Lesson 4. Push-down automata

CSIE 3110 – Formal Languages and Automata Theory

Tony Tan
Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Table of contents

Table of contents

Push-down automata (PDA)

In short, push-down automata (PDA) are NFA equipped with a stack.

Push-down automata (PDA)

In short, push-down automata (PDA) are NFA equipped with a stack.

Recall the data structure stack. You can store "stuff" in it, but you can only see and retrieve the top one.

Push-down automata (PDA)

In short, push-down automata (PDA) are NFA equipped with a stack.

Recall the data structure stack. You can store “stuff” in it, but you can only see and retrieve the top one.

In PDA the stack can only store symbols.

Push-down automata (PDA)

In short, push-down automata (PDA) are NFA equipped with a stack.

Recall the data structure stack. You can store “stuff” in it, but you can only see and retrieve the top one.

In PDA the stack can only store symbols.

A PDA works as follows.

- It moves from one state to another state (like an NFA).
- The move depends on the input symbol it currently reads and the symbol on top of the stack.
- During the move, it can pop the top symbol in the stack and push some other symbol into it.

Push-down automata (PDA) – The formal definition

(Def.) A *push-down automaton* (PDA) is a system $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$, where each of the component is as follows.

- Σ is a finite alphabet, called the *input* alphabet, whose elements are called *input symbols*.
- Γ is a finite alphabet, called the *stack* alphabet, whose elements are called *stack symbols*.
- Q is a finite set of states, $q_0 \in Q$ is the initial state. and $F \subseteq Q$ is the set of accepting states.
- δ is the set of transitions of the form: $(p, x, \mathrm{pop}(y)) \to (q, \mathrm{push}(z))$ where $x \in \Sigma \cup \{\varepsilon\}$ and $y, z \in \Gamma \cup \{\varepsilon\}$.

Push-down automata (PDA) – The formal definition

(Def.) A *push-down automaton* (PDA) is a system $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$, where each of the component is as follows.

- Σ is a finite alphabet, called the *input* alphabet, whose elements are called *input symbols*.
- Γ is a finite alphabet, called the *stack* alphabet, whose elements are called *stack symbols*.
- Q is a finite set of states, $q_0 \in Q$ is the initial state. and $F \subseteq Q$ is the set of accepting states.
- δ is the set of transitions of the form: $(p, x, \mathrm{pop}(y)) \to (q, \mathrm{push}(z))$ where $x \in \Sigma \cup \{\varepsilon\}$ and $y, z \in \Gamma \cup \{\varepsilon\}$.
  Intuitively, it means when a PDA is in state p reading x from the input and the top of the stack is y, it can "pop" y from the top of the stack, "push" z into the stack and moves to state q.

Push-down automata (PDA) – The formal definition

(Def.) A *push-down automaton* (PDA) is a system $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$, where each of the component is as follows.

- Σ is a finite alphabet, called the *input* alphabet, whose elements are called *input symbols*.
- Γ is a finite alphabet, called the *stack* alphabet, whose elements are called *stack symbols*.
- Q is a finite set of states, $q_0 \in Q$ is the initial state. and $F \subseteq Q$ is the set of accepting states.
- δ is the set of transitions of the form: $(p, x, \mathrm{pop}(y)) \to (q, \mathrm{push}(z))$ where $x \in \Sigma \cup \{\varepsilon\}$ and $y, z \in \Gamma \cup \{\varepsilon\}$.
  Intuitively, it means when a PDA is in state p reading x from the input and the top of the stack is y, it can "pop" y from the top of the stack, "push" z into the stack and moves to state q.

(Note) The stack can only contain stack symbols.

Illustration of the transition $(p, x, \mathbf{pop}(y)) \rightarrow (q, \mathbf{push}(z))$ in PDA

$$p \xrightarrow[\text{push}(z)]{x,\ \text{pop}(y)} q$$

Illustration of the transition $(p, x, \mathbf{pop}(y)) \;\rightarrow\; (q, \mathbf{push}(z))$ in PDA

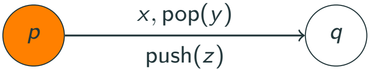


Illustration of the transition $(p, x, \mathbf{pop}(y)) \rightarrow (q, \mathbf{push}(z))$ in PDA

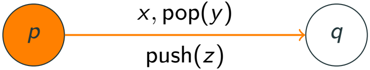
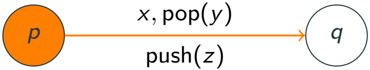


Illustration of the transition $(p, x, \mathbf{pop}(y)) \rightarrow (q, \mathbf{push}(z))$ in PDA



The input: $\ldots\ldots\ldots x \ldots\ldots\ldots$

Illustration of the transition $(p, x, \mathbf{pop}(y)) \rightarrow (q, \mathbf{push}(z))$ in PDA

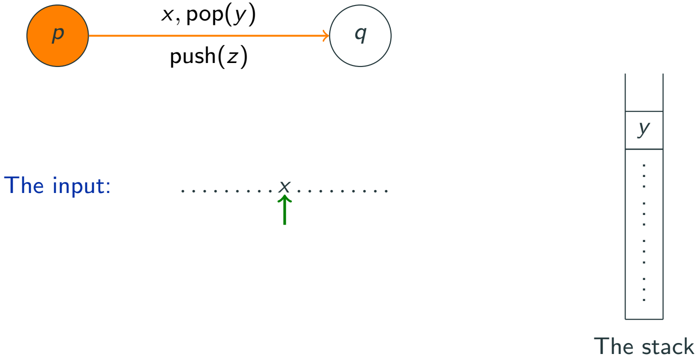
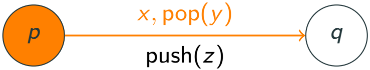
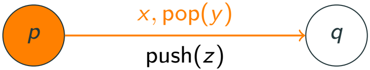


Illustration of the transition $(p, x, \mathbf{pop}(y)) \rightarrow (q, \mathbf{push}(z))$ in PDA



The input: $\ldots\ldots\ldots x \ldots\ldots\ldots$

Illustration of the transition $(p, x, \textbf{pop}(y)) \rightarrow (q, \textbf{push}(z))$ in PDA



The input: $\ldots\ldots\ldots x \ldots\ldots\ldots$

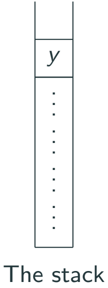
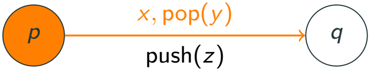


Illustration of the transition $(p, x, \mathbf{pop}(y)) \rightarrow (q, \mathbf{push}(z))$ in PDA



The input: . . . . x . . . .

The stack

Illustration of the transition $(p, x, \textbf{pop}(y)) \rightarrow (q, \textbf{push}(z))$ in PDA

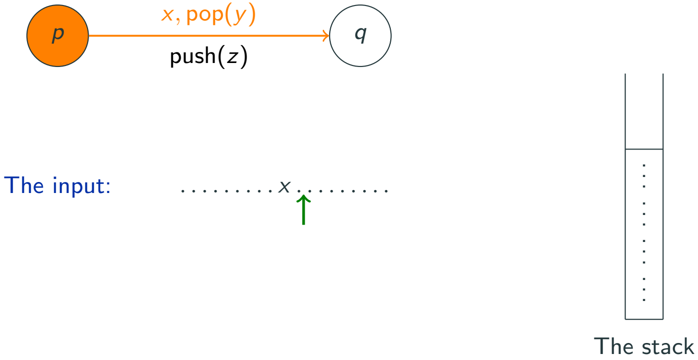
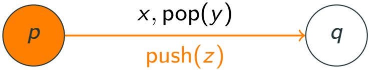


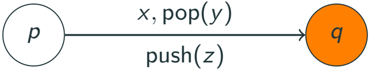
Illustration of the transition $(p, x, \mathbf{pop}(y)) \;\rightarrow\; (q, \mathbf{push}(z))$ in PDA



The input: $\ldots\ldots\ldots x \ldots\ldots\ldots$

The stack

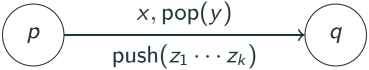
Illustration of the transition $(p, x, \mathbf{pop}(y)) \rightarrow (q, \mathbf{push}(z))$ in PDA



The input: $\ldots\ldots\ldots x \ldots\ldots\ldots$

The stack

Illustration of PDA transitions – continued



The PDA “pushes” a string onto its stack:

- Read the symbol x.
- Pop y (if possible) from the stack.
- Push z_1 into the stack, then z_2, then z_3 and so on until z_k.
- Move into state q.

Illustration of PDA transitions – continued

$$p \xrightarrow[\text{push}(z)]{x,\ \text{pop}(\varepsilon)} q$$

When $y = \varepsilon$, the PDA does not check the content the stack.

- Read the symbol x.
- Push z into the stack.
- Move into state q.

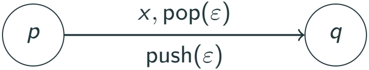
Illustration of PDA transitions – continued

$$p \xrightarrow[\text{push}(\varepsilon)]{x,\ \text{pop}(y)} q$$

When $z = \varepsilon$, the PDA does not push anything to the stack.

- Read the symbol x.
- Pop y (if possible) from the stack.
- Move into state q.

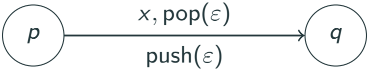
Illustration of PDA transitions – continued



When $y = z = \varepsilon$, the PDA does not do anything to the stack.

- Read the symbol x.
- Move into state q.

Illustration of PDA transitions – continued

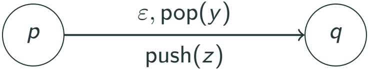


When $y = z = \varepsilon$, the PDA does not do anything to the stack.

- Read the symbol x.
- Move into state q.

(Note) We can view NFA as a special case of PDA where the transitions are all of the form: $(p, x, \text{pop}(\varepsilon)) \to (q, \text{push}(\varepsilon))$.

Illustration of PDA transitions – continued



When $x = \varepsilon$, the PDA does not read any input symbol.

- Pop y (if possible) from the stack.
- Push z into the stack.
- Move into state q.

The criteria for the acceptance of a word by a PDA

Suppose $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ is a PDA.

The criteria for the acceptance of a word by a PDA

Suppose $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ is a PDA.

On input word w, it works as follows.

The criteria for the acceptance of a word by a PDA

Suppose $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ is a PDA.

On input word w, it works as follows.

- It starts from the initial state q_0 and the stack is initially empty.

The criteria for the acceptance of a word by a PDA

Suppose $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ is a PDA.

On input word w, it works as follows.

- It starts from the initial state q_0 and the stack is initially empty.
- Then, it moves (non-deterministically) from one state to another using its transitions.

The criteria for the acceptance of a word by a PDA

Suppose $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ is a PDA.

On input word w, it works as follows.

- It starts from the initial state q_0 and the stack is initially empty.
- Then, it moves (non-deterministically) from one state to another using its transitions.
- When it finishes reading w and it is in one of the accepting states, then we say that w is accepted by $\mathcal{A}$.

The criteria for the acceptance of a word by a PDA

Suppose $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ is a PDA.

On input word w, it works as follows.

- It starts from the initial state q_0 and the stack is initially empty.
- Then, it moves (non-deterministically) from one state to another using its transitions.
- When it finishes reading w and it is in one of the accepting states, then we say that w is accepted by $\mathcal{A}$.

See the formal definition of accepting run for PDA in the note.

The criteria for the acceptance of a word by a PDA

Suppose $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ is a PDA.

On input word w, it works as follows.

- It starts from the initial state q_0 and the stack is initially empty.
- Then, it moves (non-deterministically) from one state to another using its transitions.
- When it finishes reading w and it is in one of the accepting states, then we say that w is accepted by $\mathcal{A}$.

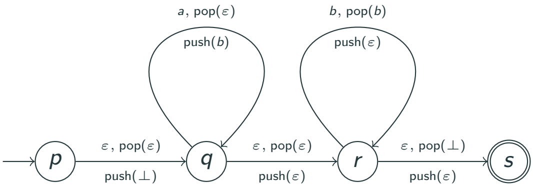
See the formal definition of accepting run for PDA in the note.

