Theoretical Computer Science

Tutorial Week 6

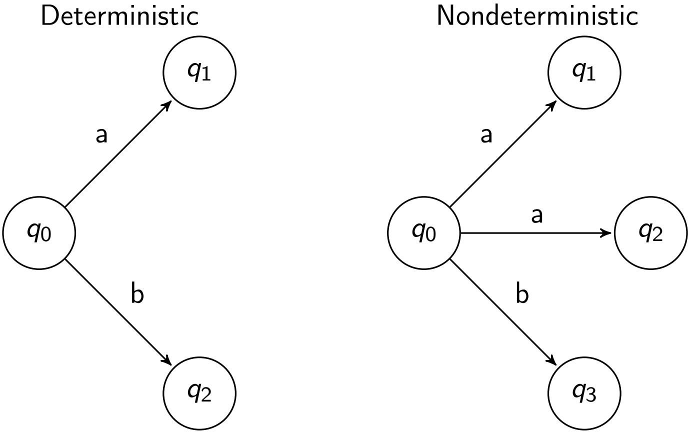
Prof. Andrey Frolov

INNOPOLIS UNIVERSITY

Agenda

- **Determinism vs Non-determinism**
  - Definition of NDFSA
  - Examples
- Deterministic Pushdown Automata
  - Definition
  - Examples
  - Pumping lemma for PDA
- Nondeterministic Pushdown Automata

Determinism vs Non-determinism



Agenda

- Determinism vs Non-determinism
  - **Definition of NDFSA**
  - Examples
- Deterministic Pushdown Automata
  - Definition
  - Examples
  - Pumping lemma for PDA
- Nondeterministic Pushdown Automata

Non-deterministic Finite State Automata (NDFSA)

Definition: NDFSA

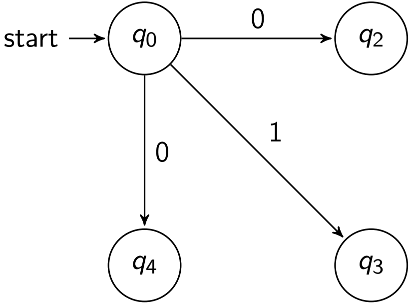
A NDFSA is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where Q, Σ, q_0, A are defined as in (D)FSA and the transition function is defined as

$$\delta : Q \times \Sigma \rightarrow \mathbb{P}(Q)$$

$\mathbb{P}$ is the power set (i.e., the set of all possible subsets)

A NDFSA modifies the definition of a FSA to permit transitions at each stage to either zero, one, or more than one states.

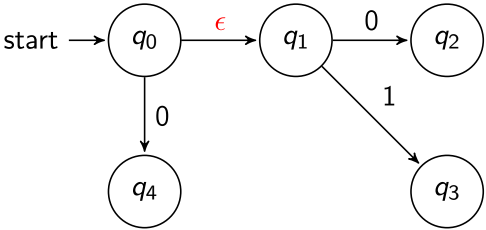
Example



δ	0	1
$\rightarrow q_0$	$\{q_2, q_4\}$	q_3
q_1	$\emptyset$	$\emptyset$
q_2	$\emptyset$	$\emptyset$
q_3	$\emptyset$	$\emptyset$

NDFSA with ϵ

What about ϵ-transition???



NDFSA with ϵ

Could we add ϵ?

Definition: NDFSA

A NDFSA is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where Q, Σ, q_0, A are defined as in (D)FSA and the transition function is defined as

$$\delta : Q \times \Sigma \cup \{\epsilon\}??? \rightarrow \mathbb{P}(Q)$$

$\mathbb{P}$ is the powerset function (i.e., the set of all possible subsets)

NDFSA with ϵ

Could we add ϵ?

