# CS310 : Automata Theory 2019

## Lecture 16: Pushdown automata

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2019-02-10

# Back to automata

We have explored a strictly bigger(why?) class than regular languages, namely CFLs.

Now we look at a class of automata that will recognize the languages.

# Finite to infinite memory

All the automata seen so far had finite number of states.

Now we will add infinite memory, but strict limits on access patterns.

- the infinite memory is accessible like a stack
- only top element can be read
- elements can be pushed or popped

The stack gives significant extra power to the automata.

Exercise 16.1

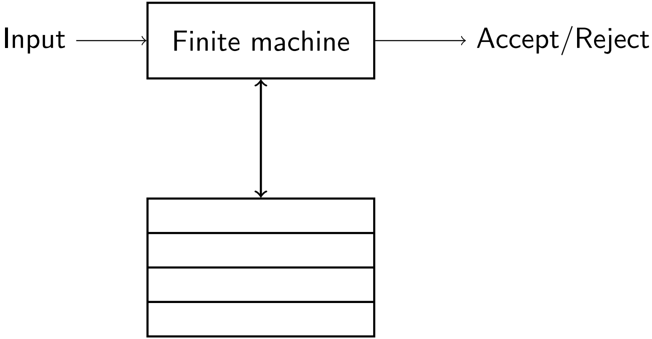
*a. What kinds of memories a process have in modern computers?*

*b. What are their relative performances?*

**Commentary:** It is unrealistic to expect random access to infinite memory, however the above limitation can still be further relaxed.

# Adding stacks to the automata

- Our usual states are still there along with the new stack
- At each transition we read the top stack symbol along with the input symbol and take actions
- Along with jumping states, we can push or pop symbols on the stack

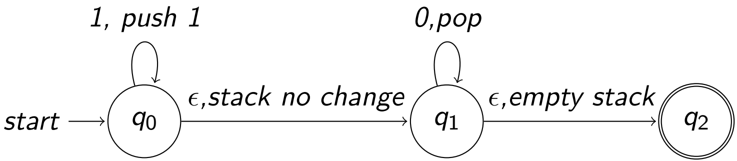


# Example : an automaton with stack

## Example 16.1

*Consider language* $\{1^n0^n | n \geq 0\}$, *which was proven to be not regular.*

*Let us use stack to recognize the language.*



# Stack symbols

We may need to push and pop symbols from the stack.

We define a new set of symbols, denoted $\Gamma$, that may be pushed in the stack.

There is a special symbol $Z_0 \in \Gamma$ that is always present at the bottom of the stack at the start.

$Z_0$ marks bottom of the stack.

# Pushdown automaton (PDA)

**Definition 16.1**
*A pushdown automaton(PDA) P is a seven-tuple*

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

*where*

- $Q$ *is a finite set of states,*
- $\Sigma$ *is a finite set of input symbols,*
- $\Gamma$ *is a finite set of stack symbols,*
- $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow \mathfrak{p}(Q \times \Gamma^*)$ *is a transition function,*
- $q_0 \in Q$ *is the start/initial state,*
- $Z_0 \in \Gamma$ *is the start stack symbol, and*
- $F \subseteq Q$ *is a set of accepting states.*

Nondeterministic

**Exercise 16.2**
*a. Can* $\Gamma$ *be empty?* *b. How many inputs* $\delta$ *has?*

# Transitions of pushdown automaton

$$\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow \mathfrak{p}(Q \times \Gamma^*)$$

Inputs:

1. current state
2. input symbol or $\epsilon$
3. top symbol on the stack

Output: The set of possible

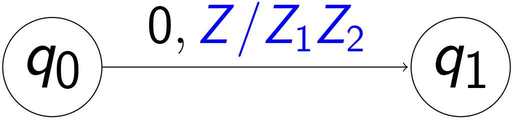
- next states and
- stack symbols word that will replace the top symbol

