نظریه‌ی زبان‌ها و ماشین‌ها

درس‌نامه‌ی ۷

کاظم فولادی
http://kazim.fouladi.ir

ویراست اول: ۱۳۸۵
ویراست دوم: ۱۳۸۷
ویراست سوم: ۱۳۹۳

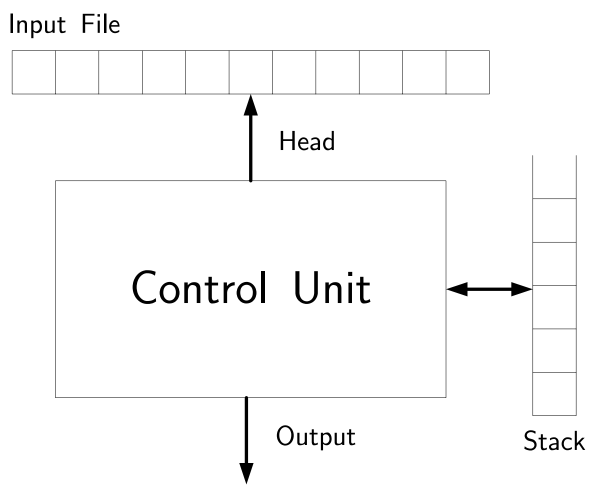
فهرست مطالب

۷ آتوماتای پشته‌ای

PUSHDOWN AUTOMATA

۷-۱ آتوماتای پشته‌ای غیر قطعی

در این درس پذیرنده‌ی زبان‌های مستقل از متن را معرفی می‌کنیم.
این پذیرنده، مشابه FA است با این تفاوت که حافظه‌ی پشته به ساختار ماشین اضافه می‌شود.



- هر حرکت واحد کنترل، یک حرف از نوار ورودی می‌خواند و محتوای پشته نیز با عملیات معمول پشته (push، pop) تغییر می‌کند.
- هر حرکت واحد کنترل بر اساس ورودی فعلی و نماد فعلی بالای پشته تعیین می‌شود.
- نتیجه‌ی هر حرکت یک حالت جدید برای واحد کنترل و تغییر در بالای پشته است.

۷-۱-۱ تعریف آتوماتون پشته‌ای غیر قطعی

تعریف آتوماتون پشته‌ای غیر قطعی **(non-deterministic pushdown accepter: NPDA)**.
یک پذیرنده‌ی پشته‌ای غیر قطعی به صورت زیر تعریف می‌شود:

$$M = (Q, \Sigma, \Gamma, \delta, q_{\circ}, z, F)$$

که در آن
Q: مجموعه‌ای متناهی از **حالات** واحد کنترل

Σ: **الفبای ورودی**
Γ: **الفبای پشته**
δ: **تابع گذر** که به صورت زیر تعریف می‌شود:

$$Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to \text{a finite subset of } (Q \times \Gamma^*)$$

q_0: **حالت شروع** واحد کنترل که $q_0 \in Q$
z: **علامت شروع پشته** که $z \in \Gamma$
F: **مجموعه‌ی حالات نهایی** که $F \subseteq Q$

ملاحظاتی در مورد دامنه و برد تابع δ

- آرگومان‌های تابع گذر، δ، حالت فعلی واحد کنترل، حرف ورودی فعلی و علامت بالای پشته به صورت سه‌تایی (q, a, α) است.
- ورودی فعلی a می‌تواند λ باشد (حرکت بدون مصرف ورودی، گذر λ)
- اگر پشته خالی باشد، یعنی $\alpha = \lambda$، حرکت ممکن نیست.
- برد δ، مجموعه‌ای از زوج‌ها به صورت (q, β) است که در آن:
  q حالت بعدی و
  β رشته‌ای است که به جای حرف فعلی بالای پشته می‌نشیند.

مثال

$$\delta(q_1, a, b) = \{(q_2, cd), (q_3, \lambda)\}$$

هرگاه واحد کنترل در حالت q_1، نماد ورودی a و نماد بالای پشته b باشد، آنگاه یکی از دو حالت زیر اتفاق می‌افتد:

۱) واحد کنترل به حالت q_2 می‌رود و رشته‌ی cd جایگزین b در بالای پشته می‌شود.

۲) واحد کنترل به حالت q_3 می‌رود و حرف b از بالای پشته حذف می‌شود.

قرار دادن یک رشته در داخل پشته حرف به حرف انجام می‌شود و از سمت راست رشته شروع می‌شود.

نکته اگر برای یک آرگومان δ مقداری مشخص نشده باشد، NPDA در آنجا متوقف می‌شود و رشته‌ی ورودی پذیرفته نمی‌شود.

