Pushdown automata

Pushdown automaton

Example: Consider the language over the alphabet $\Sigma = \{(,),[,],<,>\}$ consisting of "correctly parenthesised", i.e., the sequences where every left parenthesis has a corresponding right parenthesis, and where paretheses do not "cross" (as for example in the word `<[>]`).

This language is generated by a context-free grammar

$$A \rightarrow \varepsilon \mid (A) \mid [A] \mid <A> \mid AA$$

A typical example of a word that belongs to this language:

`<[](()[<>])>[]`

It is not hard to show that this language is not regular.

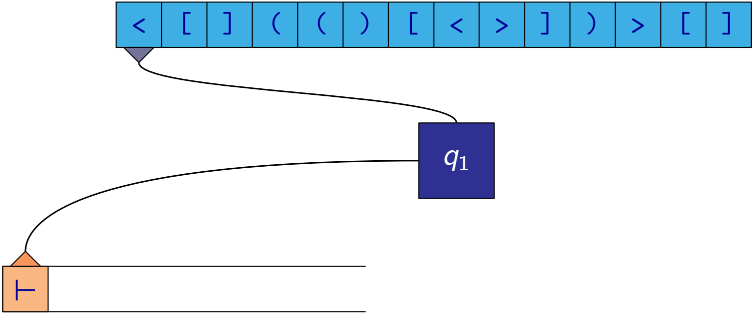
Pushdown automaton

We would like to construct a device, similar to a finite automaton, that would be able to recognize words from this language.

An approppriate possibility seems to be to use a **stack** (of unbounded size) for this recognition.

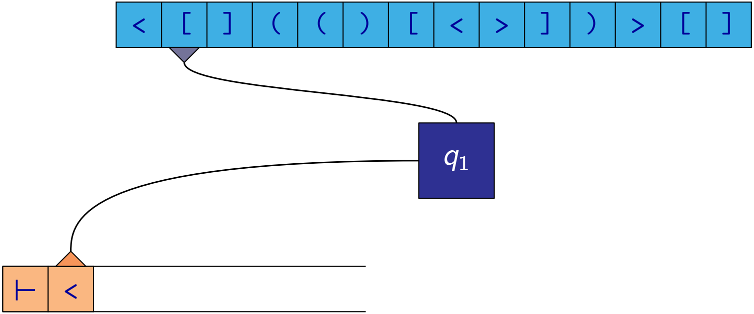
Pushdown automaton

- Word <[](()[<>])>[] belongs to the language.



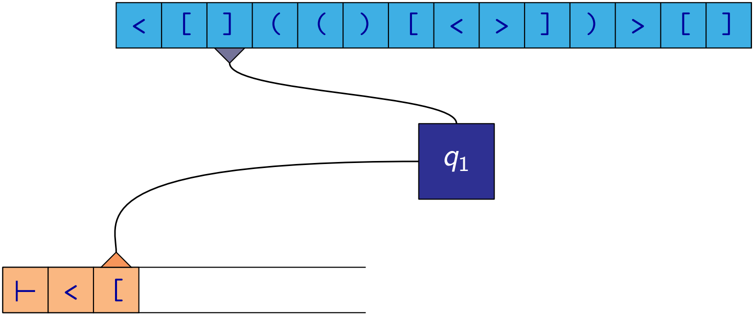
Pushdown automaton

- Word `<[]((()[<>])>[]` belongs to the language.



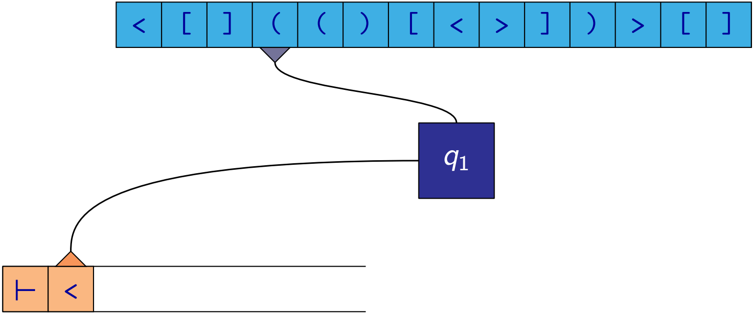
Pushdown automaton

- Word `<[](()[<>])>[]` belongs to the language.



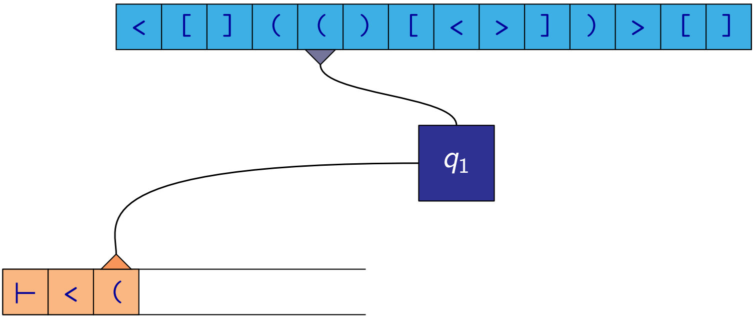
Pushdown automaton

- Word `<[]((()[<>])>[]` belongs to the language.



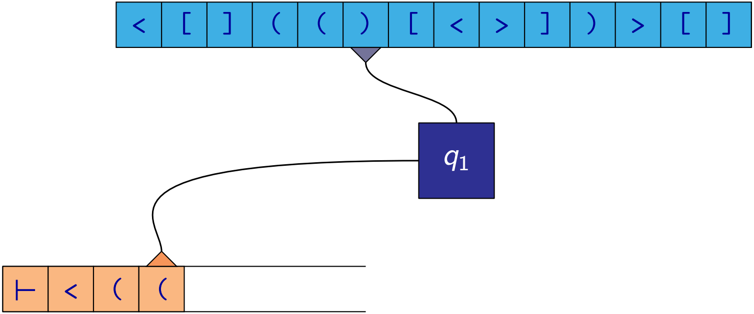
Pushdown automaton

- Word <[](()[<>])>[] belongs to the language.



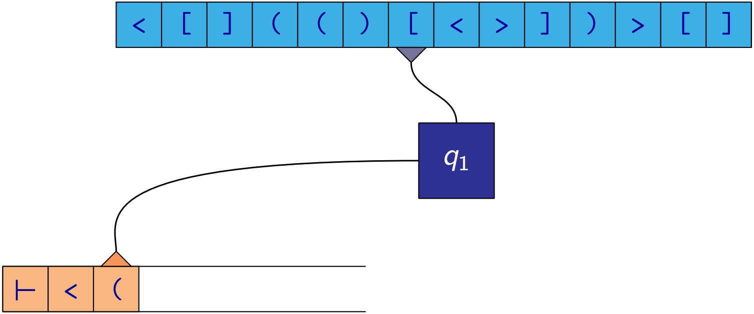
Pushdown automaton

- Word `<[](()[<>])>[]` belongs to the language.



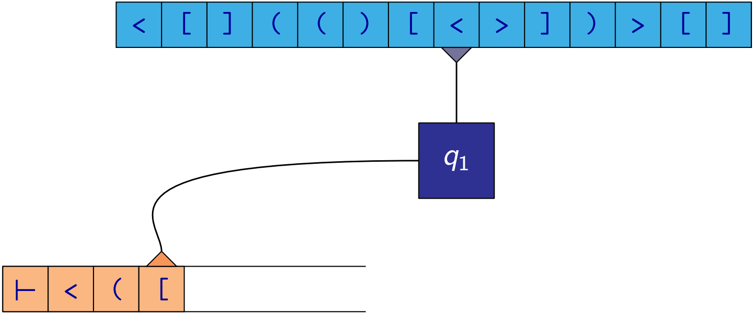
Pushdown automaton

- Word `<[](()[<>])>[]` belongs to the language.



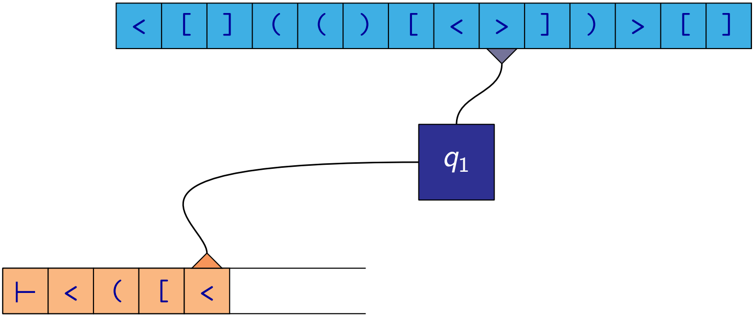
Pushdown automaton

- Word `<[](()[<>])>[]` belongs to the language.



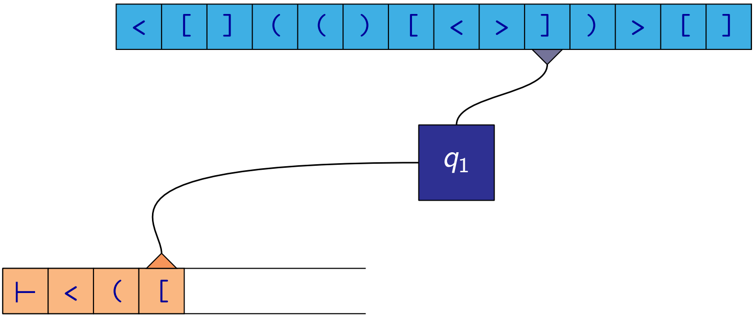
Pushdown automaton

- Word `<[](()[<>])>[]` belongs to the language.



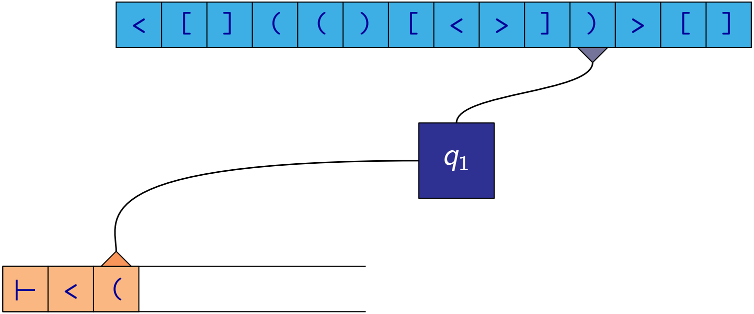
Pushdown automaton

- Word <[](()[<>])>[] belongs to the language.



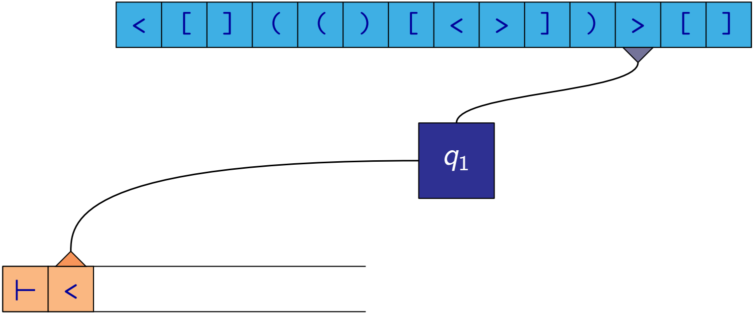
Pushdown automaton

- Word `<[](()[<>])>[]` belongs to the language.



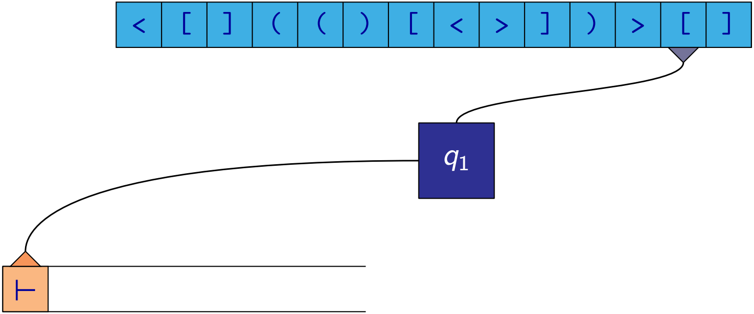
Pushdown automaton

- Word `<[](()[<>])>[]` belongs to the language.



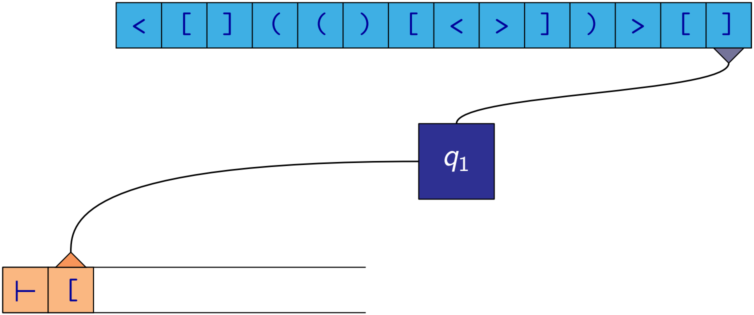
Pushdown automaton

- Word <[](()[<>])>[] belongs to the language.



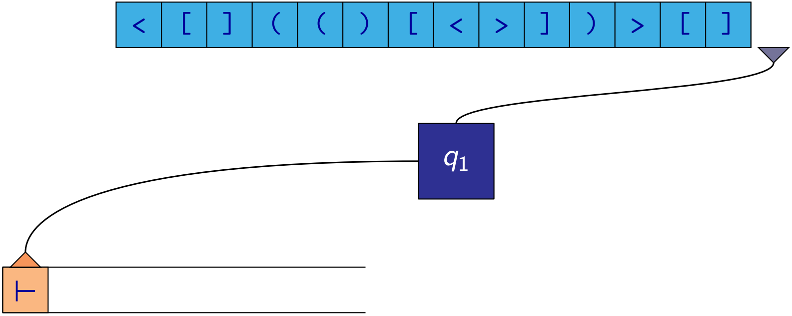
Pushdown automaton

- Word `<[](()[<>])>[]` belongs to the language.



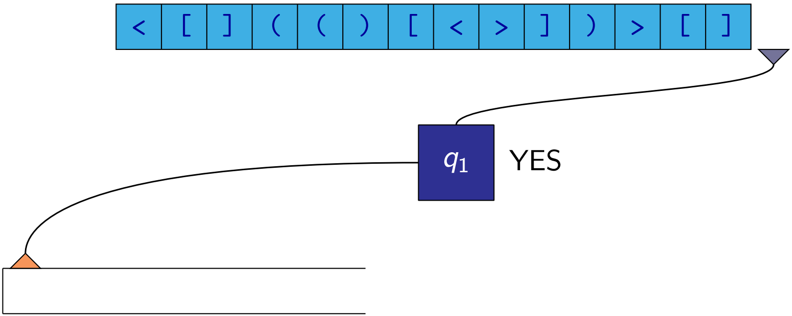
Pushdown automaton

- Word `<[](()[<>])>[]` belongs to the language.



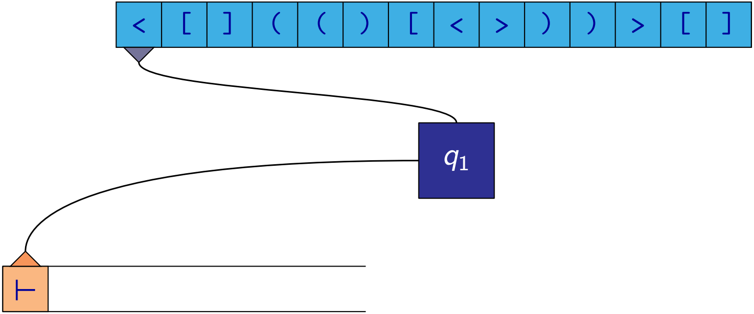
Pushdown automaton

- Word <[](()[<>])>[] belongs to the language.
- The automaton has read the whole word and ends with an empty stack, and so the word is accepted by the automaton.



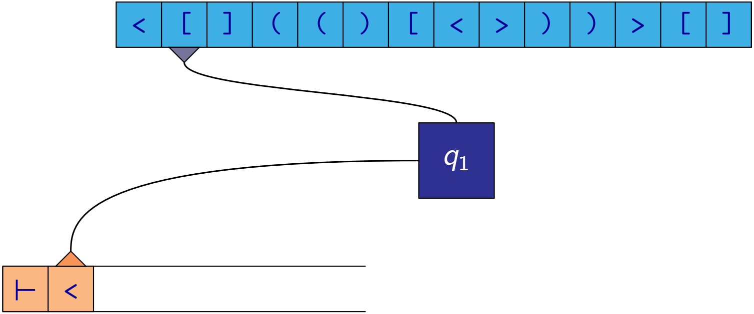
Pushdown automaton

- Word `<[](()[<>))>[]` does not belong to the language.



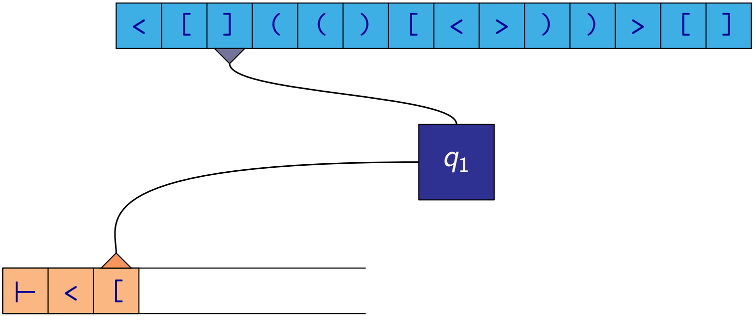
Pushdown automaton

- Word `<[](()[<>))>[]` does not belong to the language.



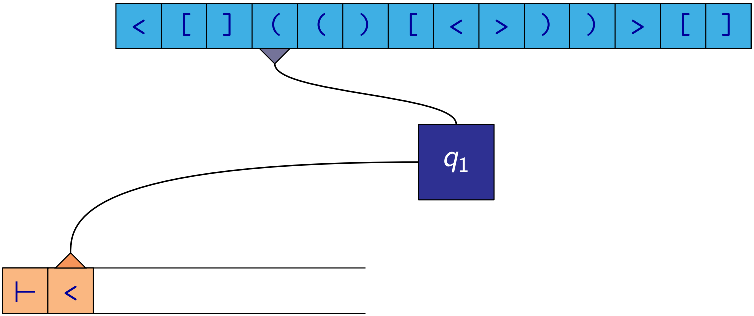
Pushdown automaton

- Word `<[](()[<>))>[]` does not belong to the language.



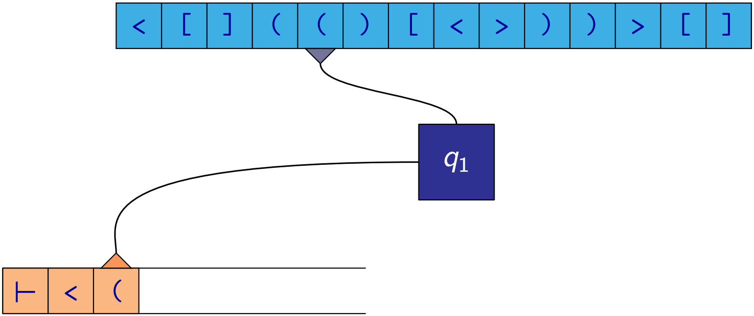
Pushdown automaton

- Word <[](()[<>))>[] does not belong to the language.



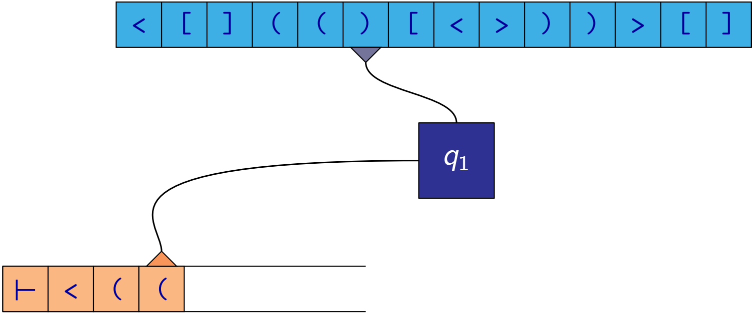
Pushdown automaton

- Word `<[](()[<>))>[]` does not belong to the language.



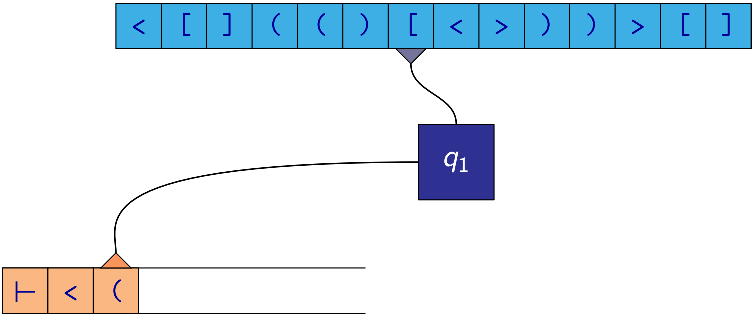
Pushdown automaton

- Word `<[](()[<>))>[]` does not belong to the language.



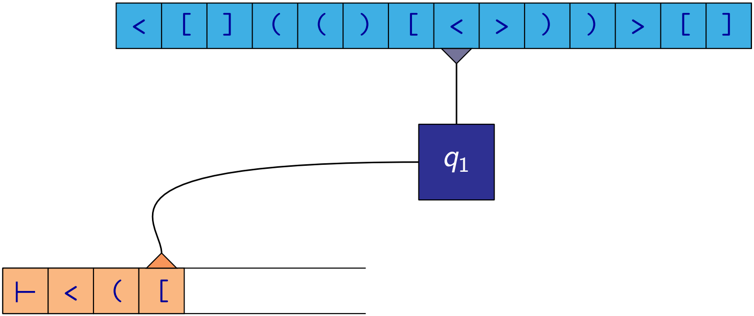
Pushdown automaton

- Word <[](()[<>))>[] does not belong to the language.



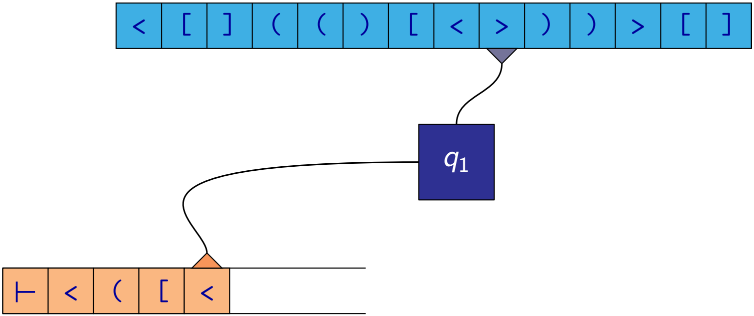
Pushdown automaton

- Word <[](()[<>))>[] does not belong to the language.



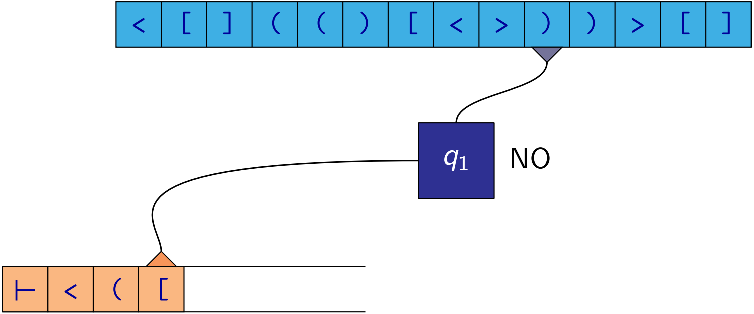
Pushdown automaton

- Word `<[](()[<>))>[]` does not belong to the language.



Pushdown automaton

- Word `<[](()[<>))>[]` does not belong to the language.
- The automaton has found a parenthesis that does not match, so the word is not accepted.



Pushdown automaton

Example:

- We would like to recognize language $L = \{a^n b^n \mid n \geq 1\}$

Again, it is a typical example of a non-regular language.

Pushdown automaton

Example:

- We would like to recognize language $L = \{a^n b^n \mid n \geq 1\}$

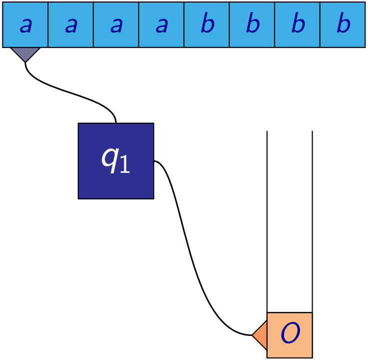
Again, it is a typical example of a non-regular language.

A stack can be used as a counter:

- Symbols of one kind (called for example I) will be pushed to it.
- A number of occurrences of these symbols I on the stack repsents a value of the counter.

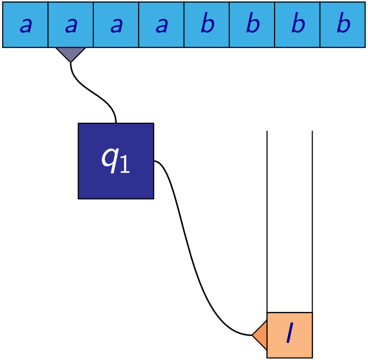
Pushdown automaton

- Word *aaaabbbb* belongs to the language $L = \{a^n b^n \mid n \geq 1\}$



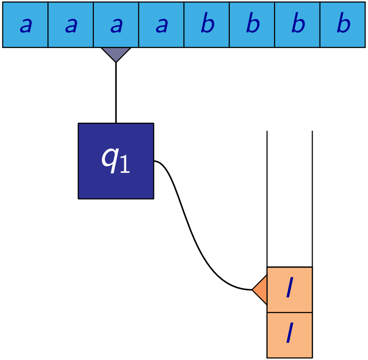
Pushdown automaton

- Word *aaaabbbb* belongs to the language $L = \{a^n b^n \mid n \geq 1\}$



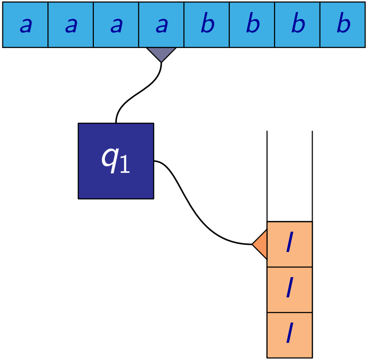
Pushdown automaton

- Word *aaaabbbb* belongs to the language $L = \{a^n b^n \mid n \geq 1\}$



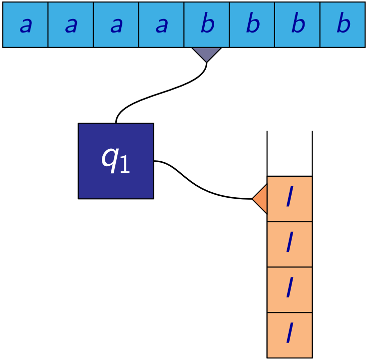
Pushdown automaton

- Word *aaaabbbb* belongs to the language $L = \{a^n b^n \mid n \geq 1\}$



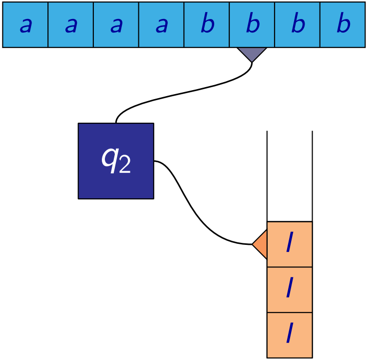
Pushdown automaton

- Word *aaaabbbb* belongs to the language $L = \{a^n b^n \mid n \geq 1\}$



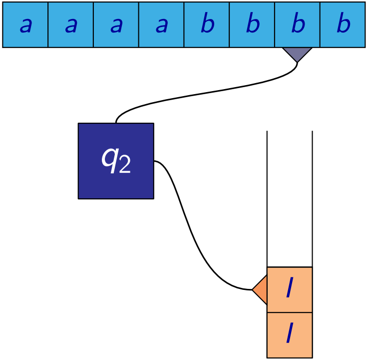
Pushdown automaton

- Word *aaaabbbb* belongs to the language $L = \{a^n b^n \mid n \geq 1\}$



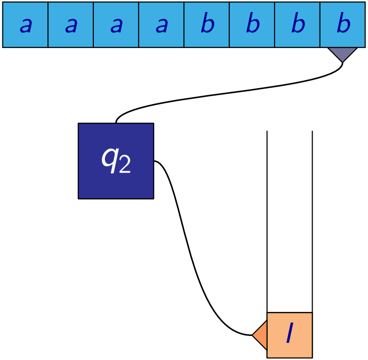
Pushdown automaton

- Word *aaaabbbb* belongs to the language $L = \{a^n b^n \mid n \geq 1\}$



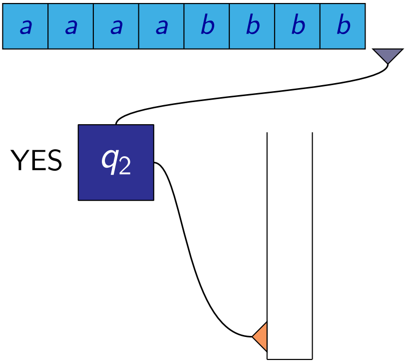
Pushdown automaton

- Word *aaaabbbb* belongs to the language $L = \{a^n b^n \mid n \geq 1\}$



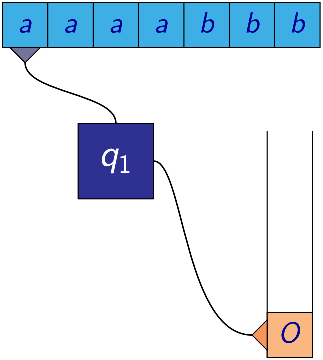
Pushdown automaton

- Word *aaaabbbb* belongs to the language $L = \{a^n b^n \mid n \geq 1\}$
- The automaton has read the whole word and ends with an empty stack, and so the word is accepted by the automaton.



