Automata and Grammars (BIE-AAG)

8. Pushdown automata

Jan Holub

Department of Theoretical Computer Science
Faculty of Information Technology
Czech Technical University in Prague

© Jan Holub, 2020

Pushdown automaton

Definition

Pushdown automaton is a 7-tuple $R = (Q, \Sigma, G, \delta, q_0, Z_0, F)$, where:

- Q is a finite set of states,
- Σ is a finite input alphabet,
- G is a finite pushdown store alphabet,
- δ is a mapping from a finite subset of $Q \times (\Sigma \cup \{\varepsilon\}) \times G^*$ into set of finite subsets $Q \times G^*$,
- $q_0 \in Q$ is the initial state,
- $Z_0 \in G$ is the initial pushdown store symbol,
- $F \subseteq Q$ is a set of final states.

Pushdown automaton

Example

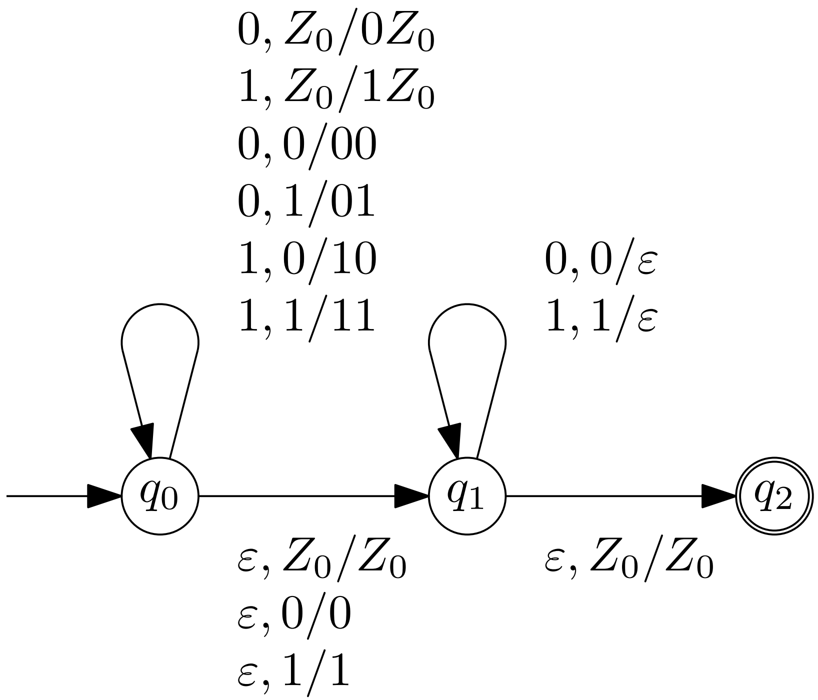
$P = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, Z_0\}, \delta, q_0, Z_0, \{q_2\})$, where δ:

$$
\begin{array}{rclcrcl}
\delta(q_0, 0, Z_0) & = & \{(q_0, 0Z_0)\} & & \delta(q_0, 1, Z_0) & = & \{(q_0, 1Z_0)\} \\
\delta(q_0, 0, 0) & = & \{(q_0, 00)\} & & \delta(q_0, 0, 1) & = & \{(q_0, 01)\} \\
\delta(q_0, 1, 0) & = & \{(q_0, 10)\} & & \delta(q_0, 1, 1) & = & \{(q_0, 11)\} \\
\delta(q_0, \varepsilon, Z_0) & = & \{(q_1, Z_0)\} & & \delta(q_0, \varepsilon, 0) & = & \{(q_1, 0)\} \\
\delta(q_0, \varepsilon, 1) & = & \{(q_1, 1)\} & & \delta(q_1, 0, 0) & = & \{(q_1, \varepsilon)\} \\
\delta(q_1, 1, 1) & = & \{(q_1, \varepsilon)\} & & \delta(q_1, \varepsilon, Z_0) & = & \{(q_2, Z_0)\}
\end{array}
$$

Pushdown automaton

Example

$P = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, Z_0\}, \delta, q_0, Z_0, \{q_2\})$, where δ:



Pushdown automaton

Configuration of PDA R: $(q, w, \alpha) \in Q \times \Sigma^* \times G^*$, where

- q is the current state,
- w is the yet unprocessed part of the input string,
- α is the pushdown store content.

