Theoretical Computer Science

Tutorial Week 6

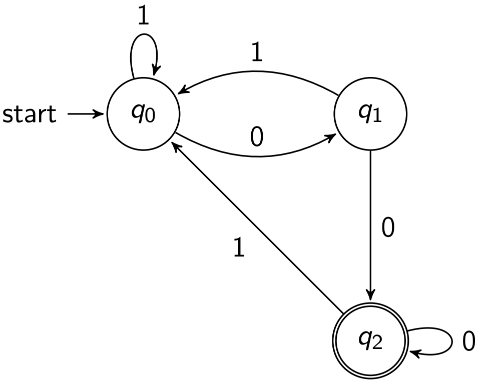
Prof. Andrey Frolov

Agenda

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

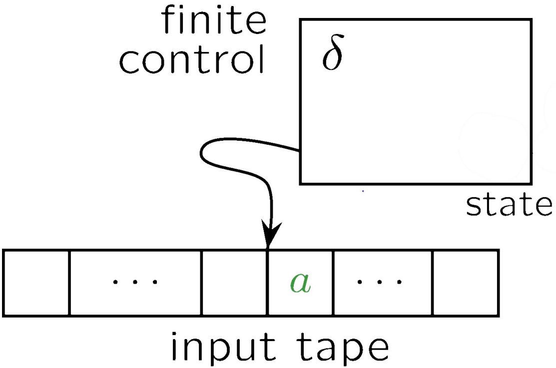
Informally

Finite State Automata

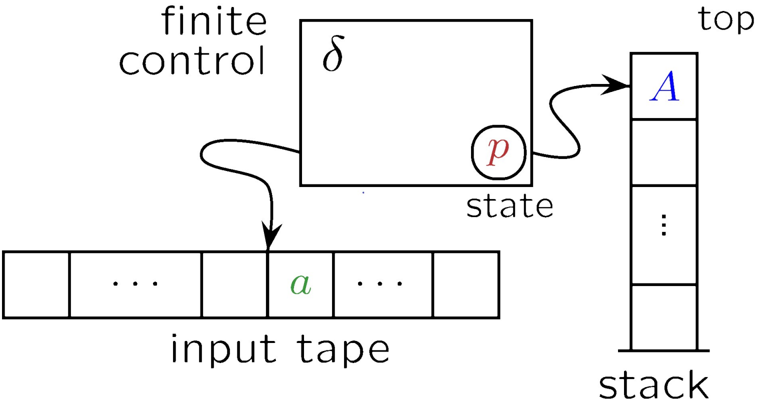


States are memory! Non-rewritable memory!

Finite State Automata



Pushdown Automata



Agenda

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

FSA (Formal definition)

Definition

A (complete) Finite State Automaton 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 (total) *transition* function.

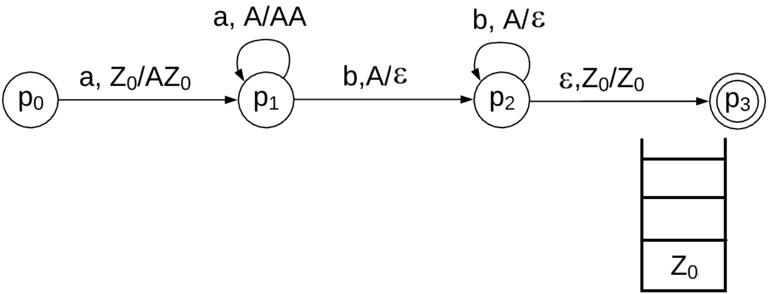
PDA (Formal Definition)

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;
- $\delta : Q \times (I \cup \{\epsilon\}) \times \Gamma \rightarrow Q \times \Gamma^*$ 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.

Pushdown automata



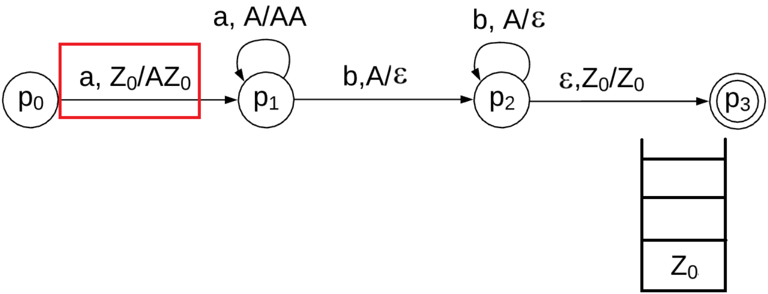
$\Gamma = \{Z_0, A\}$, Z_0 is the initial stack symbol.

