Theory of Computation

Lab Session 2

February 04 , 2016

Innopolis University

News

Technical Report

The technical report (essay) is 20% of the final mark and will be divided in 2 parts:

1. An essay on a topic no more than 5 pages long.
2. A live presentation on the topic.

There are extra points, 5 out the final 100 marks, (optional) for a video on the topic.

News

Essay description

- The essay is on a specific topic. We will publish a list of topics by this Friday on moodle.
- You are free to choose another topic: your TA needs to approve it.
- The essay is 5 pages long.
- The essay will be in groups of 3 or 4 students.
- You need to communicate your TA the name of the students that compose each group: by February 18th.
- The essay is to be submitted on April 04th.

News

Presentation description

- The presentation will be about the essay.
- All students in each group need to present.
- The presentation is max 20 minutes.
- The day of the presentation is to be defined (we will notify later).

Agenda

- Exercises on Finite State Automaton (FSA)

FSA

- Finite State Automaton is a model of computation.
- It is a simple computing device: it acts as a language acceptor.

FSA

- Finite State Automaton is a model of computation.
- It is a simple computing device: it acts as a language acceptor.

Let's see an example.

FSA - Example (intuition)

If $\Sigma = \{a, b\}$ and L_1 is defined as

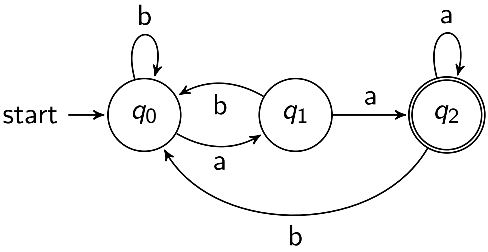
$$L_1 = \{x \in \Sigma^* \mid x \text{ ends with } aa\}$$

FSA - Example (intuition)

If $\Sigma = \{a, b\}$ and L_1 is defined as

$$L_1 = \{x \in \Sigma^* \mid x \text{ ends with } aa\}$$

Does the following FSA accepts all strings represented by the language L_1?



Example (informally)

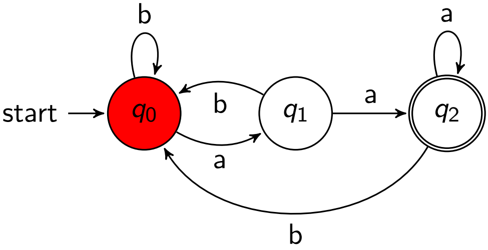
String $x = ababaa$ belongs to L_1. Meaning, $x \in \Sigma^*$ and it ends with aa. Let's see if the FSA accepts the string x

Example (informally)

String $x = ababaa$ belongs to L_1. Meaning, $x \in \Sigma^*$ and it ends with aa. Let's see if the FSA accepts the string x

q_0 is the starting point (it is graphically denoted by *start*). So the FSA starts in state q_0

$x = ababaa$



Example (informally)

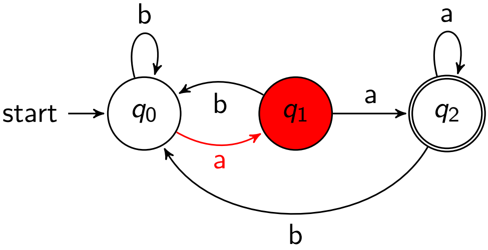
Then, we go through each character of x, following the transitions in the FSA

Example (informally)

Then, we go through each character of x, following the transitions in the FSA

$x = \mathbf{a}babaa$

From state q_0 and label a, we reach state q_1

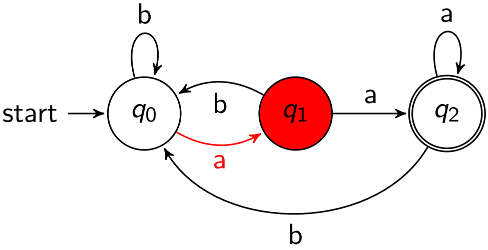


Example (informally)