Definition: NDFSA

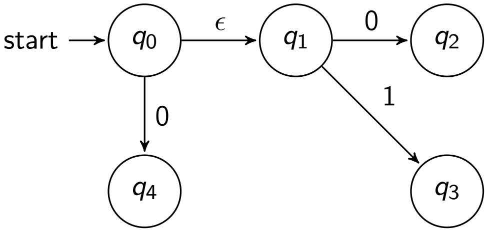
A NDFSA is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where Q, Σ, q_0, A are defined as in (D)FSA and the transition function is defined as

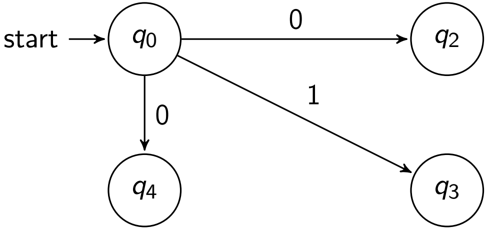
$$\delta : Q \times \Sigma \cup \{\epsilon\}??? \rightarrow \mathbb{P}(Q)$$

$\mathbb{P}$ is the powerset function (i.e., the set of all possible subsets)

Yes, but it is not necessary!

Example with ϵ



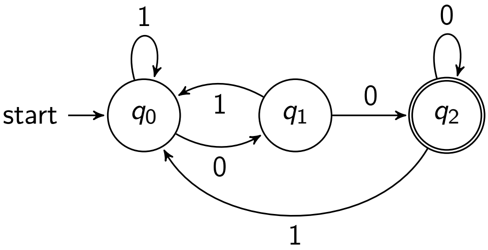


Agenda

- Determinism vs Non-determinism
  - Definition of NDFSA
  - **Examples**
- Deterministic Pushdown Automata
  - Definition
  - Examples
  - Pumping lemma for PDA
- Nondeterministic Pushdown Automata

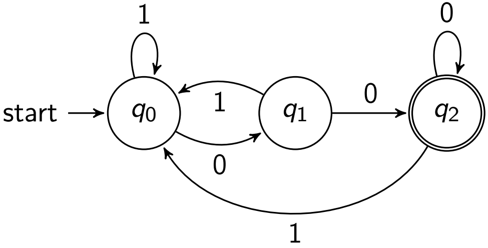
FSA vs NDFSA

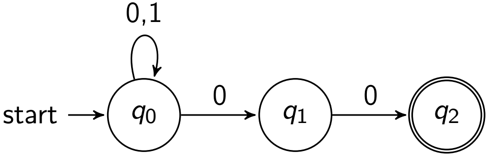
The FSA and NDFSA accepting strings ending with 00



FSA vs NDFSA

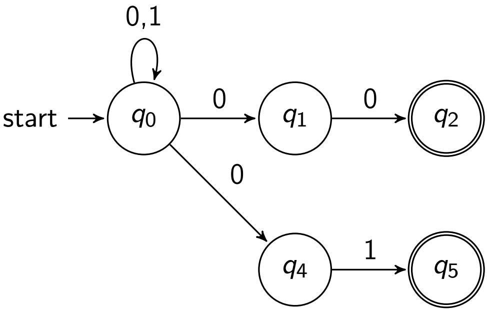
The FSA and NDFSA accepting strings ending with 00





Example

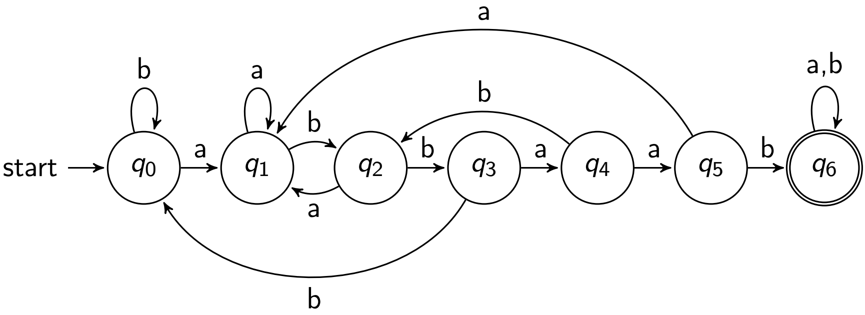
The NDFSA accepting strings ending with 00 or 01