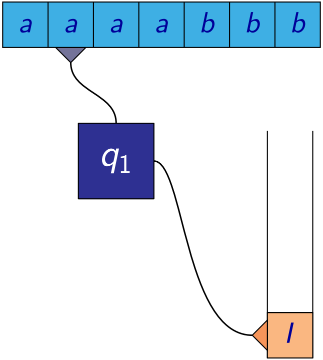
Pushdown automaton

- Word *aaaabbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



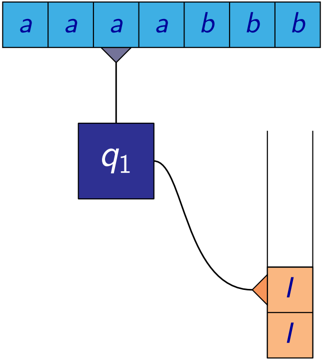
Pushdown automaton

- Word *aaaabbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



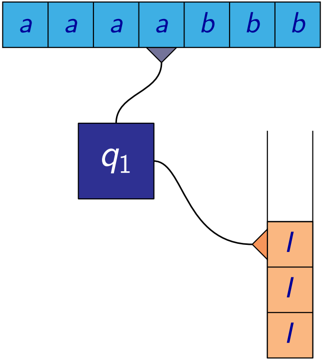
Pushdown automaton

- Word *aaaabbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



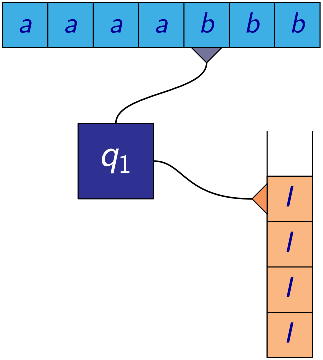
Pushdown automaton

- Word *aaaabbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



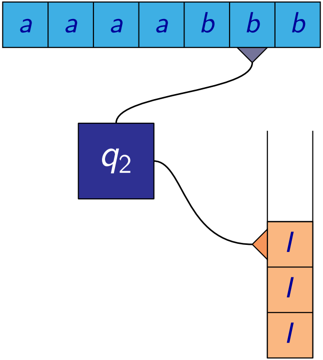
Pushdown automaton

- Word *aaaabbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



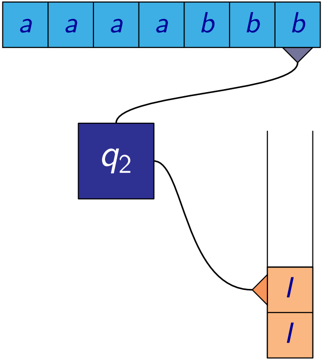
Pushdown automaton

- Word *aaaabbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



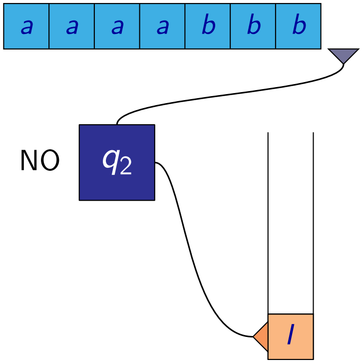
Pushdown automaton

- Word *aaaabbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



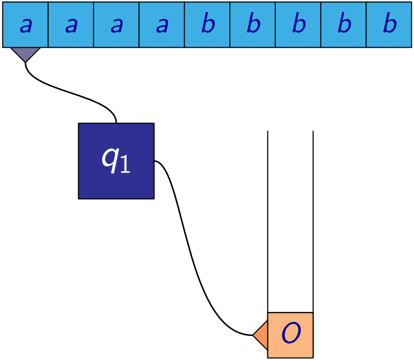
Pushdown automaton

- Word *aaaabbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$
- The automaton has read all word but the stack is not empty and so the word is not accepted by the automaton.



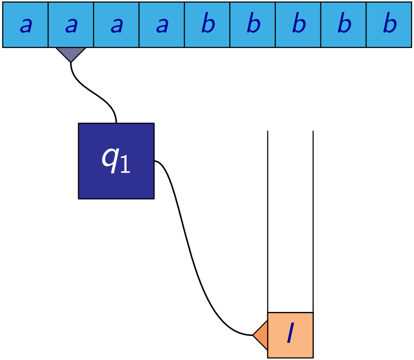
Pushdown automaton

- Word *aaaabbbbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



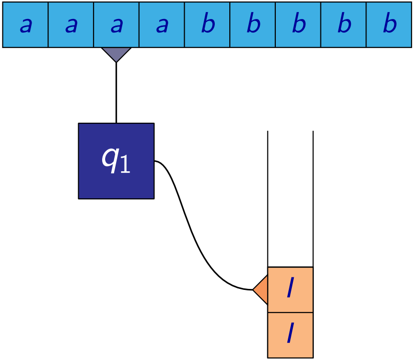
Pushdown automaton

- Word *aaaabbbbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



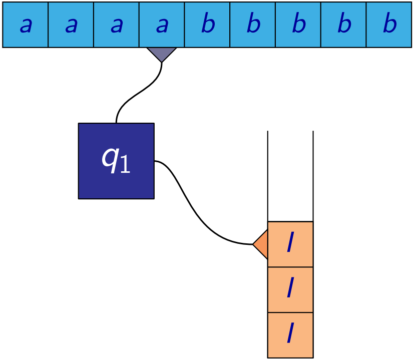
Pushdown automaton

- Word *aaaabbbbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



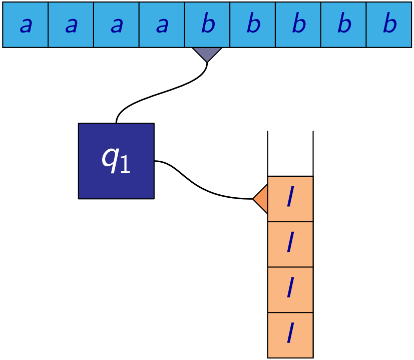
Pushdown automaton

- Word *aaaabbbbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



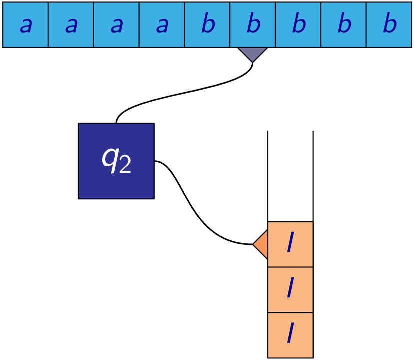
Pushdown automaton

- Word *aaaabbbbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



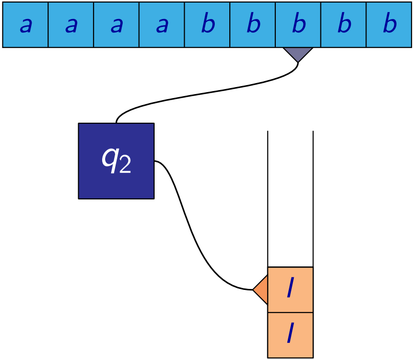
Pushdown automaton

- Word *aaaabbbbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



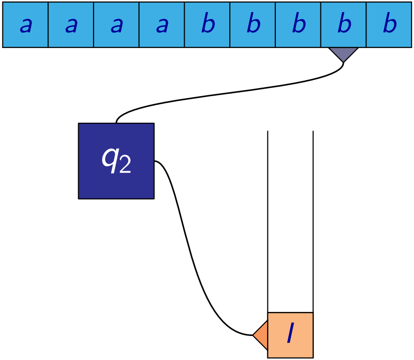
Pushdown automaton

- Word *aaaabbbbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



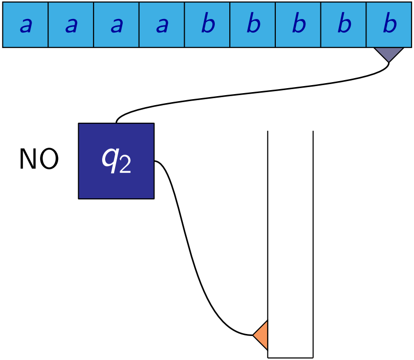
Pushdown automaton

- Word *aaaabbbbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



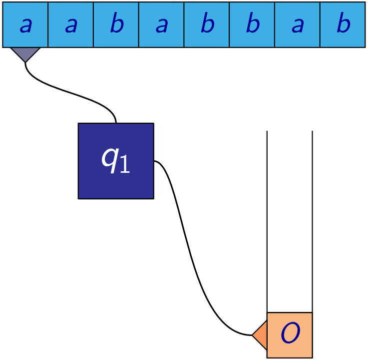
Pushdown automaton

- Word *aaaabbbbb* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$
- The automaton reads b, it should remove a symbol from the stack but there is no symbol there. So the word is not accepted.



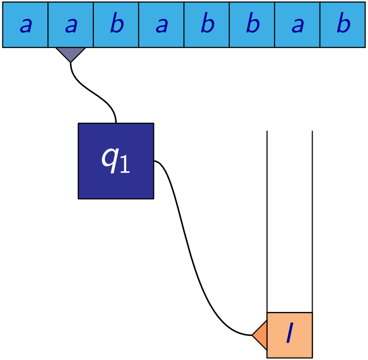
Pushdown automaton

- Word *aababbab* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



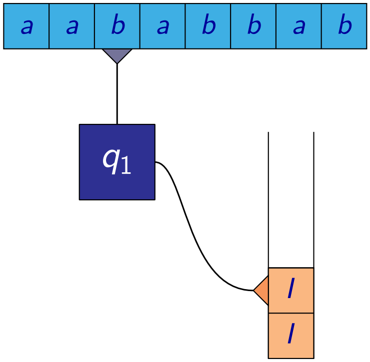
Pushdown automaton

- Word *aababbab* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



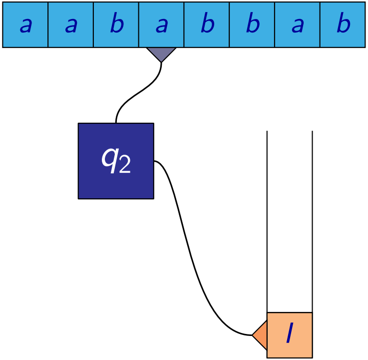
Pushdown automaton

- Word *aababbab* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



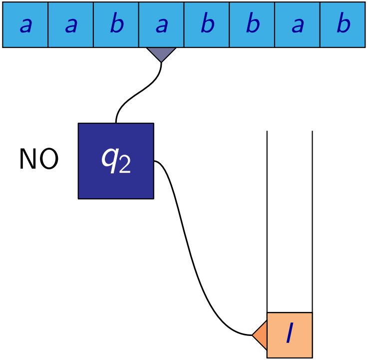
Pushdown automaton

- Word *aababbab* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$



Pushdown automaton

- Word *aababbab* does not belong to language $L = \{a^n b^n \mid n \geq 1\}$
- The automaton has read *a* but it is already in the state where it removes symbols from the stack, and so the word is not accepted.



Pushdown automaton

- A pushdown automaton can be nondeterministic and it can have ε-transitions.

Pushdown automaton

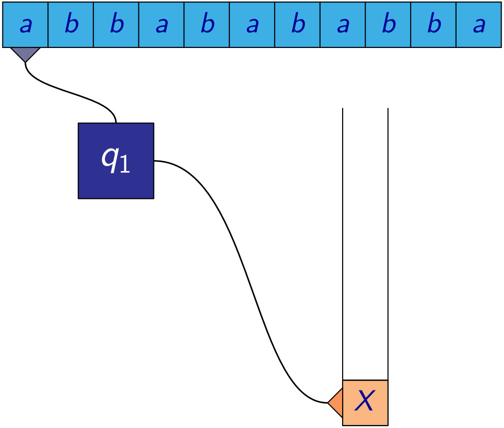
- A pushdown automaton can be nondeterministic and it can have ε-transitions.

Example:

- Let us consider the language $L = \{w \in \{a, b\}^* \mid w = w^R\}$.
- The first half of a word can be stored on the stack.
- When reading the second part, the automaton removes the symbols from the stack if they are same as symbols in the input.
- If the stack is empty after reading all word, the second is the same (the reverse of) the first.
- The automaton can nondeterministically guess the position of the "boundery" between the first and the second half of the word. Those computations where the automaton guesses wrong are nonaccepting.

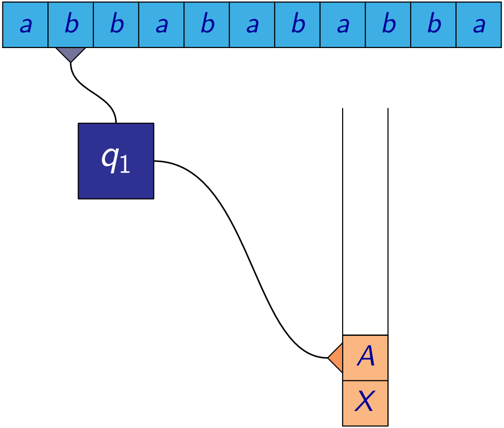
Pushdown automaton

- Word *abbabababba* belongs to the language $L = \{w \in \{a, b\}^* \mid w = w^R\}$



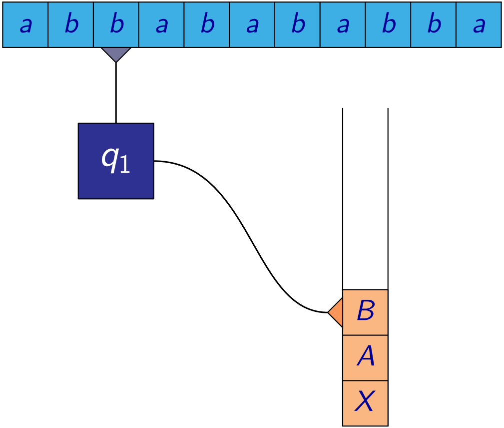
Pushdown automaton

- Word *abbabababba* belongs to the language $L = \{w \in \{a, b\}^* \mid w = w^R\}$



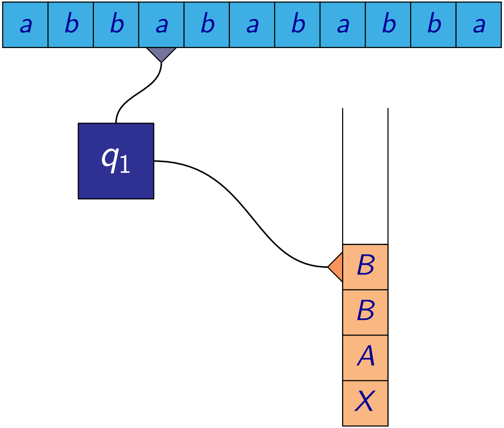
Pushdown automaton

- Word *abbababababba* belongs to the language $L = \{w \in \{a,b\}^* \mid w = w^R\}$



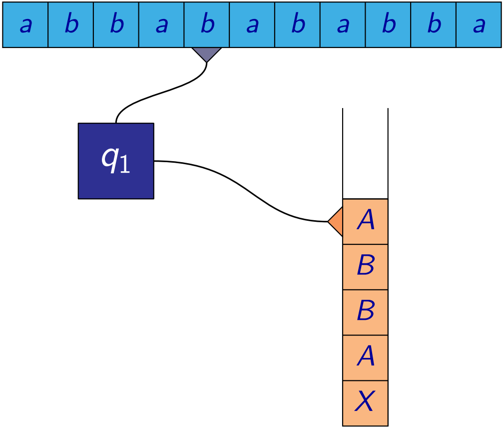
Pushdown automaton

- Word *abbababbba* belongs to the language $L = \{w \in \{a,b\}^* \mid w = w^R\}$



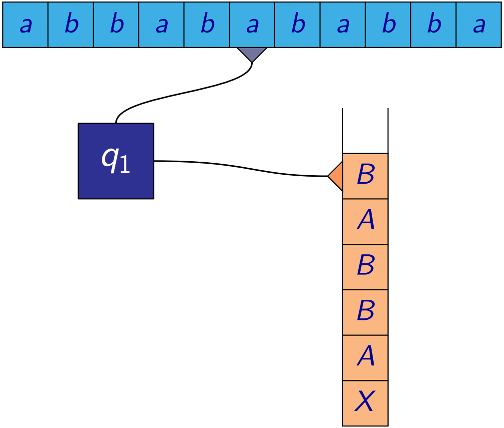
Pushdown automaton

- Word *abbabababba* belongs to the language $L = \{w \in \{a,b\}^* \mid w = w^R\}$



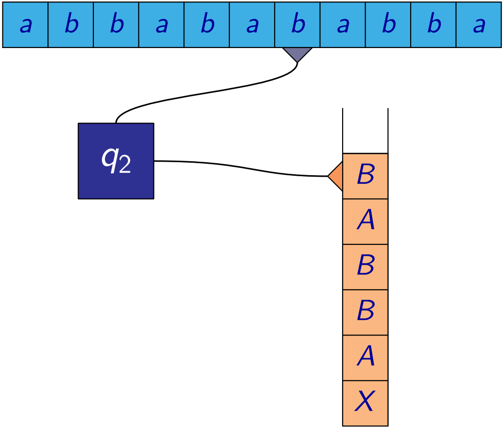
Pushdown automaton

- Word *abbababababba* belongs to the language
  $L = \{w \in \{a,b\}^* \mid w = w^R\}$



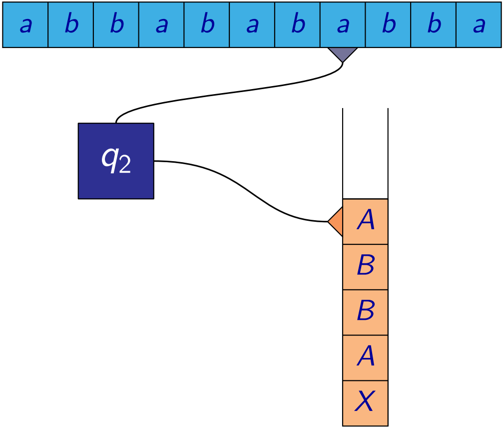
Pushdown automaton

- Word *abbababababba* belongs to the language
  $L = \{w \in \{a,b\}^* \mid w = w^R\}$



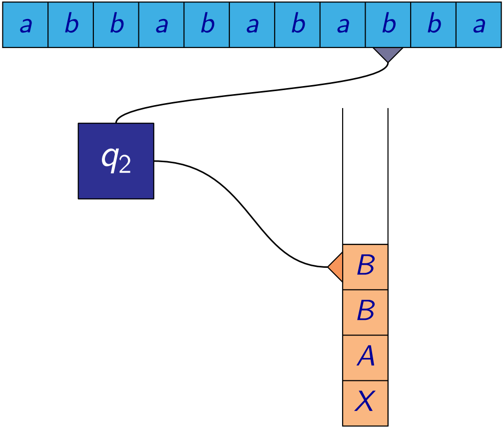
Pushdown automaton

- Word *abbabababba* belongs to the language
  $L = \{w \in \{a, b\}^* \mid w = w^R\}$



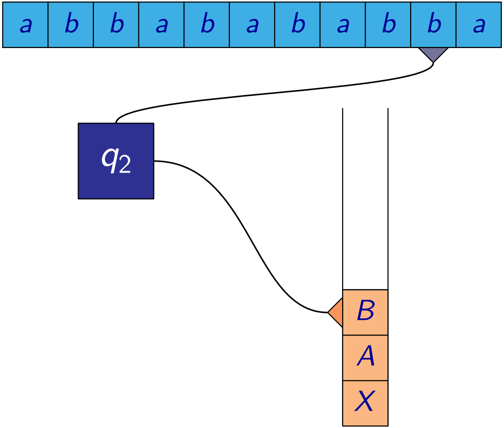
Pushdown automaton

- Word *abbababababba* belongs to the language
  $L = \{w \in \{a,b\}^* \mid w = w^R\}$



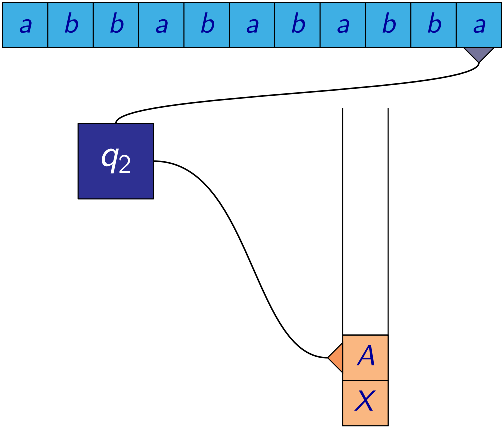
Pushdown automaton

- Word *abbababbabba* belongs to the language
  $L = \{w \in \{a, b\}^* \mid w = w^R\}$



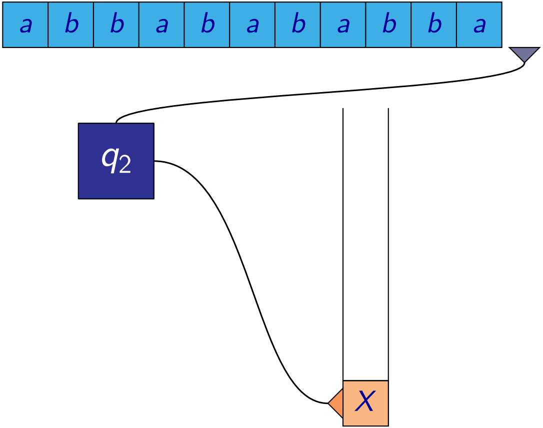
Pushdown automaton

- Word *abbababababba* belongs to the language
  $L = \{w \in \{a,b\}^* \mid w = w^R\}$



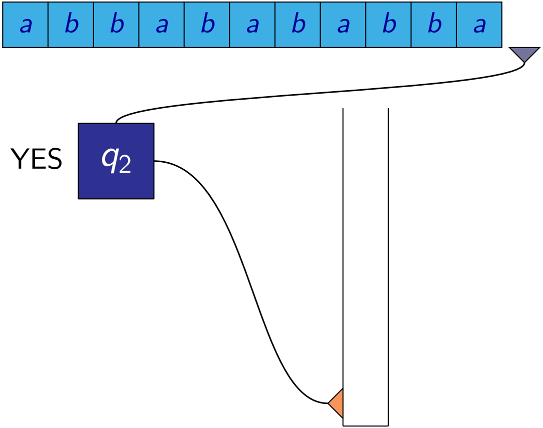
Pushdown automaton

- Word *abbabababba* belongs to the language
  $L = \{w \in \{a,b\}^* \mid w = w^R\}$



Pushdown automaton

- Word *abbababababba* belongs to the language
  $L = \{w \in \{a,b\}^* \mid w = w^R\}$



Pushdown automaton

Definition

A **pushdown automaton** (**PDA**) is a tuple $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, X_0)$ where

- Q is a finite non-empty set of states
- Σ is a finite non-empty set called an input alphabet
- Γ is a finite non-empty set called a stack alphabet
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}(Q \times \Gamma^*)$ is a (nondeterministic) transition function
- $q_0 \in Q$ is the initial state
- $X_0 \in \Gamma$ is the initial stack symbol

Pushdown automaton

Example: $L = \{ a^n b^n \mid n \geq 1 \}$

$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, O)$ where

- $Q = \{q_1, q_2\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{O, I\}$
- $\delta(q_1, a, O) = \{(q_1, I)\}$ $\quad\delta(q_1, b, O) = \emptyset$
  $\delta(q_1, a, I) = \{(q_1, II)\}$ $\quad\delta(q_1, b, I) = \{(q_2, \varepsilon)\}$
  $\delta(q_2, a, I) = \emptyset$ $\quad\delta(q_2, b, I) = \{(q_2, \varepsilon)\}$
  $\delta(q_2, a, O) = \emptyset$ $\quad\delta(q_2, b, O) = \emptyset$

Remark: We often omit those values of transition function δ that are $\emptyset$.

Pushdown automaton

To represent transition functions, we will use a notation where a transition function is viewed as a set of **rules**:

- For every $q, q' \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, $X \in \Gamma$, and $\alpha \in \Gamma^*$, where
$$(q', \alpha) \in \delta(q, a, X)$$
there is a corresponding rule
$$qX \xrightarrow{a} q'\alpha\,.$$

Example: If

$$\delta(q_5, b, C) = \{(q_3, ACC), (q_5, BB), (q_{13}, \varepsilon)\}$$

it can be represented as three rules:

$$q_5C \xrightarrow{b} q_3ACC \qquad q_5C \xrightarrow{b} q_5BB \qquad q_5C \xrightarrow{b} q_{13}$$

Pushdown automaton

Example: The automaton, recognizing the language $L = \{ a^n b^n \mid n \geq 1 \}$, that was described before:

$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, O)$ where

- $Q = \{q_1, q_2\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{O, I\}$
- $$q_1 O \xrightarrow{a} q_1 I$$
  $$q_1 I \xrightarrow{a} q_1 II$$
  $$q_1 I \xrightarrow{b} q_2$$
  $$q_2 I \xrightarrow{b} q_2$$

Pushdown automaton

Example: $L = \{ w \in \{a, b\}^* \mid w = w^R \}$

$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where

- $Q = \{q_1, q_2\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{X, A, B\}$
- $\delta(q_1, a, X) = \{(q_1, AX), (q_2, X)\}$ $\quad$ $\delta(q_1, b, X) = \{(q_1, BX), (q_2, X)\}$
  $\delta(q_1, a, A) = \{(q_1, AA), (q_2, A)\}$ $\quad$ $\delta(q_1, b, A) = \{(q_1, BA), (q_2, A)\}$
  $\delta(q_1, a, B) = \{(q_1, AB), (q_2, B)\}$ $\quad$ $\delta(q_1, b, B) = \{(q_1, BB), (q_2, B)\}$
  $\delta(q_1, \varepsilon, X) = \{(q_2, X)\}$ $\quad$ $\delta(q_2, \varepsilon, X) = \{(q_2, \varepsilon)\}$
  $\delta(q_1, \varepsilon, A) = \{(q_2, A)\}$ $\quad$ $\delta(q_2, \varepsilon, A) = \emptyset$
  $\delta(q_1, \varepsilon, B) = \{(q_2, B)\}$ $\quad$ $\delta(q_2, \varepsilon, B) = \emptyset$
  $\delta(q_2, a, A) = \{(q_2, \varepsilon)\}$ $\quad$ $\delta(q_2, b, A) = \emptyset$
  $\delta(q_2, a, B) = \emptyset$ $\quad$ $\delta(q_2, b, B) = \{(q_2, \varepsilon)\}$
  $\delta(q_2, a, X) = \emptyset$ $\quad$ $\delta(q_2, b, X) = \emptyset$

Pushdown automaton

Example: $L = \{ w \in \{a, b\}^* \mid w = w^R \}$

$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where

- $Q = \{q_1, q_2\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{X, A, B\}$
- $q_1 X \xrightarrow{a} q_1 AX$ $\quad q_1 X \xrightarrow{b} q_1 BX$ $\quad q_2 X \xrightarrow{\varepsilon} q_2$
  $q_1 A \xrightarrow{a} q_1 AA$ $\quad q_1 A \xrightarrow{b} q_1 BA$ $\quad q_2 A \xrightarrow{a} q_2$
  $q_1 B \xrightarrow{a} q_1 AB$ $\quad q_1 B \xrightarrow{b} q_1 BB$ $\quad q_2 B \xrightarrow{b} q_2$
  $q_1 X \xrightarrow{a} q_2 X$ $\quad q_1 X \xrightarrow{b} q_2 X$ $\quad q_1 X \xrightarrow{\varepsilon} q_2 X$
  $q_1 A \xrightarrow{a} q_2 A$ $\quad q_1 A \xrightarrow{b} q_2 A$ $\quad q_1 A \xrightarrow{\varepsilon} q_2 A$
  $q_1 B \xrightarrow{a} q_2 B$ $\quad q_1 B \xrightarrow{b} q_2 B$ $\quad q_1 B \xrightarrow{\varepsilon} q_2 B$

Computation of a Pushdown Automaton

Let $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, X_0)$ be a pushdown automaton.

Configurations of $\mathcal{M}$:

- A **configuration** of a PDA is a triple

$$(q, w, \alpha)$$

  where $q \in Q$, $w \in \Sigma^*$, and $\alpha \in \Gamma^*$.

- An **initial configuration** is a configuration (q_0, w, X_0), where $w \in \Sigma^*$.

Computation of a Pushdown Automaton

Steps performed by $\mathcal{M}$:

- Binary relation $\longrightarrow$ on configurations of $\mathcal{M}$ represents the possible steps of computation performed by PDA $\mathcal{M}$.

  That $\mathcal{M}$ can go from configuration (q, w, α) to configuration (q', w', α') is written as

  $$(q, w, \alpha) \longrightarrow (q', w', \alpha').$$

- The relation $\longrightarrow$ is defined as follows:

  $$(q, aw, X\beta) \longrightarrow (q', w, \alpha\beta) \quad \text{iff} \quad (q', \alpha) \in \delta(q, a, X)$$

  where $q, q' \in Q$, $a \in (\Sigma \cup \{\varepsilon\})$, $w \in \Sigma^*$, $X \in \Gamma$, and $\alpha, \beta \in \Gamma^*$.

