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

COSE215: Theory of Computation

Jihyeok Park

PLRG

2026 Spring

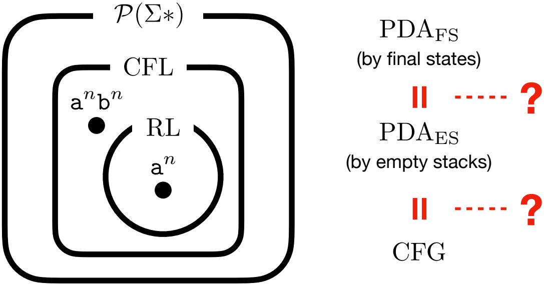
Recall

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**



Contents

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

Contents

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

PDA_{FS} to PDA_{ES}

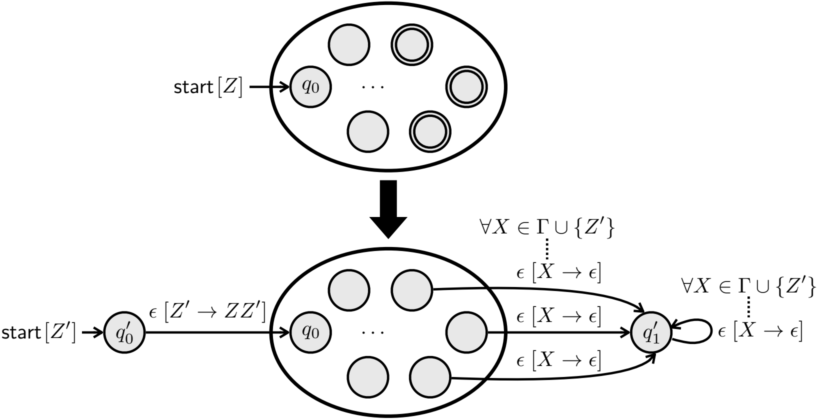
Theorem (PDA_{FS} to PDA_{ES})

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

PDA_{FS} to PDA_{ES}

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

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



PDA_{FS} to PDA_{ES}

PLRG

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

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

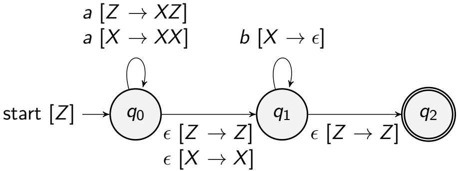
where

$$\begin{array}{lcl}
\delta'(q'_0, \epsilon, Z') & = & \{(q_0, ZZ')\} \\
\delta'(q \in Q, a \in \Sigma, X \in \Gamma) & = & \delta(q, a, X) \\
\delta'(q \in Q, \epsilon, X \in \Gamma \cup \{Z'\}) & = & \begin{cases} \delta(q, \epsilon, X) \cup \{(q'_1, \epsilon)\} & \text{if } q \in F \\ \delta(q, \epsilon, X) & \text{otherwise} \end{cases} \\
\delta'(q'_1, \epsilon, X \in \Gamma \cup \{Z'\}) & = & \{(q'_1, \epsilon)\}
\end{array}$$

PDA_{FS} to PDA_{ES} – Example

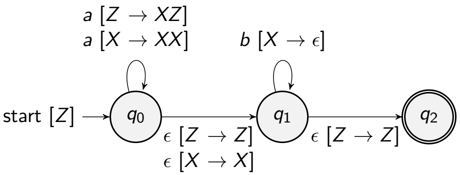
$$L_F(P) = L_E(P') = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$$