Then, we go through each character of x, following the transitions in the FSA

$x = \mathbf{a}baba a$

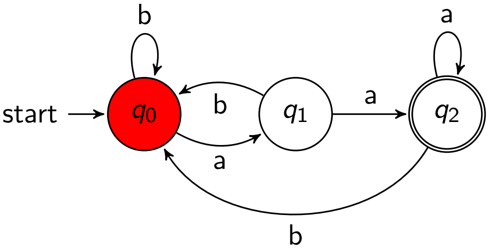
From state q_0 and label a, we reach state q_1



We repeat the process for all characters in x. If at the end we reach a final state (graphically denoted by the double circle state), we say the FSA accepts the string x.

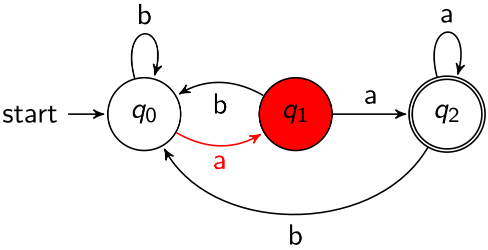
Example (informally) (1)

$x = ababaa$



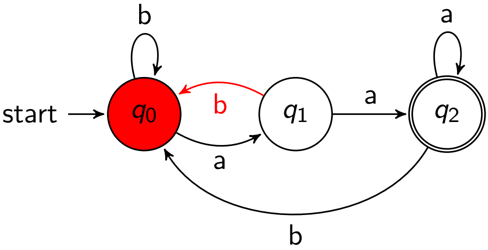
Example (informally) (2)

$x =$ **a***babaa*



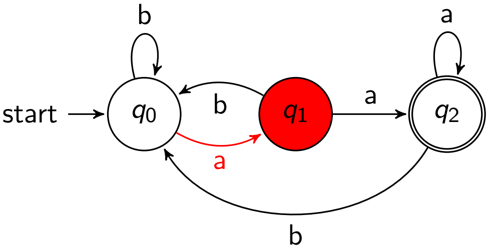
Example (informally) (3)

$x = a\mathbf{b}abaa$



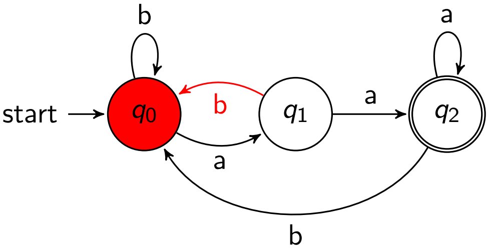
Example (informally) (4)

$x = ab\mathbf{a}baa$



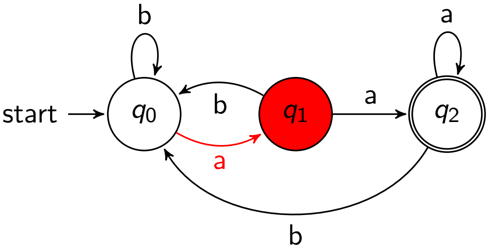
Example (informally) (5)

$x = aba\mathbf{b}aa$



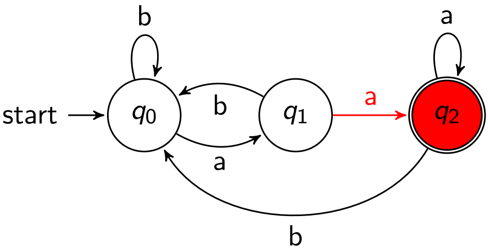
Example (informally) (6)

$x = abab\mathbf{a}a$



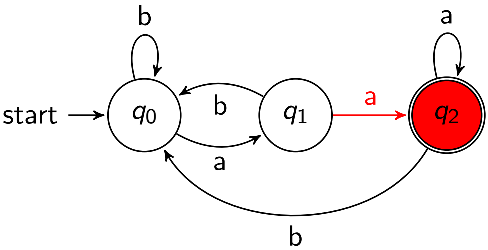
Example (informally) (7)

$x = ababa\mathbf{a}$



Example (informally) (7)

$x = ababa\mathbf{a}$



We went through all characters of x and ended up in a final state: string x belongs to language L_1.