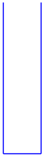
The language accepted by a PDA $\mathcal{A}$ is denoted by $L(\mathcal{A})$:

$$L(\mathcal{A}) \quad := \quad \{w \in \Sigma^* \mid \text{there is an accepting run of } \mathcal{A} \text{ on } w\}$$

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

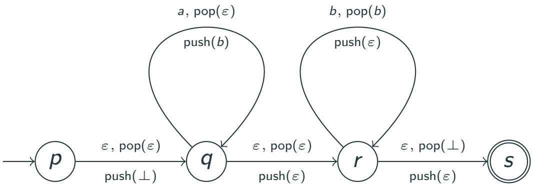




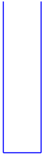
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



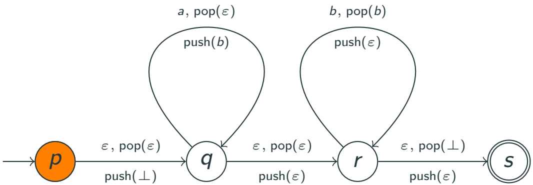
Input word: ε



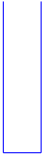
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



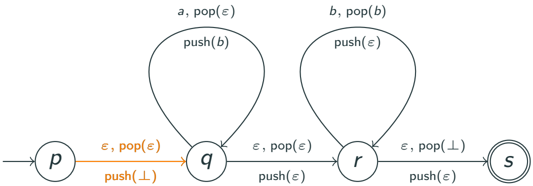
Input word: ε



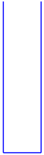
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

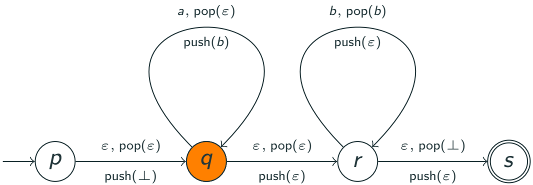


Input word: ε

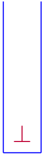


An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

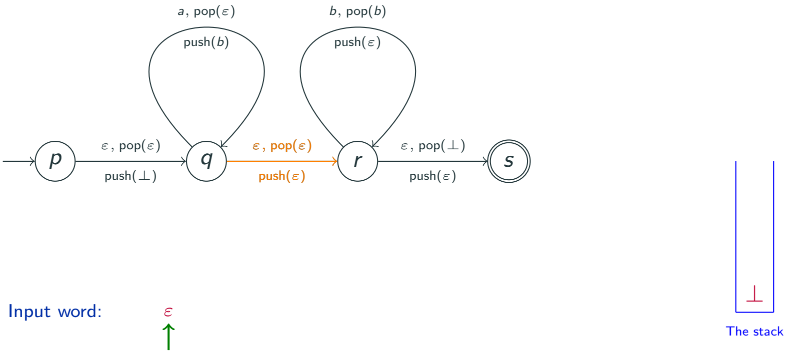


Input word: ε



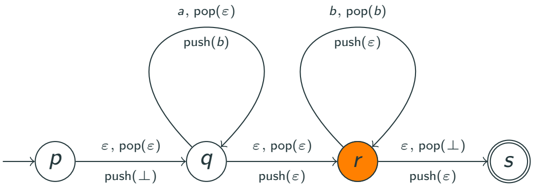
An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

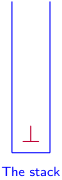


An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

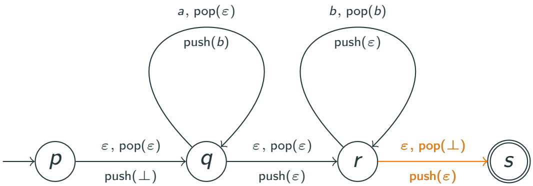


Input word: ε

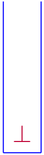


An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

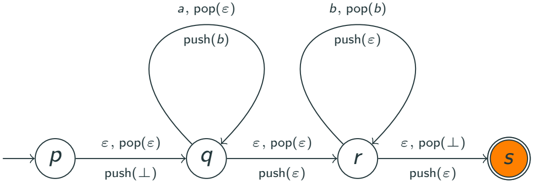


Input word: ε

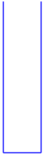


An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

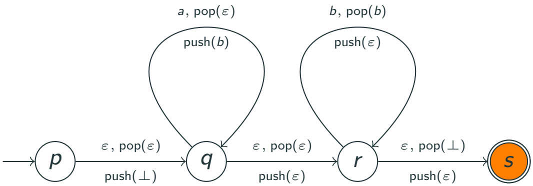


Input word: ε

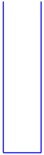


An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



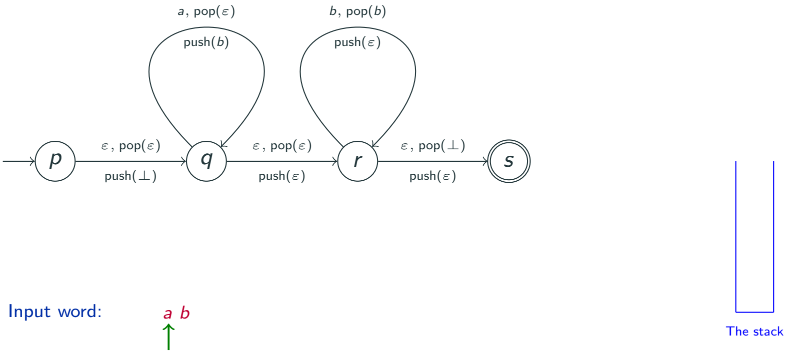
Input word: ε



This is an accepting run of $\mathcal{A}$! So, $\varepsilon \in L(\mathcal{A})$.

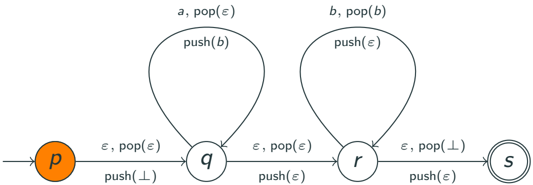
An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

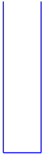
$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



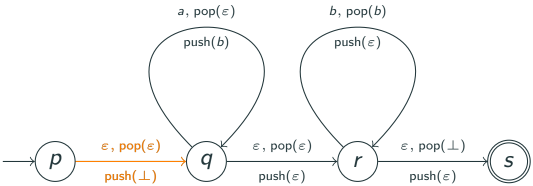


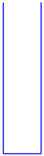
The stack

Input word: $a\ b$

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.





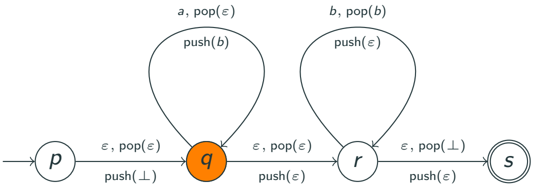
Input word: $a\ b$

$\uparrow$

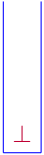
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



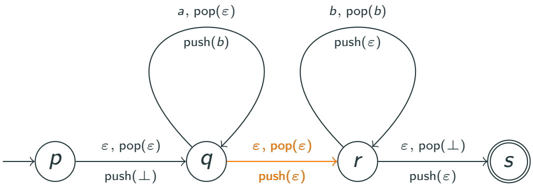
Input word: $a\ b$



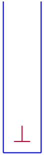
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

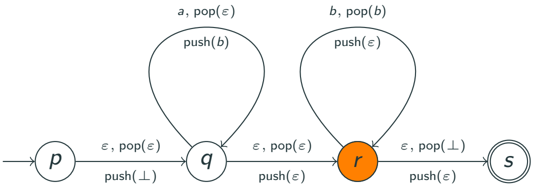


Input word: $a\ b$

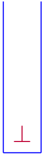


An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



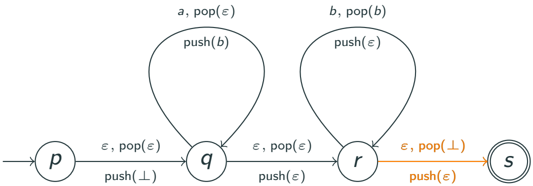
Input word: $a\ b$

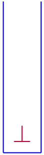


The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \bot\}$.



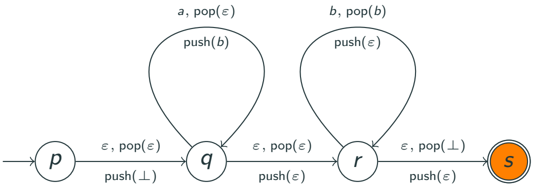


The stack

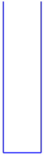
Input word: $a\ b$

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

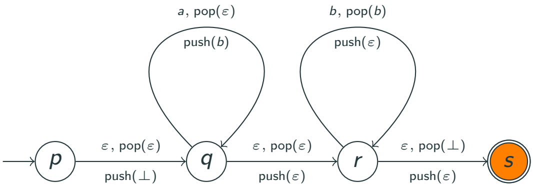


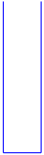
Input word: $a\ b$



An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.





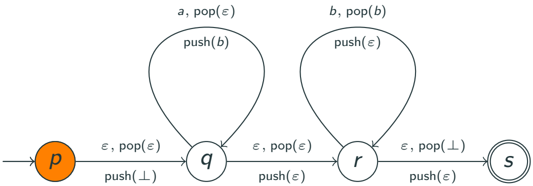
Input word: $a\ b$

The stack

This is not an accepting run of $\mathcal{A}$, because the PDA does not finish reading the input word.

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

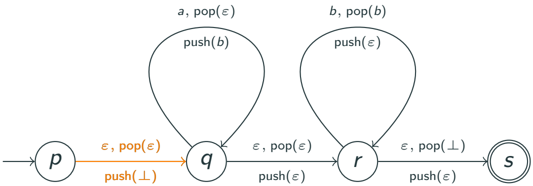


Input word: $a\ b$

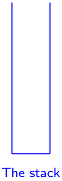
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

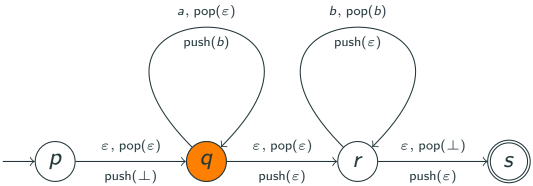


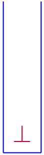
Input word: $a\ b$



An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



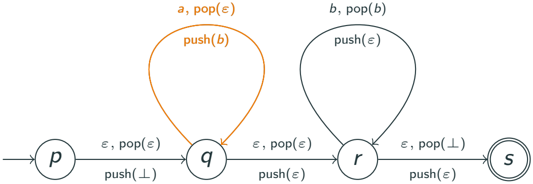


The stack

Input word: $a\ b$

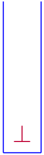
An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



Input word: $a\ b$

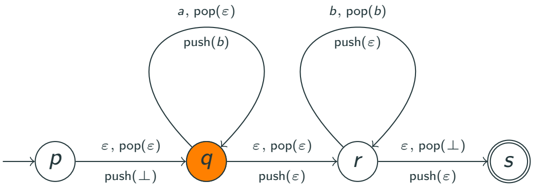
$\uparrow$



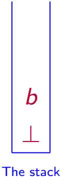
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

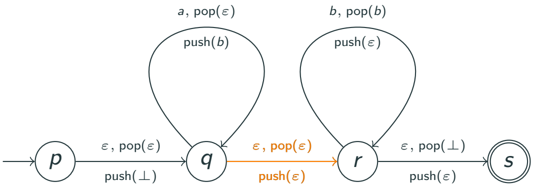


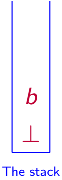
Input word: $a\ b$



An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

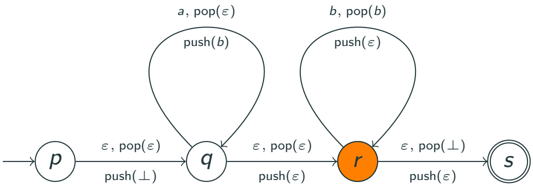


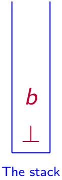


Input word: $a\ b$

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

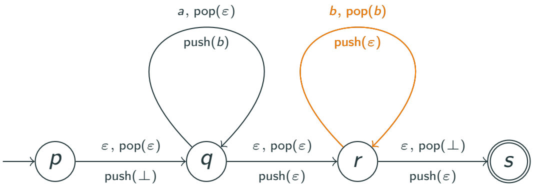


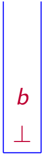


Input word: $a\ b$

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



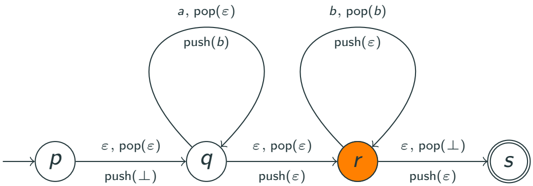


The stack

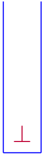
Input word: $a\ b$

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



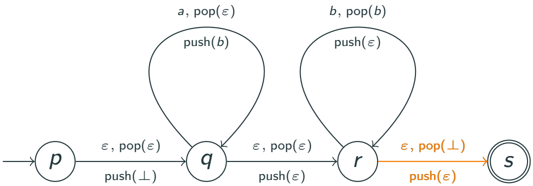
Input word: $a\ b$



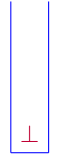
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