The initial configuration of PDA R: $(q_0, w, Z_0), w \in \Sigma^*$

$\delta(q, a, \alpha) = \{(p_1, \gamma_1), (p_2, \gamma_2), \ldots, (p_m, \gamma_m)\}$: PDA in state q reads symbol a, moves into state $p_i, i \in \{1, 2, \ldots, m\}$, and string α on top of the pushdown store is replaced by string γ_i.

$\delta(q, \varepsilon, \alpha) = \{(p_1, \gamma_1), (p_2, \gamma_2), \ldots, (p_m, \gamma_m)\}$: transition into a new state and change of pushdown store content without reading an input symbol.

Pushdown automaton

Definition (Move of pushdown automaton)
Let $R = (Q, \Sigma, G, \delta, q_0, Z_0, F)$ be a pushown automaton. Let $\vdash_R$ is a relation over $Q \times \Sigma^* \times G^*$ (i.e., subset of $(Q \times \Sigma^* \times G^*) \times (Q \times \Sigma^* \times G^*)$) such that $(q, w, \alpha\beta) \vdash_R (p, w', \gamma\beta)$ iff $w = aw'$ and $(p, \gamma) \in \delta(q, a, \alpha)$ for some $a \in \Sigma \cup \{\varepsilon\}$, $w \in \Sigma^*, \alpha, \beta, \gamma \in G^*$. An element of relation $\vdash_R$ is called *move in pushdown automaton* R.

$\vdash^k$: k-th power of relation $\vdash$,
$\vdash^+$: transitive closure of relation $\vdash$,
$\vdash^*$: transitive and reflexive closure of relation $\vdash$

Pushdown automaton

Definition

Language defined (accepted) by PDA $R = (Q, \Sigma, G, \delta, q_0, Z_0, F)$:

1. *by transition into a final state*
   $L(R) = \{w : w \in \Sigma^*, \exists \gamma \in G^*, \exists q \in F, (q_0, w, Z_0) \vdash^* (q, \varepsilon, \gamma)\}$,
2. by *empty pushdown store*
   $L_\varepsilon(R) = \{w : w \in \Sigma^*, \exists q \in Q, (q_0, w, Z_0) \vdash^* (q, \varepsilon, \varepsilon)\}$.

Pushdown automaton

Example

PDA accepting language $L_\varepsilon(\mathrm{PDA}) = \{ww^R : w \in \{a,b\}^*\}$:

$R = (\{q,p\}, \{a,b\}, \{a,b,S,Z\}, \delta, q, Z, \emptyset)$, where

$\delta(q,a,\varepsilon) = \{(q,a)\}$,

$\delta(q,b,\varepsilon) = \{(q,b)\}$,

$\delta(q,\varepsilon,\varepsilon) = \{(q,S)\}$,

$\delta(q,\varepsilon,aSa) = \{(q,S)\}$,

$\delta(q,\varepsilon,bSb) = \{(q,S)\}$,

$\delta(q,\varepsilon,SZ) = \{(p,\varepsilon)\}$.