Computation of a Pushdown Automaton

Computations of $\mathcal{M}$:

- We define binary relation $\longrightarrow^*$ on configurations of $\mathcal{M}$ as the reflexive and transitive closure of $\longrightarrow$, i.e.,

$$(q, w, \alpha) \longrightarrow^* (q', w', \alpha')$$

  if there is a sequence of configurations

$$(q_0, w_0, \alpha_0), (q_1, w_1, \alpha_1), \ldots, (q_n, w_n, \alpha_n)$$

  such that

  - $(q, w, \alpha) = (q_0, w_0, \alpha_0)$,
  - $(q', w', \alpha') = (q_n, w_n, \alpha_n)$, and
  - $(q_i, w_i, \alpha_i) \longrightarrow (q_{i+1}, w_{i+1}, \alpha_{i+1})$ for each $i = 0, 1, \ldots, n-1$, i.e.,

$$(q_0, w_0, \alpha_0) \longrightarrow (q_1, w_1, \alpha_1) \longrightarrow \quad \cdots \quad \longrightarrow (q_n, w_n, \alpha_n)$$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$$q_1X \xrightarrow{a} q_1AX \qquad q_1X \xrightarrow{b} q_1BX$$
$$q_1A \xrightarrow{a} q_1AA \qquad q_1A \xrightarrow{b} q_1BA$$
$$q_1B \xrightarrow{a} q_1AB \qquad q_1B \xrightarrow{b} q_1BB$$
$$q_1X \xrightarrow{a} q_2X \qquad q_1X \xrightarrow{b} q_2X$$
$$q_1A \xrightarrow{a} q_2A \qquad q_1A \xrightarrow{b} q_2A$$
$$q_1B \xrightarrow{a} q_2B \qquad q_1B \xrightarrow{b} q_2B$$
$$q_1X \xrightarrow{\varepsilon} q_2X$$
$$q_1A \xrightarrow{\varepsilon} q_2A$$
$$q_1B \xrightarrow{\varepsilon} q_2B$$
$$q_2X \xrightarrow{\varepsilon} q_2$$
$$q_2A \xrightarrow{a} q_2$$
$$q_2B \xrightarrow{b} q_2$$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$(q_1,\ abbababababba,\ X)$

$$q_1X \xrightarrow{a} q_1AX \qquad q_1X \xrightarrow{b} q_1BX$$
$$q_1A \xrightarrow{a} q_1AA \qquad q_1A \xrightarrow{b} q_1BA$$
$$q_1B \xrightarrow{a} q_1AB \qquad q_1B \xrightarrow{b} q_1BB$$
$$q_1X \xrightarrow{a} q_2X \qquad q_1X \xrightarrow{b} q_2X$$
$$q_1A \xrightarrow{a} q_2A \qquad q_1A \xrightarrow{b} q_2A$$
$$q_1B \xrightarrow{a} q_2B \qquad q_1B \xrightarrow{b} q_2B$$
$$q_1X \xrightarrow{\varepsilon} q_2X$$
$$q_1A \xrightarrow{\varepsilon} q_2A$$
$$q_1B \xrightarrow{\varepsilon} q_2B$$
$$q_2X \xrightarrow{\varepsilon} q_2$$
$$q_2A \xrightarrow{a} q_2$$
$$q_2B \xrightarrow{b} q_2$$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$(q_1,\ abbababababba,\ X)$
$\longrightarrow\ (q_1,\ bbababababba,\ AX)$

$q_1X \xrightarrow{a} q_1AX$ $\qquad$ $q_1X \xrightarrow{b} q_1BX$

$q_1A \xrightarrow{a} q_1AA$ $\qquad$ $q_1A \xrightarrow{b} q_1BA$

$q_1B \xrightarrow{a} q_1AB$ $\qquad$ $q_1B \xrightarrow{b} q_1BB$

$q_1X \xrightarrow{a} q_2X$ $\qquad$ $q_1X \xrightarrow{b} q_2X$

$q_1A \xrightarrow{a} q_2A$ $\qquad$ $q_1A \xrightarrow{b} q_2A$

$q_1B \xrightarrow{a} q_2B$ $\qquad$ $q_1B \xrightarrow{b} q_2B$

$q_1X \xrightarrow{\varepsilon} q_2X$

$q_1A \xrightarrow{\varepsilon} q_2A$

$q_1B \xrightarrow{\varepsilon} q_2B$

$q_2X \xrightarrow{\varepsilon} q_2$

$q_2A \xrightarrow{a} q_2$

$q_2B \xrightarrow{b} q_2$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$(q_1,\ abbababababba,\ X)$
$\longrightarrow\ (q_1,\ bbababababba,\ AX)$
$\longrightarrow\ (q_1,\ bababababba,\ BAX)$

$q_1X \xrightarrow{a} q_1AX$ $\quad$ $q_1X \xrightarrow{b} q_1BX$
$q_1A \xrightarrow{a} q_1AA$ $\quad$ $q_1A \xrightarrow{b} q_1BA$
$q_1B \xrightarrow{a} q_1AB$ $\quad$ $q_1B \xrightarrow{b} q_1BB$
$q_1X \xrightarrow{a} q_2X$ $\quad$ $q_1X \xrightarrow{b} q_2X$
$q_1A \xrightarrow{a} q_2A$ $\quad$ $q_1A \xrightarrow{b} q_2A$
$q_1B \xrightarrow{a} q_2B$ $\quad$ $q_1B \xrightarrow{b} q_2B$
$q_1X \xrightarrow{\varepsilon} q_2X$
$q_1A \xrightarrow{\varepsilon} q_2A$
$q_1B \xrightarrow{\varepsilon} q_2B$
$q_2X \xrightarrow{\varepsilon} q_2$
$q_2A \xrightarrow{a} q_2$
$q_2B \xrightarrow{b} q_2$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$(q_1,\ abbababababba,\ X)$
$\longrightarrow\ (q_1,\ bbababababba,\ AX)$
$\longrightarrow\ (q_1,\ babababba,\ BAX)$
$\longrightarrow\ (q_1,\ ababababba,\ BBAX)$

$q_1X \xrightarrow{a} q_1AX$	$q_1X \xrightarrow{b} q_1BX$
$q_1A \xrightarrow{a} q_1AA$	$q_1A \xrightarrow{b} q_1BA$
$q_1B \xrightarrow{a} q_1AB$	$q_1B \xrightarrow{b} q_1BB$
$q_1X \xrightarrow{a} q_2X$	$q_1X \xrightarrow{b} q_2X$
$q_1A \xrightarrow{a} q_2A$	$q_1A \xrightarrow{b} q_2A$
$q_1B \xrightarrow{a} q_2B$	$q_1B \xrightarrow{b} q_2B$
$q_1X \xrightarrow{\varepsilon} q_2X$	
$q_1A \xrightarrow{\varepsilon} q_2A$	
$q_1B \xrightarrow{\varepsilon} q_2B$	
$q_2X \xrightarrow{\varepsilon} q_2$	
$q_2A \xrightarrow{a} q_2$	
$q_2B \xrightarrow{b} q_2$	

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$(q_1,\ abbababababba,\ X)$
$\longrightarrow\ (q_1,\ bbababababba,\ AX)$
$\longrightarrow\ (q_1,\ bababababba,\ BAX)$
$\longrightarrow\ (q_1,\ abababba,\ BBAX)$
$\longrightarrow\ (q_1,\ bababba,\ ABBAX)$

$q_1X \xrightarrow{a} q_1AX$ $\qquad$ $q_1X \xrightarrow{b} q_1BX$

$q_1A \xrightarrow{a} q_1AA$ $\qquad$ $q_1A \xrightarrow{b} q_1BA$

$q_1B \xrightarrow{a} q_1AB$ $\qquad$ $q_1B \xrightarrow{b} q_1BB$

$q_1X \xrightarrow{a} q_2X$ $\qquad$ $q_1X \xrightarrow{b} q_2X$

$q_1A \xrightarrow{a} q_2A$ $\qquad$ $q_1A \xrightarrow{b} q_2A$

$q_1B \xrightarrow{a} q_2B$ $\qquad$ $q_1B \xrightarrow{b} q_2B$

$q_1X \xrightarrow{\varepsilon} q_2X$

$q_1A \xrightarrow{\varepsilon} q_2A$

$q_1B \xrightarrow{\varepsilon} q_2B$

$q_2X \xrightarrow{\varepsilon} q_2$

$q_2A \xrightarrow{a} q_2$

$q_2B \xrightarrow{b} q_2$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$(q_1,\ abbababababba,\ X)$
$\longrightarrow\ (q_1,\ bbababababba,\ AX)$
$\longrightarrow\ (q_1,\ babababba,\ BAX)$
$\longrightarrow\ (q_1,\ abababba,\ BBAX)$
$\longrightarrow\ (q_1,\ bababba,\ ABBAX)$
$\longrightarrow\ (q_1,\ ababba,\ BABBAX)$

$q_1X \xrightarrow{a} q_1AX$ $\quad$ $q_1X \xrightarrow{b} q_1BX$
$q_1A \xrightarrow{a} q_1AA$ $\quad$ $q_1A \xrightarrow{b} q_1BA$
$q_1B \xrightarrow{a} q_1AB$ $\quad$ $q_1B \xrightarrow{b} q_1BB$
$q_1X \xrightarrow{a} q_2X$ $\quad$ $q_1X \xrightarrow{b} q_2X$
$q_1A \xrightarrow{a} q_2A$ $\quad$ $q_1A \xrightarrow{b} q_2A$
$q_1B \xrightarrow{a} q_2B$ $\quad$ $q_1B \xrightarrow{b} q_2B$
$q_1X \xrightarrow{\varepsilon} q_2X$
$q_1A \xrightarrow{\varepsilon} q_2A$
$q_1B \xrightarrow{\varepsilon} q_2B$
$q_2X \xrightarrow{\varepsilon} q_2$
$q_2A \xrightarrow{a} q_2$
$q_2B \xrightarrow{b} q_2$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$(q_1,\ abbababababba,\ X)$
$\longrightarrow\ (q_1,\ bbababababba,\ AX)$
$\longrightarrow\ (q_1,\ babababba,\ BAX)$
$\longrightarrow\ (q_1,\ abababba,\ BBAX)$
$\longrightarrow\ (q_1,\ bababba,\ ABBAX)$
$\longrightarrow\ (q_1,\ ababba,\ BABBAX)$
$\longrightarrow\ (q_2,\ babba,\ BABBAX)$

$q_1X \xrightarrow{a} q_1AX$
$q_1A \xrightarrow{a} q_1AA$
$q_1B \xrightarrow{a} q_1AB$
$q_1X \xrightarrow{a} q_2X$
$q_1A \xrightarrow{a} q_2A$
$q_1B \xrightarrow{a} q_2B$
$q_1X \xrightarrow{\varepsilon} q_2X$
$q_1A \xrightarrow{\varepsilon} q_2A$
$q_1B \xrightarrow{\varepsilon} q_2B$
$q_2X \xrightarrow{\varepsilon} q_2$
$q_2A \xrightarrow{a} q_2$
$q_2B \xrightarrow{b} q_2$

$q_1X \xrightarrow{b} q_1BX$
$q_1A \xrightarrow{b} q_1BA$
$q_1B \xrightarrow{b} q_1BB$
$q_1X \xrightarrow{b} q_2X$
$q_1A \xrightarrow{b} q_2A$
$q_1B \xrightarrow{b} q_2B$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$(q_1,\ abbababababba,\ X)$
$\longrightarrow\ (q_1,\ bbababababba,\ AX)$
$\longrightarrow\ (q_1,\ bababababba,\ BAX)$
$\longrightarrow\ (q_1,\ ababababba,\ BBAX)$
$\longrightarrow\ (q_1,\ babababba,\ ABBAX)$
$\longrightarrow\ (q_1,\ abababba,\ BABBAX)$
$\longrightarrow\ (q_2,\ babba,\ BABBAX)$
$\longrightarrow\ (q_2,\ abba,\ ABBAX)$

$q_1 X \xrightarrow{a} q_1 AX$ $\quad$ $q_1 X \xrightarrow{b} q_1 BX$

$q_1 A \xrightarrow{a} q_1 AA$ $\quad$ $q_1 A \xrightarrow{b} q_1 BA$

$q_1 B \xrightarrow{a} q_1 AB$ $\quad$ $q_1 B \xrightarrow{b} q_1 BB$

$q_1 X \xrightarrow{a} q_2 X$ $\quad$ $q_1 X \xrightarrow{b} q_2 X$

$q_1 A \xrightarrow{a} q_2 A$ $\quad$ $q_1 A \xrightarrow{b} q_2 A$

$q_1 B \xrightarrow{a} q_2 B$ $\quad$ $q_1 B \xrightarrow{b} q_2 B$

$q_1 X \xrightarrow{\varepsilon} q_2 X$

$q_1 A \xrightarrow{\varepsilon} q_2 A$

$q_1 B \xrightarrow{\varepsilon} q_2 B$

$q_2 X \xrightarrow{\varepsilon} q_2$

$q_2 A \xrightarrow{a} q_2$

$q_2 B \xrightarrow{b} q_2$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$(q_1,\ abbababababba,\ X)$
$\longrightarrow\ (q_1,\ bbababababba,\ AX)$
$\longrightarrow\ (q_1,\ bababababba,\ BAX)$
$\longrightarrow\ (q_1,\ abababba,\ BBAX)$
$\longrightarrow\ (q_1,\ bababba,\ ABBAX)$
$\longrightarrow\ (q_1,\ ababba,\ BABBAX)$
$\longrightarrow\ (q_2,\ babba,\ BABBAX)$
$\longrightarrow\ (q_2,\ abba,\ ABBAX)$
$\longrightarrow\ (q_2,\ bba,\ BBAX)$

$q_1X \xrightarrow{a} q_1AX$	$q_1X \xrightarrow{b} q_1BX$
$q_1A \xrightarrow{a} q_1AA$	$q_1A \xrightarrow{b} q_1BA$
$q_1B \xrightarrow{a} q_1AB$	$q_1B \xrightarrow{b} q_1BB$
$q_1X \xrightarrow{a} q_2X$	$q_1X \xrightarrow{b} q_2X$
$q_1A \xrightarrow{a} q_2A$	$q_1A \xrightarrow{b} q_2A$
$q_1B \xrightarrow{a} q_2B$	$q_1B \xrightarrow{b} q_2B$
$q_1X \xrightarrow{\varepsilon} q_2X$	
$q_1A \xrightarrow{\varepsilon} q_2A$	
$q_1B \xrightarrow{\varepsilon} q_2B$	
$q_2X \xrightarrow{\varepsilon} q_2$	
$q_2A \xrightarrow{a} q_2$	
$q_2B \xrightarrow{b} q_2$	

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$$
\begin{array}{l}
(q_1,\ abbababababba,\ X) \\
\longrightarrow\ (q_1,\ bbababababba,\ AX) \\
\longrightarrow\ (q_1,\ babababba,\ BAX) \\
\longrightarrow\ (q_1,\ ababababba,\ BBAX) \\
\longrightarrow\ (q_1,\ bababba,\ ABBAX) \\
\longrightarrow\ (q_1,\ ababba,\ BABBAX) \\
\longrightarrow\ (q_2,\ babba,\ BABBAX) \\
\longrightarrow\ (q_2,\ abba,\ ABBAX) \\
\longrightarrow\ (q_2,\ bba,\ BBAX) \\
\longrightarrow\ (q_2,\ ba,\ BAX)
\end{array}
$$

$$
\begin{array}{ll}
q_1X \xrightarrow{a} q_1AX & q_1X \xrightarrow{b} q_1BX \\
q_1A \xrightarrow{a} q_1AA & q_1A \xrightarrow{b} q_1BA \\
q_1B \xrightarrow{a} q_1AB & q_1B \xrightarrow{b} q_1BB \\
q_1X \xrightarrow{a} q_2X & q_1X \xrightarrow{b} q_2X \\
q_1A \xrightarrow{a} q_2A & q_1A \xrightarrow{b} q_2A \\
q_1B \xrightarrow{a} q_2B & q_1B \xrightarrow{b} q_2B \\
q_1X \xrightarrow{\varepsilon} q_2X & \\
q_1A \xrightarrow{\varepsilon} q_2A & \\
q_1B \xrightarrow{\varepsilon} q_2B & \\
q_2X \xrightarrow{\varepsilon} q_2 & \\
q_2A \xrightarrow{a} q_2 & \\
q_2B \xrightarrow{b} q_2 &
\end{array}
$$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$$
\begin{array}{l}
(q_1,\ abbababab ba,\ X) \\
\longrightarrow\ (q_1,\ bbabababba,\ AX) \\
\longrightarrow\ (q_1,\ babababba,\ BAX) \\
\longrightarrow\ (q_1,\ ababab ba,\ BBAX) \\
\longrightarrow\ (q_1,\ bababba,\ ABBAX) \\
\longrightarrow\ (q_1,\ ababba,\ BABBAX) \\
\longrightarrow\ (q_2,\ babba,\ BABBAX) \\
\longrightarrow\ (q_2,\ abba,\ ABBAX) \\
\longrightarrow\ (q_2,\ bba,\ BBAX) \\
\longrightarrow\ (q_2,\ ba,\ BAX) \\
\longrightarrow\ (q_2,\ a,\ AX)
\end{array}
$$

$$
\begin{array}{ll}
q_1X \xrightarrow{a} q_1AX & q_1X \xrightarrow{b} q_1BX \\
q_1A \xrightarrow{a} q_1AA & q_1A \xrightarrow{b} q_1BA \\
q_1B \xrightarrow{a} q_1AB & q_1B \xrightarrow{b} q_1BB \\
q_1X \xrightarrow{a} q_2X & q_1X \xrightarrow{b} q_2X \\
q_1A \xrightarrow{a} q_2A & q_1A \xrightarrow{b} q_2A \\
q_1B \xrightarrow{a} q_2B & q_1B \xrightarrow{b} q_2B \\
q_1X \xrightarrow{\varepsilon} q_2X & \\
q_1A \xrightarrow{\varepsilon} q_2A & \\
q_1B \xrightarrow{\varepsilon} q_2B & \\
q_2X \xrightarrow{\varepsilon} q_2 & \\
q_2A \xrightarrow{a} q_2 & \\
q_2B \xrightarrow{b} q_2 &
\end{array}
$$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$(q_1,\ abbabababba,\ X)$
$\longrightarrow\ (q_1,\ bbabababba,\ AX)$
$\longrightarrow\ (q_1,\ babababba,\ BAX)$
$\longrightarrow\ (q_1,\ ababbabba,\ BBAX)$
$\longrightarrow\ (q_1,\ bababba,\ ABBAX)$
$\longrightarrow\ (q_1,\ ababba,\ BABBAX)$
$\longrightarrow\ (q_2,\ babba,\ BABBAX)$
$\longrightarrow\ (q_2,\ abba,\ ABBAX)$
$\longrightarrow\ (q_2,\ bba,\ BBAX)$
$\longrightarrow\ (q_2,\ ba,\ BAX)$
$\longrightarrow\ (q_2,\ a,\ AX)$
$\longrightarrow\ (q_2,\ \varepsilon,\ X)$

$q_1X \xrightarrow{a} q_1AX$
$q_1A \xrightarrow{a} q_1AA$
$q_1B \xrightarrow{a} q_1AB$
$q_1X \xrightarrow{a} q_2X$
$q_1A \xrightarrow{a} q_2A$
$q_1B \xrightarrow{a} q_2B$
$q_1X \xrightarrow{\varepsilon} q_2X$
$q_1A \xrightarrow{\varepsilon} q_2A$
$q_1B \xrightarrow{\varepsilon} q_2B$
$q_2X \xrightarrow{\varepsilon} q_2$
$q_2A \xrightarrow{a} q_2$
$q_2B \xrightarrow{b} q_2$

$q_1X \xrightarrow{b} q_1BX$
$q_1A \xrightarrow{b} q_1BA$
$q_1B \xrightarrow{b} q_1BB$
$q_1X \xrightarrow{b} q_2X$
$q_1A \xrightarrow{b} q_2A$
$q_1B \xrightarrow{b} q_2B$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$(q_1,\ abbababababba,\ X)$
$\longrightarrow\ (q_1,\ bbababababba,\ AX)$
$\longrightarrow\ (q_1,\ babababba,\ BAX)$
$\longrightarrow\ (q_1,\ abababba,\ BBAX)$
$\longrightarrow\ (q_1,\ bababba,\ ABBAX)$
$\longrightarrow\ (q_1,\ ababba,\ BABBAX)$
$\longrightarrow\ (q_2,\ babba,\ BABBAX)$
$\longrightarrow\ (q_2,\ abba,\ ABBAX)$
$\longrightarrow\ (q_2,\ bba,\ BBAX)$
$\longrightarrow\ (q_2,\ ba,\ BAX)$
$\longrightarrow\ (q_2,\ a,\ AX)$
$\longrightarrow\ (q_2,\ \varepsilon,\ X)$
$\longrightarrow\ (q_2,\ \varepsilon,\ \varepsilon)$

$q_1X \xrightarrow{a} q_1AX$ $\qquad$ $q_1X \xrightarrow{b} q_1BX$
$q_1A \xrightarrow{a} q_1AA$ $\qquad$ $q_1A \xrightarrow{b} q_1BA$
$q_1B \xrightarrow{a} q_1AB$ $\qquad$ $q_1B \xrightarrow{b} q_1BB$
$q_1X \xrightarrow{a} q_2X$ $\qquad$ $q_1X \xrightarrow{b} q_2X$
$q_1A \xrightarrow{a} q_2A$ $\qquad$ $q_1A \xrightarrow{b} q_2A$
$q_1B \xrightarrow{a} q_2B$ $\qquad$ $q_1B \xrightarrow{b} q_2B$
$q_1X \xrightarrow{\varepsilon} q_2X$
$q_1A \xrightarrow{\varepsilon} q_2A$
$q_1B \xrightarrow{\varepsilon} q_2B$
$q_2X \xrightarrow{\varepsilon} q_2$
$q_2A \xrightarrow{a} q_2$
$q_2B \xrightarrow{b} q_2$

Computation of a Pushdown Automaton

In the previous definition, the set of configurations was defined as

$$Conf = Q \times \Sigma^* \times \Gamma^*$$

and relation $\longrightarrow$ was a subset of the set $Conf \times Conf$.

Computation of a Pushdown Automaton

Alternatively, we could define configurations in such a way that they do not contain an input word:

$$Conf = Q \times \Gamma^*$$

The relation $\longrightarrow$ is then defined as a subset of the set $Conf \times (\Sigma \cup \{\varepsilon\}) \times Conf$, where the notation

$$q\alpha \xrightarrow{a} q'\alpha'$$

that after reading symbol a (or reading nothing when $a = \varepsilon$), the given pushdown automaton can go from configuration (q, α) to configuration (q', α'), i.e.,

$$qX\beta \xrightarrow{a} q'\gamma\beta \quad \text{iff} \quad (q', \gamma) \in \delta(q, a, X)$$

where $q, q' \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, $X \in \Gamma$, and $\beta, \gamma \in \Gamma^*$.

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$$q_1X \xrightarrow{a} q_1AX \qquad q_1X \xrightarrow{b} q_1BX$$
$$q_1A \xrightarrow{a} q_1AA \qquad q_1A \xrightarrow{b} q_1BA$$
$$q_1B \xrightarrow{a} q_1AB \qquad q_1B \xrightarrow{b} q_1BB$$
$$q_1X \xrightarrow{a} q_2X \qquad q_1X \xrightarrow{b} q_2X$$
$$q_1A \xrightarrow{a} q_2A \qquad q_1A \xrightarrow{b} q_2A$$
$$q_1B \xrightarrow{a} q_2B \qquad q_1B \xrightarrow{b} q_2B$$
$$q_1X \xrightarrow{\varepsilon} q_2X$$
$$q_1A \xrightarrow{\varepsilon} q_2A$$
$$q_1B \xrightarrow{\varepsilon} q_2B$$
$$q_2X \xrightarrow{\varepsilon} q_2$$
$$q_2A \xrightarrow{a} q_2$$
$$q_2B \xrightarrow{b} q_2$$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$q_1 X$

$$q_1 X \xrightarrow{a} q_1 AX \qquad q_1 X \xrightarrow{b} q_1 BX$$
$$q_1 A \xrightarrow{a} q_1 AA \qquad q_1 A \xrightarrow{b} q_1 BA$$
$$q_1 B \xrightarrow{a} q_1 AB \qquad q_1 B \xrightarrow{b} q_1 BB$$
$$q_1 X \xrightarrow{a} q_2 X \qquad q_1 X \xrightarrow{b} q_2 X$$
$$q_1 A \xrightarrow{a} q_2 A \qquad q_1 A \xrightarrow{b} q_2 A$$
$$q_1 B \xrightarrow{a} q_2 B \qquad q_1 B \xrightarrow{b} q_2 B$$
$$q_1 X \xrightarrow{\varepsilon} q_2 X$$
$$q_1 A \xrightarrow{\varepsilon} q_2 A$$
$$q_1 B \xrightarrow{\varepsilon} q_2 B$$
$$q_2 X \xrightarrow{\varepsilon} q_2$$
$$q_2 A \xrightarrow{a} q_2$$
$$q_2 B \xrightarrow{b} q_2$$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$,
$\Gamma = \{X, A, B\}$

$$q_1X \xrightarrow{a} q_1AX$$

$$\begin{array}{ll}
q_1X \xrightarrow{a} q_1AX & q_1X \xrightarrow{b} q_1BX \\
q_1A \xrightarrow{a} q_1AA & q_1A \xrightarrow{b} q_1BA \\
q_1B \xrightarrow{a} q_1AB & q_1B \xrightarrow{b} q_1BB \\
q_1X \xrightarrow{a} q_2X & q_1X \xrightarrow{b} q_2X \\
q_1A \xrightarrow{a} q_2A & q_1A \xrightarrow{b} q_2A \\
q_1B \xrightarrow{a} q_2B & q_1B \xrightarrow{b} q_2B \\
q_1X \xrightarrow{\varepsilon} q_2X & \\
q_1A \xrightarrow{\varepsilon} q_2A & \\
q_1B \xrightarrow{\varepsilon} q_2B & \\
q_2X \xrightarrow{\varepsilon} q_2 & \\
q_2A \xrightarrow{a} q_2 & \\
q_2B \xrightarrow{b} q_2 &
\end{array}$$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$$q_1X \xrightarrow{a} q_1AX$$
$$\xrightarrow{b} q_1BAX$$

$$q_1X \xrightarrow{a} q_1AX \qquad q_1X \xrightarrow{b} q_1BX$$
$$q_1A \xrightarrow{a} q_1AA \qquad q_1A \xrightarrow{b} q_1BA$$
$$q_1B \xrightarrow{a} q_1AB \qquad q_1B \xrightarrow{b} q_1BB$$
$$q_1X \xrightarrow{a} q_2X \qquad q_1X \xrightarrow{b} q_2X$$
$$q_1A \xrightarrow{a} q_2A \qquad q_1A \xrightarrow{b} q_2A$$
$$q_1B \xrightarrow{a} q_2B \qquad q_1B \xrightarrow{b} q_2B$$
$$q_1X \xrightarrow{\varepsilon} q_2X$$
$$q_1A \xrightarrow{\varepsilon} q_2A$$
$$q_1B \xrightarrow{\varepsilon} q_2B$$
$$q_2X \xrightarrow{\varepsilon} q_2$$
$$q_2A \xrightarrow{a} q_2$$
$$q_2B \xrightarrow{b} q_2$$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$$q_1X \xrightarrow{a} q_1AX \xrightarrow{b} q_1BAX \xrightarrow{b} q_1BBAX$$

$$q_1X \xrightarrow{a} q_1AX \qquad q_1X \xrightarrow{b} q_1BX$$
$$q_1A \xrightarrow{a} q_1AA \qquad q_1A \xrightarrow{b} q_1BA$$
$$q_1B \xrightarrow{a} q_1AB \qquad q_1B \xrightarrow{b} q_1BB$$
$$q_1X \xrightarrow{a} q_2X \qquad q_1X \xrightarrow{b} q_2X$$
$$q_1A \xrightarrow{a} q_2A \qquad q_1A \xrightarrow{b} q_2A$$
$$q_1B \xrightarrow{a} q_2B \qquad q_1B \xrightarrow{b} q_2B$$
$$q_1X \xrightarrow{\varepsilon} q_2X$$
$$q_1A \xrightarrow{\varepsilon} q_2A$$
$$q_1B \xrightarrow{\varepsilon} q_2B$$
$$q_2X \xrightarrow{\varepsilon} q_2$$
$$q_2A \xrightarrow{a} q_2$$
$$q_2B \xrightarrow{b} q_2$$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$$q_1 X \xrightarrow{a} q_1 AX \xrightarrow{b} q_1 BAX \xrightarrow{b} q_1 BBAX \xrightarrow{a} q_1 ABBAX$$

$q_1 X \xrightarrow{a} q_1 AX$

$q_1 A \xrightarrow{a} q_1 AA$

$q_1 B \xrightarrow{a} q_1 AB$

$q_1 X \xrightarrow{a} q_2 X$

$q_1 A \xrightarrow{a} q_2 A$

$q_1 B \xrightarrow{a} q_2 B$

$q_1 X \xrightarrow{\varepsilon} q_2 X$

$q_1 A \xrightarrow{\varepsilon} q_2 A$

$q_1 B \xrightarrow{\varepsilon} q_2 B$

$q_2 X \xrightarrow{\varepsilon} q_2$

$q_2 A \xrightarrow{a} q_2$

$q_2 B \xrightarrow{b} q_2$

$q_1 X \xrightarrow{b} q_1 BX$

$q_1 A \xrightarrow{b} q_1 BA$

$q_1 B \xrightarrow{b} q_1 BB$

$q_1 X \xrightarrow{b} q_2 X$

$q_1 A \xrightarrow{b} q_2 A$

$q_1 B \xrightarrow{b} q_2 B$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$$
\begin{aligned}
q_1 X &\xrightarrow{a} q_1 AX \\
&\xrightarrow{b} q_1 BAX \\
&\xrightarrow{b} q_1 BBAX \\
&\xrightarrow{a} q_1 ABBAX \\
&\xrightarrow{b} q_1 BABBAX
\end{aligned}
$$

$$
\begin{array}{ll}
q_1 X \xrightarrow{a} q_1 AX & q_1 X \xrightarrow{b} q_1 BX \\
q_1 A \xrightarrow{a} q_1 AA & q_1 A \xrightarrow{b} q_1 BA \\
q_1 B \xrightarrow{a} q_1 AB & q_1 B \xrightarrow{b} q_1 BB \\
q_1 X \xrightarrow{a} q_2 X & q_1 X \xrightarrow{b} q_2 X \\
q_1 A \xrightarrow{a} q_2 A & q_1 A \xrightarrow{b} q_2 A \\
q_1 B \xrightarrow{a} q_2 B & q_1 B \xrightarrow{b} q_2 B \\
q_1 X \xrightarrow{\varepsilon} q_2 X & \\
q_1 A \xrightarrow{\varepsilon} q_2 A & \\
q_1 B \xrightarrow{\varepsilon} q_2 B & \\
q_2 X \xrightarrow{\varepsilon} q_2 & \\
q_2 A \xrightarrow{a} q_2 & \\
q_2 B \xrightarrow{b} q_2 &
\end{array}
$$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$$
\begin{aligned}
q_1 X &\xrightarrow{a} q_1 AX \\
&\xrightarrow{b} q_1 BAX \\
&\xrightarrow{b} q_1 BBAX \\
&\xrightarrow{a} q_1 ABBAX \\
&\xrightarrow{b} q_1 BABBAX \\
&\xrightarrow{a} q_2 BABBAX
\end{aligned}
$$

$$
\begin{array}{ll}
q_1 X \xrightarrow{a} q_1 AX & q_1 X \xrightarrow{b} q_1 BX \\
q_1 A \xrightarrow{a} q_1 AA & q_1 A \xrightarrow{b} q_1 BA \\
q_1 B \xrightarrow{a} q_1 AB & q_1 B \xrightarrow{b} q_1 BB \\
q_1 X \xrightarrow{a} q_2 X & q_1 X \xrightarrow{b} q_2 X \\
q_1 A \xrightarrow{a} q_2 A & q_1 A \xrightarrow{b} q_2 A \\
q_1 B \xrightarrow{a} q_2 B & q_1 B \xrightarrow{b} q_2 B \\
q_1 X \xrightarrow{\varepsilon} q_2 X & \\
q_1 A \xrightarrow{\varepsilon} q_2 A & \\
q_1 B \xrightarrow{\varepsilon} q_2 B & \\
q_2 X \xrightarrow{\varepsilon} q_2 & \\
q_2 A \xrightarrow{a} q_2 & \\
q_2 B \xrightarrow{b} q_2 &
\end{array}
$$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$$
\begin{aligned}
q_1X &\xrightarrow{a} q_1AX \\
&\xrightarrow{b} q_1BAX \\
&\xrightarrow{b} q_1BBAX \\
&\xrightarrow{a} q_1ABBAX \\
&\xrightarrow{b} q_1BABBAX \\
&\xrightarrow{a} q_2BABBAX \\
&\xrightarrow{b} q_2ABBAX
\end{aligned}
$$

