمت راست کلیه قوانین تولید در آن یک ترمینال و بدنبال آن به تعداد صفر یا بیشتر غیر ترمینال وجود داشته باشد:

$$A \to aX$$

یک نماد ترمینال — به تعداد صفر یا بیشتر نماد غیر ترمینال

دلایلی که نشان می‌دهد یک CFG به صورت CNF نیست.

- ✓ متغیر آغازی مثلا S در سمت راست بعضی از قوانین تولید ظاهر شده باشد.
- ✓ دارای قوانین تهی مانند $A_i \to \lambda$ باشد.
- ✓ دارای قوانین واحد(nuit rules) یعنی قوانینی که سمت راست فقط دارای یک متغیر مانند $A \to B$ باشد.
- ✓ سمت راست قوانینی فاقد فقط دو متغیر یا فاقد فقط یک ترمینال باشد.

روش تبدیل یک گرامر متن-آزاد به صورت *فرم نرمال چامسکی*

قدم اول. برای هر قانون تولید که سمت راست آن ترکیبی از ترمینال‌ها و غیر ترمینال‌ها باشد, هر ترمینال a_i را با متغیر A_i جاگزین و قانون تولید جدید $A_i \to a_i$ را به مجموعه قوانین تولید اضافه می‌کنیم.

مثال ۱.
$$Q \to aP \overset{step1}{\Rightarrow} \begin{vmatrix} Q \to A_1P \\ A_1 \to a \end{vmatrix}$$

مثال ۲.
$$\begin{vmatrix} S \to AB \\ A \to aCa \\ A \to a \\ B \to bB \\ B \to b \\ C \to D \\ D \to d \end{vmatrix} \overset{STEP1}{\Rightarrow} \begin{vmatrix} S \to AB \\ A \to A_1CA_1 \\ A \to a \\ B \to B_1B \\ B \to b \\ C \to D \\ D \to d \\ A_1 \to a \\ B_1 \to b \end{vmatrix}$$

در این مثال aCa با A_1CA_1 جایگزین و قانون جدید $A_1 \to a$ اضافه شده است. هم‌چنین bB با B_1B جایگزین و قانون جدید $B_1 \to b$ اضافه شده است.

قدم دوم. قوانین تولید بصورت کلی $A = C_1C_2 \cdots C_n (n \geq 3)$ (قوانینی که سمت راست آن دارای ۳ یا بیشتر از غیر ترمینال‌ها باشد) را با مجموعه قوانین زیر جایگزین می‌کنیم:*(معرفی متغیرهای واسطه‌ای)*

$$\begin{aligned} A &\to C_1X_1 \\ X_1 &\to C_2X_2 \\ &\vdots \\ X_{n-2} &\to C_{n-1}C_n \end{aligned}$$

قدم دوم را تا موقعی تکرار میکنیم که در هر قانون تولید بیشتر از ۲ غیر ترمینال وجود نداشته باشد.

مثال.
$$Q \to ABCDE \overset{STEP2}{\Rightarrow} \begin{vmatrix} Q \to AX_1 \\ X_1 \to BX_2 \\ X_2 \to CX_3 \\ X_3 \to DE \end{vmatrix}$$

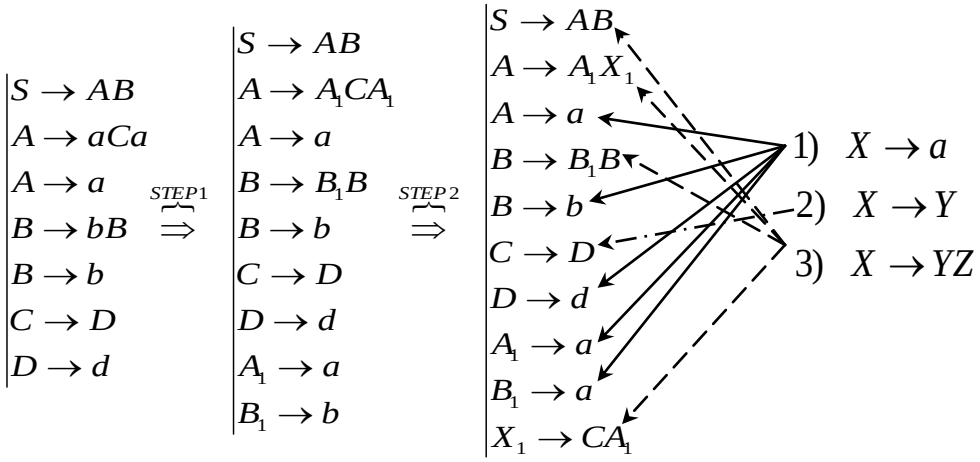
مثال. کاربرد قدم دوم به یک گرامر
$$\begin{vmatrix} S \to AB \\ A \to A_1CA_1 \\ A \to a \\ B \to B_1B \\ B \to b \\ C \to D \\ D \to d \\ A_1 \to a \\ B_1 \to b \end{vmatrix} \overset{STEP2}{\Rightarrow} \begin{vmatrix} S \to AB \\ A \to A_1X_1 \\ X_1 \to CA_1 \\ A \to a \\ B \to B_1B \\ B \to b \\ C \to D \\ D \to d \\ A_1 \to a \\ B_1 \to a \end{vmatrix}$$

در این مثال A_1CA_1 با A_1X_1 جایگزین و قانون جدید $X_1 \to CA_1$ اضافه شده

پس از بکار بردن قدم اول و دوم الگوریتم بر روی یک گرامر متن-آزاد , نتیجه نهائی یک گرامر شامل ۳ قانون به صورت کلی زیر خواهد بود:

$$1)\ X \to a \qquad 2)\ X \to Y \qquad 3)\ X \to YZ$$

نظیر قوانین سه گانه بالا در گرامر حاصل از اعمال قدم دوم الگوریتم در مثال بالا در شکل زیر با خطوط جهت دار مختلف نشان داده شده است.



از قوانین سه گانه $X \to YZ$, $X \to Y$ و $X \to a$ قوانین به صورت $X \to Y$ را **قوانین زنجیر** (chain rules) یا **قوانین واحد**(unite rules) گویند. فرم نرمال چامسکی فاقد قواعد زنجیر است .

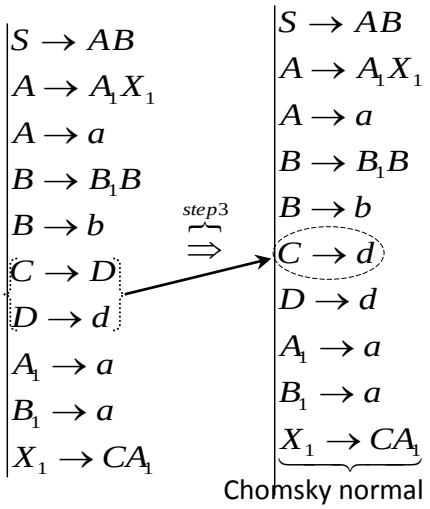
قدم سوم. از بین بردن قوانین زنجیر.

✓. اگر قانون $Y \to a$ وجود داشته باشد آنگاه $X \to Y$ را با $X \to a$ جایگزین کنید.

✓. اگر قانون $Y \to YZ$ وجود داشته باشد آنگاه $X \to Y$ را با $X \to YZ$ جایگزین کنید.

✓. اگر $X \to Z$ وجود داشته باشد آنگاه $X \to Y$ را با نتیجه جایگزینی Z جایگزین کنید.

مثال برای کاربرد قدم سوم.



مثال. $\{S \to A, A \to B, B \to b\} \overset{step3}{\Rightarrow} \{S \to b, A \to b, B \to b\}$

مثال۵۰. گرامر متن-آزاد زیر را به **فرم- نرمال– چامسکی** تبدیل کنید.

$$\begin{vmatrix} S \to ABa \\ A \to aab \\ B \to Ac \end{vmatrix} \overset{step1}{\Rightarrow} \begin{vmatrix} S \to ABA_1 \\ A \to A_1A_1B_1 \\ B \to AC_1 \\ A_1 \to a \\ B_1 \to b \\ C_1 \to c \end{vmatrix} \overset{step2}{\Rightarrow} \begin{vmatrix} S \to AV_1 \\ V_1 \to BA_1 \\ A \to A_1V_2 \\ V_2 \to A_1B_1 \\ B \to AC_1 \\ A_1 \to a \\ B_1 \to b \\ C_1 \to c \end{vmatrix}$$

بعد از انجام قدم‌های یک و دو چون نتیجه فاقد قوانین زنجیر است بنابراین نیازی به اعمال قدم سوم نیست و گرامر حاصل بعد از اعمال قدم دوم الگوریتم به صورت فرم نرمال چامسکی است.

توجه: هر گرامر متن-آزاد و فاقد محصول تهی را می‌توان به صورت "فرم نرمال چامسکی"تبدیل کرد. در ضمن در CNF متغیر آغازی مثلا S باید در سمت راست قوانین تولید نباید ظاهر شده باشد. بنابرین اگر یک گرامر متن – آزاد دارای محصول تهی مثلا $A \to \lambda$ یا $A \to \varepsilon$ بوده و یا متغیر آغازی در سمت راست قوانینی ظاهر شده باشد بایستی از روشهائی برای حذف تولید تهی و حذف متغیر آغازی از سمت راست قوانین استفاده کنیم. انجام این عمل با ذکر مثال به عهده دانشجو.

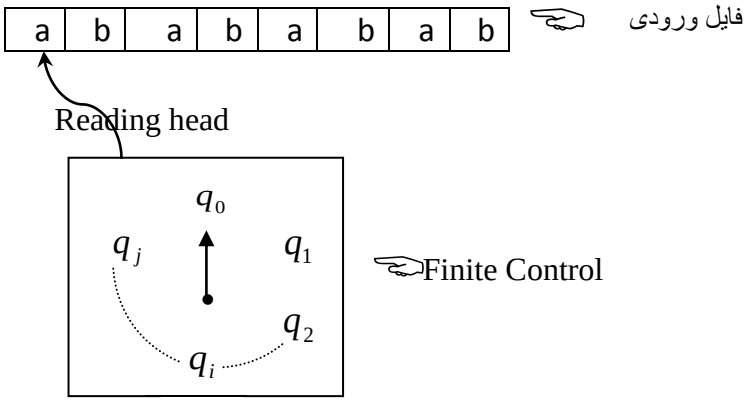
توجه: *زبان‌های منظم توسط گرامرهای منظم تولید و توسط* ***اتوماتای متناهی*** *تشخیص داده می‌شوند.*

اوتوماتای متناهی(Finite Automata)

پاسخ این پرسش که "آیا یک رشته‌ی خاص متعلق به زبان یک گرامر مور نظر هست؟" ، نه تنها در حالت کلی دشوار بلکه در بعضی موارد ناممکن است. گرامرهای متن_آزاد (CFG) و زبان‌های منظم ، دارای خصوصیاتی هستند که از طریق ساختن یک "تشخیص دهنده"(Recognizer) یا (پذیرنده) (Acceptor) برای رشته‌های متعلق به یک گرامر منظم پاسخ چنین سؤالاتی را امکان پذیر می‌کند. اکنون به معرفی مفهوم یک ماشین (اوتوماتون) به عنوان یک مدل انتزاعی از کامپیوتر که (در تئوری) توانائی انجام اغلب کارهای منتسب به کامپیوترهای رقمی از جمله توانائی تشخیص دقیق رشته‌ها در یک گرامر باقاعده را دارد می‌پردازیم. ماشین مورد نظر ماشینی است تحت عنوان ماشین با حالت متناهی یا اتوماتای متناهی **(Finite Automata)** با علامت اختصاری **FA** و دارای مکانیزمی است که می‌تواند ورودیهارا که رشته‌هائی بر روی الفباء داده شده هستند و بر روی یک " فایل ورودی" (input file) که اصطلاحا به آن ***نوار ورودی*** (input tape) نیز گفته می‌شود نوشته شده است بخواند. فایل ورودی به خانه‌هائی تقسیم بندی شده‌اند که هرخانه می‌تواند یک نماد را در خود نگه دارد. از ترکیبات مهم یک ماشین قسمتی است تحت عنوان " کنترل متناهی" (finite control) که از طریق هد خواندن

(reading head) می‌تواند نمادهای نوشته شده در خانه‌های نوار را احساس کند. کار یک FA در اصل قبول(accept) یا رد (reject) یک ورودی با توجه به الگوی (عبارت منظم)معرفی شده می‌باشد.

یک ماشین را می‌توان سیستمی تصور کرد که **ورودی** را می‌پذیرد ، احتمالا" **خروجی** را تولید می‌کند و از نوعی حافظه‌ی درونی برخوردار است که اطلاعات خاصی را در رابطه با ورودیهای قبلی ره‌گیری می‌کند. چگونگی کامل درونی ماشین و کلیه‌ی حافظه‌ی آن در هر زمان خاصی **وضعیت** یا حالت **(state)** ماشین را در آن زمان تشکیل می‌دهد. با دریافت هر ورودی، ماشین مشخص می‌کند که از وضعیت جاری کدام وضعیت بعدی را اشغال کرده و احتمالا" کدام خروجی را تولید خواهد کرد. ***اگر تعداد وضعیت متناهی باشد، آنگاه ماشین را ماشین با حالت متناهی گویند***. به شکل زیر توجه کنید.



بعد از خواندن یک نماد ورودی, head خواندن یک خانه به راست حرکت و وضعیت finite control تغییر می‌یابد. این عمل تا موقعی که هد خواندن به انتهای رشته ورودی رسیده باشد ادامه پیدا می‌کند.

اتوماتایای متناهی قطعی یا DFA (Deterministic Finite Automata)

در یک ماشین از نوع DFA به ازاء هر ورودی یک و فقط یک وضعیت قطعی و منحصر به فرد وجود دارد که ماشین از وضعیت موجود به آن تغییر وضعیت می‌دهد. در ماشین DFA تغییر وضعیت فقط در ازاء خواندن یک ورودی امکان پذیر است و یک DFA در ازاء رشته تهی تغییر وضعیت نمی‌دهد.

