Lecture 15 – Examples of Pushdown Automata

COSE215: Theory of Computation

Jihyeok Park

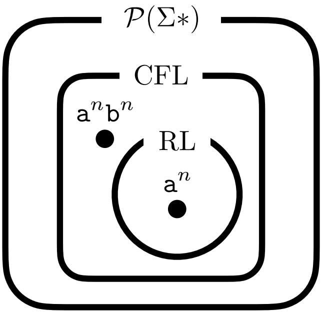
PLRG

2025 Spring

Recall

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

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



Languages	Automata	Grammars
Context-Free Language **(CFL)**	Pushdown Automata **(PDA)**	Context-Free Grammar **(CFG)**
Regular Language **(RL)**	Finite Automata **(FA)**	Regular Expression **(RE)**

Contents

Example 1: $\mathtt{a}^n\mathtt{b}^n$

Construct a PDA that accepts the language by **final states**:

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

Example 1: $a^n b^n$

Construct a PDA that accepts the language by **final states**:

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

The key idea is to **count** the number of a's using the stack.

Example 1: $\mathtt{a}^n\mathtt{b}^n$

Construct a PDA that accepts the language by **final states**:

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

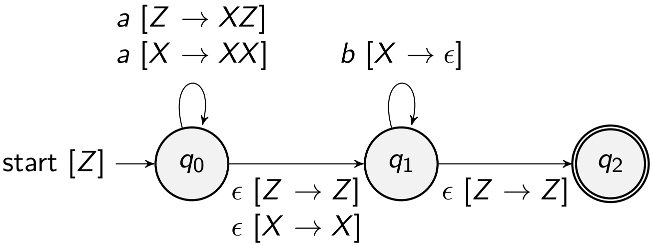
The key idea is to **count** the number of `a`'s using the stack.

1. Start with the stack only having the initial stack alphabet Z.
2. Repeatedly **push** X onto the stack for each `a`.
3. Repeatedly **pop** X from the stack for each `b`.
4. Accept when the top of the stack is Z.

Example 1: $a^n b^n$

Construct a PDA that accepts the language by **final states**:

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



https://plrg.korea.ac.kr/courses/cose215/materials/pda-an-bn-final.pdf

Example 2: $\mathtt{a}^n\mathtt{b}^{2n}$

Construct a PDA that accepts the language by **final states**:

$$L_F(P) = \{\mathtt{a}^n\mathtt{b}^{2n} \mid n \geq 0\}$$

Example 2: $\mathtt{a}^n\mathtt{b}^{2n}$

Construct a PDA that accepts the language by **final states**:

$$L_F(P) = \{\mathtt{a}^n\mathtt{b}^{2n} \mid n \geq 0\}$$

Now, we need to push **two** X's for each $\mathtt{a}$.

Example 2: $\mathtt{a}^n\mathtt{b}^{2n}$

Construct a PDA that accepts the language by **final states**:

$$L_F(P) = \{\mathtt{a}^n\mathtt{b}^{2n} \mid n \geq 0\}$$

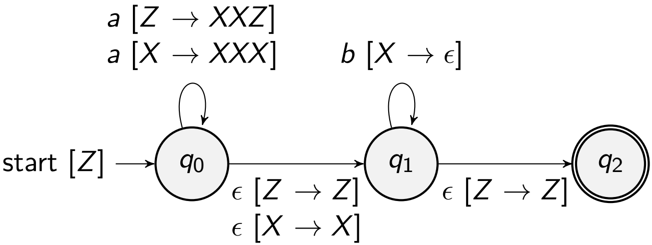
Now, we need to push **two** X's for each a.

1. Start with the stack only having the initial stack alphabet Z.
2. Repeatedly **push two** X's onto the stack for each a.
3. Repeatedly **pop** X from the stack for each b.
4. Accept when the top of the stack is Z.

Example 2: $a^n b^{2n}$

Construct a PDA that accepts the language by **final states**:

$$L_F(P) = \{a^n b^{2n} \mid n \geq 0\}$$



https://plrg.korea.ac.kr/courses/cose215/materials/pda-an-b2n-final.pdf

Example 3: ww^R

Construct a PDA that accepts the language by **final states**:

$$L_F(P) = \{ww^R \mid w \in \{\mathtt{a}, \mathtt{b}\}^*\}$$

Example 3: ww^R

Construct a PDA that accepts the language by **final states**:

$$L_F(P) = \{ww^R \mid w \in \{\mathtt{a}, \mathtt{b}\}^*\}$$

The key idea is to **store** the first half of the word and **compare** it with the second half in reverse order using the stack.