$P =$

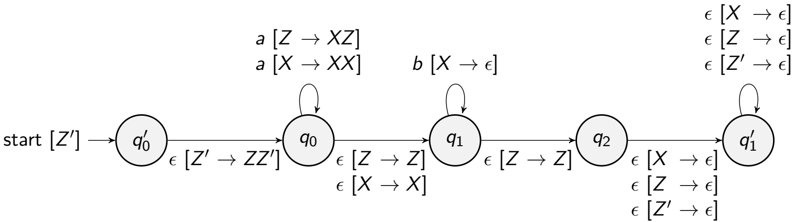


$$L_F(P) = L_E(P') = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$$

$P =$



$P' =$



PDA_{ES} to PDA_{FS}

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

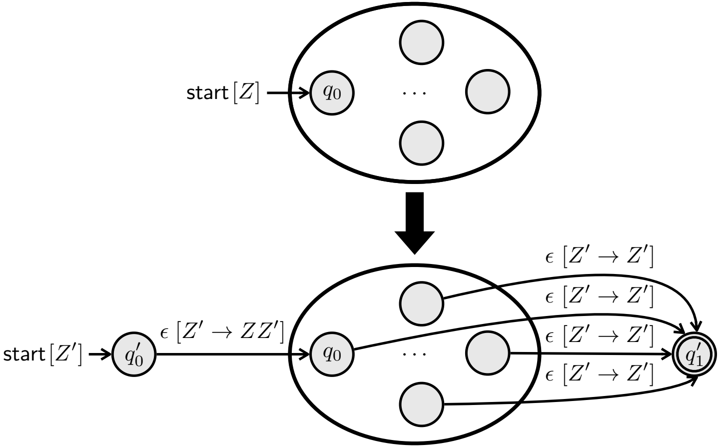
For a given PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$, $\exists$ PDA P'. $L_E(P) = L_F(P')$.

PDA$_{ES}$ to PDA$_{FS}$

PLRG

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

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

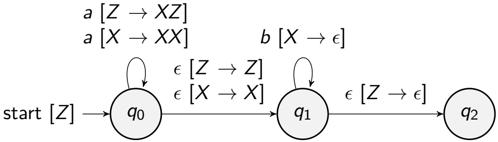
where

$$\begin{array}{rcl} \delta'(q'_0, \epsilon, Z') & = & \{(q_0, ZZ')\} \\ \delta'(q \in Q, a \in \Sigma, X \in \Gamma) & = & \delta(q, a, X) \\ \delta'(q \in Q, \epsilon, X \in \Gamma) & = & \delta(q, \epsilon, X) \\ \delta'(q \in Q, \epsilon, Z') & = & \{(q'_1, Z')\} \end{array}$$

PDA_{ES} to PDA_{FS} – Example

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

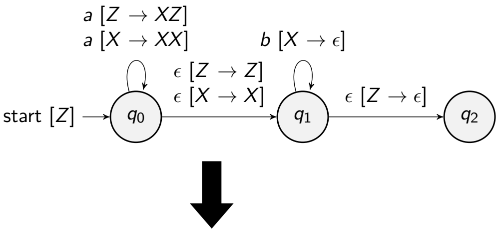
$P =$



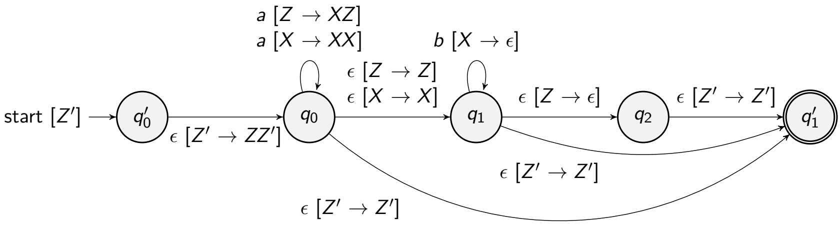
PDA_{ES} to PDA_{FS} – Example

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

$P =$



$P' =$



Contents

CFGs to PDA_{ES}

Theorem (CFGs to PDA_{ES})

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

CFGs to PDA_{ES}

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.

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

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

where

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

CFGs to PDA_{ES} – Example

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

Consider the following CFG:

$$S \to \epsilon \mid \texttt{a}S\texttt{b} \mid \texttt{b}S\texttt{a} \mid SS$$

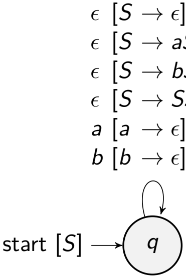
CFGs to PDA$_{ES}$ – Example

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

Consider the following CFG:

$$S \to \epsilon \mid \mathtt{a}S\mathtt{b} \mid \mathtt{b}S\mathtt{a} \mid SS$$

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



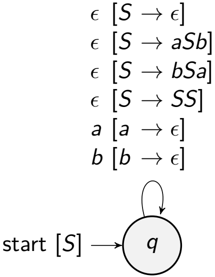
CFGs to PDA$_{\text{ES}}$ – Example

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

Consider the following CFG:

$$S \to \epsilon \mid \mathtt{a}S\mathtt{b} \mid \mathtt{b}S\mathtt{a} \mid SS$$

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



$$\begin{aligned}(q, \mathtt{abab}, S) &\vdash (q, \mathtt{abab}, \mathtt{a}S\mathtt{b}) \\ &\vdash (q, \mathtt{bab}, S\mathtt{b}) \\ &\vdash (q, \mathtt{bab}, \mathtt{b}S\mathtt{ab}) \\ &\vdash (q, \mathtt{ab}, S\mathtt{ab}) \\ &\vdash (q, \mathtt{ab}, \mathtt{ab}) \\ &\vdash (q, \mathtt{b}, \mathtt{b}) \\ &\vdash (q, \epsilon, \epsilon)\end{aligned}$$