For input string $aabbaa$ the automaton R performs this sequence of moves:

$$
\begin{array}{lll}
\quad (q, aabbaa, Z) & \vdash (q, abbaa, aZ) & \vdash (q, bbaa, aaZ) \\
\vdash (q, baa, baaZ) & \vdash (q, baa, SbaaZ) & \vdash (q, aa, bSbaaZ) \\
\vdash (q, aa, SaaZ) & \vdash (q, a, aSaaZ) & \vdash (q, a, SaZ) \\
\vdash (q, \varepsilon, aSaZ) & \vdash (q, \varepsilon, SZ) & \vdash (p, \varepsilon, \varepsilon)
\end{array}
$$

Basic properties of PDA

Theorem

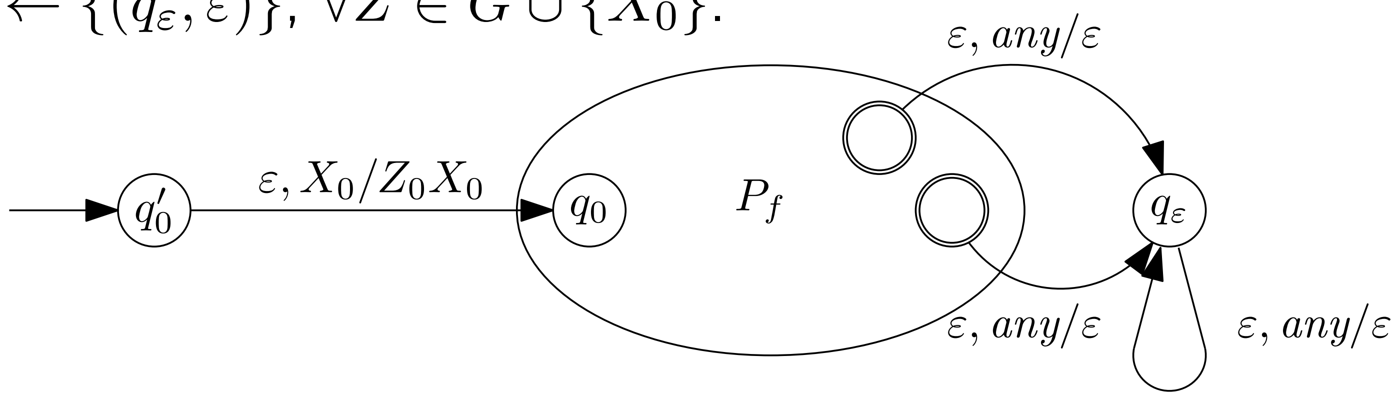
Let $P = (Q, \Sigma, G, \delta, q_0, Z_0, F)$ be a PDA. If $(q, w, A) \vdash_P^n (q', \varepsilon, \varepsilon)$, then $(q, w, A\,\alpha) \vdash_P^n (q', \varepsilon, \alpha)$, $\forall A \in G$, $\forall \alpha \in G^*$.

Basic properties of PDA

Theorem
Let L be a language. Then $\exists$ PDA P_ε accepting L by empty pushdown store iff there $\exists$ PDA P_f accepting L by transition into a final state.

Proof: First we show that $\exists P_f : L = L(P_f) \Rightarrow \exists P_\varepsilon : L = L_\varepsilon(P_\varepsilon)$.
Let $P_f = (Q, \Sigma, G, \delta, q_0, Z_0, F)$ be a PDA such that $L = L(P_f)$.
$P_\varepsilon \leftarrow (Q \cup \{q_\varepsilon, q_0'\}, \Sigma, G \cup \{X_0\}, \delta', q_0', X_0, \emptyset)$, where $\{q_\varepsilon, q_0'\} \cap Q = \emptyset$, $X_0 \notin G$, δ':

1. $\delta'(q_0', \varepsilon, X_0) \leftarrow \{(q_0, Z_0X_0)\}$,
2. $\delta'(q, a, Z) \leftarrow \delta(q, a, Z)$, $\forall q \in Q, \forall a \in \Sigma \cup \{\varepsilon\}, \forall Z \in G^*$,
3. $\delta'(q, \varepsilon, Z) \leftarrow \delta'(q, \varepsilon, Z) \cup \{(q_\varepsilon, \varepsilon)\}$, $\forall q \in F, \forall Z \in G \cup \{X_0\}$,
4. $\delta'(q_\varepsilon, \varepsilon, Z) \leftarrow \{(q_\varepsilon, \varepsilon)\}$, $\forall Z \in G \cup \{X_0\}$.