$q_1X \xrightarrow{a} q_1AX$	$q_1X \xrightarrow{b} q_1BX$
$q_1A \xrightarrow{a} q_1AA$	$q_1A \xrightarrow{b} q_1BA$
$q_1B \xrightarrow{a} q_1AB$	$q_1B \xrightarrow{b} q_1BB$
$q_1X \xrightarrow{a} q_2X$	$q_1X \xrightarrow{b} q_2X$
$q_1A \xrightarrow{a} q_2A$	$q_1A \xrightarrow{b} q_2A$
$q_1B \xrightarrow{a} q_2B$	$q_1B \xrightarrow{b} q_2B$
$q_1X \xrightarrow{\varepsilon} q_2X$	
$q_1A \xrightarrow{\varepsilon} q_2A$	
$q_1B \xrightarrow{\varepsilon} q_2B$	
$q_2X \xrightarrow{\varepsilon} q_2$	
$q_2A \xrightarrow{a} q_2$	
$q_2B \xrightarrow{b} q_2$	

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$$
\begin{aligned}
q_1 X &\xrightarrow{a} q_1 AX \\
&\xrightarrow{b} q_1 BAX \\
&\xrightarrow{b} q_1 BBAX \\
&\xrightarrow{a} q_1 ABBAX \\
&\xrightarrow{b} q_1 BABBAX \\
&\xrightarrow{a} q_2 BABBAX \\
&\xrightarrow{b} q_2 ABBAX \\
&\xrightarrow{a} q_2 BBAX
\end{aligned}
$$

$$
\begin{array}{ll}
q_1 X \xrightarrow{a} q_1 AX & q_1 X \xrightarrow{b} q_1 BX \\
q_1 A \xrightarrow{a} q_1 AA & q_1 A \xrightarrow{b} q_1 BA \\
q_1 B \xrightarrow{a} q_1 AB & q_1 B \xrightarrow{b} q_1 BB \\
q_1 X \xrightarrow{a} q_2 X & q_1 X \xrightarrow{b} q_2 X \\
q_1 A \xrightarrow{a} q_2 A & q_1 A \xrightarrow{b} q_2 A \\
q_1 B \xrightarrow{a} q_2 B & q_1 B \xrightarrow{b} q_2 B \\
q_1 X \xrightarrow{\varepsilon} q_2 X & \\
q_1 A \xrightarrow{\varepsilon} q_2 A & \\
q_1 B \xrightarrow{\varepsilon} q_2 B & \\
q_2 X \xrightarrow{\varepsilon} q_2 & \\
q_2 A \xrightarrow{a} q_2 & \\
q_2 B \xrightarrow{b} q_2 &
\end{array}
$$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$$
\begin{aligned}
q_1 X &\xrightarrow{a} q_1 AX \\
&\xrightarrow{b} q_1 BAX \\
&\xrightarrow{b} q_1 BBAX \\
&\xrightarrow{a} q_1 ABBAX \\
&\xrightarrow{b} q_1 BABBAX \\
&\xrightarrow{a} q_2 BABBAX \\
&\xrightarrow{b} q_2 ABBAX \\
&\xrightarrow{a} q_2 BBAX \\
&\xrightarrow{b} q_2 BAX
\end{aligned}
$$

$$
\begin{array}{ll}
q_1 X \xrightarrow{a} q_1 AX & q_1 X \xrightarrow{b} q_1 BX \\
q_1 A \xrightarrow{a} q_1 AA & q_1 A \xrightarrow{b} q_1 BA \\
q_1 B \xrightarrow{a} q_1 AB & q_1 B \xrightarrow{b} q_1 BB \\
q_1 X \xrightarrow{a} q_2 X & q_1 X \xrightarrow{b} q_2 X \\
q_1 A \xrightarrow{a} q_2 A & q_1 A \xrightarrow{b} q_2 A \\
q_1 B \xrightarrow{a} q_2 B & q_1 B \xrightarrow{b} q_2 B \\
q_1 X \xrightarrow{\varepsilon} q_2 X & \\
q_1 A \xrightarrow{\varepsilon} q_2 A & \\
q_1 B \xrightarrow{\varepsilon} q_2 B & \\
q_2 X \xrightarrow{\varepsilon} q_2 & \\
q_2 A \xrightarrow{a} q_2 & \\
q_2 B \xrightarrow{b} q_2 &
\end{array}
$$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$$
\begin{aligned}
q_1 X &\xrightarrow{a} q_1 AX \\
&\xrightarrow{b} q_1 BAX \\
&\xrightarrow{b} q_1 BBAX \\
&\xrightarrow{a} q_1 ABBAX \\
&\xrightarrow{b} q_1 BABBAX \\
&\xrightarrow{a} q_2 BABBAX \\
&\xrightarrow{b} q_2 ABBAX \\
&\xrightarrow{a} q_2 BBAX \\
&\xrightarrow{b} q_2 BAX \\
&\xrightarrow{b} q_2 AX
\end{aligned}
$$

$$
\begin{array}{ll}
q_1 X \xrightarrow{a} q_1 AX & q_1 X \xrightarrow{b} q_1 BX \\
q_1 A \xrightarrow{a} q_1 AA & q_1 A \xrightarrow{b} q_1 BA \\
q_1 B \xrightarrow{a} q_1 AB & q_1 B \xrightarrow{b} q_1 BB \\
q_1 X \xrightarrow{a} q_2 X & q_1 X \xrightarrow{b} q_2 X \\
q_1 A \xrightarrow{a} q_2 A & q_1 A \xrightarrow{b} q_2 A \\
q_1 B \xrightarrow{a} q_2 B & q_1 B \xrightarrow{b} q_2 B \\
q_1 X \xrightarrow{\varepsilon} q_2 X & \\
q_1 A \xrightarrow{\varepsilon} q_2 A & \\
q_1 B \xrightarrow{\varepsilon} q_2 B & \\
q_2 X \xrightarrow{\varepsilon} q_2 & \\
q_2 A \xrightarrow{a} q_2 & \\
q_2 B \xrightarrow{b} q_2 &
\end{array}
$$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$$
\begin{aligned}
q_1X &\xrightarrow{a} q_1AX \\
&\xrightarrow{b} q_1BAX \\
&\xrightarrow{b} q_1BBAX \\
&\xrightarrow{a} q_1ABBAX \\
&\xrightarrow{b} q_1BABBAX \\
&\xrightarrow{a} q_2BABBAX \\
&\xrightarrow{b} q_2ABBAX \\
&\xrightarrow{a} q_2BBAX \\
&\xrightarrow{b} q_2BAX \\
&\xrightarrow{b} q_2AX \\
&\xrightarrow{a} q_2X
\end{aligned}
$$

$$
\begin{array}{ll}
q_1X \xrightarrow{a} q_1AX & q_1X \xrightarrow{b} q_1BX \\
q_1A \xrightarrow{a} q_1AA & q_1A \xrightarrow{b} q_1BA \\
q_1B \xrightarrow{a} q_1AB & q_1B \xrightarrow{b} q_1BB \\
q_1X \xrightarrow{a} q_2X & q_1X \xrightarrow{b} q_2X \\
q_1A \xrightarrow{a} q_2A & q_1A \xrightarrow{b} q_2A \\
q_1B \xrightarrow{a} q_2B & q_1B \xrightarrow{b} q_2B \\
q_1X \xrightarrow{\varepsilon} q_2X & \\
q_1A \xrightarrow{\varepsilon} q_2A & \\
q_1B \xrightarrow{\varepsilon} q_2B & \\
q_2X \xrightarrow{\varepsilon} q_2 & \\
q_2A \xrightarrow{a} q_2 & \\
q_2B \xrightarrow{b} q_2 &
\end{array}
$$

Computation of a Pushdown Automaton

Example: $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_1, X)$ where $Q = \{q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{X, A, B\}$

$$
\begin{array}{lll}
q_1 X & \xrightarrow{a} & q_1 AX \\
& \xrightarrow{b} & q_1 BAX \\
& \xrightarrow{b} & q_1 BBAX \\
& \xrightarrow{a} & q_1 ABBAX \\
& \xrightarrow{b} & q_1 BABBAX \\
& \xrightarrow{a} & q_2 BABBAX \\
& \xrightarrow{b} & q_2 ABBAX \\
& \xrightarrow{a} & q_2 BBAX \\
& \xrightarrow{b} & q_2 BAX \\
& \xrightarrow{b} & q_2 AX \\
& \xrightarrow{a} & q_2 X \\
& \xrightarrow{\varepsilon} & q_2
\end{array}
$$

$$
\begin{array}{ll}
q_1 X \xrightarrow{a} q_1 AX & q_1 X \xrightarrow{b} q_1 BX \\
q_1 A \xrightarrow{a} q_1 AA & q_1 A \xrightarrow{b} q_1 BA \\
q_1 B \xrightarrow{a} q_1 AB & q_1 B \xrightarrow{b} q_1 BB \\
q_1 X \xrightarrow{a} q_2 X & q_1 X \xrightarrow{b} q_2 X \\
q_1 A \xrightarrow{a} q_2 A & q_1 A \xrightarrow{b} q_2 A \\
q_1 B \xrightarrow{a} q_2 B & q_1 B \xrightarrow{b} q_2 B \\
q_1 X \xrightarrow{\varepsilon} q_2 X & \\
q_1 A \xrightarrow{\varepsilon} q_2 A & \\
q_1 B \xrightarrow{\varepsilon} q_2 B & \\
q_2 X \xrightarrow{\varepsilon} q_2 & \\
q_2 A \xrightarrow{a} q_2 & \\
q_2 B \xrightarrow{b} q_2 &
\end{array}
$$

Pushdown automaton

Two different definitions acceptace of words are used:

- A pushdown automaton $\mathcal{M}$ accepting by an **empty stack** accepts a word w iff there is some computation of $\mathcal{M}$ on w such that $\mathcal{M}$ reads all symbols of w and after reading them, the stack is empty.

- A pushdown automaton $\mathcal{M}$ accepting by an **accepting state** accepts a word w iff there is some computation of $\mathcal{M}$ on w such that $\mathcal{M}$ reads all symbols of w and after reading them, the control unit of $\mathcal{M}$ is in some state from a given set of accepting states F.

Pushdown automaton

- A word $w \in \Sigma^*$ is **accepted** by PDA $\mathcal{M}$ **by empty stack** iff

$$(q_0, w, X_0) \longrightarrow^* (q, \varepsilon, \varepsilon)$$

for some $q \in Q$.

Definition

The **langugage** $\mathcal{L}(\mathcal{M})$ **accepted** by PDA $\mathcal{M}$ **by empty stack** is defined as

$$\mathcal{L}(\mathcal{M}) = \{ w \in \Sigma^* \mid (\exists q \in Q)((q_0, w, X_0) \longrightarrow^* (q, \varepsilon, \varepsilon)) \} .$$

Pushdown automaton

Let us extend the definition of PDA $\mathcal{M}$ with a set of **accepting states** F (where $F \subseteq Q$).

- A word $w \in \Sigma^*$ is **accepted** by PDA $\mathcal{M}$ **by accepting state** iff

$$(q_0, w, X_0) \longrightarrow^* (q, \varepsilon, \alpha)$$

for some $q \in F$ and $\alpha \in \Gamma^*$.

Definition

The **language** $\mathcal{L}(\mathcal{M})$ **accepted** by PDA $\mathcal{M}$ **by accepting state** is defined as

$$\mathcal{L}(\mathcal{M}) = \{ w \in \Sigma^* \mid (\exists q \in F)(\exists \alpha \in \Gamma^*)((q_0, w, X_0) \longrightarrow^* (q, \varepsilon, \alpha)) \}.$$

Pushdown automata

In the case of **nondeterministic** pushdown automata, there is no difference in the class of accepted languages between recognizing by empty stack and recognizing by accepting state.

We can easily perform the following constructions:

- To construct for a given (nondeterministic) pushdown automaton, that recognizes a language L by empty stack, an equivalent (nondeterministic) pushdown automaton recognizing this language L by accepting states.
- To construct for a given (nondeterministic) pushdown automaton, that recognizes a language L by accepting states, an equivalent (nondeterministic) pushdown automaton recognizing the language L by empty stack.

Deterministic Pushdown Automata

A pushdown automaton $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, X_0)$ is **deterministic** when:

- For each $q \in Q$, $a \in (\Sigma \cup \{\varepsilon\})$ and $X \in \Gamma$ it holds that:

$$|\delta(q, a, X)| \leq 1$$

- For each $q \in Q$ and $X \in \Gamma$ holds at most one of the following possibilities:
    - There exists a rule $qX \xrightarrow{\varepsilon} q'\alpha$ for some $q' \in Q$ and $\alpha \in \Gamma^*$.
    - There exists a rule $qX \xrightarrow{a} q'\alpha$ for some $a \in \Sigma$, $q' \in Q$ and $\alpha \in \Gamma^*$.

Deterministic Pushdown Automata

Note that **deterministic** pushdown automata accepting by empty stack are able to recognize only **prefix-free** languages, i.e., languages L where:

- if $w \in L$, then there is no word $w' \in L$ such that w is a proper prefix of w'.

Remark: Instead of language $L \subseteq \Sigma^*$, that possibly is or is not prefix-free, we can take the prefix-free language

$$L' = L \cdot \{\dashv\}$$

over the alphabet $\Sigma \cup \{\dashv\}$, where $\dashv \notin \Sigma$ is a special "marker" representing the end of a word.
I.e., instead of testing whether $w \in L$, where $w \in \Sigma^*$, we can test whether $(w \dashv) \in L'$.

Deterministic Pushdown Automata

- For each deterministic pushdown automaton recognizing by empty stack we can easily construct an equivalent deterministic pushdown automaton recognizing by accepting states.

- For each deterministic pushdown automaton recognizing language L (where $L \subseteq \Sigma^*$) by accepting states we can easily construct a deterministic pushdown automaton recognizing by empty stack the language $L \cdot \{\dashv\}$, where $\dashv \notin \Sigma$.

Equivalence of CFG and PDA

Theorem

For every context-free grammar $\mathcal{G}$ we can construct a pushdown automaton $\mathcal{M}$ (with one control state) such that $\mathcal{L}(\mathcal{M}) = \mathcal{L}(\mathcal{G})$.

Proof: For CFG $\mathcal{G} = (\Pi, \Sigma, S, P)$ we construct PDA $\mathcal{M} = (\{q_0\}, \Sigma, \Gamma, \delta, q_0, S)$, where

- $\Gamma = \Pi \cup \Sigma$
- For each rule $(X \rightarrow \alpha) \in P$ from the context-free grammar $\mathcal{G}$ (where $X \in \Pi$ a $\alpha \in (\Pi \cup \Sigma)^*$), we add a corresponing rule

$$q_0 X \xrightarrow{\varepsilon} q_0 \alpha$$

  to the trasition function δ of the pushdown automaton $\mathcal{M}$.
- For each symbol $a \in \Sigma$, we add a rule

$$q_0 a \xrightarrow{a} q_0$$

  to the trasition function δ of the pushdown automaton $\mathcal{M}$.

Equivalence of CFG and PDA

Example: Consider a context-free grammar $\mathcal{G} = (\Pi, \Sigma, S, P)$, where

- $\Pi = \{S, E, T, F\}$
- $\Sigma = \{\, \mathtt{a}, \mathtt{+}, \mathtt{*}, \mathtt{(}, \mathtt{)}, \dashv \,\}$
- The set P contains the following rules:

$$
\begin{aligned}
S &\to E \dashv \\
E &\to T \mid E\mathtt{+}T \\
T &\to F \mid T\mathtt{*}F \\
F &\to \mathtt{a} \mid \mathtt{(}E\mathtt{)}
\end{aligned}
$$

Equivalence of CFG and PDA

For the given grammar $\mathcal{G} = (\Pi, \Sigma, S, P)$ with rules

$$\begin{aligned} S &\rightarrow E \dashv \\ E &\rightarrow T \mid E{+}T \\ T &\rightarrow F \mid T{*}F \\ F &\rightarrow \mathtt{a} \mid (E) \end{aligned}$$

we construct a pushdown automaton $\mathcal{M} = (\{q_0\}, \Sigma, \Gamma, \delta, q_0, S)$, where

- $\Sigma = \{\mathtt{a}, +, *, (,), \dashv\}$
- $\Gamma = \{S, E, T, F, \mathtt{a}, +, *, (,), \dashv\}$
- The trasition function δ contains the following rules:

$$q_0 S \xrightarrow{\varepsilon} q_0 E \dashv \qquad q_0 F \xrightarrow{\varepsilon} q_0 \mathtt{a} \qquad q_0 \mathtt{a} \xrightarrow{\mathtt{a}} q_0 \qquad q_0 (\xrightarrow{(} q_0$$

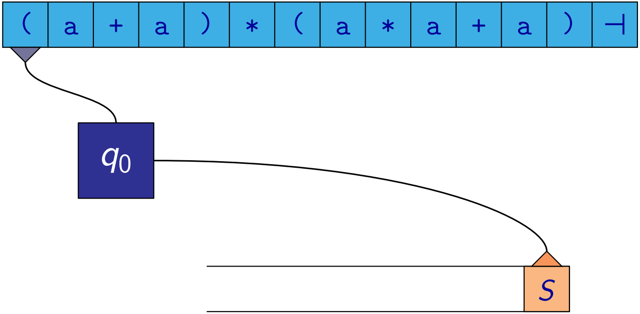
$$q_0 E \xrightarrow{\varepsilon} q_0 T \qquad q_0 F \xrightarrow{\varepsilon} q_0 (E) \qquad q_0 + \xrightarrow{+} q_0 \qquad q_0) \xrightarrow{)} q_0$$

$$q_0 E \xrightarrow{\varepsilon} q_0 E{+}T \qquad q_0 * \xrightarrow{*} q_0 \qquad q_0 \dashv \xrightarrow{\dashv} q_0$$

$$q_0 T \xrightarrow{\varepsilon} q_0 F$$

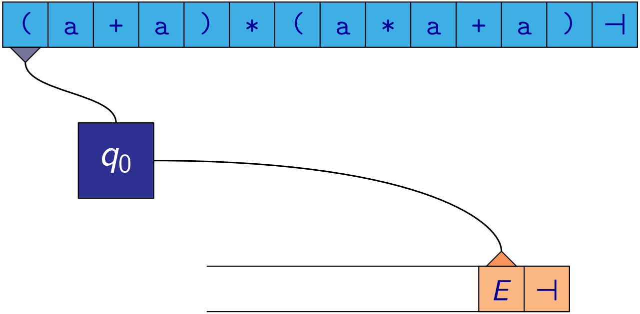
$$q_0 T \xrightarrow{\varepsilon} q_0 T{*}F$$

Equivalence of CFG and PDA



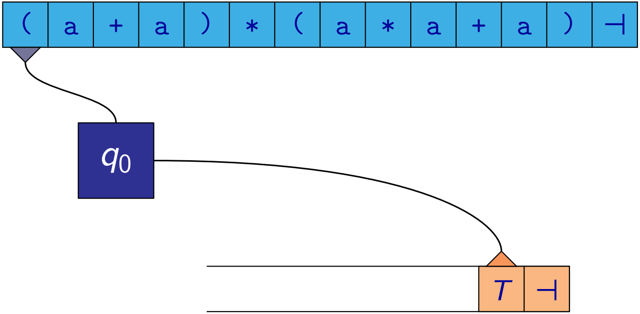
$\underline{S}$

Equivalence of CFG and PDA



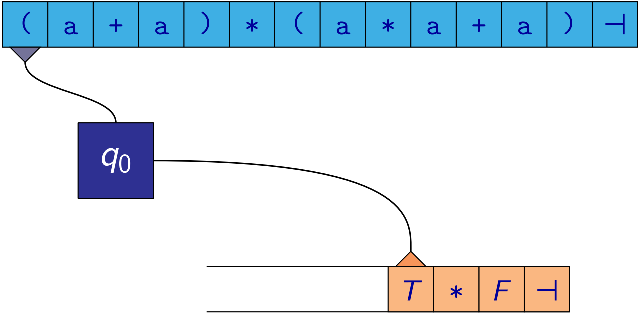
$\underline{S} \Rightarrow \underline{E}\dashv$

Equivalence of CFG and PDA



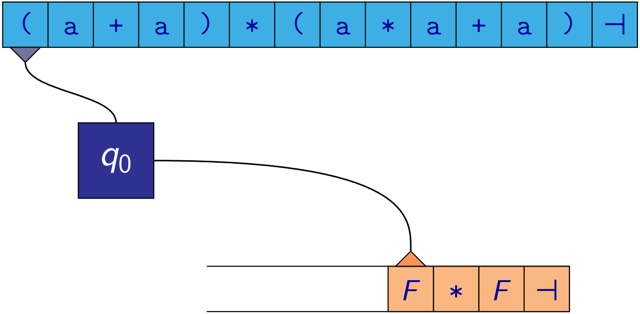
$\underline{S} \Rightarrow \underline{E}\dashv \Rightarrow \underline{T}\dashv$

Equivalence of CFG and PDA



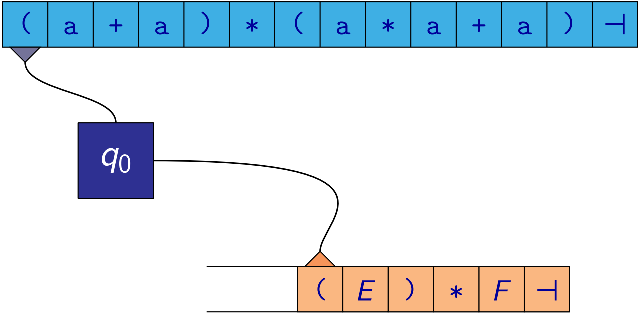
$\underline{S} \Rightarrow \underline{E}\dashv \Rightarrow \underline{T}\dashv \Rightarrow \underline{T}*F\dashv$

Equivalence of CFG and PDA



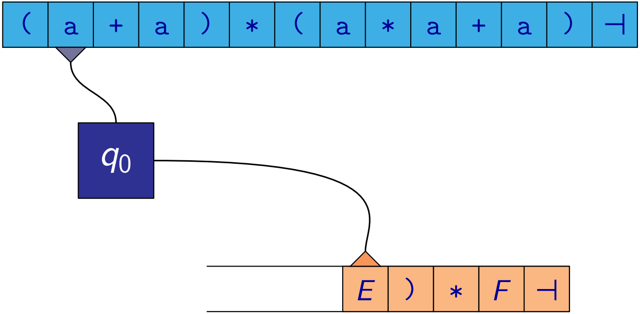
$\underline{S} \Rightarrow \underline{E}\dashv \Rightarrow \underline{T}\dashv \Rightarrow \underline{T}*F\dashv \Rightarrow \underline{F}*F\dashv$

Equivalence of CFG and PDA



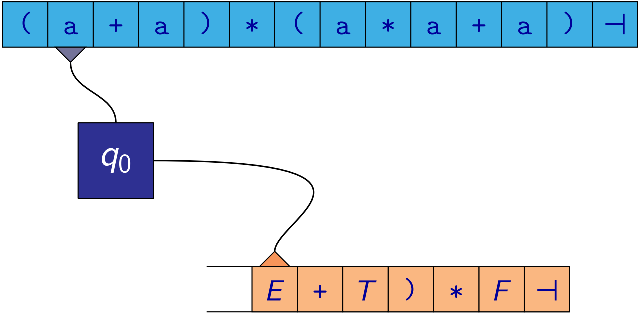
$\underline{S} \Rightarrow \underline{E}\dashv \Rightarrow \underline{T}\dashv \Rightarrow \underline{T}*F\dashv \Rightarrow \underline{F}*F\dashv \Rightarrow (\underline{E})*F\dashv$

Equivalence of CFG and PDA



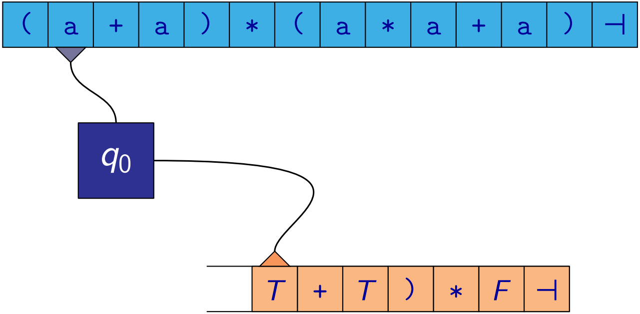
$$\underline{S} \Rightarrow \underline{E}\dashv \Rightarrow \underline{T}\dashv \Rightarrow \underline{T}*F\dashv \Rightarrow \underline{F}*F\dashv \Rightarrow (\underline{E})*F\dashv$$

Equivalence of CFG and PDA



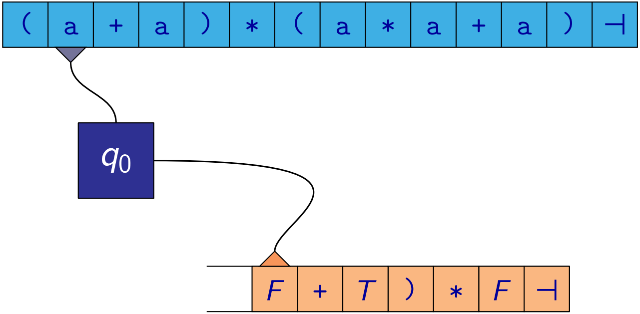
$$\cdots \Rightarrow \underline{T}\dashv \Rightarrow \underline{T}*F\dashv \Rightarrow \underline{F}*F\dashv \Rightarrow (\underline{E})*F\dashv \Rightarrow (\underline{E}+T)*F\dashv$$

Equivalence of CFG and PDA



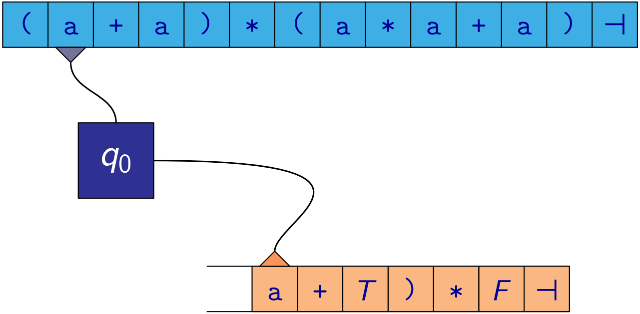
$\cdots \Rightarrow \underline{F}*F\dashv \Rightarrow (\underline{E})*F\dashv \Rightarrow (\underline{E}+T)*F\dashv \Rightarrow (\underline{T}+T)*F\dashv$

Equivalence of CFG and PDA



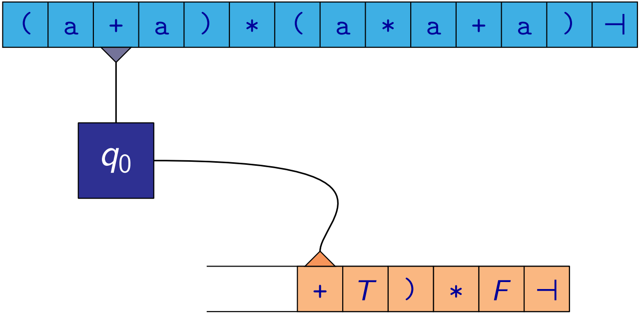
$$\cdots \Rightarrow (\underline{E})*F\dashv \Rightarrow (\underline{E}+T)*F\dashv \Rightarrow (\underline{T}+T)*F\dashv \Rightarrow (\underline{F}+T)*F\dashv$$

Equivalence of CFG and PDA



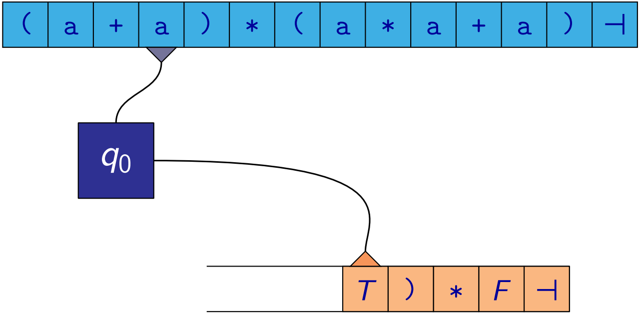
$\cdots \Rightarrow (\underline{E}+T)*F\dashv \Rightarrow (\underline{T}+T)*F\dashv \Rightarrow (\underline{F}+T)*F\dashv \Rightarrow (\mathtt{a}+\underline{T})*F\dashv$

Equivalence of CFG and PDA

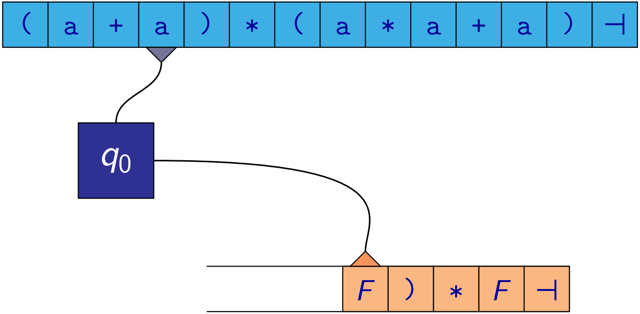


$\cdots \Rightarrow (\underline{E}+T)*F\dashv \Rightarrow (\underline{T}+T)*F\dashv \Rightarrow (\underline{F}+T)*F\dashv \Rightarrow (\mathrm{a}+\underline{T})*F\dashv$

