CS340: Theory of Computation — Sem I 2021-22

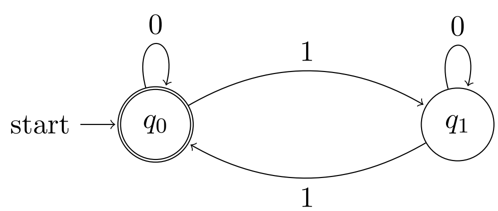
Lecture Notes 2: Deterministic Finite Automata

Raghunath Tewari — IIT Kanpur

1 Deterministic Finite Automaton

Consider the following language,

$$L = \{x \in \{0,1\}^* | x \text{ has an even no. of 1's}\}$$



State diagram of an automaton that accepts L

Definition 1.1. A *deterministic finite automaton* (in short DFA) is a 5 tuple $(Q, \Sigma, \delta, q_0, F)$, where,

- Q is a finite set called the set of *states*,
- Σ is a finite set called the *alphabet*,
- $\delta : Q \times \Sigma \to Q$ is the *transition function*,
- $q_0 \in Q$ is the *initial state*, and
- $F \subseteq Q$ is the set of *final states*.

We will now introduce some terminology related to DFA. Let $M = (Q, \Sigma, \delta, q_0, F)$.

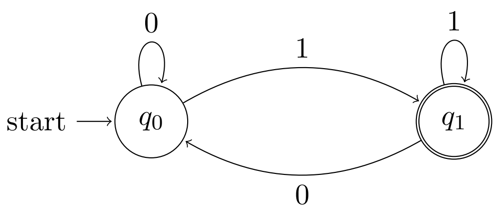
- We say M *accepts* $x \in \Sigma^*$ if M halts at an accept state when given x.
- $$L = L(M) = \{x \in \Sigma^* | M \text{ accepts } x\}$$

  In such a case, we say that M *accepts* L or M *recognizes* L. The terms 'accept' and 'recognize' will be used synonymously in this course.

Remark. If we say M accepts L it means that M accepts every string in L and every string accepted by M is in L. In other words, both the sets are equal. It is a common rookie mistake to a think of one as a proper subset of the other. Please avoid it!

Examples of DFA

1.
$$L_1 = \{x \in \{0,1\}^* | x \text{ ends with a } 1\}$$

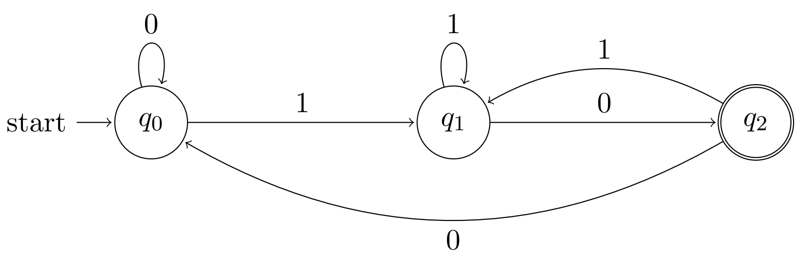


State diagram of an automaton that accepts L_1

Intuition of the various states:

- State q_0 corresponds to all those strings that end with a 0.
- State q_1 corresponds to all those strings that end with a 1.

2. Consider the following DFA



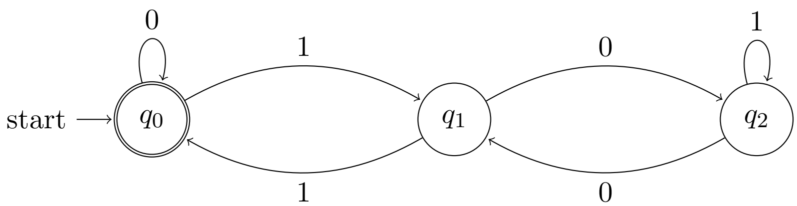
State diagram of DFA M_2

$$L(M_2) = \{x \in \{0,1\}^* | x \text{ ends with a } 10\}$$

Intuition of the various states:

- State q_0 corresponds to all those strings that end with two 0's or the string 0.
- State q_1 corresponds to all those strings that end with a 1.
- State q_2 corresponds to all those strings that end with a 10.