Example 3: ww^R

Construct a PDA that accepts the language by **final states**:

$$L_F(P) = \{ww^R \mid w \in \{\mathtt{a}, \mathtt{b}\}^*\}$$

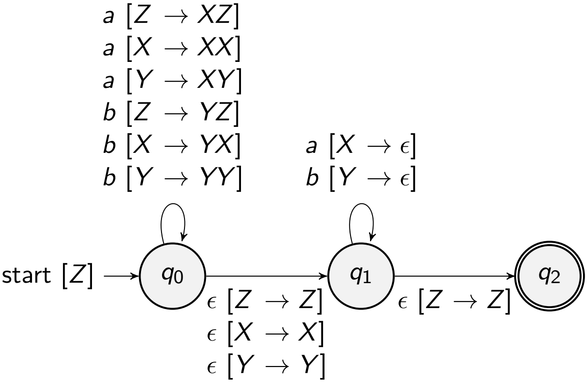
The key idea is to **store** the first half of the word and **compare** it with the second half in reverse order using the stack.

1. Start with the stack only having the initial stack alphabet Z.
2. Repeatedly **push** X (or Y) onto the stack for each a (or b).
3. Repeatedly **pop** X (or Y) from the stack for each a (or b).
4. Accept when the top of the stack is Z.

Example 3: ww^R

Construct a PDA that accepts the language by **final states**:

$$L_F(P) = \{ww^R \mid w \in \{\mathtt{a}, \mathtt{b}\}^*\}$$



https://plrg.korea.ac.kr/courses/cose215/materials/pda-w-wr-final.pdf

Example 4: Balanced Parentheses

Construct a PDA that accepts the language by **empty stacks**:

$$L_E(P) = \{w \in \{\,(,\,)\,\}^* \mid w \text{ is balanced}\}$$

Example 4: Balanced Parentheses

Construct a PDA that accepts the language by **empty stacks**:

$$L_E(P) = \{w \in \{\,(\,,\,)\,\}^* \mid w \text{ is balanced}\}$$

The key idea is to **count** the number of **unmatched** open parentheses.

Example 4: Balanced Parentheses

Construct a PDA that accepts the language by **empty stacks**:

$$L_E(P) = \{w \in \{(,)\}^* \mid w \text{ is balanced}\}$$

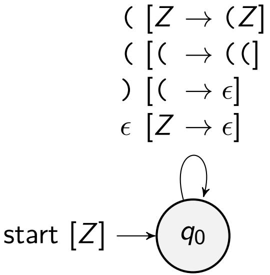
The key idea is to **count** the number of **unmatched** open parentheses.

1. Start with the stack only having the initial stack alphabet Z.
2. If the current symbol is (, push (onto the stack.
3. If the current symbol is), pop (from the stack.
4. Repeat steps 2 and 3.
5. Accept when the top of the stack is Z.

Example 4: Balanced Parentheses

Construct a PDA that accepts the language by **empty stacks**:

$$L_E(P) = \{w \in \{(,)\}^* \mid w \text{ is balanced}\}$$



https://plrg.korea.ac.kr/courses/cose215/materials/pda-balanced-empty.pdf

Example 5: Equal Number of a's and b's

Construct a PDA that accepts the language by **empty stacks**:

$$L_E(P) = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid N_{\mathtt{a}}(w) = N_{\mathtt{b}}(w)\}$$

where $N_{\mathtt{a}}(w)$ and $N_{\mathtt{b}}(w)$ are the number of a's and b's in w, respectively.

Example 5: Equal Number of $\texttt{a}$'s and $\texttt{b}$'s

Construct a PDA that accepts the language by **empty stacks**:

$$L_E(P) = \{w \in \{\texttt{a}, \texttt{b}\}^* \mid N_{\texttt{a}}(w) = N_{\texttt{b}}(w)\}$$

where $N_{\texttt{a}}(w)$ and $N_{\texttt{b}}(w)$ are the number of $\texttt{a}$'s and $\texttt{b}$'s in w, respectively.

Consider the following function $f : \{\texttt{a}, \texttt{b}\}^* \to \mathbb{N}$:

$$f(w) = N_{\texttt{a}}(w) - N_{\texttt{b}}(w)$$

Example 5: Equal Number of a's and b's

Construct a PDA that accepts the language by **empty stacks**:

$$L_E(P) = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid N_{\mathtt{a}}(w) = N_{\mathtt{b}}(w)\}$$

where $N_{\mathtt{a}}(w)$ and $N_{\mathtt{b}}(w)$ are the number of a's and b's in w, respectively.

Consider the following function $f : \{\mathtt{a}, \mathtt{b}\}^* \to \mathbb{N}$:

$$f(w) = N_{\mathtt{a}}(w) - N_{\mathtt{b}}(w)$$

The key idea is to represent the **positive value** of $f(w)$ using the number of P's and the **negative value** of $f(w)$ using the number of N's.