Equivalence of CFG and PDA

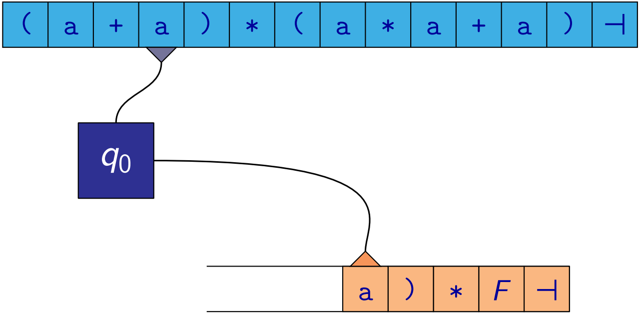


$$\cdots \Rightarrow (\underline{E}+T)*F\dashv \Rightarrow (\underline{T}+T)*F\dashv \Rightarrow (\underline{F}+T)*F\dashv \Rightarrow (\mathtt{a}+\underline{T})*F\dashv$$



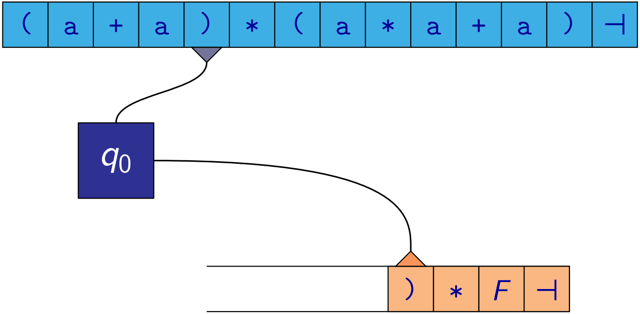
$$\cdots \Rightarrow (\underline{T}+T)*F\dashv \Rightarrow (\underline{F}+T)*F\dashv \Rightarrow (\mathtt{a}+\underline{T})*F\dashv \Rightarrow (\mathtt{a}+\underline{F})*F\dashv$$

Equivalence of CFG and PDA



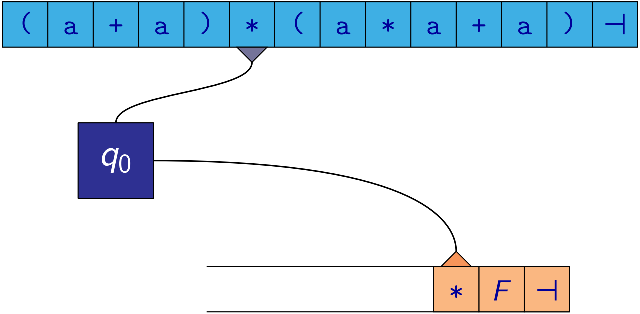
$\cdots \Rightarrow (\underline{F}+T)*F\dashv \Rightarrow (\mathrm{a}+\underline{T})*F\dashv \Rightarrow (\mathrm{a}+\underline{F})*F\dashv \Rightarrow (\mathrm{a}+\mathrm{a})*\underline{F}\dashv$

Equivalence of CFG and PDA



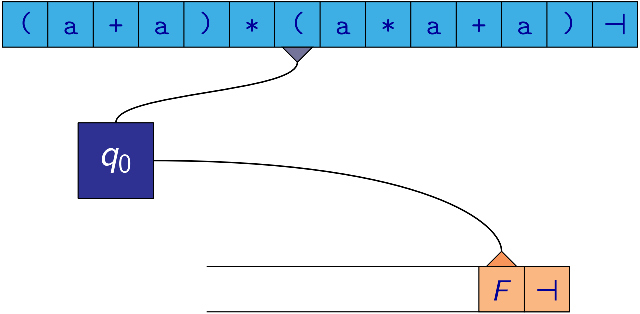
$\cdots \Rightarrow (\underline{F}+T)*F\dashv \Rightarrow (\text{a}+\underline{T})*F\dashv \Rightarrow (\text{a}+\underline{F})*F\dashv \Rightarrow (\text{a}+\text{a})*\underline{F}\dashv$

Equivalence of CFG and PDA



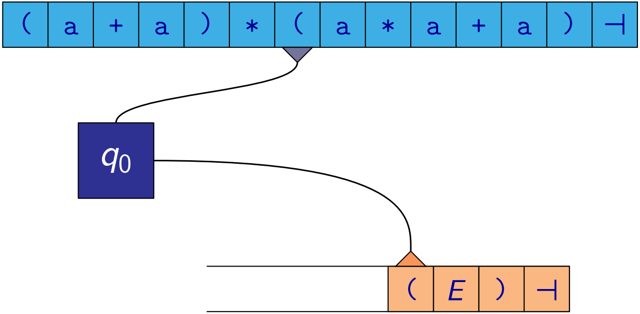
$$\cdots \Rightarrow (\underline{F}+T)*F\dashv \;\Rightarrow\; (\mathtt{a}+\underline{T})*F\dashv \;\Rightarrow\; (\mathtt{a}+\underline{F})*F\dashv \;\Rightarrow\; (\mathtt{a}+\mathtt{a})*\underline{F}\dashv$$

Equivalence of CFG and PDA



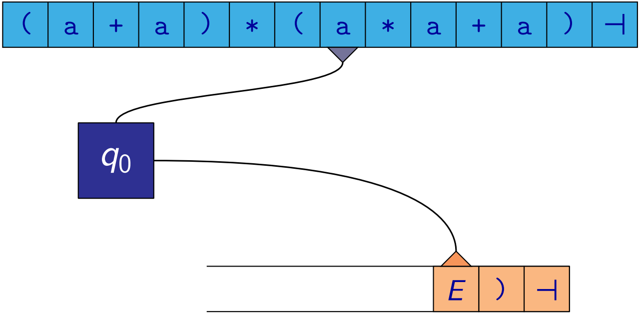
$$\cdots \Rightarrow (\underline{F}+T)*F\dashv \Rightarrow (\mathrm{a}+\underline{T})*F\dashv \Rightarrow (\mathrm{a}+\underline{F})*F\dashv \Rightarrow (\mathrm{a}+\mathrm{a})*\underline{F}\dashv$$

Equivalence of CFG and PDA



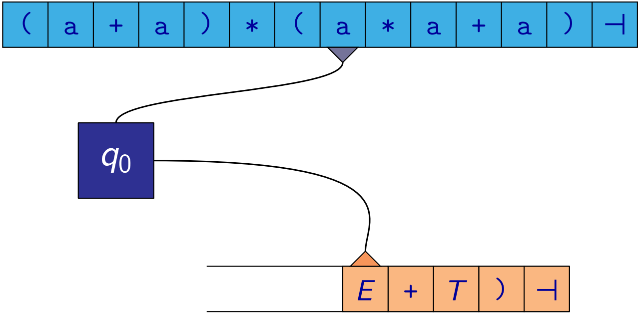
$\cdots \Rightarrow \texttt{(a+}\underline{T}\texttt{)*}F\dashv \Rightarrow \texttt{(a+}\underline{F}\texttt{)*}F\dashv \Rightarrow \texttt{(a+a)*}\underline{F}\dashv \Rightarrow \texttt{(a+a)*(}\underline{E}\texttt{)}\dashv$

Equivalence of CFG and PDA



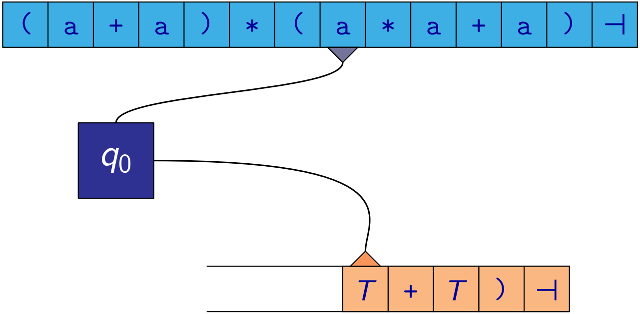
$\cdots \Rightarrow (a+\underline{T})*F\dashv \Rightarrow (a+\underline{F})*F\dashv \Rightarrow (a+a)*\underline{F}\dashv \Rightarrow (a+a)*(\underline{E})\dashv$

Equivalence of CFG and PDA



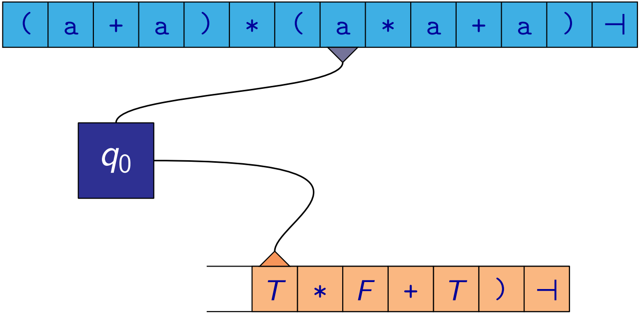
$\cdots \Rightarrow$ (a+a)$*\underline{F}\dashv \Rightarrow$ (a+a)$*$($\underline{E}$)$\dashv \Rightarrow$ (a+a)$*$($\underline{E}$+T)$\dashv$

Equivalence of CFG and PDA



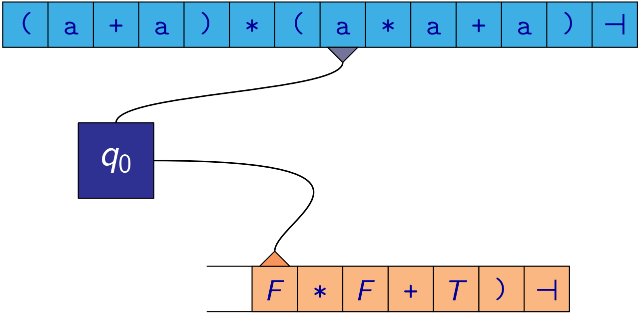
$$\cdots \Rightarrow \texttt{(a+a)*(}\underline{E}\texttt{)}\dashv \Rightarrow \texttt{(a+a)*(}\underline{E}\texttt{+}T\texttt{)}\dashv \Rightarrow \texttt{(a+a)*(}\underline{T}\texttt{+}T\texttt{)}\dashv$$

Equivalence of CFG and PDA



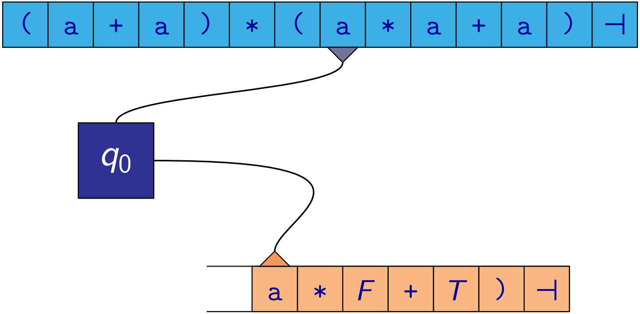
$$\cdots \Rightarrow \texttt{(a+a)*(}\underline{E}\texttt{+}T\texttt{)}\dashv \Rightarrow \texttt{(a+a)*(}\underline{T}\texttt{+}T\texttt{)}\dashv \Rightarrow \texttt{(a+a)*(}\underline{T}\texttt{*}F\texttt{+}T\texttt{)}\dashv$$

Equivalence of CFG and PDA



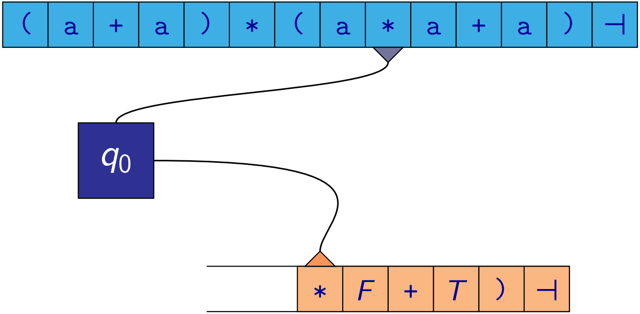
$\cdots \Rightarrow$ (a+a)*($\underline{T}$*F+T)⊣ $\Rightarrow$ (a+a)*($\underline{F}$*F+T)⊣

Equivalence of CFG and PDA



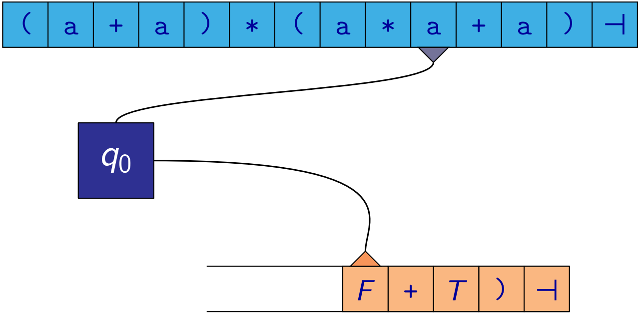
$$\cdots \;\Rightarrow\; (\mathtt{a}+\mathtt{a})*(\underline{F}*F+T)\dashv \;\Rightarrow\; (\mathtt{a}+\mathtt{a})*(\mathtt{a}*\underline{F}+T)\dashv$$

Equivalence of CFG and PDA



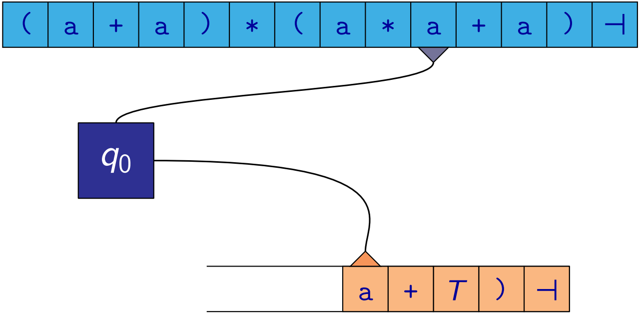
$\cdots \Rightarrow$ (a+a)*($\underline{F}$*F+T)⊣ $\Rightarrow$ (a+a)*(a*$\underline{F}$+T)⊣

Equivalence of CFG and PDA



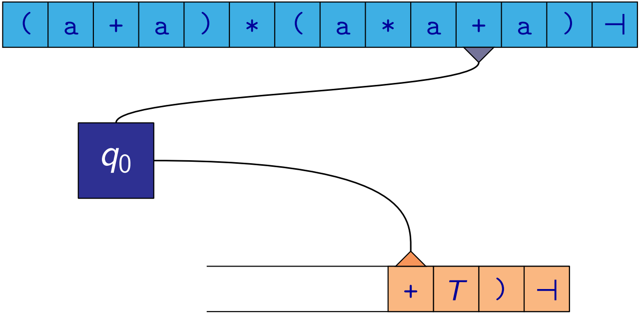
$$\cdots \Rightarrow \texttt{(a+a)*(}\underline{F}\texttt{*}F\texttt{+}T\texttt{)}\dashv \;\Rightarrow\; \texttt{(a+a)*(a*}\underline{F}\texttt{+}T\texttt{)}\dashv$$

Equivalence of CFG and PDA



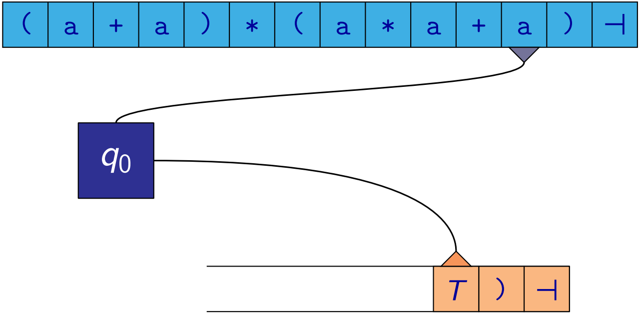
$\cdots \Rightarrow$ (a+a)*(a*$\underline{F}$+T)⊣ $\Rightarrow$ (a+a)*(a*a+$\underline{T}$)⊣

Equivalence of CFG and PDA



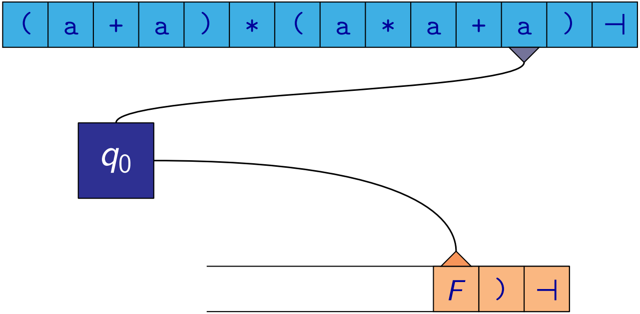
$\cdots \Rightarrow$ (a+a)*(a*$\underline{F}$+T)⊣ $\Rightarrow$ (a+a)*(a*a+$\underline{T}$)⊣

Equivalence of CFG and PDA



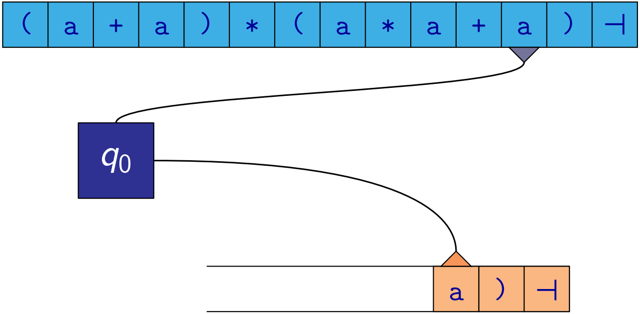
$\cdots \Rightarrow$ (a+a)*(a*$\underline{F}$+T)⊣ $\Rightarrow$ (a+a)*(a*a+$\underline{T}$)⊣

Equivalence of CFG and PDA



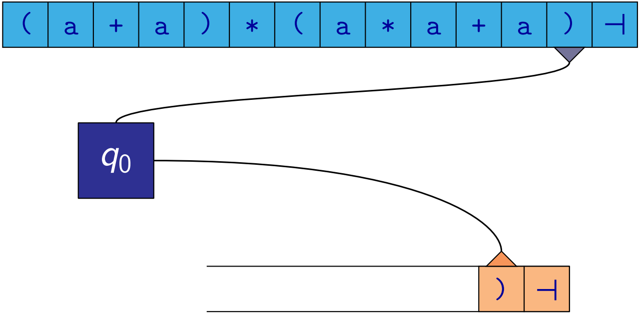
$\cdots \Rightarrow$ (a+a)*(a*a+$\underline{T}$)⊣ $\Rightarrow$ (a+a)*(a*a+$\underline{F}$)⊣

Equivalence of CFG and PDA



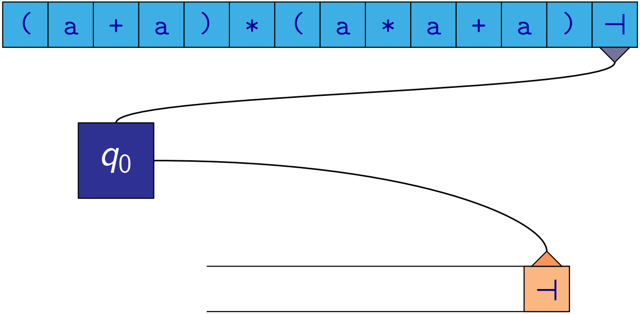
$$\cdots \Rightarrow \texttt{(a+a)*(a*a+}\underline{F}\texttt{)}\dashv \Rightarrow \texttt{(a+a)*(a*a+a)}\dashv$$

Equivalence of CFG and PDA



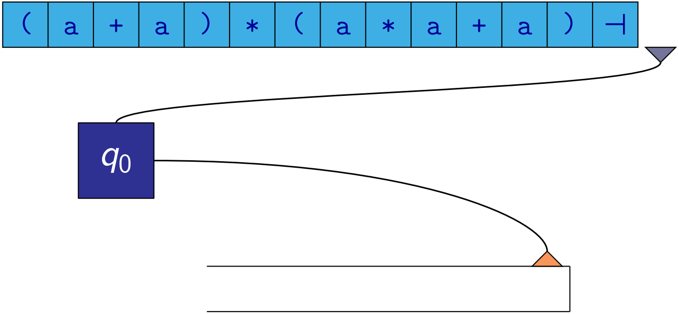
$\cdots \Rightarrow$ (a+a)*(a*a+$\underline{F}$)⊣ $\Rightarrow$ (a+a)*(a*a+a)⊣

Equivalence of CFG and PDA



$\cdots \;\Rightarrow\; (a+a)*(a*a+\underline{F})\dashv \;\Rightarrow\; (a+a)*(a*a+a)\dashv$

Equivalence of CFG and PDA



$\cdots \Rightarrow$ (a+a)*(a*a+$\underline{F}$)$\dashv$ $\Rightarrow$ (a+a)*(a*a+a)$\dashv$

Equivalence of CFG and PDA

We can see from the previous example that the pushdown automaton $\mathcal{M}$ basically performs a **left derivation** in grammar $\mathcal{G}$.

It can be easily shown that:

- For every left derivation in grammar $\mathcal{G}$ there is some corresponding computation of automaton $\mathcal{M}$.
- For every computation of automaton $\mathcal{M}$ there is some corresponding left derivation in grammar $\mathcal{G}$.

Remark: The described approach corresponds to the syntactic analysis that proceeds **top down**.

Equivalence of CFG and PDA

Alternatively, it is also possible to proceed from **bottom up**.

This could be implemented by a nondeterministic pushdown automaton $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, X_0)$ constructed for a given grammar $\mathcal{G} = (\Pi, \Sigma, S, P)$ as follows:

- $\Gamma = \Pi \cup \Sigma \cup \{\vdash\}$, where $\vdash \notin (\Pi \cup \Sigma)$
- $X_0 = \vdash$
- Q contains states corresponding to all suffixes of right-hand sides from P a also a special state $\langle S \rangle$ (where $S \in \Pi$ is the initial nonterminal of grammar $\mathcal{G}$) and a special state q_{acc}.

  A state corresponding to suffix α (where $\alpha \in (\Pi \cup \Sigma)^*$) will be denoted $\langle \alpha \rangle$.

  A special case is a state corresponding to suffix ε. This state will be denoted $\langle \rangle$.
- $q_0 = \langle \rangle$

Equivalence of CFG and PDA

- For every input symbol $a \in \Sigma$ and every stack symbol $W \in \Gamma$ the following rule is added to δ:

$$\langle\rangle W \xrightarrow{a} \langle\rangle aW$$

- For every rule $X \rightarrow Y_1 Y_2 \cdots Y_n$ from grammar $\mathcal{G}$ (where $X \in \Pi$, $n \geq 0$, and $Y_i \in (\Pi \cup \Sigma)$ for $1 \leq i \leq n$) the following set of rules is added to δ:

$$\begin{aligned}
\langle\rangle Y_n &\xrightarrow{\varepsilon} \langle Y_n \rangle \\
\langle Y_n \rangle Y_{n-1} &\xrightarrow{\varepsilon} \langle Y_{n-1} Y_n \rangle \\
\langle Y_{n-1} Y_n \rangle Y_{n-2} &\xrightarrow{\varepsilon} \langle Y_{n-2} Y_{n-1} Y_n \rangle \\
&\vdots \\
\langle Y_2 Y_3 \ldots Y_n \rangle Y_1 &\xrightarrow{\varepsilon} \langle Y_1 Y_2 Y_3 \cdots Y_n \rangle
\end{aligned}$$

and for every $W \in \Gamma$ we add the rules

$$\langle Y_1 Y_2 \cdots Y_n \rangle W \xrightarrow{\varepsilon} \langle\rangle XW$$

Equivalence of CFG and PDA

- For example if grammar $\mathcal{G}$ contains rule

$$B \rightarrow CaADb$$

the transition function δ of automaton $\mathcal{M}$ will contain rules

$$\langle\rangle b \xrightarrow{\varepsilon} \langle b\rangle$$
$$\langle b\rangle D \xrightarrow{\varepsilon} \langle Db\rangle$$
$$\langle Db\rangle A \xrightarrow{\varepsilon} \langle ADb\rangle$$
$$\langle ADb\rangle a \xrightarrow{\varepsilon} \langle aADb\rangle$$
$$\langle aADb\rangle C \xrightarrow{\varepsilon} \langle CaADb\rangle$$

and also for every $W \in \Gamma$ the will be a rule

$$\langle CaADb\rangle W \xrightarrow{\varepsilon} \langle\rangle BW$$

Equivalence of CFG and PDA

- In particular, for ε-rules of grammar $\mathcal{G}$, the corresponding rules will be as follows: for ε-rule
$$X \rightarrow \varepsilon$$
of grammar $\mathcal{G}$, where $X \in \Pi$, there will be corresponding rules
$$\langle\rangle W \xrightarrow{\varepsilon} \langle\rangle XW$$
where $W \in \Gamma$.

- We finish the construction by adding the following two special rules to δ (where $S \in \Pi$ is the initial nonterminal of grammar $\mathcal{G}$):
$$\langle\rangle S \xrightarrow{\varepsilon} \langle S\rangle \qquad\qquad \langle S\rangle \vdash \xrightarrow{\varepsilon} q_{acc}$$

Equivalence of CFG and PDA

Example: Consider the same grammar $\mathcal{G}$ as in the previous example:

$$
\begin{aligned}
S &\rightarrow E \dashv \\
E &\rightarrow T \mid E{+}T \\
T &\rightarrow F \mid T{*}F \\
F &\rightarrow \mathtt{a} \mid (E)
\end{aligned}
$$

For this grammar we construct a corresponding pushdown automaton $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, X_0)$, where

- $\Sigma = \{ \mathtt{a}, +, *, (,), \dashv \}$
- $\Gamma = \{ S, E, T, F, \mathtt{a}, +, *, (,), \dashv, \vdash \}$
- $Q = \{ \langle\rangle, \langle\dashv\rangle, \langle E \dashv\rangle, \langle T\rangle, \langle +T\rangle, \langle E{+}T\rangle, \langle F\rangle, \langle *F\rangle, \langle T{*}F\rangle,$
  $\langle \mathtt{a}\rangle, \langle)\rangle, \langle E)\rangle, \langle (E)\rangle, \langle S\rangle, q_{acc} \}$
- $q_0 = \langle\rangle$
- $X_0 = \vdash$

Equivalence of CFG and PDA

For each $X \in \Gamma$ the following rules are added to δ:

$$\langle\rangle X \xrightarrow{\mathtt{a}} \langle\rangle \mathtt{a} X$$
$$\langle\rangle X \xrightarrow{+} \langle\rangle + X$$
$$\langle\rangle X \xrightarrow{*} \langle\rangle * X$$
$$\langle\rangle X \xrightarrow{(} \langle\rangle (X$$
$$\langle\rangle X \xrightarrow{)} \langle\rangle) X$$
$$\langle\rangle X \xrightarrow{\dashv} \langle\rangle \dashv X$$

$$\langle\rangle S \xrightarrow{\varepsilon} \langle S\rangle$$
$$\langle S\rangle \vdash \xrightarrow{\varepsilon} q_{acc}$$

$$\langle\rangle \dashv \xrightarrow{\varepsilon} \langle \dashv\rangle$$
$$\langle \dashv\rangle E \xrightarrow{\varepsilon} \langle E \dashv\rangle \qquad \langle E \dashv\rangle X \xrightarrow{\varepsilon} \langle\rangle S X$$

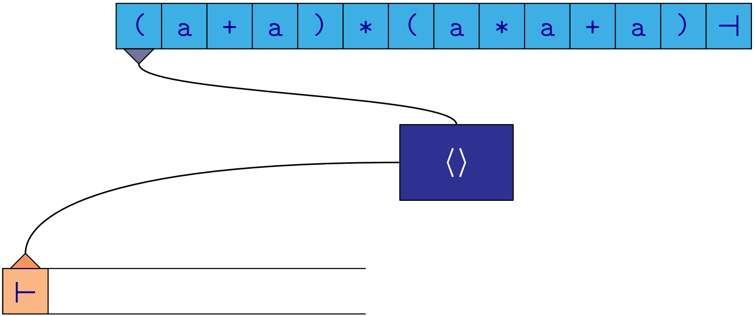
$$\langle\rangle T \xrightarrow{\varepsilon} \langle T\rangle \qquad \langle T\rangle X \xrightarrow{\varepsilon} \langle\rangle E X$$
$$\langle T\rangle + \xrightarrow{\varepsilon} \langle + T\rangle$$
$$\langle + T\rangle E \xrightarrow{\varepsilon} \langle E + T\rangle \qquad \langle E + T\rangle X \xrightarrow{\varepsilon} \langle\rangle E X$$

$$\langle\rangle F \xrightarrow{\varepsilon} \langle F\rangle \qquad \langle F\rangle X \xrightarrow{\varepsilon} \langle\rangle T X$$
$$\langle F\rangle * \xrightarrow{\varepsilon} \langle * F\rangle$$
$$\langle * F\rangle T \xrightarrow{\varepsilon} \langle T * F\rangle \qquad \langle T * F\rangle X \xrightarrow{\varepsilon} \langle\rangle T X$$

$$\langle\rangle \mathtt{a} \xrightarrow{\varepsilon} \langle \mathtt{a}\rangle \qquad \langle \mathtt{a}\rangle X \xrightarrow{\varepsilon} \langle\rangle F X$$

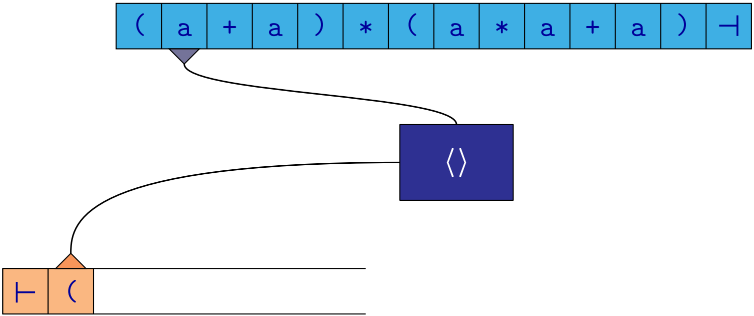
$$\langle\rangle) \xrightarrow{\varepsilon} \langle)\rangle$$
$$\langle)\rangle E \xrightarrow{\varepsilon} \langle E)\rangle$$
$$\langle E)\rangle (\xrightarrow{\varepsilon} \langle (E)\rangle \qquad \langle (E)\rangle X \xrightarrow{\varepsilon} \langle\rangle F X$$