FSA (Formal definition)

A complete Finite State Automaton

A complete Finite State Automaton is a tuple $< Q, \Sigma, q_0, A, \delta >$, where

Q is a finite set of *states*;
Σ is a finite *input alphabet*;
$q_0 \in Q$ is the *initial* state;
$A \subseteq Q$ is the set of *accepting* states;
$\delta : Q \times \Sigma \to Q$ is a total *transition* function.

For any element q of Q and any symbol $\sigma \in \Sigma$, we interpret $\delta(q, \sigma)$ as the state to which the FSA moves, if it is in state q and receives the input σ.

The extended transition δ^*

A move sequence starts from an initial state and is *accepting* if it reaches one of the final states (informally explained with the previous example).
Formally, this transition is defined recursively:

the extended transition δ^*
Let $M = < Q, \Sigma, q_0, A, \delta >$ be a complete finite state automaton.
We define the extended transition function

$$\delta^* : Q \times \Sigma^* \to Q$$

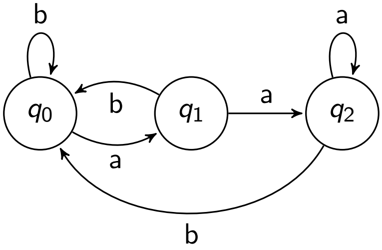
as follows:

1. For every $q \in Q$, $\delta^*(q, \epsilon) = q$
2. For every $q \in Q$, every $y \in \Sigma^*$, and every $\sigma \in \Sigma$,

$$\delta^*(q, y\sigma) = \delta(\delta^*(q, y), \sigma)$$

The extended transition (Example)

The complete FSA M contains the following transitions



$$
\begin{aligned}
\delta^*(q_1, baa) &= \delta(\delta^*(q_1, ba), a) \\
&= \delta(\delta(\delta^*(q_1, b), a), a) \\
&= \delta(\delta(\delta(\delta^*(q_1, \epsilon), b), a), a) \\
&= \delta(\delta(\delta(q_1, b), a), a) \\
&= \delta(\delta(q_0, a), a) \\
&= \delta(q_1, a) \\
&= q_2
\end{aligned}
$$

Acceptance by a FSA

The extended transition function is used to determine what it means for a FSA to accept (or reject) a string or a language. Formally:

Acceptance by a FSA

The extended transition function is used to determine what it means for a FSA to accept (or reject) a string or a language. Formally:

Acceptance by a FSA

Let $M =< Q, \Sigma, q_0, A, \delta >$ be a complete FSA, and let $x \in \Sigma^*$. The string x is accepted by M if

$$\delta^*(q_0, x) \in A$$

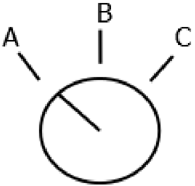
and it is rejected by M otherwise. The language accepted by M is the set

$$L(M) = \{x \in \Sigma^* \mid x \text{ is accepted by } M\}$$

If L is a language over Σ, L is accepted by M iff $L = L(M)$

Example - Three Position Switch

Consider a three-position electrical switch.The switch consists of a knob that can rotate in one of the three positions A, B, C after a shift in a clockwise or counterclockwise (shifts from A to C or C to A are not allowed). Model the operation of the switch with a complete FSA.