تعریف رسمی DFA

یک DFA را با یک پنج تائی $M = (Q, \Sigma, \delta, q_0, F)$ با مشخصه زیرنشان می‌دهیم.

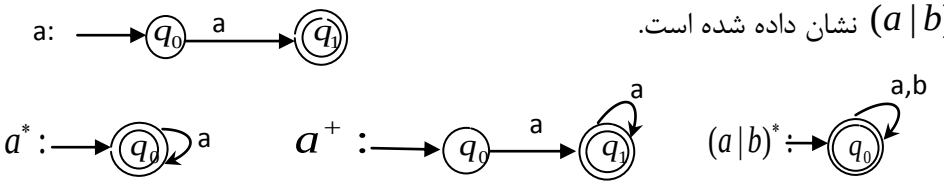
- ✓ Q: مجموعه‌ی متناهی از وضعیت‌ها: $Q = \{q_0, q_1, \ldots, q_n\}$
- ✓ Σ: مجموعه‌ی متناهی از ورودی‌ها (الفباء)
- ✓ δ:تابع تغییر وضعیت (transition function) با ضابطه $\delta : Q \times \Sigma \to Q$ که به هر زوج مرتب $< q_i, x > (x \in \Sigma)$ وضعیت جدیدی را نسبت می‌دهد.
- ✓ $q_0 \in \Sigma$: عنصر ویژه‌ای از Σ بنام **وضعیت آغازی** (initial state) .
- ✓ $F \subseteq Q$: بنام مجموعه‌ی وضعیت‌های نهائی (final states).

یک ماشین DFA ابتدا در وضعیت آغازی q_0 تنظیم شده و نماد ورودی سمت چپ‌ترین کاراکتر رشته ورودی است. در هر حرکت ماشین, مکانیزم ورودی یک سلول به راست پیشرفت کرده و یک نماد از الفباء را مصرف می‌کند. در انتهای رشته اگر ماشین در یکی از وضعیتهای نهائی قرار گرفته باشد رشته مورد نظر پذیرفته شده است(accepted) در غیر این صورت پذیرفته نشده است(rejected). مکانیزم ورودی فقط از چپ به راست حرکت کرده و در هر مرحله فقط یک نماد را می‌خواند. تغییر وضعیت از یک وضعیت درونی به وضعیت دیگر نتیجه عملکرد تابع تغییر وضعیت δ است. به عنوان مثال $\delta(q_0,a)=q_1$ به این معنی است که اگر DFA در وضعیت q_0 بوده و نماد ورودی فعلی a باشد, ماشین به q_1 تغییر وضعیت می‌دهد.

دیاگرام تغییر وضعیت(Transition diagram)

جهت تجسم بهتر عمل‌کرد یک DFA آنرا با یک گراف جهت‌دار بنام دیاگرام تغییر وضعیت نشان می-دهند. در این دیاگرام هر وضعیت را با یک دایره ، هر تغییر وضعیت را با یک خط جهت‌دار برچسب گذاری شده با یک ورودی، و هر وضعیت نهائی (وضعیت پذیرش) را با دو دایره هم‌مرکز نشان می‌دهند.

مثال ۵۱. در زیر چند نمونه از دیاگرام تغییر وضعیت ساده مربوط به عبارت‌های منظم a , a^* , a^+ و $(a\mid b)^*$ نشان داده شده است.



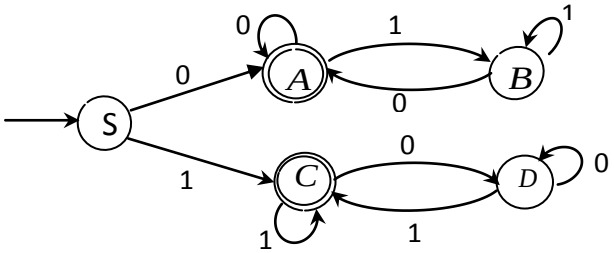
جدول تغییر وضعیت(Transition table)

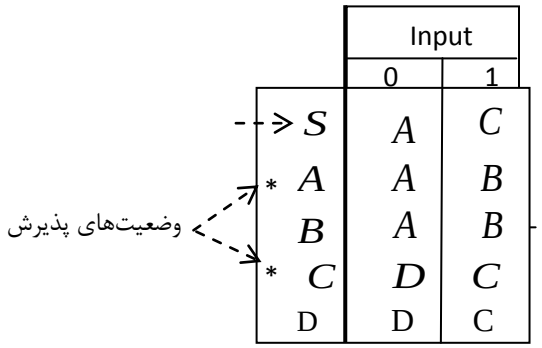
DFA را با یک جدول بنام جدول تغییر وضعیت نیز نشان می‌دهند. در این جدول سرسطرها نشان دهنده‌ی وضعیتها و سر ستونها نشان دهنده‌ی الفباء ورودی و عناصر درون جدول وضعیتهای بعدی در ازاء خوانده شدن نمادها را نشان می‌دهد.

مثال ۵۲. ماشین $M=(Q,\Sigma,\delta,q_0,F)$ با مشخصات زیر را به صورت دیاگرام تغییر وضعیت و جدول تغییر وضعیت نشان دهید. $Q=\{S,A,B,C,D\}$, $\Sigma=\{0,1\}$, $F=\{A,C\}$, $q_0=S$

$$\delta:(S,0)\to A,\quad \delta:(S,1)\to C,\quad \delta:(A,0)\to A,\quad \delta:(A,1)\to B,\delta:(B,0)\to A$$
$$\delta:(B,1)\to B,\quad \delta:(C,0)\to D,\quad \delta:(C,1)\to C,\delta:(D,0)\to D,\quad \delta:(D,1)\to C$$

این ماشین رشته‌هائی از اعداد باینری را می‌پذیرد که نماد آغازی آن همان نماد پایانی هم است.





جدول تغییر وضعیت

عملکرد ماشین را در رابطه با پذیرش یا عدم پذیرش رشته‌های 101001 و 11100 ردگیری کنید.

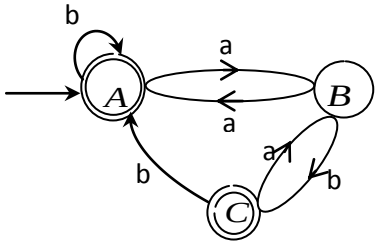
$$101001 \Rightarrow: S \xrightarrow{1} C \xrightarrow{0} D \xrightarrow{1} C \xrightarrow{0} D \xrightarrow{0} D \xrightarrow{1} C \Rightarrow accept$$

$$11100 \Rightarrow: S \xrightarrow{1} C \xrightarrow{1} C \xrightarrow{1} C \xrightarrow{0} D \xrightarrow{0} D \Rightarrow reject$$

ماشین بعد از خواندن کلیه کاراکترهای رشته 101001در وضعیت C که یک وضعیت نهائی است متوقف می‌شود و در نتیجه رشته داده شده پذیرفته می‌شود. ولی بعد از خواندن کلیه کاراکترهای رشته 11100 در وضعیت D که یک وضعیت نهائی نیست متوقف می‌شود و در نتیجه رشته داده شده پذیرفته نمی‌شود. مفهوم ماشین با حالت متناهی تعریف دیگری از زبان‌ها را به شرح زیر ارائه می‌دهد. فرض کنید M یک ماشین با حالت متناهی با الفباء Σ است. مجموعه‌ی کلیه رشته‌های متشکل نمادهای Σ که توسط ماشین M پذیرفته شده باشد را زبان $L(M)$ تولید شده توسط ماشین M گویند. اکنون پرسشی که با آن مواجه هستیم این است که آیا با در دست داشتن گرامر مفروض G می‌توان یک ماشین با حالت متناهی M طراحی کرد به قسمی که $L(G) = L(M)$. جهت پاسخ به این سوال به قضیه زیر توجه کنید.

قضیه. زبان L و یک ماشین با حالت متناهی Mمفروض است. $L = L(M)$ اگر و فقط اگر گرامر منظم G وجود داشته باشد به قسمی که $L = L(G)$

مثال۵۳.ماشین M به صورت دیاگرام زیر داده شده است. گرامر G متناظر با ماشین داده شده را نشان دهید.



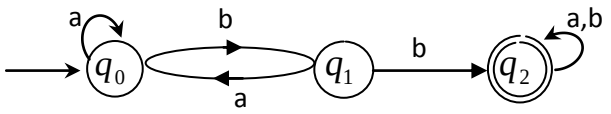
جواب.

1. $A \to bA \mid b$
2. $A \to aB \mid \lambda$
3. $B \to aA \mid a$
4. $B \to bC \mid b$
5. $C \to aB \mid bA \mid b$

مثال۵۴. یک DFA طراحی کنید که رشته‌هائی از الفباء $\Sigma = \{a,b\}$ را بپذیرد که هر رشته شامل زیر رشته bb باشد. به عبارت دیگر $L(M) = (a+b)^* bb(a+b)^*$. در ضمن عملکرد ماشین را در رابطه با پذیرش یا عدم پذیرش رشته‌های $ababba$ ردگیری کنید.

مشخصان کلی ماشین: $M: \quad Q = \{q_0, q_1, q_2\}, \quad \Sigma = \{a,b\}, \quad F = \{q_2\}$

جواب.



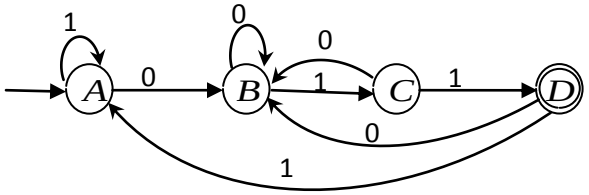
	δ	a	b
→	q_0	q_0	q_1
	q_1	q_0	q_2
*	q_2	q_2	q_2

$$ababba \Rightarrow: q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{b} q_2 \xrightarrow{a} q_2 \Rightarrow accept$$

ماشین بعد از خواندن کلیه کاراکترهای رشته $ababba$ در وضعیت q_2 که یک وضعیت نهائی است متوقف می‌شود و در نتیجه رشته داده شده پذیرفته می‌شود

مثال۵۵. یک DFA طراحی کنید که رشته‌هائی از $\Sigma = \{0,1\}$را بپذیرد که به 011 ختم شده باشد.

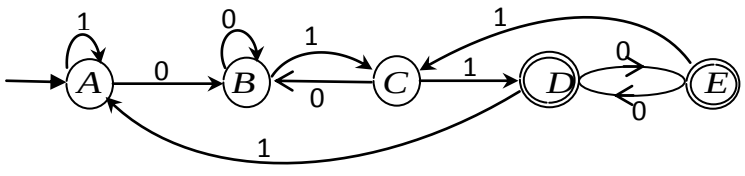
جواب.



مثال۵۶. یک اتوماتای متناهی طراحی کنید که رشته‌هائی از اعداد باینری متعلق به زبان حاصل از گرامر منظم و راست خطی به شرح زیر را بپذیرد.

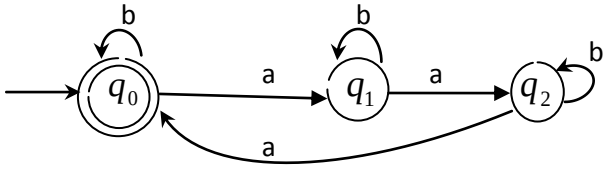
$$P = \{A \to 0B \mid 1A, \quad B \to 0B \mid 1C, \quad C \to 0B \mid 1D \mid 1, \quad D \to 0E \mid 1A \mid 0, \quad E \to 0D \mid 1C \mid 0\}$$

جواب.



مثال۵۷. یک DFA طراحی کنید که زبانی را بر روی الفباء $\Sigma = \{a,b\}$ تولید کند که در آن تعداد a ها مضربی از ۳ باشد.

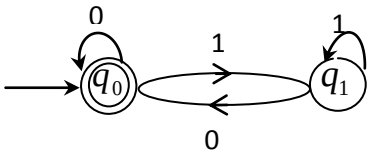
جواب.



مثال۵۸. یک DFA با حداقل وضعیت طراحی کنید که رشته‌هائی از اعداد باینری را بپذیرد که بر۲ بخش پذیر باشد.

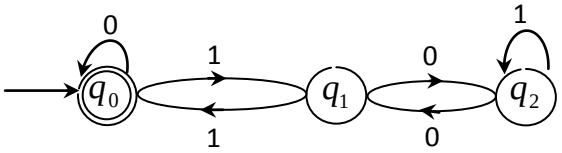
جواب. رشته‌ای از اعداد باینری بر ۲ بخش پذیر است که ان رشته به 0 ختم شده باشد.

	0	1
q_0	q_0	q_1
q_1	q_0	q_1



مثال۵۹. یک DFA طراحی کنید که رشته‌هائی از اعداد باینری را بپذیرد که بر ۳ بخش پذیر باشد.

جواب.اگر رشته بر ۳ بخش پذیر باشد باقیمانده تقسیم برابر 0 است, در غیر این صورت باقیمانده یا 1 است یا 2. پس برای رسم دیاگرام تغییر و ضعیت به سه state نیاز داریم که هر وضعیت برای یکی از باقیمانده‌ها منظور شده است.

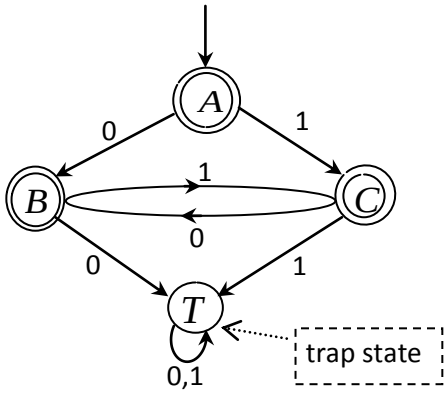


	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

توجه: اگر در دیاگرام تغییر وضعیت یک DFA ممکن است وضعیتی وجود داشته باشد که در صورت ورود به آن امکان خروج از آن وجود ندارد آن وضعیت را ***وضعیت تله*** **(trap state)** گویند.