Equivalence of CFG and PDA



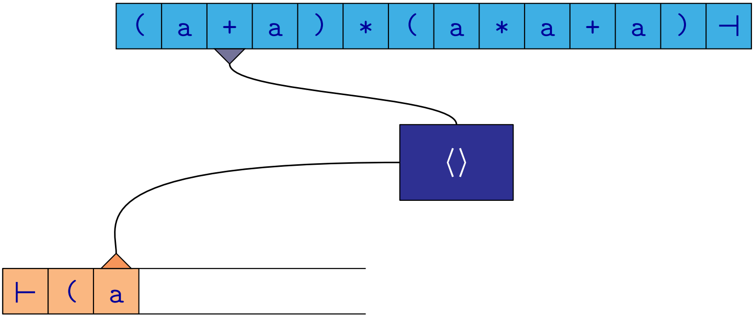
(a+a)*(a*a+a) ⊣

Equivalence of CFG and PDA



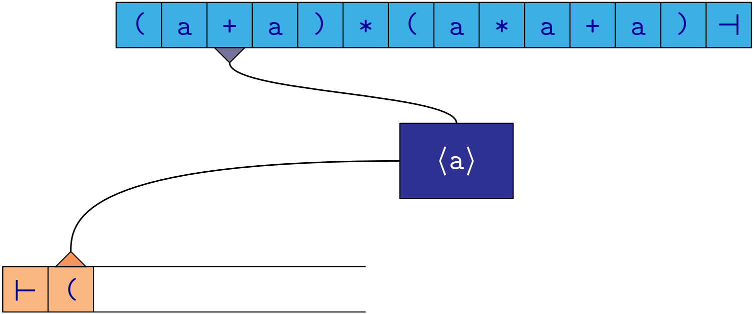
(a+a)*(a*a+a) ⊣

Equivalence of CFG and PDA



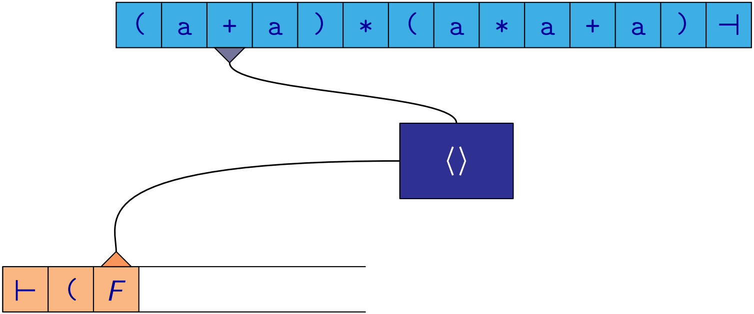
$\texttt{(a+a)*(a*a+a)} \dashv$

Equivalence of CFG and PDA



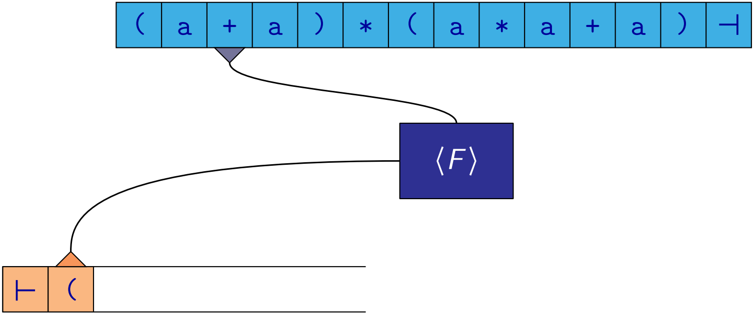
$\texttt{(a+a)*(a*a+a)}\dashv$

Equivalence of CFG and PDA



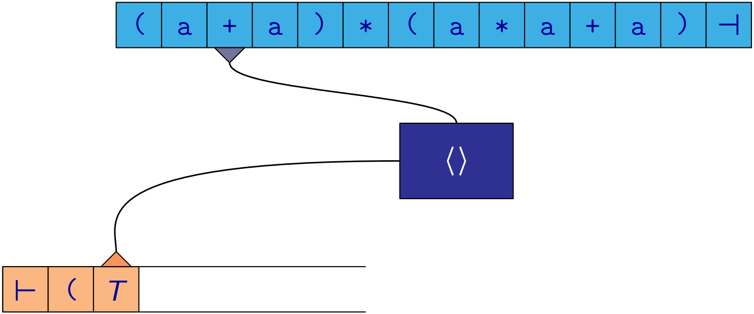
$(\underline{F}+a)*(a*a+a)\dashv \;\Rightarrow\; (a+a)*(a*a+a)\dashv$

Equivalence of CFG and PDA



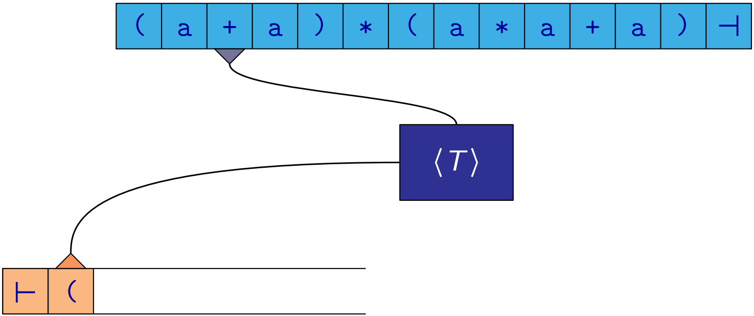
$(\underline{F}+\mathtt{a})*(\mathtt{a}*\mathtt{a}+\mathtt{a}) \dashv \;\Rightarrow\; (\mathtt{a}+\mathtt{a})*(\mathtt{a}*\mathtt{a}+\mathtt{a}) \dashv$

Equivalence of CFG and PDA



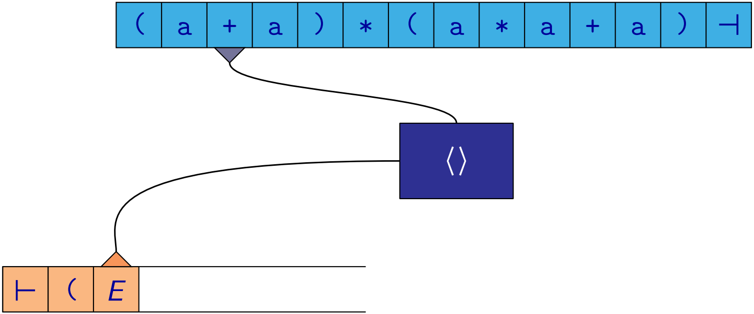
$(\underline{T}\texttt{+a)*(a*a+a)}\dashv \;\Rightarrow\; (\underline{F}\texttt{+a)*(a*a+a)}\dashv \;\Rightarrow\; \texttt{(a+a)*(a*a+a)}\dashv$

Equivalence of CFG and PDA



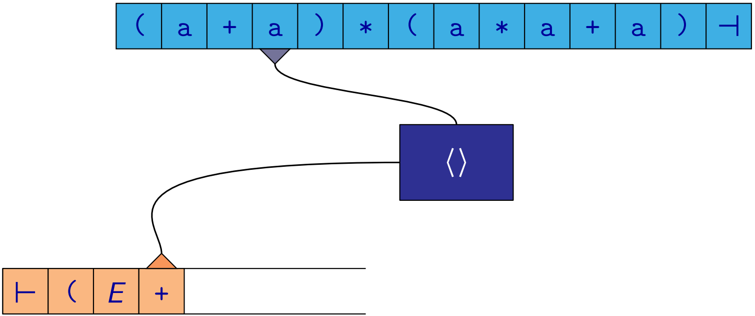
$(\underline{T}\text{+a)*(a*a+a)}\dashv \;\Rightarrow\; (\underline{F}\text{+a)*(a*a+a)}\dashv \;\Rightarrow\; \text{(a+a)*(a*a+a)}\dashv$

Equivalence of CFG and PDA



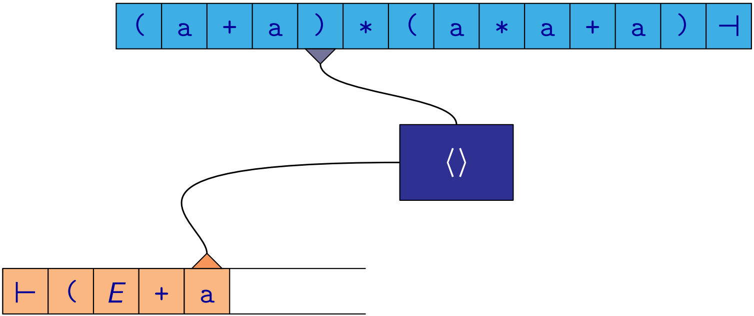
$(\underline{E}+a)*(a*a+a)\dashv \;\Rightarrow\; (\underline{T}+a)*(a*a+a)\dashv \;\Rightarrow\; (\underline{F}+a)*(a*a+a)\dashv \;\Rightarrow$
$\dots$

Equivalence of CFG and PDA



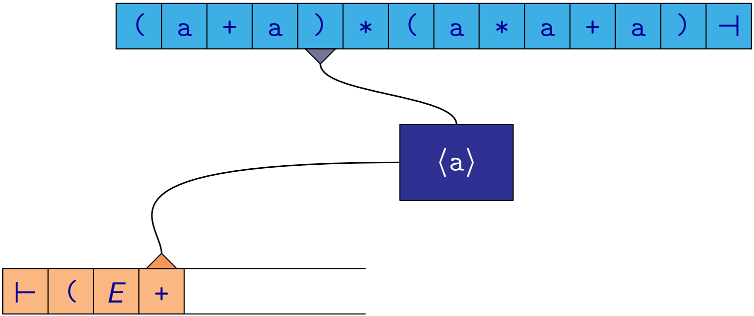
$(\underline{E}\texttt{+a)*(a*a+a)} \dashv \;\Rightarrow\; (\underline{T}\texttt{+a)*(a*a+a)} \dashv \;\Rightarrow\; (\underline{F}\texttt{+a)*(a*a+a)} \dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



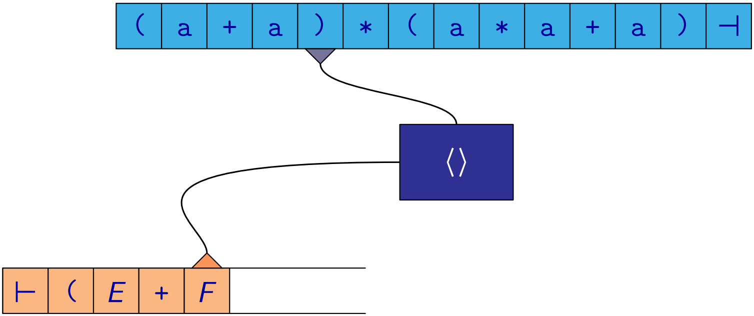
$(\underline{E}+\mathtt{a})*(\mathtt{a}*\mathtt{a}+\mathtt{a})\dashv \;\Rightarrow\; (\underline{T}+\mathtt{a})*(\mathtt{a}*\mathtt{a}+\mathtt{a})\dashv \;\Rightarrow\; (\underline{F}+\mathtt{a})*(\mathtt{a}*\mathtt{a}+\mathtt{a})\dashv \;\Rightarrow$
$\ldots$

Equivalence of CFG and PDA



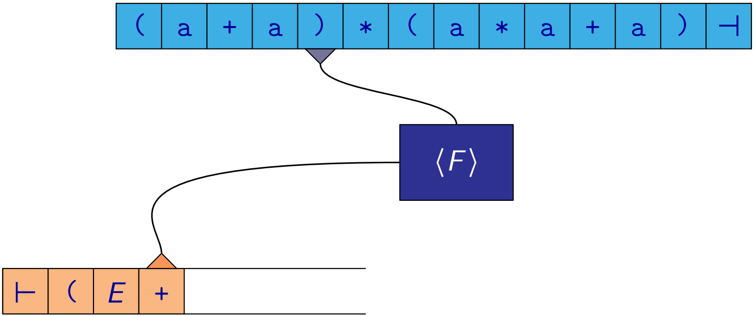
$(\underline{E}+a)*(a*a+a)\dashv \;\Rightarrow\; (\underline{T}+a)*(a*a+a)\dashv \;\Rightarrow\; (\underline{F}+a)*(a*a+a)\dashv \;\Rightarrow$
$\cdots$

Equivalence of CFG and PDA



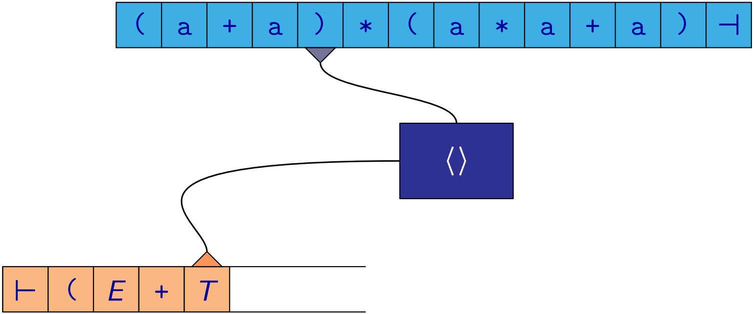
$(E+\underline{F})*(a*a+a)\dashv \;\Rightarrow\; (\underline{E}+a)*(a*a+a)\dashv \;\Rightarrow\; (\underline{T}+a)*(a*a+a)\dashv \;\Rightarrow\; \ldots$

Equivalence of CFG and PDA



$(E+\underline{F})*(a*a+a)\dashv \;\Rightarrow\; (\underline{E}+a)*(a*a+a)\dashv \;\Rightarrow\; (\underline{T}+a)*(a*a+a)\dashv \;\Rightarrow$
$\cdots$

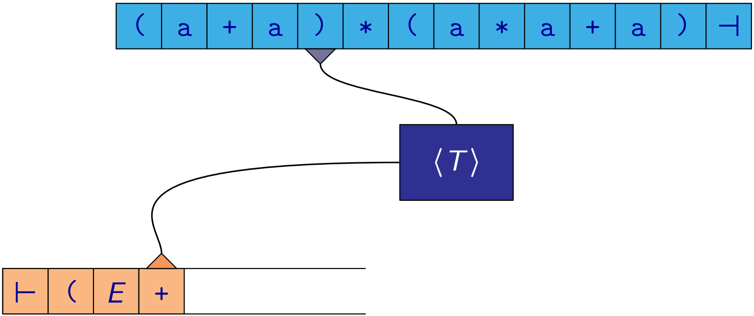
Equivalence of CFG and PDA



$(E+\underline{T})*(a*a+a) \dashv \;\Rightarrow\; (E+\underline{F})*(a*a+a) \dashv \;\Rightarrow\; (\underline{E}+a)*(a*a+a) \dashv \;\Rightarrow$

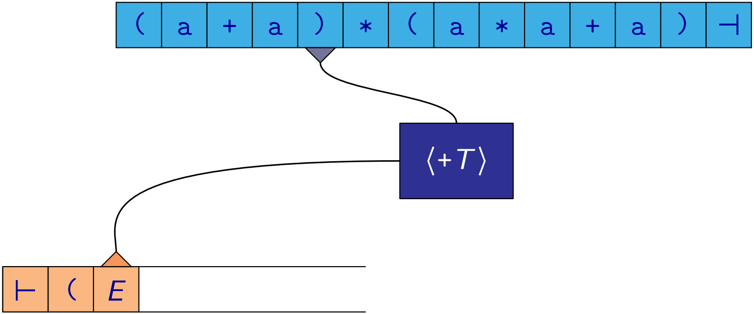
$\ldots$

Equivalence of CFG and PDA



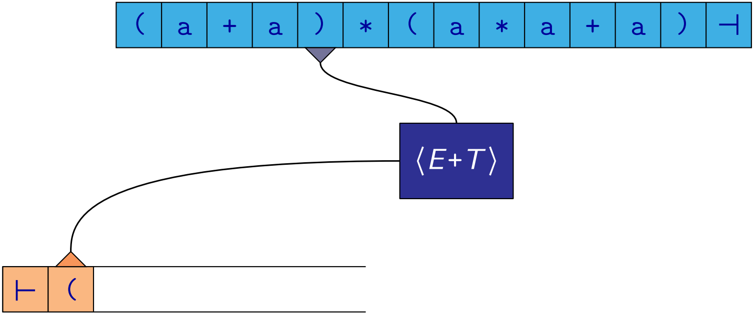
$(E{+}\underline{T}){*}(\mathtt{a{*}a{+}a}) \dashv \;\Rightarrow\; (E{+}\underline{F}){*}(\mathtt{a{*}a{+}a}) \dashv \;\Rightarrow\; (\underline{E}{+}\mathtt{a}){*}(\mathtt{a{*}a{+}a}) \dashv \;\Rightarrow$
$\cdots$

Equivalence of CFG and PDA



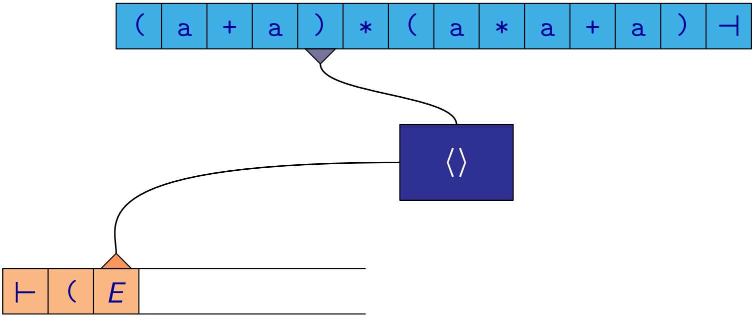
$(E+\underline{T})*(a*a+a)\dashv \;\Rightarrow\; (E+\underline{F})*(a*a+a)\dashv \;\Rightarrow\; (\underline{E}+a)*(a*a+a)\dashv \;\Rightarrow$
$\ldots$

Equivalence of CFG and PDA



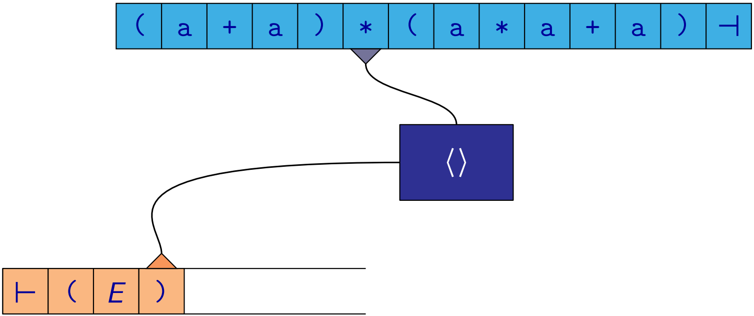
$(E+\underline{T})*(a*a+a) \dashv \;\Rightarrow\; (E+\underline{F})*(a*a+a) \dashv \;\Rightarrow\; (\underline{E}+a)*(a*a+a) \dashv \;\Rightarrow$
$\ldots$

Equivalence of CFG and PDA



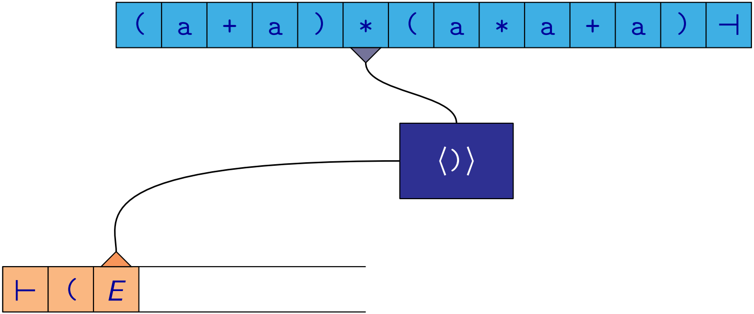
$(\underline{E})*(a*a+a)\dashv \;\Rightarrow\; (E+\underline{T})*(a*a+a)\dashv \;\Rightarrow\; (E+\underline{F})*(a*a+a)\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



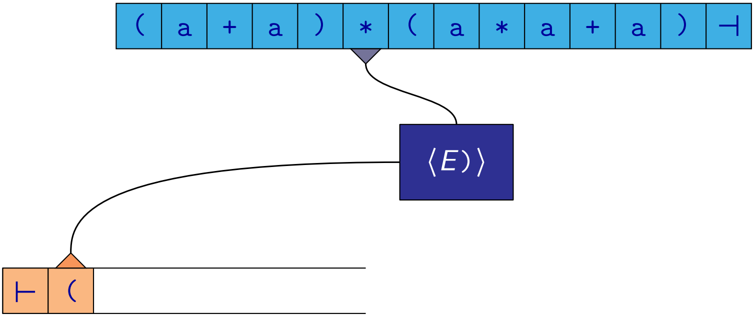
$(\underline{E})*(a*a+a)\dashv \;\Rightarrow\; (E+\underline{T})*(a*a+a)\dashv \;\Rightarrow\; (E+\underline{F})*(a*a+a)\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



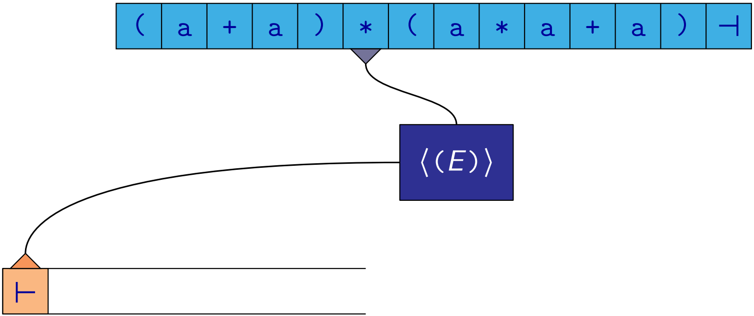
$(\underline{E})*(a*a+a)\dashv \;\Rightarrow\; (E+\underline{T})*(a*a+a)\dashv \;\Rightarrow\; (E+\underline{F})*(a*a+a)\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



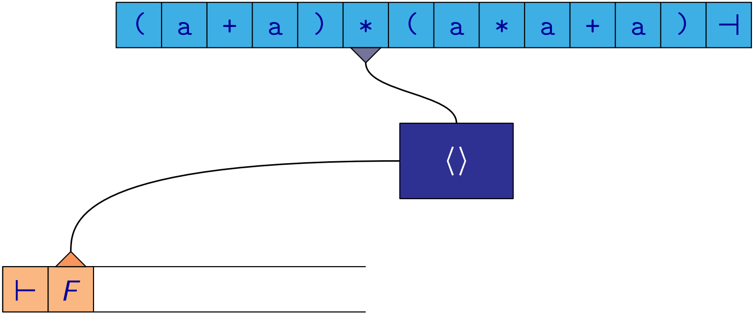
$(\underline{E})*(a*a+a)\dashv \;\Rightarrow\; (E+\underline{T})*(a*a+a)\dashv \;\Rightarrow\; (E+\underline{F})*(a*a+a)\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



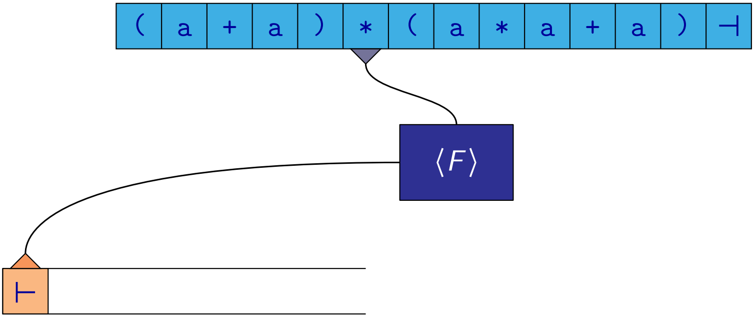
$(\underline{E})*(a*a+a)\dashv \;\Rightarrow\; (E+\underline{T})*(a*a+a)\dashv \;\Rightarrow\; (E+\underline{F})*(a*a+a)\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



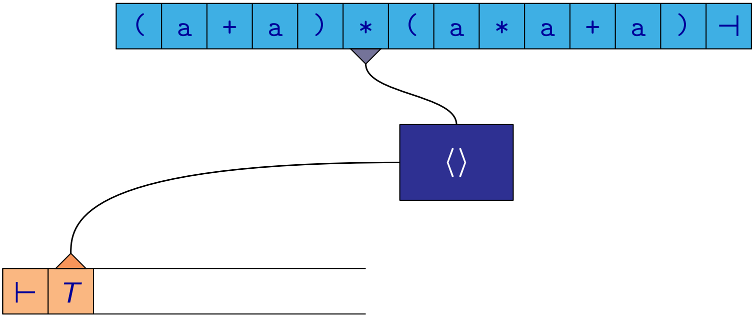
$\underline{F}$∗(a∗a+a) ⊣ ⇒ ($\underline{E}$)∗(a∗a+a) ⊣ ⇒ (E+$\underline{T}$)∗(a∗a+a) ⊣ ⇒ ⋯

Equivalence of CFG and PDA



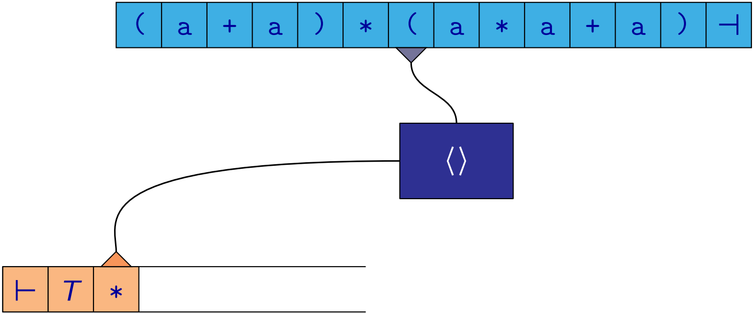
$\underline{F}*(a*a+a)\dashv \;\Rightarrow\; (\underline{E})*(a*a+a)\dashv \;\Rightarrow\; (E+\underline{T})*(a*a+a)\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



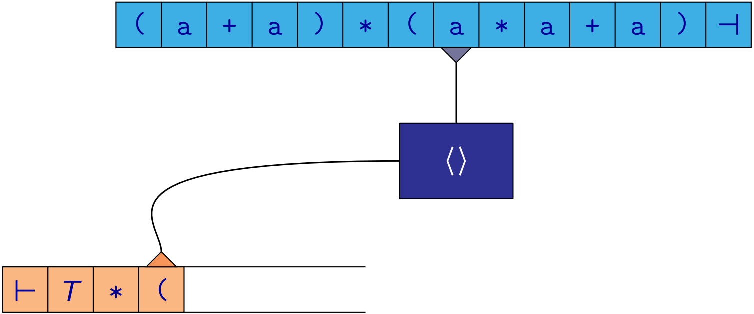
$\underline{T}*(a*a+a)\dashv \;\Rightarrow\; \underline{F}*(a*a+a)\dashv \;\Rightarrow\; (\underline{E})*(a*a+a)\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



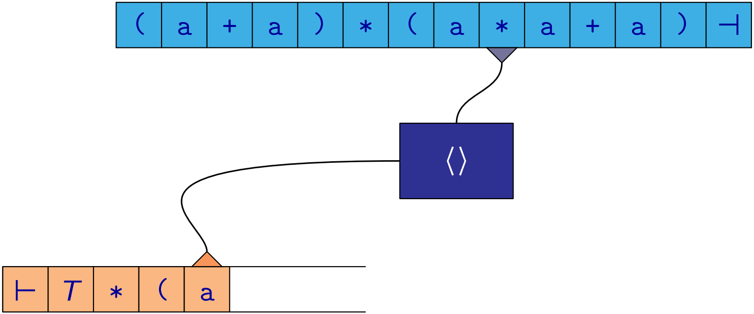
$\underline{T}$*(a*a+a)⊣ ⇒ $\underline{F}$*(a*a+a)⊣ ⇒ ($\underline{E}$)*(a*a+a)⊣ ⇒ ⋯

Equivalence of CFG and PDA



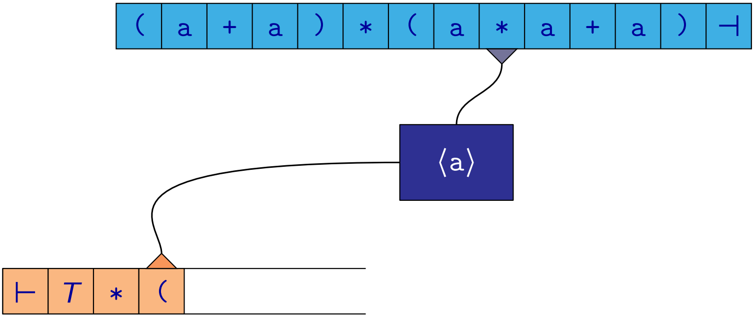
$\underline{T}*(a*a+a)\dashv \;\Rightarrow\; \underline{F}*(a*a+a)\dashv \;\Rightarrow\; (\underline{E})*(a*a+a)\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



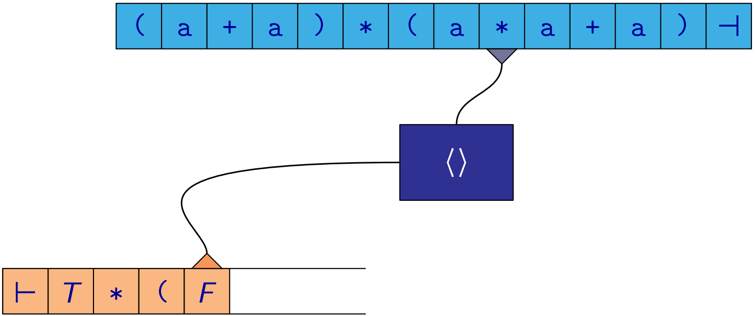
$\underline{T}*(a*a+a) \dashv \; \Rightarrow \; \underline{F}*(a*a+a) \dashv \; \Rightarrow \; (\underline{E})*(a*a+a) \dashv \; \Rightarrow \; \cdots$

