

درس مبانی نظریه محاسبه

ماشین های پشته‌ای

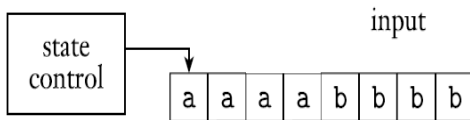
یک مدل محاسباتی برای زبانهای مستقل از متن

ماشینهای محاسباتی

- ◀ ماشین متناهی معین (زبانهای منظم)
- ◀ ماشین متناهی نامعین (زبانهای منظم)
- ◀ ماشین پشته‌ای (زبانهای مستقل از متن)
- ◀ ماشین پشته‌ای معین (زبانهای مستقل از متن معین)
- ◀ ماشین تورینگ (زبانهای شبه تصمیم پذیر)
- ◀ ...

در فصل قبل ماشین متناهی را معرفی کردیم و با زبانهای منظم آشنا شدیم. در این فصل با یک مدل قوی تر برای محاسبات به نام ماشین پشته ای آشنا می شویم که زبانهای مستقل از متن را می پذیرد.

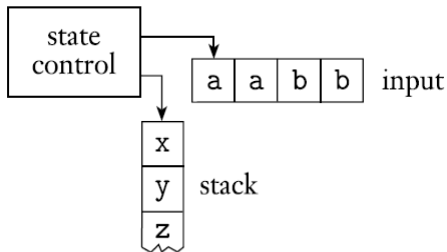
دیدیم که ماشین متناهی قدرت پذیرش زبانهای ساده ای مانند $\{a^n b^n \mid n \geq 0\}$ را ندارند. دلیل اصلی این ضعف، در متناهی بودن حافظه ماشین است. ما نیاز داریم تعداد a ها را بدانیم تا بعداً آن را با تعداد b ها مقایسه کنیم.



اولین ایده برای تقویت مدل ماشین متناهی **اضافه کردن حافظه نامحدود** به ماشین است.

حافظه نامحدود با دسترسی محدود

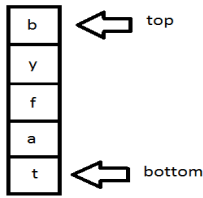
ماشین پشته‌ای تا حد زیادی شبیه متناهی است با این تفاوت که یک حافظه نامحدود به ماشین اضافه شده است. به این حافظه نامحدود پشته stack گفته می‌شود.



همانطور که از اسمش پیداست (مثل ساختار داده پشته در زبانهای برنامه‌نویسی) دسترسی به حافظه پشته کاملاً محدود و یکطرفه است.

قوانین حاکم بر پشته

- ▶ پشته یک حافظه خطی است که بالا top و پایین (کف) bottom دارد.



- ▶ ماشین فقط می‌تواند عنصر بالای پشته (کاراکتر ذخیره شده در بالای پشته) را ببیند. عناصر پایینتر در صورت حذف عناصری بالایی قابل دسترسی هستند.
- ▶ ماشین می‌تواند عنصری به بالای پشته اضافه کند یا عنصر بالای پشته را حذف کند. این معادل اعمال push و pop در ساختار داده پشته است. برای همین به ماشین پشته‌ای push-down automaton گفته می‌شود.

