MTAT.05.125 Introduction to Theoretical Computer Science

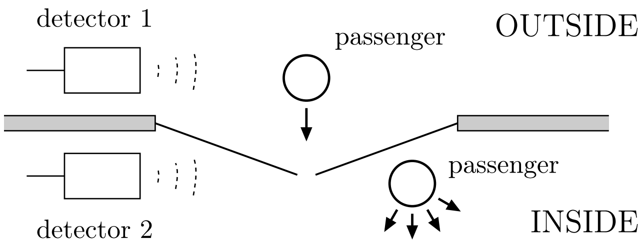
Autumn 2015

Vitaly Skachek
Estonian version by Reimo Palm
English version by Yauhen Yakimenka

Lecture 5. Deterministic automata and regular languages.

Automata theory

Example 1. Control system of airport entrance consists of two detectors, one of which is outside and another one is inside. Each detector is able to detect people nearby. We need to design a system to open/close the doors based on data from the detectors.

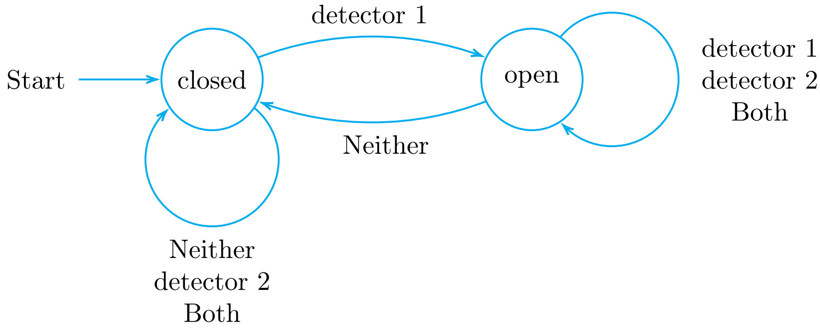


The new status will depend on two factors, what is the current status and which detectors are activated. The following table presents all variants for the new state.

Current status	Neither activated	Detector 1 activated	Detector 2 activated	Both activated
closed	closed	open	closed	closed
open	closed	open	open	open

When there are no people around (none of the detectors has been activated) the door should be closed. If a person comes into the view of detector 1, the door should be opened. However, if detector 2 observes someone (regardless of what detectors 1 sees), the situation should not change – otherwise it could hurt person inside.

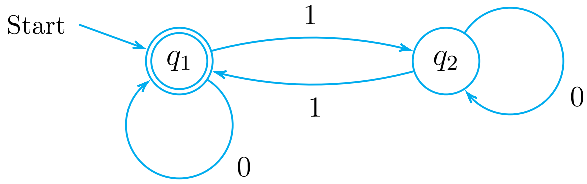
The rules of changing the states from the table could be also drawn as a state diagram:



This is more intuitive description of system "moving" between states depending on data from detectors.

Typically an automaton reads a sequence of input symbols.

Example 2. State diagram shows the automaton for checking if input sequence contains even number of ones.



For instance, let us "debug" how the automaton reads the input sequence 010010.

Step	Reads character	New state
	Start	q_1
1.	0	q_1
2.	1	q_2
3.	0	q_2
4.	0	q_2
5.	1	q_1
6.	0	q_1
	Stop	

With double we denote the *accept* states. If automaton is in such state after reading all input, we say the automaton "accepts" input sequence.

Definition. *A* finite automaton *is a 5-tuple* $(Q, \Sigma, \delta, q_0, F)$*, where*

- Q *is a finite set called* set of states*;*
- Σ *is a finite set called* alphabet*;*
- $\delta\colon Q \times \Sigma \to Q$ *is the transition function;*
- $q_0 \in Q$ *is a (unique)* start state*;*
- $F \subseteq Q$ *is a set of* accept states*.*

Example 3. In the example 2:

- $Q = \{q_1, q_2\}$.
- $\Sigma = \{0, 1\}$.
- Start state is q_1.
- $F = \{q_1\}$.
- Transition function δ:

  $$\delta(q_1, 0) = q_1, \quad \delta(q_1, 1) = q_2, \quad \delta(q_2, 0) = q_2, \quad \delta(q_2, 1) = q_1,$$

  or in the table form:

	0	1
q_1	q_1	q_2
q_2	q_2	q_1

Definition. *Let* $M = (Q, \Sigma, \delta, q_0, F)$ *be a finite automaton and* $w = a_1 a_2 \dots a_n$ *be a string, where each* $a_i \in \Sigma$*. Then we say that* M accepts *the string* w*, if there exists a sequence of states* r_0*,* r_1*, ...,* r_n*, as follows:*

1. $r_0 = q_0$*;*
2. *for each* $i = 0, \dots, n-1$ *it holds* $\delta(r_i, a_{i+1}) = r_{i+1}$*;*
3. $r_n \in F$*.*

We will call any set of strings *a language*.