FSA vs NDFSA

Let Σ be the alphabet $\Sigma = \{a, b\}$

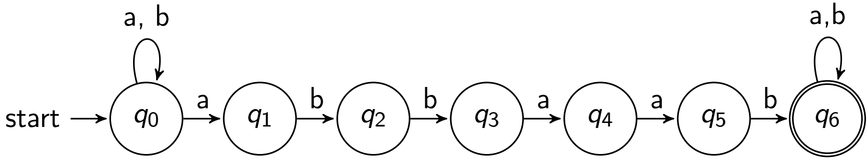
- $L = \{x \in \Sigma^* \mid \; x \text{ contains the substring } abbaab\}$;



FSA vs NDFSA

Let Σ be the alphabet $\Sigma = \{a, b\}$

- $L = \{x \in \Sigma^* \mid \; x \text{ contains the substring } abbaab\}$;



Agenda

- Determinism vs Non-determinism
  - Definition of NDFSA
  - Examples
- Deterministic Pushdown Automata
  - **Definition**
  - Examples
  - Pumping lemma for PDA
- Nondeterministic Pushdown Automata

DFSA (Formal definition)

Definition

A Deterministic Finite State Automaton (DFSA) is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where

Q is a finite set of *states*;
Σ is a finite *input alphabet*;
$q_0 \in Q$ is the *initial* state;
$A \subseteq Q$ is the set of *accepting* states;
$\delta : Q \times \Sigma \to Q$ is a *transition* function.

PDA (Formal Definition)

Definition

A (Deterministic) Pushdown Automaton (PDA) is a tuple $\langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, A\rangle$, where

- Q is a finite set of states;
- Σ and Γ are the input and stack (finite) alphabets;
- $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to Q \times \Gamma^*$ is the (partial) transition function;
- $q_0 \in Q$ is the initial state;
- $Z_0 \in \Gamma$ is the initial stack symbol;
- $A \subseteq Q$ is the set of accepting states.

Using the ϵ-rule below!

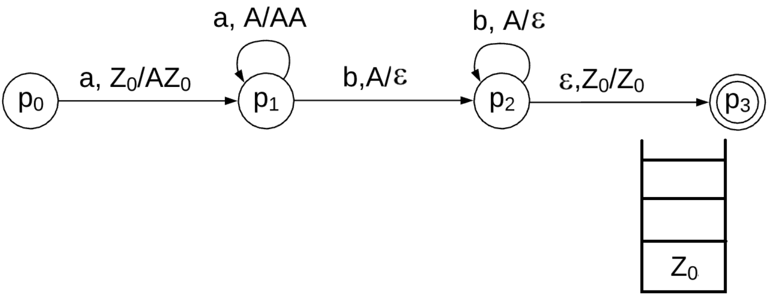
Pushdown automata

The ϵ-rule

If there is an epsilon transition from a state q, there must not be any alphabet transition from that state.

Usually, we use only $\epsilon, Z_0/Z_0$-transition at the end of PDA.

Pushdown automata

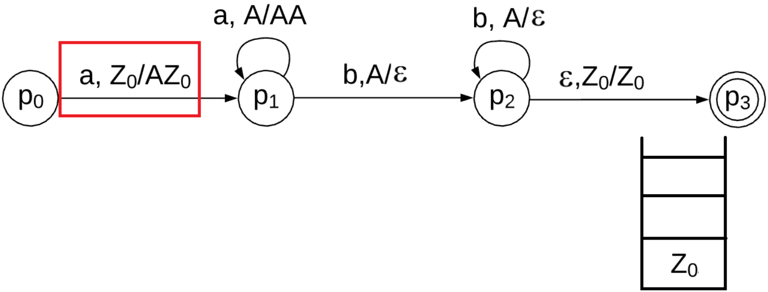


$\Gamma = \{Z_0, A\}$, Z_0 is the initial stack symbol.