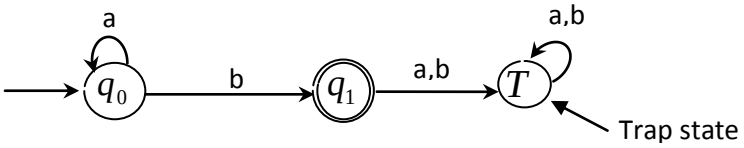
مثال۶۰. یک DFA طراحی کنید که رشته‌هائی از اعداد باینری را بپذیرد که در آن 0ها و 1ها در حال تغییر باشند. به عبارت دیگر بعد هر نماد 1 یک نماد 0 و بعد از هر نماد 0 یک نماد 1 ظاهر شده باشد.

جواب.

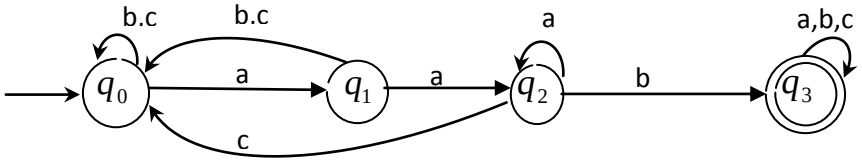


مثال۶۱. یک DFA طراحی کنید که زبان $L = \{a^n b \mid n \geq 0\}$ را تولید کند.

جواب.



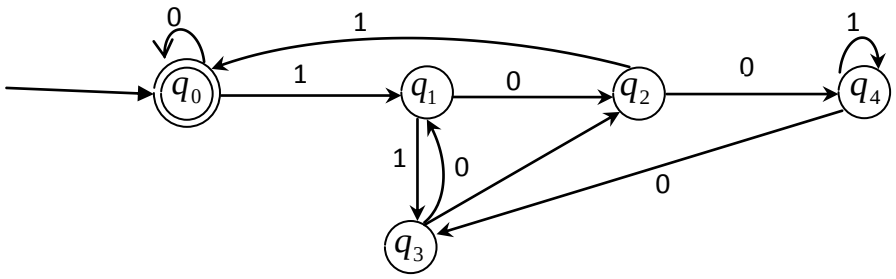
مثال۶۲. یک DFA طراحی کنید که رشته‌هائی را بر روی $\Sigma = \{a, b, c\}$ تولید کند که aab زیر رشته آنها باشد



مثال۶۳. یک DFA طراحی کنید که رشته‌هائی از اعداد باینری را بپذیرد که بر ۵ بخش پذیر باشد.

جواب. باقیمانده تقسیم یک عدد بر ۵ یکی از پنج عدد صحیح 0 تا 4 است. پس برای رسم دیاگرام تغییر وضعیت DFA مورد نظر به ۵ state و هرstate را برای یکی از باقیمانده‌ها به شرح زیر منظور می‌کنیم.

قدم اول. معرفی وضعیت‌ها. وضعیت‌های q_0, q_1, q_2, q_3 و q_4 را به ترتیب برای اعدادی که باقیمانده تقسیم آنها بر۵ برابر 0 یا 1 یا 2 یا 3 یا 4 است منظور می‌کنیم. (q_0 هم وضعیت آغازی است هم نهائی)



بعد از پردازش یک رشته اگر به وضعیت q_0 برسیم یعنی معادل اعشاری رشته مورد نظر بر ۵ بخش پذیر است. و چون 0 بر هر عددی از جمله ۵ بخش پذیر است بنابرین یک حلقه با برچسب 0 در q_0 منظور شده است و تابع تغییر وضعیت: $\delta(q_0,0) \to q_0$

قدم دوم. تعریف توابع تغییر وضعیت .

one: برای پردازش عدد باینری 1 تابع تغییر وضعیت $\delta(q_0,1) \to q_1$ را تعریف می‌کنیم.

two: نمایش باینری 10 و تابع تغییر وضعیت $\delta(q_1,0) \to q_2$. $path: \quad q_0 \overset{1}{\to} q_1 \overset{0}{\to} q_2$

three: نمایش باینری 11 و تابع تغییر وضعیت $\delta(q_1,1) \to q_3$. $path: \quad q_0 \overset{1}{\to} q_1 \overset{1}{\to} q_3$

four: نمایش باینری 100 و تابع تغییر وضعیت $\delta(q_2,0) \to q_4$. $path: \quad q_0 \overset{1}{\to} q_1 \overset{0}{\to} q_2 \overset{0}{\to} q_4$

Five: نمایش باینری 101 و تابع تغییر وضعیت $\delta(q_2,1) \to q_0$, $path: \quad q_0 \overset{1}{\to} q_1 \overset{0}{\to} q_2 \overset{1}{\to} q_0$

Six: نمایش باینری 110 و $6\%5=1$, $path:\quad q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_3 \xrightarrow{0} q_1$

Seven: نمایش باینری 111 و $7\%5=2$, $path:\quad q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_3 \xrightarrow{1} q_2$

eight: نمایش باینری 1000 و $8\%5=3$, $path:\quad q_0 \xrightarrow{100} q_4 \xrightarrow{0} q_3$

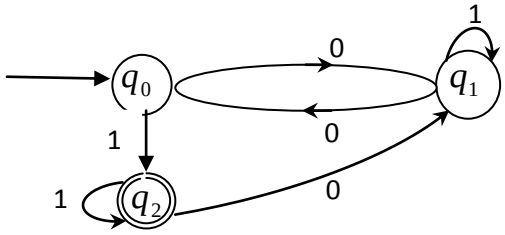
Nine: نمایش باینری 1001 و $9\%5=4$, $path:\quad q_0 \xrightarrow{100} q_4 \xrightarrow{1} q_4$

: دیاگرام تغییر وضعیت

	0	1
q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_4	q_0
q_3	q_1	q_2
q_4	q_3	q_4

مثال۶۴. یک DFA طراحی کنید که رشته‌هائی از اعداد باینری را به پذیرد که تعداد 0 ها زوج بوده و 0ها به بدنبال 1ها منظور شده باشند. عملکرد ماشین را در رابطه با رشته 000101 ردگیری کنید.

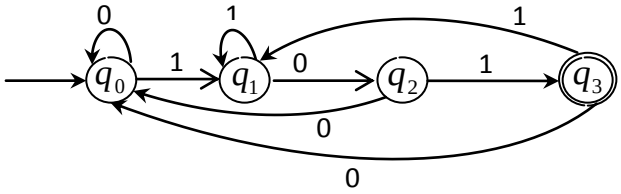
جواب.



$$000101 \Rightarrow 0q_1 00101 \Rightarrow 00q_0 0101 \Rightarrow 000q_1 101 \Rightarrow 0001q_1 01$$
$$\Rightarrow 00010q_0 1 \Rightarrow 000101q_2 \quad accept$$

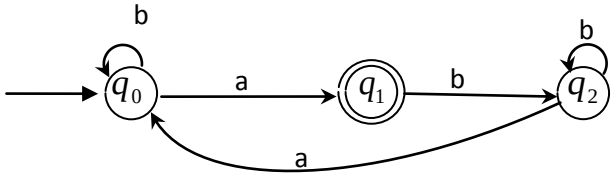
مثال۶۵. یک DFA طراحی کنید که رشته‌هائی از اعداد باینری را به پذیرد که به 101 ختم شود.

جواب.



مثال۶۶. یک DFA طراحی کنید که معادل عبارت منظم $b^*a(b^+ab^*a)^*$ باشد.

جواب.



مینیمم کردن ماشین DFA (DFA minimization)

منظور از مینیمم کردن یک DFA یعنی طراحی یک ماشین معادل آن ولی با حداقل وضعیت. به همین منظور دو الگوریتم به شرح زیر را مورد بر رسی قرار خواهیم داد

الف. "الگوریتم پر کردن جدول " (Table-Filling Algorithm)

ب. الگوریتم افراز وضعیت‌ها.

الف. الگوریتم پر کردن جدول

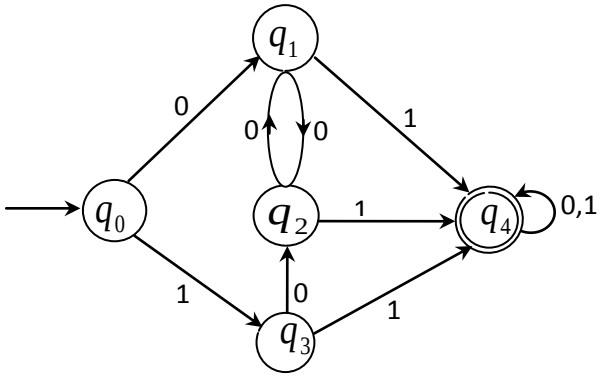
۱. کلیه وضعیتهای غیر قابل دسترس از وضعیت آغازی را حذف می‌کنیم.

۲. به ازاء کلیه وضعیتهای موجود در DFA مورد نظر لیست زوج‌هائی از وضعیت‌ها مانند (q_i, q_j) را که تعداد آنها برابر $\frac{n(n-1)}{2}$ می‌باشد تنظیم می‌کنیم.

۳. از زوج‌های لیست شده آنهائی‌که فقط یکی از وضعیتها وضعیت نهائی و دیگری وضعیت نهائی نباشد را به عنوان *زوج‌های ادغام ناپذیر* علامت گذاری می‌کنیم. (زوجهائی مانند (q_i, q_j) که در آن هم q_i و هم q_j متعلق به وضعیت‌های نهائی باشند را علامت‌گذاری نمی‌کنیم)

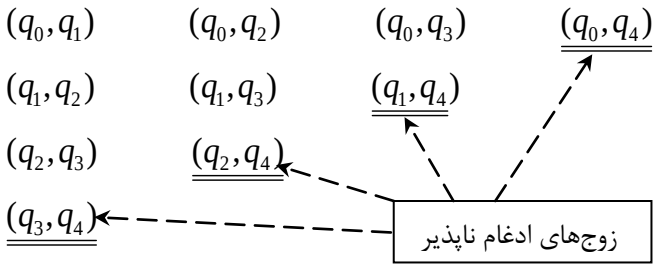
۴. به ازاء کلیه (q_i, q_j) های علامت گذاری نشده خروجیهای q_i و q_j را به ازاء تمامی حروف الفباء بررسی می‌کنیم. اگر به ازاء عنصری مانند $a \in \Sigma$, $\delta(q_i, a) = p$ و $\delta(q_j, a) = q$ و زوج (p,q) علامت گذاری شده باشد آنگاه نتیجه می‌گیریم q_i و q_j نیز ادغام ناپذیرند و زوج (q_i, q_j) را به عنوان زوج ادغام ناپذیر علامت گذاری می‌کنیم.

مثال ۶۷ . DFA ارائه شده مطابق شکل زیر را مینی‌مایز کنید.



جواب.

تعداد زوجهای لیست شده به تعداد $\frac{n(n-1)}{2} = \frac{5 \times 4}{2} = 10$ به شرح زیر است .



✓. $(q_3,q_4)\overset{1}{\leftarrow}(q_0,q_1)\overset{0}{\rightarrow}(q_1,q_2)$ چون q_3 و q_4 ادغام ناپذیرند پس q_0 و q_1 نیز ادغام ناپذیرند و زوج (q_0,q_1) را نیز علامت گذاری می‌کنیم.

✓. $(q_3,q_4)\overset{1}{\leftarrow}(q_0,q_2)\overset{0}{\rightarrow}(q_1,q_1)$ چون q_3 و q_4 ادغام ناپذیرند پس q_0 و q_2 نیز ادغام ناپذیرند و زوج (q_0,q_2) را نیز علامت گذاری می‌کنیم.

✓. $(q_3,q_4)\overset{1}{\leftarrow}(q_0,q_3)\overset{0}{\rightarrow}(q_1,q_2)$ چون q_3 و q_4 ادغام ناپذیرند پس q_0 و q_3 نیز ادغام ناپذیرند و زوج (q_0,q_3) را نیز علامت گذاری می‌کنیم.

✓. $(q_4,q_4)\overset{1}{\leftarrow}(q_1,q_2)\overset{0}{\rightarrow}(q_2,q_1)$ چون q_2 و q_2 باهم و q_4 و q_4 نیز باهم ادغام پذیرند پس زوج (q_1,q_2) را علامت گذاری نمی‌کنیم.

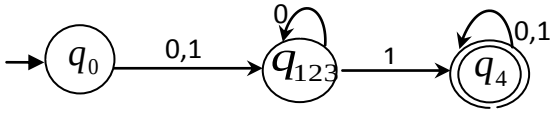
✓. $(q_4,q_4)\overset{1}{\leftarrow}(q_1,q_3)\overset{0}{\rightarrow}(q_2,q_2)$ چون q_2 و q_1 ادغام پذیرند پس q_3 و q_1 نیز ادغام پذیرند و زوج (q_1,q_3) را علامت گذاری نمی‌کنیم.

✓. $(q_4,q_4)\overset{1}{\leftarrow}(q_2,q_3)\overset{0}{\rightarrow}(q_1,q_2)$ چون q_1 و q_2 ادغام پذیرند پس q_2 و q_3 نیز ادغام پذیرند و زوج (q_2,q_3) را علامت گذاری نمی‌کنیم.

الگوریتم را با پر کردن جدول به شکل زیر بکار می‌بریم . خانه‌های خالی جدول خانه‌هائی ادغام پذیرند.

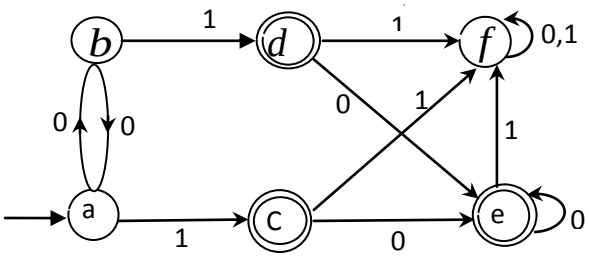
q_1	×			
q_2	×			
q_3	×			
q_4	×	×	×	×
	q_0	q_1	q_2	q_3

وضعیتهای q_1,q_2,q_3 ادغام پذیرند. دیاگرام بهینه شده به صورت زیر است.



