

Lecture 16 – Equivalence of Pushdown Automata and Context-Free Grammars

COSE215: Theory of Computation

Jihyeok Park



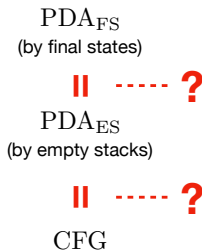
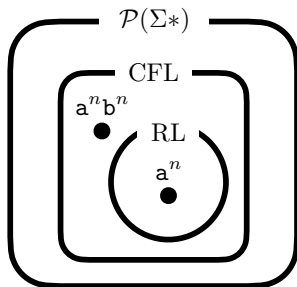
2026 Spring

A **context-free grammar** is a 4-tuple:

$$G = (V, \Sigma, S, R)$$

A **pushdown automaton (PDA)** is a finite automaton with a **stack**.

- Acceptance by **final states**
- Acceptance by **empty stacks**



1. Equivalence of PDA by Final States and Empty Stacks

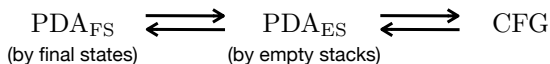
PDA_{FS} to PDA_{ES}

PDA_{ES} to PDA_{FS}

2. Equivalence of PDA and CFGs

CFGs to PDA_{ES}

PDA_{ES} to CFGs



1. Equivalence of PDA by Final States and Empty Stacks

PDA_{FS} to PDA_{ES}

PDA_{ES} to PDA_{FS}

2. Equivalence of PDA and CFGs

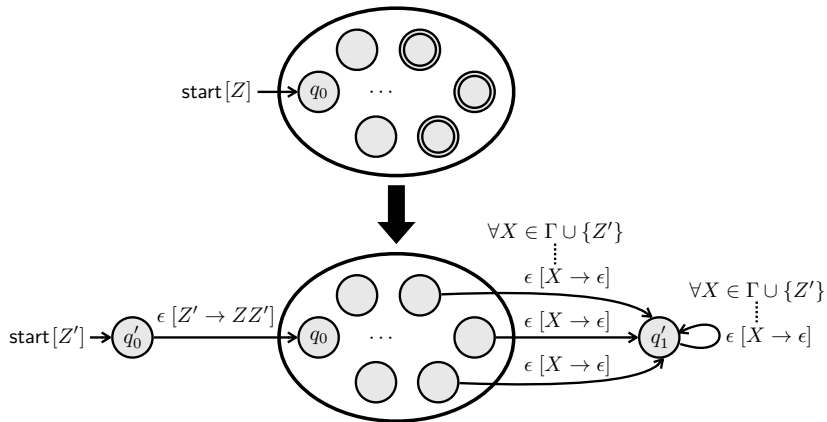
CFGs to PDA_{ES}

PDA_{ES} to CFGs



Theorem (PDA_{FS} to PDA_{ES})

For a given PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$, \exists PDA P' . $L_F(P) = L_E(P')$.



Theorem (PDA_{FS} to PDA_{ES})

For a given PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$, \exists PDA P' . $L_F(P) = L_E(P')$.

Define a PDA

$$P' = (Q \cup \{q'_0, q'_1\}, \Sigma, \Gamma \cup \{Z'\}, \delta', q'_0, Z', \emptyset)$$

where

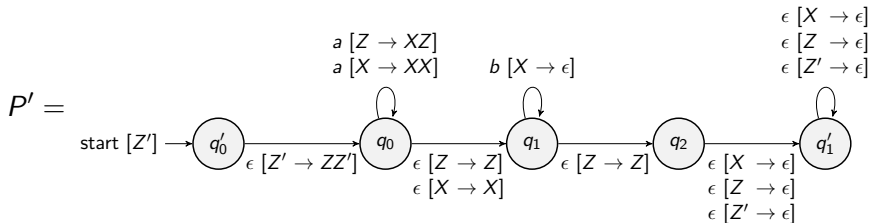
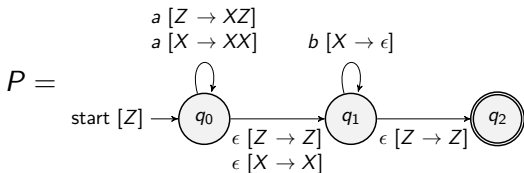
$$\delta'(q'_0, \epsilon, Z') = \{(q_0, ZZ')\}$$

$$\delta'(q \in Q, a \in \Sigma, X \in \Gamma) = \delta(q, a, X)$$

$$\delta'(q \in Q, \epsilon, X \in \Gamma \cup \{Z'\}) = \begin{cases} \delta(q, \epsilon, X) \cup \{(q'_1, \epsilon)\} & \text{if } q \in F \\ \delta(q, \epsilon, X) & \text{otherwise} \end{cases}$$