Pushdown automata

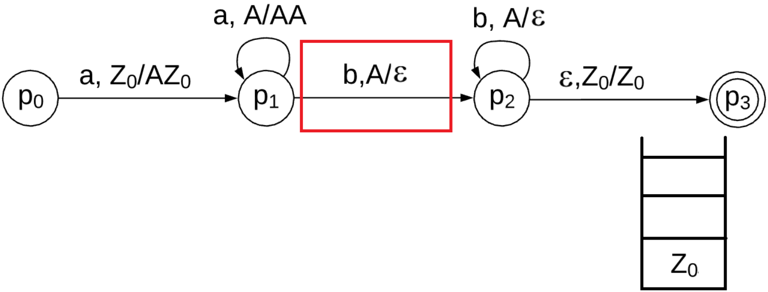
Conditions on Z_0

- the stack contains at least one symbol: Z_0;
- Z_0 is never removed;
- no additional copies of Z_0 are pushed onto the stack.



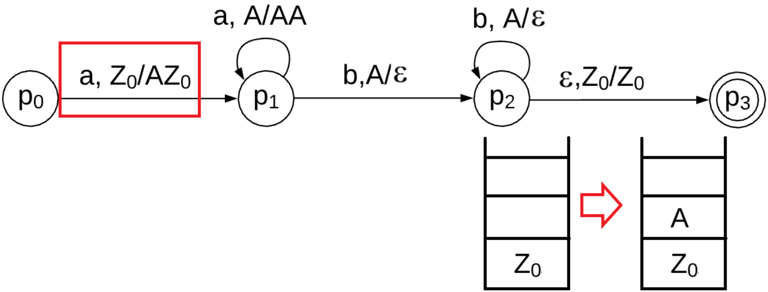
$$\delta(p_0, a, Z_0) = (p_1, AZ_0)$$

Pushdown automata



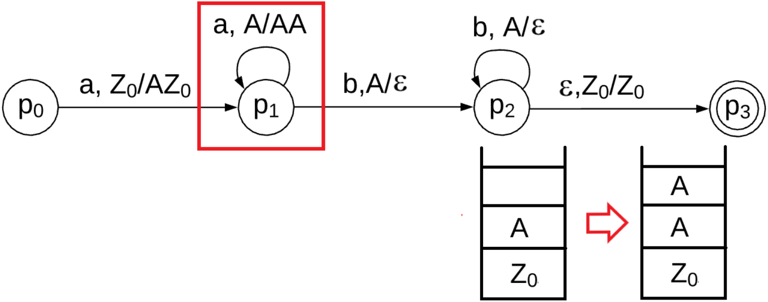
$$\delta(p_1, b, A) = (p_2, \epsilon)$$

Pushdown automata

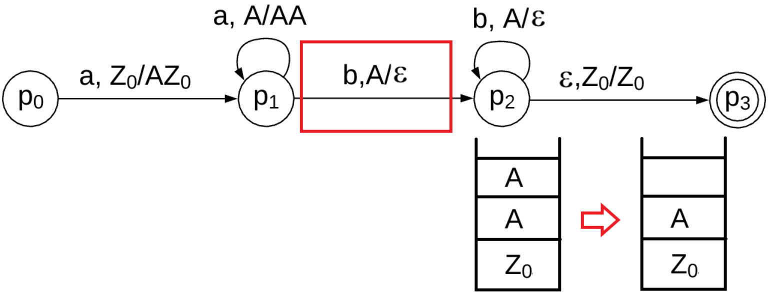


aabb

Pushdown automata

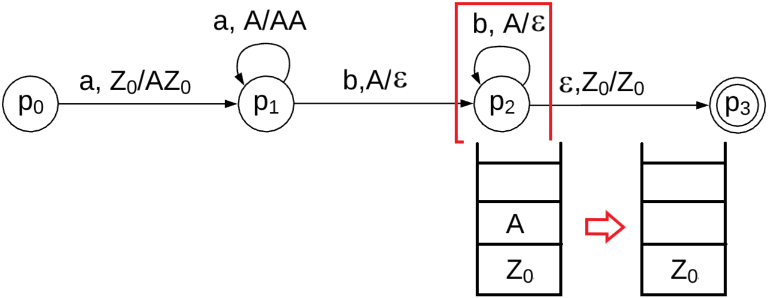


aabb



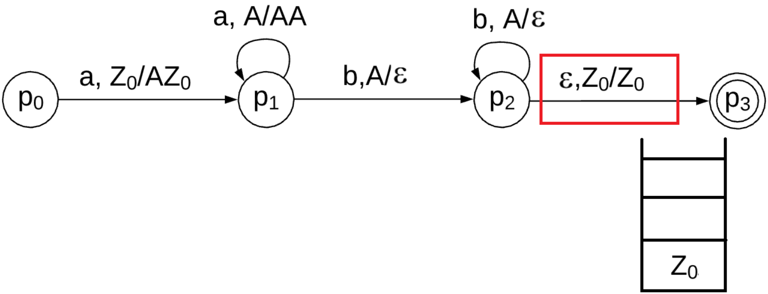
aabb

Pushdown automata



aabb

