Unit 2

Finite Automata

Regular Languages

Reading: Sipser, chapter 1
State Diagram

A simplified version of Finite Automaton
**Automatic door controller**

- Let’s design a controller for the automatic door in our neighborhood supermarket

![Diagram of automatic door with front and rear pads]
- Front pad detects a person about to walk through.
- Rear pad detects a person standing behind the door.
- Two *states* for the door:
  - Open
  - Close
- Four possible *inputs*:
  - Front
  - Rear
  - Both
  - Neither
State Diagram for automatic door controller
State Transition Table for automatic door controller

<table>
<thead>
<tr>
<th>State</th>
<th>Neither</th>
<th>Front</th>
<th>Rear</th>
<th>Both</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closed</td>
<td>closed</td>
<td>open</td>
<td>closed</td>
<td>closed</td>
</tr>
<tr>
<td>Open</td>
<td>closed</td>
<td>open</td>
<td>open</td>
<td>open</td>
</tr>
</tbody>
</table>
Please Welcome
our first computation model:
Finite Automata
The machine accepts a string if the process ends in a double circled state.
Finite Automata (FA)

- Our first formal model of computation.
- Consists of states (nodes) and moves (edges).
- The transition between states is according to an input word. Each symbol dictates one move.
- Some states are ‘good’ (accepting states) and some are ‘bad’ (rejecting states).
- A word is accepted by the Automaton if the transition it dictates ends in an accepting state.
Graphical Representation

• A circle represents a state of the automaton.
• The transition function is represented by directed and labeled edges between states.
• Accepting states have a double circle.
• The start state has an incoming arrow.
Formal Definition

A *Finite Automaton* (FA) is a 5-tuple \((Q, \Sigma, \delta, q_s, F)\):

- \(Q\) - a finite set called the *states*.
- \(\Sigma\) - a finite set called the *alphabet*.
- \(\delta:Q \times \Sigma \rightarrow Q\) is the *transition function*.
- \(q_s \in Q\) is the *start state*.
- \(F \subseteq Q\) is the set of *accepting states*. 
Example:

\[ A = (Q, \Sigma, \delta, q_s, F) \]

- \( Q = \{ q_1, q_2, q_3 \} \)
- \( \Sigma = \{ 0, 1 \} \)
- \( q_1 \) is the starting state
- \( F = \{ q_2 \} \)
- \( \delta \) is defined as

\[
\begin{array}{c|cc}
\text{q} & 0 & 1 \\
\hline
q_1 & q_1 & q_2 \\
q_2 & q_3 & q_2 \\
q_3 & q_2 & q_2 \\
\end{array}
\]
Another Example

- Formal definition: \( B = (Q, \Sigma, \delta, q_s, F) \)
  - continue in class
How does an Automaton work?

• Reads one symbol of the input word (i.e. one letter) in each time slot.

• In each time slot it moves to a new state (defined in transition function) according to the current state and the symbol read

• The automaton stops after the last symbol in the input word is read.

• If the state in which it stops is accepting - the automaton accepts the input word.
Formal Definition of Computation

A finite automaton \( M = (Q, \Sigma, \delta, q_s, F) \) accepts a string/word \( w = w_1\ldots w_n \) if and only if there is a sequence \( r_0\ldots r_n \) of states in \( Q \) such that:

1) \( r_0 = q_s \)

2) \( \delta(r_i, w_{i+1}) = r_{i+1} \) for all \( i = 0,\ldots,n-1 \)

3) \( r_n \in F \)
The Language of an Automaton

• The *language* of an automaton *A* is denoted *L(A)*

• *L(A)* consists of all the words that *A* accepts, i.e. when reading them it stops in an accepting state.
Recognizing the Language of an Automaton

The language of B:

$L(B) = \{ w \mid w \text{ is over } \Sigma = \{0,1\} \text{ and } |w| \text{ is even} \}$
More Examples

• An automaton accepting the language $\Sigma^*$

• an automaton accepting the empty language $\emptyset$
The language of A:

\[ L(A) = \{ w \mid w \text{ contains at least one } 1 \text{ and even number of } 0 \text{ follows the last } 1 \} \]
Designing an Automaton

Construct an automaton $B$ accepting the following language:

$L(B) = \{ w \mid w \text{ is over } \Sigma = \{a,b,c\} \text{ and } |w| \text{ is odd} \}$

B:

![Automaton Diagram]

- $q_0$ is the start state.
- $q_1$ is the accepting state.
- Transitions:
  - $a,b,c$ from $q_0$ to $q_1$.
  - $a,b,c$ from $q_1$ to $q_0$.