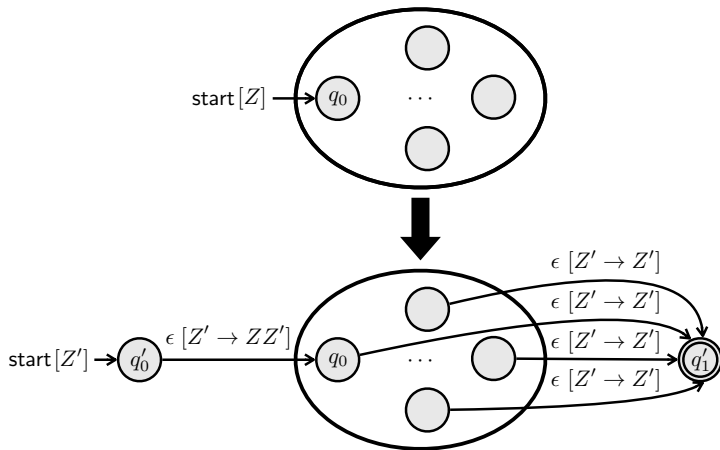
$$\delta'(q'_1, \epsilon, X \in \Gamma \cup \{Z'\}) = \{(q'_1, \epsilon)\}$$

$$L_F(P) = L_E(P') = \{a^n b^n \mid n \geq 0\}$$



Theorem (PDA_{ES} to PDA_{FS})

For a given PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$, \exists PDA P' . $L_E(P) = L_F(P')$.



Theorem (PDA_{ES} to PDA_{FS})

For a given PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$, \exists PDA P' . $L_E(P) = L_F(P')$.

Define a PDA

$$P' = (Q \cup \{q'_0, q'_1\}, \Sigma, \Gamma \cup \{Z'\}, \delta', q'_0, Z', \{q'_1\})$$

where

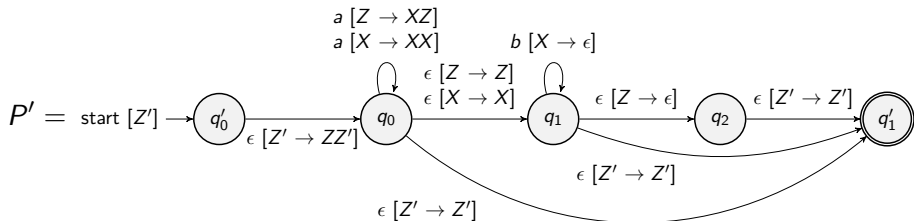
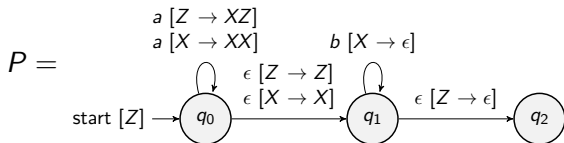
$$\delta'(q'_0, \epsilon, Z') = \{(q_0, ZZ')\}$$

$$\delta'(q \in Q, a \in \Sigma, X \in \Gamma) = \delta(q, a, X)$$

$$\delta'(q \in Q, \epsilon, X \in \Gamma) = \delta(q, \epsilon, X)$$

$$\delta'(q \in Q, \epsilon, Z') = \{(q'_1, Z')\}$$

$$L_E(P) = L_F(P') = \{a^n b^n \mid n \geq 0\}$$



1. Equivalence of PDA by Final States and Empty Stacks

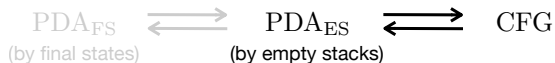
PDA_{FS} to PDA_{ES}

PDA_{ES} to PDA_{FS}

2. Equivalence of PDA and CFGs

CFGs to PDA_{ES}

PDA_{ES} to CFGs



Theorem (CFGs to PDA_{ES})

For a given CFG $G = (V, \Sigma, S, R)$, \exists PDA P . $L(G) = L_E(P)$.

The key idea is to represent **sentential forms** of the CFG during the derivation process as the content of the **stack** of the PDA.