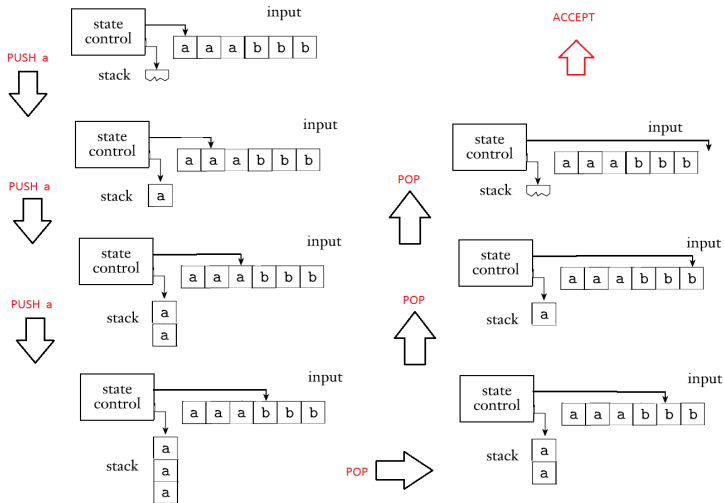
یک ماشین پشته‌ای برای زبان $a^n b^n$

چگونه از پشته برای تشخیص یک رشته با فرم $a^n b^n$ استفاده کنیم؟

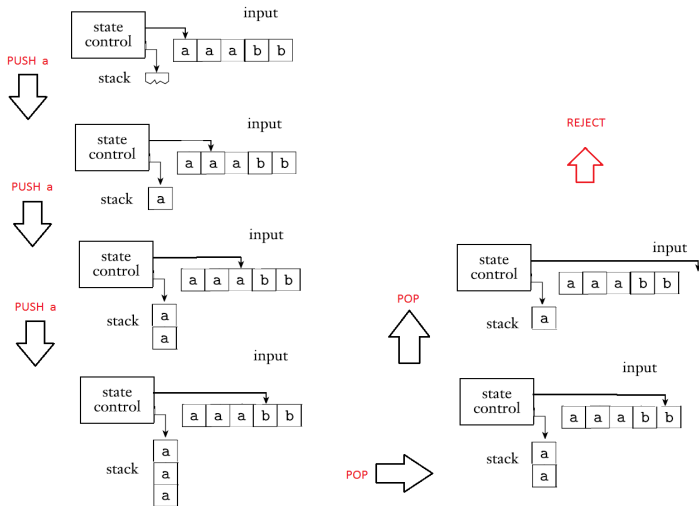
توصیف کلی ماشین:

- ◀ موقع خواندن رشته، تا رسیدن به اولین b ، همه a ها را در پشته ذخیره کن.
- ◀ با دیدن اولین b یک a از بالای پشته بردار. به این کار ادامه بده (با دیدن هر b یک a از بالای پشته بردار).
- ◀ اگر اولین کاراکتر b بود یا بعد از b یک a آمد، به حالت عدم پذیرش برو.
- ◀ در صورتیکه در انتهای رشته ورودی، پشته نیز خالی باشد به وضعیت پذیرش برو.

ماشین پشته‌ای برای زبان $a^n b^n$: یک نمونه اجرا



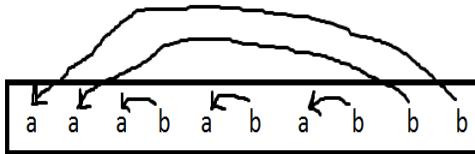
ماشین پشته‌ای برای زبان $a^n b^n$: یک نمونه اجرا



ماشین پشته‌ای برای زبان

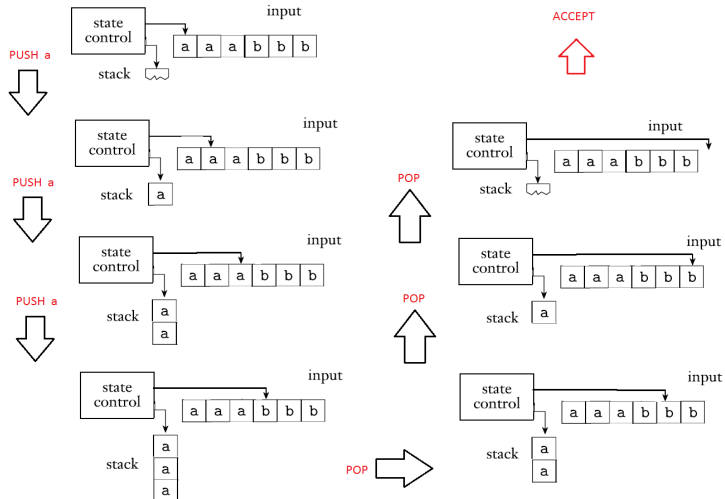
$$\{w \in \{a,b\}^* \mid n_w(a) = n_w(b)\}$$

توصیف کلی ماشین: با دیدن یک b اگر بالای پشته a باشد a را از بالای پشته بردار در غیر این صورت b را روی پشته بگذار. به همین ترتیب، با دیدن a اگر بالای پشته b باشد، b را از بالای پشته بردار در غیر این صورت a را بالای پشته بگذار. فقط در صورتیکه در انتهای رشته ورودی، پشته نیز خالی باشد به وضعیت پذیرش برو.



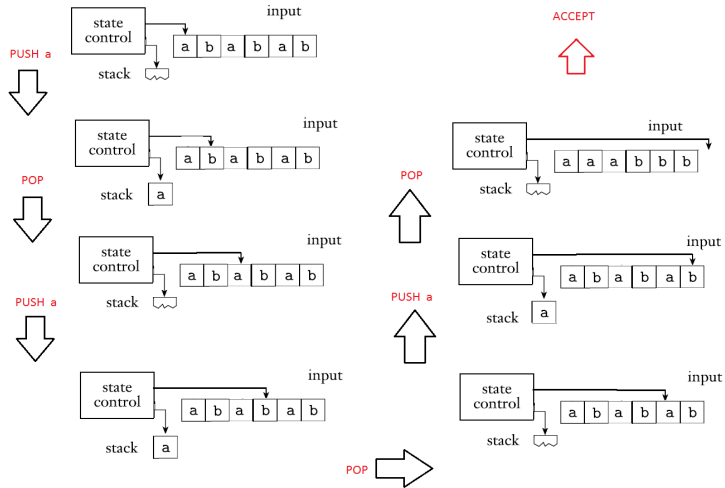
ماشین پشته‌ای برای زبان

$\{w \in \{a,b\}^* \mid n_w(a) = n_w(b)\}$: یک نمونه اجرا



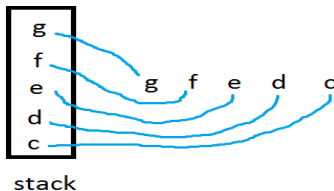
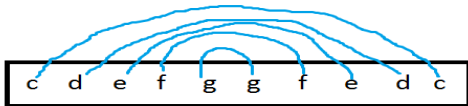
ماشین پشته‌ای برای زبان

