Definition of push-down automaton

A **push-down automaton**, or PDA for short, is a tuple

$$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F),$$

with

- Q finite set of **states**
- Σ finite **input alphabet**
- Γ finite **stack alphabet**
- $\delta : Q \times \Sigma \cup \{\epsilon\} \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$ is a **transition** function, always using **finite** subsets of $2^{Q \times \Gamma^*}$
- $q_0 \in Q$ is the initial state
- $Z_0 \in \Gamma$ is the initial stack symbol

  with no symbol in the stack δ is undefined
- $F \subseteq Q$ is the set of final states

Example

The PDA for L_{wwr} is defined as

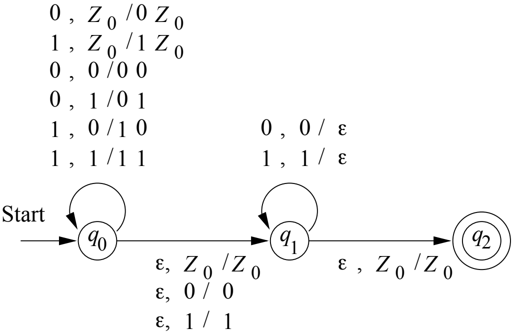
$$P = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, Z_0\}, \delta, q_0, Z_0, \{q_2\}),$$

where δ is specified by the following transition table (omitting curley brackets; stack represented as string with top at the left)

	$0, Z_0$	$1, Z_0$	0,0	0,1	1,0	1,1	ϵ, Z_0	$\epsilon, 0$	$\epsilon, 1$
$\rightarrow q_0$	$q_0, 0Z_0$	$q_0, 1Z_0$	$q_0, 00$	$q_0, 01$	$q_0, 10$	$q_0, 11$	q_1, Z_0	$q_1, 0$	$q_1, 1$
q_1			q_1, ϵ			q_1, ϵ	q_2, Z_0		
$\star q_2$									

Example

The transition function δ can also be represented in **graphical** notation, using the convention that $(p, \alpha) \in \delta(q, a, X)$ is associated with an arc from state q to state p with label $a, X/\alpha$



Instantaneous description

Informally, a **computation** of a PDA is a sequence of "configurations" of the automaton obtained one from the other by consuming an input symbol or else by reading ϵ

In order to formalize the configuration of a PDA we introduce the mathematical notion of **instantaneous description**

To formalize the computation of a PDA we then introduce a binary relation over instantaneous descriptions called **moves**

Instantaneous description

An **instantaneous description**, or ID for short, is a triple

$$(q, w, \gamma)$$

where

- q is the current state
- w is the part of the input still to be read
- γ is the stack content, with **topmost symbol** at the left

In this lecture, we will interchangeably use terms instantaneous description and configuration

Computation

Let $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA. We define a binary relation over the set of IDs called **moves**, written $\underset{P}{\vdash}$ or also $\vdash$

$\forall w \in \Sigma^*,\ \beta \in \Gamma^*$:

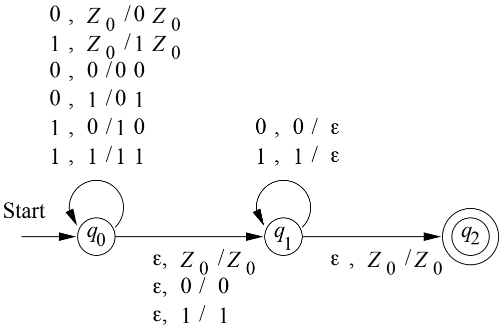
$$(p, \alpha) \in \delta(q, a, X) \Rightarrow (q, aw, X\beta) \vdash (p, w, \alpha\beta)$$
$$(p, \alpha) \in \delta(q, \epsilon, X) \Rightarrow (q, w, X\beta) \vdash (p, w, \alpha\beta)$$

We define $\overset{*}{\underset{P}{\vdash}}$ as the reflexive and transitive closure of $\underset{P}{\vdash}$. We use $\overset{*}{\underset{P}{\vdash}}$ to define a **computation** of a PDA

