Pushdown Automata

Deepak D'Souza

Department of Computer Science and Automation
Indian Institute of Science, Bangalore.

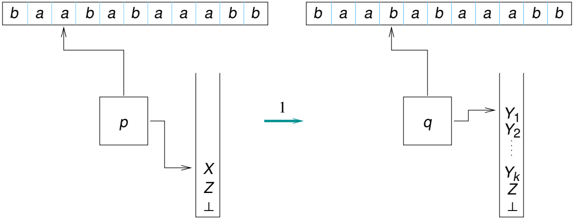
21 October 2019

Outline

Pushdown Automata + CFG: history

- CFG's were introduced by Noam Chomsky in 1956.
- Oettinger introduced PDA's for parsing applications in 1961.
- Chomsky, Schutzenberger, and Evey showed equivalence of CFG's and PDA's in 1962.

How a PDA works



Each step of the PDA looks like:

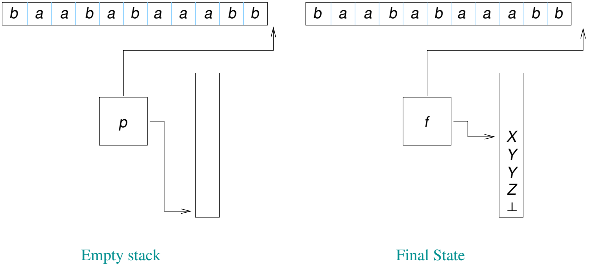
- Read current symbol and advance head;
- Read and pop top-of-stack symbol;
- Push in a string of symbols on the stack;
- Change state.

Each transition Looks like

$$(p, a, X) \to (q, Y_1 Y_2 \cdots Y_k).$$

Two mechanisms of acceptance

Mechanism used must be specified a priori in the PDA definition.



Accept input if

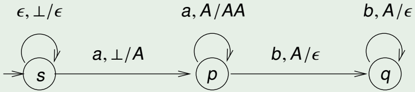
- Input is consumed and stack is empty (Acceptance by Empty Stack).
- Or, input is consumed and PDA is in a final state (Acceptance by Final State).

Example PDA

Example PDA (acceptance by empty stack) for $\{a^n b^n \mid n \geq 0\}$

$$\begin{array}{lcl} (s, \epsilon, \perp) & \rightarrow & (s, \epsilon) \\ (s, a, \perp) & \rightarrow & (p, A) \\ (p, a, A) & \rightarrow & (p, AA) \\ (p, b, A) & \rightarrow & (q, \epsilon). \\ (q, b, A) & \rightarrow & (q, \epsilon). \end{array}$$

Diagram representation

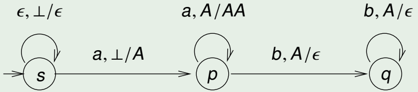


Example PDA

Example PDA (acceptance by empty stack) for $\{a^n b^n \mid n \geq 0\}$

$$
\begin{array}{lcl}
(s, \epsilon, \perp) & \to & (s, \epsilon) \\
(s, a, \perp) & \to & (p, A) \\
(p, a, A) & \to & (p, AA) \\
(p, b, A) & \to & (q, \epsilon). \\
(q, b, A) & \to & (q, \epsilon).
\end{array}
$$

Diagram representation



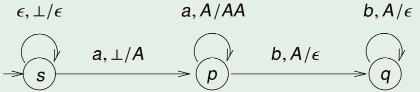
Illustrate run on input "*aaabbb*".

Example PDA

Example PDA (acceptance by empty stack) for $\{a^n b^n \mid n \geq 0\}$

$$\begin{array}{lcl} (s, \epsilon, \bot) & \rightarrow & (s, \epsilon) \\ (s, a, \bot) & \rightarrow & (p, A) \\ (p, a, A) & \rightarrow & (p, AA) \\ (p, b, A) & \rightarrow & (q, \epsilon). \\ (q, b, A) & \rightarrow & (q, \epsilon). \end{array}$$

Diagram representation



Illustrate run on input "*aaabbb*".

What happens on input "*aaabbbb*"?

PDA's more formally

A Pushdown Automaton is a structure of the form

$$\mathcal{M} = (Q, A, \Gamma, s, \delta, \perp, F)$$

where

- Q is a finite set of states,
- A is the input alphabet,
- Γ is the stack alphabet,
- $s \in Q$ is the start state,
- $\delta \subseteq_{fin} Q \times (A \cup \{\epsilon\}) \times \Gamma \times Q \times \Gamma^*$ is the (non-deterministic) transition relation,
- $\perp \in \Gamma$ is the bottom-of-stack symbol,
- $F \subseteq Q$ is the set of final states.

