درس مبانی نظریه محاسبه

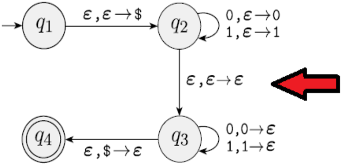
جلسه یازدهم

ماشین های پشته‌ای - ادامه

یک pda برای $\{ww^R \mid w \in \{0,1\}^*\}$

w = 011110

stack =	state = q1	input = 0
stack = $	state = q2	input = 0
stack = 0$	state = q2	input = 1
stack = 10$	state = q2	input = 1
stack = 110$	state = q2	input = 1
stack = 110$	state = q3	input = 1
stack = 10$	state = q3	input = 1
stack = 0$	state = q3	input = 0
stack = $	state = q3	input =
stack =	state = q4	input =



حدس وسط رشته

زبان یک ماشین پشته‌ای

توجه: ماشین پشته‌ای pda مانند nfa یک ماشین غیر قطعی است.

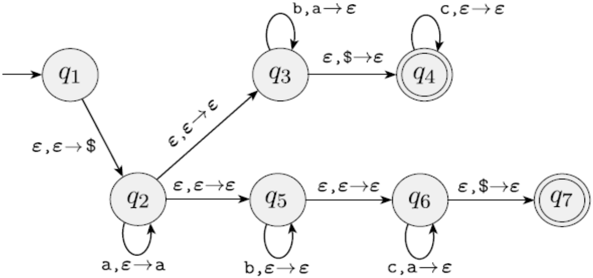
تعریف: ماشین پشته‌ای $M = (Q, \Sigma, \Gamma, \delta, q_1, F)$ داده شده است. رشته w توسط ماشین پشته‌ای M پذیرفته می‌شود اگر و فقط اگر برای رشته w یک مسیر از وضعیت شروع M تا یکی از وضعیتهای پذیرش M داشته باشیم. در این حالت گوییم که w عضو زبان M است.

توجه : ماشین باید کل رشته ورودی را بخواند و اگر پس از طی یکی از مسیرهای ممکن در یک وضعیت پذیرش قرار گرفت آنگاه رشته توسط ماشین پذیرفته شده است.

$w \in L(M)$ $\Leftrightarrow$ برای رشته w از وضعیت q_1 به یکی وضعیتهای داخل F یک مسیر وجود داشته باشد.

برای مثال، در ماشین پشته‌ای صفحه قبل، رشته 1001 عضو زبان است اما رشته 0011 عضو زبان نیست.

ماشین پشته‌ای: یک مثال دیگر



State diagram for PDA M_2 that recognizes

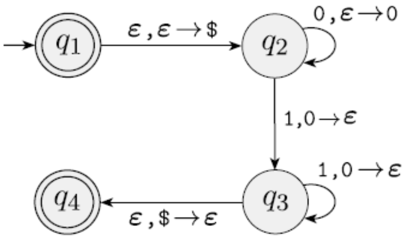
$$\{a^i b^j c^k \mid i,j,k \geq 0 \text{ and } i=j \text{ or } i=k\}$$

$$L(M_2) = \{a^n b^n c^* \mid n \geq 0\} \cup \{a^n b^* c^n \mid n \geq 0\}$$

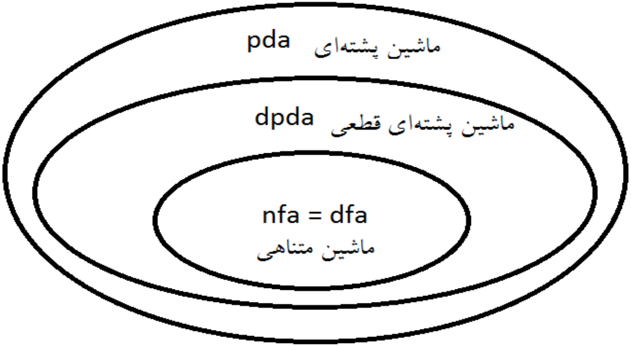
بر خلاف ماشینهای متناهی، عنصر عدم قطعیت در ماشینهای پشته‌ای اساسی و غیر قابل حذف است. به عبارت دیگر، بر خلاف ماشینهای متناهی که برای هر nfa یک معادل dfa وجود داشت، برای هر ماشین پشته‌ای غیرقطعی یک معادل قطعی وجود ندارد. برای مثال برای زبان ماشین پشته‌ای M_2 اسلاید قبل یک ماشین پشته‌ای قطعی وجود ندارد.