**Commentary:** Only $Z_0$ is present in the stack at the start. The symbol is used to mark the empty stack. We defined $Z_0$ only because we can give a clean definition of $\delta$

# Transitions in pushdown automaton

Let $\Sigma = \{0, 1\}$, $\Gamma = \{Z_0, Z, Z_1, Z_2, ...\}$ and $\delta(q_0, 0, Z) \ni (q_1, Z_1 Z_2)$

We draw the PDA transitions as follows




- To execute the transition, top symbol of the stack must be $Z$
- After transition $Z$ will be popped and word $Z_1Z_2$ will be pushed

# Understanding stack interaction due to $\delta$

- Popping $Z$ in stack :

$$\delta(q_1, a, Z) \ni (q_2, \epsilon)$$

- Pushing $X$ in stack :

$$\delta(q_1, a, Z) \ni (q_2, XZ)$$

Exercise 16.3
*Give a PDA transition such that the stack content remains unchanged.*

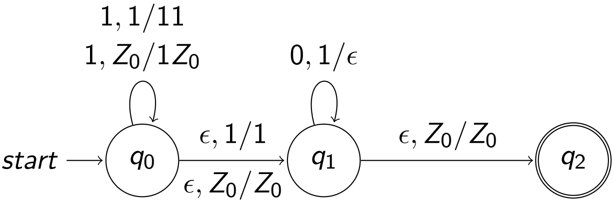
Exercise 16.4
*Can PDA move with empty stack?*

# Example: PDA

## Example 16.2

*The following is the formal definition of a PDA recognizing* $\{1^n0^n | n \geq 0\}$.

$A = (\{q_0, q_1, q_2\}, \{0, 1\}, \{Z_0, 1\}, \delta, q_0, Z_0, \{q_2\})$, *where* $\delta$ *is illustrated as follows*




## Exercise 16.5

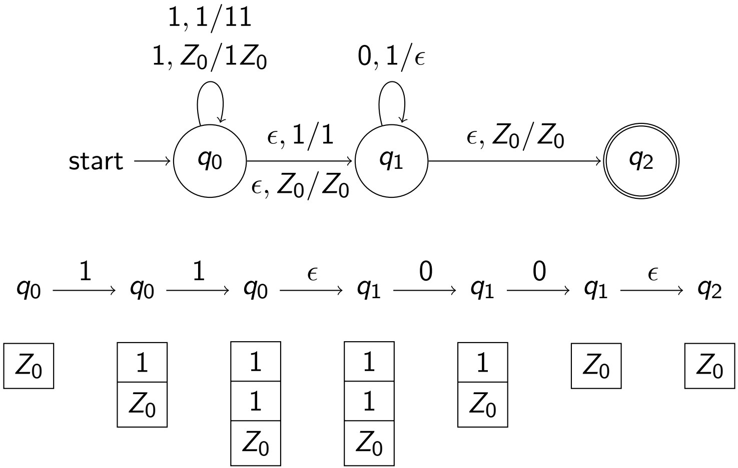
*What is the value of the transition function?*

- $\delta(q_0, 0, 1) =$
- $\delta(q_0, 1, Z_0) =$
- $\delta(q_1, \epsilon, Z_0) =$
- $\delta(q_1, 0, 1) =$

# Running PDA

## Example 16.3

*Consider the PDA again*



# Instantaneous description

We define something like extended transitions, but more suited for PDA.

