TWO-WAY FINITE AUTOMATA

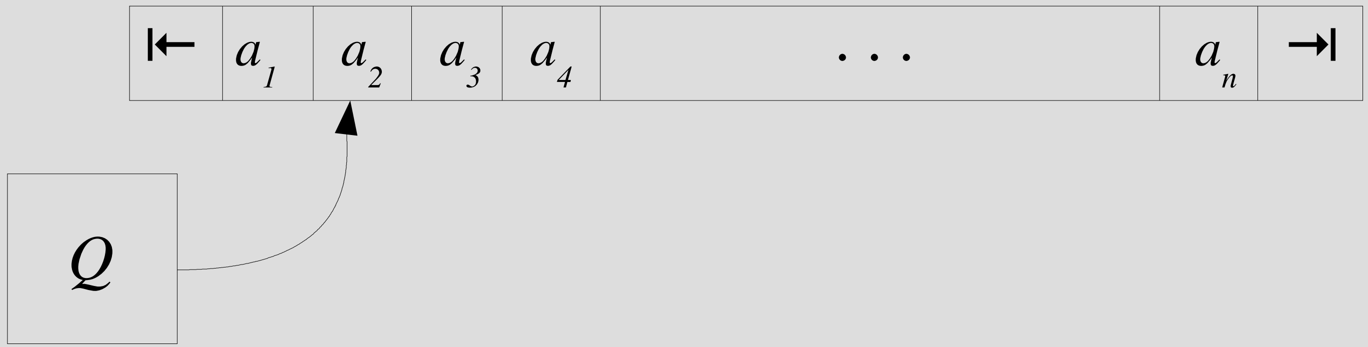
K VAMSI KRISHNA
KOTESWARA RAO VEMU

CONTENTS

- Introduction
- Formal Definition
- Example
- Equivalence with FA
- Constructing DFA

Introduction

Two-way FA are similar to finite automata except that they can read the input string in either direction.



Formal Definition

- $M = (Q, \Sigma, \vdash, \dashv, \delta, s, t, r)$
  - Q is a finite set (the *states*)
  - Σ is a finite set (the *input alphabet*)
  - $\vdash$ is the *left endmarker* , $\vdash \notin \Sigma$
  - $\dashv$ is the *right endmarker*, $\dashv \notin \Sigma$
  - $\delta : Q \times (\Sigma \cup \{\vdash, \dashv\}) \rightarrow (Q \times \{L, R\})$ is the *transition function*
  - $s \in Q$ is the *start state*
  - $t \in Q$ is the *accept state*
  - $r \in Q$ is the *reject state*, $r \neq t$

Formal Definition (Contd...)

- for all states q
  - $\delta(q, \vdash) = (u, R)$ for some $u \in Q$,
  - $\delta(q, \dashv) = (v, L)$ for some $v \in Q$
- for all symbols $b \in \Sigma \cup \{\vdash\}$
  - $\delta(t, b) = (t, R)$
  - $\delta(t, \dashv) = (t, L)$
  - $\delta(r, b) = (r, R)$
  - $\delta(r, \dashv) = (r, L)$

Formal Definition(Contd...)

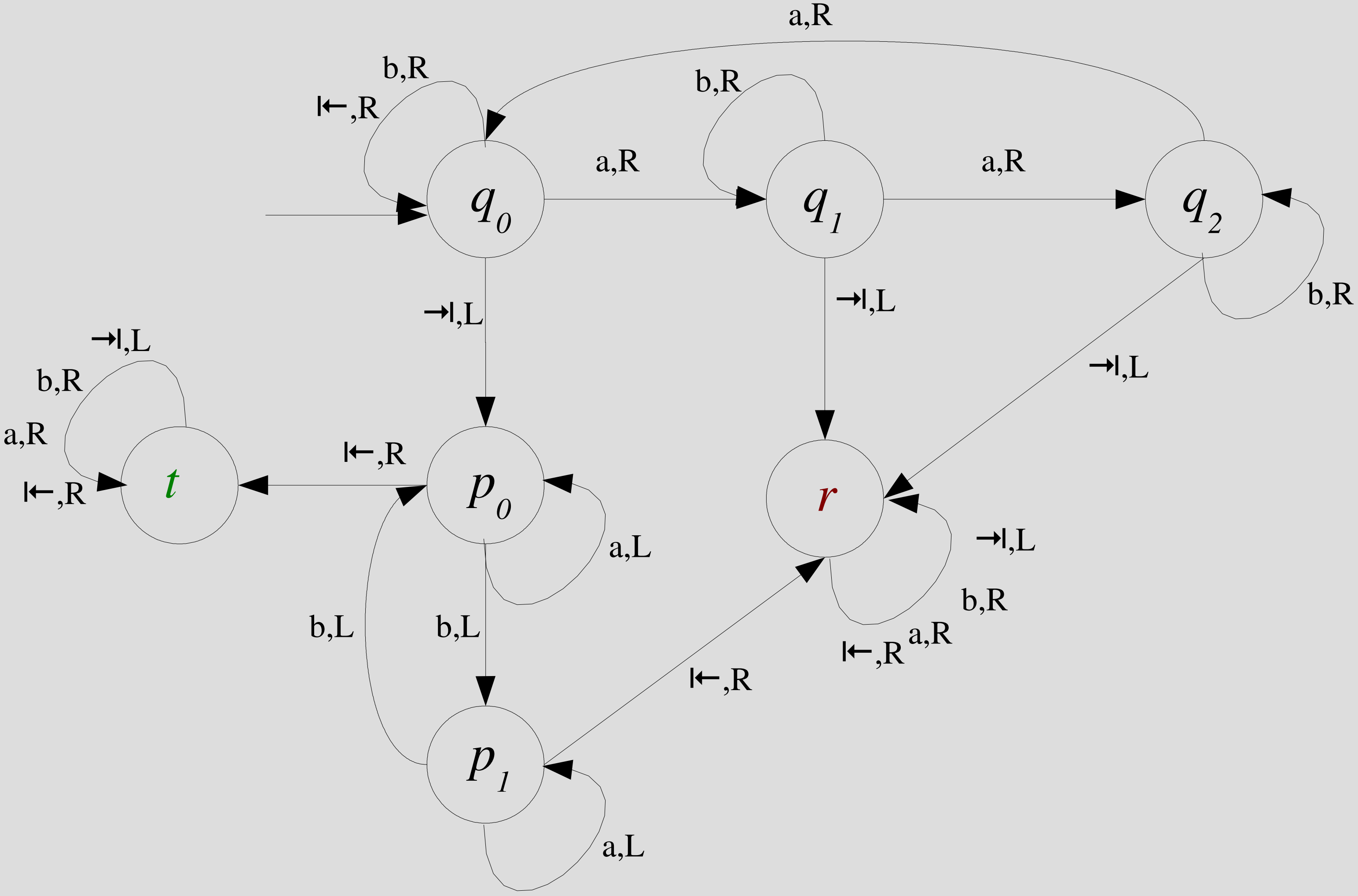
- Let $a_0 a_1 a_2 \ldots a_n a_{n+1} = \vdash x \dashv$ for any input x
- A *configuration* of the machine on input x is a pair (q,i) such that $q \in Q$ and $0 \leq i \leq n+1$
- The *start configuration* is given as $(s, 0)$
- The *next configuration* relation (denoted with $\xrightarrow[x]{1}$)
  - $\delta(p, a_i) = (q, \text{L}) \Rightarrow (p, i) \xrightarrow[x]{1} (q, i-1)$
  - $\delta(p, a_i) = (q, \text{R}) \Rightarrow (p, i) \xrightarrow[x]{1} (q, i+1)$