3.
$$L_3 = \{x \in \{0,1\}^* | x \text{ is divisible by } 3\}$$



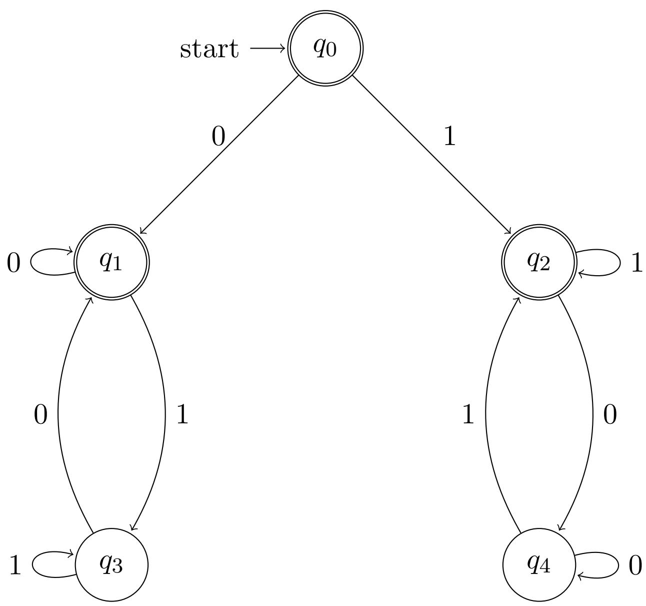
State diagram of DFA M_3 such that $L_3 = L(M_3)$

Intuition of the various states:

- q_0: all strings x such that $x \equiv 0 \mod 3$.
- q_1: all strings x such that $x \equiv 1 \mod 3$.
- q_2: all strings x such that $x \equiv 2 \mod 3$.

4.

$$L_4 = \{x \in \{0,1\}^* | x \text{ begins and ends with the same symbol}\}$$



State diagram of DFA M_4 such that $L_4 = L(M_4)$

Intuition of the various states:

- q_0: accepts ϵ.
- q_1: all strings beginning and ending with 0.
- q_3: all strings beginning with 0 and ending with 1.
- q_2 and q_4 are symmetric to q_1 and q_3.

Note 1. - Given a language there exist many DFAs (infact infinitely many) that accept the given language. However, given a DFA, there is only a unique language that the DFA accepts.

- The size of an automaton is independent of the length of the input string given to it. Strings can be arbitrarily long. This is the meaning of the term *finite* in a DFA.
- An automaton reads one bit at a time.
- At any instant while reading an input, the automaton does not know when the string will terminate. Its action at that point depends only on the (i) current state, and (ii) currently read input symbol.
- For every state p and every input symbol a, there is a unique state q that a DFA will go to from p on seeing the symbol a. This means that once a string is fixed, the automaton has a unique walk starting from the start state. If the last state is an accept state then the string is accepted, otherwise not. This is the meaning of the term *deterministic* in a DFA.
- The states of a DFA informally correspond to some property. Since a DFA has a finite number of states, hence it can "remember" only a finite amount of information of the input string. For example, a DFA cannot count the number of 1's present in a binary input string. We will prove such limitations of a DFA later in this course.

2 Definition of Computation of a DFA

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA. Let $w = a_1 a_2 \ldots a_n$ be a string in Σ^*.

- We say that M *accepts* w if there exists a sequence of states $r_0, r_1, \ldots, r_n \in Q$ (not necessarily distinct), such that

  1. $r_0 = q_0$,
  2. $r_i = \delta(r_{i-1}, a_i) \qquad$ for $i = 1, 2, \ldots n$
  3. $r_n \in F$.

- We say that M *accepts* $A \subseteq \Sigma^*$ if

$$A = \{w \in \Sigma^* | M \text{ accepts } w\}$$

  Denoted as $L(M)$. Once again note that M accepts a language A if and only if M accepts *exactly* those strings that belong to A.