تعریف : ماشین پشته‌ای M را قطعی گوییم اگر برای هر رشته فقط یک مسیر محاسباتی در ماشین وجود داشته باشد. یک ماشین پشته‌ای قطعی را با نماد dpda نشان می‌دهیم.

برای مثال ماشین زیر یک ماشین پشته‌ای قطعی برای زبان $\{0^n1^n \mid n \geq 0\}$ است.



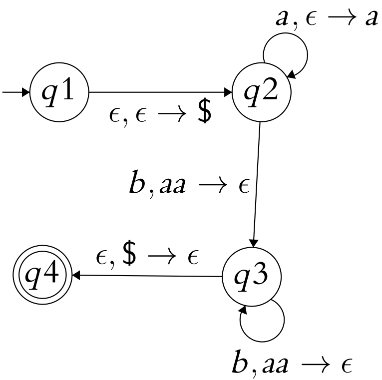
ماشینهای محاسباتی از لحاظ قدرت پذیرش



یک ماشین پشته‌ای برای زبان زیر طراحی کنید.

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

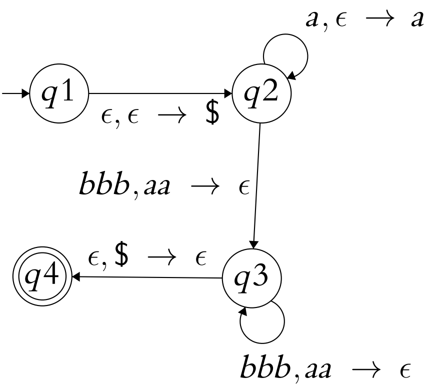
توضیح: ماشین همه a ها را در پشته قرار می‌دهد و سپس به ازای هر b دو تا a از بالای پشته برمی‌دارد.



یک ماشین پشته‌ای برای زبان زیر طراحی کنید.

$$L = \{a^{2n}b^{3n} \mid n \geq 1\}$$

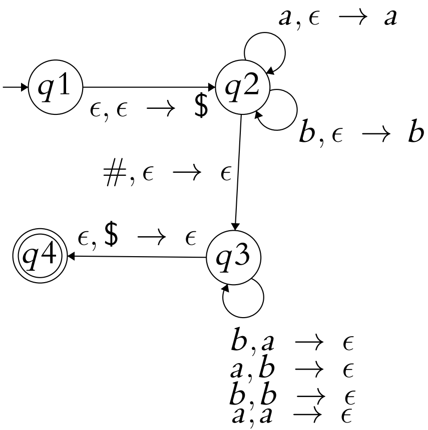
توضیح: ماشین همه a ها را در پشته قرار می‌دهد و پس از خواندن هر bbb در رشته ورودی یک aa از بالای پشته برمی‌دارد.



یک ماشین پشته‌ای برای زبان زیر طراحی کنید.

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

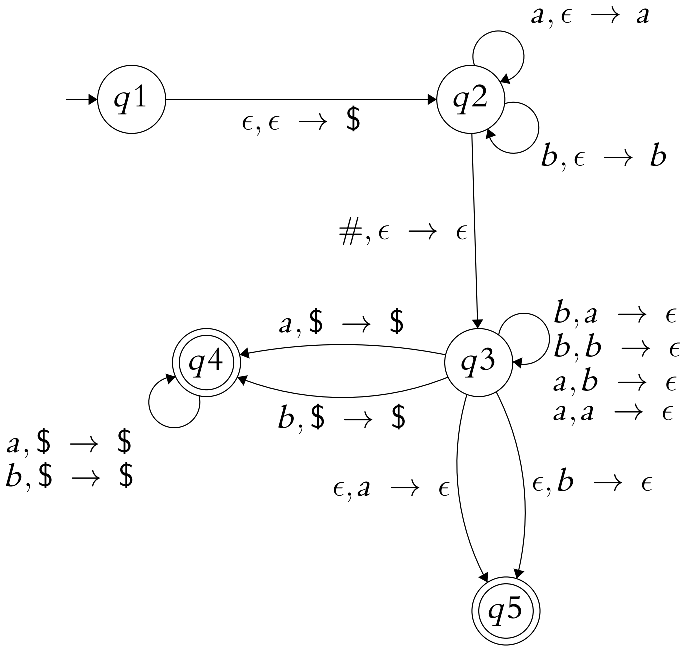
توضیح: ماشین همه کاراکترهای ورودی را تا رسیدن به علامت # در پشته قرار می‌دهد، سپس به ازای هر کارکتر ورودی یک کاراکتر از بالای پشته برمی‌دارد.



