Equivalence of CFGs and PDAs

Deepak D'Souza

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

05 March 2024

Outline

CFG = PDA

Theorem (Chomsky-Evey-Schutzenberger 1962)

The class of languages definable by Context-Free Grammars and Pushdown Automata coincide.

From CFG to PDA

Leftmost derivation: A derivation in which at each step the left-most non-terminal is rewritten.

CFG G_4

$$S \rightarrow (S) \mid SS \mid \epsilon.$$

Leftmost derivation in G_4:

$$\underline{S}$$

From CFG to PDA

Leftmost derivation: A derivation in which at each step the left-most non-terminal is rewritten.

CFG G_4

$$S \rightarrow (S) \mid SS \mid \epsilon.$$

Leftmost derivation in G_4:

$$\underline{S} \Rightarrow (\underline{S})$$

From CFG to PDA

Leftmost derivation: A derivation in which at each step the left-most non-terminal is rewritten.

CFG G_4

$$S \rightarrow (S) \mid SS \mid \epsilon.$$

Leftmost derivation in G_4:

$$\begin{aligned} \underline{S} &\Rightarrow (\underline{S}) \\ &\Rightarrow (\underline{S}S) \end{aligned}$$

From CFG to PDA

Leftmost derivation: A derivation in which at each step the left-most non-terminal is rewritten.

CFG G_4

$$S \quad \rightarrow \quad (S) \mid SS \mid \epsilon.$$

Leftmost derivation in G_4:

$$\begin{aligned} \underline{S} &\Rightarrow (\underline{S}) \\ &\Rightarrow (\underline{S}S) \\ &\Rightarrow (\underline{S}SS) \end{aligned}$$

From CFG to PDA

Leftmost derivation: A derivation in which at each step the left-most non-terminal is rewritten.

CFG G_4

$$S \rightarrow (S) \mid SS \mid \epsilon.$$

Leftmost derivation in G_4:

$$\begin{aligned} \underline{S} &\Rightarrow (\underline{S}) \\ &\Rightarrow (\underline{S}S) \\ &\Rightarrow (\underline{S}SS) \\ &\Rightarrow ((\underline{S})SS) \end{aligned}$$

From CFG to PDA

Leftmost derivation: A derivation in which at each step the left-most non-terminal is rewritten.

CFG G_4

$$S \rightarrow (S) \mid SS \mid \epsilon.$$

Leftmost derivation in G_4:

$$\begin{aligned} \underline{S} &\Rightarrow (\underline{S}) \\ &\Rightarrow (\underline{S}S) \\ &\Rightarrow (\underline{S}SS) \\ &\Rightarrow ((\underline{S})SS) \\ &\Rightarrow ((\underline{S}S)SS) \end{aligned}$$

From CFG to PDA

Leftmost derivation: A derivation in which at each step the left-most non-terminal is rewritten.

CFG G_4

$$S \rightarrow (S) \mid SS \mid \epsilon.$$

Leftmost derivation in G_4:

$$\begin{aligned}
\underline{S} &\Rightarrow (\underline{S}) \\
&\Rightarrow (\underline{S}S) \\
&\Rightarrow (\underline{S}SS) \\
&\Rightarrow ((\underline{S})SS) \\
&\Rightarrow ((\underline{S}S)SS) \\
&\Rightarrow (((\underline{S})S)SS)
\end{aligned}$$

From CFG to PDA

Leftmost derivation: A derivation in which at each step the left-most non-terminal is rewritten.

CFG G_4

$$S \rightarrow (S) \mid SS \mid \epsilon.$$

Leftmost derivation in G_4:

$$\begin{aligned}
\underline{S} &\Rightarrow (\underline{S}) \\
&\Rightarrow (\underline{S}S) \\
&\Rightarrow (\underline{S}SS) \\
&\Rightarrow ((\underline{S})SS) \\
&\Rightarrow ((\underline{S}S)SS) \\
&\Rightarrow (((\underline{S})S)SS) \\
&\Rightarrow ((()\underline{S})SS)
\end{aligned}$$

From CFG to PDA

Leftmost derivation: A derivation in which at each step the left-most non-terminal is rewritten.