ماشین فوق زبان عبارت منظم $(0+1)0^*1(0+1)^*$ را تولید می‌کند.

مثال۶۸. DFA زیر را مینیمایز کنید.



جواب. تعداد زوجهای لیست شده به تعداد $\frac{n(n-1)}{2}=\frac{6\times5}{2}=15$ به شرح زیر است .

(a,b) $\underline{\underline{(a,c)}}$ $\underline{\underline{(a,d)}}$ $\underline{\underline{(a,e)}}$ (a,f)

$\underline{\underline{(b,c)}}$ $\underline{\underline{(b,d)}}$ $\underline{\underline{(b,e)}}$ (b,f)

(c,d) (c,e) $\underline{\underline{(c,f)}}$

(d,e) $\underline{\underline{(d,f)}}$

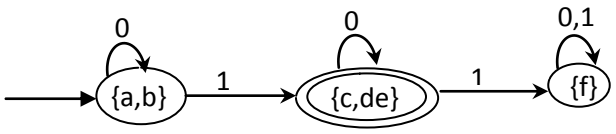
$\underline{\underline{(e,f)}}$

رسم جدول.

جدول زیر در دو مرحله تنظیم شده است. در مرحله اول زوجهائی که در بالا با دو زیر خط مثلا $\underline{\underline{(e,f)}}$ و نظایر آن مشخص شده اند علامت گذاری شده اند و علامت گذاری در مرحله دوم با استفاده از بند ۴ الگوریتم انجام شده است. وخانه‌هائی که درون آنها خالی است وضعیتهای درج شده در سطر و ستون آنها ادغام پذیرند.

b					
c	✓	✓			
d	✓	✓			
e	✓	✓			
f	✓	✓	✓	✓	✓
	a	b	c	d	e

رسم DNF مینیمایز شده



ب.بهینه کردن DFA با استفاده از الگوریتم افراز وضعیت‌ها

۱. ابتدا کلیه وضعیتهای قابل دسترس از وضعیت آغازی را به دو بلوک , یک بلوک شامل کلیه وضعیت های پایانی و بلوک دیگر شامل کلیه وضعیت‌های غیر پایانی افراز می‌کنیم.

۲. فرض کنید A و B دو وضعیت از یک بلوک هستند و به ازاء $x \in \Sigma$ (Σ الفباء مفروض) نتیجه بگیریم $\delta(A,x) = p$ و $\delta(B,x) = q$. اگر $p = q$ یا p و q هردو متعلق به یک بلوک باشند آنگاه p و q معادلند $(p \approx q)$. درغیر اینصورت A و B به دو بلوک متمایز جدید تفکیک خواهد شد.

۳. با تکرار عمل مذکور در بند ۲ بالا برای کلیه‌ی زوجهائی از وضعیت‌های هر بلوک به این نتیجه میرسیم که امکان تفکیک بلوک‌های افراز امکان پذیر نیست. ودر نتیجه به پایان الگوریتم رسیده‌ایم.

مثال۶۹ . DFA ارائه شده در مثال ۶۸ را با استفاده از الگوریتم افراز وضعیت‌ها مینیمایز کنید.

الف. افراز DFA داده شده به ۲ بلوک, شامل وضعیتهای پایانی و وضعیتهای غیر پایانی می‌کنیم.

$$0 - Equivalenc: \Pi_0 = \{\{a,b,f\},\{c,d,e\}\}$$

جدول تغییر وضعیت به تفکیک بلوک‌ها.

	q	δ(q,0)	δ(q,1)
بلوک 1 شامل وضعیتهای غیر پایانی	a	b	c
	b	a	d
	f	f	f
بلوک 2 شامل وضعیتهای پایانی	c	e	f
	d	e	f
	e	e	f

ب. ✓ $\delta(a,0) = b$ و $\delta(b,0) = a$ و a و b هردو متعلق به یک بلوک یعنی بلوک۱ هستند.

$\delta(a,1) = c$ و $\delta(b,1) = d$ و c و d هردو متعلق به یک بلوک یعنی بلوک ۲ هستند.

پس a و b معادلند $(a \approx b)$ و هردو در یک بلوک باقی می‌مانند.

✓ $\delta(a,0) = b$ و $\delta(f,0) = f$ و b و f هردو متعلق به یک بلوک یعنی بلوک ۱ هستند. ولی $\delta(a,1) = c$ و $\delta(f,1) = f$ که c و d *متعلق به دو بلوک متفاوت هستند*. پس a و f معادل نیست. بنابراین بلوک ۱ را به دو بلوک مجزا مثلا بلوک ۱ شامل $\{a,b\}$ و بلوک ۳ شامل $\{f\}$ تفکیک می‌کنیم.

$$1 - Equivalence: \Pi_1 = \{\{a,b\},\{f\},\{c,d,e\}\}$$

جدول تغییر وضعیت به تفکیک بلوک‌های جدید.

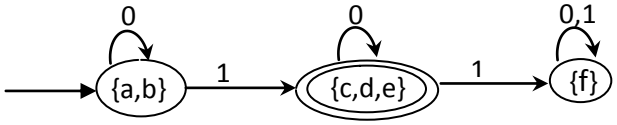
	q	δ(q,0)	δ(q,1)
بلوک 1	a	b	c
	b	a	d
بلوک 3	f	f	f
بلوک 2 شامل وضعیتهای پایانی	c	e	f
	d	e	f
	e	e	f

ج. با توجه به جدول تغییر وضعیت بالا

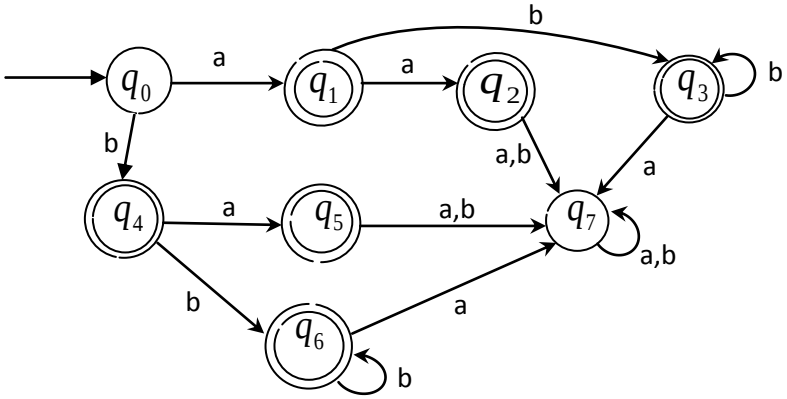
✓. $\delta(c,0)=e$ و $\delta(d,0)=e$ و $\delta(c,1)=f$ و $\delta(d,1)=f$ پس $(c \approx d)$

✓. $\delta(d,0)=e$ و $\delta(e,0)=e$ و $\delta(d,1)=f$ و $\delta(e,1)=f$ پس $(d \approx e)$

بنا به خاصیت تعدی $(c \approx d)\ \ and\ \ (d \approx e) \Rightarrow (c \approx e)$ پس وضعیتهای c وd و e معادلند و هرسه درهمان بلوک ۲ باقی میمانند. بنابراین $\Pi_1 = \Pi_2 = \{\{a,b\},\{f\},\{c,d,e\}\}$ و پایان الگوریتم و دیاگرام تغییر و ضعیت DFA بهینه شده به صورت زیر است.



مثال۷۰ . DFA زیر را با استفاده از الگوریتم افراز وضعیتها مینی‌مایز کنید.

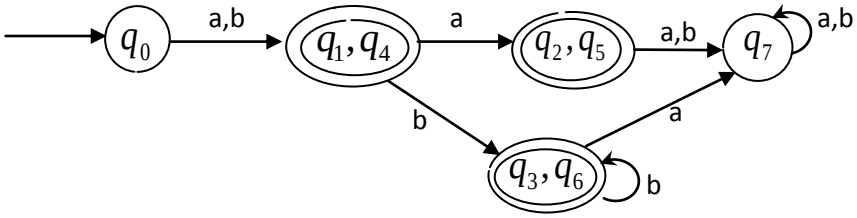


افراز وضعیت‌ها به صورت ذهنی انجام شده است.

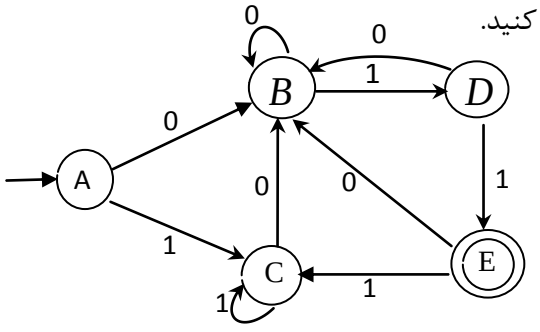
$\Pi_0 = \{\{q_0,q_7\},\{q_1,q_2,q_3,q_4,q_5,q_6\}\}$, $\Pi_1 = \{\{q_0\},\{q_7\},\{q_1,q_2,q_3,q_4,q_5,q_6\}\}$

$\Pi_2 = \{\{q_0\},\{q_7\},\{q_1,q_4\},\{q_2,q_3,q_5,q_6\}\}$, $\Pi_3 = \{\{q_0\},\{q_7\},\{q_1,q_4\},\{q_2,q_5\},q_3,q_6\}\}$

DFA بهینه: چون $\Pi_2 = \Pi_3$ پس پایان الگوریتم.



مثال ۷۱. DFA زیر را می‌نیمایز کنید.

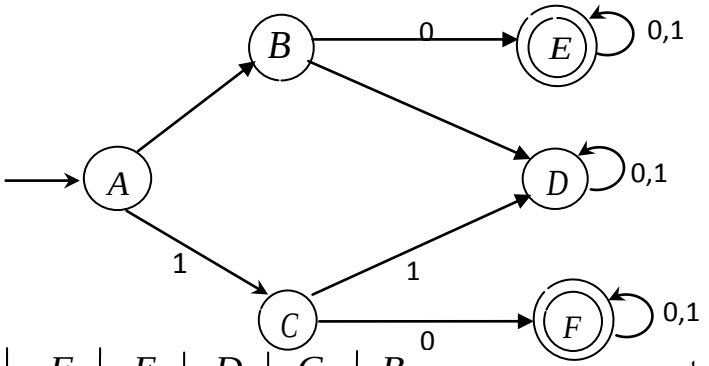


$0Equivalenc: \Pi_0 = \{\{A,B,C,D\},\{E\}\}$ ، $1Equivalenc: \Pi_1 = \{\{A,B,C\},\{D\},\{E\}\}$

$3Equivalenc: \Pi_3 = \{\{A,C\},\{B\},\{D\},\{E\}\}$ ، $2Equivalenc: \Pi_2 = \{\{A,C\},\{B\},\{D\},\{E\}\}$

چون $\Pi_2 = \Pi_3 = \{\{A,C\},\{B\},\{D\},\{E\}\}$ پس پایان الگوریتم (رسم دیاگرام بهینه بعهده دانشجو)

مثال۷۲. DFA ارائه شده مطابق شکل زیر را با استفاده از الگوریتم غربال مینی‌مایز کنید.

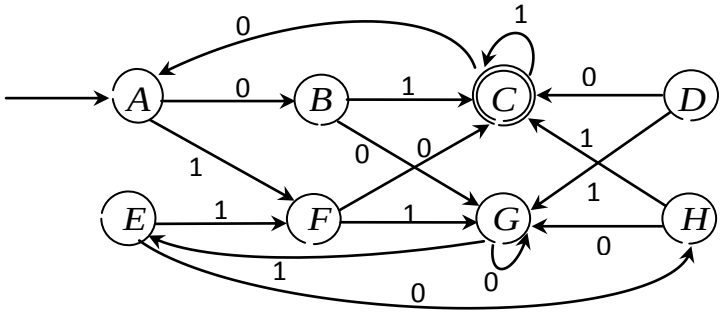


جواب.

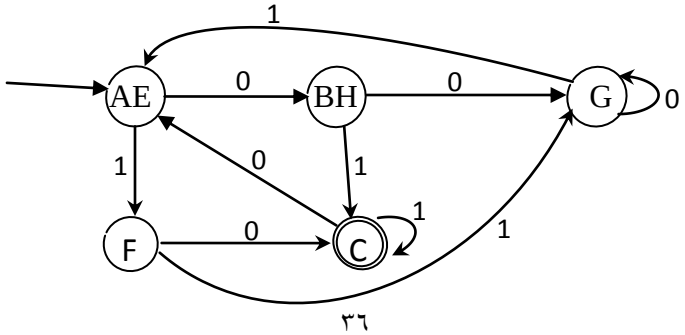
	F	E	D	C	B
A	×	×	×	×	×
B	×	×	×		
C	×	×	×		
D	×	×			
E					

رسم دیاگرام DFA بهینه و حل کامل مسئله به روش‌های مختلف به عهده دانشجو.

مثال۷۳. DFA زیر را مینی‌مایز کنید. (توجه: ابتدا وضعیت D را که از وضعیت آغازی قابل دسترس نیست حذف کنید.)



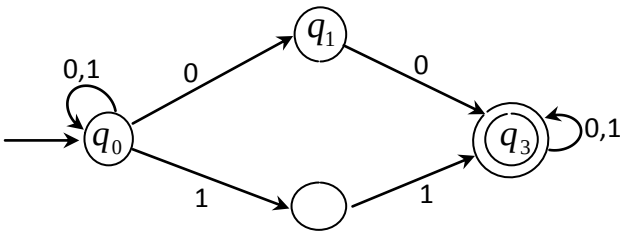
جواب.شکل DFA بهینه در زیر ارائه شده است. حل کامل به عهده دانشجو.