Definition. *Let* M *be a finite automaton. We say that* L *is a language of* M *if* L *is exactly the set of all strings accepted by* M*. In this case we will also say that* M accepts (or recognises) L*.*

In other words, M accepts language L, if and only if

$$L = \{w \mid M \text{ accepts } w\}.$$

Example 4. Let M be the automat from Example 2. Then the language accepted by M is $L = \{\varepsilon, 0, 00, 11, \ldots, 0101, \ldots, 011110, \ldots\}$, where ε stands for empty string.

Definition. *A language L is called* regular *if there exists a finite automaton that recognises L.*

Definition. *Let L_1 and L_2 be two languages. We define operations* union*,* concatenation *and* star *as follows:*

- *Union:*
$$L_1 \cup L_2 = \{w \mid w \in L_1 \text{ or } w \in L_2\}$$
- *Concatenation:*
$$L_1 \circ L_2 = \{w_1 w_2 \mid w_1 \in L_1 \text{ and } w_2 \in L_2\}$$
- *Star:*
$$L_1^* = \{w_1 w_2 \ldots w_k \mid k \geq 0 \text{ and for each } i = 1, \ldots, k \text{ it holds } w_i \in L_1\}$$

Example 5. Let $L_1 = \{00, 010\}$, $L_2 = \{11\}$. Then

$$L_1 \cup L_2 = \{00, 010, 11\}$$
$$L_1 \circ L_2 = \{0011, 01011\}$$
$$L_1^* = \{\varepsilon, 00, 010, 0000, 00010, 01000, \ldots\}$$

Theorem. *If L_1 and L_2 are two regular languages, then so is $L_1 \cup L_2$.*

Proof. L_1 and L_2 are regular languages therefore there exist automata recognising the languages, call them M_1 and M_2, respectively. We want to construct autmaton M that recognises $L_1 \cup L_2$.

Idea: M "simulates" M_1 and M_2 in parallel. If one of them accepts the input, then so M does. The states of M will be pairs (state of M_1, state of M_2).

Assume $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$. Let then define $M = (Q, \Sigma, \delta, q_0, F)$ as follows.

- Set of states
$$Q = Q_1 \times Q_2 = \{(r_1, r_2) \mid r_1 \in Q_1, r_2 \in Q_2\}$$
- Alphabet is the same, Σ.

- Transition function δ:

$$\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$$

  for any input symbol a.

- Start state $q_0 = (q_1, q_2)$.

- Set of accept states $F \subset Q_1 \times Q_2$:

$$F = \{(r_1, r_2) \mid r_1 \in F_1 \text{ or } r_2 \in F_2\}.$$

□

Practise session

1. Consider the automaton $M = (Q, \Sigma, \delta, q_0, F)$, where $Q = \{q_1, q_2\}$, $\Sigma = \{0, 1\}$, $q_0 = q_1$, $F = \{q_2\}$ and δ is defined in the table:

	0	1
q_1	q_1	q_2
q_2	q_1	q_2

What language the automaton recognises? (The automaton is similar to the one from the lecture, but δ is different.)

2. What language the automaton $M = (Q, \Sigma, \delta, q_0, F)$ recognises? $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{0, 1, 2, \#\}$, start state is q_0, $F = \{q_0\}$ and δ is defined in the table:

	0	1	2	#
q_0	q_0	q_1	q_2	q_0
q_1	q_1	q_2	q_0	q_0
q_2	q_2	q_0	q_1	q_0

3. Design a finite automaton M_0, that recognises the regular language of all binary strings that contain 001 as a substring.

For example, it accepts 001, 000011, 1111001111, but it does not accept 101.