PDA_{ES} to CFGs

Theorem (PDA_{ES} to CFGs)

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

PDA_{ES} to CFGs

PLRG

Theorem (PDA_{ES} to CFGs)

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

The key idea is defining a variable $A^X_{i,j}$ 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:

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

PDA$_{\text{ES}}$ to CFGs

Theorem (PDA$_{\text{ES}}$ to CFGs)

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

The key idea is defining a variable $A^X_{i,j}$ 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:

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

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

$$S \rightarrow A^Z_{0,0} \mid A^Z_{0,1} \mid \cdots \mid A^Z_{0,n-1}$$

PLRG

Theorem (PDA_{ES} to CFGs)

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

The key idea is defining a variable $A^X_{i,j}$ 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:

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

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

$$S \rightarrow A^Z_{0,0} \mid A^Z_{0,1} \mid \cdots \mid A^Z_{0,n-1}$$

Then, how to define production rules for $A^X_{i,j}$?

PDA_{ES} to CFGs

PLRG

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

PDA_{ES} to CFGs

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

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

PDA$_{\text{ES}}$ to CFGs

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 $X_1 \cdots X_m$.

PDA_{ES} to CFGs

We can define production rules for $A^X_{i,j}$ 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.

PDA_{ES} to CFGs

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.

$\mathrm{PDA}_{\mathrm{ES}}$ to CFGs

We can define production rules for $A^X_{i,j}$ 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 $X_1 \cdots X_m$.

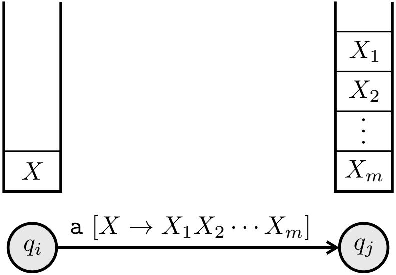
Then, we need to pop $X_1, \cdots, X_m$ from the stack to make the stack empty.

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^X_{i,k_m} as follows:

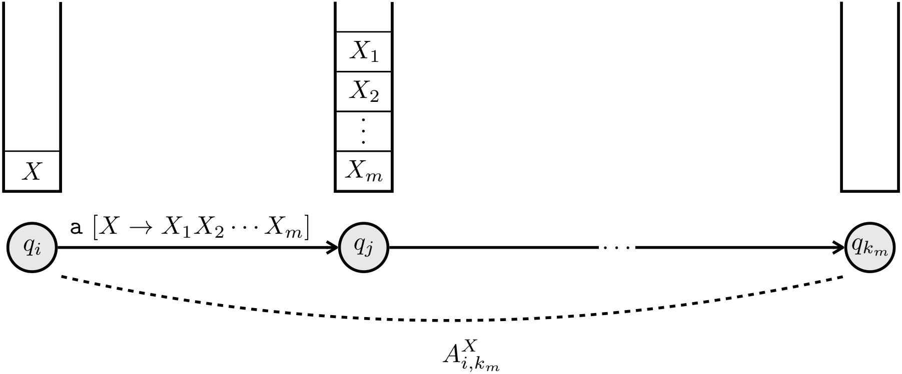
$$A^X_{i,k_m} \rightarrow a \; A^{X_1}_{j,k_1} \; A^{X_2}_{k_1,k_2} \cdots A^{X_m}_{k_{m-1},k_m} \text{ for all } 0 \leq k_1, \cdots, k_m < n$$

PDA$_{\text{ES}}$ to CFGs



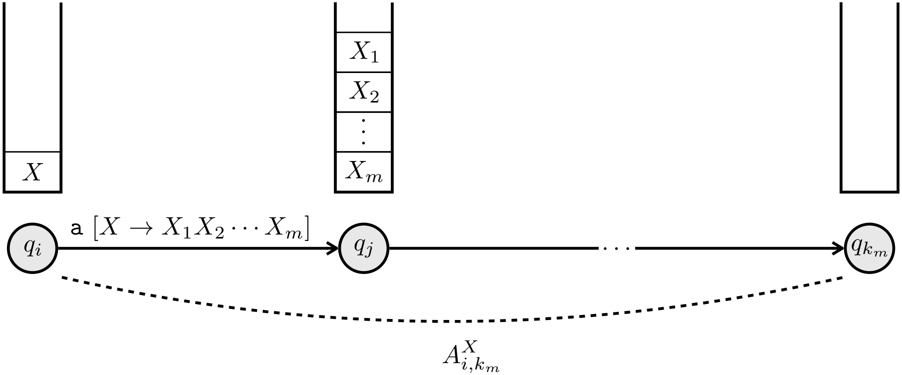
Consider a transition $(q_j, X_1 \cdots X_m) \in \delta(q_i, a, X)$.