مثال

NPDAی M به صورت $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ با

$$
\begin{aligned}
Q &= \{q_0, q_1, q_2, q_3\} \\
\Sigma &= \{a, b\} \\
\Gamma &= \{0, 1\} \\
z &= 0 \\
F &= \{q_3\}
\end{aligned}
$$

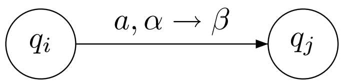
و تابع گذر

$$
\begin{aligned}
\delta(q_0, a, 0) &= \{(q_1, 10), (q_3, \lambda)\} \\
\delta(q_0, \lambda, 0) &= \{(q_3, \lambda)\} \\
\delta(q_1, a, 1) &= \{(q_1, 11)\} && \text{افزودن یک ۱ به پشته در صورت مشاهده‌ی یک } a \\
\delta(q_1, b, 1) &= \{(q_2, \lambda)\} && \text{حذف یک ۱ از پشته و ورود به حالت بعدی در صورت مشاهده‌ی اولین } b \\
\delta(q_2, b, 1) &= \{(q_2, \lambda)\} && \text{حذف یک ۱ از پشته در صورت مشاهده‌ی یک } b \\
\delta(q_2, \lambda, 0) &= \{(q_3, \lambda)\}
\end{aligned}
$$

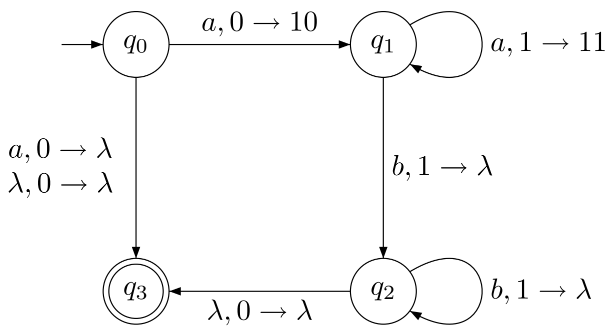
زبان $L = \{a^n b^n : n \geq 0\} \cup \{a\}$ را می‌پذیرد.

بازنمایی یک NPDA با دیاگرام گذر حالت نیز انجام می‌شود:

$$\delta(q_i, a, \alpha) = (q_j, \beta)$$



برای مثال، دیاگرام گذر حالت برای NPDA مثال فوق به صورت زیر خواهد بود:



تعریف یک توصیف بلافصل پیکربندی *(instantaneuos description of configuration)* برای یک آتوماتون پشته‌ای، یک سه‌تایی به صورت

$$(q, w, u)$$

است که در آن
q: حالت واحد کنترل
w: باقیمانده‌ی رشته‌ی ورودی
u: محتوای پشته
است.

تعریف **حرکت.** حرکت از یک توصیف بلافصل پیکربندی به دیگری به صورت زیر نشان داده می‌شود:

$$(q_1, aw, bx) \vdash (q_2, w, yx) \quad \textbf{iff} \quad (q_2, y) \in \delta(q_1, a, b)$$

از نمادگذاری زیر استفاده می‌شود:

$\vdash^*$	حرکت در صفر یا چند قدم
$\vdash^+$	حرکت در یک یا چند قدم
$\vdash_M$	حرکت توسط آتوماتون M

۷-۱-۲ زبان پذیرفته شده توسط یک آتوماتون پشته‌ای

تعریف اگر $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ یک آتوماتای پشته‌ای غیر قطعی *(NPDA)* باشد، زبان پذیرفته شده توسط M عبارت است از:

$$L(M) = \{w \in \Sigma^* : \exists p \in F \ (q_0, w, z) \vdash_M^* (p, \lambda, u), u \in \Gamma^*\}$$

یعنی، زبان پذیرفته شده توسط M مجموعه‌ی تمام رشته‌هایی است که می‌توانند در پایان خود، M را به یک حالت نهایی ببرند؛ «محتوای پشته در انتهای پذیرش اهمیتی ندارد [تعریف کتاب]».

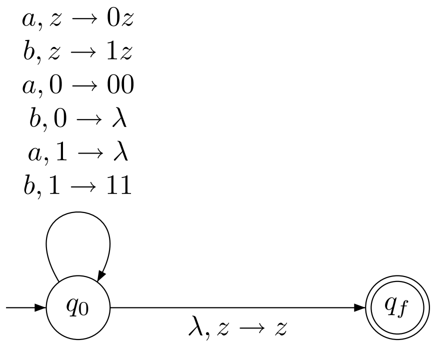
مثال

NPDAی M به صورت $M = (\{q_0, q_f\}, \{a, b\}, \{0, 1, z\}, \delta, q_0, z, \{q_f\})$ برای پذیرش زبان

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

با تابع گذر زیر تعریف می‌شود:

$$
\begin{aligned}
\delta(q_۰, \lambda, z) &= \{(q_f, z)\} && \text{پذیرش رشته‌ی تهی} \\
\delta(q_۰, a, z) &= \{(q_۰, ۰z)\} && \text{افزودن یک ۰ به پشته در صورت مشاهده‌ی } a \\
\delta(q_۰, b, z) &= \{(q_۰, ۱z)\} && \text{افزودن یک ۱ به پشته در صورت مشاهده‌ی } b \\
\delta(q_۰, a, ۰) &= \{(q_۰, ۰۰)\} \\
\delta(q_۰, b, ۰) &= \{(q_۰, \lambda)\} \\
\delta(q_۰, a, ۱) &= \{(q_۰, \lambda)\} \\
\delta(q_۰, b, ۱) &= \{(q_۰, ۱۱)\}
\end{aligned}
$$



حرکت‌های لازم برای پذیرش رشته‌ی $abab$ در این NPDA به صورت زیر است:

$$(q_۰, abab, z) \vdash (q_۰, bab, ۰z) \vdash (q_۰, ab, z) \vdash (q_۰, b, ۰z) \vdash (q_۰, \lambda, z) \vdash (q_f, \lambda, z)$$