Example - Three Position Switch

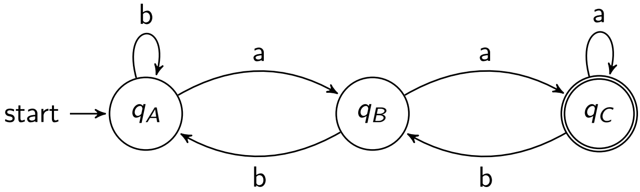
The complete Finite State Automaton

$$
\begin{tikzcd}
A \arrow[loop left, "\text{Counter}"] \arrow[r, bend left, "\text{Clockwise}"] & B \arrow[l, bend left, "\text{Counter}"] \arrow[r, bend left, "\text{Clockwise}"] & C \arrow[l, bend left, "\text{Counter}"] \arrow[loop right, "\text{Clockwise}"]
\end{tikzcd}
$$

A complete FSA models the possible configurations of a system with states, and events or external actions (inputs) that can cause a configuration change with transitions. An automaton that solves the proposed exercise is shown. It has a set of states $Q = (A, B, C)$ to represent the position and a transition function δ that models changes in position of the switch following the application of an appropriate motion to the knob.

Example - Three Position Switch

The Automaton Acceptor



An automaton acceptor of the language sequences of moves that lead from A to C.

Exercises

Exercises (first part)

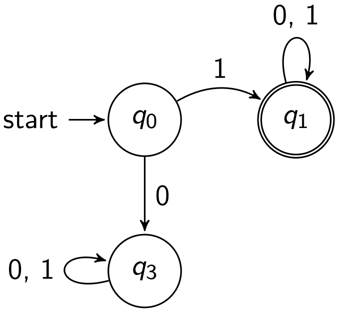
Build complete FSAs that recognise the following languages:
Let Σ be the alphabet $\Sigma = \{0, 1\}$

- $L_0 = \{x \in \Sigma^* \mid \ x \text{ starts with } 1\}$;
- $L_1 = \{x \in \Sigma^* \mid \ x \text{ does not begin with } 1\}$;
- $L_2 = \{x \in \Sigma^* \mid \text{ any } 0 \text{ in } x \text{ is followed by at least a } 1\}$.
  Strings example: 010111, 1111, 01110111011.
- $L_3 = \{x \in \Sigma^* \mid \ x \text{ ends with } 00\}$;
- $L_4 = \{x \in \Sigma^* \mid \ x \text{ contains exactly 3 zeros}\}$;

Solution (1)

Let Σ be the alphabet $\Sigma = \{0, 1\}$

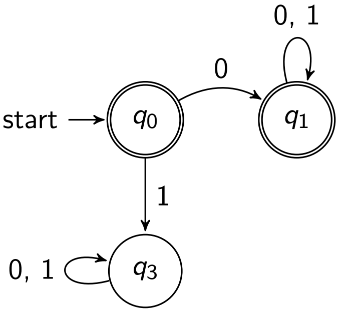
- $L_0 = \{x \in \Sigma^* \mid x \text{ starts with } 1\}$;



Solution (2)

Let Σ be the alphabet $\Sigma = \{0, 1\}$

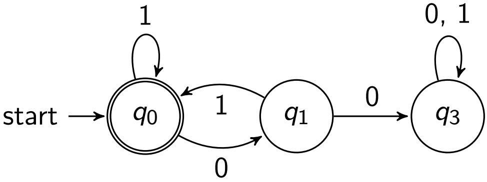
- $L_1 = \{x \in \Sigma^* \mid \ x \text{ does not begin with } 1\}$;



Solution (3)

Let Σ be the alphabet $\Sigma = \{0, 1\}$

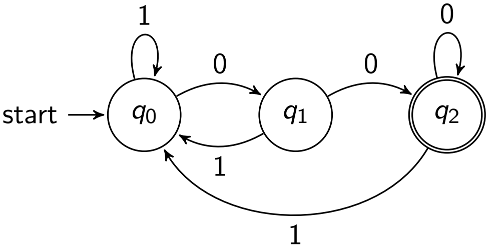
- $L_2 = \{x \in \Sigma^* \mid \text{ any 0 in } x \text{ is followed by at least a 1}\}$.
  Strings example: 010111, 1111, 01110111011.