اوتوماتای متناهی غیر قطعی یا NFA (Nondeterministic Finite Automata)

در یک ماشین از نوع NFA برای هر وضعیت و به ازاء بعضی از نمادهای الفباء ممکن است و ضعیت بعدی و قطعی مشخص نبوده و یا وضعیت‌های بعدی متعددی داشته باشد. در ضمن ماشین NFA ممکن است طوری طراحی شود که بدون خواندن یک ورودی (خواندن رشته تهی) نیز به وضعیت جدیدی تغییر یابد. ماشین NFA که با خواندن رشته تهی تغییر وضعیت دهد را اصطلاحا λ–Transition گویند. اگر ماشین در وضعیتی قرار گیرد که با خواندن یک نماد شاخه‌های متعددی از وضعیتها پیش بینی شده باشد ممکن است بعضی از شاخه‌ها موجب رد(reject) و بعضی دیگر موجب قبول(accept) ورودی مورد نظر گردد.

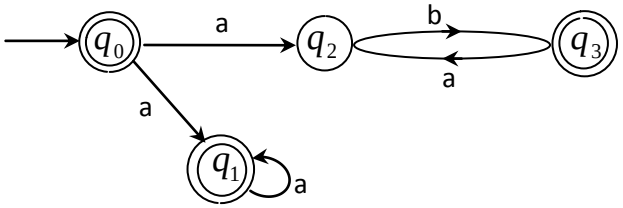
مثال۷۴. چرا ماشین مطابق دیاگرام زیر از نوع غیر قطعی (NFA) است؟ و ماشین داده شده چه نوع رشته‌هائی از اعداد باینری را می‌پذرد؟



جواب. در ماشین داده شده $\delta(q_0,0) \to q_0$ و $\delta(q_0,0) \to q_1$ هم‌چنین $\delta(q_0,1) \to q_0$ و $\delta(q_0,1) \to q_2$. ماشین از نوع NFA است و رشته‌هائی می‌پذیرد که شامل زیر رشته 00 یا 11 است.

مثال۷۵. یک NFA طراحی کنید که رشته‌های $a^* + (ab)^*$ را به‌پذیرد.

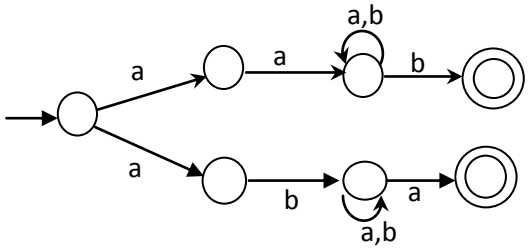
جواب.



مثال۷۶. یک NFA طراحی کنید که بر روی الفباء {a,b} رشته‌هائی را به پذیرد که با aa آغاز و با b خاتمه پذیرد یا با ab آغاز و با a خاتمه پذیرد. به عبارت دیگر

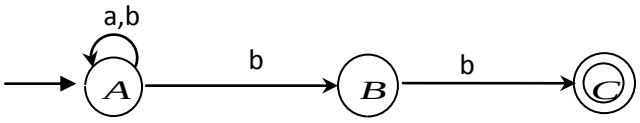
$$L = \{aaxb \mid x \in \{a,b\}^*\} \cup \{abya \mid x \in \{a,b\}^*\}$$

جواب.



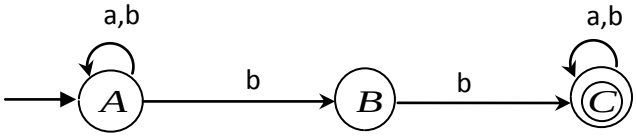
مثال۷۷. بر روی الفباء {a,b} یک NFA طراحی کنید که زبان $(a+b)^*bb$ بپذیرد.

جواب.



مثال۷۸. بر روی $\Sigma=\{a,b\}$ یک NFA طراحی کنید که رشته‌هائی را به پذیردکه bb زیر رشته آن است

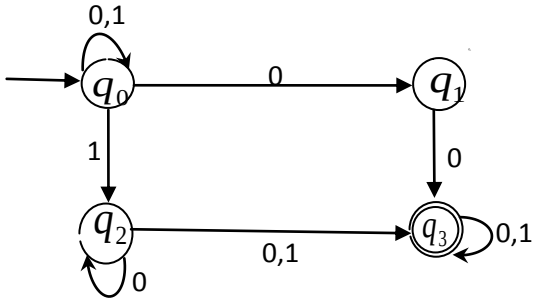
جواب.



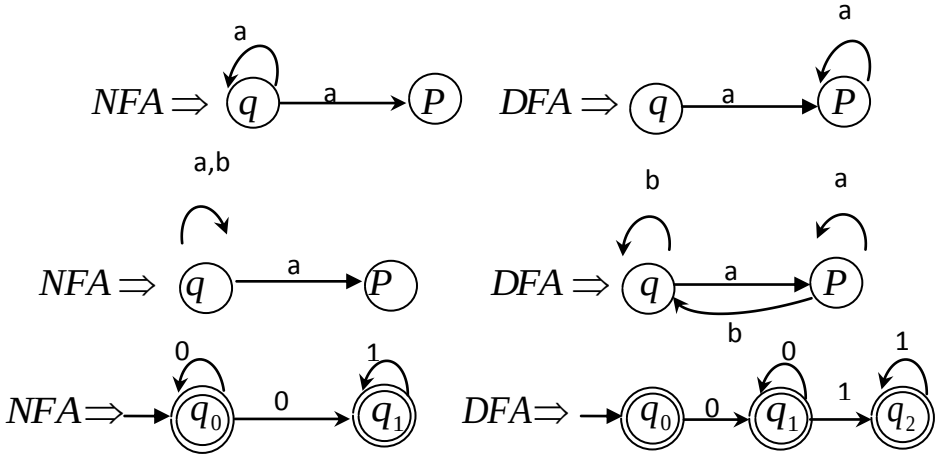
مثال۷۹. یک NFA به صورت جدول تغییر وضعیت در زیر ارائه شده است. NFA مورد نظر را به صورت دیاگرام تغییر وضعیت نشان دهید. q_3 وضعیت نهائی است.

		0	1
→	q_0	$\{q_0,q_1\}$	$\{q_0,q_2\}$
	q_1	$\{q_3\}$	–
	q_2	$\{q_2,q_3\}$	$\{q_3\}$
*	q_3	$\{q_3\}$	$\{q_3\}$

جواب.

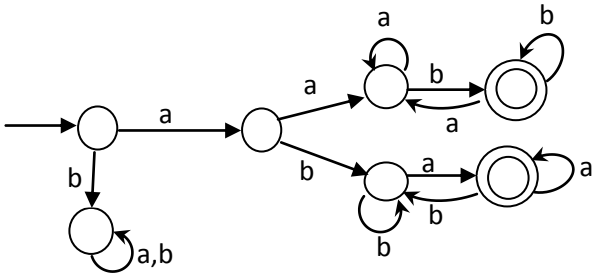


مثال ۸۰ . تبدیل NFA به DFA به روش ذهنی (HEURISTIC)

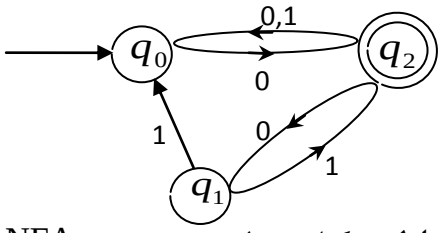


مثال۸۱. NFA مورد نظر در مثال ۷۶ را به DFA تبدیل کنید

جواب.



مثال۸۲. NFA زیر را به DFA تبدیل کنید.



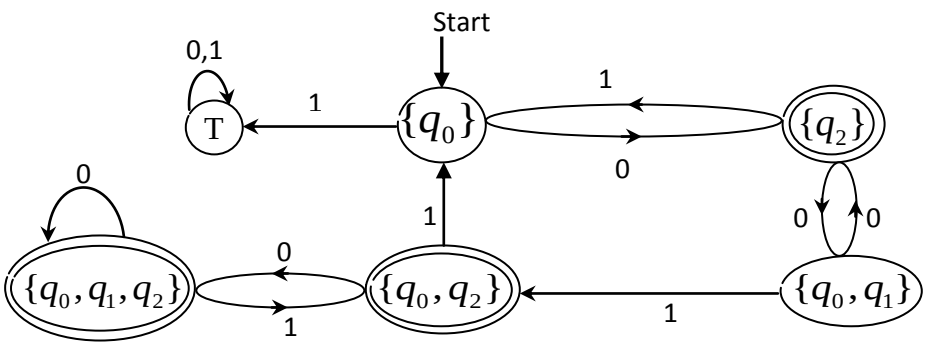
جواب. مرحله اول.تشکیل جدول تغییر وضعیت NFA.

		0	1
→	q_0	$\{q_2\}$	$\emptyset$
	q_1	$\emptyset$	$\{q_0,q_2\}$
*	q_2	$\{q_0,q_1\}$	$\{q_0\}$

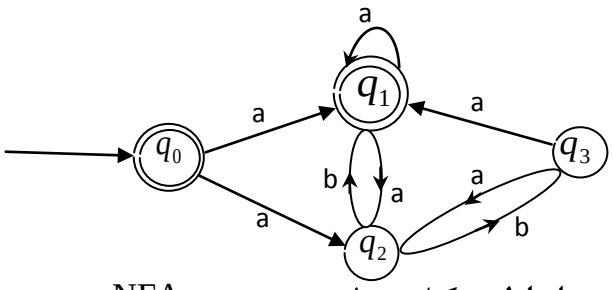
مرحله دوم.تشکیل جدول تغییر وضعیت DFA را متناظر.

		0	1
→	$\{q_0\}$	$\{q_2\}$	$\emptyset$
*	$\{q_2\}$	$\{q_0,q_1\}$	$\{q_0\}$
	$\{q_0,q_1\}$	$\{q_2\}$	$\{q_0,q_2\}$
*	$\{q_0,q_2\}$	$\{q_0,q_1,q_2\}$	$\{q_0\}$
*	$\{q_0,q_1,q_2\}$	$\{q_0,q_1,q_2\}$	$\{q_0,q_2\}$

مرحله سوم.تشکیل دیاگرام تغییر وضعیت DFA را متناظر.



مثال۸۳. NFA زیر را به DFA تبدیل کنید.



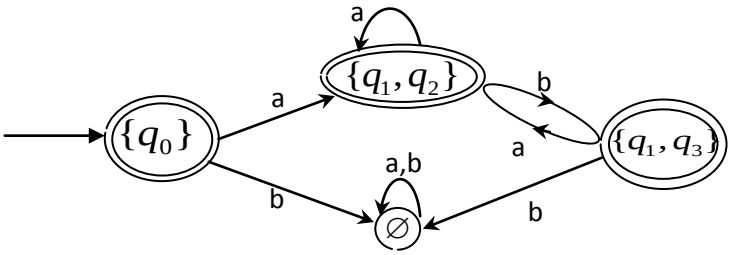
جواب. مرحله اول.تشکیل جدول تغییر وضعیت NFA.

		a	b
→ *	q_0	$\{q_1, q_2\}$	$\varnothing$
*	q_1	$\{q_1, q_2\}$	$\varnothing$
	q_2	$\varnothing$	$\{q_1, q_3\}$
	q_3	$\{q_1, q_2\}$	$\varnothing$

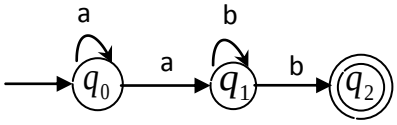
مرحله دوم.تشکیل جدول تغییر وضعیت DFA متناظر.

	a	b
→ * $\{q_0\}$	$\{q_1, q_2\}$	$\varnothing$
* $\{q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_1, q_3\}$
* $\{q_1, q_3\}$	$\{q_1, q_2\}$	$\varnothing$

مرحله سوم.تشکیل دیاگرام تغییر وضعیت DFA را متناظر.



مثال۸۴. NFA زیر را به DFA تبدیل کنید. (این ماشین زبان a^+b^+ می‌پذیرد.)

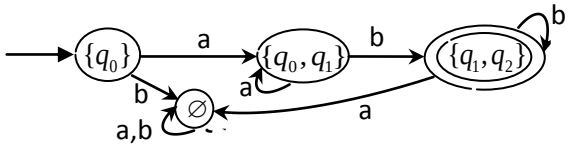


جدول تغییر وضعیت NFA

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

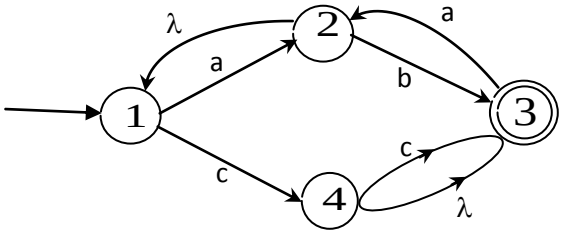
جدول تغییر وضعیت DFA

	a	b
$\{q_0)$	$\{q_0, q_1\}$	$\varnothing$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\varnothing$
$\{q_1, q_2\}$	$\varnothing$	$\{q_1, q_2\}$



تبدیل λ-NFA به DFA

مثال۸۵. λ-NFA زیر را به DFA تبدیل کنید.

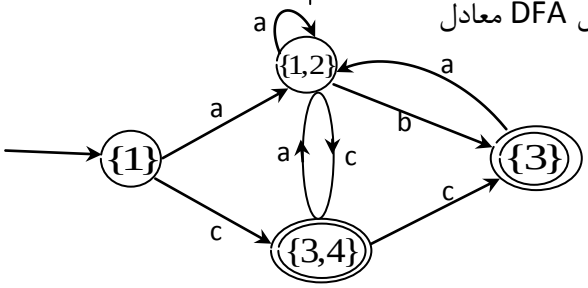


$\lambda - closure(1) = \{1\}, \quad \lambda - closure(2) = \{2,1\}, \quad \lambda - closure(3) = \{3\}, \quad \lambda - closure(4) = \{3,4\}$

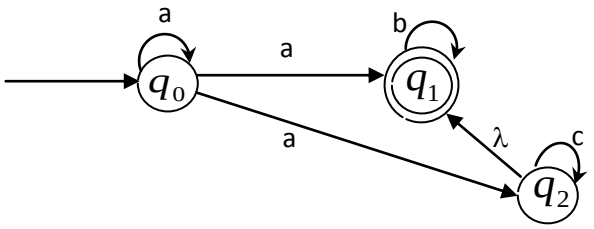
جدول تغییر وضعیت DFA

	δ	a	b	c
→	{1}	{1,2}	∅	{3,4}
	{1,2}	{1,2}	{3}	{3,4}
*	{3}	{1,2}	∅	∅
*	{3,4}	{1,2}	∅	{3}

شکل DFA معادل



مثال۸۶. λ-NFA زیر را به DFA تبدیل کنید.