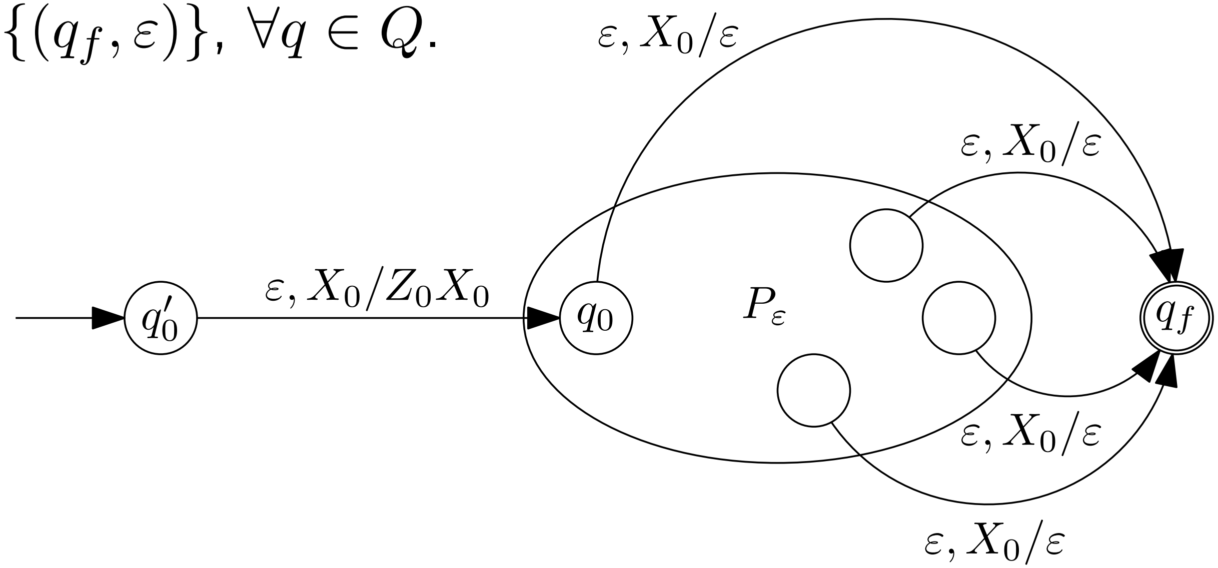


Basic properties of PDA

Theorem
Let L be a language. Then $\exists$ PDA P_ε accepting L by empty pushdown store iff there $\exists$ PDA P_f accepting L by transition into a final state.

Proof (cont.): We now show that $\exists P_\varepsilon : L = L_\varepsilon(P_\varepsilon) \Rightarrow \exists P_f : L = L(P_f)$.
Let $P_\varepsilon = (Q, \Sigma, G, \delta, q_0, Z_0, \emptyset)$ be a PDA such that $L = L_\varepsilon(P_\varepsilon)$.
$P_f \leftarrow (Q \cup \{q'_0, q_f\}, \Sigma, G \cup \{X_0\}, \delta, q'_0, X_0, \{q_f\})$, where $\{q'_0, q_f\} \cap Q = \emptyset$, $X_0 \notin G$, δ':

1. $\delta'(q'_0, \varepsilon, X_0) \leftarrow \{(q_0, Z_0X_0)\}$,
2. $\delta'(q, a, Z) \leftarrow \delta(q, a, Z)$, $\forall q \in Q, \forall a \in \Sigma \cup \{\varepsilon\}, \forall Z \in G^*$,
3. $\delta'(q, \varepsilon, X_0) \leftarrow \{(q_f, \varepsilon)\}$, $\forall q \in Q$. $\square$



Parsing

Definition (Leftmost/Rightmost derivation)
Given $G = (N, \Sigma, P, S)$. Derivation $S \Rightarrow^* \alpha, \alpha \in (N \cup \Sigma)^*$ is called *leftmost derivation* (*rightmost derivation*), if the leftmost (rightmost) nonterminal symbol in a sentential form is replaced in each step.

