MTL783: Theory of Computation	1st Semester, 2025-2026

Lecture 3 — 31/07/2025

Lecturer: Prof Minati De *Scribe: Team 3*

Scribed by:

1. Shivaani Hari (2022MT11273)
2. Siya Gupta (2022MT11274)
3. Shrenik Mohan Sakala (2022MT11920)
4. Aahna Jain (2022MT11930)
5. Nilay Sharma (2022MT12007)

1 Optimisation Problems as Decision Problems

Any optimisation problem can be represented as a **decision problem**, and can be called a function that takes some input and returns "yes" or "no" as output.

Examples

1. **Connected Graph Problem:**
   - **Input:** Any graph
   - **Output:** Yes if the graph is connected, else no.
2. **Checking if an integer is prime:**
   - **Input:** An integer
   - **Output:** Yes if prime, else no.

The solutions to a decision problem can be represented in the form of a language which comprises of the set of all inputs such that the output for the decision problem is yes, i.e., if the input is a valid word in the language, return yes.

2 Automata

An automaton is an abstract mathematical model. It has a mechanism for reading input; the input is a string over a given alphabet, written on an input file, which the automaton can read but not change. Some key points about automata-

- An automaton whose output response is limited to a simple "yes" or "no" is called an **accepter**. Presented with an input string, an accepter either accepts the string or rejects it.
- A more general automaton, capable of producing strings of symbols as output, is called a **transducer**.
- In case of an acceptor, when we finish reading the string (done from left to right), a yes/no decision is made and returned.

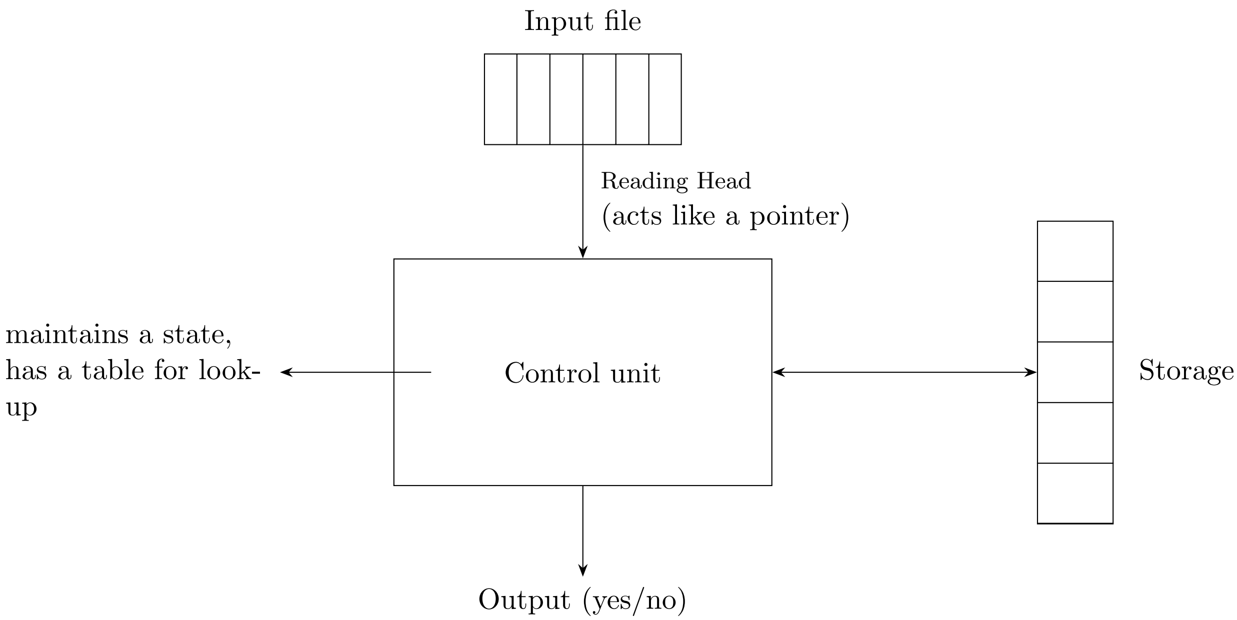


Figure 1: Basic Structure of Automata

3 Deterministic Finite Accepters (DFA)

Definition 3.1: 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
- $\delta : Q \times \Sigma \to Q$ 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

4 Transition Graph

- Each state in Q corresponds to a vertex in the transition graph.
- It is a directed graph.
- An incoming edge from nowhere is drawn to the initial state node.