جواب. الف. تعیین $\lambda - closure$ وضعیت‌ها.

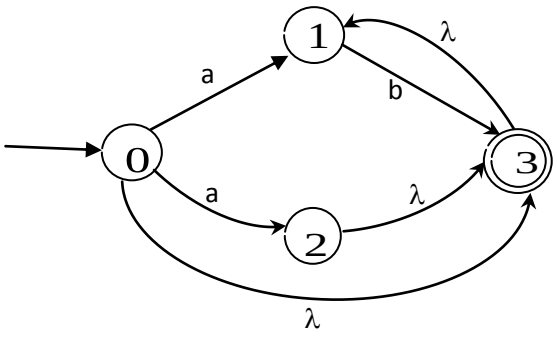
$\lambda - closure(q_0) = \{q_0\}, \quad \lambda - closure(q_1) = \{q_1\}, \quad \lambda - closure(q_2) = \{q_1, q_2\}$

ب.تشکیل جدول تغییر وضعیت DFA با استفاده از وضعیتهای بدست آمده از $\lambda - closure$ وضعیتها.

	a	b	c
→ $\{q_0\}$	$\{q_0, q_1, q_2\}$	∅	∅
* $q_1\}$	∅	$\{q_1\}$	∅
* $q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1\}$	$\{q_1, q_2\}$
* $\{q_1, q_2\}$	∅	$\{q_1\}$	$\{q_1, q_2\}$

تشکیل دیاگرام تغییر وضعیت DFA. به عهده دانشجو

مثال۸۷. λ-NFA زیر را به DFA تبدیل کنید.

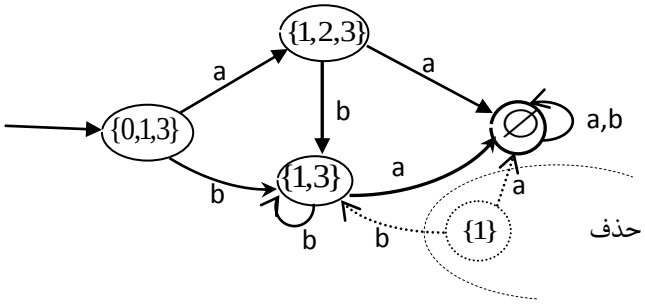


$$\lambda - closure(0) = \{0,3,1\}, \quad \lambda - closure(1) = \{1\},$$
$$\lambda - closure(2) = \{2,3,1\}, \quad \lambda - closure(3) = \{3,1\}$$

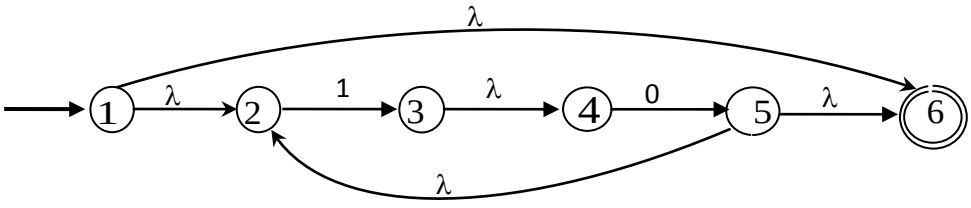
جدول تغییر وضعیت DFA

	δ	a	b
→ *	[0,3,1}	{1,2,3}	{1,3}
	{1}	∅	{1,3}}
*	{1,3}	∅	{1,3}
*	{1,2,3}	∅	{1,3}

دیاگرام تغییر وضعیت DFA



مثال۸۷. λ-NFA زیر را به DFA تبدیل کنید.

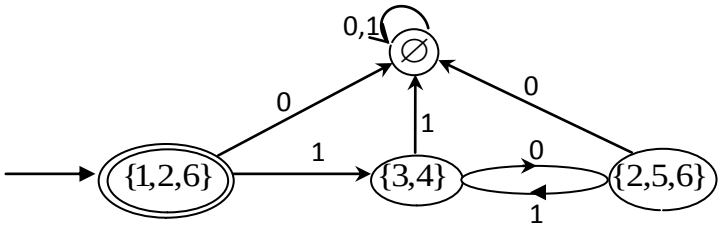


الف. تعیین $\lambda - closure$ وضعیتها.

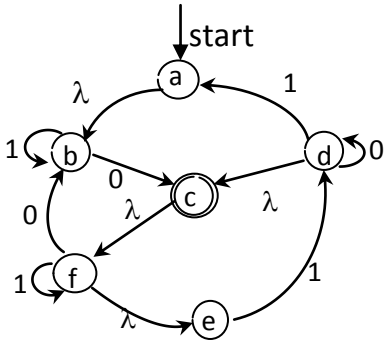
$$\lambda - closure(1) = \{1,2,6\}, \quad \lambda - closure(2) = \{2\}, \quad \lambda - closure(3) = \{3,4\},$$
$$\lambda - closure(4) = \{4\}, \lambda - closure(5) = \{2,5,6\}, \quad \lambda - closure(6) = \{6\}$$

ب.تشکیل جدول تغییر وضعیت DFA با استفاده از وضعیتهای بدست آمده از $\lambda-closure$ وضعیتها

δ	0	1
{1,2,6}	$\varnothing$	{3,4}
{3,4}	{2,5,6}	$\varnothing$
{2,5,6}	$\varnothing$	{3,4}

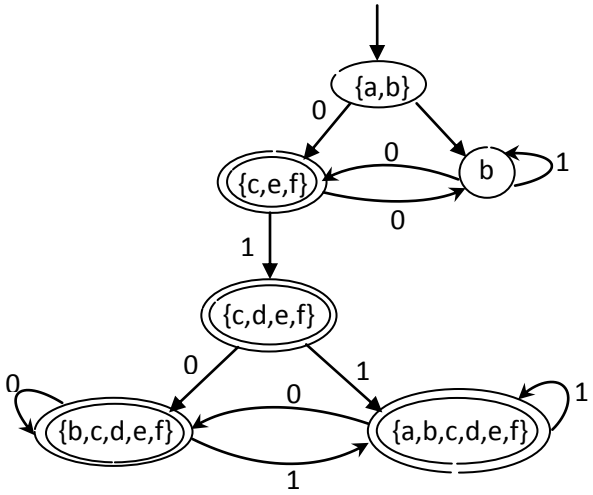


مثال۸۸. λ-NFA زیر را به DFA تبدیل کنید.

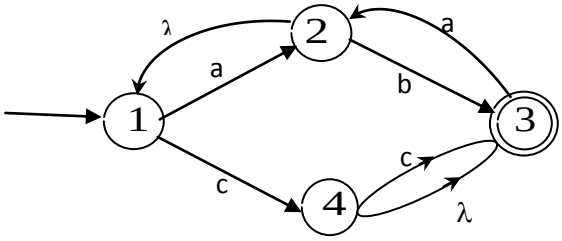


جواب. تعیین $\lambda-closure$ وضعیت‌ها.

$$\lambda-closure(a)=\{a,b\},\quad \lambda-closure(b)=\{b\},\quad \lambda-closure(c)=\{c,e,f\},$$
$$\lambda-closure(d)=\{c,d,e,f\},\quad \lambda-closure(e)=\{e\},\quad \lambda-closure(f)=\{e,f\}$$



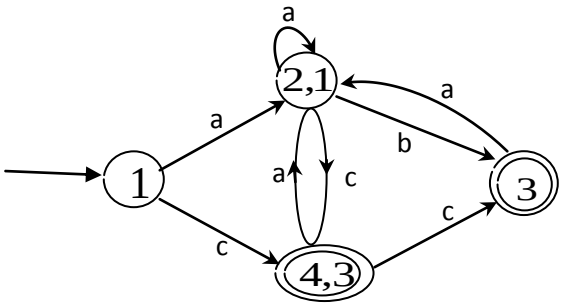
مثال۸۹. λ-NFA زیر را به DFA تبدیل کنید.



$\lambda - closure(1) = \{1\}, \quad \lambda - closure(2) = \{1,2\}, \quad \lambda - closure(3) = \{3\},$
$\lambda - closure(4) = \{4,3\},$

	a	b	c
→ {1}	{2,1}	–	{4,3}
{2,1}	{2,1}	{3}	{4,3}
*{4,3}	{2,1}	–	{3}
*{3}	{2,1}	–	–

شکل DFA معادل



ماشین پشته‌ای (PushDown Automaton) یا PDA

در این قسمت نوعی از مدل کامپیوتر را تحت عنوان PDA مورد بررسی قرار می‌دهیم که در مقایسه با DFA به مراتب قوی‌تر بوده و با اضافه کردن یک نوع حافظه بنام پشته(stack) به DFA ایجاد می‌شود.

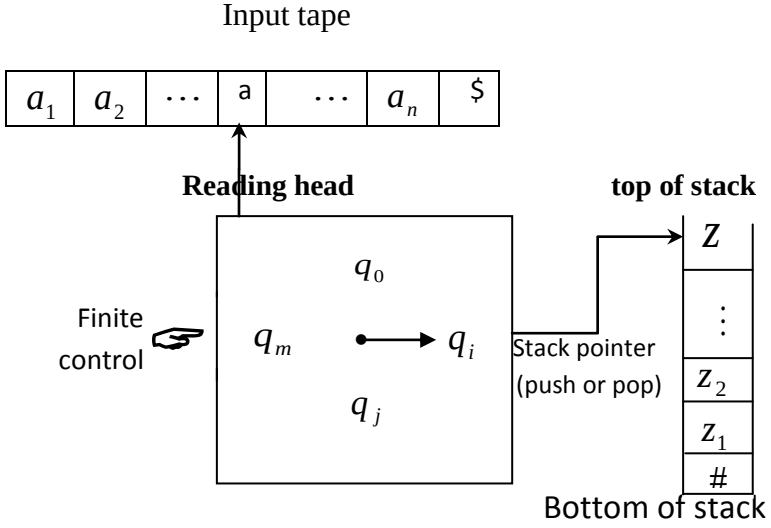
هر PDA از سه قسمت شامل یک نوار ورودی ,یک واحد کنترل متناهی و یک پشته با سایز متناهی تشکیل شده است.

یک پشته در PDA دو عمل به شرح زیر انجام می‌دهد:

✓ PUSH: اضافه کردن یک نماد جدید به ابتدای پشته.

✓ POP: خواندن نماد بالای پشته و برداشتن آن.

بنابراین پشته مورد نظر وسیله‌ایست جهت ذخیره اطلاعات که بر اساس LIFO (**L**ast **I**n **F**irst **O**ut) طراحی شده است. به این معنی که هرچیزی که به پشته اضافه (push) یا از آن برداشته شود (pop) به ابتدا یا اصطلاحا بالای(top) آن اضافه شده یا از ابتدای آن برداشته می‌شوند.*(به شکل زیر توجه کنید)*



یک PDA با خواندن یک نماد و با توجه به موار زیر:

✓. وضعیتی فعلی ماشین (در شکل بالا وضعیت q_i)

✓. نماد قرار گرفته در بالای پشته (در شکل بالا نماد Z)

✓. نماد خوانده شده (در شکل بالا نماد a)

می‌تواند یک یا چند فعالیت به شرح زیر را انجام دهد:

۱. تغییر وضعیت.

۲. حرکت هد خواندن یک خانه به راست.

۳. جایگزینی نماد بالای پشته (نماد Z در شکل) با نمادهای پشته: عمل POP یا PUSH یک نماد و تنظیم مجدد اشاره‌گر پشته به TOP (بالای) پشته.

۴. توقف (Halt)

اتوماتای پشته‌ای می‌تواند قطعی یا غیر قطعی باشد. اتوماتای پشته‌ای غیر قطعی را با نماد NPDA مخفف عبارت Nondeterministic PDA نشان می‌دهیم.(یکPDA می‌تواند بدون خواندن ورودی هم عملPOP یا PUSH را انجام دهد).

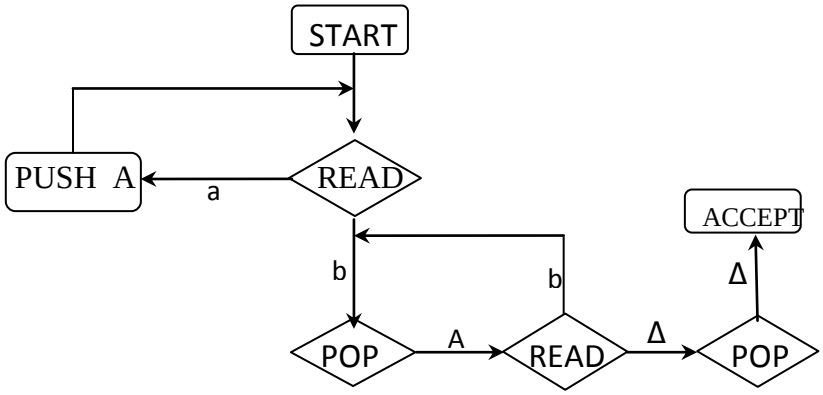
توجه: در پیکربندی آغازی, نماد # در انتهای پشته و نماد $ در سمت راست هر رشته ورودی مانند σ منظور شده است. با منظور نمودن نمادهای # و $, به ترتیب , ماشین انتهای پشته و پایان رشته ورودی را تشخیص می‌دهد. برنامه یک PDA را می‌توان با یک نمای گردشی با مشخصات به شرح زیر نشان داد:

- A single start state
- A single **halt-and-accept** state
- A **reader** box
- A **POP** box
- A **PUSH** box

در نمای گردشی ماشین از نماد Δ جهت نشان دادن پایان رشته ورودی و هم‌چنین نتیجه عمل POP وقتی که پشته خالی شده باشد استفاده شده است.