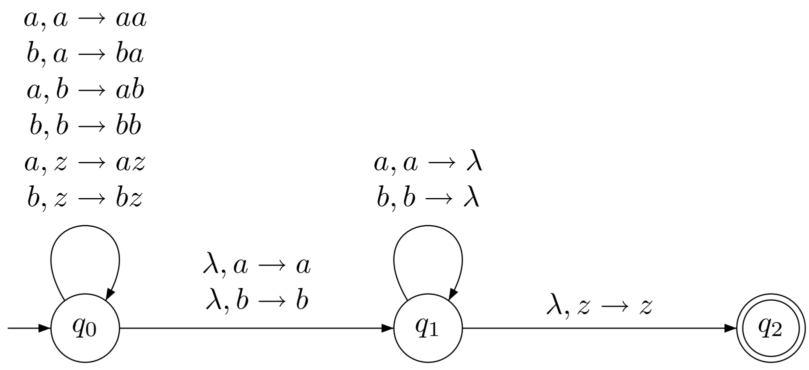
مثال

NPDAی M به صورت برای پذیرش زبان $M = (\{q_۰, q_۱, q_۲\}, \{a, b\}, \{a, b, z\}, \delta, q_۰, z, \{q_۲\})$

$$L = \{ww^R : w \in \{a, b\}^+\}$$

با تابع گذر زیر تعریف می‌شود:

$$
\begin{aligned}
\delta(q_۰, a, a) &= \{(q_۰, aa)\} \\
\delta(q_۰, b, a) &= \{(q_۰, ba)\} \\
\delta(q_۰, a, b) &= \{(q_۰, ab)\} \\
\delta(q_۰, b, b) &= \{(q_۰, bb)\} \\
\delta(q_۰, a, z) &= \{(q_۰, az)\} \\
\delta(q_۰, b, z) &= \{(q_۰, bz)\} \\
\delta(q_۰, \lambda, a) &= \{(q_۱, a)\} && \text{حدس زدن وسط رشته به صورت غیرقطعی} \\
\delta(q_۰, \lambda, b) &= \{(q_۱, b)\} && \text{حدس زدن وسط رشته به صورت غیرقطعی} \\
\delta(q_۱, a, a) &= \{(q_۱, \lambda)\} \\
\delta(q_۱, b, b) &= \{(q_۱, \lambda)\} \\
\delta(q_۱, \lambda, z) &= \{(q_۲, z)\}
\end{aligned}
$$



▲

۲-۷ آتوماتای پشته‌ای و زبان‌های مستقل از متن

۱-۲-۷ آتوماتای پشته‌ای برای زبان‌های مستقل از متن

اگر G گرامر تولید کننده‌ی زبان مستقل از متن L در فرم نرمال گریباخ باشد، یک NPDA با سه حالت q_0 (حالت شروع)، q_1 (حالت میانی) و q_f (حالت نهایی) در نظر می‌گیریم.

- در ابتدای کار علامت شروع S به پشته انتقال می‌یابد.
- وقتی گرامر قاعده‌ای به شکل $X \to aY_1Y_2 \dots Y_n$ دارد، با خواندن a از ورودی، X از بالای پشته حذف می‌شود و به جای آن $Y_1Y_2 \dots Y_n$ قرار می‌گیرد.

$$\underbrace{X}_{\text{pop}} \to a\,\underbrace{Y_1Y_2 \dots Y_n}_{\text{push}}$$

- ظاهر شدن نماد شروع پشته در بالای پشته، به معنی پایان اشتقاق و قرارگرفتن آتوماتون پشته‌ای در حالت نهایی است.

قضیه به ازای هر زبان مستقل از متن L، یک $NPDA$ به نام M وجود دارد که $L = L(M)$.

ساخت NPDA از روی گرامر مربوط به زبان مستقل از متن. اگر $G = (V, T, S, P)$ یک گرامر مستقل از متن خالی از λ باشد، به ترتیب زیر عمل می‌کنیم:

- گرامر معادل به فرم گریباخ را می‌نویسیم.
- یک NPDA با سه حالت را به صورت زیر می‌سازیم:

$$M = (\{q_0, q_1, q_f\}, T, V \cup \{z\}, \delta, q_0, z, \{q_f\})$$

که در آن الفبای ورودی آتوماتون مجموعه‌ی پایانه‌های G و الفبای پشته مجموعه‌ی متغیرهای G و z می‌باشد.

- تابع گذر حالت را به صورت زیر می‌سازیم:

$$\delta(q_0, \lambda, z) = \{(q_1, Sz)\}$$

به ازای هر قاعده‌ی گرامر به شکل $A \rightarrow au$ که در آن $a \in T$ و $u \in V^*$،

$$(q_1, u) \in \delta(q_1, a, A)$$

و گذر

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

اگر $\lambda \in L(G)$، گذر زیر را می‌افزاییم:

$$(q_f, z) \in \delta(q_0, \lambda, z)$$

مثال

برای به دست آوردن یک NPDA برای پذیرش زبان تولید شده با گرامر

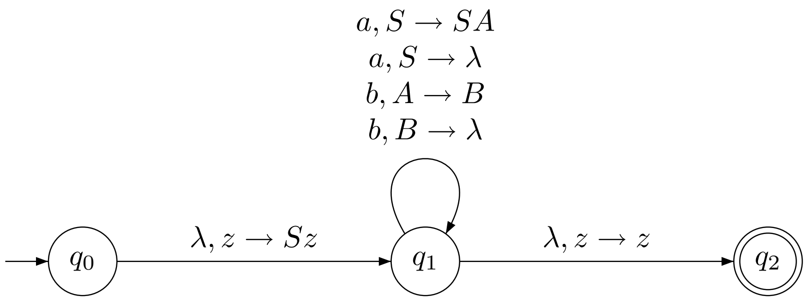
$$S \rightarrow aSbb \mid a$$

ابتدا گرامر را به فرم معادل نرمال گریباخ تبدیل می‌کنیم:

$$\begin{aligned} S &\rightarrow aSA \mid a \\ A &\rightarrow bB \\ B &\rightarrow b \end{aligned}$$