$\{w \in \{a,b\}^* \mid n_w(a) = n_w(b)\}$: نمونه اجرای دیگر



ماشین پشته‌ای برای زبان $\{ww^R \mid w \in \Sigma^*\}$

توصیف کلی ماشین: ماشین وسط رشته را حدس می‌زند (استفاده از عدم قطعیت). فرض کنید حدس ماشین درست باشد. ماشین تا وسط رشته را در پشته قرار می‌دهد، سپس نصفه دوم رشته را با محتوای پشته مطابقت می‌دهد. هر جا که اختلافی پیدا شد به وضعیت عدم پذیرش می‌رود.



تعریف رسمی ماشین پشته‌ای

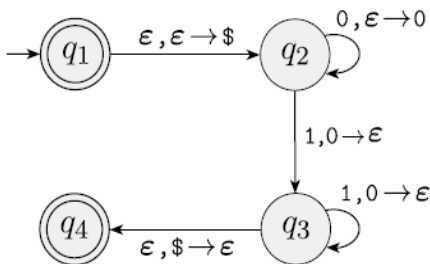
به یک ماشین پشته‌ای، یک pda نیز گفته می‌شود.

A *pushdown automaton* is a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$, where Q , Σ , Γ , and F are all finite sets, and

1. Q is the set of states,
2. Σ is the input alphabet,
3. Γ is the stack alphabet,
4. $\delta: Q \times \Sigma_{\varepsilon} \times \Gamma_{\varepsilon} \rightarrow \mathcal{P}(Q \times \Gamma_{\varepsilon})$ is the transition function,
5. $q_0 \in Q$ is the start state, and
6. $F \subseteq Q$ is the set of accept states.

فرض می‌کنیم الفبای پشته Γ همیشه حاوی علامت \$ است. از علامت \$ برای نشان دادن کف پشته استفاده می‌کنیم.

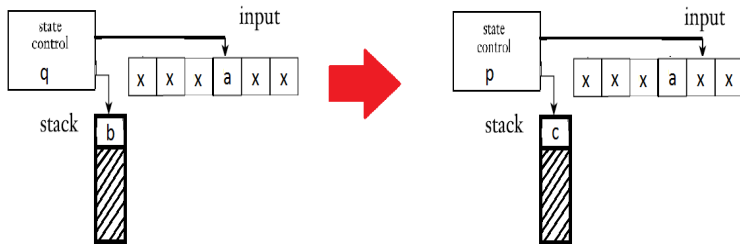
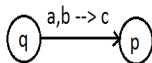
نمودار یک pda برای زبان $0^n 1^n$



توضیحات:

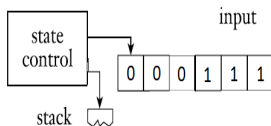
- ▶ در نمودار بالا q_1 وضعیت شروع است
- ▶ وضعیتهای q_1 و q_4 وضعیتهای پذیرش هستند.

برچسب $a, b \rightarrow c$ روی فلش بین وضعیتهای q و p به این معنی است که اگر حرف ورودی a باشد و علامت بالای پشته b باشد، حرف بالای پشته را با c عوض کن در حالیکه ماشین از وضعیت q به وضعیت p می‌رود.



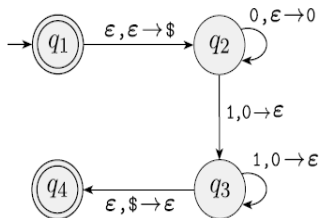
توضیح چند حالت خاص:

- ◀ اگر $a = \epsilon$ یعنی بدون مصرف رشته ورودی اگر حرف بالای پشته b است آن را با c عوض کن.
- ◀ اگر $b = \epsilon$ یعنی اگر حرف ورودی a است، حرف c را بالای پشته قرار بده.
- ◀ اگر $a = \epsilon$ و $b = \epsilon$ یعنی بدون مصرف رشته ورودی، حرف c را بالای پشته قرار بده.
- ◀ اگر $a = \epsilon$ و $c = \epsilon$ یعنی بدون اینکه چیزی از رشته ورودی بخوانی حرف بالای پشته را حذف کن.
- ◀ اگر $a = \epsilon$ و $b = \epsilon$ و $c = \epsilon$ یعنی بدون اینکه چیزی از رشته ورودی بخوانی و تغییری در پشته بدهی به وضعیت بعدی برو.



$w = 000111$

stack =	state = q_1	input = 0
stack = \$	state = q_2	input = 0
stack = 0\$	state = q_2	input = 0
stack = 00\$	state = q_2	input = 0
stack = 000\$	state = q_2	input = 1
stack = 00\$	state = q_3	input = 1
stack = 0\$	state = q_3	input = 1
stack = \$	state = q_3	input =
stack =	state = q_4	input =



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