CFG G_4

$$S \rightarrow (S) \mid SS \mid \epsilon.$$

Leftmost derivation in G_4:

$$\begin{aligned}
\underline{S} &\Rightarrow (\underline{S}) \\
&\Rightarrow (\underline{S}S) \\
&\Rightarrow (\underline{S}SS) \\
&\Rightarrow ((\underline{S})SS) \\
&\Rightarrow ((\underline{S}S)SS) \\
&\Rightarrow (((\underline{S})S)SS) \\
&\Rightarrow ((()\underline{S})SS) \\
&\Rightarrow ((())\underline{S}S)
\end{aligned}$$

From CFG to PDA

Leftmost derivation: A derivation in which at each step the left-most non-terminal is rewritten.

CFG G_4

$$S \rightarrow (S) \mid SS \mid \epsilon.$$

Leftmost derivation in G_4:

$$\begin{aligned}
\underline{S} &\Rightarrow (\underline{S}) \\
&\Rightarrow (\underline{S}S) \\
&\Rightarrow (\underline{S}SS) \\
&\Rightarrow ((\underline{S})SS) \\
&\Rightarrow ((\underline{S}S)SS) \\
&\Rightarrow (((\underline{S})S)SS) \\
&\Rightarrow ((()\underline{S})SS) \\
&\Rightarrow ((())\underline{S}S) \\
&\Rightarrow ((())(\underline{S})S)
\end{aligned}$$

From CFG to PDA

Leftmost derivation: A derivation in which at each step the left-most non-terminal is rewritten.

CFG G_4

$$S \rightarrow (S) \mid SS \mid \epsilon.$$

Leftmost derivation in G_4:

$$\begin{aligned}
\underline{S} &\Rightarrow (\underline{S}) \\
&\Rightarrow (\underline{S}S) \\
&\Rightarrow (\underline{S}SS) \\
&\Rightarrow ((\underline{S})SS) \\
&\Rightarrow ((\underline{S}S)SS) \\
&\Rightarrow (((\underline{S})S)SS) \\
&\Rightarrow ((()\underline{S})SS) \\
&\Rightarrow ((())\underline{S}S) \\
&\Rightarrow ((())(\underline{S})S) \\
&\Rightarrow ((())()\underline{S}) \quad \Rightarrow \quad ((())()).
\end{aligned}$$

From CFG to PDA

Let $G = (N, A, S, P)$ be a CFG. Assume WLOG that all rules of G are of the form

$$X \to cB_1 B_2 \cdots B_k$$

where $c \in A \cup \{\epsilon\}$ and $k \geq 0$.

- Idea: Define a PDA M that simulates a leftmost derivation of G on the given input.
- Define $M = (\{s\}, A, N, s, \delta, S)$ where δ is given by:

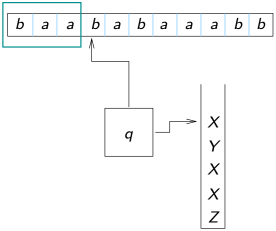
  $$(s, c, X) \quad \to \quad (s, B_1 B_2 \cdots B_k),$$

  whenever $X \to cB_1 B_2 \cdots B_k$ is a production in G.
- M accepts by empty stack.

CFG to PDA

$$\boxed{b \quad a \quad a}\ X \quad Y \quad X \quad X \quad Z$$

Leftmost sentential form of G



Corresponding configuration of M

Exercise

Construct a PDA for the CFG below.

CFG G_4

$$\begin{aligned} S &\rightarrow (SR \mid SS \mid \epsilon \\ R &\rightarrow) \end{aligned}$$

Simulate it on the input "((())())".

From PDA to CFG

Given a PDA M, how would you construct an “equivalent” context-free grammar from M?

