

Equivalence of CFG's and PDA's

Deepak D'Souza

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

11 October 2013

Outline

1 From CFG to PDA

2 From PDA to CFG

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 :

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 :

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

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 :

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

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 :

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

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 :

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

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 :

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

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 :

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

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 :

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

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 :

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

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 :

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

From CFG to PDA

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

$$X \rightarrow cB_1B_2 \cdots B_k$$

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

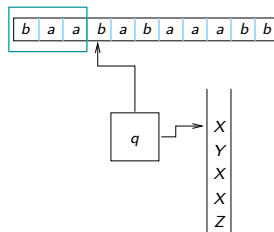
- Idea: Define a PDA M that simulates a leftmost derivation of G .
- Define $M = (\{s\}, A, N, s, \delta, S)$ where δ is given by:

$$(s, c, X) \rightarrow (s, B_1B_2 \cdots B_k),$$

whenever $X \rightarrow cB_1B_2 \cdots B_k$ is a production in G .

CFG to PDA

b a a X Y X X Z

Leftmost sentential form of G Corresponding configuration of M

Exercise

Construct a PDA for the CFG below.

CFG G_4

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

Simulate it on the input “((()))”.

From PDA to CFG

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

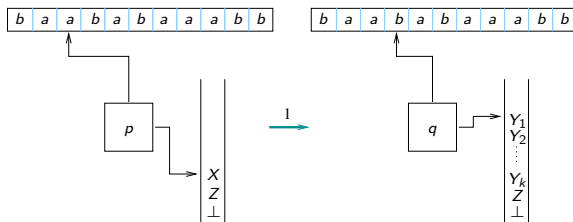
From PDA to CFG

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



- Add the rule $X \rightarrow aY_1Y_2\cdots Y_k$ in G .
- If $(s, c, \perp) \rightarrow (s, \alpha)$ then add $S \rightarrow c\alpha$ in G .

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

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

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

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

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

Example to illustrate construction

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

$$(s, a, \perp) \rightarrow (p, A\perp)$$
$$(p, a, A) \rightarrow (p, AA)$$
$$(p, b, A) \rightarrow (q, \epsilon).$$
$$(p, c, A) \rightarrow (r, \epsilon).$$
$$(q, b, A) \rightarrow (q, \epsilon).$$
$$(r, c, A) \rightarrow (r, \epsilon).$$
$$(q, b, \perp) \rightarrow (t, \epsilon).$$
$$(r, c, \perp) \rightarrow (t, \epsilon).$$