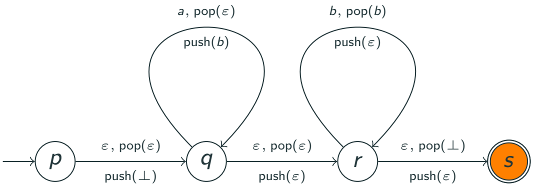


Input word: $a\ b$

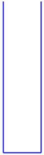


An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



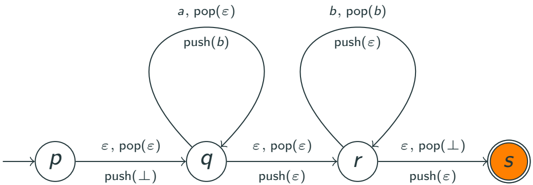
Input word: $a \ b$

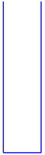


The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



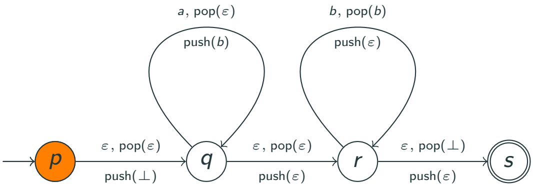


Input word: $a\ b$

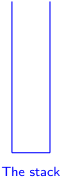
This is an accepting run of $\mathcal{A}$, because the PDA finishes reading the input word and ends in an accepting state.

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

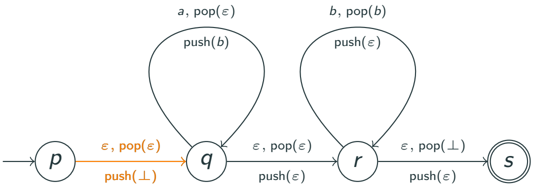


Input word: $a\ a\ a\ b\ b\ b$

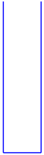


An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



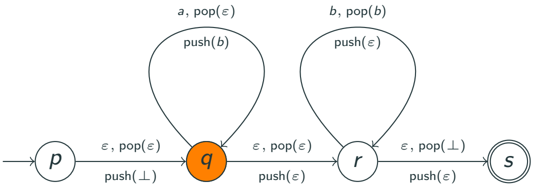
Input word: $a\ a\ a\ b\ b\ b$



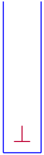
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



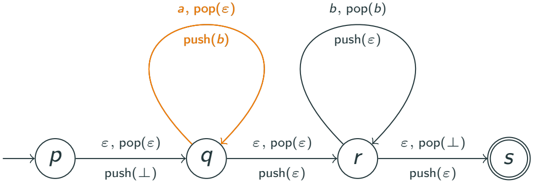
Input word: $a\ a\ a\ b\ b\ b$



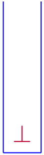
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



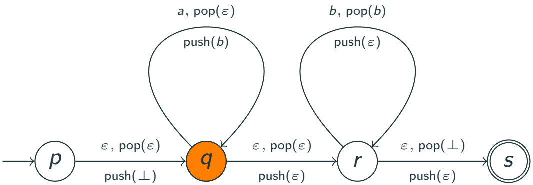
Input word: $a\ a\ a\ b\ b\ b$



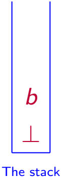
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

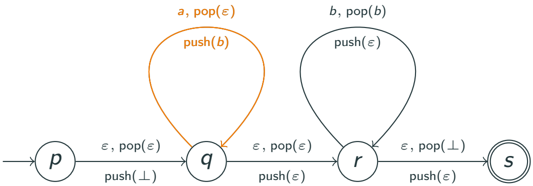


Input word: $a\ a\ a\ b\ b\ b$

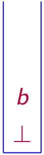


An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



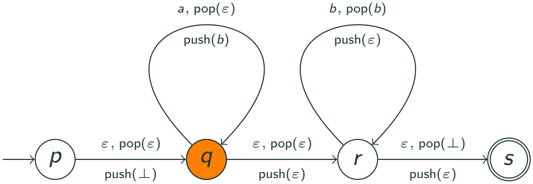
Input word: $a\ a\ a\ b\ b\ b$



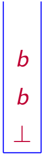
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



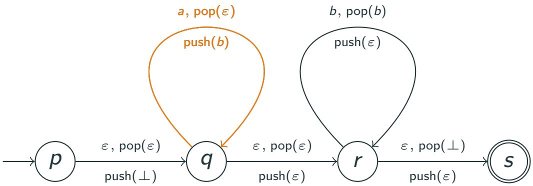
Input word: $a\ a\ a\ b\ b\ b$



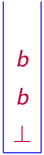
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



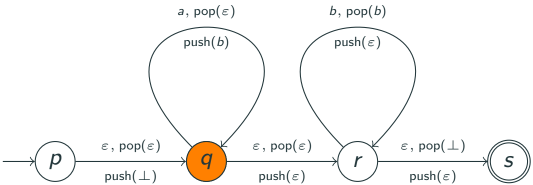
Input word: $a\ a\ a\ b\ b\ b$



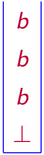
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



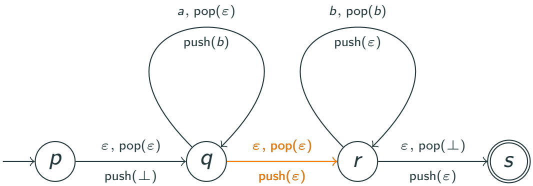
Input word: $a\ a\ a\ b\ b\ b$

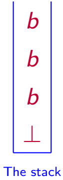


The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

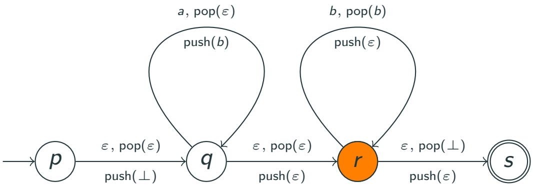




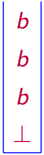
Input word: $a\ a\ a\ b\ b\ b$

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \bot\}$.



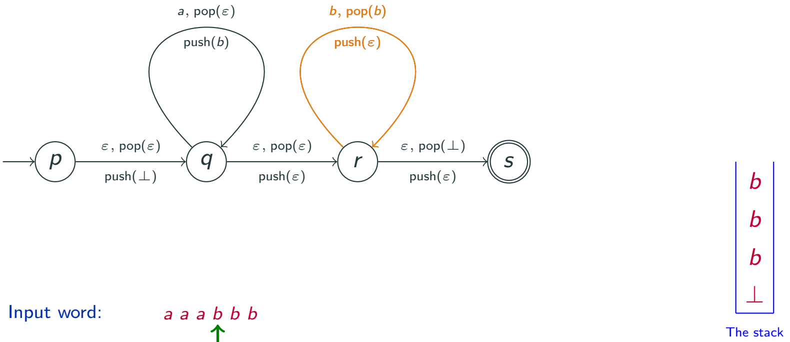
Input word: $a\ a\ a\ b\ b\ b$



The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

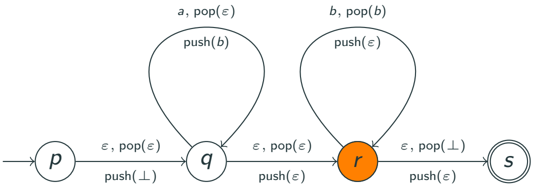
$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



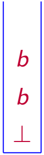
Input word: $a\ a\ a\ b\ b\ b$

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



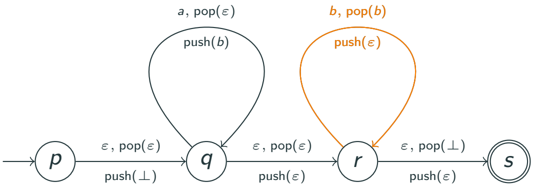
Input word: $a\ a\ a\ b\ b\ b$



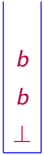
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



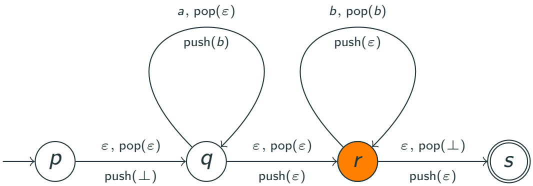
Input word: $a\ a\ a\ b\ b\ b$



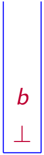
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



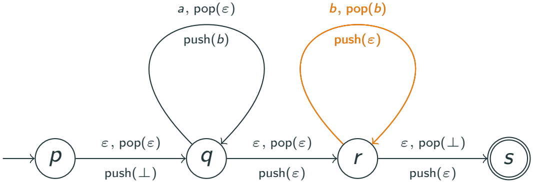
Input word: $a\ a\ a\ b\ b\ b$



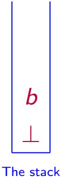
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

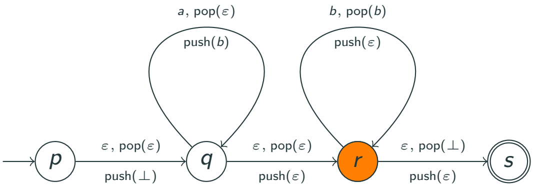


Input word: $a\ a\ a\ b\ b\ b$

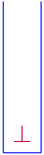


An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



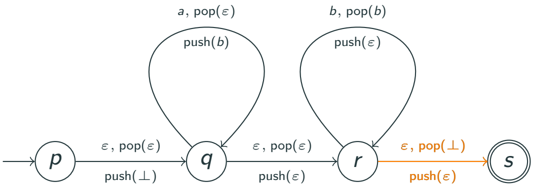
Input word: $a\ a\ a\ b\ b\ b$



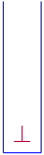
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



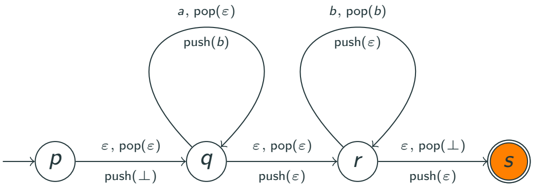
Input word: $a\ a\ a\ b\ b\ b$



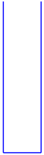
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

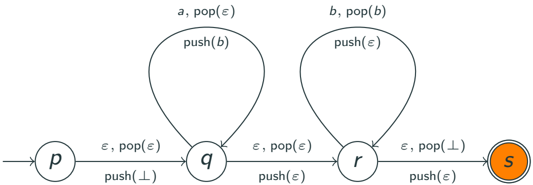


Input word: $a\ a\ a\ b\ b\ b$

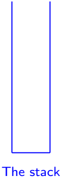


An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



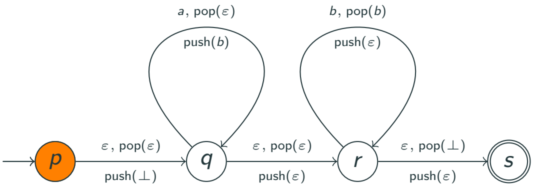
Input word: $a\ a\ a\ b\ b\ b$



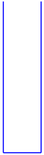
This is an accepting run of $\mathcal{A}$, because the PDA finishes reading the input word and ends in an accepting state.

