CS 4510: Automata and Complexity 08/25/2022

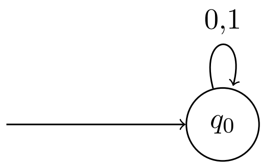
Lecture 2: Deterministic Finite Automata

Lecturer: Zvi Galil *Author: Austin Peng*

Contents

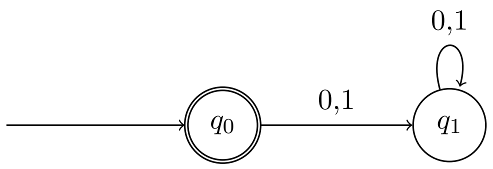
1 DFA Examples

Example 1. M_2



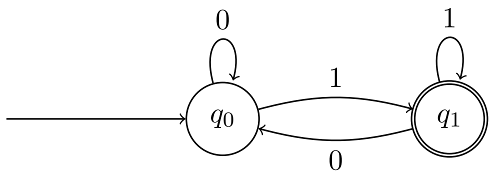
$L(M_1) = \varphi$. The language is empty because there are no accepting states.

Example 2. M_2



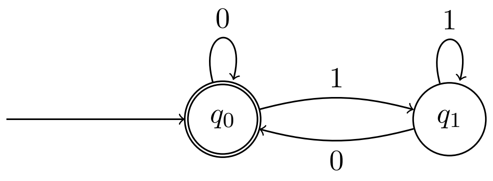
$L(M_2) = \{\varepsilon\}$.
Note the difference between M_1 and M_2. They recognize different langauges.

Example 3. M_3



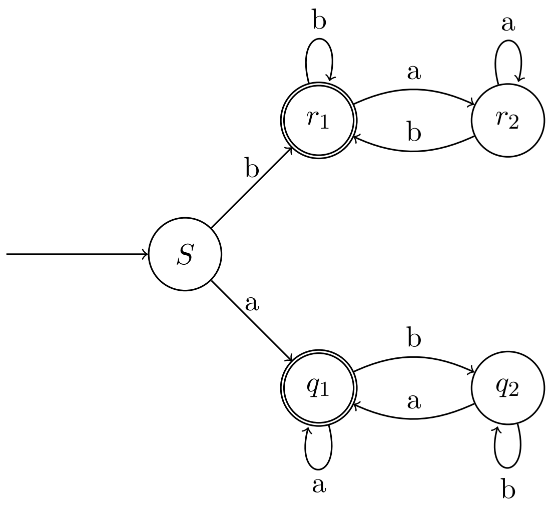
$L(M_3) = \{w|w \text{ ends in a } 1\}$

Example 4. M_4



$L(M_4) = \{\varepsilon \cup \text{ strings ending with } 0\}$. Note this is the complement of $L(M_3)$.

Example 5. M_5



$L(M_5)$ is the set of all strings that start and end with the same character. Note: $\Sigma = \{a, b\}$

2 Applications Of DFA

2.1 Modular Arithmetic

Let $w \in \{0,1\}^*$ (aka any binary string). We define $\overline{w}$ to be the value of the string as a binary number. Then, for $w \in \{0,1\}^*$ and $a \in \{0,1\}$, we have the following properties:

- $\overline{a} = a$
- $\overline{wa} = 2\overline{w} + a$

We can use a DFA to recognize modular arithmetic. For the following example, we will consider the following transition table of $\overline{w}$ mod 3. Note that the start state of our transition table is marked with an arrow.

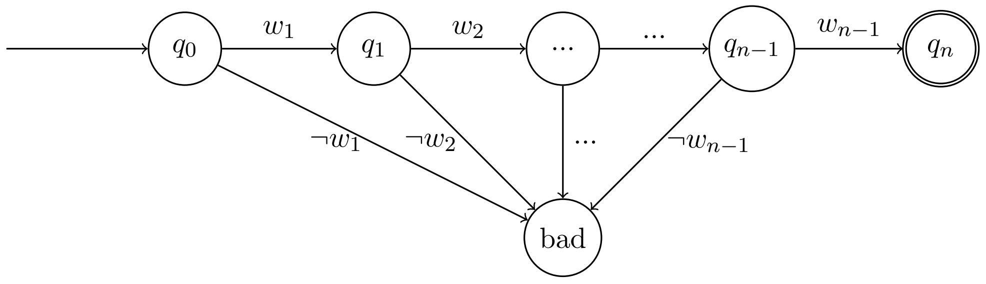
input a / $\overline{w}$ mod 3	$\overline{w0}$ mod 3	$\overline{w1}$ mod 3	state
0 (state q_0)	0	1	$\rightarrow q_0$
1 (state q_1)	2	0	q_1
2 (state q_2)	1	2	q_2

If we set the accepting state to be q_1 then this DFA will accept exactly those strings which are $\equiv 1$ modulo 3 (aka congruent to 1 modulo 3).

2.2 String Matching

2.2.1 Recognizing A Single String

For a string w, we can create a DFA for the language $L_w\{w\}$ as follows:



2.2.2 Recognizing A Suffix

Let L'_w be the set of strings that end in w. An example string from this language is $1101001 \in L'_{001}$, because it ends in 001. We can use the following transition table:

Q \ input	0	1
$\rightarrow$ bad	q_0	bad
q_0	q_{00}	bad
q_{00}	q_{00}	q_{001}
q_{001}	q_0	bad

We define q_{001} to be our only accepting state.
In the general case, we need to keep track of the longest suffix seen so far. We will use the states $\{\text{bad}, q_0, ..., q_n\}$.
The DFA will be in state q_i if $w_1...w_i$ is the longest suffix of the input seen so far that is a prefix of w. If we are in state q_i, then we have to see $n - i$ more symbols until we find the string. The transition function is defined as follows:

- $\delta(q_{i-1}, w_i) = q_i$
- $\delta(q_{i-1}, a \neq w_i) = q_j$, where $w_1w_2...w_j$ is the largest prefix of w that is a suffix of the current input (including a).