A leftmost (rightmost) derivation corresponds to only one parse tree and vice versa. Therefore there is a linear representation of a parse tree called *a parse*.

Parsing

Definition (Left/Rights parse of sentential form)
Given $G = (N, \Sigma, P, S)$. Let's assume the rules in P are numbered by $1, 2, \ldots, |P|$. *The parse of a sentential form* α in G is the sequence of the rule numbers used in the derivation $S \Rightarrow^* \alpha$.
The left parse of a sentential form α in G is the sequence of the rule numbers used in the leftmost derivation $S \Rightarrow^* \alpha$.
The right parse of a sentential form α in G is the reverse sequence of the rule numbers used in the rightmost derivation $S \Rightarrow^* \alpha$.

Parsing

Parsing (Syntactic analysis) = construction of parse tree

Methods of parsing:

1. *top down* (LL),
2. *bottom up* (LR).

Definition
Top-down parsing is a process of finding a left parse of a given sentence in a given grammar.

Definition
Bottom-up parsing is a process of finding a right parse of a given sentence in a given grammar.

Relationship betw. CFG and PDA

Theorem
Given CFG $G = (N, \Sigma, P, S)$ is given, we can create a PDA R such that $L(G) = L_\varepsilon(R)$.

A. Construction of PDA (model of top-down parsing):
$R = (\{q\}, \Sigma, N \cup \Sigma, \delta, q, S, \emptyset)$, where δ:

1. $\delta(q, \varepsilon, A) \leftarrow \{(q, \alpha) : (A \to \alpha) \in P\}$, $\forall A \in N$, $\quad \triangleright$ $(\textbf{expansion})$
2. $\delta(q, a, a) \leftarrow \{(q, \varepsilon)\}$, $\forall a \in \Sigma$. $\quad \triangleright$ $(\textbf{comparison})$

Top of the pushdown store for this type of automaton is **always on the left**.

Relationship betw. CFG and PDA

Example

CFG $G = (\{E, T, F\}, \{+, *, (,), \mathrm{a}\}, P, E)$, where P:

$(1)\ E \to E + T$	$(2)\ E \to T$	$(3)\ T \to T * F$
$(4)\ T \to F$	$(5)\ F \to (E)$	$(6)\ F \to \mathrm{a}.$

PDA $R = (\{q\}, \{+, *, (,), \mathrm{a}\}, \{+, *, (,), \mathrm{a}, E, T, F\}, \delta, q, E, \emptyset)$, where δ:

$$\delta(q, \varepsilon, E) = \{(q, E + T), (q, T)\}$$
$$\delta(q, \varepsilon, T) = \{(q, T * F), (q, F)\}$$
$$\delta(q, \varepsilon, F) = \{(q, (E)), (q, \mathrm{a})\}$$
$$\delta(q, b, b) = \{(q, \varepsilon)\},\ \forall b \in \{\mathrm{a}, +, *, (,)\}.$$

Relationship betw. CFG and PDA

Example (cont.)

Sentence $\mathrm{a} + \mathrm{a} * \mathrm{a}$ has the following left derivation in grammar G:

$$\begin{aligned}
E &\Rightarrow E + T && (1)\\
&\Rightarrow T + T && (2)\\
&\Rightarrow F + T && (4)\\
&\Rightarrow \mathrm{a} + T && (6)\\
&\Rightarrow \mathrm{a} + T * F && (3)\\
&\Rightarrow \mathrm{a} + F * F && (4)\\
&\Rightarrow \mathrm{a} + \mathrm{a} * F && (6)\\
&\Rightarrow \mathrm{a} + \mathrm{a} * \mathrm{a} && (6)
\end{aligned}$$