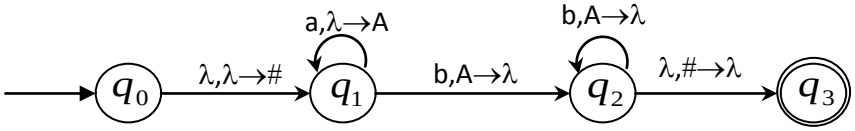
مثال۸۸ . یک PDA برای زبان $L = \{a^n b^n : n > 0\}$ رسم کنید.

یک PDA در واقع از پشته به عنوان یک شمارشگر استفاده می‌کند.)



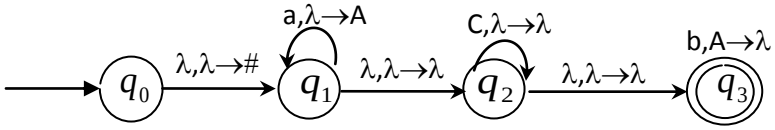
مثال۸۹. ماشین PDA برای پذیرش زبان $L = \{a^n b^n : n > 0\}$ که در مثال ۸۸ بصورت نمای گردشی نشان داده شده است بصورت دیاگرام تغییر وضعیت نشان دهید.

جواب.



دستور $\lambda,\lambda\rightarrow\#$ یعنی قرار دادن نماد # در انتهای پشته و ورود به وضعیت q_1 .

مثال۹۰. یک ماشین پشته‌ای طراحی کنید که زبان $\{a^i c^j b^i \mid i, j \geq 0\}$ را به پذیرد.



تعریف رسمی اتوماتای پشته‌ای

یک PDA را می‌توان با یک ۷ تائی $M = (Q, \Sigma, S, \delta, q_0, z, F)$ با مشخصات زیر تعریف کرد:

Q : مجموعه متناهی از وضعیتهای درونی واحد کنترل.

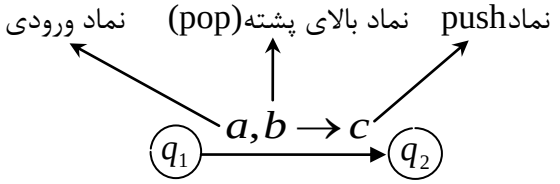
Σ : الفباء ورودی.

S : الفباء پشته.

δ : تابع تغییر وضعیت که زیرمجموعه متناهی است از $(Q \times (\Sigma \cup \{\lambda\}) \times S^*) \times (Q \times S^*)$

$q_0 \in Q$: وضعیت آغازی. $Z \in S$: نماد شروع پشته. $F \subseteq Q$: مجموعه وضعیتهای نهائی.

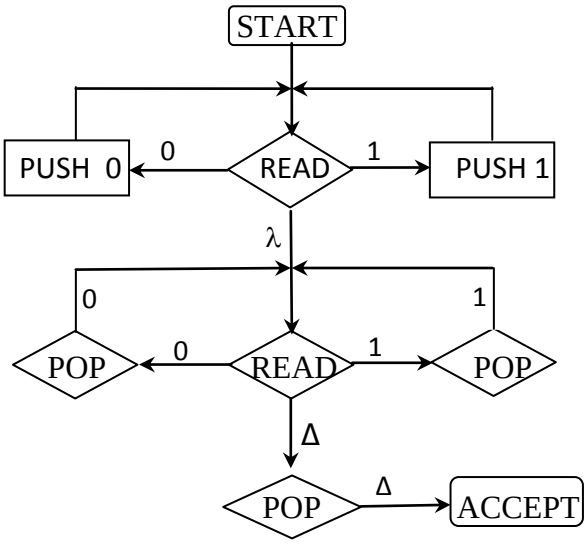
در شکل زیر تغییر وضعیت یک PDA از وضعیت q_1 به q_2 به صورت $a,b \to c$ نشان داده شده است.



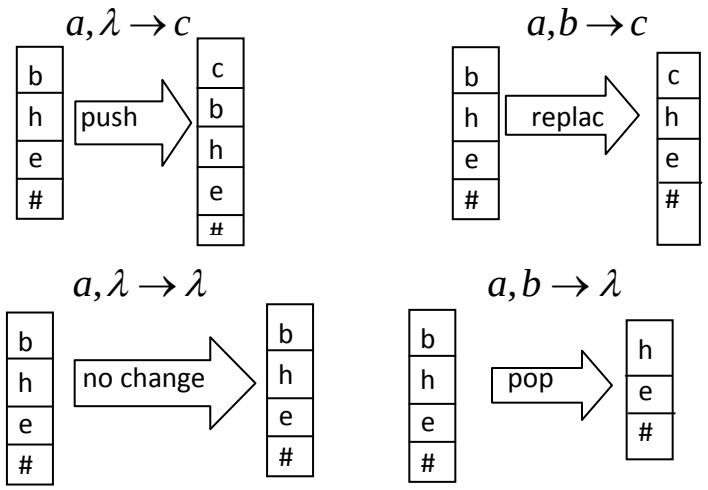
شکل بالا نشان دهنده این است که در وضعیت q_1 اگر بانماد ورودی a مواجه شده و نماد بالای پشته b باشد آنگاه b را pop , c را push و به q_2 تغییر وضعیت داده شده است. شکل فوق معادل عبارت $((q_1, a, b), (q_2, c)) \in \delta$ یا $\delta(q_1, a, b) = \{[q_2, c]\}$ بوده و به این معنی است که اگر ماشین M در وضعیت q_1 است و b در ابتدای (top) پشته بوده و ماشین نماد a را از نوار ورودی بخواند آنگاه c را با b جایگزین کرده و وارد و ضعیت q_2 می‌شود. به عبارت دیگر ابتدا عمل pop b و سپس عمل $push$ c انجام می‌گیرد. زوجهائی به صورت $((q_1, a, b), (q_2, c))$ را تغییر وضعیت ماشین M گویند. در تغییر وضعیت ارائه شده $q_1, q_2 \in Q$, $b \in S^*$, $c \in S^*$ و $a \in \Sigma \cup \{\lambda\}$.

توجه داشته باشید که ماشین M مورد نظر ماشین غیر قطعی یا به اختصار ماشین NPDA است به این معنی که بدون استفاده از ورودی نیز می‌تواند تغییر وضعیت دهد.

مثال۹۱. یک PDA بر روی $\Sigma = \{0,1\}$ طراحی کنید که رشته‌هائی به طول زوج به پذیرد که وارون رشته با خود رشته برابر باشد(even length palindromes) . PDA را هم به صورت نمای گردشی و هم به صورت دیاگرام تغییر وضعیت نشان دهید.



مثل۹۲. در یک PDA نماد ورودی a و پشته مورد نظر به صورت زیر است. عملکرد دستورات زیر را بر روی پشته داده شده به تفکیک توضیح داده و پس از اجرای هر دستور پشته جدید را با شکل نشان دهید.



توجه: در شروع فعالیت PDA, رشته ورودی در نوار ورودی درج شده, ماشین در وضعیت آغازی q_0 و هد خواندن در سمت چپ‌ترین خانه نوار تنظیم شده و پشته خالی است. رشته ورودی X توسط ماشین پذیرفته می‌شود اگر ماشین در یک وضعیت نهائی متوقف و پشته خالی از نمادهای S باشد.

ماشین تورینگ (TM)

ماشین تورینگ هم مانند مانند اتوماتای پشته‌ای دارای یک انباره(storage) اضافی است. در PDA از یک پشته به عنوان یک انباره استفاده می‌شود و لی در TM از یک نوار که از هردو طرف بی انتها است استفاده می‌شود. نوار در TM دارای خانه‌هائی است که هر خانه می‌تواند یک نماد را در خود نگهداری کند.هد خواندن و نوشتن ماشین می‌تواند یک نماد را از نوار بخواند, یک نماد را در نوار بنویسد و سپس نوار را به اندازه یک خانه به چپ یا راست حرکت دهد. دو نوع TM وجود دارد قطعی و غیر قطعی. در این قسمت فقط نوع قطعی ماشین تورینگ را مورد بررسی قرار می‌دهیم. نوار آغازی ماشین دارای نمادهائی است که ماشین بعد شروع به کار بعضی یا تعدادی از آنهارا ممکن است بخواند یا هیچ‌کدام از نمادهارا نخواند. نوار آغازی ماشین را می‌توان به عنوان ورودی ماشین در نظر گرفت. بعد از اتمام کار ماشین نتیجه محاسبات اعم از پذیرفته شده یا رد شده به عنوان خروجی بر روی نوار می‌ماند.

تعریف ماشین تورینگ

یک ماشین تورینگ را می‌توان با یک ۷تائی به صورت $(Q,\Sigma,\Gamma,\delta,q_0,\#,F)$ با مشخصات زیر تعریف کرد.

Q: مجموعه متناهی از وضعیتها. Σ: الفباء ورودی.

Γ: مجموعه متناهی از نمادهای نوار(tape alphabet)

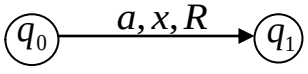
δ: تابع تغییر وضعیت با ضابطه‌ی $\delta : (Q\times\Gamma)\to(Q\times\Gamma)\times\{L,R,S\}$

$q_0\in Q$: وضعیت آغازی. #: نماد blank و $F\subseteq Q$: مجموعه وضعیتهای نهائی.

مثال۱. دستور $\delta:(q_0,a)\to(q_1,x,R)$ را به صورت جدول تغییر وضعیت و دیاگرام تغییر وضعیت نشان دهید.

جواب. جدول تغییر وضعیت. دیاگرام تغییر وضعیت.

	a
q_0	q_1xR



مثال۲. جدول را تفسیر کنید

	#	a	b
q_0	q_1aR	q_2L	$-$

✓. اگر ماشین در وضعیت q_0 بوده و frame خالی (blank) است آنگاه نماد a را در آن frame نوشته , به q_1 تغییر وضعیت داده و نوار را یک خانه به سمت راست حرکت دهد.

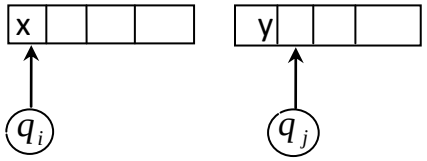
✓. اگر ماشین در وضعیت q_0 بوده و frame دارای نماد a است آنگاه به q_2 تغییر وضعیت داده بدون نوشتن و پاک کردن, نوار را یک خانه به سمت چپ حرکت دهد.

✓. اگر ماشین در وضعیت q_0 بوده و frame دارای نماد b است عملی انجام نگیرد(وضعیت HALT).

مثال۳. دستورالعمل $\delta(q_i, x) = [q_j, y, L]$ را تفسیر کرده و عملکرد ماشین را با شکل نشان دهید.

جواب.

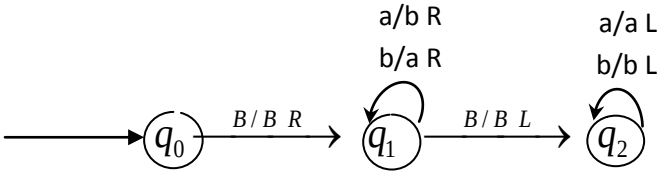
reads the x in the current cell, enters state q_j, writes a y in this cell, and moves one cell left



توجه. دستور $\delta(q_i, x) = [q_j, y, L]$ را با یک پنج تائی به صورت (q_i, x, q_j, y, L) نیز نشان می‌دهند. اگر تابع δ برای (q_i, x) تعریف نشده باشد آنگاه TM به وضعیت halt رسیده است.

مثال۴. تابع تغییر وضعیت یک ماشین تورینگ با الفباء ورودی {a,b} مطابق جدول زیر داده شده است. دیاگرام تغییر وضعیت ماشین را رسم کنید.

δ	B	a	b
q_0	q_1, B, R		
q_1	q_2, B, L	q_1, b, R	q_1, a, R
q_2		q_2, a, L	q_2, b, L



پیکربندی یک ماشین تورینگ شامل موارد زیر است:

✓ وضعیتی که در آن قرار دارد. ✓ محتویات نوار. ✓ موقعیت هد نوار بر روی نوار.

موارد گفته شده را می‌توان به صورت $x_i \cdots x_j q_m x_k \cdots x_j$ نشان داد که در آن x ها نمادهای روی نوار، q_m وضعیت فعلی، و هد نوار بر روی خانه‌ای که شامل x_k است (نماد بلافاصله بعد از q_m) قرار دارد. با استفاده از پیکربندی یک ماشین می‌توان محاسبات آن را ردیابی کرد. برای ارائه دو حرکت متوالی از نماد $\vdash_M$ استفاده می‌کنیم. به عنوان مثال عبارت $\delta(q_5, b) = (q_8, c, R)$ را با نماد $abbabq_5babb \vdash_M abbabcq_8abb$ نشان می‌دهیم. نماد $uq_ivB \vdash^* xq_jyB$ به این معنی است که xq_jyB را می‌توان از uq_ivB با استفاده از تعداد متناهی و امکانا به تعداد صفر از تابع تغییر وضعیت بدست آورد.

مثال۵. در ماشین تورینگ مطابق جدول زیر که نمادهای a و b را با هم تعویض می‌کند محاسبه ماشین را برای ورودی رشته $abab$ رهگیری کنید.