یک ماشین پشته‌ای برای زبان زیر طراحی کنید.

$$L = \{x\#y \mid x, y \in \{a, b\}^*, |x| \neq |y|\}$$

توضیح: ماشین همه کاراکترهای ورودی را تا رسیدن به علامت # در پشته قرار می‌دهد، سپس به ازای هر کارکتر ورودی یک کاراکتر از بالای پشته برمی‌دارد. اگر رشته تمام شد و پشته خالی نباشد به وضعیت q_5 میرود که یک وضعیت پذیرش است. همچنین اگر پشته خالی شد در حالیکه از رشته باقی مانده به وضعیت q_4 می‌رود که یک وضعیت پذیرش است.

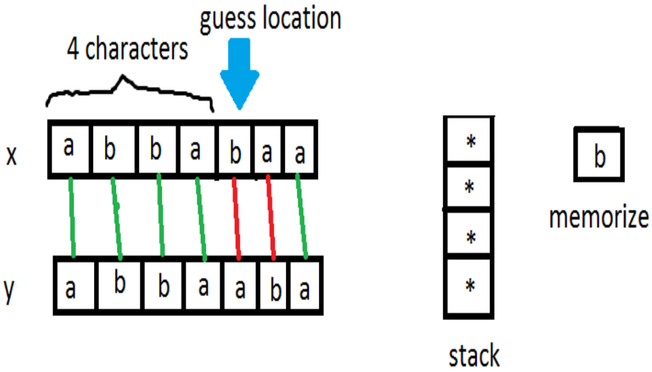

$q1$
$q2$
$q3$
$q4$
$q5$
$\epsilon, \epsilon \rightarrow \$$
$a, \epsilon \rightarrow a$
$b, \epsilon \rightarrow b$
$\#, \epsilon \rightarrow \epsilon$
$b, a \rightarrow \epsilon$
$b, b \rightarrow \epsilon$
$a, b \rightarrow \epsilon$
$a, a \rightarrow \epsilon$
$a, \$ \rightarrow \$$
$b, \$ \rightarrow \$$
$a, \$ \rightarrow \$$
$b, \$ \rightarrow \$$
$\epsilon, a \rightarrow \epsilon$
$\epsilon, b \rightarrow \epsilon$

تمرین: یک ماشین پشته‌ای برای زبان زیر طراحی کنید.

$$L = \{x\#y \mid x, y \in \{a,b\}^*, x \neq y$$ [redacted] $$\}$$

راهنمایی: اگر $x \neq y$ آنگاه وجود دارد i بطوریکه $x_i \neq y_i$.

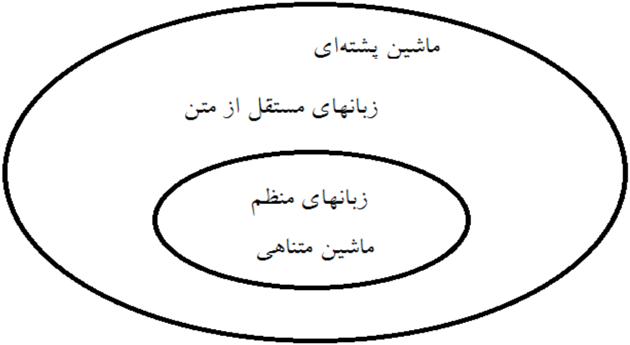
راهنمایی: از عدم قطعیت استفاده کنید.


guess location
4 characters
x
a b b a b a a
y
a b b a a b a
*
*
*
*
stack
b
memorize

ماشین پشته‌ای و زبانهای مستقل از متن

تعریف: زبان L را یک زبان مستقل از متن گوییم اگر و فقط اگر توسط یک ماشین پشته‌ای پذیرفته شود.