Example 5: Equal Number of a's and b's

Construct a PDA that accepts the language by **empty stacks**:

$$L_E(P) = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid N_{\mathtt{a}}(w) = N_{\mathtt{b}}(w)\}$$

where $N_{\mathtt{a}}(w)$ and $N_{\mathtt{b}}(w)$ are the number of a's and b's in w, respectively.

Consider the following function $f : \{\mathtt{a}, \mathtt{b}\}^* \to \mathbb{N}$:

$$f(w) = N_{\mathtt{a}}(w) - N_{\mathtt{b}}(w)$$

The key idea is to represent the **positive value** of $f(w)$ using the number of P's and the **negative value** of $f(w)$ using the number of N's.

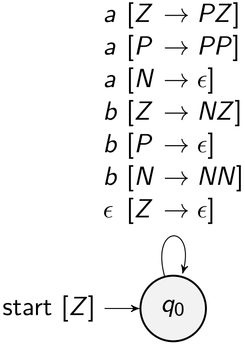
1. Start with the stack only having the initial stack alphabet Z.
2. If the current symbol is a, **push** P or **pop** N.
3. If the current symbol is b, **push** N or **pop** P.
4. Repeat steps 2 and 3.
5. Accept when the top of the stack is Z.

Example 5: Equal Number of a's and b's

Construct a PDA that accepts the language by **empty stacks**:

$$L_E(P) = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid N_\mathtt{a}(w) = N_\mathtt{b}(w)\}$$

where $N_\mathtt{a}(w)$ and $N_\mathtt{b}(w)$ are the number of a's and b's in w, respectively.



https://plrg.korea.ac.kr/courses/cose215/materials/pda-eq-a-b-empty.pdf

Example 6: Unequal Number of a's and b's

Construct a PDA that accepts the language by **empty stacks**:

$$L_E(P) = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid N_{\mathtt{a}}(w) \neq N_{\mathtt{b}}(w)\}$$

where $N_{\mathtt{a}}(w)$ and $N_{\mathtt{b}}(w)$ are the number of a's and b's in w, respectively.

Example 6: Unequal Number of a's and b's

Construct a PDA that accepts the language by **empty stacks**:

$$L_E(P) = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid N_{\mathtt{a}}(w) \neq N_{\mathtt{b}}(w)\}$$

where $N_{\mathtt{a}}(w)$ and $N_{\mathtt{b}}(w)$ are the number of a's and b's in w, respectively.

The key idea is same but we accept the top of the stack is P or N.

Example 6: Unequal Number of a's and b's

Construct a PDA that accepts the language by **empty stacks**:

$$L_E(P) = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid N_{\mathtt{a}}(w) \neq N_{\mathtt{b}}(w)\}$$

where $N_{\mathtt{a}}(w)$ and $N_{\mathtt{b}}(w)$ are the number of a's and b's in w, respectively.

The key idea is same but we accept the top of the stack is P or N.

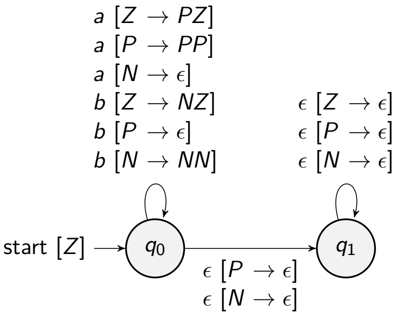
1. Start with the stack only having the initial stack alphabet Z.
2. If the current symbol is a, **push** P or **pop** N.
3. If the current symbol is b, **push** N or **pop** P.
4. Repeat steps 2 and 3.
5. Accept when the top of the stack is P or N.

Example 6: Unequal Number of a's and b's

Construct a PDA that accepts the language by **empty stacks**:

$$L_E(P) = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid N_{\mathtt{a}}(w) \neq N_{\mathtt{b}}(w)\}$$

where $N_{\mathtt{a}}(w)$ and $N_{\mathtt{b}}(w)$ are the number of a's and b's in w, respectively.



https://plrg.korea.ac.kr/courses/cose215/materials/pda-uneq-a-b-empty.pdf

Example 7: Not of the Form ww

Construct a PDA that accepts the language by **empty stacks**:

$$L_E(P) = \{x \in \{\mathtt{a}, \mathtt{b}\}^* \mid x \text{ is not of the form } ww\}$$

Example 7: Not of the Form ww

Construct a PDA that accepts the language by **empty stacks**:

$$L_E(P) = \{x \in \{\mathtt{a}, \mathtt{b}\}^* \mid x \text{ is not of the form } ww\}$$

There are two cases of $x \in L_E(P)$:

1. x is an **odd-length** word or
2. x is divided into two **same-length** but **unequal** words.