Relationship betw. CFG and PDA

Example (cont.)

$$
\begin{array}{lllrrr}
(q, \mathrm{a}+\mathrm{a}*\mathrm{a}, E) & \vdash & (q, & \mathrm{a}+\mathrm{a}*\mathrm{a}, & E+T) & (1) \\
& \vdash & (q, & \mathrm{a}+\mathrm{a}*\mathrm{a}, & T+T) & (2) \\
& \vdash & (q, & \mathrm{a}+\mathrm{a}*\mathrm{a}, & F+T) & (4) \\
& \vdash & (q, & \mathrm{a}+\mathrm{a}*\mathrm{a}, & \mathrm{a}+T) & (6) \\
& \vdash & (q, & +\mathrm{a}*\mathrm{a}, & +T) & \\
& \vdash & (q, & \mathrm{a}*\mathrm{a}, & T) & \\
& \vdash & (q, & \mathrm{a}*\mathrm{a}, & T*F) & (3) \\
& \vdash & (q, & \mathrm{a}*\mathrm{a}, & F*F) & (4) \\
& \vdash & (q, & \mathrm{a}*\mathrm{a}, & a*F) & (6) \\
& \vdash & (q, & *\mathrm{a}, & *F) & \\
& \vdash & (q, & \mathrm{a}, & F) & \\
& \vdash & (q, & \mathrm{a}, & \mathrm{a}) & (6) \\
& \vdash & (q, & \varepsilon, & \varepsilon) &
\end{array}
$$

The left parse of sentence $\mathrm{a}+\mathrm{a}*\mathrm{a}$: 1, 2, 4, 6, 3, 4, 6, 6.

Relationship betw. CFG and PDA

Theorem
Given CFG $G = (N, \Sigma, P, S)$ is given, we can create a PDA R such that $L(G) = L(R)$.

B. Construction of PDA (bottom-up parsing):
$R = (\{q, r\}, \Sigma, N \cup \Sigma \cup \{\#\}, \delta, q, \#, \{r\})$, where δ:

1. $\delta(q, a, \varepsilon) \leftarrow \{(q, a)\}, \forall a, a \in \Sigma,$ $\triangleright$ $(\textbf{shift})$
2. $\delta(q, \varepsilon, \alpha) \leftarrow \{(q, A) : (A \to \alpha) \in P\},$ $\triangleright$ $(\textbf{reduce})$
3. $\delta(q, \varepsilon, \#S) \leftarrow \{(r, \varepsilon)\}.$ $\triangleright$ $(\textbf{accept})$

Compared to the definition of the pushdown automaton and its configurations, the **top of the pushdown store** for this type of pushdown automaton is always **on the right**. $\square$

Remark
By reverting all strings concerning pushdown store we get a PDA exactly following its definition.

Relationship betw. CFG and PDA

Example

Let us have CFG $G = (\{E, T, F\}, \{+, *, (,), \mathrm{a}\}, P, E)$, where P:

$$
\begin{array}{lll}
(1)\ E \to E + T & (2)\ E \to T & (3)\ T \to T * F \\
(4)\ T \to F & (5)\ F \to (E) & (6)\ F \to \mathrm{a}.
\end{array}
$$

$R = (\{q, r\}, \{+, *, (,), \mathrm{a}\}, \{E, T, F, +, *, (,), \mathrm{a}, \#\}, \delta, q, \#, \{r\})$, where δ:

$\delta(q, b, \varepsilon) = \{(q, b)\}$, $\forall b \in \{\mathrm{a}, +, *, (,)\}$

$\delta(q, \varepsilon, E + T) = \{(q, E)\}$

$\delta(q, \varepsilon, T) = \{(q, E)\}$

$\delta(q, \varepsilon, T * F) = \{(q, T)\}$

$\delta(q, \varepsilon, F) = \{(q, T)\}$

$\delta(q, \varepsilon, (E)) = \{(q, F)\}$

$\delta(q, \varepsilon, \mathrm{a}) = \{(q, F)\}$

$\delta(q, \varepsilon, \#E) = \{(r, \varepsilon)\}$.