Formal Definition (Contd...)

- Lets define $\rightarrow_x^n$ inductively for $n \geq 0$
  - $(p, i) \rightarrow_x^0 (p, i)$
  - if $(p, i) \rightarrow_x^n (q, j)$ and $(q, j) \rightarrow_x^1 (u, k)$ then $(p, i) \rightarrow_x^{n+1} (u, k)$
  - $(p, i) \rightarrow_x^* (q, j) \Leftrightarrow \exists n \geq 0\ (p, i) \rightarrow_x^n (q, j)$
- The machine *accepts* the input x if $(s, 0) \rightarrow_x^* (t, i)$ for some i
- The machine *rejects* the input x if $(s, 0) \rightarrow_x^* (r, i)$ for some i
- The machine *halts* on input x if it either *accepts* or *rejects* x, otherwise it is said to *loop* on x
- The set $L(M)$ is defined to be the set of strings *accepted* by M

Example

- $L = \{\ x \in \{a,b\}^* \mid \#a(x) \text{ is a multiple of 3 and } \#b(x) \text{ is even }\}$



Equivalence with FA

- Basic idea for the proof
  - consider the string $w = xz$
- $T_x : (Q \cup \{\bullet\}) \rightarrow (Q \cup \{\perp\})$, where
  - $T_x(\bullet)$ is the state in which the machine crosses x for the first time into z and $T_x(\bullet) = \perp$ if the machine never *emerges* from x
  - At sometime the machine may move back into x in state q from z and later either *emerge* from x in state p or may *never emerge*
  - In the first case we define $T_x(q) = p$; in the second case $T_x(q) = \perp$
  - There can be only finitely many possible tables T

Equivalence with FA (Contd...)

- if $T_x = T_y$ and M accepts xz then M accepts yz
- $T_x = T_y \Rightarrow \forall z \;(M \text{ accepts } xz \Leftrightarrow M \text{ accepts } yz)$

$$\Leftrightarrow \forall z \;(xz \in L(M) \Leftrightarrow yz \in L(M))$$

$$\Leftrightarrow x \equiv_{L(M)} y$$

where $\equiv_{L(M)}$ is a Myhill Nerode relation with finite index, as the number of tables is finite

$\therefore$ L(M) is regular.

Constructing DFA

- Lets define DFA M'
  - $Q' = \{ \mathrm{T} : (Q \cup \{\bullet\}) \to (Q \cup \{\perp\}) \}$
  - $s' = \mathrm{T}_{\varepsilon}$
  - $\delta'(\mathrm{T}_x, a) = \mathrm{T}_{xa}$
  - $F' = \{ \mathrm{T}_x \;/ x \in \mathrm{L}(M) \}$
- We can prove that, $\delta'^{*}(\mathrm{T}_x , y) = \mathrm{T}_{xy}$ (by induction)
- $x \in \mathrm{L}(M') \Leftrightarrow \delta'^{*}(s', x) \in F'$

$$\Leftrightarrow \delta'^{*}(\mathrm{T}_{\varepsilon}, x) \in F'$$

$$\Leftrightarrow \mathrm{T}_x \in F'$$

$$\Leftrightarrow x \in \mathrm{L}(M) \qquad \therefore\ \mathrm{L}(M') = \mathrm{L}(M)$$

THANK YOU