Theory of Machines and Languages

Fatemeh Deldar

1403-1404

Three Basic Concepts

2. Grammars

$$S \to Ab,$$
$$A \to aAb,$$
$$A \to \lambda.$$

$$L = \{a^n b^{n+1} : n \geq 0\}$$

$$S \to SS,$$
$$S \to \lambda,$$
$$S \to aSb,$$
$$S \to bSa$$

$$L = \{w : n_a(w) = n_b(w)\}$$

Three Basic Concepts

2. Grammars

- Two grammars G_1 and G_2 are equivalent if they generate the same language, that is, if $L(G_1) = L(G_2)$

Consider the grammar $G_1 = (\{A, S\}, \{a, b\}, S, P_1)$, with P_1 consisting of the productions

$$S \rightarrow aAb|\lambda,$$
$$A \rightarrow aAb|\lambda.$$

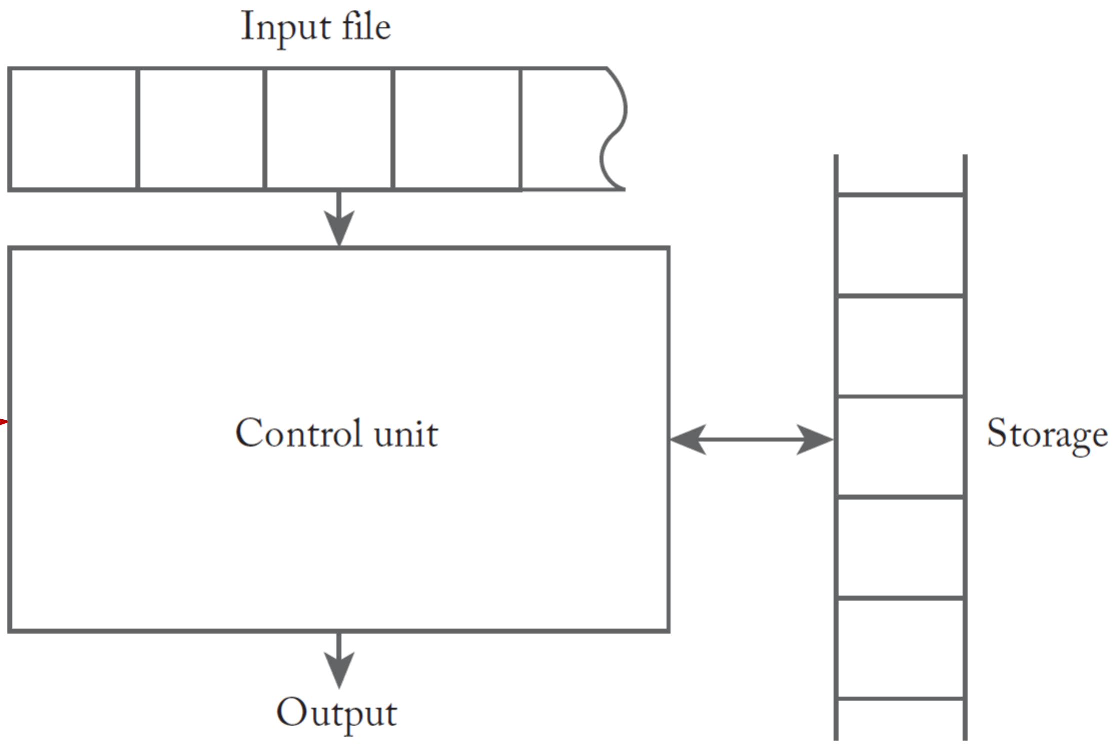
$$L(G_1) = \{a^n b^n : n \geq 0\}.$$

Three Basic Concepts

3. Automata

- An automaton is an abstract model of a digital computer

Control unit can be in any one of a finite number of internal states



Three Basic Concepts

- **Example**

Find a grammar that generates the language

$$L = \left\{ww^R : w \in \{a,b\}^+\right\}.$$

Find grammars for the following languages on $\Sigma = \{a\}$.

(a) $L = \{w : |w| \mod 3 > 0\}$.

(b) $L = \{w : |w| \mod 3 = 2\}$.

(c) $w = \{|w| \mod 5 = 0\}$.

Three Basic Concepts

- **Example**

Let $\Sigma = \{a, b\}$. For each of the following languages, find a grammar that generates it.

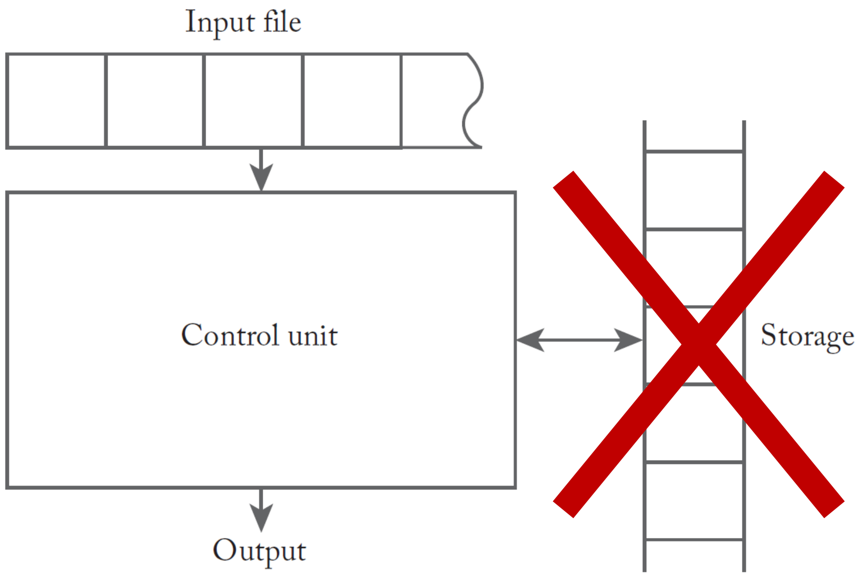
(a) $L_1 = \{a^n b^m : n \geq 0, m < n\}$.

(b) $L_2 = \{a^{3n} b^{2n} : n \geq 2\}$.

Finite Automata

Finite Automata

- This type of automaton is characterized by:
  1. Having no temporary storage
  2. An input file cannot be rewritten
  3. A finite amount of information can be retained in the control unit by placing the unit into a specific state
     - Since the number of such states is finite, a finite automaton can only deal with situations in which the information to be stored at any time is strictly bounded



Deterministic Finite Accepter

A **deterministic finite accepter** or **dfa** is defined by the quintuple

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

where

Q is a finite set of **internal states**,
Σ is a finite set of symbols called the **input alphabet**,
$\delta : Q \times \Sigma \rightarrow Q$ is a total function called the **transition function**,
$q_0 \in Q$ is the **initial state**,
$F \subseteq Q$ is a set of **final states**.

- **The transitions from one internal state to another are governed by the transition function δ**
  - **For example, if $\delta(q_0, a) = q_1$ then if the dfa is in state q_0 and the current input symbol is a, the dfa will go into state q_1**

Deterministic Finite Accepter

- Transition graphs are used to visualize and represent finite automata
  - Vertices represent states
  - Edges represent transitions
  - The labels on the vertices are the names of the states
  - The labels on the edges are the current values of the input symbol

- **Example**

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

$$\begin{aligned} \delta(q_0, 0) &= q_0, & \delta(q_0, 1) &= q_1, \\ \delta(q_1, 0) &= q_0, & \delta(q_1, 1) &= q_2, \\ \delta(q_2, 0) &= q_2, & \delta(q_2, 1) &= q_1. \end{aligned}$$