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

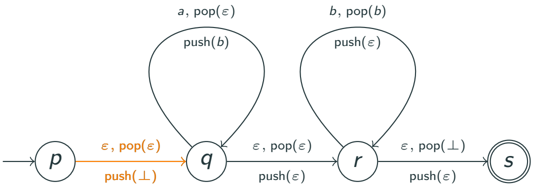


Input word: $a\ a\ a\ b\ b\ b\ a\ b$

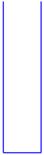


An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



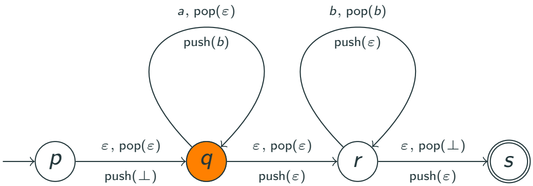
Input word: $a\ a\ a\ b\ b\ b\ a\ b$



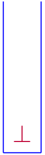
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



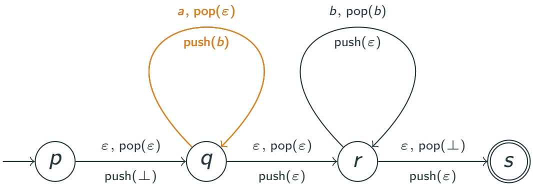
Input word: $a\ a\ a\ b\ b\ b\ a\ b$



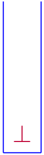
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



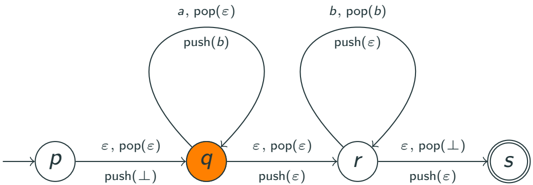
Input word: $a\ a\ a\ b\ b\ b\ a\ b$

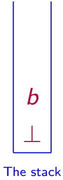


The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

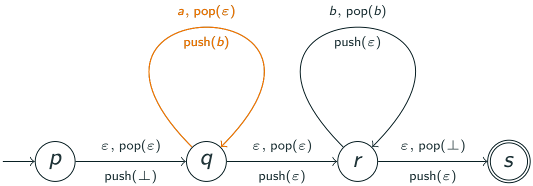




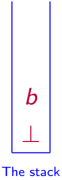
Input word: $a\ a\ a\ b\ b\ b\ a\ b$

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.

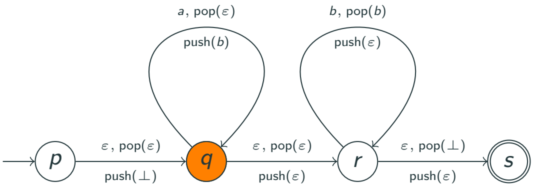


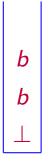
Input word: $a\ a\ a\ b\ b\ b\ a\ b$



An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



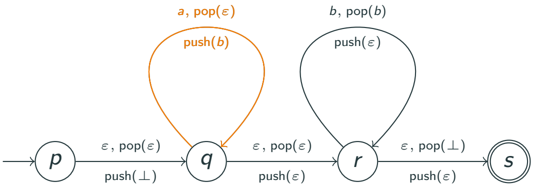


The stack

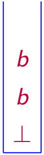
Input word: $a\ a\ a\ b\ b\ b\ a\ b$

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



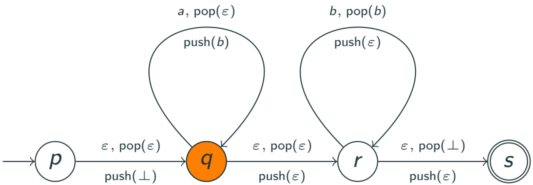
Input word: $a\ a\ a\ b\ b\ b\ a\ b$



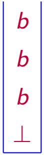
The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



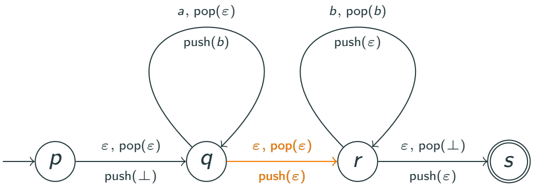
Input word: $a\ a\ a\ b\ b\ b\ a\ b$

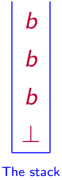


The stack

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



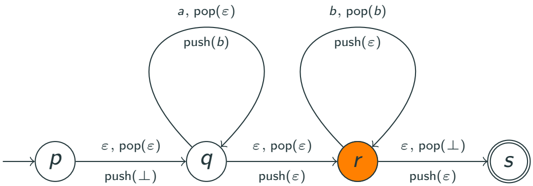


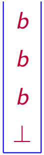
Input word: $a\ a\ a\ b\ b\ b\ a\ b$

This is not an accepting run of $\mathcal{A}$.

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.





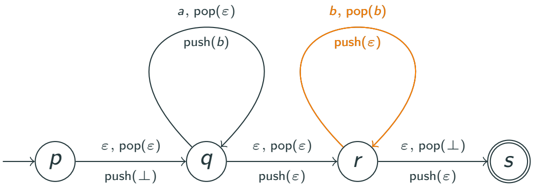
The stack

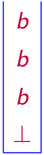
Input word: $a\ a\ a\ b\ b\ b\ a\ b$

This is not an accepting run of $\mathcal{A}$.

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.





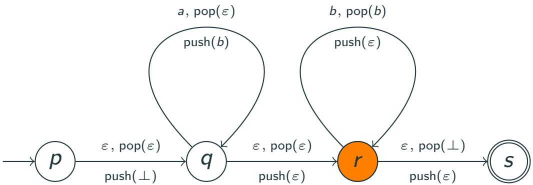
The stack

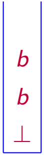
Input word: $a\ a\ a\ b\ b\ b\ a\ b$

This is not an accepting run of $\mathcal{A}$.

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.





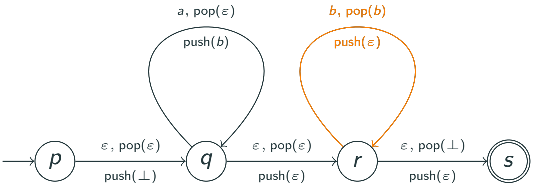
The stack

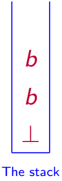
Input word: $a\ a\ a\ b\ b\ b\ a\ b$

This is not an accepting run of $\mathcal{A}$.

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



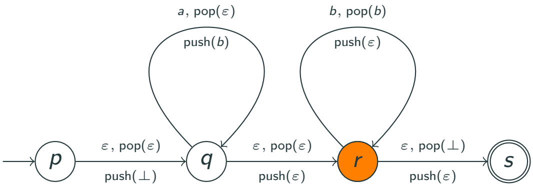


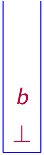
Input word: $a\ a\ a\ b\ b\ b\ a\ b$

This is not an accepting run of $\mathcal{A}$.

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.





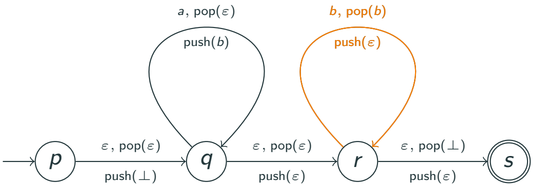
The stack

Input word: $a\ a\ a\ b\ b\ b\ a\ b$

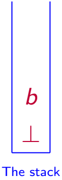
This is not an accepting run of $\mathcal{A}$.

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



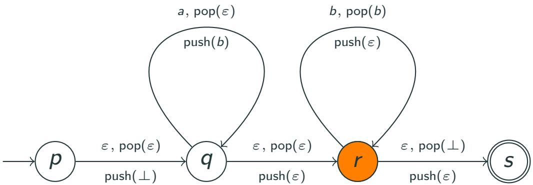
Input word: $a\ a\ a\ b\ b\ b\ a\ b$



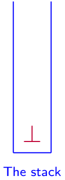
This is not an accepting run of $\mathcal{A}$.

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



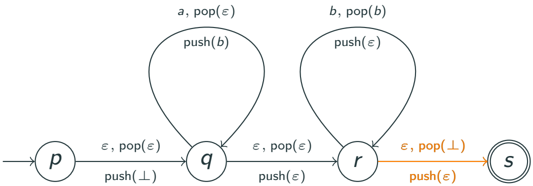
Input word: $a\ a\ a\ b\ b\ b\ a\ b$

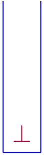


This is not an accepting run of $\mathcal{A}$.

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.





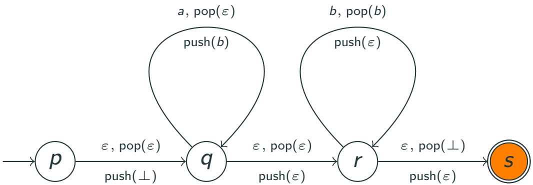
The stack

Input word: $a\ a\ a\ b\ b\ b\ a\ b$

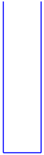
This is not an accepting run of $\mathcal{A}$.

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



Input word: $a\ a\ a\ b\ b\ b\ a\ b$

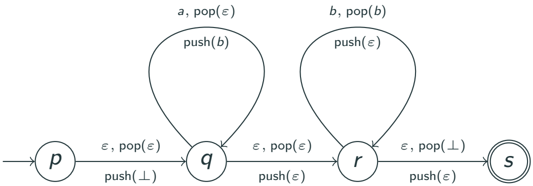


The stack

This is not an accepting run of $\mathcal{A}$.

An example: A PDA $\mathcal{A}$ for $L = \{a^n b^n | n \geqslant 0\}$

$\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \perp\}$.



Note: $w \in L(\mathcal{A})$ if and only if w is of the form $a^n b^n$.

The equivalence between CFG and PDA

Theorem 4.1

- *For every CFG $\mathcal{G}$, there is a PDA $\mathcal{A}$ such that $L(\mathcal{A}) = L(\mathcal{G})$.*
- *For every PDA $\mathcal{A}$, there is a CFG $\mathcal{G}$ such that $L(\mathcal{A}) = L(\mathcal{G})$.*

Table of contents

From CFG to PDA

Theorem (The first item of Theorem 4.1)
For every CFG $\mathcal{G}$, there is a PDA $\mathcal{A}$ such that $L(\mathcal{A}) = L(\mathcal{G})$.

From CFG to PDA

Theorem (The first item of Theorem 4.1)
For every CFG $\mathcal{G}$, there is a PDA $\mathcal{A}$ such that $L(\mathcal{A}) = L(\mathcal{G})$.

The idea is to use a PDA to "simulate" the derivations of the words generated by the CFG.

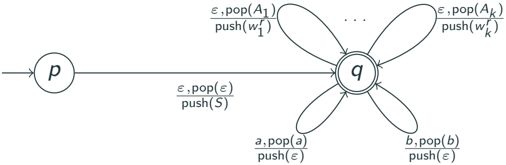
Proof of Theorem 4.1: From CFG to PDA

Let $\mathcal{G} = \langle \Sigma, V, R, S \rangle$ where $\Sigma = \{a, b\}$ and $R = \{A_1 \to w_1, \ldots, A_k \to w_k\}$.

Proof of Theorem 4.1: From CFG to PDA

Let $\mathcal{G} = \langle \Sigma, V, R, S \rangle$ where $\Sigma = \{a, b\}$ and $R = \{A_1 \to w_1, \ldots, A_k \to w_k\}$.

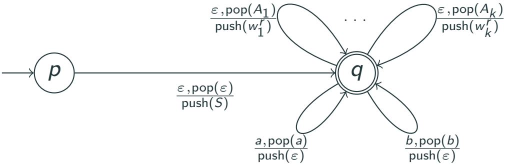
Consider the following PDA with $\Sigma = \{a, b\}$ and $\Gamma = V \cup \Sigma$:



Proof of Theorem 4.1: From CFG to PDA

Let $\mathcal{G} = \langle \Sigma, V, R, S \rangle$ where $\Sigma = \{a, b\}$ and $R = \{A_1 \to w_1, \ldots, A_k \to w_k\}$.