Pushdown automata

Conditions on Z_0

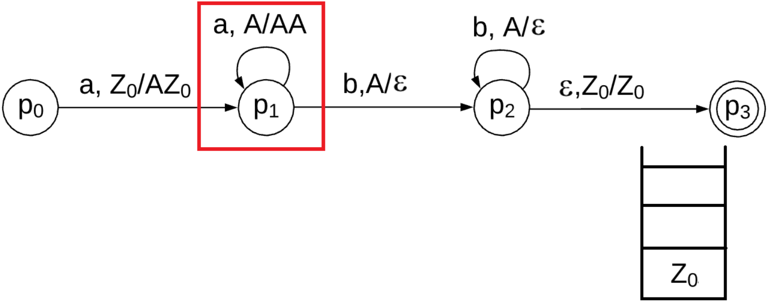
- the stack contains at least one symbol: Z_0;
- Z_0 is never removed;
- no additional copies of Z_0 are pushed onto the stack.

Pushdown automata

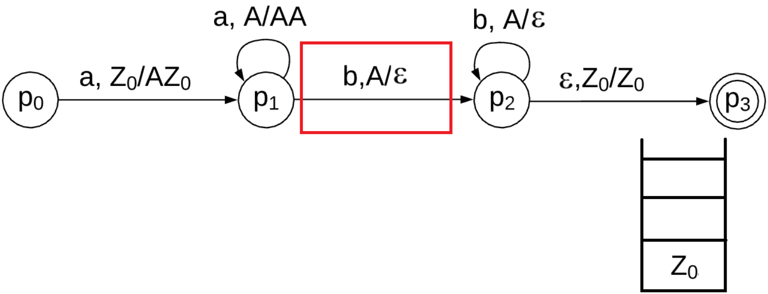


$$\delta(p_0, a, Z_0) = (p_1, AZ_0)$$

Pushdown automata

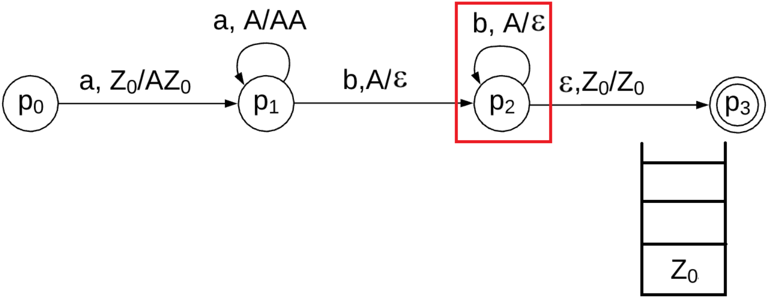


$$\delta(p_1, a, A) = (p_1, AA)$$



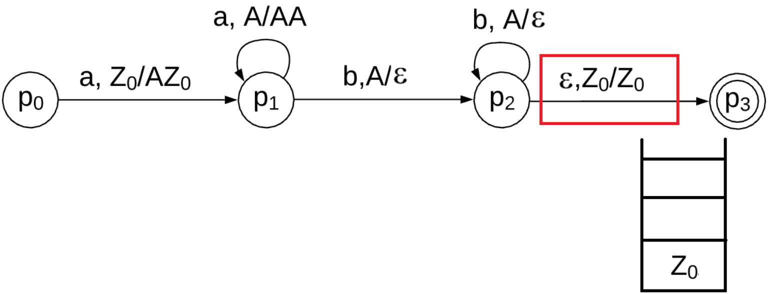
$$\delta(p_1, b, A) = (p_2, \epsilon)$$

Pushdown automata



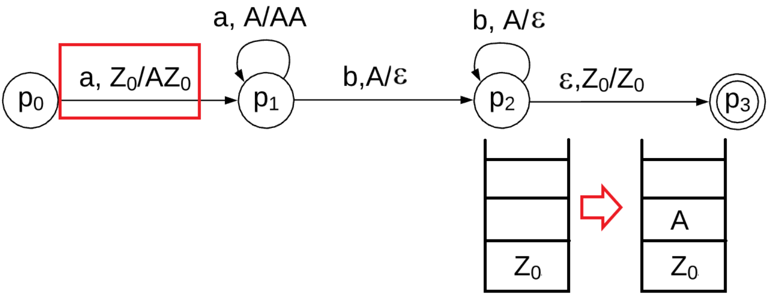
$$\delta(p_2, b, A) = (p_2, \epsilon)$$

Pushdown automata



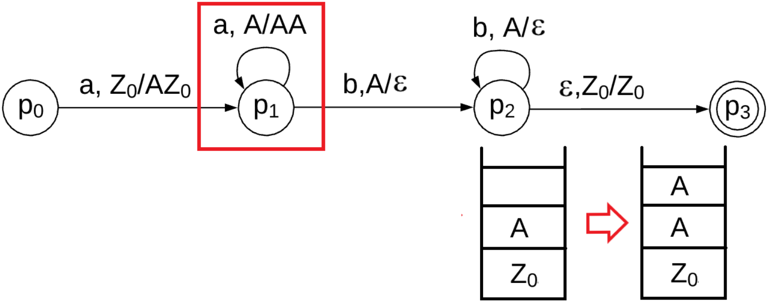