The graph in Figure 2 represents the DFA

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

where δ is given by:

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

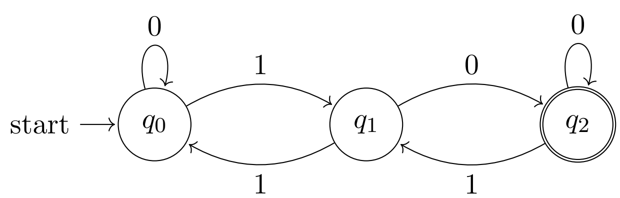


Figure 2: Simple Example of a DFA

This DFA accepts the string `010`. Starting in state q_0, the symbol 0 is read first: the automaton remains in q_0. Next, 1 is read and the automaton transitions to q_1. Next, 0 is read and the automaton transitions to q_2. We are now at the end of the string and in a final state q_2, so the string is accepted.

The DFA does not accept the string `101`, since it first goes to q_1 after reading 1, then to q_2 after reading 0, but then returns back to q_1 on reading 1, and the string ends here, leading to termination at a non-final state.

5 Transition Table

We can represent a transition function in the form of a table. There is a one-to-one relationship between M and its graph representation.

6 Extended Transition Function

It's convenient to introduce the **extended transition function** $\delta^* : Q \times \Sigma^* \to Q$. Here, the second argument is a string (not just a symbol) and the value is the state after reading the string.

$$\delta(q_0, a) = q_1 \quad \text{and} \quad \delta(q_1, b) = q_2,$$

then

$$\delta^*(q_0, ab) = q_2.$$

Formally, we define δ^* recursively as:

$$\delta^*(q, \lambda) = q \tag{2.1}$$

$$\delta^*(q, wa) = \delta(\delta^*(q, w), a) \tag{2.2}$$

for all $q \in Q$, $w \in \Sigma^*$, $a \in \Sigma$.

Example application:

$$\delta^*(q_0, ab) = \delta(\delta^*(q_0, a), b)$$

Now, $\delta^*(q_0, a) = \delta(\delta^*(q_0, \lambda), a) = \delta(q_0, a) = q_1$. So,

$$\delta^*(q_0, ab) = \delta(q_1, b) = q_2.$$

7 Language Accepted by a DFA

Definition 3.2: The language accepted by a DFA $M = (Q, \Sigma, \delta, q_0, F)$ is the set of all strings on Σ accepted by M:

$$L(M) = \{w \in \Sigma^* : \delta^*(q_0, w) \in F\}$$

Non-acceptance means the DFA stops in a non-final state; that is,

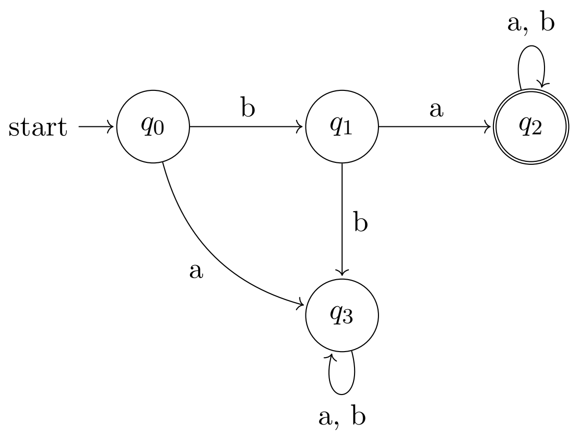
$$\overline{L(M)} = \{w \in \Sigma^* : \delta^*(q_0, w) \notin F\}$$

8 Questions

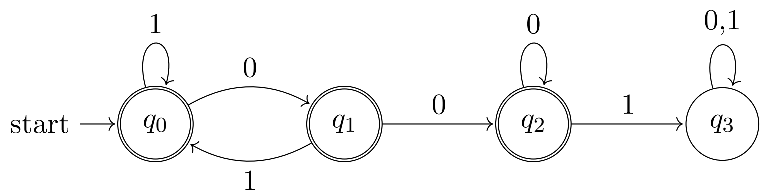
Q1: If you have a DFA, how can we construct a DFA whose language is the complement of that of the original DFA?

Ans: Make all the final states non-final and vice versa to get a new DFA.