Equivalence of CFG and PDA



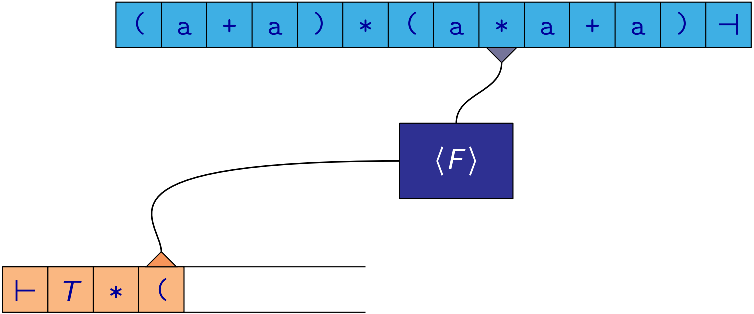
$\underline{T}$∗(a∗a+a) ⊣ ⇒ $\underline{F}$∗(a∗a+a) ⊣ ⇒ ($\underline{E}$)∗(a∗a+a) ⊣ ⇒ ⋯

Equivalence of CFG and PDA



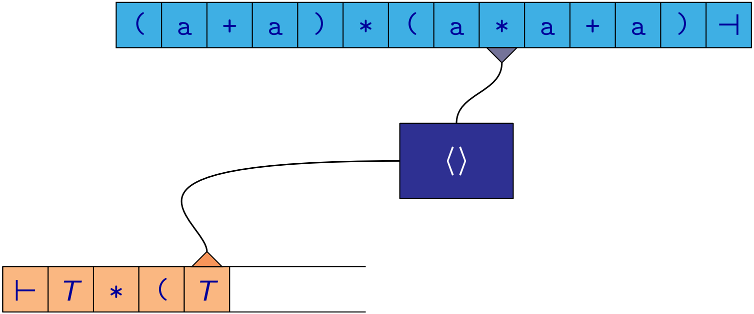
$T*(\underline{F}*\mathtt{a}+\mathtt{a})\dashv \;\Rightarrow\; \underline{T}*(\mathtt{a}*\mathtt{a}+\mathtt{a})\dashv \;\Rightarrow\; \underline{F}*(\mathtt{a}*\mathtt{a}+\mathtt{a})\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



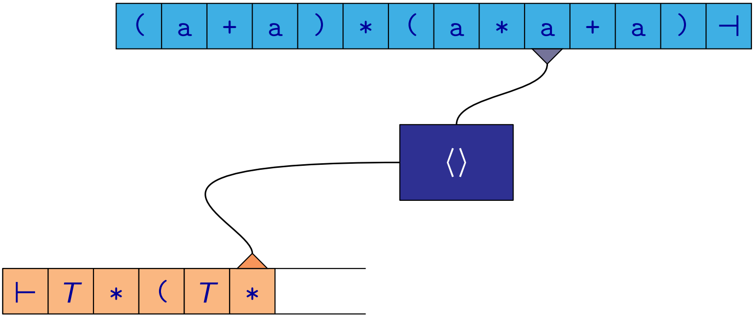
$T*(\underline{F}*a+a)\dashv \;\Rightarrow\; \underline{T}*(a*a+a)\dashv \;\Rightarrow\; \underline{F}*(a*a+a)\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



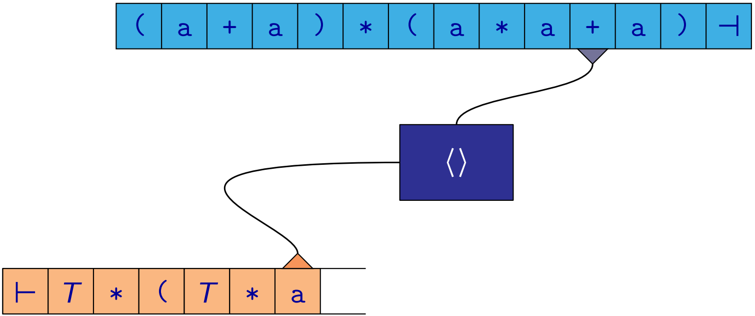
$T*(\underline{T}*\mathtt{a}+\mathtt{a})\dashv \;\Rightarrow\; T*(\underline{F}*\mathtt{a}+\mathtt{a})\dashv \;\Rightarrow\; \underline{T}*(\mathtt{a}*\mathtt{a}+\mathtt{a})\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



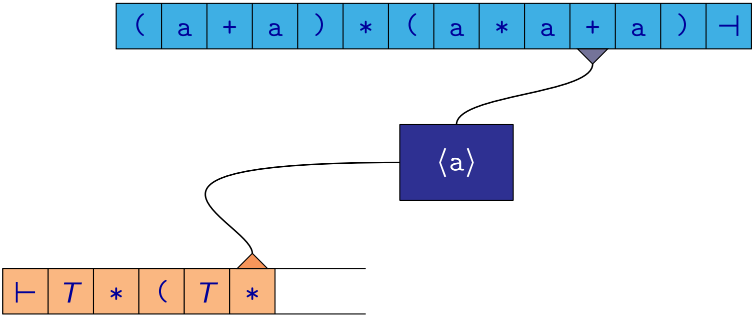
$T*(\underline{T}*\mathtt{a}+\mathtt{a})\dashv \;\Rightarrow\; T*(\underline{F}*\mathtt{a}+\mathtt{a})\dashv \;\Rightarrow\; \underline{T}*(\mathtt{a}*\mathtt{a}+\mathtt{a})\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



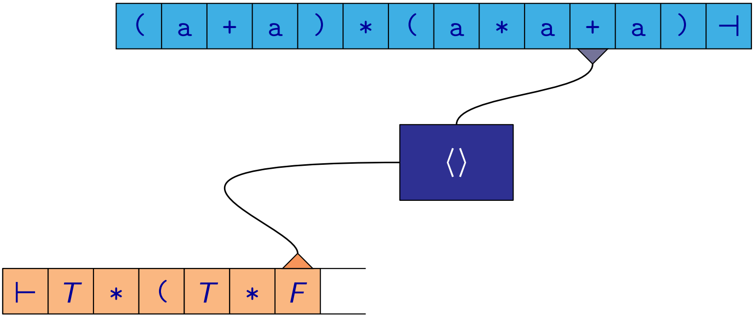
$$T*(\underline{T}*\mathtt{a}+\mathtt{a})\dashv \;\Rightarrow\; T*(\underline{F}*\mathtt{a}+\mathtt{a})\dashv \;\Rightarrow\; \underline{T}*(\mathtt{a}*\mathtt{a}+\mathtt{a})\dashv \;\Rightarrow\; \cdots$$

Equivalence of CFG and PDA



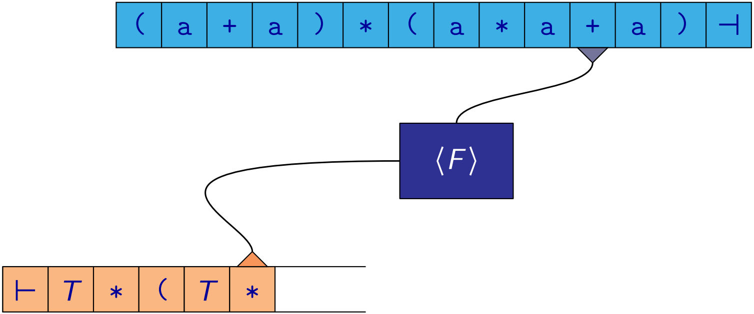
$T*(\underline{T}*a+a)\dashv \;\Rightarrow\; T*(\underline{F}*a+a)\dashv \;\Rightarrow\; \underline{T}*(a*a+a)\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



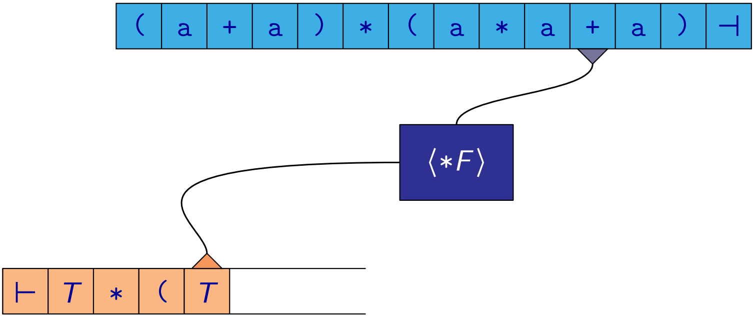
$$T*(T*\underline{F}+\mathrm{a})\dashv \;\Rightarrow\; T*(\underline{T}*\mathrm{a}+\mathrm{a})\dashv \;\Rightarrow\; T*(\underline{F}*\mathrm{a}+\mathrm{a})\dashv \;\Rightarrow\; \cdots$$

Equivalence of CFG and PDA



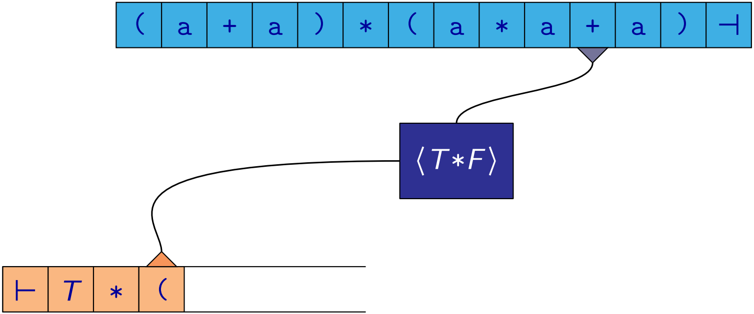
$T{*}(T{*}\underline{F}{+}a)\dashv \;\Rightarrow\; T{*}(\underline{T}{*}a{+}a)\dashv \;\Rightarrow\; T{*}(\underline{F}{*}a{+}a)\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



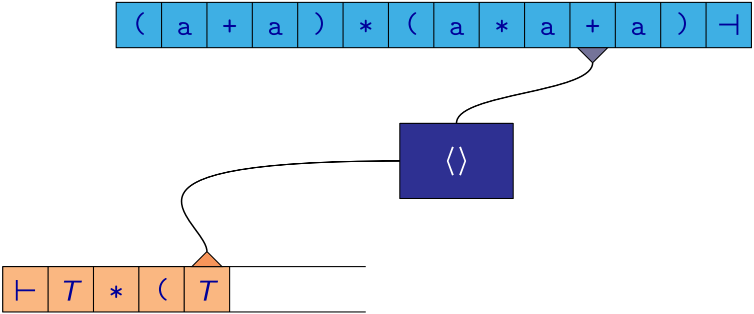
$T*(T*\underline{F}+\mathrm{a})\dashv \;\Rightarrow\; T*(\underline{T}*\mathrm{a}+\mathrm{a})\dashv \;\Rightarrow\; T*(\underline{F}*\mathrm{a}+\mathrm{a})\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



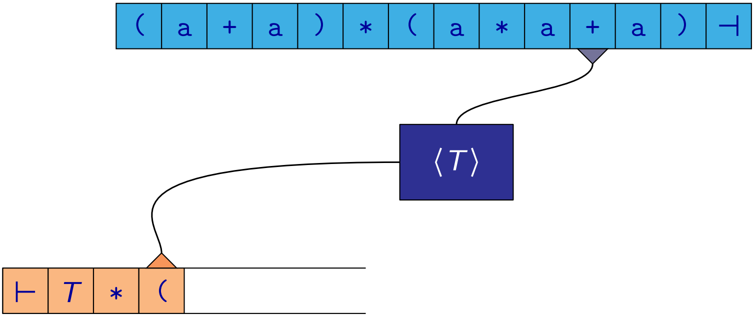
$T*(T*\underline{F}+\mathrm{a})\dashv \;\Rightarrow\; T*(\underline{T}*\mathrm{a}+\mathrm{a})\dashv \;\Rightarrow\; T*(\underline{F}*\mathrm{a}+\mathrm{a})\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



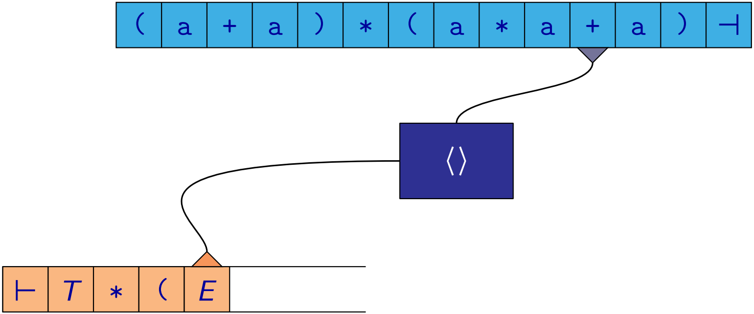
$T*(\underline{T}+\mathrm{a})\dashv \;\Rightarrow\; T*(T*\underline{F}+\mathrm{a})\dashv \;\Rightarrow\; T*(\underline{T}*\mathrm{a}+\mathrm{a})\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



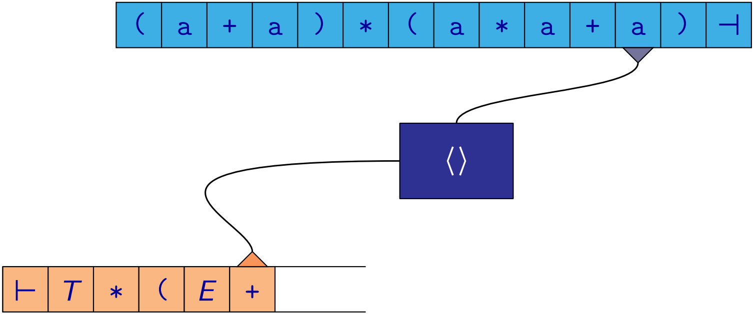
$$T*(\underline{T}+\mathrm{a})\dashv \;\Rightarrow\; T*(T*\underline{F}+\mathrm{a})\dashv \;\Rightarrow\; T*(\underline{T}*\mathrm{a}+\mathrm{a})\dashv \;\Rightarrow\; \cdots$$

Equivalence of CFG and PDA



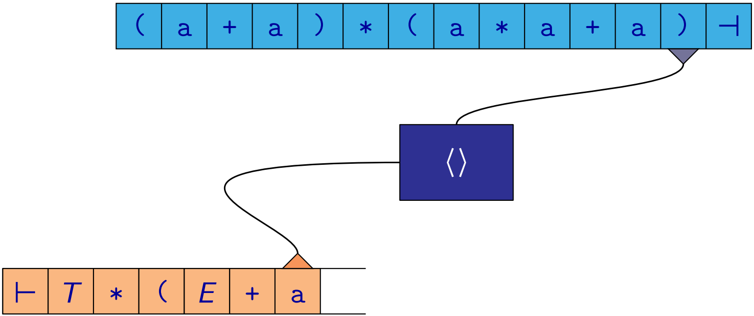
$T*(\underline{E}+\mathtt{a})\dashv \;\Rightarrow\; T*(\underline{T}+\mathtt{a})\dashv \;\Rightarrow\; T*(T*\underline{F}+\mathtt{a})\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



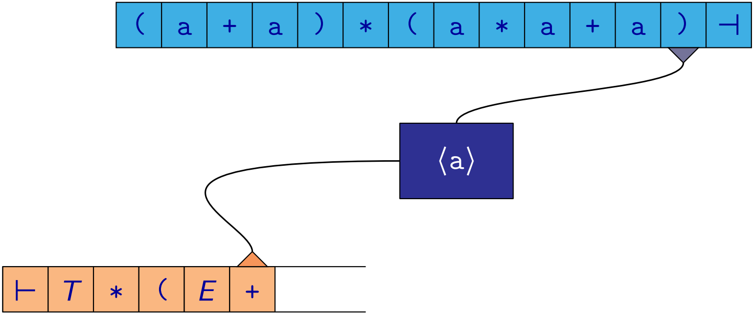
$$T*(\underline{E}+\mathtt{a})\dashv \;\Rightarrow\; T*(\underline{T}+\mathtt{a})\dashv \;\Rightarrow\; T*(T*\underline{F}+\mathtt{a})\dashv \;\Rightarrow\; \cdots$$

Equivalence of CFG and PDA



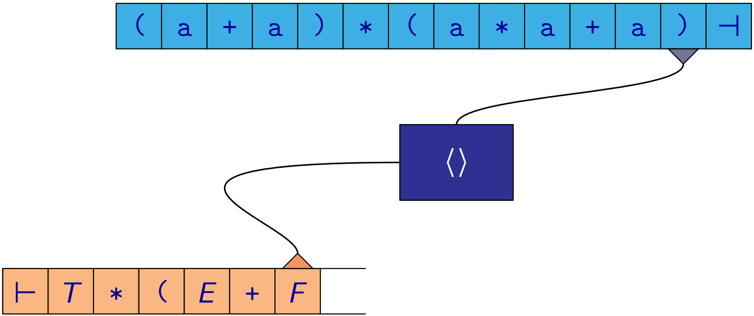
$T * (\underline{E} + \mathtt{a}) \dashv \;\Rightarrow\; T * (\underline{T} + \mathtt{a}) \dashv \;\Rightarrow\; T * (T * \underline{F} + \mathtt{a}) \dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



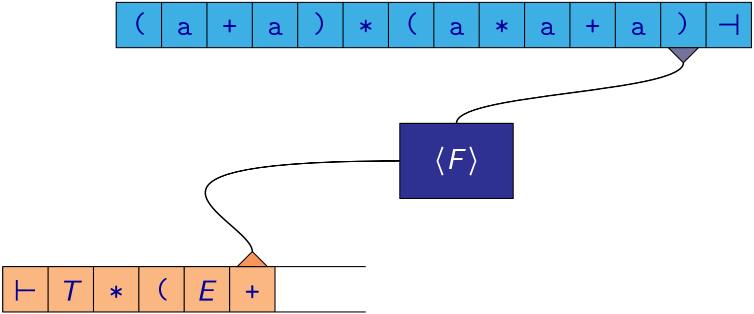
$T*(\underline{E}+\mathtt{a})\dashv \;\Rightarrow\; T*(\underline{T}+\mathtt{a})\dashv \;\Rightarrow\; T*(T*\underline{F}+\mathtt{a})\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



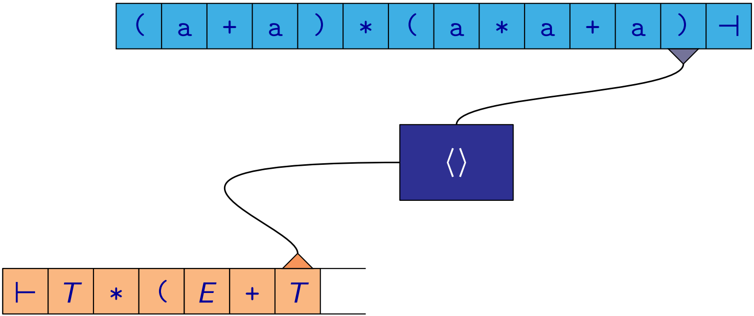
$T * (E + \underline{F}) \dashv \; \Rightarrow \; T * (\underline{E} + \mathtt{a}) \dashv \; \Rightarrow \; T * (\underline{T} + \mathtt{a}) \dashv \; \Rightarrow \; T * (T * \underline{F} + \mathtt{a}) \dashv \; \Rightarrow$
$\cdots$

Equivalence of CFG and PDA



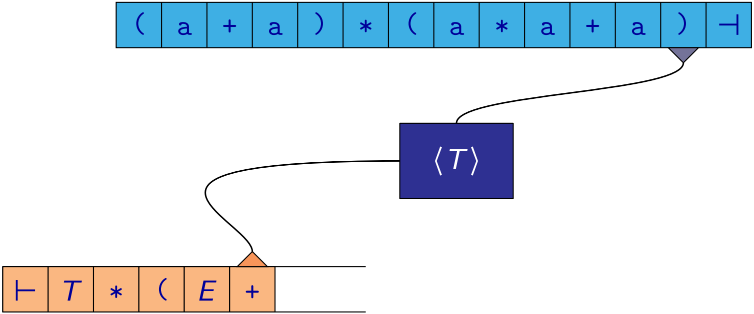
$T*(E+\underline{F})\dashv \;\Rightarrow\; T*(\underline{E}+\mathtt{a})\dashv \;\Rightarrow\; T*(\underline{T}+\mathtt{a})\dashv \;\Rightarrow\; T*(T*\underline{F}+\mathtt{a})\dashv \;\Rightarrow$
$\cdots$

Equivalence of CFG and PDA



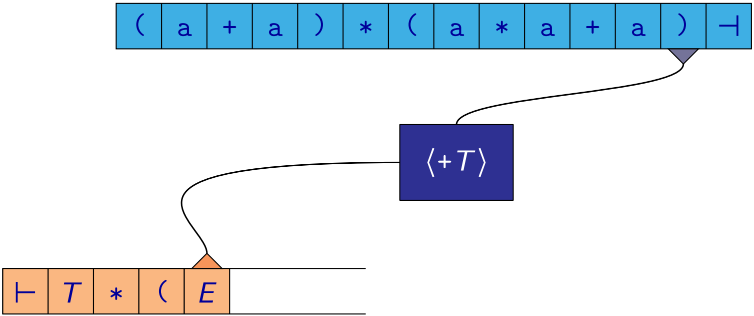
$$T*(E+\underline{T})\dashv \;\Rightarrow\; T*(E+\underline{F})\dashv \;\Rightarrow\; T*(\underline{E}+\mathtt{a})\dashv \;\Rightarrow\; T*(\underline{T}+\mathtt{a})\dashv \;\Rightarrow\; \cdots$$

Equivalence of CFG and PDA



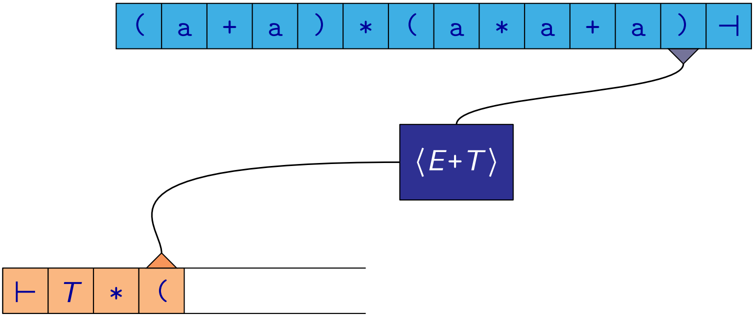
$$T*(E+\underline{T})\dashv \;\Rightarrow\; T*(E+\underline{F})\dashv \;\Rightarrow\; T*(\underline{E}+\mathrm{a})\dashv \;\Rightarrow\; T*(\underline{T}+\mathrm{a})\dashv \;\Rightarrow\; \cdots$$

Equivalence of CFG and PDA



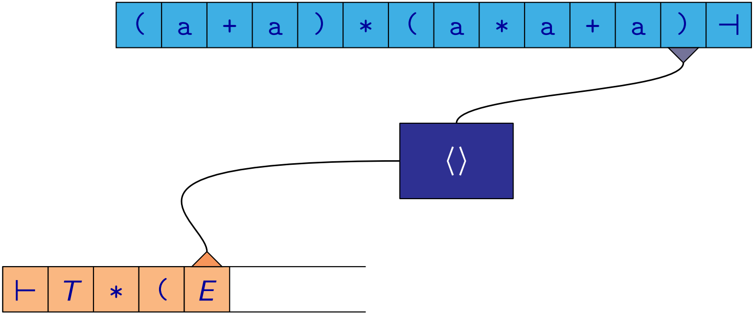
$T*(E+\underline{T})\dashv \;\Rightarrow\; T*(E+\underline{F})\dashv \;\Rightarrow\; T*(\underline{E}+\mathrm{a})\dashv \;\Rightarrow\; T*(\underline{T}+\mathrm{a})\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



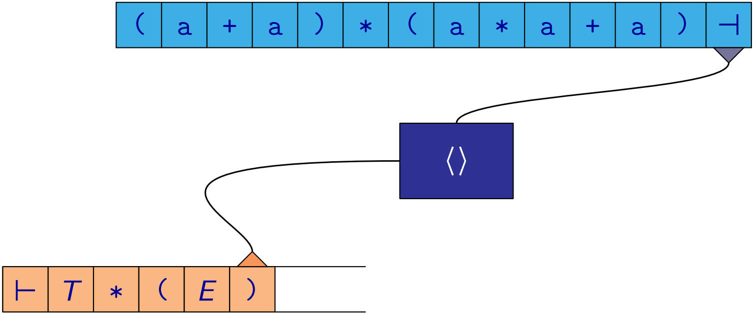
$$T*(E+\underline{T})\dashv \;\Rightarrow\; T*(E+\underline{F})\dashv \;\Rightarrow\; T*(\underline{E}+\mathtt{a})\dashv \;\Rightarrow\; T*(\underline{T}+\mathtt{a})\dashv \;\Rightarrow\; \cdots$$

Equivalence of CFG and PDA



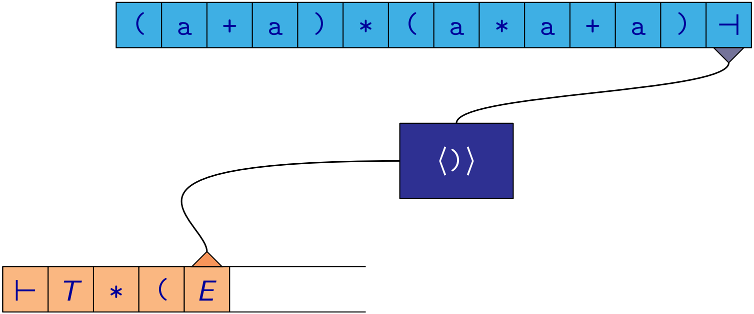
$$T*(\underline{E})\dashv \;\Rightarrow\; T*(E+\underline{T})\dashv \;\Rightarrow\; T*(E+\underline{F})\dashv \;\Rightarrow\; T*(\underline{E}+\mathtt{a})\dashv \;\Rightarrow\; \cdots$$

Equivalence of CFG and PDA



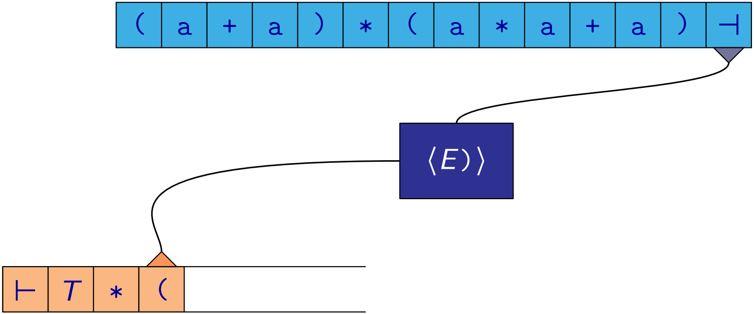
$T*(\underline{E})\dashv \;\Rightarrow\; T*(E+\underline{T})\dashv \;\Rightarrow\; T*(E+\underline{F})\dashv \;\Rightarrow\; T*(\underline{E}+\mathtt{a})\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



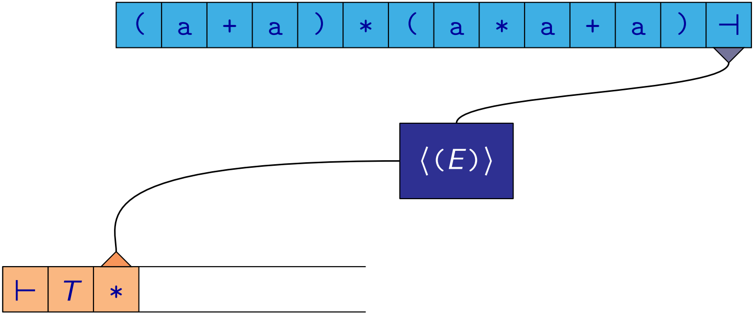
$$T*(\underline{E})\dashv \;\Rightarrow\; T*(E+\underline{T})\dashv \;\Rightarrow\; T*(E+\underline{F})\dashv \;\Rightarrow\; T*(\underline{E}+\mathrm{a})\dashv \;\Rightarrow\; \cdots$$

Equivalence of CFG and PDA



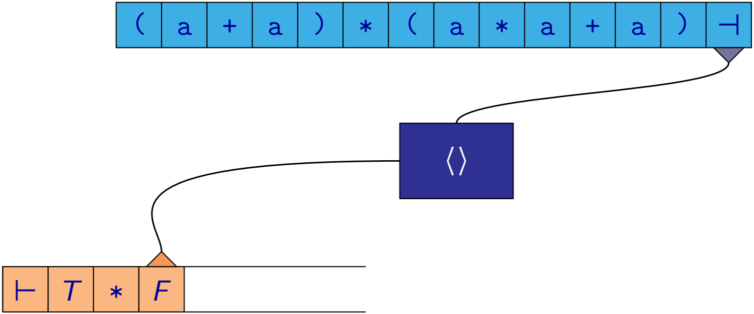
$$T*(\underline{E})\dashv \;\Rightarrow\; T*(E+\underline{T})\dashv \;\Rightarrow\; T*(E+\underline{F})\dashv \;\Rightarrow\; T*(\underline{E}+\mathrm{a})\dashv \;\Rightarrow\; \cdots$$

Equivalence of CFG and PDA



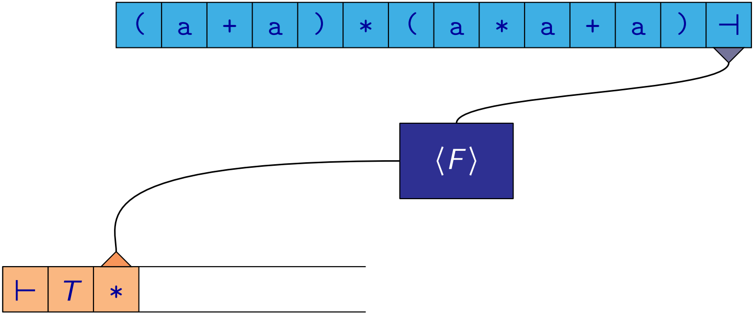
$T*(\underline{E})\dashv \;\Rightarrow\; T*(E+\underline{T})\dashv \;\Rightarrow\; T*(E+\underline{F})\dashv \;\Rightarrow\; T*(\underline{E}+\mathrm{a})\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