Define a PDA

$$P = (\{q\}, \Sigma, V \cup \Sigma, \delta, q, S, \emptyset)$$

where

$$\delta(q, \epsilon, A \in V) = \{(q, \alpha) \mid A \rightarrow \alpha \in R\}$$

$$\delta(q, a \in \Sigma, a \in \Sigma) = \{(q, \epsilon)\}$$

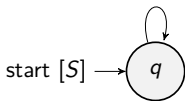
$$\begin{aligned} \delta(q, \epsilon, A \in V) &= \{(q, \alpha) \mid A \rightarrow \alpha \in R\} \\ \delta(q, a \in \Sigma, a \in \Sigma) &= \{(q, \epsilon)\} \end{aligned}$$

Consider the following CFG:

$$S \rightarrow \epsilon \mid aSb \mid bSa \mid SS$$

Then, the equivalent PDA (by empty stacks) is:

$\epsilon [S \rightarrow \epsilon]$	$(q, abab, S) \vdash (q, abab, aSb)$
$\epsilon [S \rightarrow aSb]$	$\vdash (q, bab, Sb)$
$\epsilon [S \rightarrow bSa]$	$\vdash (q, bab, bSab)$
$\epsilon [S \rightarrow SS]$	$\vdash (q, ab, Sab)$
$a [a \rightarrow \epsilon]$	$\vdash (q, ab, ab)$
$b [b \rightarrow \epsilon]$	$\vdash (q, b, b)$
	$\vdash (q, \epsilon, \epsilon)$



Theorem (PDA_{ES} to CFGs)

For a given PDA $P = (Q = \{q_0, \dots, q_{n-1}\}, \Sigma, \Gamma, \delta, q_0, Z, F)$,
 \exists CFG G . $L_E(P) = L(G)$.

The key idea is defining a variable $A_{i,j}^X$ for each $0 \leq i, j < n$ and $X \in \Gamma$ that generates all words causing the PDA to move from q_i to q_j by popping X :

$$A_{i,j}^X \Rightarrow^* w \quad \text{if and only if} \quad (q_i, w, X) \vdash^* (q_j, \epsilon, \epsilon)$$

With this idea, we can define a CFG that generates all words accepted by the PDA P with empty stacks as follows:

$$S \rightarrow A_{0,0}^Z \mid A_{0,1}^Z \mid \dots \mid A_{0,n-1}^Z$$

Then, how to define production rules for $A_{i,j}^X$?

We can define production rules for $A_{i,j}^X$ as follows.

Consider a transition $(q_j, X_1 \cdots X_m) \in \delta(q_i, a, X)$ for all $q_i \in Q$, $a \in \Sigma \cup \{\epsilon\}$, $X \in \Gamma$.

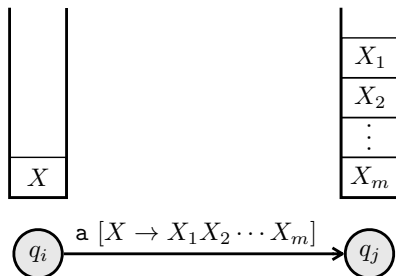
It makes PDA move from q_i to q_j by replacing X with $X_1 \cdots X_m$.

Then, we need to pop X_1, \cdots, X_m from the stack to make the stack empty.

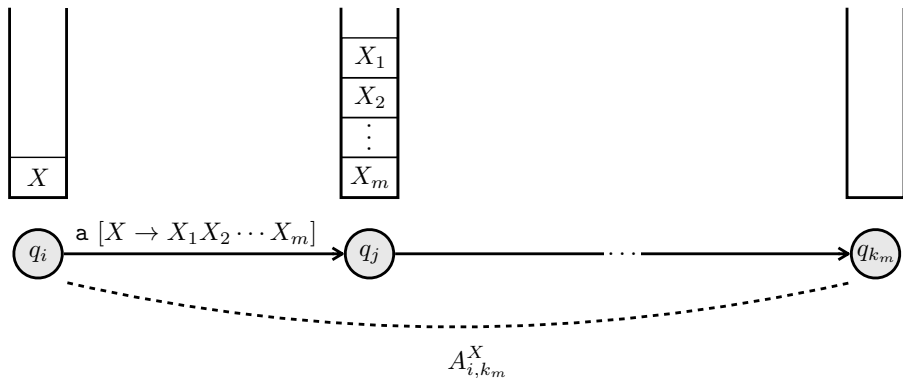
Let k_1, \cdots, k_m be the states that the PDA moves to after popping X_1, \cdots, X_m , respectively.