در این صورت NPDAی مورد نظر به صورت زیر خواهد بود:

$$\begin{aligned} \delta(q_0, \lambda, z) &= \{(q_1, Sz)\} \\ \delta(q_1, a, S) &= \{(q_1, SA), (q_1, \lambda)\} \\ \delta(q_1, b, A) &= \{(q_1, B)\} \\ \delta(q_1, b, B) &= \{(q_1, \lambda)\} \\ \delta(q_1, \lambda, z) &= \{(q_2, z)\} \end{aligned}$$



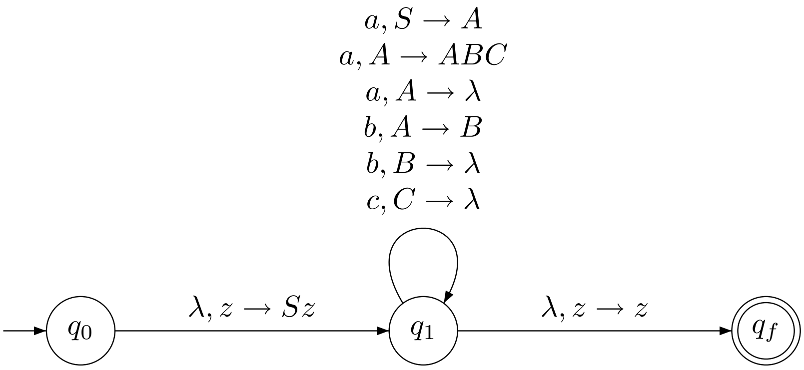
مثال

یک NPDA برای پذیرش زبان تولید شده با گرامر

$$\begin{array}{lcl} S & \rightarrow & aA \\ A & \rightarrow & aABC \mid bB \mid a \\ B & \rightarrow & b \\ C & \rightarrow & c \end{array}$$

به صورت زیر خواهد بود:

$$\begin{array}{lcl} \delta(q_0, \lambda, z) & = & \{(q_1, Sz)\} \\ \delta(q_1, a, S) & = & \{(q_1, A)\} \\ \delta(q_1, a, A) & = & \{(q_1, ABC), (q_1, \lambda)\} \\ \delta(q_1, b, A) & = & \{(q_1, B)\} \\ \delta(q_1, b, B) & = & \{(q_1, \lambda)\} \\ \delta(q_1, c, C) & = & \{(q_1, \lambda)\} \\ \delta(q_1, \lambda, z) & = & \{(q_f, z)\} \end{array}$$



◀ این NPDA اشتقاق‌های چپ‌ترین را برای گرامر مورد نظر شبیه‌سازی می‌کند.

◀ این شبیه‌سازی به گونه‌ای است که قسمت پردازش نشده‌ی فرم جمله‌ای درون پشته قرار دارد و به علاوه پیشوند پایانه‌ای هر فرم جمله‌ای با پیشوند مربوط در ورودی تطبیق پیدا می‌کند.

◀ لزومی ندارد که G حتماً در فرم نرمال گریباخ باشد، برای مثال:

– $A \rightarrow Bx$: A را از بالای پشته برمی‌داریم و بدون مصرف ورودی آن را با Bx جایگزین می‌کنیم.

– $A \rightarrow abCx$: ابتدا ab را در ورودی با رشته‌ی مشابه در پشته تطبیق می‌دهیم و بعد A را با Cx جایگزین می‌نماییم.

ساخت NPDA از روی گرامر مربوط به زبان مستقل از متن بدون نیاز به تبدیل به فرم گریباخ.
اگر $G = (V, T, S, P)$ یک گرامر مستقل از متن باشد، به ترتیب زیر عمل می‌کنیم:

- یک NPDA با سه حالت را به صورت زیر می‌سازیم:

$$M = (\{q_۰, q_۱, q_f\}, T, T \cup V \cup \{z\}, \delta, q_۰, z, \{q_f\})$$

که در آن الفبای ورودی آتوماتون مجموعه‌ی پایانه‌های G و الفبای پشته مجموعه‌ی پایانه‌ها و متغیرهای G و z می‌باشد.

- تابع گذر حالت را به صورت زیر می‌سازیم:

$$\delta(q_۰, \lambda, \lambda) = \{(q_۱, S)\}$$

به ازای هر قاعده‌ی گرامر به شکل $A \to x$ که در آن $A \in V$ و $x \in (V \cup T)^*$، گذر

$$(q_۱, \lambda) \in \delta(q_۱, A, x)$$

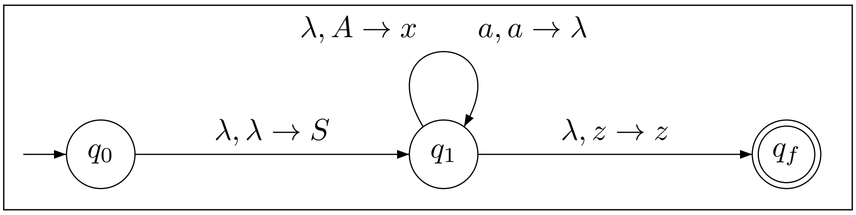
و به ازای هر پایانه‌ی گرامر $a \in T$ گذر

$$(q_۱, a) \in \delta(q_۱, a, \lambda)$$

و نیز گذر

$$\delta(q_۱, \lambda, z) = \{(q_f, z)\}$$

را می‌افزاییم.