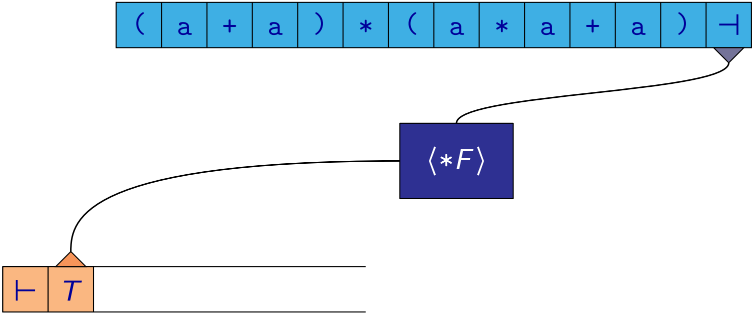
$T * \underline{F} \dashv \;\Rightarrow\; T * (\underline{E}) \dashv \;\Rightarrow\; T * (E + \underline{T}) \dashv \;\Rightarrow\; T * (E + \underline{F}) \dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



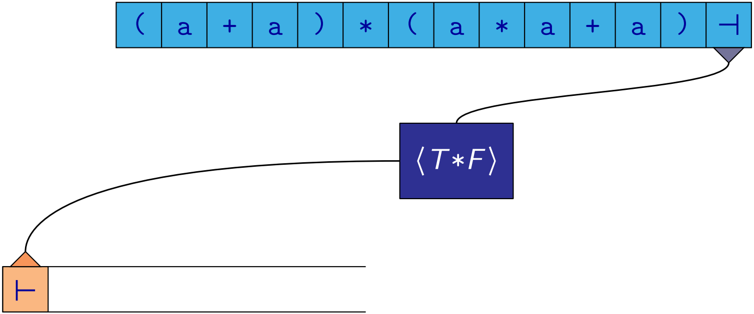
$T * \underline{F} \dashv \;\Rightarrow\; T * (\underline{E}) \dashv \;\Rightarrow\; T * (E + \underline{T}) \dashv \;\Rightarrow\; T * (E + \underline{F}) \dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



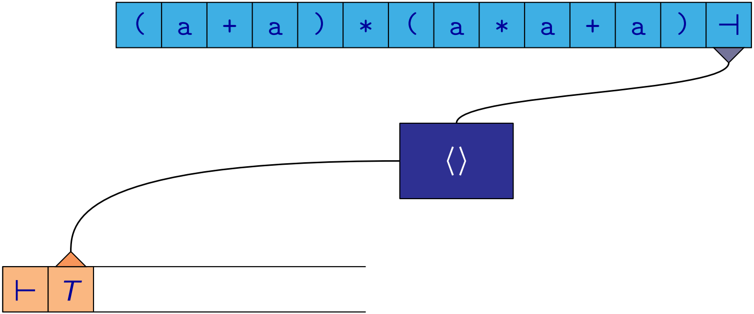
$T{*}\underline{F}\dashv \;\Rightarrow\; T{*}(\underline{E})\dashv \;\Rightarrow\; T{*}(E{+}\underline{T})\dashv \;\Rightarrow\; T{*}(E{+}\underline{F})\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



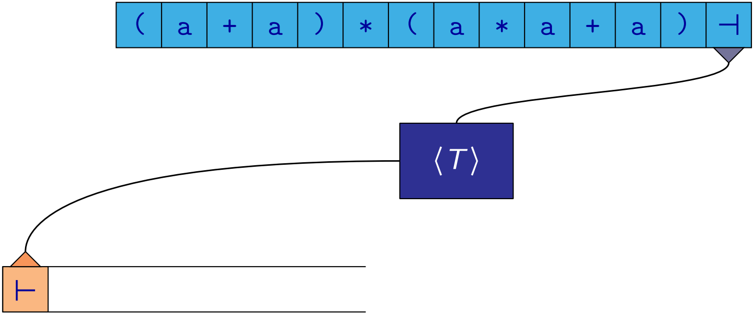
$T * \underline{F} \dashv \;\Rightarrow\; T * (\underline{E}) \dashv \;\Rightarrow\; T * (E + \underline{T}) \dashv \;\Rightarrow\; T * (E + \underline{F}) \dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



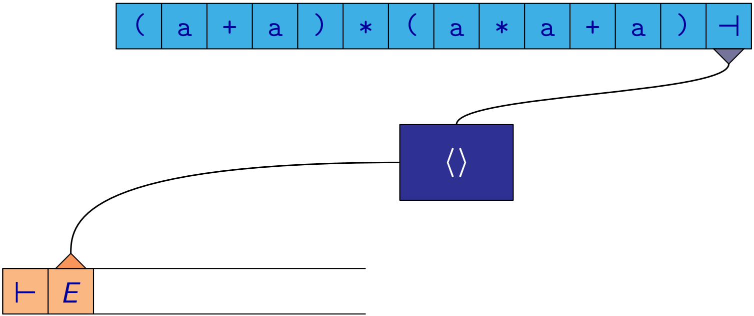
$\underline{T}\dashv \Rightarrow T*\underline{F}\dashv \Rightarrow T*(\underline{E})\dashv \Rightarrow T*(E+\underline{T})\dashv \Rightarrow T*(E+\underline{F})\dashv \Rightarrow$
$\cdots$

Equivalence of CFG and PDA



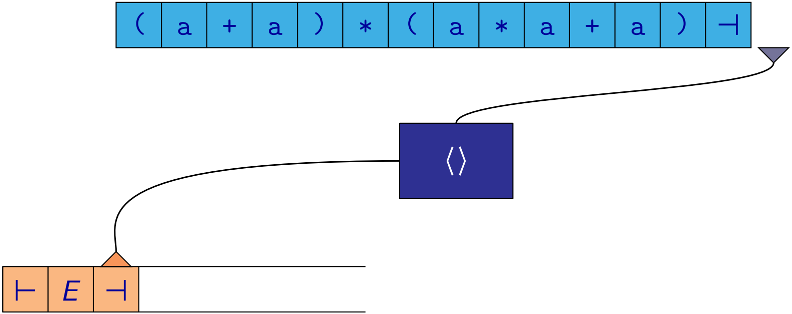
$\underline{T}\dashv \;\Rightarrow\; T{*}\underline{F}\dashv \;\Rightarrow\; T{*}(\underline{E})\dashv \;\Rightarrow\; T{*}(E{+}\underline{T})\dashv \;\Rightarrow\; T{*}(E{+}\underline{F})\dashv \;\Rightarrow$
$\ldots$

Equivalence of CFG and PDA



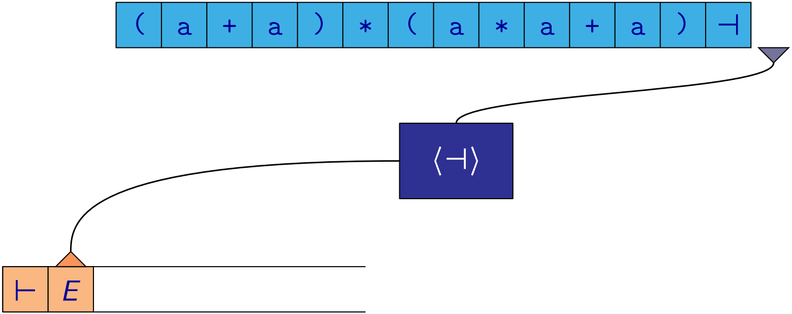
$$\underline{E}\dashv \;\Rightarrow\; \underline{T}\dashv \;\Rightarrow\; T*\underline{F}\dashv \;\Rightarrow\; T*(\underline{E})\dashv \;\Rightarrow\; T*(E+\underline{T})\dashv \;\Rightarrow\; \cdots$$

Equivalence of CFG and PDA



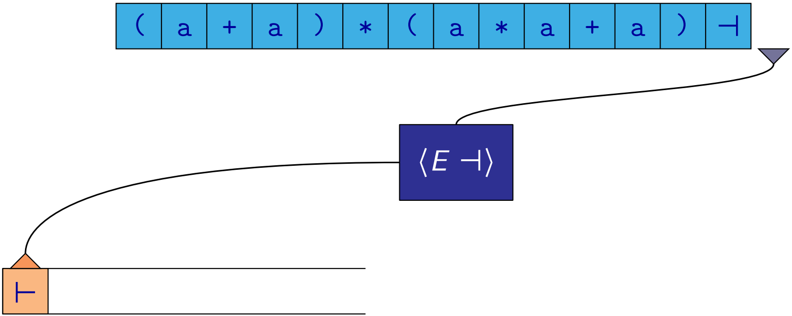
$\underline{E}\dashv \Rightarrow \underline{T}\dashv \Rightarrow T*\underline{F}\dashv \Rightarrow T*(\underline{E})\dashv \Rightarrow T*(E+\underline{T})\dashv \Rightarrow \cdots$

Equivalence of CFG and PDA



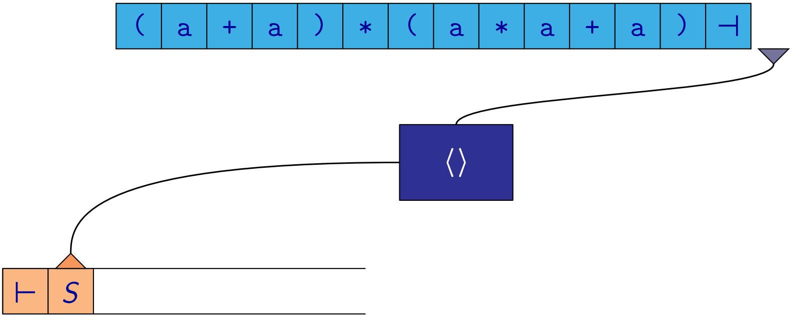
$\underline{E}\dashv \;\Rightarrow\; \underline{T}\dashv \;\Rightarrow\; T*\underline{F}\dashv \;\Rightarrow\; T*(\underline{E})\dashv \;\Rightarrow\; T*(E+\underline{T})\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



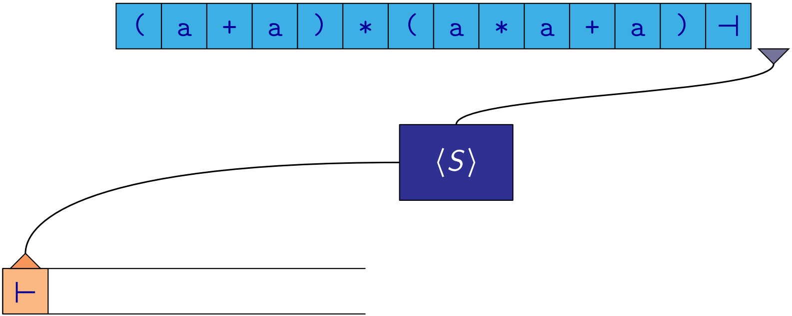
$\underline{E}\dashv \;\Rightarrow\; \underline{T}\dashv \;\Rightarrow\; T*\underline{F}\dashv \;\Rightarrow\; T*(\underline{E})\dashv \;\Rightarrow\; T*(E+\underline{T})\dashv \;\Rightarrow\; \cdots$

Equivalence of CFG and PDA



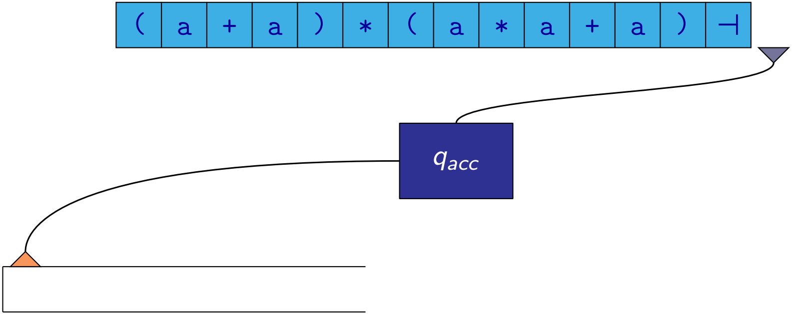
$\underline{S} \Rightarrow \underline{E}\dashv \Rightarrow \underline{T}\dashv \Rightarrow T*\underline{F}\dashv \Rightarrow T*(\underline{E})\dashv \Rightarrow T*(E+\underline{T})\dashv \Rightarrow \cdots$

Equivalence of CFG and PDA



$\underline{S} \Rightarrow \underline{E}\dashv \Rightarrow \underline{T}\dashv \Rightarrow T{*}\underline{F}\dashv \Rightarrow T{*}(\underline{E})\dashv \Rightarrow T{*}(E{+}\underline{T})\dashv \Rightarrow \cdots$

Equivalence of CFG and PDA



$\underline{S} \Rightarrow \underline{E}\dashv \Rightarrow \underline{T}\dashv \Rightarrow T{*}\underline{F}\dashv \Rightarrow T{*}(\underline{E})\dashv \Rightarrow T{*}(E{+}\underline{T})\dashv \Rightarrow \cdots$

Equivalence of CFG and PDA

As we can see from the previous example, the pushdown automaton $\mathcal{M}$ basically performs a **right derivation** in grammar $\mathcal{G}$ in reverse order.

Other Classes of Context-Free Grammars

There exist a lot of different classes of context-free grammars, for which it is possible to construct a corresponding pushdown automaton in such a way that this automaton is deterministic:

- **Top-down** approach — constructs a left derivation:
  - LL(0), LL(1), LL(2), ...
- **Bottom-up** approach — constructs a right derivation in a reverse order:
  - LR(0), LR(1), LR(2), ...
  - LALR (resp. LALR(1), ...)
  - SLR (resp. SLR(1), ...)

Parser Generators

Parser generators — tools that allow for a description of a context-free grammar to automatically generate a code in some programming language basically implementing behaviour of a corresponding pushdown automaton.

Examples of parser generators:

- Yacc
- Bison
- ANTLR
- JavaCC
- Menhir
- ...

Equivalence of CFG and PDA

Theorem

For every pushdown automaton $\mathcal{M}$ with one control state, there is a corresponding CFG $\mathcal{G}$ such $\mathcal{L}(\mathcal{G}) = \mathcal{L}(\mathcal{M})$.

Proof: For PDA $\mathcal{M} = (\{q_0\}, \Sigma, \Gamma, \delta, q_0, X_0)$, where $\Sigma \cap \Gamma = \emptyset$, we construct CFG $\mathcal{G} = (\Gamma, \Sigma, X_0, P)$, where

$$(A \to a\alpha) \in P \qquad \text{iff} \qquad (q_0, \alpha) \in \delta(q_0, a, A)$$

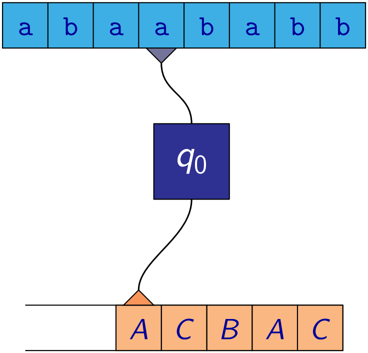
for all $A \in \Gamma$, $a \in \Sigma \cup \{\varepsilon\}$, and $\alpha \in \Gamma^*$.

It can be proved by induction that

$$X_0 \Rightarrow^* u\alpha \ \ (\text{in } \mathcal{G}) \qquad \text{iff} \qquad q_0 X_0 \xrightarrow{u} q_0\alpha \ \ (\text{in } \mathcal{M})$$

where $u \in \Sigma^*$ and $\alpha \in \Gamma^*$ (in $\mathcal{G}$, we consider only left derivations).

Equivalence of CFG and PDA



$\mathcal{M}$:

$\vdots$

$$q_0 A \xrightarrow{a} q_0 BC$$

$$q_0 B \xrightarrow{b} q_0$$

$\vdots$

$\mathcal{G}$:

$\vdots$

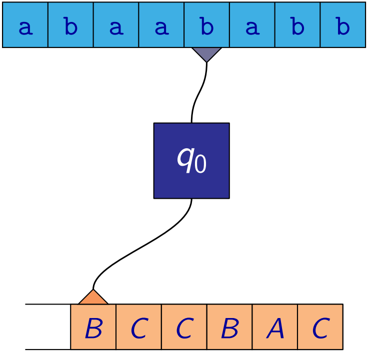
$$A \rightarrow aBC$$

$$B \rightarrow b$$

$\vdots$

$\mathtt{a\,b\,a}\ \underline{A}\ C\ B\ A\ C$

Equivalence of CFG and PDA



$\mathcal{M}$:

$\vdots$

$q_0 A \xrightarrow{a} q_0 BC$

$q_0 B \xrightarrow{b} q_0$

$\vdots$

$\mathcal{G}$:

$\vdots$

$A \rightarrow aBC$

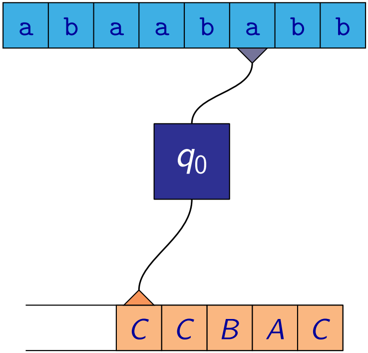
$B \rightarrow b$

$\vdots$

$\mathtt{aba}\,\underline{A}\,C\,B\,A\,C$

$\Rightarrow \mathtt{abaa}\,\underline{B}\,C\,C\,B\,A\,C$

Equivalence of CFG and PDA



$\mathcal{M}$:

$\vdots$

$q_0 A \xrightarrow{a} q_0 BC$

$q_0 B \xrightarrow{b} q_0$

$\vdots$

$\mathcal{G}$:

$\vdots$

$A \rightarrow aBC$

$B \rightarrow b$

$\vdots$

$$
\begin{aligned}
& \mathtt{aba}\,\underline{A}\,C\,B\,A\,C \\
\Longrightarrow\ & \mathtt{abaa}\,\underline{B}\,C\,C\,B\,A\,C \\
\Longrightarrow\ & \mathtt{abaab}\,\underline{C}\,C\,B\,A\,C
\end{aligned}
$$

Equivalence of CFG and PDA

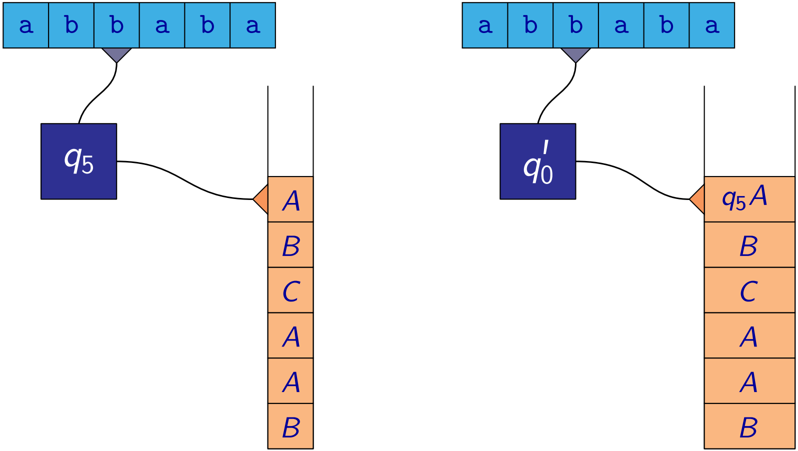
Theorem

For every pushdown automaton $\mathcal{M}$ there exists a pushdown automaton $\mathcal{M}'$ with one control state such that $\mathcal{L}(\mathcal{M}') = \mathcal{L}(\mathcal{M})$.

Proof idea:

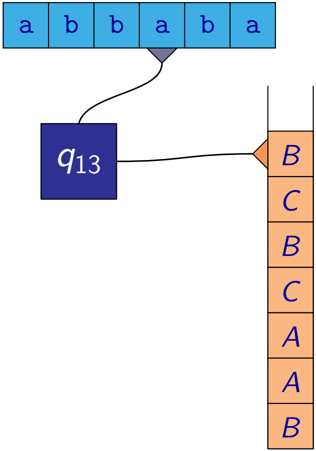
- The control state of $\mathcal{M}$ is stored on the top of the stack of $\mathcal{M}'$.
- For $\delta(q, a, X) = \{(q', \varepsilon)\}$ we must ensure that the new control state on the stack of $\mathcal{M}'$ is q'. (Other cases are straightforward.)
- Stack symbols of $\mathcal{M}'$ are triples of the form (q, A, q') where q represents the control state of $\mathcal{M}$ when that symbol is on the top, A is the stack symbol of $\mathcal{M}$, and q' is the first control state in the triple below it.
- PDA $\mathcal{M}'$ nondeterministically "guesses" the control states to which $\mathcal{M}$ goes when the given stack symbols becomes the top of the stack.

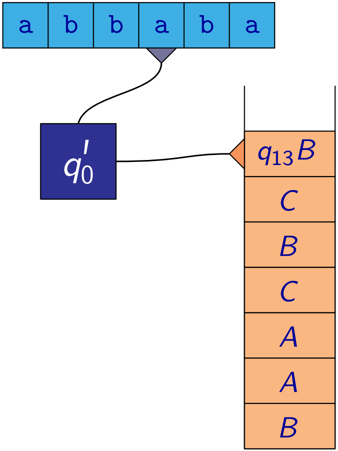
Incorrect idea:



Equivalence of CFG and PDA

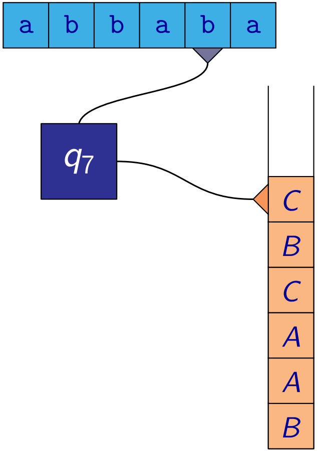
Incorrect idea:

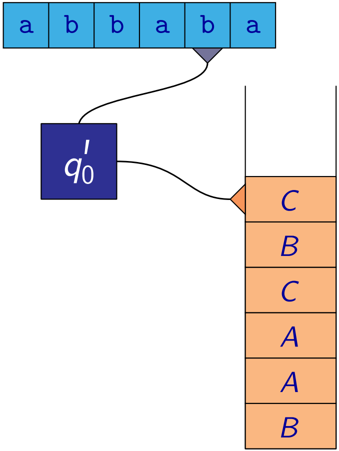




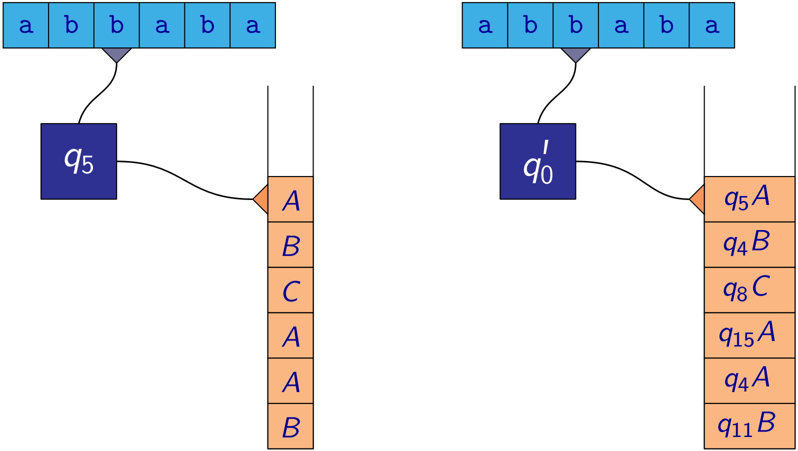
Equivalence of CFG and PDA

Incorrect idea:



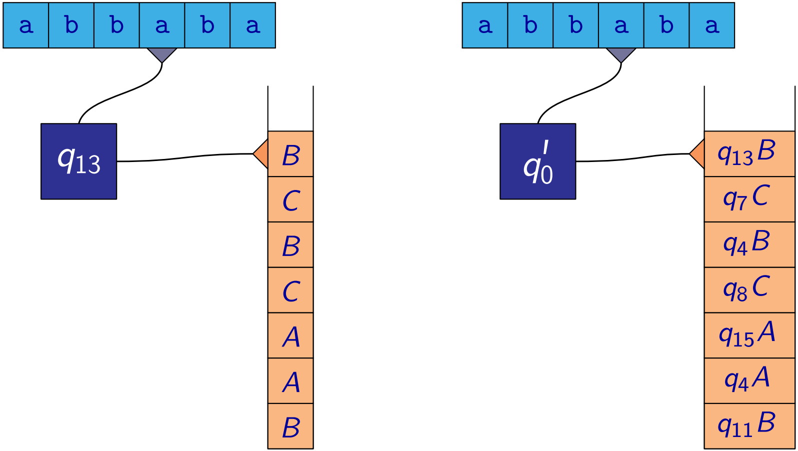


Other incorrect idea:

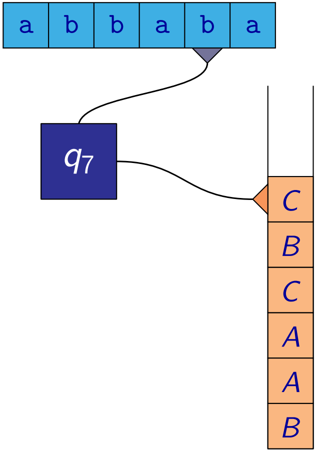


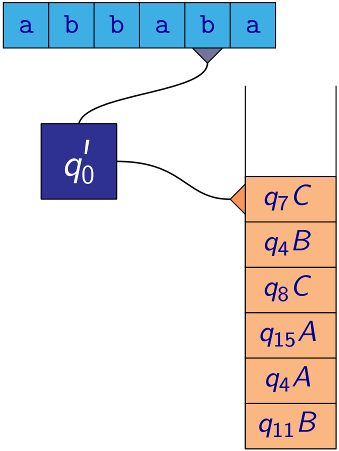
Equivalence of CFG and PDA

Other incorrect idea:



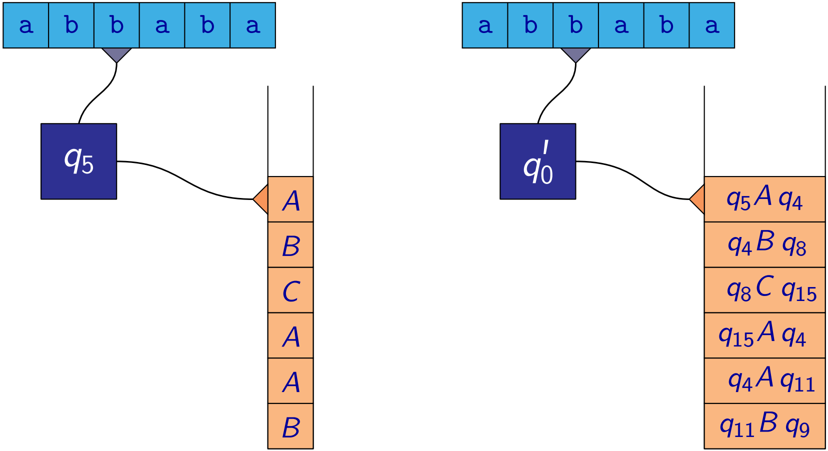
Other incorrect idea:





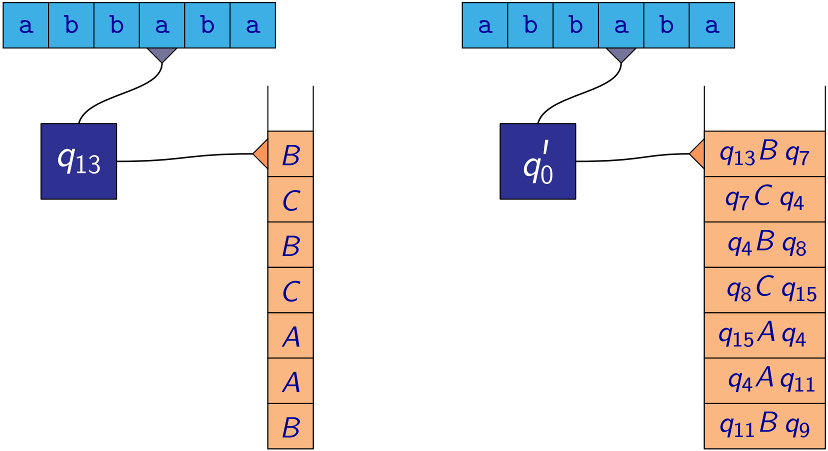
Equivalence of CFG and PDA

The correct construction:



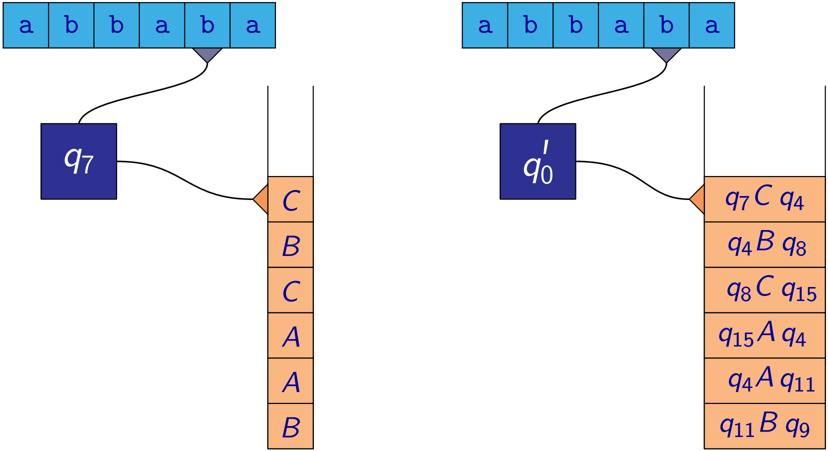
Equivalence of CFG and PDA

The correct construction:



Equivalence of CFG and PDA

The correct construction:



Equivalence of CFG and PDA

Proposition

For every context-free grammar $\mathcal{G}$ there is some (nondeterministic) pushdown automaton $\mathcal{M}$ such that $\mathcal{L}(\mathcal{G}) = \mathcal{L}(\mathcal{M})$.

Proposition

For every pushdown automaton $\mathcal{M}$ there is some context-free grammar $\mathcal{G}$ such that $\mathcal{L}(\mathcal{M}) = \mathcal{L}(\mathcal{G})$.