Consider the following PDA with $\Sigma = \{a, b\}$ and $\Gamma = V \cup \Sigma$:

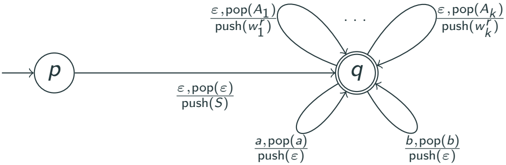


- The stack contains the strings to derive.

Proof of Theorem 4.1: From CFG to PDA

Let $\mathcal{G} = \langle \Sigma, V, R, S \rangle$ where $\Sigma = \{a, b\}$ and $R = \{A_1 \to w_1, \ldots, A_k \to w_k\}$.

Consider the following PDA with $\Sigma = \{a, b\}$ and $\Gamma = V \cup \Sigma$:

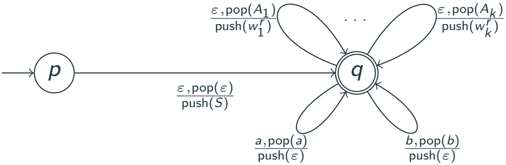


- The stack contains the strings to derive.
- The transition $(q, \varepsilon, \text{pop}(A_i)) \to (q, \text{push}(w_i^r))$ simulates the rule $A_i \to w_i$.

Proof of Theorem 4.1: From CFG to PDA

Let $\mathcal{G} = \langle \Sigma, V, R, S \rangle$ where $\Sigma = \{a, b\}$ and $R = \{A_1 \to w_1, \ldots, A_k \to w_k\}$.

Consider the following PDA with $\Sigma = \{a, b\}$ and $\Gamma = V \cup \Sigma$:

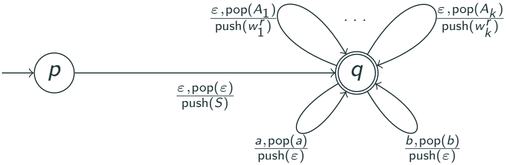


- The stack contains the strings to derive.
- The transition $(q, \varepsilon, \mathrm{pop}(A_i)) \to (q, \mathrm{push}(w_i^r))$ simulates the rule $A_i \to w_i$.
- The transitions $(q, a, \mathrm{pop}(\varepsilon)) \to (q, \mathrm{push}(\varepsilon))$ and $(q, b, \mathrm{pop}(\varepsilon)) \to (q, \mathrm{push}(\varepsilon))$ enforce the input word to be the same as the terminals that are already derived.

Proof of Theorem 4.1: From CFG to PDA

Let $\mathcal{G} = \langle \Sigma, V, R, S \rangle$ where $\Sigma = \{a, b\}$ and $R = \{A_1 \to w_1, \ldots, A_k \to w_k\}$.

Consider the following PDA with $\Sigma = \{a, b\}$ and $\Gamma = V \cup \Sigma$:

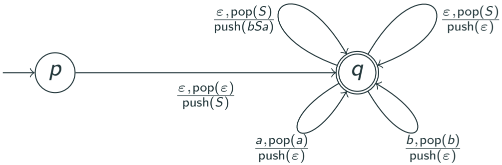


- The stack contains the strings to derive.
- The transition $(q, \varepsilon, \text{pop}(A_i)) \to (q, \text{push}(w_i^r))$ simulates the rule $A_i \to w_i$.
- The transitions $(q, a, \text{pop}(\varepsilon)) \to (q, \text{push}(\varepsilon))$ and $(q, b, \text{pop}(\varepsilon)) \to (q, \text{push}(\varepsilon))$ enforce the input word to be the same as the terminals that are already derived.
- The transition $(p, \varepsilon, \text{pop}(\varepsilon)) \to (q, \text{push}(S))$ enforces the words are derived only from the start variable S.

An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

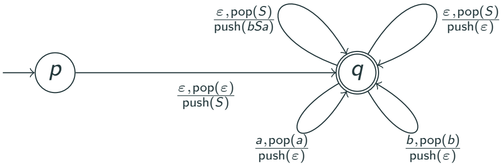
The PDA is:



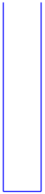
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b$

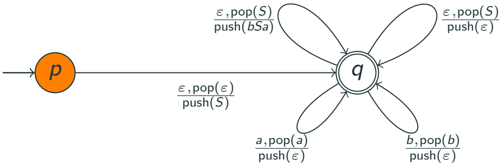


The stack

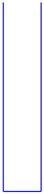
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b$

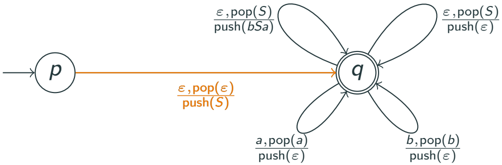


The stack

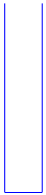
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \rightarrow aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b$

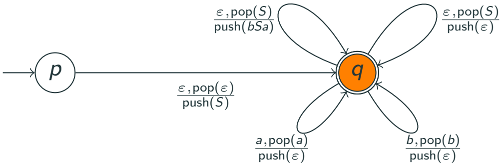


The stack

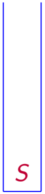
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



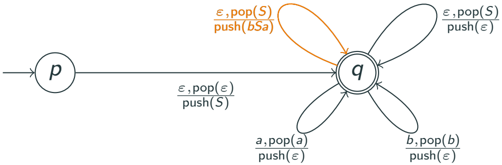
Input word: $a\ a\ a\ b\ b\ b$



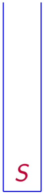
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b$

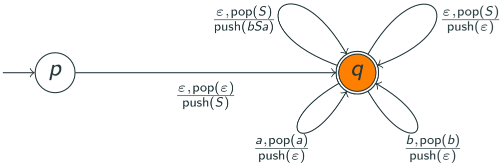


The stack

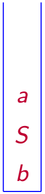
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b$

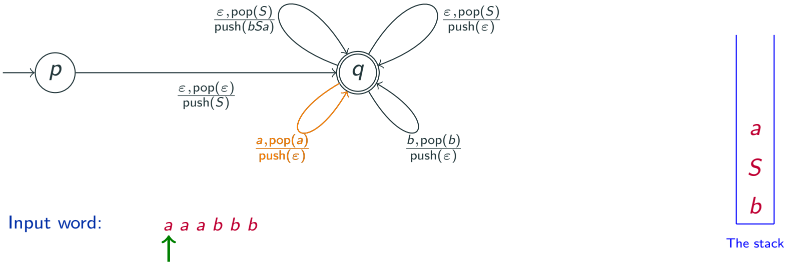


The stack

An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

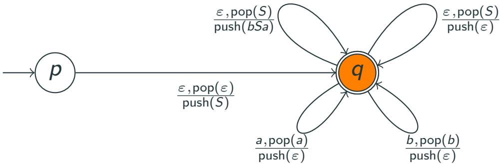
The PDA is:



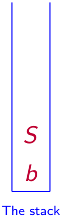
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



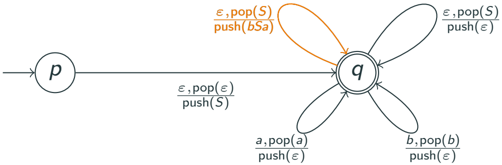
Input word: $a\ a\ a\ b\ b\ b$



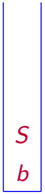
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b$

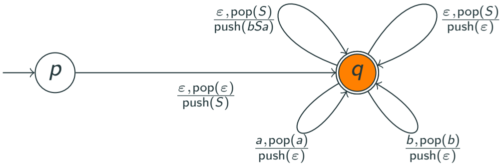


The stack

An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



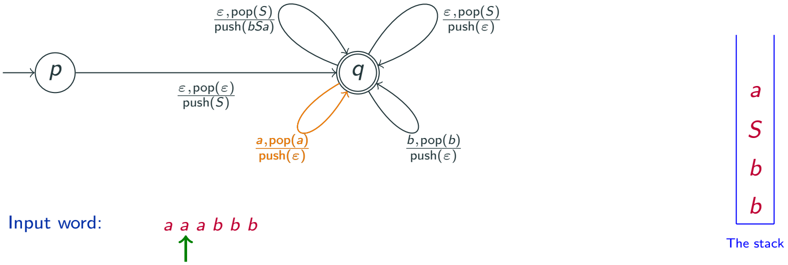
Input word: $a\ a\ a\ b\ b\ b$ ↑

The stack

An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

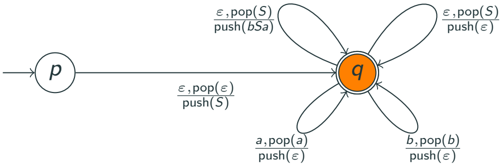
The PDA is:



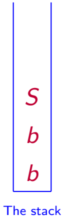
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



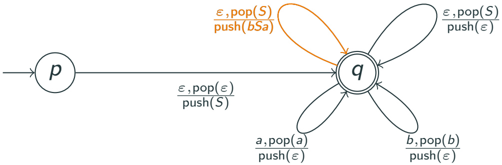
Input word: $a\ a\ a\ b\ b\ b$



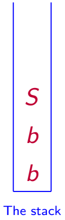
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



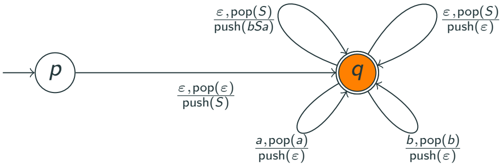
Input word: $a\ a\ a\ b\ b\ b$



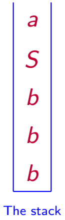
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



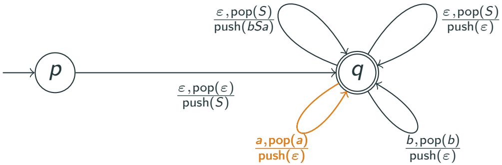
Input word: $a\ a\ a\ b\ b\ b$



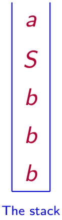
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



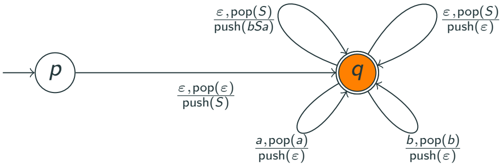
Input word: $a\ a\ a\ b\ b\ b$



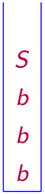
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \rightarrow aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b$

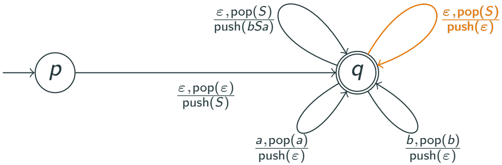


The stack

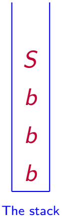
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



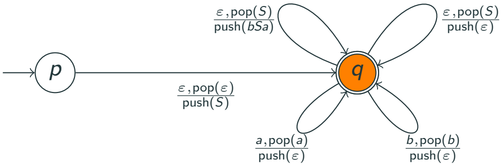
Input word: $a\ a\ a\ b\ b\ b$

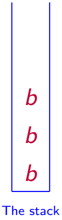


An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



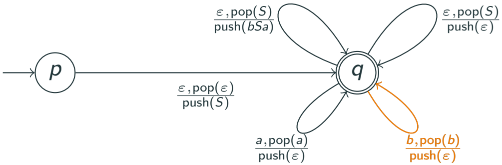


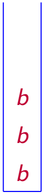
Input word: $a\ a\ a\ b\ b\ b$

An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



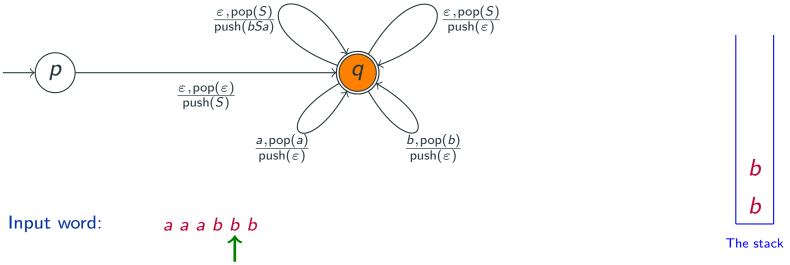


Input word: $a\ a\ a\ b\ b\ b$

An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \rightarrow aSb \mid \varepsilon$.