Relationship betw. CFG and PDA

Example (cont)

Sentence $\mathrm{a} + \mathrm{a} * \mathrm{a}$ has the following right derivation in grammar G:

$$
\begin{aligned}
E &\Rightarrow E + T && (1)\\
&\Rightarrow E + T * F && (3)\\
&\Rightarrow E + T * \mathrm{a} && (6)\\
&\Rightarrow E + F * \mathrm{a} && (4)\\
&\Rightarrow E + \mathrm{a} * \mathrm{a} && (6)\\
&\Rightarrow T + \mathrm{a} * \mathrm{a} && (2)\\
&\Rightarrow F + \mathrm{a} * \mathrm{a} && (4)\\
&\Rightarrow \mathrm{a} + \mathrm{a} * \mathrm{a} && (6)
\end{aligned}
$$

Relationship betw. CFG and PDA

Example (cont)

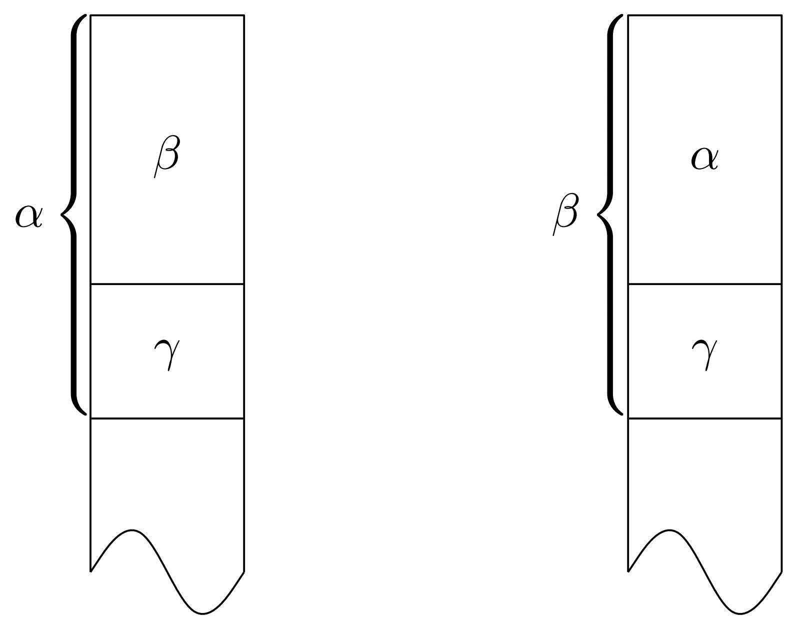
$$
\begin{array}{llrll}
(q, \mathrm{a}+\mathrm{a}*\mathrm{a}, \#) & \vdash & (q, & +\mathrm{a}*\mathrm{a}, & \#\mathrm{a} &) & \\
& \vdash & (q, & +\mathrm{a}*\mathrm{a}, & \#F &) & (6) \\
& \vdash & (q, & +\mathrm{a}*\mathrm{a}, & \#T &) & (4) \\
& \vdash & (q, & +\mathrm{a}*\mathrm{a}, & \#E &) & (2) \\
& \vdash & (q, & \mathrm{a}*\mathrm{a}, & \#E+ &) & \\
& \vdash & (q, & *\mathrm{a}, & \#E+\mathrm{a} &) & \\
& \vdash & (q, & *\mathrm{a}, & \#E+F &) & (6) \\
& \vdash & (q, & *\mathrm{a}, & \#E+T &) & (4) \\
& \vdash & (q, & \mathrm{a}, & \#E+T* &) & \\
& \vdash & (q, & \varepsilon, & \#E+T*\mathrm{a} &) & \\
& \vdash & (q, & \varepsilon, & \#E+T*F &) & (6) \\
& \vdash & (q, & \varepsilon, & \#E+T &) & (3) \\
& \vdash & (q, & \varepsilon, & \#E &) & (1) \\
& \vdash & (r, & \varepsilon, & \varepsilon &) &
\end{array}
$$

The right parse of sentence $\mathrm{a}+\mathrm{a}*\mathrm{a}$: 6, 4, 2, 6, 4, 6, 3, 1.