Example 7: Not of the Form ww

Construct a PDA that accepts the language by **empty stacks**:

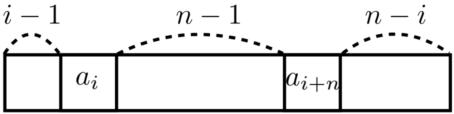
$$L_E(P) = \{x \in \{\mathtt{a}, \mathtt{b}\}^* \mid x \text{ is not of the form } ww\}$$

There are two cases of $x \in L_E(P)$:

1. x is an **odd-length** word or
2. x is divided into two **same-length** but **unequal** words.

In the second case, assume $x = a_1 \cdots a_{2n}$. Then,

$$\exists 1 \leq i \leq n.\ a_i \neq a_{i+n}$$



Example 7: Not of the Form ww

Construct a PDA that accepts the language by **empty stacks**:

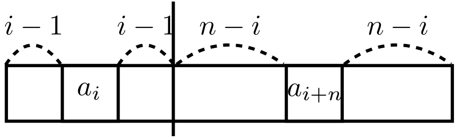
$$L_E(P) = \{x \in \{\mathtt{a}, \mathtt{b}\}^* \mid x \text{ is not of the form } ww\}$$

There are two cases of $x \in L_E(P)$:

1. x is an **odd-length** word or
2. x is divided into two **same-length** but **unequal** words.

In the second case, assume $x = a_1 \cdots a_{2n}$. Then,

$$\exists 1 \leq i \leq n.\ a_i \neq a_{i+n}$$



Example 7: Not of the Form ww

Construct a PDA that accepts the language by **empty stacks**:

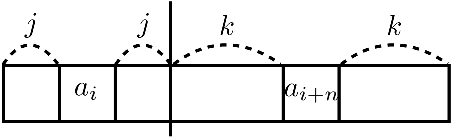
$$L_E(P) = \{x \in \{\mathtt{a}, \mathtt{b}\}^* \mid x \text{ is not of the form } ww\}$$

There are two cases of $x \in L_E(P)$:

1. x is an **odd-length** word or
2. x is divided into two **same-length** but **unequal** words.

In the second case, assume $x = a_1 \cdots a_{2n}$. Then,

$$\exists 1 \leq i \leq n.\ a_i \neq a_{i+n}$$



Example 7: Not of the Form ww

Construct a PDA that accepts the language by **empty stacks**:

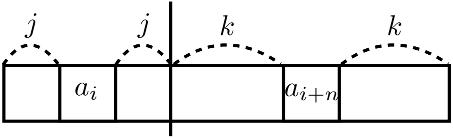
$$L_E(P) = \{x \in \{\mathtt{a}, \mathtt{b}\}^* \mid x \text{ is not of the form } ww\}$$

There are two cases of $x \in L_E(P)$:

1. x is an **odd-length** word or
2. x is divided into two **odd-length** words whose **centers** are different.

In the second case, assume $x = a_1 \cdots a_{2n}$. Then,

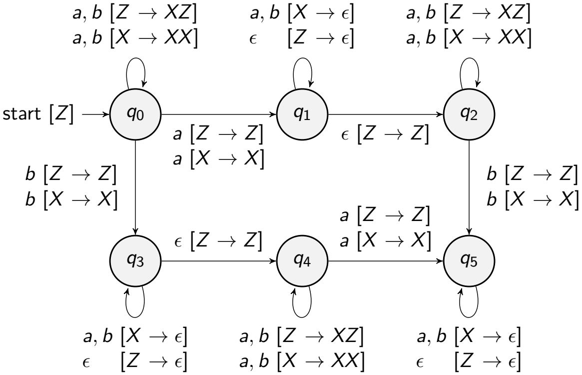
$$\exists 1 \leq i \leq n.\ a_i \neq a_{i+n}$$



Example 7: Not of the Form ww

Construct a PDA that accepts the language by **empty stacks**:

$$L_E(P) = \{x \in \{\mathtt{a}, \mathtt{b}\}^* \mid x \text{ is not of the form } ww\}$$



https://plrg.korea.ac.kr/courses/cose215/materials/pda-not-w-w-empty.pdf

Summary

1. Examples of Pushdown Automata

   Example 1: $\mathtt{a}^n\mathtt{b}^n$

   Example 2: $\mathtt{a}^n\mathtt{b}^{2n}$

   Example 3: ww^R

   Example 4: Balanced Parentheses

   Example 5: Equal Number of a's and b's

   Example 6: Unequal Number of a's and b's

   Example 7: Not of the Form ww

Homework #4

- Please see this document on GitHub:

https://github.com/ku-plrg-classroom/docs/tree/main/cose215/pda-examples

- The due date is 23:59 on May 19 (Mon.).

- Please only submit `Implementation.scala` file to **LMS**.

Next Lecture

- Equivalence of Pushdown Automata and Context-Free Grammars

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