$$\delta(p_2, \epsilon, Z_0) = (p_3, Z_0)$$

Pushdown automata



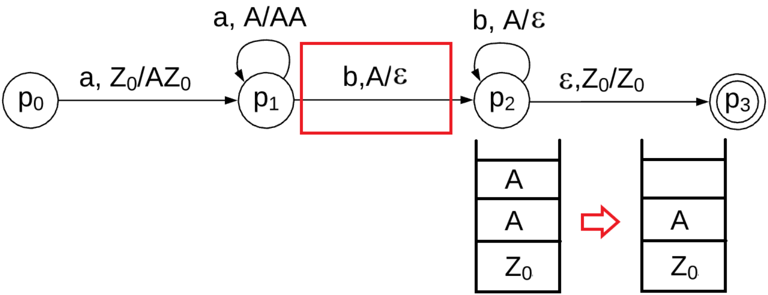
$aabb$

Pushdown automata



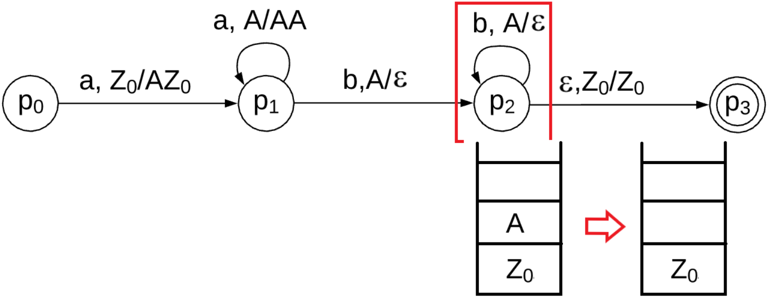
$a\,a\,b\,b$

Pushdown automata



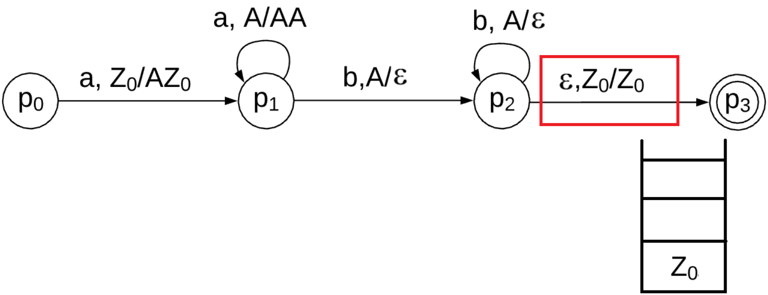
aabb

Pushdown automata



aabb

Pushdown automata



$aabb \in L_1$

PDA. Transition

Definition

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

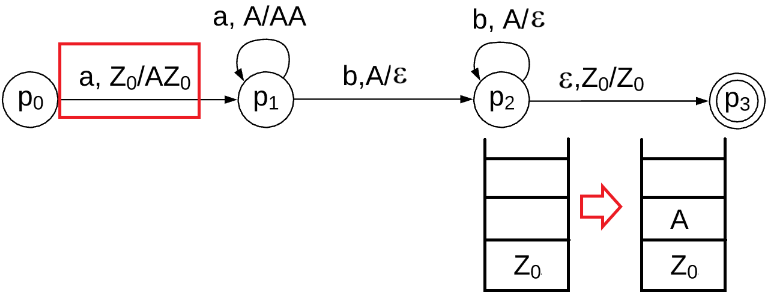
Definition

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

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

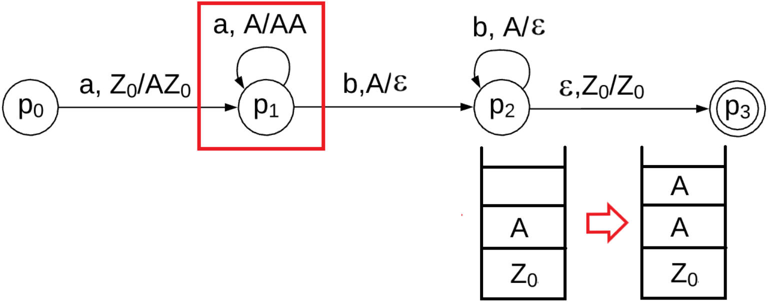
where $a \in \Sigma, x \in \Sigma^*, Z \in \Gamma, Z' \in \Gamma^*$.