Compare the above with the two relations rewrite and derivation for a CFG

Example

Given our PDA for L_{wwr}



describe the computation of the automaton for the input 1111

Example

The PDA nondeterministically performs the following computations

$$
\begin{tikzcd}
(q_0, 1111, Z_0) \arrow[d] \arrow[dr] & & \\
(q_0, 111, 1Z_0) \arrow[d] \arrow[dr] & (q_1, 1111, Z_0) \arrow[r] & (q_2, 1111, Z_0) \\
(q_0, 11, 11Z_0) \arrow[d] \arrow[dr] & (q_1, 111, 1Z_0) \arrow[r] & (q_1, 11, Z_0) \arrow[d] \\
(q_0, 1, 111Z_0) \arrow[d] \arrow[dr] & (q_1, 11, 11Z_0) \arrow[dr] & (q_2, 11, Z_0) \\
(q_0, \varepsilon, 1111Z_0) \arrow[d] & (q_1, 1, 111Z_0) \arrow[d] & (q_1, 1, 1Z_0) \arrow[d] \\
(q_1, \varepsilon, 1111Z_0) & (q_1, \varepsilon, 11Z_0) & (q_1, \varepsilon, Z_0) \arrow[d] \\
 & & (q_2, \varepsilon, Z_0)
\end{tikzcd}
$$

Notational conventions for PDAs

We use the following notational conventions

- a, b, c, ..., a_1, a_2, ..., a_i, ... symbols from the input alphabet
- p, q, r, ..., q_1, q_2, ..., q_i, ... states of the automaton
- u, w, x, y, z input strings
- X, Y, Z stack symbols
- α, β, γ, ... stack contents (strings of stack symbols)

Properties of computations

Intuitively, stack or input symbols that are not read/consumed by the PDA **do not affect** the computation :

- if an ID sequence is **valid** (relation $\vdash$), then so is the sequence obtained by adding any string to the tail of the input
- if an ID sequence is valid, then so is the sequence obtained by adding any string to the bottom of the stack
- if an ID sequence is valid and some tail of the input is not consumed, then so is the sequence obtained by removing that tail in every ID in the sequence

Properties of computations

Theorem $\forall w \in \Sigma^*, \gamma \in \Gamma^*$:

$$(q, x, \alpha) \overset{*}{\vdash} (p, y, \beta) \Rightarrow (q, xw, \alpha\gamma) \overset{*}{\vdash} (p, yw, \beta\gamma)$$

Note :

- if $\gamma = \epsilon$ we get property 1, and if $w = \epsilon$ we get property 2 from previous slide
- the inverse of the above theorem does not hold: γ can be used in the computation and 'reconstructed' afterward

Theorem $\forall w \in \Sigma^*$:

$$(q, xw, \alpha) \overset{*}{\vdash} (p, yw, \beta) \Rightarrow (q, x, \alpha) \overset{*}{\vdash} (p, y, \beta)$$

Acceptance by final state

Let $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA

The **language accepted by final state** by P is

$$L(P) = \{w \mid (q_0, w, Z_0) \overset{*}{\vdash} (q, \epsilon, \alpha),\ q \in F\}$$

Note :

- The stack does not necessarily need to be empty at the end of the computation
- The PDA cannot test the end of the string: this is an external condition in the definition of $L(P)$

Acceptance by empty stack

Let $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be some PDA. The **language accepted by empty stack** by P is

$$N(P) = \{w \mid (q_0, w, Z_0) \overset{*}{\vdash} (q, \epsilon, \epsilon)\}$$

for any state q

Note : Since final states are no longer relevant in this case, set F is **not used** in the definition

From empty stack to final state

Theorem If $L = N(P_N)$ for some PDA $P_N = (Q, \Sigma, \Gamma, \delta_N, q_0, Z_0)$, then there exists a PDA P_F such that $L = L(P_F)$

Proof Let

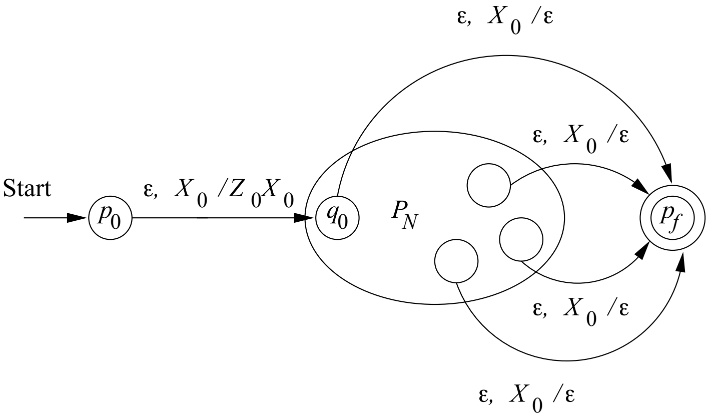
$$P_F = (Q \cup \{p_0, p_f\}, \Sigma, \Gamma \cup \{X_0\}, \delta_F, p_0, X_0, \{p_f\})$$

