Automata & languages

A primer on the Theory of Computation

Roland Schmid

nsg.ee.ethz.ch

ETH Zürich (D-ITET)

19 October 2023

Part 5 out of 5

Last week was all about

Context-Free Languages

Context-Free Languages

a superset of Regular Languages

Example $\{0^n 1^n \mid n \geq 0\}$ is a CFL but not a RL

We saw the concept of

Context-Free Grammars

$$(\mathcal{V}, \Sigma, \mathcal{R}, \mathcal{S})$$

- $\mathcal{V}$: variables
- Σ: alphabet
- $\mathcal{R}$: fine set of rules
- $\mathcal{S}$: start symbol

$$v \rightarrow w$$

$v \in \mathcal{V}$

$w \in (\Sigma_\epsilon \cup \mathcal{V})^*$

As for Regular Languages,

Context-Free Languages are recognized by “machines”

Language	Regular	Context-Free
Machine	DFA/NFA	Push Down Automata

Push-Down Automatas are pretty similar to DFAs

alphabet

start state

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

states

accepting states

Push-Down Automatas are pretty similar to DFAs

except for… the stack and transition function

stack alphabet

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

transition function

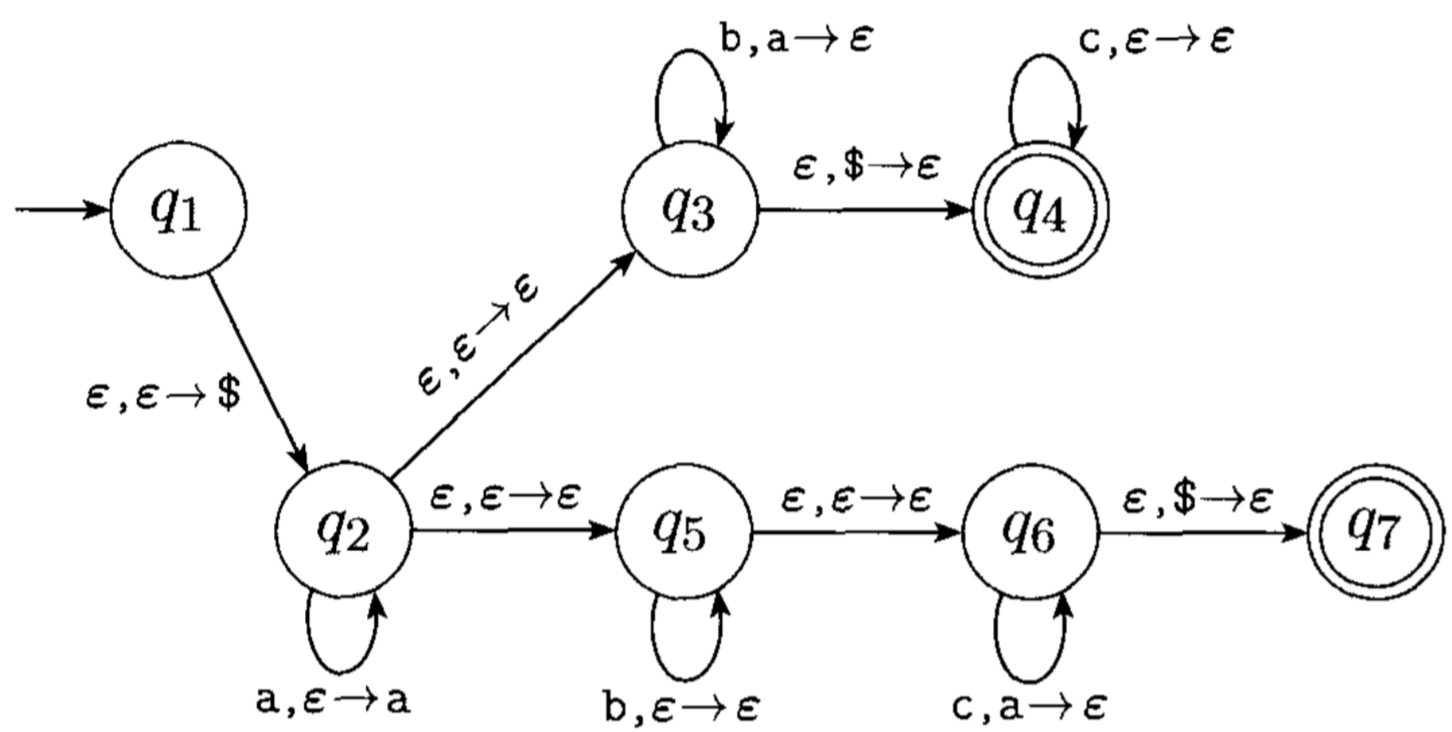
$$Q \times \Sigma_\epsilon \times \Gamma_\epsilon \rightarrow P(Q \times \Gamma_\epsilon)$$

Language

$L = \{a^i b^j c^k \mid i,j,k \geq 0 \text{ and } i=j \text{ or } i=k\}$

Machine

(PDA)



But before that, we'll prove

some extra properties about Context-Free Languages

Today's plan

1. PDA $\asymp$ CFG
2. Pumping lemma for CFL