Pushdown automata



$aabb \in L_1$

PDA. Transition

Definition

A tuple (q, x, Z) is called **configuration**, where $q \in Q, x \in \Sigma^*, Z \in \Gamma^*$.

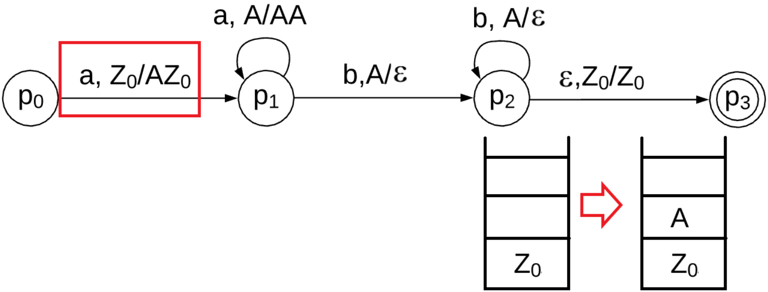
Definition

For a PDA $\langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, A \rangle$,
if $\delta(q, a, Z) = (q', Z')$ then

$$(q, ax, Z\gamma) \vdash \left(q', x, Z'\gamma\right),$$

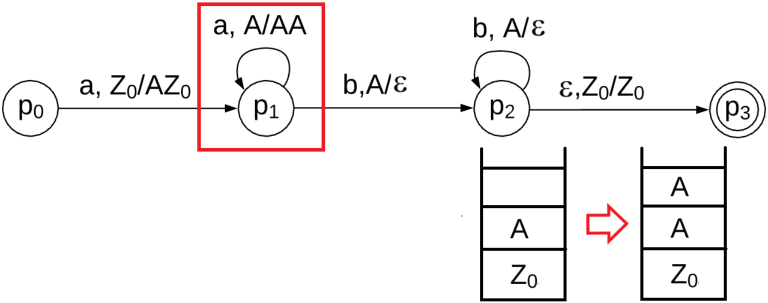
where $a \in \Sigma, x \in \Sigma^*, Z \in \Gamma, Z' \in \Gamma^*$.

Pushdown automata



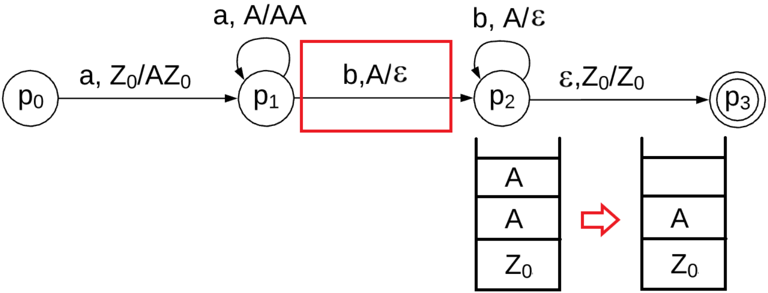
$(p_0, {\color{red}a}abb, Z_0) \vdash (p_1, abb, AZ_0)$