Configurations, runs, etc. of a PDA

- A configuration of $\mathcal{M}$ is of the form $(p, u, \gamma) \in Q \times A^* \times \Gamma^*$, which says "$\mathcal{A}$ is in state p, with unread input u, and stack contents γ".
- Initial configuration of $\mathcal{M}$ on input w is $(s, w, \perp)$.
- 1-step transition of $\mathcal{M}$: If $(p, a, X) \to (q, \alpha)$ is a transition in δ, then
$$(p, au, X\beta) \overset{1}{\Rightarrow} (q, u, \alpha\beta).$$
- Similarly, if $(p, \epsilon, X) \to (q, \alpha)$ is a transition in δ, then
$$(p, u, X\beta) \overset{1}{\Rightarrow} (q, u, \alpha\beta).$$
- $\mathcal{M}$ accepts w by empty stack if $(s, w, \perp) \overset{*}{\Rightarrow} (q, \epsilon, \epsilon)$.
- $\mathcal{M}$ accepts w by final state if $(s, w, \perp) \overset{*}{\Rightarrow} (f, \epsilon, \gamma)$ for some $f \in F$ and $\gamma \in \Gamma^*$.
- Language accepted by $\mathcal{M}$ is denoted $L(\mathcal{M})$.

Exercise

Design PDA's for the following languages:

- Balanced Parenthesis
- $\{a, b\}^* - \{ww \mid w \in \{a, b\}^*\}$.

Solution

PDA (acceptance by empty stack) for BP

$$
\begin{aligned}
(s, \epsilon, \bot) &\rightarrow (s, \epsilon) \\
(s, (, \bot) &\rightarrow (s, A\bot) \\
(s, (, A) &\rightarrow (s, AA) \\
(s,), A) &\rightarrow (s, \epsilon).
\end{aligned}
$$

Equivalence of acceptance criteria

Claim

- Given a PDA $\mathcal{M}$ that accepts by Final State we can give a PDA $\mathcal{M}'$ that accepts by Empty Stack such that $L(\mathcal{M}') = L(\mathcal{M})$.
- Conversely, given a PDA $\mathcal{M}$ that accepts by Empty Stack we can give a PDA $\mathcal{M}'$ that accepts by Final State such that $L(\mathcal{M}') = L(\mathcal{M})$.

In fact given a PDA $\mathcal{M}$ we can construct a PDA $\mathcal{M}'$ that accepts the same language as $\mathcal{M}$, by both acceptance criteria.

From Final State to ES/FS

What is the problem in doing this?

From Final State to ES/FS

What is the problem in doing this?

- $\mathcal{M}$ may reject an input by not entering a final state, yet emptying its stack.
- $\mathcal{M}$ may accept an input by reaching a final state, but not emptying its stack.

From Final State to ES/FS

What is the problem in doing this?

- $\mathcal{M}$ may reject an input by not entering a final state, yet emptying its stack.
- $\mathcal{M}$ may accept an input by reaching a final state, but not emptying its stack.

Let $\mathcal{M} = (Q, A, \Gamma, s, \delta, \perp, F)$.

Define $\mathcal{M}' = (Q \cup \{s', t\}, A, \Gamma \cup \{\perp\!\!\!\perp\}, s', \delta', \perp\!\!\!\perp, \{t\})$, where δ' is δ plus the transitions:

$$
\begin{array}{lcll}
(s', \epsilon, \perp\!\!\!\perp) & \rightarrow & (s, \perp\perp\!\!\!\perp) & \\
(s, a, \perp) & \rightarrow & (p, A) & \text{original transition in } \delta \\
(f, \epsilon, X) & \rightarrow & (t, X) & \text{for } X \in \Gamma \cup \{\perp\!\!\!\perp\} \\
(t, \epsilon, X) & \rightarrow & (t, \epsilon) & \text{for } X \in \Gamma \cup \{\perp\!\!\!\perp\}.
\end{array}
$$

- Argue that if $w \in L(M)$ then $w \in L(M')$.
- Argue that if $w \in L(M')$ then $w \in L(M)$.

From Empty Stack to ES/FS

- Let $\mathcal{M} = (Q, A, \Gamma, s, \delta, \perp)$.
- Define $\mathcal{M}' = (Q \cup \{s', t\}, A, \Gamma \cup \{\perp\!\!\!\perp\}, s', \delta', \perp\!\!\!\perp, \{t\})$, where δ' is δ plus the transitions:

$$
\begin{array}{lcll}
(s', \epsilon, \perp\!\!\!\perp) & \rightarrow & (s, \perp\perp\!\!\!\perp) & \\
(q, \epsilon, \perp\!\!\!\perp) & \rightarrow & (t, \perp\!\!\!\perp) & \text{for } q \in Q \\
(t, \epsilon, \perp\!\!\!\perp) & \rightarrow & (t, \epsilon). &
\end{array}
$$

- Argue that if $w \in L(M)$ then $w \in L(M')$.
- Argue that if $w \in L(M')$ then $w \in L(M)$.