4. Let $\Sigma = \{0, 1\}$ and L be a regular language. Show that L' is also a regular language,

$$L' = \{01w \mid w \in L\}.$$

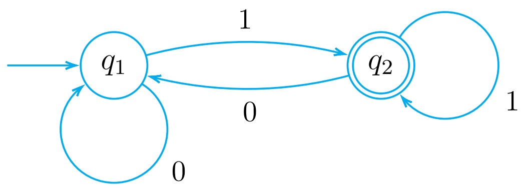
Solutions for the practise session questions

1. Consider the automaton $M = (Q, \Sigma, \delta, q_0, F)$, where $Q = \{q_1, q_2\}$, $\Sigma = \{0, 1\}$, $q_0 = q_1$, $F = \{q_2\}$ and δ is defined in the table:

	0	1
q_1	q_1	q_2
q_2	q_1	q_2

What language the automaton recognises? (The automaton is similar to the one from the lecture, but δ is different.)

Solution. State diagram is the following:

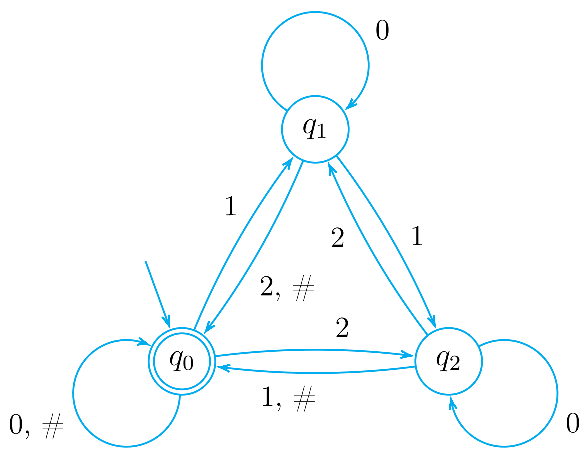


We observe that after reading "1" the automaton always stays in the accept state and after reading "0" it is always not in the accept state. Therefore it accepts all the words that end with "1". The language that M recognises: the set of all binary strings ending with "1".

2. What language the automaton $M = (Q, \Sigma, \delta, q_0, F)$ recognises? $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{0, 1, 2, \#\}$, start state is q_0, $F = \{q_0\}$ and δ is defined in the table:

	0	1	2	#
q_0	q_0	q_1	q_2	q_0
q_1	q_1	q_2	q_0	q_0
q_2	q_2	q_0	q_1	q_0

Solution. State diagram of the automaton:

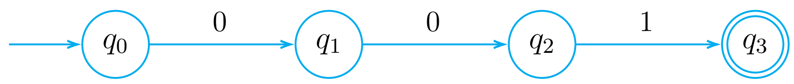


M keeps a sum of the values of inputs modulo 3. Every time is encounters symbol $\#$, it resets counter to zero. M accepts if the sum is 0 modulo 3 (in other words, the sum is divisible by 3).

3. Design a finite automaton M_0, that recognises the regular language of all binary strings that contain 001 as a substring.

For example, it accepts 001, 000011, 1111001111, but it does not accept 101.

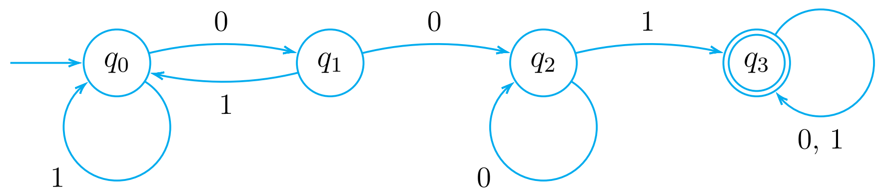
Solution. Let's try:



What happens if we read a symbol, that is not from the pattern 001?

- If M_0 is in the state q_0 or q_1 and reads 1, it should go back to q_0.
- If M_0 is in the state q_2 and reads 0, it means that we saw 000 – stay in q_2.
- If M_0 is in the state q_3 (it means we already encountered the pattern 001), it stays in q_3 forever.