Pushdown automata



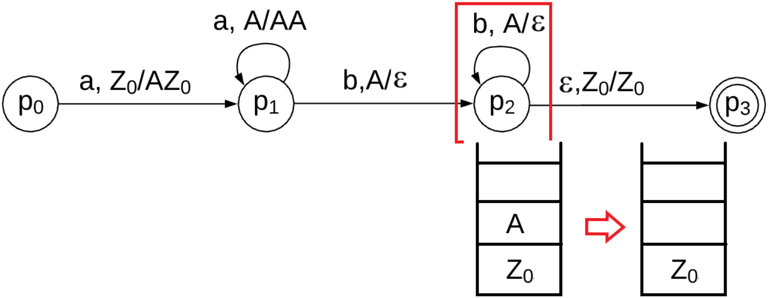
$(p_0, aabb, Z_0) \vdash (p_1, \textcolor{red}{a}bb, AZ_0) \vdash (p_1, bb, AAZ_0)$

Pushdown automata



$$(p_0, aabb, Z_0) \vdash (p_1, abb, AZ_0) \vdash (p_1, \textcolor{red}{b}b, AAZ_0) \vdash (p_2, b, AZ_0)$$

Pushdown automata



$(p_0, aabb, Z_0) \vdash (p_1, abb, AZ_0) \vdash (p_1, bb, AAZ_0) \vdash (p_2, \textcolor{red}{b}, AZ_0) \vdash (p_3, \epsilon, Z_0)$

PDA. Recognized Languages

Definition

For configurations $c_1, c_2, \dots c_k$, if

$$c_1 \vdash c_2 \vdash \cdots \vdash c_k,$$

then we define

$$c_1 \vdash^* c_k$$

PDA. Recognized Languages

Definition

For configurations $c_1, c_2, \ldots c_k$, if

$$c_1 \vdash c_2 \vdash \cdots \vdash c_k,$$

then we define

$$c_1 \vdash^* c_k$$

Definition

A language L is recognized by a PDA $\langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, A \rangle$, if

$$L = \{x \in \Sigma^* \mid (q_0, x, Z_0) \vdash^* (q, \epsilon, \gamma), \text{ where } q \in A, \gamma \in \Gamma^*\}$$

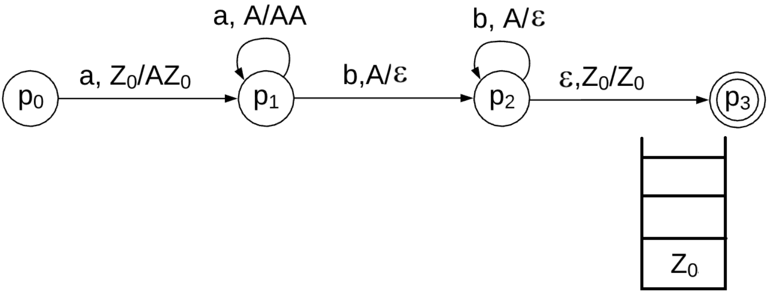
Agenda

- Deterministic Pushdown Automata
  - Definition
  - **Examples**
  - Pumping lemma for PDA
- Nondeterministic Pushdown Automata

PDA. Examples

Example 1

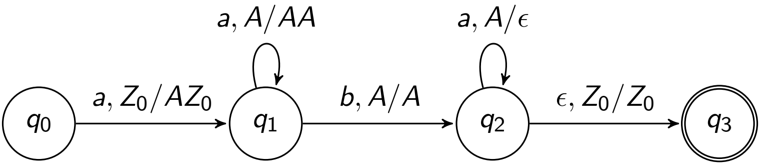
$L_1 = \{a^n b^n \mid n \geq 1\}$ is not regular, but is recognized by a PDA.



PDA. Examples

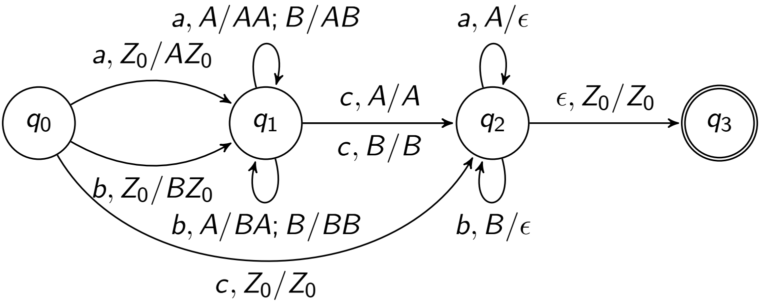
Example 2

$L_2 = \{a^n b a^n \mid n \in \mathbb{N}\}$ is not regular, but is recognized by a PDA.