Solution (4)

Let Σ be the alphabet $\Sigma = \{0, 1\}$

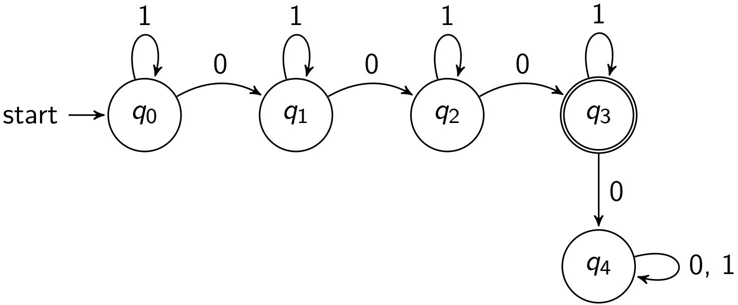
- $L_3 = \{x \in \Sigma^* \mid \ x \text{ ends with } 00\}$;



Solution (5)

Let Σ be the alphabet $\Sigma = \{0, 1\}$

- $L_4 = \{x \in \Sigma^* \mid x \text{ contains exactly 3 zeros}\}$;



Exercises (second part)

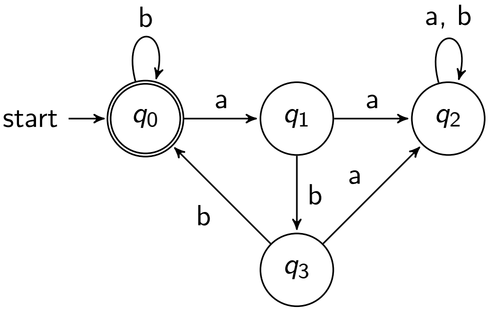
Build complete FSAs that recognise the following languages:
Let Σ be the alphabet $\Sigma = \{a, b\}$

- $L_5 = \{x \in \Sigma^* \mid$ every a in x (if there are any) is followed immediately by $bb\}$.
- $L_6 = \{x \in \Sigma^* \mid$ x ends with b and does not contain the substring $aa\}$.
- $L_7 = \{x \in \Sigma^* \mid$ x contains the substring $abbaab\}$;
- $L_8 = \{x \in \Sigma^* \mid$ x has an even number of 0's and an even number of 1's$\}$;

Solution (6)

Let Σ be the alphabet $\Sigma = \{a, b\}$

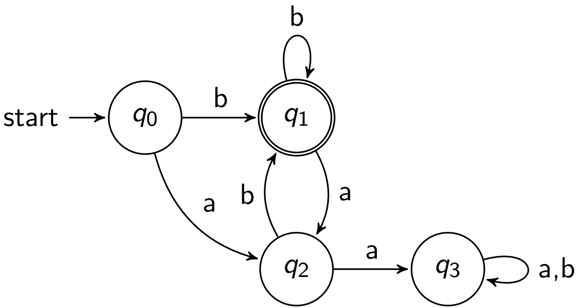
- $L_5 = \{x \in \Sigma^* \mid$
  every a in x (if there are any) is followed immediately by $bb\}$.



Solution (7)

Let Σ be the alphabet $\Sigma = \{a, b\}$

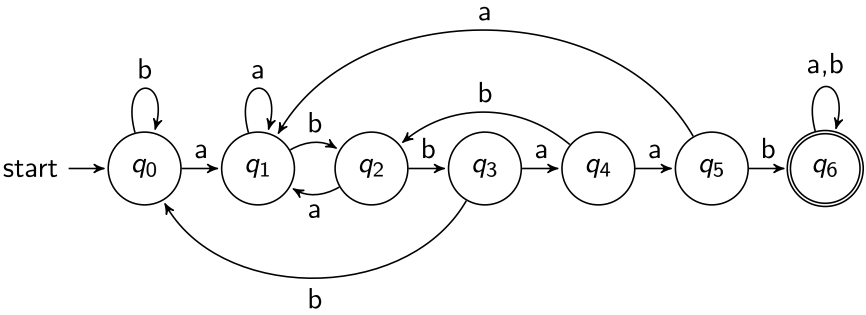
- $L_6 = \{x \in \Sigma^* \mid$
  x ends with b and does not contain the substring $aa\}$.