The automaton $B$ accepts strings over the alphabet $\Sigma = \{a,b,c\}$ with an odd length.
Another Example

• Construct an automaton accepting the following language:

\[ L = \{ w \in \{0,1\}^* \mid |w| \mod 4 = 0 \} \]
Another Example

• Construct an automaton $C$ accepting the following language:

$L(C) = \{ w \over \Sigma = \{a,b,c\} \mid w \in \Sigma^* \text{ and the last letter of } w \text{ is } c \}$

$C:$

![Automaton diagram]
Exercise

Construct an automaton accepting the language

\[ L = \{ \text{w over } \Sigma = \{0,1\} \mid |w| = 1 \text{ or } |w| \geq 3 \} \]

Answer: in class
State Diagram

• A graphical representation of a finite automaton FA is also called a state diagram.
• Given a formal definition of the FA one can draw its state diagram.
• Given a state diagram one can write a formal definition of the FA.
From state diagram to a formal description

- Consider the finite-state automaton $A$ defined over \{a, b\} by the state diagram shown below:

$$
\begin{align*}
Q_A &= \{q_0, q_1, q_2\}, \quad \Sigma_A = \{a, b\}, \\
\text{start state is } q_0, \\
F_A &= \{q_0, q_1\}, \\
\delta_A &- \text{ see table}
\end{align*}
$$

What is $L(A)$? What is $\sim L(A)$

<table>
<thead>
<tr>
<th>$\delta_A$</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>q0</td>
<td>q1</td>
<td>q0</td>
</tr>
<tr>
<td>q1</td>
<td>q1</td>
<td>q2</td>
</tr>
<tr>
<td>q2</td>
<td>q1</td>
<td>q0</td>
</tr>
</tbody>
</table>
Transition function

• The transition function \( \delta \) is a function
  \[ \delta: Q \times \Sigma \rightarrow Q \]
• It defines the movement of an automaton from one state to another.
• Transition function input is an ordered pair:
  (current state, current input symbol).
• For each pair of "current state" and "current input symbol" the transition function produces as output the next state in the automaton.
Example

- \( \delta(q_0, a) = q_1 \) means that state \( q_1 \) is *reachable* from state \( q_0 \) by the transition labeled by the input symbol \( a \).

- where
  - \( q_0 \) is the input state also called “current state”
  - \( a \) is the input symbol
  - \( q_1 \) is the output state also called “next state”
From formal description to a state diagram

• Draw a state diagram according to given $Q$, $\Sigma$, $\delta$, $q_0$, and $F$.

$Q=\{q_0,q_1,q_2,q_3\}$,
$\Sigma=\{0,1\}$,
$q_0$ initial,
$F=\{q_1\}$.

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$q_1$</td>
<td>$q_3$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$q_1$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$q_1$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>$q_3$</td>
<td>$q_3$</td>
<td>$q_3$</td>
</tr>
</tbody>
</table>

• Answer: In class
Extended transition function

• Let $\delta: Q \times \Sigma \rightarrow Q$ a transition function
• We define an \textit{extended transition function} $\delta': Q \times \Sigma^* \rightarrow Q$
• The extended transition function $\delta'$ defines the movement of an automaton on words.
• Let $w = u\sigma$ ($w, u$- words, $\sigma$- symbol)
• $\delta'(q, w) = \delta'(q, u\sigma) = \delta (\delta'(q, u), \sigma) = \delta (p, \sigma) = r$
Extended transition function

Formal definition:

• For all \( q \in Q \), and for all \( \sigma \in \Sigma \),
  \[ \delta'(q,\sigma) = \delta(q,\sigma) \quad \text{and} \quad \delta'(q,\varepsilon) = q \]

• \( \delta'(q,u\sigma) = \delta'\left(\delta'(q,u), \sigma\right) \)
The Language of an Automaton

• **Informally**: \( L(A) \) is the words that \( A \) accepts.

• **Formally**:

  Let \( A = \{Q, \Sigma, \delta, q_0, F\} \),

  \( L(A) = \{x | x \in \Sigma^*, \delta'(q_0, x) \in F\} \)

• We say that \( A \) **recognizes** \( L(A) \)
Regular Languages

A language is called a *regular language* iff some finite automaton recognizes it.
Two FA Questions

• Given the description of a finite automaton $M = (Q,\Sigma,\delta,q,F)$, what is the language $L(M)$ that it recognizes?

• In general, what kind of languages can be recognized by finite automata? (What are the regular languages?)
The automaton of a language

In-class exercise:

1. Describe formally and graphically an automaton, $A$, for the language
   \[ L = \{ w \mid \Sigma = \{0, 1\} \text{ and } |_1(w) \text{ is even} \} \]

2. Prove that $L(A) = L$. 
More Exercises

Give DFA state diagrams for the following languages over $\Sigma=\{0,1\}$:

1. $\{ w \mid w \text{ begins with 1 and ends with 0} \}$
2. $\{ w \mid w \text{ does not contain substring 110} \}$
3. $\{ w \mid w \text{ contains 110 as a substring} \}$