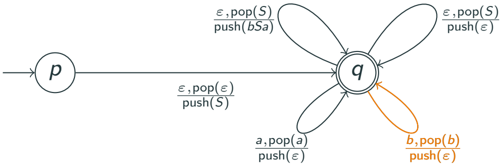
The PDA is:



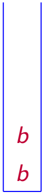
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b$

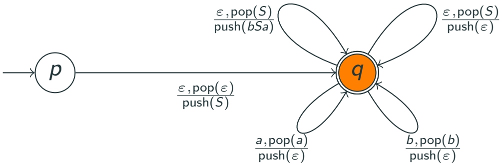


The stack

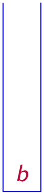
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b$

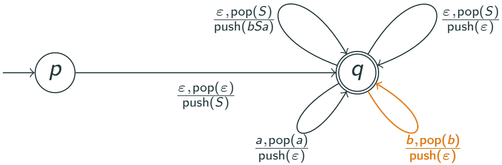


The stack

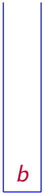
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b$

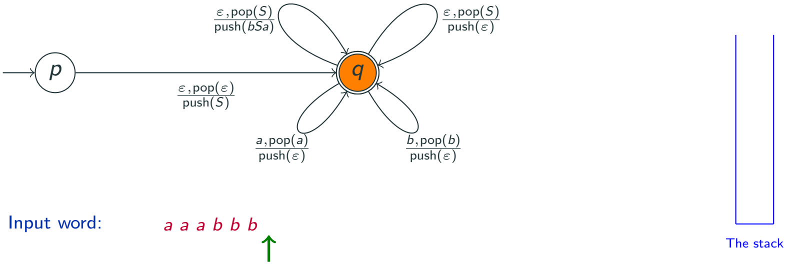


The stack

An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

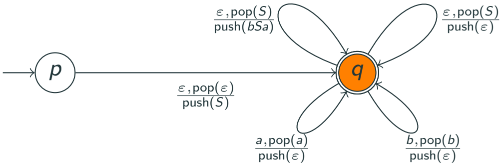
The PDA is:



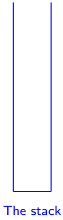
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: *a a a b b b*

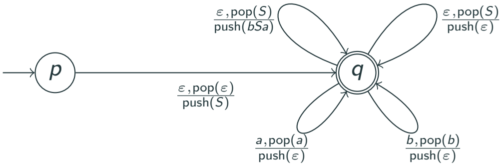


So, the word *aaabbb* is accepted by the PDA.

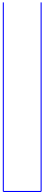
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b\ a\ b$

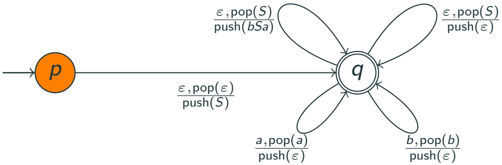


The stack

An example: From CFG to PDA

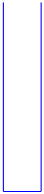
Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b\ a\ b$

$\uparrow$

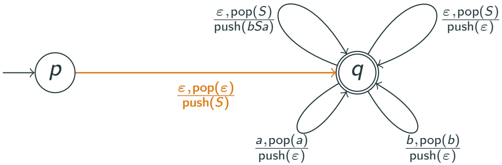


The stack

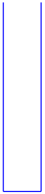
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b\ a\ b$

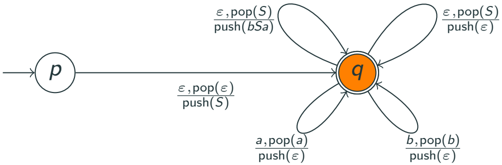


The stack

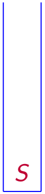
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b\ a\ b$

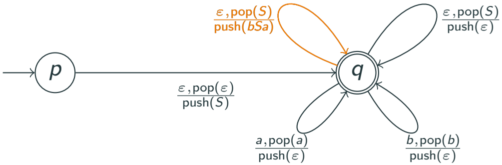


The stack

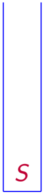
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \rightarrow aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b\ a\ b$

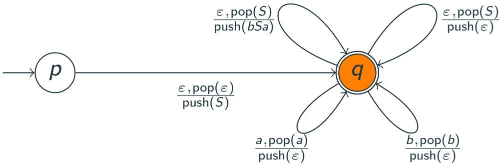


The stack

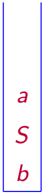
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b\ a\ b$

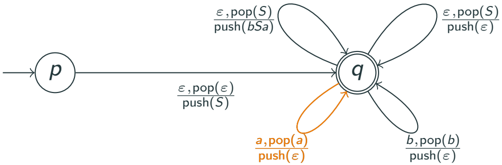


The stack

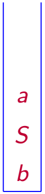
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b\ a\ b$

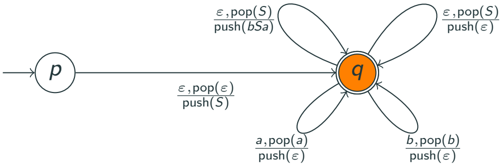


The stack

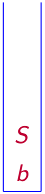
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b\ a\ b$

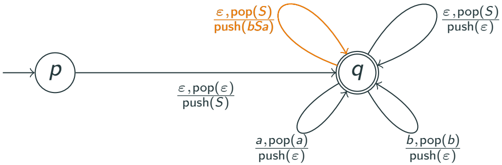


The stack

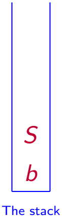
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



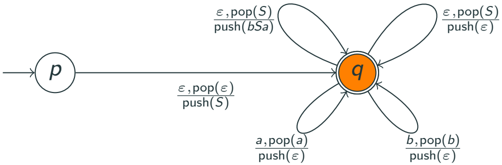
Input word: $a\ a\ a\ b\ b\ b\ a\ b$



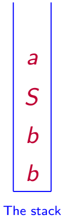
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \rightarrow aSb \mid \varepsilon$.

The PDA is:



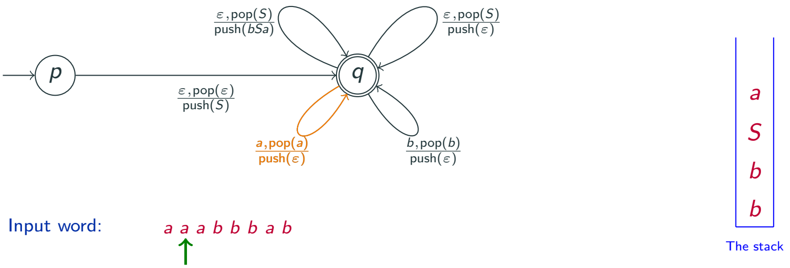
Input word: $a\ a\ a\ b\ b\ b\ a\ b$



An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \rightarrow aSb \mid \varepsilon$.

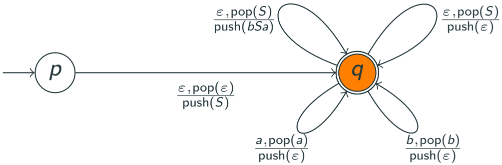
The PDA is:



An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



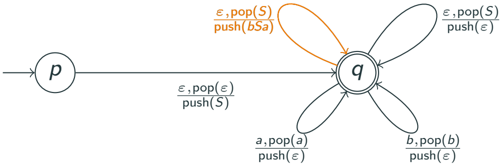
The stack

Input word: $a\ a\ a\ b\ b\ b\ a\ b$

An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



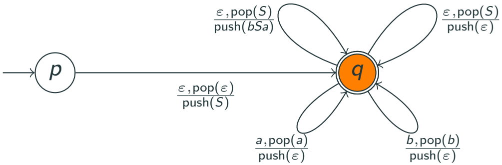
Input word: $a\ a\ a\ b\ b\ b\ a\ b$

The stack

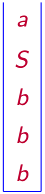
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \rightarrow aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b\ a\ b$

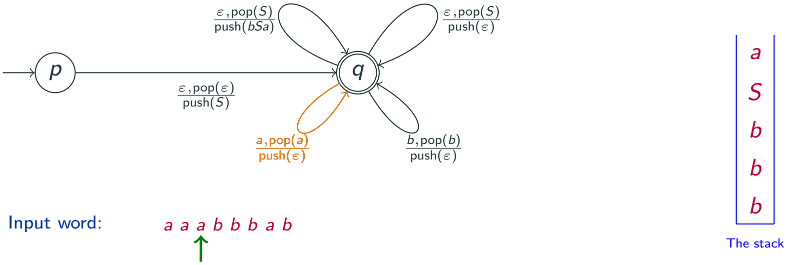


The stack

An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

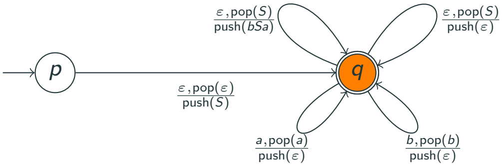
The PDA is:

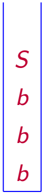


An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:





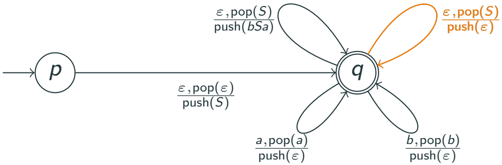
The stack

Input word: $a\ a\ a\ b\ b\ b\ a\ b$

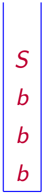
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b\ a\ b$

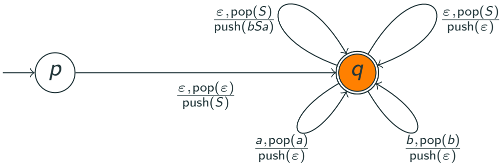


The stack

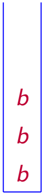
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b\ a\ b$

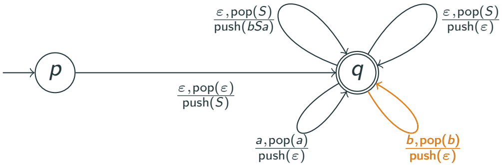


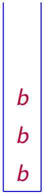
The stack

An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:





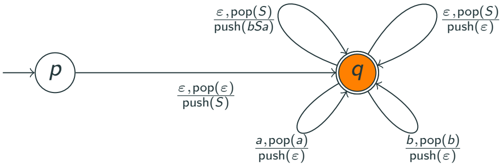
The stack

Input word: $a\ a\ a\ b\ b\ b\ a\ b$

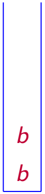
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \rightarrow aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b\ a\ b$

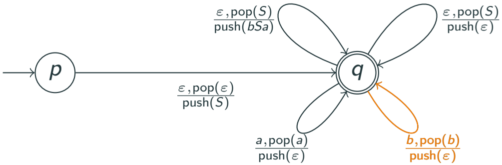


The stack

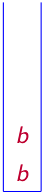
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



Input word: $a\ a\ a\ b\ b\ b\ a\ b$

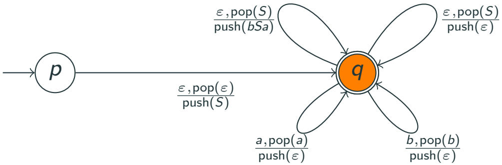


The stack

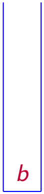
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



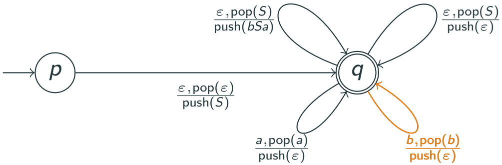
Input word: $a\ a\ a\ b\ b\ b\ a\ b$

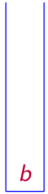


An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:





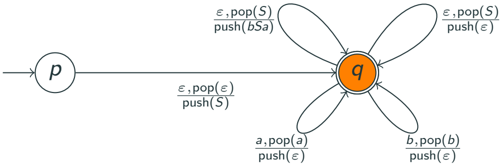
The stack

Input word: $a\ a\ a\ b\ b\ b\ a\ b$

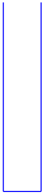
An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:



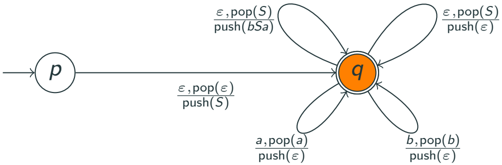
Input word: $a\ a\ a\ b\ b\ b\ a\ b$

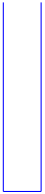


An example: From CFG to PDA

Let $\mathcal{G}$ be the CFG with rules: $S \to aSb \mid \varepsilon$.