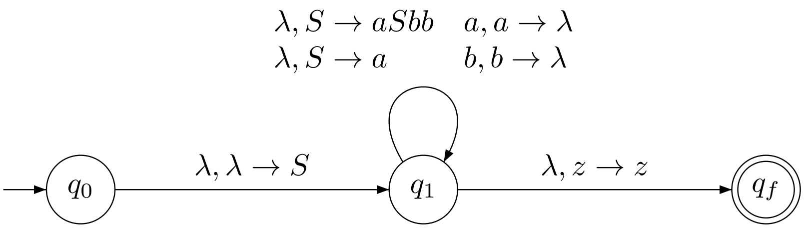


مثال

یک NPDA برای پذیرش زبان تولید شده با گرامر

$$S \to aSbb \mid a$$

به صورت زیر است:



۲-۲-۷ گرامرهای مستقل از متن برای آتوماتای پشته‌ای

برای یافتن گرامر مستقل از متن معادل با یک آتوماتون پشته‌ای، روال قبلی را معکوس می‌کنیم. در اینجا باید

- محتوای پشته در قسمت متغیر فرم جمله‌ای منعکس شده باشد.
- ورودی پردازش شده، پیشوند پایانه‌ای فرم جمله‌ای باشد.

	پردازش‌شده					
ورودی	a_1	a_2	a_3	a_4	a_5	a_6
فرم‌جمله‌ای	S_1	S_2	S_3	S_4	S_5	S_6
						پشته
						قسمت متغیر فرم جمله‌ای

فرض می‌کنیم که NPDA مورد بحث، $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$، شرایط زیر را دارد:

۱) فقط یک حالت نهایی q_f وجود دارد که تنها در صورت خالی بودن پشته وارد آن می‌شویم.

۲) همه‌ی گذر حالت‌ها باید به شکل $\delta(q_i, a, A) = \{c_1, c_2, \ldots, c_n\}$ باشد که در آن $c_i = (q_i, \lambda)$ یا $c_i = (q_j, BC)$ می‌باشد. یعنی هر حرکت محتوای پشته را یک حرف افزایش یا یک حرف کاهش می‌دهد

می‌خواهیم گرامری را بیابیم که متغیرهای آن به صورت (q_iAq_j) باشد و قواعد آن به گونه‌ای باشد که $(q_iAq_j) \Rightarrow^* v$ اگر و فقط اگر M با خواندن v و رفتن از q_i به q_j، A را از روی پشته پاک کند.[1]

$$q_i \xrightarrow[\text{clear } A]{v} q_j$$

$$(q_iAq_j) \Rightarrow^* v$$

در این گرامر با انتخاب متغیر (q_0zq_f) به عنوان نماد شروع خواهیم داشت:

$(q_0zq_f) \Rightarrow^* w$ اگر و فقط اگر M با خواندن w و رفتن از q_0 به q_f، پشته را خالی کند.

مراحل ساخت گرامر مستقل از متن از روی NPDA.

- اگر NPDA دارای گذری به شکل $(q_j, \lambda) \in \delta(q_i, a, A)$ باشد گرامر دارای قاعده‌ای به شکل

$$(q_iAq_j) \to a$$

خواهد بود.

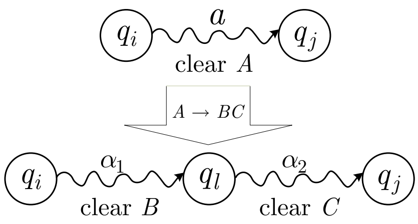
$$q_i \xrightarrow[\text{clear } A]{a} q_j$$

- اگر NPDA دارای گذری به شکل $(q_j, BC) \in \delta(q_i, a, A)$ باشد گرامر دارای قاعده‌ای به شکل

$$(q_iAq_k) \to a(q_jBq_l)(q_lCq_k)$$

خواهد بود که در آن q_l و q_k تمامی مقادیر ممکن در Q را می‌گیرند.

[1] پاک کردن، یعنی این که A و تاثیرات آن (مثلاً همه‌ی رشته‌هایی که با آنها جایگزین شده است) از پشته خارج شوند و نماد زیر A در بالا قرار گیرد.



زیرا وقتی a را می‌خوانیم، برای پاک کردن A هنگامی که از q_i به q_j می‌رویم،
ابتدا آن را با BC جایگزین می‌کنیم
سپس از q_j به q_l می‌رویم و B را پاک می‌کنیم
و پس از آن از q_l به q_k می‌رویم و C را پاک می‌کنیم.

قضیه اگر در یک $NPDA$ مانند M داشته باشیم $L = L(M)$، آن‌گاه L یک زبان مستقل از متن است.

◄ **تذکر** هر آتوماتون متناهی مانند یک NPDA است که از پشته استفاده نمی‌کند، بنابراین هر زبان منظم با یک NPDA قابل پذیرش است و بنابراین هر زبان منظم یک زبان مستقل از متن است.