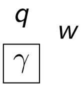
## Definition 16.2

*For a PDA, an instantaneous description(ID) is a triple*

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

*where*

- $q$ *is a current state,*
- $w$ *is the word remaining to be processed, and*
- $\gamma$ *is the current stack.*




## Example 16.4

$(q_0, 10, Z_0)$ *is an ID in our running example.*

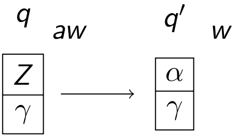
# Instantaneous description II

## Definition 16.3

*For a PDA $P = (Q, \Sigma, \Gamma, \delta, Z_0, z_0, F)$, let us define $\vdash_P$ move of PDA $P$ over IDs as follows*

$$(q, aw, Z\gamma) \vdash_P (q', w, \alpha\gamma)$$

*if $(q', \alpha) \in \delta(q, a, Z)$. $a \in \Sigma \cup \{\epsilon\}$.*



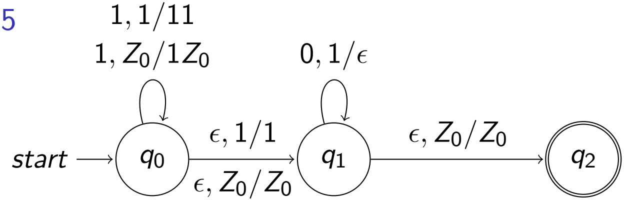

*We write $\vdash_P$ as $\vdash$ only if $P$ is understood.*

# Instantaneous description III

## Definition 16.4

*$(q_0, w_0, \gamma_1) \vdash^* (q_n, w_n, \gamma_n)$ if there are IDs $(q_0, w_0, \gamma_1), \dots, (q_n, w_n, \gamma_n)$ such that $(q_{i-1}, w_i, \gamma_i) \vdash (q_i, w_{i+1}, \gamma_{i+1})$ for each $i \in 1..n$.*

## Example 16.5




$$(q_0, 10, Z_0) \xrightarrow{1} (q_0, 0, 1Z_0) \xrightarrow{\epsilon} (q_1, 0, 1Z_0) \xrightarrow{0} (q_1, \epsilon, Z_0) \xrightarrow{\epsilon} (q_2, \epsilon, Z_0)$$

$$(q_0, 10, Z_0) \xrightarrow{\epsilon} (q_1, 10, Z_0) \xrightarrow[\epsilon]{} (q_2, 10, Z_0) \longrightarrow \textit{stuck}$$

## Exercise 16.6

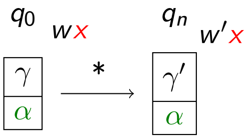
*How many stuck runs for word* 1100*?*

# Properties of IDs

## Theorem 16.1

*If* $(q_0, w, \gamma) \vdash^* (q_n, w', \gamma')$*, then the following holds.*

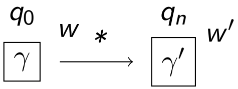
$$(q_0, wx, \gamma\alpha) \vdash^* (q_n, w'x, \gamma'\alpha)$$



## Theorem 16.2

*If* $(q_0, ww'', \gamma) \vdash^* (q_n, w'w'', \gamma')$*, then the following holds.*

$$(q_0, w, \gamma) \vdash^* (q_n, w', \gamma')$$



# Language of PDA

We can define two ways of accepting words

1. by final state
2. by empty stack

### Definition 16.5

*Let $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA. Then $L(P)$, the language recognized by $P$ by final state, is*

$$\{w|(q_0, w, Z_0)\vdash^*(q, \epsilon, \gamma') \wedge q \in F\}.$$

### Definition 16.6

*Let $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA. Then $L^\epsilon(P)$, the language recognized by $P$ by empty stack, is*

$$\{w|(q_0, w, Z_0)\vdash^*(q, \epsilon, \epsilon)\}.$$

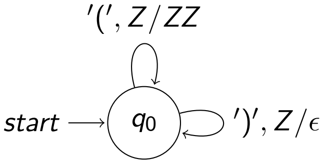
$F$ is useless in the definition of $L^\epsilon(P)$

# Example: recognize by stack

## Example 16.6

*Consider the following PDA.*

$$P = (\{q_0\}, \{'(', ')'\}, \{Z\}, \delta, q_0, Z, \{\})$$




## Exercise 16.7

*What language is recognized by P by empty stack?*

# End of Lecture 16