Pushdown automata



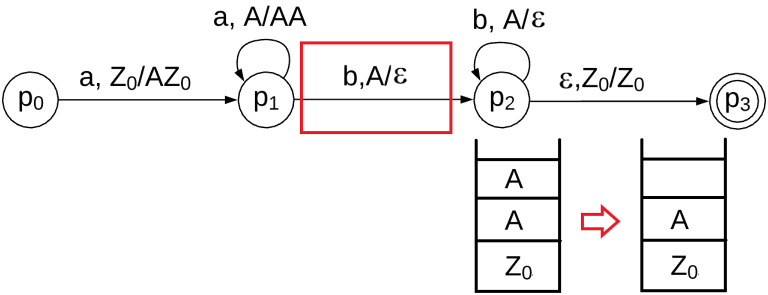
$(p_0, \textcolor{red}{a}abb, Z_0) \vdash (p_1, abb, AZ_0)$

Pushdown automata



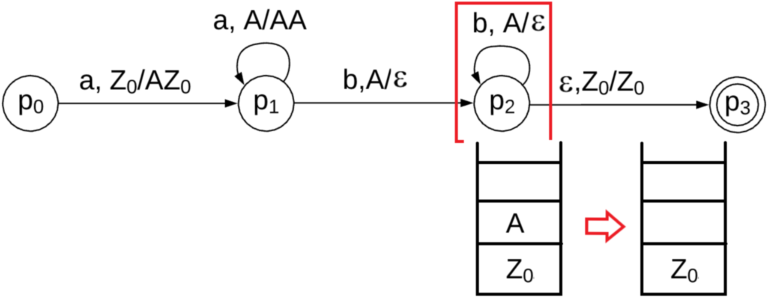
$$(p_0, aabb, Z_0) \vdash (p_1, abb, AZ_0) \vdash (p_1, bb, AAZ_0)$$

Pushdown automata



$$(p_0, aabb, Z_0) \vdash (p_1, abb, AZ_0) \vdash (p_1, bb, AAZ_0) \vdash (p_2, b, AZ_0)$$

Pushdown automata



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

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

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

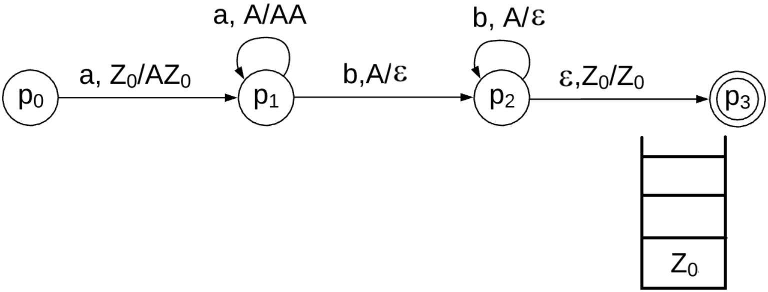
Agenda

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

PDA. Examples

Example 1

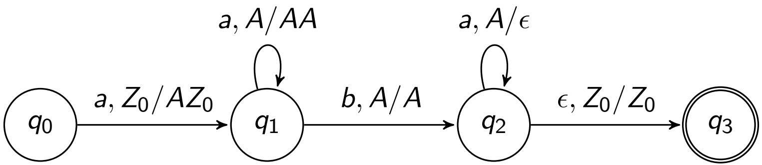
$L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ is not regular, but is recognized by a PDA.



PDA. Examples

Example 2

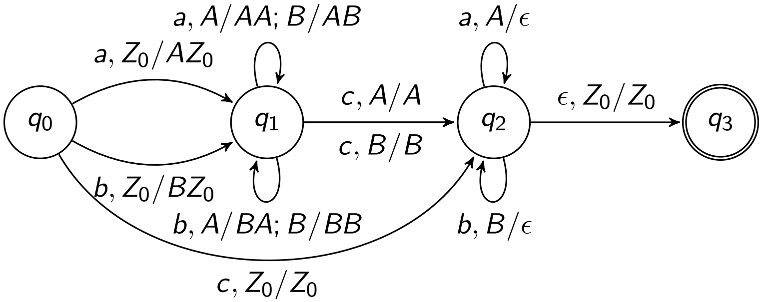
$L_2 = \{a^n b a^n \mid n \in \mathbb{N}\}$ is not regular, but is recognized by a PDA.