δ	B	a	b
q_0	q_1, B, R		
q_1	q_2, B, L	q_1, b, R	q_1, a, R
q_2		q_2, a, L	q_2, b, L

جواب.

$$(q_0 Babab B) \quad (Bq_1 abab B) \vdash_M (Bbq_1 bab B) \vdash_M (Bbaq_1 ab B) \vdash_M (Bbabq_1 b B) \vdash_M (Bbabaq_1 B) \vdash_M (Bbabq_2 aB) \vdash_M (Bbaq_2 baB) \vdash_M (Bbq_2 abaB) \vdash_M (Bq_2 babaB) \vdash_M (q_2 BbabaB) \vdash_M$$

مثال۶. ماشین تورینگ T با ۵تائی‌هائی به صورت زیر برای نوارهای آغازی به شکل زیر تعریف شده است. نوار نهائی را وقتی ماشین به حالت HALT رسید را مشخص کنید.

$$(q_0,0,q_1,0,R),\ (q_0,1,q_1,0,L),\ (q_0,B,q_1,1,R),\ (q_1,0,q_2,1,R),\ (q_1,1,q_1,1,R),\ (q_1,B,q_2,0,R),\ (q_2,B,q_3,0,R)$$

a)	…	B	B	0	1	0	1	B	…
b)	…	B	B	1	1	1	B	B	…
c)	…	B	0	0	B	0	0	B	…
d)	…	B	B	B	B	B	B	B	…

جواب. a: 0111 , b: 0011, c: 01000, d: 100

مسائل متفرقه

1. اگر $L=\{b,ba,bab\}$ و $L'=\{\lambda,b,bb,abb\}$ آنگاه

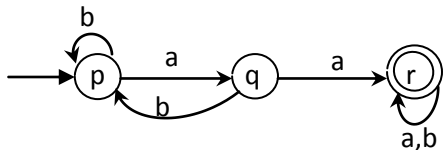
$$LL'=\{b,ba,bab\}\{\lambda,b,bb,abb\}$$
$$=\{b,ba,bb,bab,bbb,babb,baabb,bababb\}$$

۲. حاصل عمل* بر روی زبان $L=\{01\}$ را نشان دهید.

جواب. $\{01\}^*=\{\lambda,01,0101,010101,\ldots\}=\{(01)^n \mid n\geq 0\}$

۳. یک DFA طراحی کنید که بر روی {a,b} رشته‌هائی را بپذیرد که aa زیر رشته آن است.

δ	a	b
→ p	q	p
q	r	p
* r	r	r



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

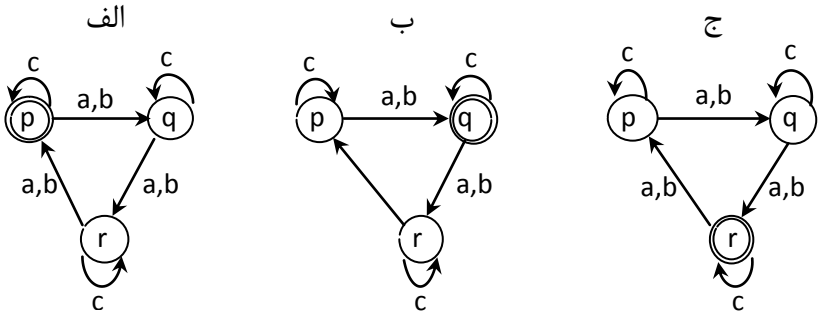
۴. یک DFA طراحی کنید که بر روی {a,b,c} رشته‌هائی را به پذیرد که حاصل جمع تعداد aها و تعداد bها در هر رشته واجد ویژگی زیر باشد.

الف. $L=\{x\in\{a,b,c\}^* \parallel x|_a+|x|_b\equiv 0 \mod(3)\}$

ب. $L=\{x\in\{a,b,c\}^* \parallel x|_a+|x|_b\equiv 1 \mod(3)\}$

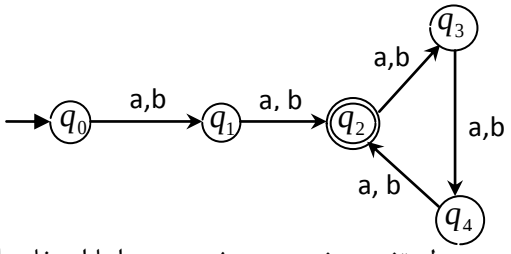
ج. $L=\{x\in\{a,b,c\}^* \parallel x|_a+|x|_b\equiv 2 \mod(3)\}$

جواب.



۵. زبان مفروض L بر روی $\{a,b\}$ زبانی است که طول رشته‌های آن متعلق به تصاعد حسابی به صورت $P=\{3,8,11,\ldots\}=\{2+3n \mid n\geq 0\}$ است. به عبارت دیگر $L=\{x\in\{a,b\}^* \mid |x|\in P\}$
یک DFA طراحی کنید که زبان مورد نظر را بپذیرد.

جواب. برای حل این مسئله ابتدا رشته‌های به طول ۲ را مورد توجه قرار داده سپس جهت پذیرش رشته هائی که طول آن مضربی از باشد از بند الف مثال ۴ در بالا استفاده می‌کنیم.

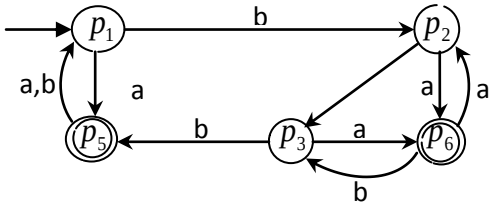


مثال۶ . DFA ارائه شده بصورت جدول تغییر وضعیت به شرح زیر را با استفاده از الگوریتم افراز وضعیت-ها مینی‌مایز کنید.(وضعیت نهائی با نماد * مشخص شده است)

δ	a	b
$\rightarrow q_0$	q_3	q_2
q_1	q_6	q_2
q_2	q_8	q_5
$*\ q_3$	q_0	q_1
$*\ q_4$	q_2	q_5
q_5	q_4	q_3
$*\ q_6$	q_4	q_3
q_7	q_4	q_6
$*\ q_8$	q_2	q_7
q_9	q_7	q_{10}
q_{10}	q_5	q_9

$$\Pi_0 = \{\{q_0,q_1,q_2,q_5,q_7,q_9,q_{10}\},\{q_3,q_4,q_6,q_8\}\}$$
$$\Pi_1 = \{\{q_0,q_1,q_2\},\{q_5,q_7\},\{q_9,q_{10}\},\{q_3,q_4,q_6,q_8\}\}$$
$$\Pi_2 = \{\{q_0,q_1,q_2\},\{q_5,q_7\},\{q_9,q_{10}\},\{q_3,q_6\},\{q_4,q_8\}\}$$
$$\Pi_3 = \{\{q_0,q_1\},\{q_2\},\{q_5,q_7\},\{q_9,q_{10}\},\{q_3,q_6\},\{q_4,q_8\}\}$$
$$\Pi_4 = \{\underbrace{\{q_0,q_1\}}_{p_1},\underbrace{\{q_2\}}_{p_2},\underbrace{\{q_5,q_7\}}_{p_3},\underbrace{\{q_9,q_{10}\}}_{p_4},\underbrace{\{q_3,q_6\}}_{p_5},\underbrace{\{q_4,q_8\}}_{p_6}\}$$

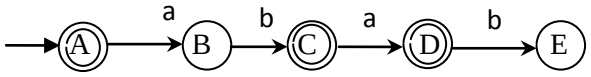
چون $\Pi_3 = \Pi_4$ پس پایان الگوریتم. DFA بهینه به صورت زیر است. وضعیت p_4 از وضعیت آغازی قابل دسترس نیست و از دیاگرام حذف است.



مثال۷. DFA ارائه شده بصورت جدول تغییر وضعیت به شرح زیر را با استفاده از الگوریتم افراز وضعیت‌ها مینی‌مایز کنید.

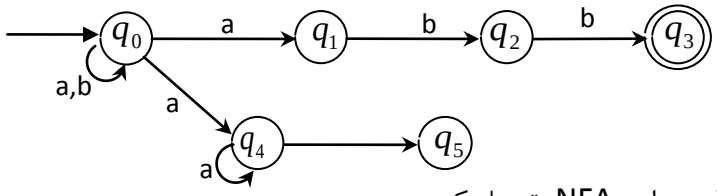
δ	a	b
q_0	q_1	q_0
q_1	q_0	q_3
q_2	q_4	q_5
q_3	q_4	q_1
q_4	q_2	q_6
q_5	q_0	q_2
q_6	q_3	q_4

$$\Pi_0 = \{\{q_0,q_2,q_3,q_4\},\{q_1,q_5,q_6\}\}$$
$$\Pi_1 = \{\{q_0\},\{q_2,q_3,q_4\},\{q_1,q_5,q_6\}\}$$
$$\Pi_2 = \{\{q_0\},\{q_2,q_3,q_4\},\{q_1,q_5\},\{q_6\}\}$$
$$\Pi_3 = \{\{q_0\},\{q_2,q_3\},\{q_4\},\{q_1,q_5\},\{q_6\}\}$$
$$\Pi_4 = \{\underbrace{\{q_0\}}_{A},\underbrace{\{q_2,q_3\}}_{C},\underbrace{\{q_4\}}_{D},\underbrace{\{q_1,q_5\}}_{B},\underbrace{\{q_6\}}_{E}\}$$

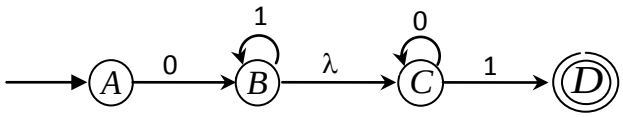


مثال۸. یک ماشین طراحی کنید که بر روی $\{a,b\}$ زبان $L(M)=(a+b)^*(abb+a^+b)$ را تولید کند.

جواب.



مثال۹. λ-NFA زیر را به NFA تبدیل کنید.



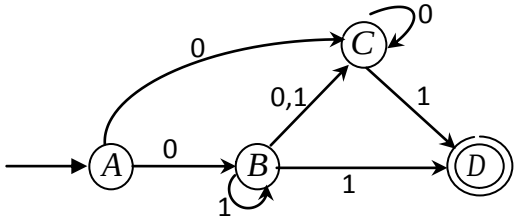
جواب.

	λ^*	0	λ^*
A	A	B	**B,C**
B	B,C	∅,C	∅,**C**
C	C	C	**C**
D	D	∅	∅

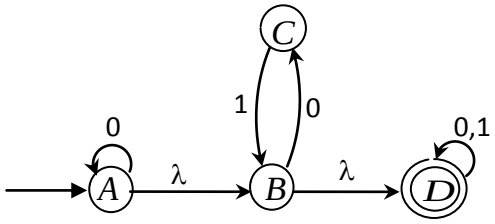
	λ^*	1	λ^*
A	A	∅	∅
B	B,C	B,D	**B,C,D**
C	C	D	**D**
D	D	∅	∅

	0	1
A	{B,C}	∅
B	{C}	{B,C,D}
C	{C}	{D}
D	∅	∅

دیاگرام NFA .



مثال۱۰. λ-NFA زیر را به NFA تبدیل کنید.



جواب.

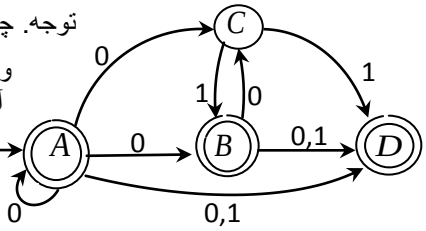
	λ^*	0	λ^*
A	A,B,D	A,C ,D	A,B, C,D
B	B,D	C,D	C,D
C	C	∅	∅
D	D	D	D

	λ^*	1	λ^*
A	A,B,D	D	D
B	B,D	∅,D	D
C	C	B	B,D
D	D	D	D

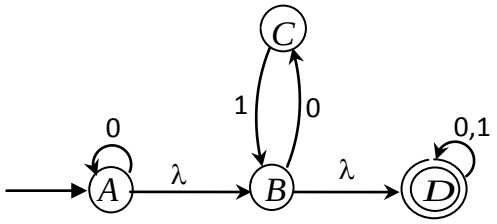
	0	1
A	{A,B,C,D}	∅
B	{C}	{B,C,D}
C	{C}	{D}
D	∅	∅

دیاگرام NFA

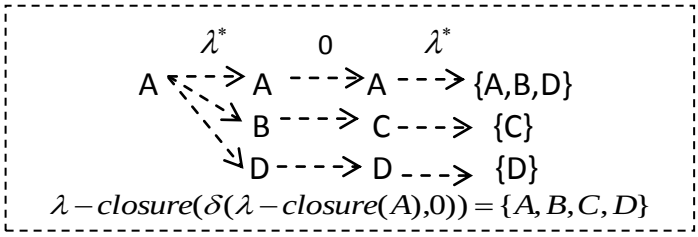
توجه. چون وضعیت D یک وضعیت نهائی است وضعیت‌های A و B نیز که وضعیت D از آنها قابل دسترس است نیز وضعیت‌های نهائی خواهند بود.



مثال۱۰. λ-NFA زیر را به NFA تبدیل کنید



با فرض $\Sigma = \{0,1\}$ و A یکی از وضعیتها باشد ابتدا $\lambda - closure(\delta(\lambda - closure(A),0))$ و $\lambda - closure(\delta(\lambda - closure(A),1))$ را پیدا کرده و این عمل را برای کلیه وضعیتها اعمال و نتیجه را در یک جدول درج می‌کنیم. به عنوان مثال با توجه به شکل بالا



جواب

	λ^*	0	λ^*	1	λ^*
A	A,B,D	A,C ,D	{A,B, C,D}	D	{D}
B	B,D	C,D	{C,D}	D	{D}
C	C	∅	∅	B	{B,D}
D	D	D	D	D	{D}

	0	1
A	{A,B,C,D}	{D}
B	{C,D}	{D}
C	∅	{B,D}
D	{D}	∅

دیاگرام NFA

توجه. چون وضعیت D یک وضعیت نهائی است وضعیت‌های A وB نیز که وضعیت D از آنها قابل دسترس است نیز وضعیت‌های نهائی خواهند بود.