مثال

NPDA با تابع گذر حالت

$$
\begin{aligned}
\delta(q_0, a, z) &= \{(q_0, Az)\} \\
\delta(q_0, a, A) &= \{(q_0, A)\} \\
\delta(q_0, b, A) &= \{(q_1, \lambda)\} \\
\delta(q_1, \lambda, z) &= \{(q_2, \lambda)\}
\end{aligned}
$$

را داریم. ابتدا با معرفی حالت جدید q_3 شرط دوم را نیز برقرار می‌سازیم: ابتدا A را از پشته حذف می‌کنیم و سپس آن را در حرکت بعدی جایگزین می‌سازیم.

$$
\begin{aligned}
\delta(q_0, a, z) &= \{(q_0, Az)\} \\
\delta(q_0, a, A) &= \{(q_3, \lambda)\} \\
\delta(q_3, \lambda, z) &= \{(q_0, Az)\} \\
\delta(q_0, b, A) &= \{(q_1, \lambda)\} \\
\delta(q_1, \lambda, z) &= \{(q_2, \lambda)\}
\end{aligned}
$$

گرامر معادل زیر تبدیل می‌شود:

$$
\begin{array}{lll}
(q_0Aq_3) & \rightarrow & a \\
(q_0Aq_1) & \rightarrow & b \\
(q_1zq_2) & \rightarrow & \lambda \\
(q_0zq_0) & \rightarrow & a(q_0Aq_0)(q_0zq_0) \mid a(q_0Aq_1)(q_1zq_0) \mid a(q_0Aq_2)(q_2zq_0) \mid a(q_0Aq_3)(q_3zq_0) \\
(q_0zq_1) & \rightarrow & a(q_0Aq_0)(q_0zq_1) \mid a(q_0Aq_1)(q_1zq_1) \mid a(q_0Aq_2)(q_2zq_1) \mid a(q_0Aq_3)(q_3zq_1) \\
(q_0zq_2) & \rightarrow & a(q_0Aq_0)(q_0zq_2) \mid a(q_0Aq_1)(q_1zq_2) \mid a(q_0Aq_2)(q_2zq_2) \mid a(q_0Aq_3)(q_3zq_2) \\
(q_0zq_3) & \rightarrow & a(q_0Aq_0)(q_0zq_3) \mid a(q_0Aq_1)(q_1zq_3) \mid a(q_0Aq_2)(q_2zq_3) \mid a(q_0Aq_3)(q_3zq_3) \\
(q_3zq_0) & \rightarrow & (q_0Aq_0)(q_0zq_0) \mid (q_0Aq_1)(q_1zq_0) \mid (q_0Aq_2)(q_2zq_0) \mid (q_0Aq_3)(q_3zq_0) \\
(q_3zq_1) & \rightarrow & (q_0Aq_0)(q_0zq_1) \mid (q_0Aq_1)(q_1zq_1) \mid (q_0Aq_2)(q_2zq_1) \mid (q_0Aq_3)(q_3zq_1) \\
(q_3zq_2) & \rightarrow & (q_0Aq_0)(q_0zq_2) \mid (q_0Aq_1)(q_1zq_2) \mid (q_0Aq_2)(q_2zq_2) \mid (q_0Aq_3)(q_3zq_2) \\
(q_3zq_3) & \rightarrow & (q_0Aq_0)(q_0zq_3) \mid (q_0Aq_1)(q_1zq_3) \mid (q_0Aq_2)(q_2zq_3) \mid (q_0Aq_3)(q_3zq_3)
\end{array}
$$

▲

۳-۷ آتوماتای پشته‌ای قطعی و زبان‌های مستقل از متن قطعی

پذیرنده‌ی پشته‌ای قطعی (deterministic pushdown accepter: DPDA) یک آتوماتون پشته‌ای است که در انجام حرکت‌هایش چند گزینه ندارد.

تعریف آتوماتون پشته‌ای $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ را در صورتی قطعی می‌گوییم که مطابق با تعریف $NPDA$ باشد و همچنین برای هر $q \in Q$، $a \in \Sigma \cup \{\lambda\}$ و $b \in \Gamma$

۱) $\delta(q, a, b)$ حداکثر دارای یک عنصر باشد.

۲) اگر $\delta(q, \lambda, b)$ تهی نباشد، آن‌گاه $\delta(q, c, b)$ برای هر $c \in \Sigma$ تهی باشد.

شرط (۱): به ازای هر پیکربندی فقط یک حرکت قابل انجام است.
شرط (۲): هرگاه حرکت λ ممکن می‌شود، آن‌گاه برای آن پیکربندی هیچ حرکتی با مصرف ورودی نباید ممکن باشد. (برعکس DFA می‌توانیم حرکت λ داشته باشیم و در نتیجه وجود حرکت λ به معنی عدم قطعیت نیست.)

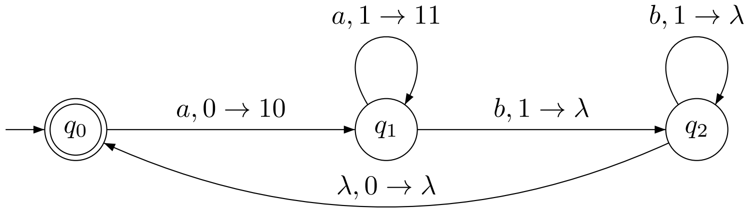
تعریف زبان L را مستقل از متن قطعی گوییم هرگاه برای آن یک $DPDA$ با نام M موجود باشد که $L = L(M)$.