PDA_{ES} to CFGs



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

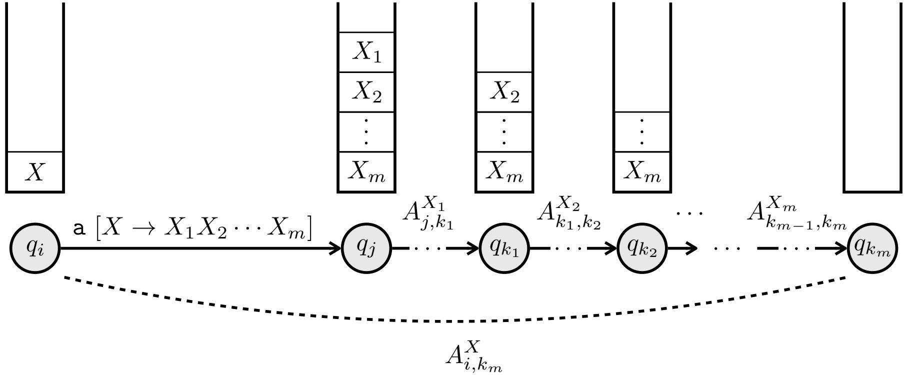
PDA_{ES} to CFGs



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

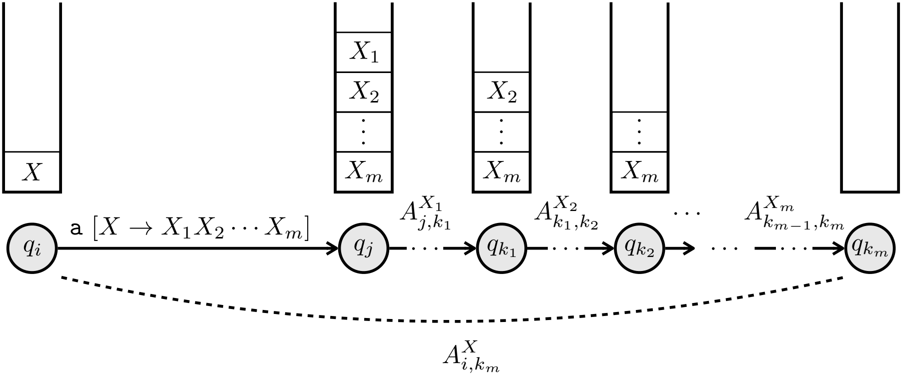
Then, we need to pop $X_1, \cdots, X_m$ from the stack to make the stack empty when moving from q_j to q_{k_m}.

PDA$_{ES}$ to CFGs



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

PDA$_{\text{ES}}$ to CFGs



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

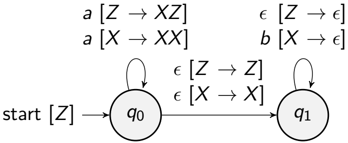
Thus, we get the following production rule for A^X_{i,k_m}:

$$A^X_{i,k_m} \to a\ A^{X_1}_{j,k_1}\ A^{X_2}_{k_1,k_2} \cdots A^{X_m}_{k_{m-1},k_m} \text{ for all } 0 \leq k_1, \cdots, k_m < n$$

PDA_{ES} to CFGs – Example

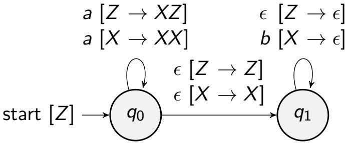
$$S \to A^{Z}_{0,j} \qquad\qquad A^{X}_{i,k_m} \to a\ A^{X_1}_{j,k_1}\ A^{X_2}_{k_1,k_2} \cdots A^{X_m}_{k_{m-1},k_m}$$

Consider the following PDA (by empty stacks):



$$S \to A^{Z}_{0,j} \qquad\qquad A^{X}_{i,k_m} \to a\ A^{X_1}_{j,k_1}\ A^{X_2}_{k_1,k_2} \cdots A^{X_m}_{k_{m-1},k_m}$$