Example 3

$L_3 = \{vcv^R \mid v \in \{a,b\}^*\}$ is not regular, but is recognized by a PDA.



Agenda

- Deterministic Pushdown Automata
  - Definition
  - Examples
  - **Pumping lemma for PDA**
- Nondeterministic Pushdown Automata

Pumping lemma for FSA

Pumping lemma

If $L \subseteq \Sigma^*$ is a regular language then there exists $m \geq 1$ such that any $w \in L$ with $|w| \geq m$ can be represented as $w = xyz$ such that

- $y \neq \epsilon$,
- $|xy| \leq m$,
- $xy^i z \in L$ for any $i \geq 0$.

Pumping lemma. For practice

Corollary

If for any $m \geq 1$ there is $w \in L$ such that $|w| \geq m$ and for any representation $w = xyz$ with $y \neq \epsilon$ and $|xy| \leq m$

$$xy^i z \notin L \text{ for some } i \geq 0.$$

Then L is not a regular language.

Pumping lemma. For practice

Corollary

If for any $m \geq 1$ there is $w \in L$ such that $|w| \geq m$ and for any representation $w = xyz$ with $y \neq \epsilon$ and $|xy| \leq m$

$$xy^i z \notin L \text{ for some } i \geq 0.$$

Then L is not a regular language.

Player 1 (opponent)	Player 2 (you)
any $m \geq 1$	$w \in L$ such that $\lvert w \rvert \geq m$
any $xyz = w$ with $y \neq \epsilon$ and $\lvert xy \rvert \leq m$	$xy^i z \notin L$ for some $i \geq 0$

Pumping lemma for PDA

Bar-Hillel lemma

If $L \subseteq \Sigma^*$ is a recognized by a PDA language then there exists $m \geq 1$ such that any $w \in L$ with $|w| \geq m$ can be represented as $w = x_1 x_2 x_3 x_4 x_5$ such that

- $|x_2 x_4| > 0$,
- $|x_2 x_3 x_4| \leq m$,
- $x_1 x_2^i x_3 x_4^i x_5 \in L$ for any $i \geq 0$.

Pumping lemma for PDA

Bar-Hillel lemma (corollary)

If for any $m \geq 1$ such that there is $w \in L$ such that $|w| \geq m$ and for any representation $w = x_1 x_2 x_3 x_4 x_5$ such that $|x_2 x_4| > 0$, $|x_2 x_3 x_4| \leq m$,

$$x_1 x_2^i x_3 x_4^i x_5 \notin L \text{ for some } i \geq 0,$$

then $L \subseteq \Sigma^*$ is not recognized by any PDA.

Pumping lemma for PDA

Bar-Hillel lemma (corollary)

If for any $m \geq 1$ such that there is $w \in L$ such that $|w| \geq m$ and for any representation $w = x_1x_2x_3x_4x_5$ such that $|x_2x_4| > 0$, $|x_2x_3x_4| \leq m$,

$$x_1x_2^ix_3x_4^ix_5 \notin L \text{ for some } i \geq 0,$$

then $L \subseteq \Sigma^*$ is not recognized by any PDA.

Example

$L = \{a^nb^nc^n \mid n \in \mathbb{N}\}$ is not recognized by a PDA.

Agenda

- Deterministic Pushdown Automata
  - Definition
  - Examples
  - Pumping lemma for PDA
- **Nondeterministic Pushdown Automata**

PDA (Formal Definition)

Definition

A Deterministic Pushdown Automaton (DPDA) is a tuple $\langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, A \rangle$, where

- Q is a finite set of states;
- Σ and Γ are the input and stack (finite) alphabets;
- $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow Q \times \Gamma^*$ is the (partial) transition function;
- $q_0 \in Q$ is the initial state;
- $Z_0 \in \Gamma$ is the initial stack symbol;
- $A \subseteq Q$ is the set of accepting states.