برخلاف آتوماتای متناهی، آتوماتای پشته‌ای قطعی و غیرقطعی هم‌ارز نیستند،
بنابراین زبان‌های مستقل از متنی وجود دارند که قطعی نیستند.

مثال ▼

زبان $L = \{a^nb^n : n \geq 0\}$ یک زبان مستقل از متن قطعی است، زیرا DPDAی زیر برای آن وجود دارد:

$$
\begin{aligned}
\delta(q_۰, a, ۰) &= \{(q_۱, ۱۰)\} \\
\delta(q_۱, a, ۱) &= \{(q_۱, ۱۱)\} \\
\delta(q_۱, b, ۱) &= \{(q_۲, \lambda)\} \\
\delta(q_۲, b, ۱) &= \{(q_۲, \lambda)\} \\
\delta(q_۲, \lambda, ۰) &= \{(q_۰, \lambda)\}
\end{aligned}
$$



مثال

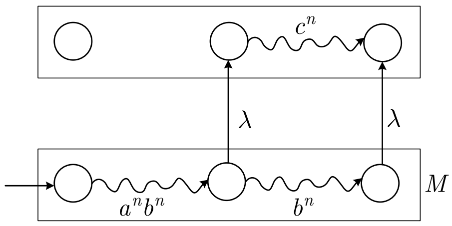
زبان $L = \{ww^R : w \in \{a, b\}^+\}$ یک زبان مستقل از متن قطعی نیست.

زیرا تشخیص وسط رشته به صورت قطعی با PDA ممکن نیست.

خانواده‌ی زبان‌های مستقل از متن قطعی و خانواده‌ی زبان‌های مستقل از متن غیر قطعی معادل نیستند.

مثال

زبان $L = \{a^n b^n : n \geq ۰\} \cup \{a^n b^{۲n} : n \geq ۰\}$ یک زبان مستقل از متن قطعی نیست.



$\hat{M}$

زبان L مستقل از متن است، زیرا اجتماع دو زبان مستقل از متن زیر است:

$$L_۱ = \{a^n b^n : n \geq ۰\}, \qquad L_۲ = \{a^n b^{۲n} : n \geq ۰\}$$

اما قطعی نیست. زیرا PDA در مقابل هر a باید یک یا دو a را مطابقت دهد، سپس باید در ابتدا تعیین کند که رشته‌ی ورودی در $L_۱$ یا $L_۲$ است، اما برای تصمیم‌گیری به طور قطعی اطلاعاتی در ابتدای کار وجود ندارد.

اگر L یک زبان مستقل از متن باشد، آنگاه زبان $\hat{L} = L \cup \{a^n b^n c^n : n \geq ۰\}$ مستقل از متن خواهد بود. این را با ساختن یک DPDA با نام $\hat{M}$ برای $\hat{L}$ نشان می‌دهیم.

برای این منظور، به واحد کنترل M قسمتی را اضافه می‌کنیم که تغییر حالت‌ها با ورودی b را به تغییر حالت با ورودی c تبدیل می‌نماید.
ورود به این قسمت پس از خواندن a^nb^n ممکن می‌شود.
چون فرایند قسمت دوم به c^n همانند b^n پاسخ می‌دهد، فرایند پذیرنده‌ی a^nb^n، $a^nb^nc^n$ را هم می‌پذیرد:

$$a^nb^n \in L(M) \Rightarrow (q_0, a^nb^n, z) \vdash_M^* (q_f, \lambda, u), \quad q_f \in F$$

چون M قطعی است، پس

$$(q_0, a^nb^{2n}, z) \vdash_M^* (q_f, b^n, u)$$

برای پذیرش a^nb^{2n} به این تغییر پیکربندی نیاز است:

$$(q_f, b^n, u) \vdash_M^* (q'_f, \lambda, u'), \quad q_f, q'_f \in F$$

همچنین بر اساس ساختار جدید داریم:

$$(\hat{q}_f, b^n, u) \vdash_{\hat{M}}^* (\hat{q}'_f, \lambda, u'), \quad \hat{q}_f, \hat{q}'_f \in F$$

می‌توان نشان داد که $\hat{L} = L(\hat{M})$ و در نتیجه $\hat{L}$ مستقل از متن است.
اما $\hat{L}$ مستقل از متن نیست (در فصل بعد ثابت می‌شود) و فرض ما مبنی بر قطعی بودن L نادرست است.

▲

مثال فوق نشان می‌دهد که زبان‌هایی وجود دارند که مستقل از متن هستند، اما مستقل از متن قطعی نیستند. پس خانواده‌ی زبان‌های مستقل از متن قطعی زیرمجموعه‌ی محضی از خانواده‌ی زبان‌های مستقل از متن است.

۴-۷ گرامرهای مربوط به زبان‌های مستقل از متن قطعی

اهمیت زبان‌های مستقل از متن قطعی، امکان تجزیه‌ی کارآمد آنها (بدون عقب‌گرد) است.
البته به دلیل وجود حرکات λ نمی‌توان بلافاصله ادعا کرد که تجزیه‌گرهای زمان خطی برای آنها وجود دارد.
برای مثال S-grammar برای تجزیه بسیار مناسب است، اما بیش از حد محدود است.