Consider the following PDA (by empty stacks):



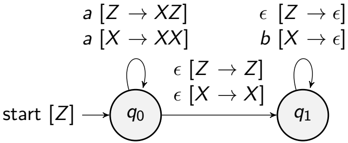
Then, the equivalent CFG is:

$$
\begin{array}{lcl}
S & \to & A^{Z}_{0,0} \mid A^{Z}_{0,1} \\
A^{Z}_{0,0} & \to & \text{a } A^{X}_{0,0}\ A^{Z}_{0,0} \mid \text{a } A^{X}_{0,1}\ A^{Z}_{1,0} \mid A^{Z}_{1,0} \\
A^{Z}_{0,1} & \to & \text{a } A^{X}_{0,0}\ A^{Z}_{0,1} \mid \text{a } A^{X}_{0,1}\ A^{Z}_{1,1} \mid A^{Z}_{1,1} \\
A^{X}_{0,0} & \to & \text{a } A^{X}_{0,0}\ A^{X}_{0,0} \mid \text{a } A^{X}_{0,1}\ A^{X}_{1,0} \mid A^{X}_{1,0} \\
A^{X}_{0,1} & \to & \text{a } A^{X}_{0,0}\ A^{X}_{0,1} \mid \text{a } A^{X}_{0,1}\ A^{X}_{1,1} \mid A^{X}_{1,1} \\
A^{Z}_{1,1} & \to & \epsilon \\
A^{X}_{1,1} & \to & b
\end{array}
$$

PDA_{ES} to CFGs – Example

$$S \to A^{Z}_{0,j} \qquad\qquad A^{X}_{i,k_m} \to a\ A^{X_1}_{j,k_1}\ A^{X_2}_{k_1,k_2} \cdots A^{X_m}_{k_{m-1},k_m}$$

Consider the following PDA (by empty stacks):



Then, the equivalent CFG is:

$$\begin{array}{lcl}
S & \to & A^{Z}_{0,0} \mid A^{Z}_{0,1} \\
A^{Z}_{0,0} & \to & \mathtt{a}\ A^{X}_{0,0}\ A^{Z}_{0,0} \mid \mathtt{a}\ A^{X}_{0,1}\ A^{Z}_{1,0} \mid A^{Z}_{1,0} \\
A^{Z}_{0,1} & \to & \mathtt{a}\ A^{X}_{0,0}\ A^{Z}_{0,1} \mid \mathtt{a}\ A^{X}_{0,1}\ A^{Z}_{1,1} \mid A^{Z}_{1,1} \\
A^{X}_{0,0} & \to & \mathtt{a}\ A^{X}_{0,0}\ A^{X}_{0,0} \mid \mathtt{a}\ A^{X}_{0,1}\ A^{X}_{1,0} \mid A^{X}_{1,0} \\
A^{X}_{0,1} & \to & \mathtt{a}\ A^{X}_{0,0}\ A^{X}_{0,1} \mid \mathtt{a}\ A^{X}_{0,1}\ A^{X}_{1,1} \mid A^{X}_{1,1} \\
A^{Z}_{1,1} & \to & \epsilon \\
A^{X}_{1,1} & \to & b
\end{array}$$

$$\begin{array}{lcl}
S & \Rightarrow & A^{Z}_{0,1} \\
 & \Rightarrow & \mathtt{a}\ A^{X}_{0,1}\ A^{Z}_{1,1} \\
 & \Rightarrow & \mathtt{aa}\ A^{X}_{0,1}\ A^{X}_{1,1}\ A^{Z}_{1,1} \\
 & \Rightarrow & \mathtt{aa}\ A^{X}_{1,1}\ A^{X}_{1,1}\ A^{Z}_{1,1} \\
 & \Rightarrow & \mathtt{aab}\ A^{X}_{1,1}\ A^{Z}_{1,1} \\
 & \Rightarrow & \mathtt{aabb}\ A^{Z}_{1,1} \\
 & \Rightarrow & \mathtt{aabb}
\end{array}$$

Summary

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

$$\text{PDA}_{\text{FS}} \rightleftarrows \text{PDA}_{\text{ES}} \rightleftarrows \text{CFG}$$

(by final states) (by empty stacks)

Next Lecture

- Deterministic Pushdown Automata (DPDA)

Jihyeok Park
jihyeok_park@korea.ac.kr
https://plrg.korea.ac.kr