Solution (8)

Let Σ be the alphabet $\Sigma = \{a, b\}$

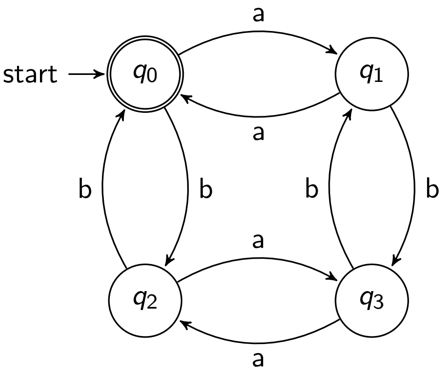
- $L_7 = \{x \in \Sigma^* \mid \ x \text{ contains the substring } abbaab\}$;



Solution (9)

Let Σ be the alphabet $\Sigma = \{a, b\}$

- $L_8 = \{x \in \Sigma^* \mid$
  x has an even number of a's and an even number of b's$\}$;



Exercises - Homework (1)

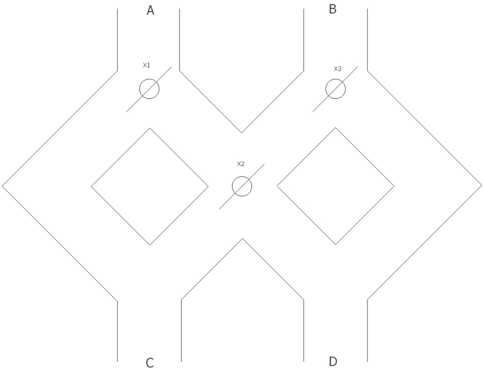
Build complete FSAs accepting the following languages over the alphabet $\Sigma = \{0, 1\}$

- $L_a = \{x \in \Sigma^* \mid \ x$ is the a binary representation of an integer, and it is divisible by $3\}$;
- $L_b = \{x \in \Sigma^* \mid$ x begins with a 1 that, when interpreted as a binary integer, is multiple of $5\}$;
- $L_c = \{x \in \Sigma^* \mid |x| \geq 2 \wedge$ whose final two symbols are the same$\}$;

Build a complete FSA accepting the following languages over the alphabet $\Sigma = \{a, b, c\}$

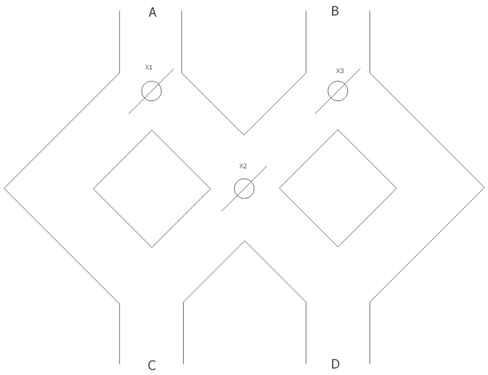
- $L_d = \{x \in \Sigma^* \mid$ the substring *abc* in x occurs an odd number of times$\}$;

Exercises - Homework (2a)



The figure is a marble toy. A marble is dropped at A or B. Levers $x1$, $x2$, and $x3$ cause the marble to fall either to the left of to the right. Whenever a marble encounters a lever, it causes the lever to reverse after the marble passes, so the next marble will take the opposite branch.

Exercises - Homework (2b)



Model this toy by a complete FSA. Let the inputs A and B represents the input into which the marble is dropped. Let acceptance corresponds to the marble existing at D; nonacceptance represents a marble exiting at C.

Exercises - Homework (3)

Implement, in the programming language of your choice, the FSAs of the previous examples and exercises (give an elegant solution to them!).