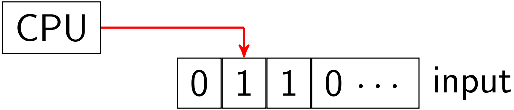
Pushdown automata I

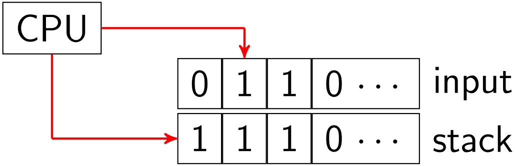
- Context-free languages are more general than regular languages
- For regular languages, by definition, there are automata to recognize them
- What are machines to recognize CFL?
- Pushdown automata (PDA)

  It's more powerful by having a stack
- DFA (or NFA):

Pushdown automata II



- Pushdown automata:



- What is a stack?

  We know they are like plates in a cafeteria

Pushdown automata III

An important property: last in first out

- Let's see how stack can help to recognize

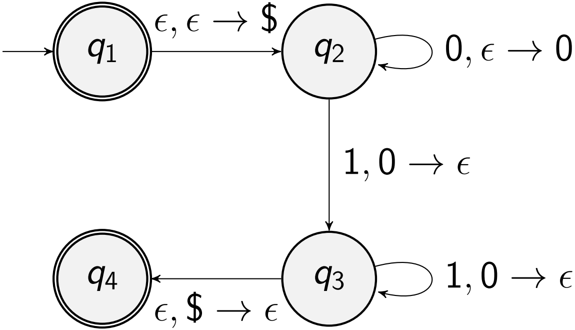
$$\{0^n 1^n \mid n \geq 0\}$$

- If 0 is read, 0 is pushed to stack
- If 1 is read, 0 is popped up
- By checking (0, 1) pairs, we know if the input is $0^n 1^n$

Example 2.14 I

- Consider the following language

$$\{0^n 1^n \mid n \geq 0\}$$



Example 2.14 II

- \$: a special symbol to indicate the initial state of stack
- How it works:

  $q_2 \rightarrow q_2$, put 0 into stack

  $q_2 \rightarrow q_3$ and $q_3 \rightarrow q_3$, read 1 and pop 0 up
- The input

$$0011$$

is the same as

$$\epsilon 0011 \epsilon$$

Example 2.14 III

- Steps:

$$
\begin{aligned}
&q_1, \emptyset, \epsilon \\
&q_2, \{\$\}, 0 \\
&q_2, \{0, \$\}, 0 \\
&q_2, \{0, 0, \$\}, 1 \\
&q_3, \{0, \$\}, 1 \\
&q_3, \{\$\}, \epsilon \\
&q_4, \{\}
\end{aligned}
$$

$\{\}$: contents of the stack before processing the input character

Example 2.14 IV

- We see that \$ can be used to check if the stack is empty
- Consider 00011

  Steps:

$$
\begin{aligned}
& q_1, \emptyset, \epsilon \\
& \vdots \\
& q_2, \{0, 0, 0, \$\}, 1 \\
& q_3, \{0, 0, \$\}, 1 \\
& q_3, \{0, \$\}, ?
\end{aligned}
$$

  Cannot reach $q_4 \Rightarrow$ rejected

Formal definition of pushdown automata I

- $(Q, \Sigma, \Gamma, \delta, q_0, F)$
  Q, Σ, Γ, F: finite sets
  1. Q: states
  2. Σ: alphabet
  3. Γ: stack alphabet
  4. δ:
  $$Q \times \Sigma_\epsilon \times \Gamma_\epsilon \rightarrow P(Q \times \Gamma_\epsilon)$$
  5. $q_0 \in Q$: start state
  6. $F \subset Q$: set of accept states
- We rely on

Formal definition of pushdown automata II

state, input, top of stack

to decide the move

$$q_1 \xrightarrow{a,b \to c} q_2$$

From q_1, read a, and replace top of stack b with c

Formal definition of example 2.14 I

- The language is

$$\{0^n 1^n \mid n \geq 0\}$$

- $M_1 = (Q, \Sigma, \Gamma, \delta, q_1, F)$

$$\begin{aligned} Q &= \{q_1, q_2, q_3, q_4\} \\ \Sigma &= \{0, 1\} \\ \Gamma &= \{0, \$\} \\ F &= \{q_1, q_4\} \end{aligned}$$

Formal definition of example 2.14 II

	0			1			ϵ		
	0	\$	ϵ	0	\$	ϵ	0	\$	ϵ
q_1									$\{(q_2, \$)\}$
q_2			$\{(q_2, 0)\}$	$\{(q_3, \epsilon)\}$					
q_3				$\{(q_3, \epsilon)\}$				$\{(q_4, \epsilon)\}$	
q_4									

- In the definition of δ we have $\Sigma_\epsilon \times \Gamma_\epsilon$

  Thus 9 columns in the table