To cover all possible combinations of k_1, \cdots, k_m , we need to define a production rule for A_{i,k_m}^X as follows:

$$A_{i,k_m}^X \rightarrow a A_{j,k_1}^{X_1} A_{k_1,k_2}^{X_2} \cdots A_{k_{m-1},k_m}^{X_m} \text{ for all } 0 \leq k_1, \cdots, k_m < n$$

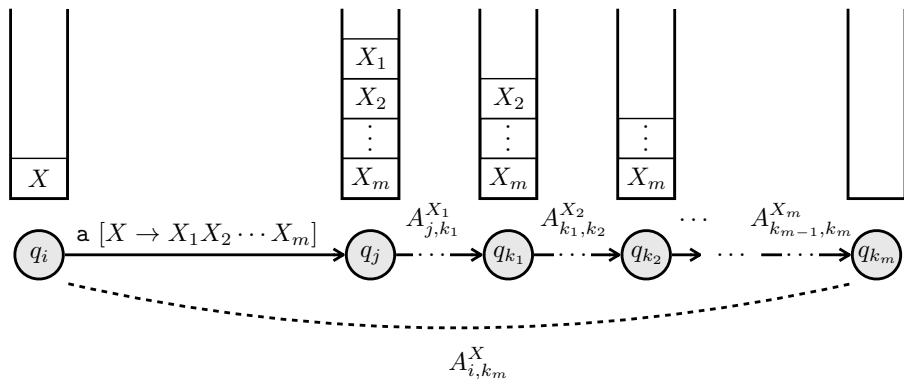


Consider a transition $(q_j, X_1 \dots X_m) \in \delta(q_i, a, X)$.



Using this transition, let's define a production rule for A_{i, k_m}^X for some k_m .

Then, we need to pop X_1, \dots, X_m from the stack to make the stack empty when moving from q_j to q_{k_m} .



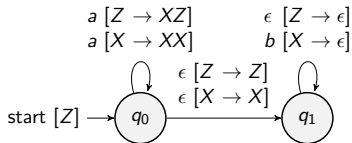
We can do it by popping X_i when moving to q_{k_i} for each $1 \leq i \leq m$.

Thus, we get the following production rule for A_{i,k_m}^X :

$$A_{i,k_m}^X \rightarrow a A_{j,k_1}^{X_1} A_{k_1,k_2}^{X_2} \dots A_{k_{m-1},k_m}^{X_m} \text{ for all } 0 \leq k_1, \dots, k_m < n$$

$$S \rightarrow A_{0,j}^Z \qquad A_{i,k_m}^X \rightarrow a A_{j,k_1}^{X_1} A_{k_1,k_2}^{X_2} \dots A_{k_{m-1},k_m}^{X_m}$$

Consider the following PDA (by empty stacks):



Then, the equivalent CFG is:

$$\begin{array}{l}
 S \rightarrow A_{0,0}^Z \mid A_{0,1}^Z \\
 A_{0,0}^Z \rightarrow a A_{0,0}^X A_{0,0}^Z \mid a A_{0,1}^X A_{1,0}^Z \mid A_{1,0}^Z \\
 A_{0,1}^Z \rightarrow a A_{0,0}^X A_{0,1}^Z \mid a A_{0,1}^X A_{1,1}^Z \mid A_{1,1}^Z \\
 A_{0,0}^X \rightarrow a A_{0,0}^X A_{0,0}^X \mid a A_{0,1}^X A_{1,0}^X \mid A_{1,0}^X \\
 A_{0,1}^X \rightarrow a A_{0,0}^X A_{0,1}^X \mid a A_{0,1}^X A_{1,1}^X \mid A_{1,1}^X \\
 A_{1,1}^Z \rightarrow \epsilon \\
 A_{1,1}^X \rightarrow b
 \end{array}$$

$$\begin{array}{l}
 S \Rightarrow A_{0,1}^Z \\
 \Rightarrow a A_{0,1}^X A_{1,1}^Z \\
 \Rightarrow aa A_{0,1}^X A_{1,1}^X A_{1,1}^Z \\
 \Rightarrow aa A_{1,1}^X A_{1,1}^X A_{1,1}^Z \\
 \Rightarrow aab A_{1,1}^X A_{1,1}^Z \\
 \Rightarrow aabb A_{1,1}^Z \\
 \Rightarrow aabb
 \end{array}$$

1. Equivalence of PDA by Final States and Empty Stacks

PDA_{FS} to PDA_{ES}

PDA_{ES} to PDA_{FS}

2. Equivalence of PDA and CFGs

CFGs to PDA_{ES}

PDA_{ES} to CFGs



- Deterministic Pushdown Automata (DPDA)

Jihyeok Park

jihyeok_park@korea.ac.kr

<https://plrg.korea.ac.kr>