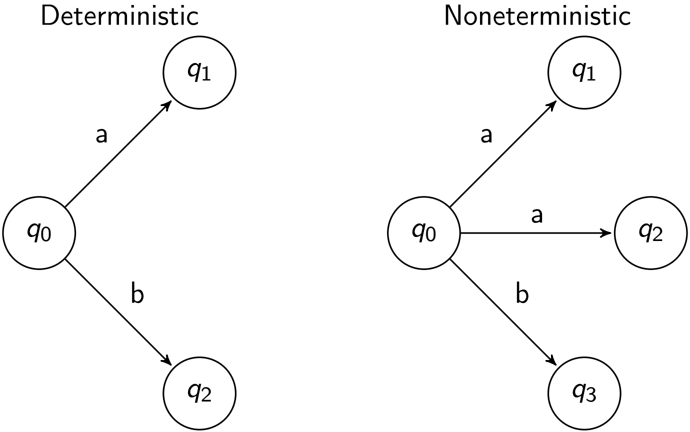
Nondeterministic PDA (Formal Definition)

Definition

A Nondeterministic Pushdown Automaton (NPDA) is a tuple $\langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, A \rangle$, where

- Q is a finite set of states;
- Σ and Γ are the input and stack (finite) alphabets;
- $\delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \times Q \times \Gamma^*$ is the transition relation;
- $q_0 \in Q$ is the initial state;
- $Z_0 \in \Gamma$ is the initial stack symbol;
- $A \subseteq Q$ is the set of accepting states.

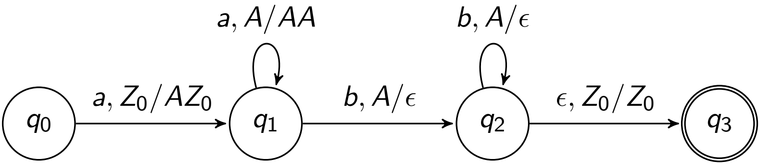
Nondeterministic PDA



PDA. Examples

Example 1

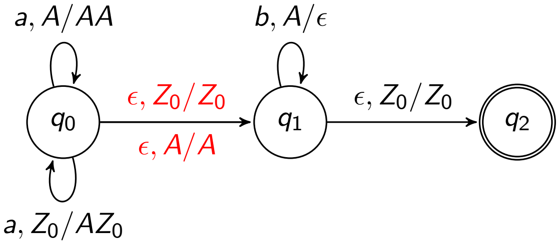
$L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ is recognized by a PDA.



NPDA. Examples

Example 1

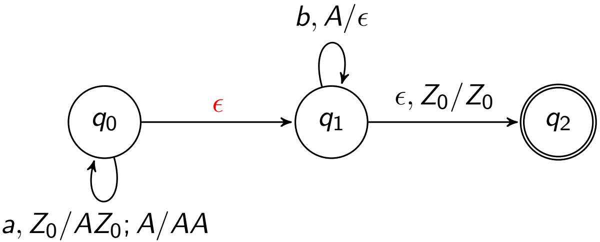
$L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ is recognized by a NPDA.



NPDA. Examples

Example 1

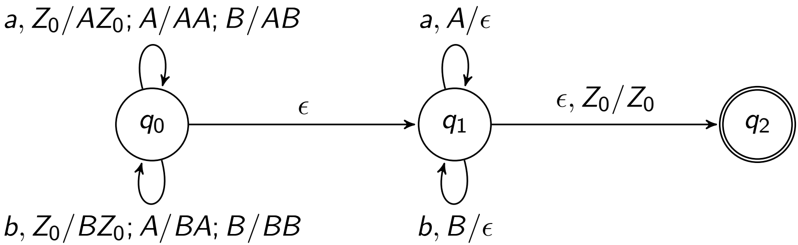
$L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ is recognized by a NPDA.



NPDA. Examples

Example 2

$L_2 = \{vv^R \mid v \in \{a,b\}^*\}$ is recognized by a NPDA.



NPDA. Examples

Example 2

$\{vv^R \mid v \in \{a, b\}^*\}$ is recognized by a NPDA.

Question

Is $\{vv^R \mid v \in \{a, b\}^*\}$ recognized by a DPDA?

Thank you for your attention!