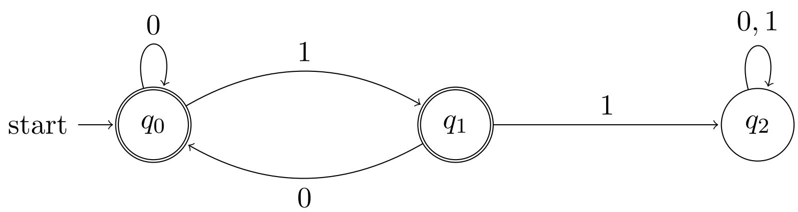
- A language L is said to be *regular* if $L = L(M)$ for some DFA M.

So far we have seen several regular languages. Here is an example of a non-regular language. We will prove this later.

$$L = \{0^n 1^n | n \geq 0\}$$

Some more examples

1. $$L_1 = \{x \in \{0,1\}^* | x \text{ does not contain the substring } 11\}$$



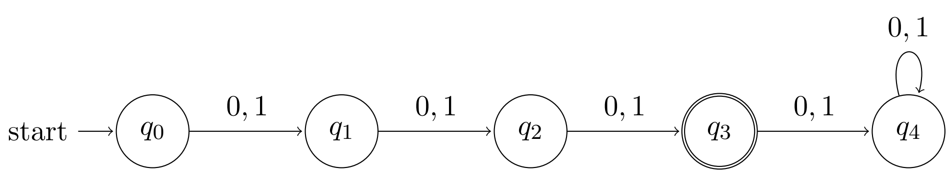
State diagram of DFA M_1 such that $L_1 = L(M_1)$

Intuition of the various states:

- q_0: strings that do not contain 11 and end with a 0.
- q_1: strings that do not contain 11 and end with a 1.
- q_2: strings that contain 11.

Remark. States from which the automaton can never reach an accept state are known as *dump states*. The state q_2 in the above DFA is an example of a dump state.

2. $$L_2 = \{x \in \{0,1\}^* || x| = 3\}$$



DFA for L_2

Intuition of the various states:

- q_0: string of length 0.
- q_1: strings of length 1.
- q_2: strings of length 2.
- q_3: strings of length 3.
- q_4: strings of length ≥ 4.

3 Regular Operations

Let us define some operations on languages and study their properties. Let $A, B \subseteq \Sigma^*$ be two languages.

1. Union operation:
$$A \cup B = \{x \mid x \in A \text{ or } x \in B\}$$

2. Concatenation operation:
$$A \cdot B = \{xy \mid x \in A \text{ and } y \in B\}$$
This is also denoted as AB.

3. Star operation:
$$A^* = \{x_1 x_2 \dots x_k \mid k \geq 0 \text{ and } x_i \in A\}$$

Example: Let $A = \{1, 2\}$ and $B = \{a, b, c\}$. Then

$$\begin{aligned} A \cup B &= \{1, 2, a, b, c\} \\ AB &= \{1a, 1b, 1c, 2a, 2b, 2c\} \\ A^* &= \{\epsilon, 1, 2, 11, 12, 21, 22, 111, 112, 121, 122, 211, 212, 221, 222, \dots\} \end{aligned}$$

Note that AB need not be the same as BA.

3.1 Closure Properties

We say that a class of objects is *closed* under some operation if applying that operation to objects of the class, produces an object that belongs to the same class as well.
Example: $\mathbb{N}$ is closed under addition and multiplication but not closed under subtraction.

Theorem 1. *Let A, B be two regular languages. Then $A \cup B$, AB and A^* are also regular. In other words, regular languages are closed under the union, concatenation and star operations.*

We will see a proof of this theorem later in the course.

Exercise 1. Construct DFAs for the following languages.

1. $L_1 = \{w \in \{0, 1\}^* | \#_0(w) \text{ is even and } \#_1(w) \text{ is odd}\}$
2. $L_2 = \{w \in \{0\}^* | |w| \text{ is divisible by 2 or 7}\}$
3. $L_3 = \{w \in \{0, 1\}^* | w \text{ is divisible by 5}\}$

Remark. $\#_0(w)$ denotes the number of occurrences of 0 in w. Similarly $\#_1(w)$.

Exercise 2. Read chapter 1.1 from textbook.