We have:



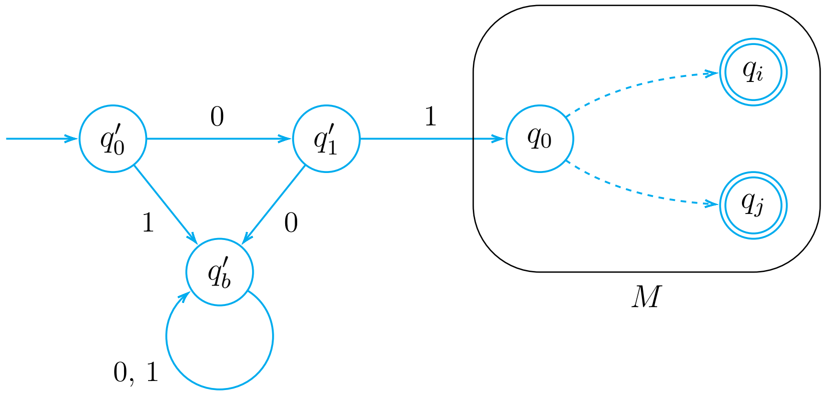
$Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}$, start state is q_0, $F = \{q_3\}$ and δ is

	0	1
q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_3
q_3	q_3	q_3

4. Let $\Sigma = \{0, 1\}$ and L be a regular language. Show that L' is also a regular language,

$$L' = \{01w \mid w \in L\}.$$

Idea of solution. L is a regular language hence there exists automaton M that accepts L. We need to design an automaton that accepts L'. Take state diagram of M and add the part, that handles the prefix 01:



If the two first input symbols are 01, the automaton successfully proceeds to the beginning of M block and handles the rest of the input. If two first input symbols are 00 or the first input symbol is 1, the automaton goes to a "sink" state which never accepts regardless of further input.

Solution. Let $M = (Q, \Sigma, \delta, q_0, F)$ is an automaton accepting L. Let us define $M' = (Q', \Sigma, \delta', q'_0, F')$ in the following way: $Q' = Q \cup \{q'_0, q'_1, q'_b\}$, q'_0 is a new start state, $F' = F$, Σ is the same alphabet and transition function δ' works on new states as follows: $\delta'(q'_0, 0) = q'_1$, $\delta'(q'_0, 1) = q'_b$, $\delta'(q'_1, 0) = q'_b$, $\delta'(q'_1, 1) = q_0$, $\delta'(q'_b, 0) = \delta'(q'_b, 1) = q'_b$ and for old states: for each $q \in Q$ and $a \in \Sigma$ we set $\delta'(q, a) = \delta(q, a)$.

Now we need to prove that M' accepts exactly L'.

1) Assume $w' \in L'$. Then $w' = 01w$, where $w \in L$. From construction of M', after reading 01 from the input, M' will be in state q_0. After reading w it will end up in an accept state of M which is also an accept state of M'. Thus M' accepts w'.

2) Now assume that M' accepts w'. From the construction, it should pass through the state q_0 (because it is the only way from q'_0 to F in M'). It can be seen from the definition of δ', that the only way to get from q'_0 to q_0 is by reading 01. Therefore $w' = 01w$. From here, M' arrives to accept state from q_0 by reading w. This means, that M accepted w, i.e. $w \in L$. We conclude that $w' \in L'$.

Additional exercises

5. What language is accepted by an automaton $M = (Q, \Sigma, \delta, q_0, F)$, where $Q = \{q_1, q_2\}$, $\Sigma = \{0, 1\}$, $q_0 = q_1$, $F = \{q_2\}$ and δ is

	0	1
q_1	q_1	q_2
q_2	q_1	q_1

6. What language is accepted by an automaton $M = (Q, \Sigma, \delta, q_0, F)$, where $Q = \{q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}$, $q_0 = q_1$, $F = \{q_3\}$ and δ is

	0	1
q_1	q_2	q_1
q_2	q_3	q_1
q_3	q_3	q_1

7. Design a finite automaton that accepts binary strings with only one 1.

8. Design a finite automaton that accepts binary strings with even number of zeroes and odd number of ones?

9. Design a finite automaton that accepts binary strings that denote integer numbers divisible by 5. For example 0, 101, 1010, 1111, 11001 are accepted, but 1, 10, 11, 100, 110, 1110 not.

10. Find all the pairs of languages L_1, L_2, for which

$$L_1 \circ L_2 = \{10, 11, 1000, 1010, 10111, 101000\}.$$

Hints. 5., 6. Problems 1 and 2 from this practise session. 7., 8. Problem 3 from this practise session. 9. Consider automaton with 5 states and switch between them according to yet counted residue. 10. There is more than one solution.