Deterministic PDA

Definition (Deterministic PDA)
Pushdown automaton $R = (Q, \Sigma, G, \delta, q_0, Z_0, F)$ is *deterministic*, if:

1. $|\delta(q, a, \gamma)| \leq 1$, $\forall q \in Q, \forall a \in (\Sigma \cup \{\varepsilon\}), \forall \gamma \in G^*$.
2. If $\delta(q, a, \alpha) \neq \emptyset$, $\delta(q, a, \beta) \neq \emptyset$ and $\alpha \neq \beta$, then α is not a prefix of β and β is not a prefix of α (i.e., $\alpha\gamma \neq \beta, \alpha \neq \beta\gamma, \gamma \in G^*$).
3. If $\delta(q, a, \alpha) \neq \emptyset$, $\delta(q, \varepsilon, \beta) \neq \emptyset$, then α is not a prefix of β and β is not a prefix of α (i.e., $\alpha\gamma \neq \beta, \alpha \neq \beta\gamma, \gamma \in G^*$).



Deterministic PDA

Construction of deterministic PDA by the top-down method (**A**):

Input: CFG $G = (N, \Sigma, P, S)$, where all rules are of form $A \to a\alpha$, $a \in \Sigma, \alpha \in (N \cup \Sigma)^*$ and for each two different rules $\{A \to a\alpha$, $A \to b\beta\} \subset P$ it holds that $a \neq b$.

$R \leftarrow (\{q\}, \Sigma, N \cup \Sigma, \delta, q, S, \emptyset)$, where
$\delta(q, a, A) \leftarrow \{(q, \alpha) : (A \to a\alpha) \in P\}$, $\forall A \in N$,
$\delta(q, a, a) \leftarrow \{(q, \varepsilon)\}$, $\forall a \in \Sigma$.

Deterministic PDA

Construction of deterministic PDA by a bottom-up method (**B**):

Example

CFG $G = (\{S, A, B\}, \{a, b, c, d\}, P, S)$, where P:

$S \rightarrow Aa \qquad A \rightarrow Bb \mid c \qquad B \rightarrow d$

$R = (\{q, r\}, \{a, b, c, d\}, \{S, A, B, a, b, c, d, \#\}, \delta, q, \#, \{r\})$, where δ:

$$
\begin{array}{ll}
(1) & \delta(q, a, \varepsilon) = (q, a) \\
 & \delta(q, b, \varepsilon) = (q, b) \\
 & \delta(q, c, \varepsilon) = (q, c) \\
 & \delta(q, d, \varepsilon) = (q, d) \\
(2) & \delta(q, \varepsilon, Aa) = (q, S) \\
 & \delta(q, \varepsilon, Bb) = (q, A) \\
 & \delta(q, \varepsilon, c) = (q, A) \\
 & \delta(q, \varepsilon, d) = (q, B) \\
(3) & \delta(q, \varepsilon, \#S) = (r, \varepsilon)
\end{array}
$$

Deterministic PDA

Example (cont.)

PDA is nondeterministic due to shifts by (1). These shifts can be made depending on the contents of the pushdown store:

$(1)'$ $\delta(q, a, A) = \{(q, Aa)\}$ – symbol a is present in the sentential form only after symbol A,

$\delta(q, b, B) = \{(q, Bb)\}$ – symbol b is present in the sentential form only after symbol B,

$\delta(q, c, \#) = \{(q, \#c)\}$,

$\delta(q, d, \#) = \{(q, \#d)\}$ – symbols c, d can be present only at the beginning of the sentential form.