۱-۴-۷ گرامر LL

خاصیت گرامر LL این است که در هر گام تجزیه، با نگاه کردن به بخش محدودی از ورودی می‌توان پیشگویی کرد که از کدام قاعده‌ی گرامر بایستی استفاده شود.

L	**L**
↙	↘
اشتقاق‌های چپ‌ترین ساخته می‌شوند	ورودی از چپ به راست دریافت می‌شود

◄ **نکته** هر S-grammar یک گرامر LL است.

تعریف می‌گوییم یک گرامر $LL(k)$ است، اگر بتوانیم در هر مرحله از تجزیه با در دست داشتن نماد فعلی ورودی و با نگاه کردن به جلو بر روی $k-1$ نماد بعدی قاعده‌ی صحیحی را که باید استفاده شود، تشخیص داد.

مثال

گرامر $S \to aSb \mid ab$ یک گرامر LL(۲) است.

زیرا برای تعیین قاعده‌ی مورد استفاده در هر گام باید به دو نماد بعدی ورودی نگاه کنیم:

$aa\ldots$: قاعده‌ی $S \to aSb$

$ab\ldots$: قاعده‌ی $S \to ab$

مثال

گرامر $S \to SS \mid aSb \mid ab$ برای هر k یک گرامر LL(k) نیست.

این گرامر بستار مثبت زبان مورد اشاره در مثال قبلی را تولید می‌کند.

به رشته‌هایی به طول بزرگتر از ۲ نگاه می‌کنیم:

دو قاعده‌ی $S \to SS \mid aSb$ را داریم، اما از روی نماد جاری نمی‌توان قاعده‌ی درست را تشخیص داد. با نماد بعدی هم نمی‌توان این کار را انجام داد (مثلاً aa پیشوند $aabb$ و $aabbab$ و ... است.)

هر قدر هم که به جلو برویم، باز مواردی وجود دارد که قاعده‌ی صحیح بر اساس آن قابل تشخیص نیست (اثبات به استقرا روی k).

می‌توان گرامر فوق را به صورت $S \to aSbS \mid \lambda$ بازنویسی کرد و برای حذف λ از آن و معادل شدن آن با گرامر اصلی نوشت:

$$S' \to aSbS, \qquad S \to aSbS \mid \lambda$$

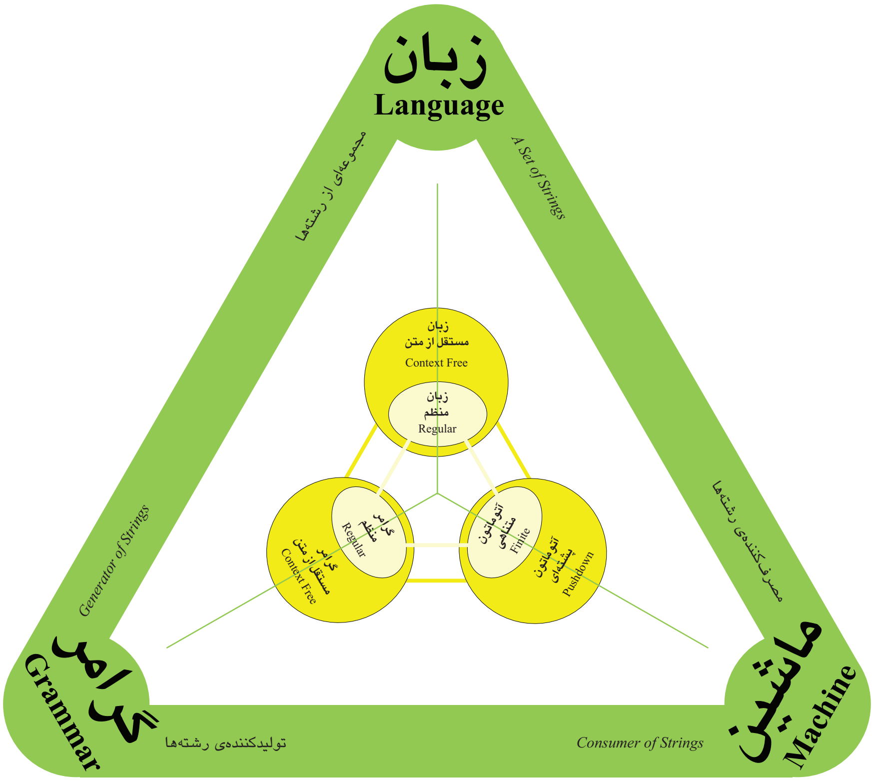
که این گرامر LL(۱) است.

بسیاری از زبان‌های برنامه‌سازی را می‌توان با گرامرهای LL تعریف نمود.

اما گرامرهای LL دارای عمومیت کافی برای کار با تمامی زبان‌های مستقل از متن قطعی نیستند.

گرامرهای LR قادر به پوشش کلیه‌ی زبان‌های مستقل از متن قطعی هستند و امکان تجزیه‌ی کاراتر را فراهم می‌سازند.

۵-۷ خانواده‌ی زبان‌های نوع دو: زبان، گرامر و ماشین



مراجع

[1] P. Linz, **An Introduction to Formal Languages and Automata**, 5th Ed., Jones and Bartletts, 2012.

[2] M. Sipser, **Introduction to the Theory of Computation**, 3rd Ed., Cengage Learning, 2013.