context free languages



ماشینهای پشته‌ای و گرامرهای مستقل از متن معادل هستند

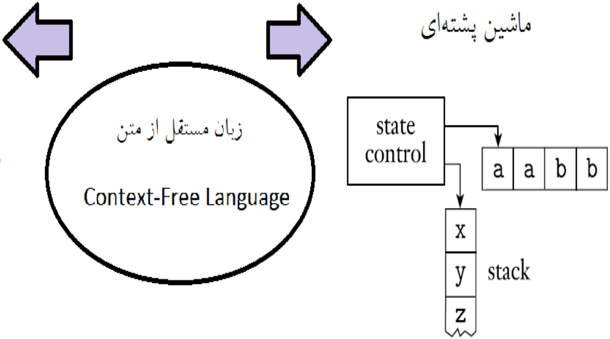
یادآوری: زبان A را مستقل از متن گوییم اگر گرامر مستقل از متن G وجود داشته باشد بطوریکه $L(G) = A$

قضیه: زبان A مستقل از متن است اگر و فقط اگر ماشین پشته‌ای M وجود داشته باشد بطوریکه $L(M) = A$

گرامر مستقل از متن

$$
\begin{aligned}
S_0 &\rightarrow AA_1 \mid UB \mid \mathrm{a} \mid SA \mid AS \\
S &\rightarrow AA_1 \mid UB \mid \mathrm{a} \mid SA \mid AS \\
A &\rightarrow \mathrm{b} \mid AA_1 \mid UB \mid \mathrm{a} \mid SA \mid AS \\
A_1 &\rightarrow SA \\
U &\rightarrow \mathrm{a} \\
B &\rightarrow \mathrm{b}
\end{aligned}
$$

ماشین پشته‌ای



قضیه اسلاید قبل را می‌توانیم در قالب دو لم زیر بنویسیم:

لم ۱ : اگر G یک گرامر مستقل از متن باشد آنگاه ماشین پشته‌ای M وجود دارد بطوریکه $L(M) = L(G)$

$$G \quad \Rightarrow \quad M$$

لم ۲ : اگر M یک ماشین پشته‌ای باشد آنگاه گرامر مستقل از متن G وجود دارد بطوریکه
$L(G) = L(M)$

$$M \quad \Rightarrow \quad G$$

اثبات لم ۱ آسانتر است و ما ایده اثبات آن را بطور مختصر ارائه می‌کنیم اما اثبات لم ۲ به آسانی لم ۱ نیست و نیاز به قدم ها و جزئیات بسیار دارد. اثبات این لم را می‌توانید در کتاب مرجع ببینید.

چگونه یک گرامر را به یک ماشین پشته‌ای تبدیل کنیم؟

$$
\begin{aligned}
S &\to ASA \mid \mathtt{a}B \\
A &\to B \mid S \\
B &\to \mathtt{b} \mid \varepsilon
\end{aligned}
$$

فرض کنید گرامر G داده شده است. می‌خواهیم ماشین پشته‌ای M را بسازیم بطوریکه اگر رشته w توسط گرامر G تولید شود، توسط ماشین M نیز پذیرفته شود. علاوه بر این اگر رشته w توسط گرامر G تولید نشود، توسط ماشین پشته‌ای M هم پذیرفته نشود.

از ابزار عدم قطعیت در ماشین پشته‌ای استفاده می‌کنیم. با داشتن قوانین جایگذاری گرامر G، ماشین پشته‌ای M در صورت لزوم هر بار قانون جایگذاری بعدی را حدس می‌زند و اگر این حدسها منجر به تولید رشته ورودی شد آنگاه توسط ماشین M هم پذیرفته می‌شود.

طرز کار ماشین M را می‌توان بصورت زیر خلاصه کرد:

- رشته ورودی w روی نوار input قرار گرفته است.
- متغیر شروع را در پشته قرار می‌دهیم و سپس پروسه زیر را تکرار می‌کنیم:
- اگر حرف بالای پشته یک حرف الفبا باشد، آن را با حرف روی نوار input مطابقت می‌دهیم. اگر یکسان بودند، حرف بالای پشته را برمی‌داریم و نوک خواندن روی نوار input یک واحد به جلو می‌رود.
اگر علامت بالای پشته یک متغیر بود، یکی از قوانین که طرف چپ آن متغیر مذکور است را انتخاب می‌کنیم و بالای پشته را با متغیر مورد نظر جایگزین می‌کنیم.
- پروسه بالا را آنقدر تکرار می‌کنیم تا اینکه پشته خالی شود و رشته ورودی کاملا خوانده شود.

$$
\begin{aligned}
S &\to ASA \mid \mathtt{a}B \\
A &\to B \mid S \\
B &\to \mathtt{b} \mid \varepsilon
\end{aligned}
$$