Q2: Find a deterministic finite accepter that recognizes the set of all strings on $\Sigma = \{a, b\}$ starting with the prefix `ba`.



Q3: Find a DFA that accepts all the strings on $\{0,1\}$ except those containing the substring `001`.



You can obtain this by drawing a DFA that accepts strings containing `001` and then taking its complement.

9 A Theorem Connecting DFA and Graphs

Theorem 3.1: *Let $M = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite accepter, and let G_M be its associated transition graph. Then for every $q_i, q_j \in Q$, and $w \in \Sigma^+$,*

$\delta^*(q_i, w) = q_j$ **if and only if** there is in G_M a walk with label w from q_i to q_j.

Proof: (Sketch) The proof can be done by induction on the length of w. Assume the claim is true for all v with $|v| \leq n$. For w of length $n+1$, write $w = va$. Suppose $\delta^*(q_i, v) = q_k$. Since $|v| = n$, there is a walk in G_M labeled v from q_i to q_k. If $\delta^*(q_i, w) = q_j$, then M must have a transition $\delta(q_k, a) = q_j$, so by construction G_M has an edge (q_k, q_j) with label a. Thus, there is a walk in G_M labeled $va = w$ from q_i to q_j.

Since the result is obviously true for $n = 1$, we can claim by induction that for every $w \in \Sigma^+$,

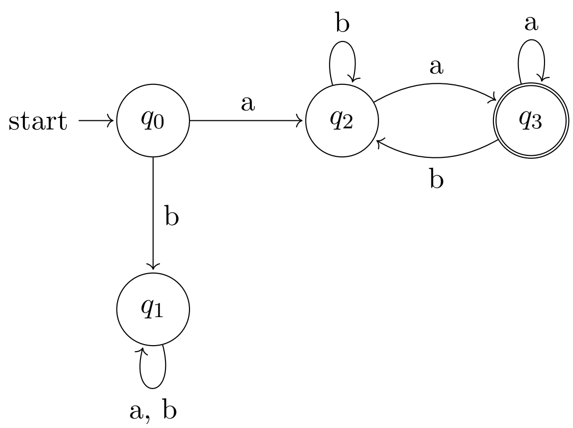
$$\delta^*(q_i, w) = q_j \quad \Longrightarrow \quad \text{there is a walk in } G_M \text{ from } q_i \text{ to } q_j \text{ labeled } w.$$

The argument can be turned around in a straightforward way to show that the existence of such a path implies $\delta^*(q_i, w) = q_j$, thus completing the proof.

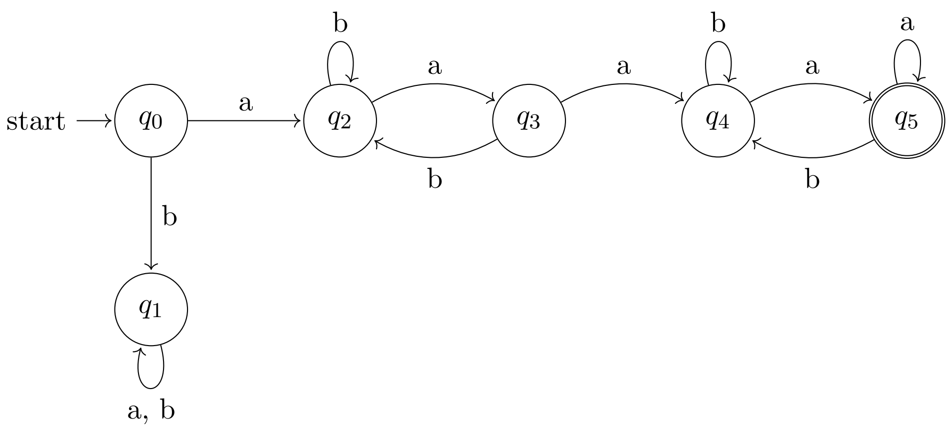
10 Regular Languages

Definition 3.3: A language L is called **regular** if and only if there exists some DFA M such that $L = L(M)$.

Example 3.1 Show that the language $L = \{awa : w \in \{a,b\}^*\}$ is regular.



Example 3.2 Show that $L_2 = \{aw_1aaw_2a : w_1, w_2 \in \{a,b\}^*\}$ is regular.



References

- Peter Linz, *An Introduction to Formal Languages and Automata*, 6th Edition, Jones & Bartlett Learning, 2016