PDA. Examples

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 $i \geq 1$.

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 $w = x_1x_2x_3x_4x_5$ such that

- $|x_2x_4| > 0$,
- $|x_2x_3x_4| \leq m$,
- $x_1x_2^ix_3x_4^ix_5 \in L$ for any $i \geq 1$.

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 $i \geq 1$.

Example

$L = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ 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;
- $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to Q \times \Gamma^*$ 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.

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;

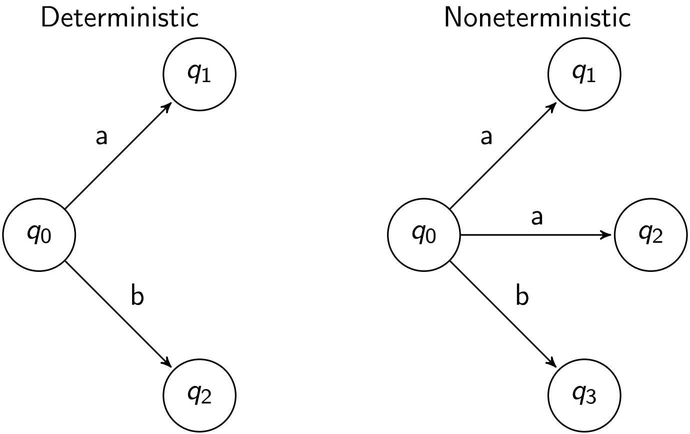
$\delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \times Q \times \Gamma^*$ is the transition relation;

$q_0 \in Q$ is the initial state;

$Z_0 \in \Gamma$ is the initial stack symbol;

$A \subseteq Q$ is the set of accepting states.

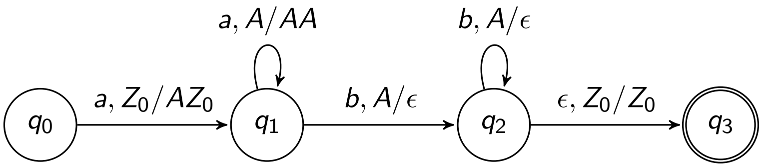
Nondeterministic PDA



PDA. Examples

Example 1

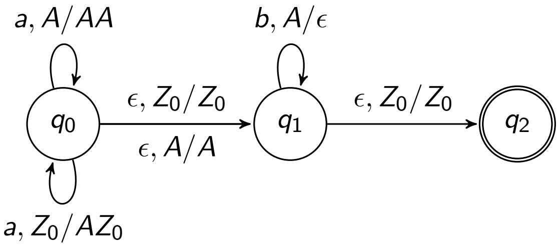
$L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ is recognized by a PDA.



NPDA. Examples

Example 1

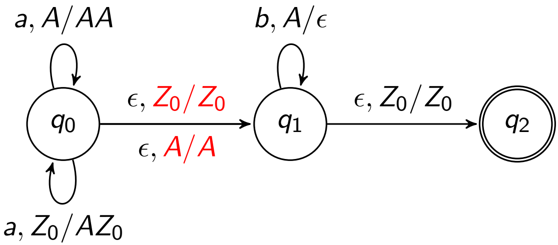
$L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ is recognized by a NPDA.



NPDA. Examples

Example 1

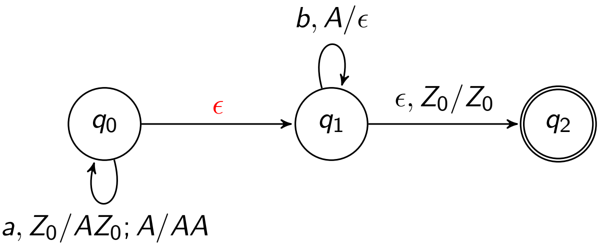
$L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ is recognized by a NPDA.



NPDA. Examples

Example 1

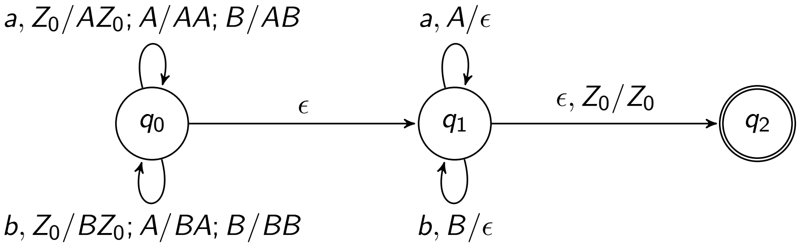
$L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ is recognized by a NPDA.



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!