where

- $\delta_F(p_0, \epsilon, X_0) = \{(q_0, Z_0X_0)\}$
- for each $q \in Q$, $a \in \Sigma \cup \{\epsilon\}$, $Y \in \Gamma$ we let $\delta_F(q, a, Y) = \delta_N(q, a, Y)$
- for each $q \in Q$ we let $(p_f, \epsilon) \in \delta_F(q, \epsilon, X_0)$

From empty stack to final state

Graphical representation of PDA P_F such that $L = L(P_F)$



From empty stack to final state

We need to prove $L(P_F) = N(P_N)$

(part $\supseteq$) Let $w \in N(P_N)$. Then

$$(q_0, w, Z_0) \overset{*}{\underset{N}{\vdash}} (q, \epsilon, \epsilon),$$

for some q. From a previous theorem

$$(q_0, w, Z_0 X_0) \overset{*}{\underset{N}{\vdash}} (q, \epsilon, X_0)$$

Since $\delta_N \subset \delta_F$, we have

$$(q_0, w, Z_0 X_0) \overset{*}{\underset{F}{\vdash}} (q, \epsilon, X_0)$$

From empty stack to final state

We thus conclude

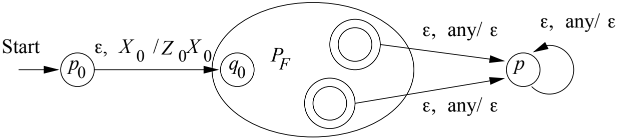
$$(p_0, w, X_0) \underset{F}{\vdash} (q_0, w, Z_0X_0) \overset{*}{\underset{F}{\vdash}} (q, \epsilon, X_0) \underset{F}{\vdash} (p_f, \epsilon, \epsilon)$$

(part $\subseteq$) By inspecting P_F diagram, any accepting computation for w in P_F embeds an accepting computation for w in P_N $\square$

From final state to empty stack

Theorem Let $L = L(P_F)$ for some PDA $P_F = (Q, \Sigma, \Gamma, \delta_F, q_0, Z_0, F)$. There exists a PDA P_N such that $L = N(P_N)$

Construction diagram for P_N from P_F



From final state to empty stack

Proof Let

$$P_N = (Q \cup \{p_0, p\}, \Sigma, \Gamma \cup \{X_0\}, \delta_N, p_0, X_0)$$

where

- $\delta_N(p_0, \epsilon, X_0) = \{(q_0, Z_0X_0)\}$
- $\delta_N(q, a, Y) = \delta_F(q, a, Y)$ for each $q \in Q$, $a \in \Sigma \cup \{\epsilon\}$, $Y \in \Gamma$
- $(p, \epsilon) \in \delta_N(q, \epsilon, Y)$, for each $q \in F$, $Y \in \Gamma \cup \{X_0\}$
- $\delta_N(p, \epsilon, Y) = \{(p, \epsilon)\}$, for each $Y \in \Gamma \cup \{X_0\}$

From final state to empty stack

We now prove $N(P_N) = L(P_F)$

(part $\subseteq$) By inspecting P_N diagram, any accepting computation for w in P_N embeds an accepting computation for w in P_F

(part $\supseteq$) Let $w \in L(P_F)$. Then

$$(q_0, w, Z_0) \overset{*}{\underset{F}{\vdash}} (q, \epsilon, \alpha)$$

for some $q \in F, \alpha \in \Gamma^*$

From final state to empty stack

Since $\delta_F \subseteq \delta_N$, and from a previous theorem stating that X_0 can be added to the bottom of the stack, we have

$$(q_0, w, Z_0 X_0) \overset{*}{\underset{N}{\vdash}} (q, \epsilon, \alpha X_0)$$

Then P_N can compute

$$(p_0, w, X_0) \underset{N}{\vdash} (q_0, w, Z_0 X_0) \overset{*}{\underset{N}{\vdash}} (q, \epsilon, \alpha X_0) \overset{*}{\underset{N}{\vdash}} (p, \epsilon, \epsilon)$$

□