The PDA is:





The stack

Input word: $a\ a\ a\ b\ b\ b\ a\ b$

The word *aaabbbab* is not accepted by the PDA, because the PDA cannot finish reading it.

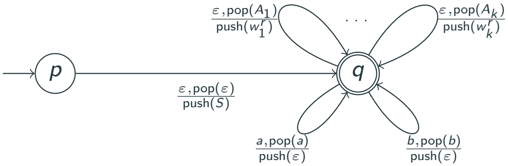
Proof of Theorem 4.1: From CFG to PDA – continued

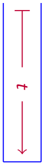
Let $\mathcal{G} = \langle \Sigma, V, R, S \rangle$ where $\Sigma = \{a, b\}$ and $R = \{A_1 \to w_1, \ldots, A_k \to w_k\}$.

Proof of Theorem 4.1: From CFG to PDA – continued

Let $\mathcal{G} = \langle \Sigma, V, R, S \rangle$ where $\Sigma = \{a, b\}$ and $R = \{A_1 \to w_1, \ldots, A_k \to w_k\}$.

Consider the following PDA with $\Sigma = \{a, b\}$ and $\Gamma = V \cup \Sigma \cup \{\bot\}$:





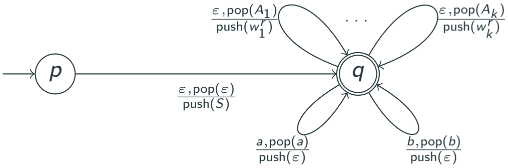
The stack

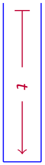
Input word: $\longleftarrow u \longrightarrow\longleftarrow v \longrightarrow$

Proof of Theorem 4.1: From CFG to PDA – continued

Let $\mathcal{G} = \langle \Sigma, V, R, S \rangle$ where $\Sigma = \{a, b\}$ and $R = \{A_1 \to w_1, \ldots, A_k \to w_k\}$.

Consider the following PDA with $\Sigma = \{a, b\}$ and $\Gamma = V \cup \Sigma \cup \{\bot\}$:





The stack

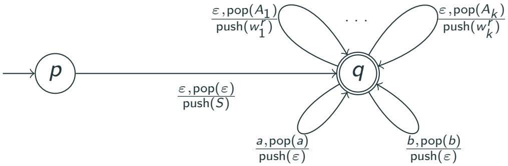
Input word:

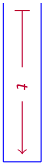
(Claim.) $t \Rightarrow^* v$ if and only if $(q, t) \vdash_v^* (q, \varepsilon)$.

Proof of Theorem 4.1: From CFG to PDA – continued

Let $\mathcal{G} = \langle \Sigma, V, R, S \rangle$ where $\Sigma = \{a, b\}$ and $R = \{A_1 \rightarrow w_1, \ldots, A_k \rightarrow w_k\}$.

Consider the following PDA with $\Sigma = \{a, b\}$ and $\Gamma = V \cup \Sigma \cup \{\perp\}$:





The stack

Input word:

(Claim.) $t \Rightarrow^* v$ if and only if $(q, t) \vdash_v^* (q, \varepsilon)$.

In particular, when $t = S$ and $u = \varepsilon$, the following holds:
$S \Rightarrow^* v$ if and only if $(q, S) \vdash_v^* (q, \varepsilon)$.

Table of contents

From PDA to CFG

Theorem (The second item of Theorem 4.1)

For every PDA $\mathcal{A}$, there is a CFG $\mathcal{G}$ such that $L(\mathcal{A}) = L(\mathcal{G})$.

From PDA to CFG

Theorem (The second item of Theorem 4.1)
For every PDA $\mathcal{A}$, there is a CFG $\mathcal{G}$ such that $L(\mathcal{A}) = L(\mathcal{G})$.

The idea is to use variables in CFG to "simulate" the runs of the PDA.

Proof of Theorem 4.1: From PDA to CFG

Let $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ be a PDA.

Proof of Theorem 4.1: From PDA to CFG

Let $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ be a PDA.

Without loss of generality, we can assume the following.

Proof of Theorem 4.1: From PDA to CFG

Let $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ be a PDA.

Without loss of generality, we can assume the following.

- It has only one final state, say q_f. That is, $F = \{q_f\}$.

Proof of Theorem 4.1: From PDA to CFG

Let $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ be a PDA.

Without loss of generality, we can assume the following.

- It has only one final state, say q_f. That is, $F = \{q_f\}$.
- The stack is empty before accepting an input word.

Proof of Theorem 4.1: From PDA to CFG

Let $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ be a PDA.

Without loss of generality, we can assume the following.

- It has only one final state, say q_f. That is, $F = \{q_f\}$.
- The stack is empty before accepting an input word.
  More precisely, on every word w, if $\mathcal{A}$ accepts w, there is an accepting run of $\mathcal{A}$ on w from the initial configuration (q_0, ε) to a final configuration (q_f, ε) where the content of the stack is empty.

Proof of Theorem 4.1: From PDA to CFG

Let $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ be a PDA.

Without loss of generality, we can assume the following.

- It has only one final state, say q_f. That is, $F = \{q_f\}$.
- The stack is empty before accepting an input word.
  More precisely, on every word w, if $\mathcal{A}$ accepts w, there is an accepting run of $\mathcal{A}$ on w from the initial configuration (q_0, ε) to a final configuration (q_f, ε) where the content of the stack is empty.
- In each transition, $\mathcal{A}$ either pushes a symbol into the stack or pops one from the stack, but it cannot do both.

Proof of Theorem 4.1: From PDA to CFG

Let $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ be a PDA.

Without loss of generality, we can assume the following.

- It has only one final state, say q_f. That is, $F = \{q_f\}$.
- The stack is empty before accepting an input word.
  More precisely, on every word w, if $\mathcal{A}$ accepts w, there is an accepting run of $\mathcal{A}$ on w from the initial configuration (q_0, ε) to a final configuration (q_f, ε) where the content of the stack is empty.
- In each transition, $\mathcal{A}$ either pushes a symbol into the stack or pops one from the stack, but it cannot do both.
  More precisely, every transition can only be of the forms:

$$
\begin{array}{rcl}
(p, x, \mathrm{pop}(y)) & \rightarrow & (q, \mathrm{push}(\varepsilon)) \\
(p, x, \mathrm{pop}(\varepsilon)) & \rightarrow & (q, \mathrm{push}(z))
\end{array}
$$

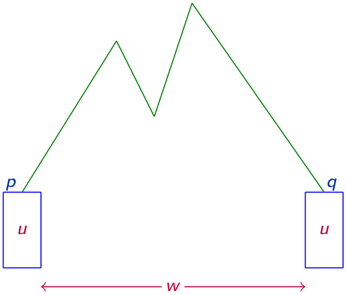
Proof of Theorem 4.1: From PDA to CFG – continued

Let $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$ be a PDA.

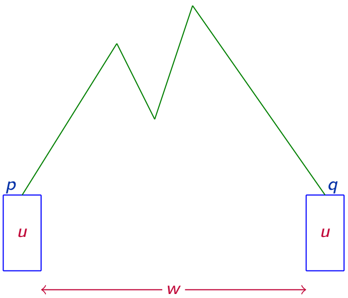
Consider the following CFG $\mathcal{G} = \langle \Sigma, V, R, S \rangle$:

- $V = \{A_{p,q} \mid p, q \in Q\}$.
- A_{q_0,q_f} is the start variable.
- The set of rules R will be defined later.

The intuitive meaning of the variable $A_{p,q}$



The intuitive meaning of the variable $A_{p,q}$



Then, $A_{p,q} \Rightarrow^* w$.

Defining the rules in R – part. 1

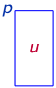
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \to (r, \mathrm{push}(z)) \quad \text{and} \quad (s, b, \mathrm{pop}(z)) \to (q, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \ \to \ a \ A_{r,s} \ b$$

Defining the rules in R – part. 1



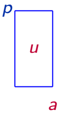
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \to (r, \mathrm{push}(z)) \quad \text{and} \quad (s, b, \mathrm{pop}(z)) \to (q, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \;\to\; a \; A_{r,s} \; b$$

Defining the rules in R – part. 1



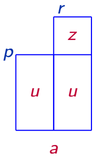
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \to (r, \mathrm{push}(z)) \quad \text{and} \quad (s, b, \mathrm{pop}(z)) \to (q, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \ \to \ a \ A_{r,s} \ b$$

Defining the rules in R – part. 1



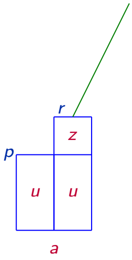
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \to (r, \mathrm{push}(z)) \quad \text{and} \quad (s, b, \mathrm{pop}(z)) \to (q, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \;\to\; a\; A_{r,s}\; b$$

Defining the rules in R – part. 1



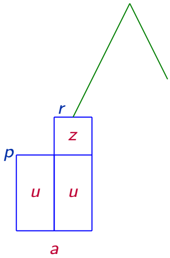
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \to (r, \mathrm{push}(z)) \quad \text{and} \quad (s, b, \mathrm{pop}(z)) \to (q, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \;\to\; a\; A_{r,s}\; b$$

Defining the rules in R – part. 1



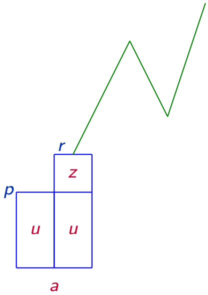
For every transition:

$$(p, a, \text{pop}(\varepsilon)) \rightarrow (r, \text{push}(z)) \quad \text{and} \quad (s, b, \text{pop}(z)) \rightarrow (q, \text{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \;\rightarrow\; a \; A_{r,s} \; b$$

Defining the rules in R – part. 1



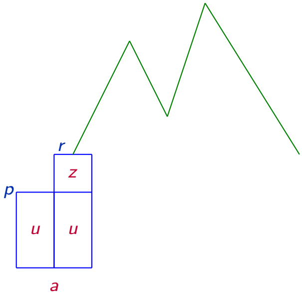
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \to (r, \mathrm{push}(z)) \quad \text{and} \quad (s, b, \mathrm{pop}(z)) \to (q, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \ \to \ a \ A_{r,s} \ b$$

Defining the rules in R – part. 1



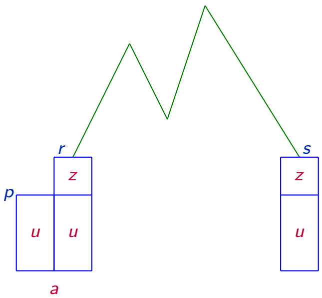
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \rightarrow (r, \mathrm{push}(z)) \quad \text{and} \quad (s, b, \mathrm{pop}(z)) \rightarrow (q, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \;\rightarrow\; a \; A_{r,s} \; b$$

Defining the rules in R – part. 1



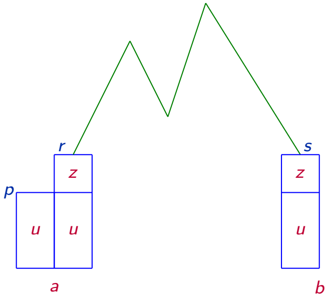
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \to (r, \mathrm{push}(z)) \quad \text{and} \quad (s, b, \mathrm{pop}(z)) \to (q, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \;\to\; a \; A_{r,s} \; b$$

Defining the rules in R – part. 1



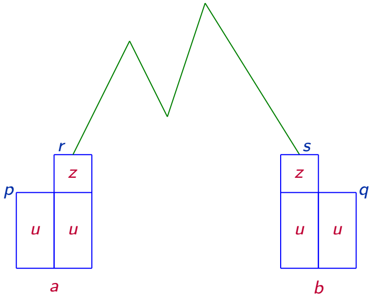
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \to (r, \mathrm{push}(z)) \quad \text{and} \quad (s, b, \mathrm{pop}(z)) \to (q, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \;\to\; a\ A_{r,s}\ b$$