From PDA to CFG

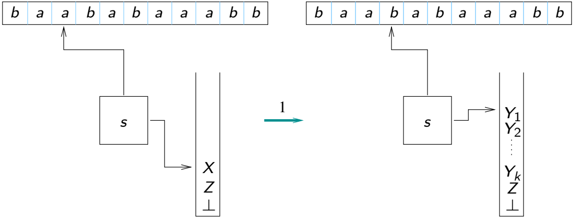
Given a PDA M, how would you construct an "equivalent" context-free grammar from M?

One approach:

- First show that we can go over to a PDA M' with a single state.
- Then simulate M' by a CFG.

Simulating a single-state PDA by a CFG

If:



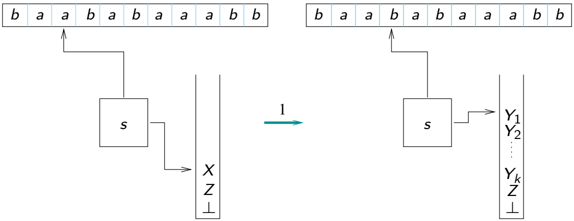
Then: add the rule $X \to aY_1 Y_2 \cdots Y_k$ in G.

In particular, if $(s, c, \perp) \to (s, \alpha)$ we add $S \to c\alpha$ in G.

Start symbol?

Simulating a single-state PDA by a CFG

If:



Then: add the rule $X \to aY_1Y_2\cdots Y_k$ in G.

In particular, if $(s, c, \perp) \to (s, \alpha)$ we add $S \to c\alpha$ in G.

Start symbol? $\perp$.

From PDA to single-state PDA

- Let $M = (Q, A, \Gamma, s, \delta, \perp, \{t\})$ be the given PDA which WLOG accepts by final state t and can empty its stack in t.
- Define $M' = (\{u\}, A, Q \times \Gamma \times Q, u, \delta', (s, \perp, t), \emptyset)$, which accepts by empty stack and where δ' is given by

$$(u, c, (p, A, r)) \to (u, (q_0 B_1 q_1)(q_1 B_2 q_2) \cdots (q_{k-1} B_k q_k))$$

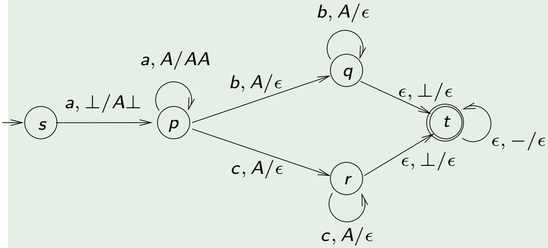
whenever $(p, c, A) \to (q, (B_1 B_2 \cdots B_k))$ is a transition of M, and $q_0 = q$ and $q_k = r$. In particular:

$$(u, c, (p, A, q)) \to (u, \epsilon)$$

if $(p, c, A) \to (q, \epsilon)$ is a transition of M.

Example to illustrate construction

PDA (acceptance by final state t) for $\{a^n b^n \mid n \geq 1\} \cup \{a^n c^n \mid n \geq 1\}$



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

Correctness of construction

To show that $L(M') = L(M)$, sufficient to show that:

Claim 1

In M, $(s, x, A) \overset{*}{\Rightarrow} (t, \epsilon, \epsilon)$ iff in M' $(u, x, (s, A, t)) \overset{*}{\Rightarrow} (u, \epsilon, \epsilon)$.

For this in turn, it is sufficient to show that:

Claim 2

$(p, x, B_1 B_2 \cdots B_k) \overset{n}{\Rightarrow} (q, \epsilon, \epsilon)$ in M iff exists $q_0, \ldots, q_k$ such that $q_0 = p$, $q_k = q$, and
$(u, x, (\langle q_0 B_1 q_1 \rangle \langle q_1 B_2 q_2 \rangle \cdots \langle q_{k-1} B_k q_k \rangle)) \overset{n}{\Rightarrow} (u, \epsilon, \epsilon)$

Proof is easy by induction on n.