CS361, Fall 1995

Deterministic Finite Automata

Instructor: Guo-Qiang Zhang

BY: Anurag Mishra and Jin Zhi

Sept-25, 1995, Monday.

1 Introduction

In this class the concept of DFA was introduced, the DFA was formally defined and some examples of the DFA's were given. In the end the topic of "Regular Expression " was started.

2 Definition of a DFA

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

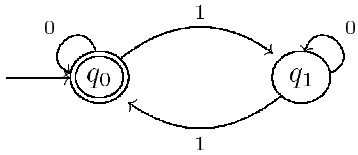
$$M = (Q, \Sigma, \delta, q_0, F)$$

where

- Q is a finite set of internal states,
- Σ is a finite set of symbols called the input alphabet,
- $\delta : Q \times \Sigma \longrightarrow Q$ is a (total) function called the transition function,
- q_0 is the initial state belongs to the set Q,
- and $F \subseteq Q$ is a set of final states.

3 Example of DFA

This DFA decides if a binary string has an even number of 1's.



The DFA drawn above can be formally written as:

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

with the transition function δ specified by

$$\delta(q_0, 0) = q_0 \quad \delta(q_0, 0) = q_1$$
$$\delta(q_0, 1) = q_1 \quad \delta(q_1, 1) = q_0$$

The transition function can also be represented in a table.

δ	0	1
q_0	q_0	q_1
q_1	q_1	q_0

A string $w = a_1 a_2 \cdots a_n$ with $a_i \in \Sigma$ for each a_i is accepted by a DFA if

$$\delta(q_0, a_1) = q_1$$
$$\delta(q_1, a_2) = q_2$$
$$\ldots$$
$$\delta(q_{n-1}, a_n) = q_n$$

and $q_n \in F$. This can be rewritten as

$$\delta^*(q_0, w) = q_n \;\&\; q_n \in F.$$

Thus, the empty string is accepted by a DFA only if the initial state is a final state.

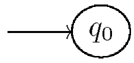
In our example, we have

$$\delta^*(q_0, 01101) = \delta^*(q_0, 1101) = \delta^*(q_1, 101) = \delta^*(q_0, 01) = \delta^*(q_0, 1) = q_1.$$

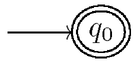
However, q_1 is not a final state, so the string 01101 is not accepted by this DFA.

4 Which of the diagrams represent a DFA?

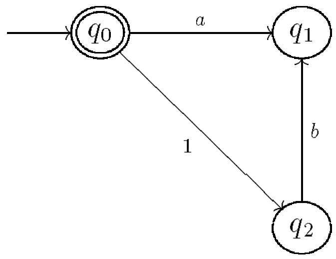
Example 1.



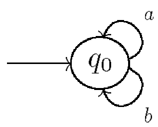
Example 2.



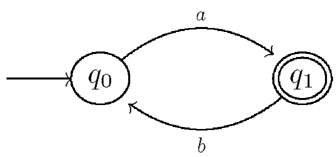
Example 3.



Example 4.



Example5.



1. It is not an DFA because the transition relation is not a function.

2. It is not an DFA because it does not fully specify a transition function.

3. It is not an DFA because the transition relation is not a function. From the initial state (also the final state) 'a' has two way to go.

4. It is an DFA with the empty set as it's final state.

5. It is not an DFA because the transition relation is not a function. From the initial state if there is a 'b' it can find no where to go.

Two of the most basic questions are:

- Given a language L, try to design a DFA M such that

$$L = L(M) := \{w \mid w \text{ is accepted by } M\}.$$

- Given a DFA, try to find what is the language accepted by it.

5 Regular expressions

Regular expressions are a special kind of notation for specifying languages in a concise way.

Given Σ, we have

- $\emptyset$, a (with $a \in \Sigma$) are regular expressions.
- If r_1 and r_2 are regular expressions then so are $r_1 + r_2$ (for union), $r_1 r_2$ (for concatenation), r_1^*, r_2^* (for Kleene star).

Example. Fix $\Sigma = \{0, 1\}$. We have the following correspondence from regular expressions to languages.

Regular Expression	Language
$0 + 1$ $\emptyset^*$	$\{0,1\}$ $\{\lambda\}$
$0^* + 1^*$	$0^* \cup 1^*$
$0^*1^* + 1^*0^*$	$(\{0\}^*\{1\}^*) \cup (\{1\}^*\{0\}^*)$