Defining the rules in R – part. 1



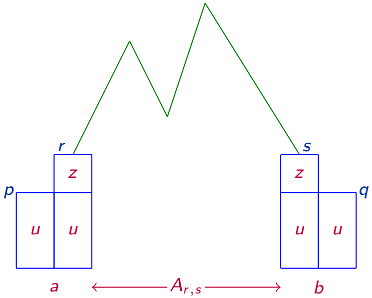
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \to (r, \mathrm{push}(z)) \quad \text{and} \quad (s, b, \mathrm{pop}(z)) \to (q, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \ \to \ a \ A_{r,s} \ b$$

Defining the rules in R – part. 1



For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \to (r, \mathrm{push}(z)) \quad \text{and} \quad (s, b, \mathrm{pop}(z)) \to (q, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \ \to \ a \ A_{r,s} \ b$$

Defining the rules in R – part. 2

For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \;\rightarrow\; (r, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \;\rightarrow\; a \; A_{r,q}$$

Defining the rules in R – part. 2

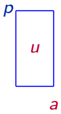
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \;\rightarrow\; (r, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \;\rightarrow\; a\ A_{r,q}$$

Defining the rules in R – part. 2



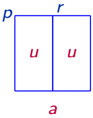
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \;\rightarrow\; (r, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \;\rightarrow\; a\ A_{r,q}$$

Defining the rules in R – part. 2



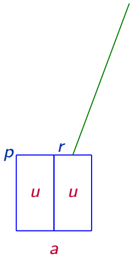
For every transition:

$$(p, a, \text{pop}(\varepsilon)) \;\rightarrow\; (r, \text{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \;\rightarrow\; a\ A_{r,q}$$

Defining the rules in R – part. 2



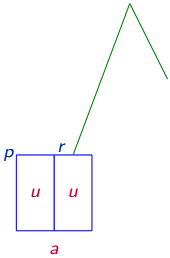
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \ \rightarrow \ (r, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \ \rightarrow \ a \ A_{r,q}$$

Defining the rules in R – part. 2



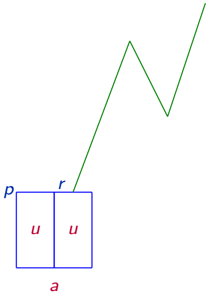
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \;\rightarrow\; (r, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \;\rightarrow\; a\ A_{r,q}$$

Defining the rules in R – part. 2



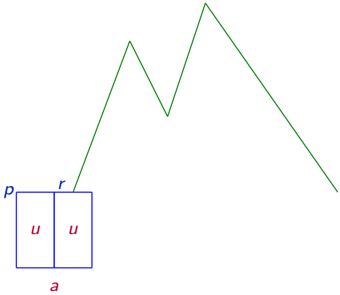
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \;\rightarrow\; (r, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \;\rightarrow\; a\ A_{r,q}$$

Defining the rules in R – part. 2



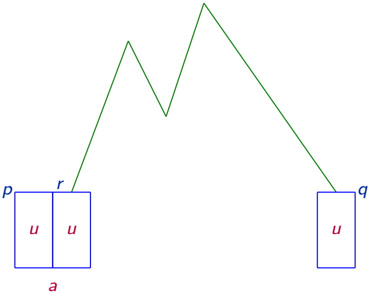
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \;\rightarrow\; (r, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \;\rightarrow\; a\ A_{r,q}$$

Defining the rules in R – part. 2



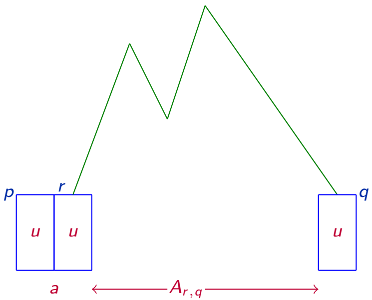
For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \;\rightarrow\; (r, \mathrm{push}(\varepsilon))$$

the following rule is in R:

$$A_{p,q} \;\rightarrow\; a\ A_{r,q}$$

Defining the rules in R – part. 2



For every transition:

$$(p, a, \mathrm{pop}(\varepsilon)) \;\rightarrow\; (r, \mathrm{push}(\varepsilon))$$

the following rule is in R:

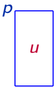
$$A_{p,q} \;\rightarrow\; a\; A_{r,q}$$

Defining the set of rules in R – part. 3

For every $r \in Q$, the following rule is in R:

$$A_{p,q} \;\rightarrow\; A_{p,r}\; A_{r,q}$$

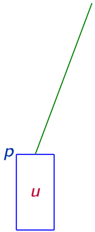
Defining the set of rules in R – part. 3



For every $r \in Q$, the following rule is in R:

$$A_{p,q} \ \rightarrow \ A_{p,r} \ A_{r,q}$$

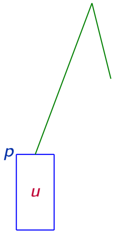
Defining the set of rules in R – part. 3



For every $r \in Q$, the following rule is in R:

$$A_{p,q} \;\rightarrow\; A_{p,r}\ A_{r,q}$$

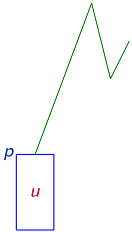
Defining the set of rules in R – part. 3



For every $r \in Q$, the following rule is in R:

$$A_{p,q} \;\rightarrow\; A_{p,r}\; A_{r,q}$$

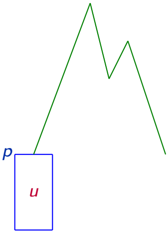
Defining the set of rules in R – part. 3



For every $r \in Q$, the following rule is in R:

$$A_{p,q} \;\rightarrow\; A_{p,r}\; A_{r,q}$$

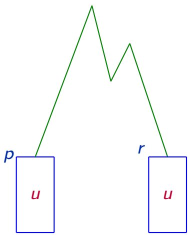
Defining the set of rules in R – part. 3



For every $r \in Q$, the following rule is in R:

$$A_{p,q} \ \rightarrow \ A_{p,r} \ A_{r,q}$$

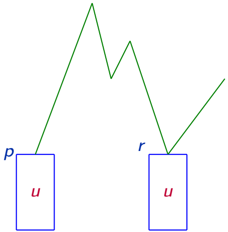
Defining the set of rules in R – part. 3



For every $r \in Q$, the following rule is in R:

$$A_{p,q} \rightarrow A_{p,r}\ A_{r,q}$$

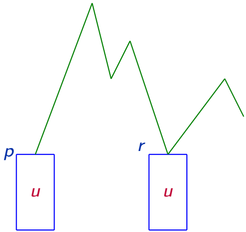
Defining the set of rules in R – part. 3



For every $r \in Q$, the following rule is in R:

$$A_{p,q} \;\rightarrow\; A_{p,r}\; A_{r,q}$$

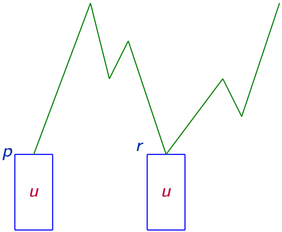
Defining the set of rules in R – part. 3



For every $r \in Q$, the following rule is in R:

$$A_{p,q} \;\rightarrow\; A_{p,r}\; A_{r,q}$$

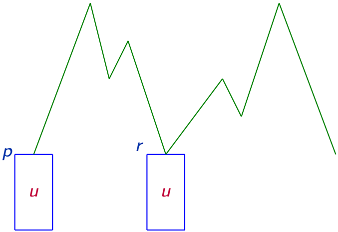
Defining the set of rules in R – part. 3



For every $r \in Q$, the following rule is in R:

$$A_{p,q} \rightarrow A_{p,r}\ A_{r,q}$$

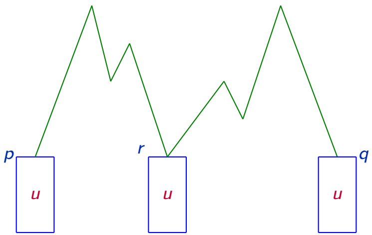
Defining the set of rules in R – part. 3



For every $r \in Q$, the following rule is in R:

$$A_{p,q} \rightarrow A_{p,r} \, A_{r,q}$$

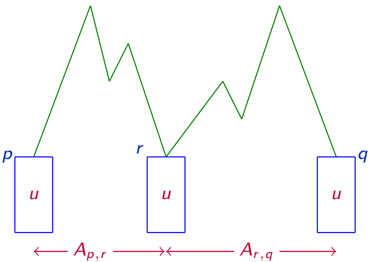
Defining the set of rules in R – part. 3



For every $r \in Q$, the following rule is in R:

$$A_{p,q} \rightarrow A_{p,r}\ A_{r,q}$$

Defining the set of rules in R – part. 3



For every $r \in Q$, the following rule is in R:

$$A_{p,q} \rightarrow A_{p,r}\ A_{r,q}$$

Defining the set of rules in R – part. 4

For every $p \in Q$, the following rule is in R:

$$A_{p,p} \rightarrow \varepsilon$$

Defining the set of rules in R – part. 4

For every $p \in Q$, the following rule is in R:

$$A_{p,p} \rightarrow \varepsilon$$

Proof of Theorem 4.1: From PDA to CFG – continued

For a PDA $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$, its equivalent CFG is:

Proof of Theorem 4.1: From PDA to CFG – continued

For a PDA $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$, its equivalent CFG is:

$\mathcal{G} = \langle \Sigma, V, R, S \rangle$, where $V = \{A_{p,q} \mid p, q \in Q\}$ and A_{q_0,q_f} is the start variable.

Proof of Theorem 4.1: From PDA to CFG – continued

For a PDA $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$, its equivalent CFG is:

$\mathcal{G} = \langle \Sigma, V, R, S \rangle$, where $V = \{A_{p,q} \mid p, q \in Q\}$ and A_{q_0,q_f} is the start variable.

The rules in R are as follows.

- For every transition $(p, a, \mathrm{pop}(\varepsilon)) \to (r, \mathrm{push}(z))$ and $(s, b, \mathrm{pop}(z)) \to (q, \mathrm{push}(\varepsilon))$, the rule $A_{p,q} \; \to \; a \; A_{r,s} \; b$ is in R.
- For every transition $(p, a, \mathrm{pop}(\varepsilon)) \; \to \; (r, \mathrm{push}(\varepsilon))$, the rule $A_{p,q} \; \to \; a \; A_{r,q}$ is in R.
- For every $r \in Q$, the rule $A_{p,q} \; \to \; A_{p,r} \; A_{r,q}$ is in R.
- For every $p \in Q$, the rule $A_{p,p} \; \to \; \varepsilon$ is in R.

Proof of Theorem 4.1: From PDA to CFG – continued

For a PDA $\mathcal{A} = \langle \Sigma, \Gamma, Q, q_0, F, \delta \rangle$, its equivalent CFG is:

$\mathcal{G} = \langle \Sigma, V, R, S \rangle$, where $V = \{A_{p,q} \mid p, q \in Q\}$ and A_{q_0,q_f} is the start variable.

The rules in R are as follows.

- For every transition $(p, a, \text{pop}(\varepsilon)) \to (r, \text{push}(z))$ and $(s, b, \text{pop}(z)) \to (q, \text{push}(\varepsilon))$, the rule $A_{p,q} \to a\ A_{r,s}\ b$ is in R.
- For every transition $(p, a, \text{pop}(\varepsilon)) \to (r, \text{push}(\varepsilon))$, the rule $A_{p,q} \to a\ A_{r,q}$ is in R.
- For every $r \in Q$, the rule $A_{p,q} \to A_{p,r}\ A_{r,q}$ is in R.
- For every $p \in Q$, the rule $A_{p,p} \to \varepsilon$ is in R.

Claim

$L(\mathcal{A}) = L(\mathcal{G})$.

From PDA to CFG

Theorem (The second item of Theorem 4.1)

For every PDA $\mathcal{A}$, there is a CFG $\mathcal{G}$ such that $L(\mathcal{A}) = L(\mathcal{G})$.

To conclude:

CFG and PDA express exactly the same class of languages.

To conclude:

CFG and PDA express exactly the same class of languages.

Theorem 4.1

- *For every CFG $\mathcal{G}$, there is a PDA $\mathcal{A}$ such that $L(\mathcal{A}) = L(\mathcal{G})$.*
- *For every PDA $\mathcal{A}$, there is a CFG $\mathcal{G}$ such that $L(\mathcal{A}) = L(\mathcal{G})$.*

End of Lesson 4