cs2001
Introductio Theory of Computation

ecture 7

Ruth Urner

January 30, 2023

Finite Automata

Reading: ITC Section 1.1

Finite Automata

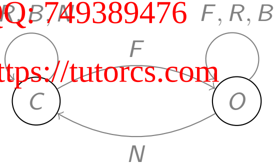
A finite automaton or finite state machine is a simple computational model.

We will work with this model of computation for the next part of this course.

A simple automaton–sliding door example

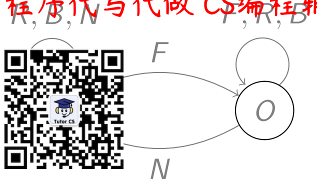
Consider an automatic sliding door sliding door with two pads that receive signals if someone is standing on them:

We can model the controller of the sliding door as a simple automaton:



Here we use: $C = CLOSED$, $O = OPEN$, $F = FRONT$, $R = REAR$, $B = BOTH$, $N = NEITHER$

A simple automaton–sliding door example



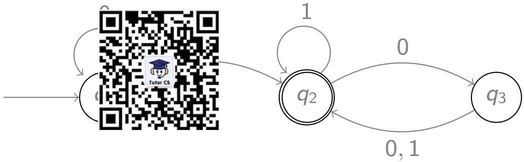
Here we use: $C = CLOSED$, $O = OPEN$, $F = FRONT$, $R = REAR$, $B = BOTH$, $N = NEITHER$

The behavior of the door can be described in terms of the following transition function:

	NEITHER	FRONT	REAR	BOTH
CLOSED	CLOSED	OPEN	CLOSED	CLOSED
OPEN	CLOSED	OPEN	OPEN	OPEN

State diagram of M_1

We can use a state diagram to describe a finite automaton M_1:



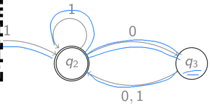
Interpretation of the state diagram: The arrow "coming out of nowhere" going into the leftmost state, signals, that this state is the start state. This automaton can read letters from the alphabet $\Sigma = \{0, 1\}$. Being in some state q, receiving letter σ, the computation finds the outgoing edge from q that has a label σ, and moves along that arrow to a new state.

Examples:

- If we feed the string 10010 to M_1, we move through the states $q_1, q_2, q_3, q_2, q_2, q_3$, and end up in state q_3, which is not an accept state.
- If we feed the string 1101 to M_1, we end up in state q_2, which is an accept state (accept states are the nodes with a double circle).
- If we feed the empty string ϵ to M_1, we end up in state q_1, which is not an accept state.

State diagram of M_1

We can use a st… scribe a finite automaton M_1:



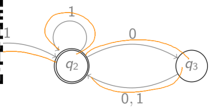
Interpretation of the state diagram: The arrow "coming out of nowhere" going into the leftmost state, signals, that this state is the start state. This automaton can read letters from the alphabet $\Sigma = \{0, 1\}$. Being in some state q, receiving letter σ, the computation finds the outgoing edge from q that has a label σ, and moves along that arrow to a new state.

Examples:

- If we feed the string 10010 to M_1

State diagram of M_1

We can use a st... scribe a finite automaton M_1:



Interpretation of the state diagram: The arrow "coming out of nowhere" going into the leftmost state, signals, that this state is the start state. This automaton can read letters from the alphabet $\Sigma = \{0, 1\}$. Being in some state q, receiving letter σ, the computation finds the outgoing edge from q that has a label σ, and moves along that arrow to a new state.

Examples:

- If we feed the string 10010 to M_1, we move through the states $q_1, q_2, q_3, q_2, q_2, q_3$, and end up in state q_3, which is not an accept state.
- If we feed the string 1101 to M_1, ends up in q_2, string is accepted.

Definition

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

1. Q is a finite set called the set of states,
2. Σ is a finite set called the alphabet,
3. $\delta : Q \times \Sigma \to Q$ is the transition function,
4. $q_0 \in Q$ is the start state, and
5. $F \subseteq Q$ is the set of accept states.

Formal definition of a finite

Definition

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

1. Q is a finite set called the set of states,
2. Σ is a finite set called the alphabet,
3. $\delta : Q \times \Sigma \to Q$ is the transition function,
4. $q_0 \in Q$ is the start state, and
5. $F \subseteq Q$ is the set of accept states.

Formal description of M_1

The above state diagram corresponds to the following formal description:
$M_1 = (Q, \Sigma, \delta, q_1, F)$, where

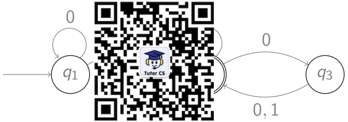
1. $Q = \{q_1, q_2, q_3\}$,
2. $\Sigma = \{0, 1\}$,
3. δ is defined by the following table:

δ	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

4. q_1 is the start state, and
5. $F \subseteq \{q_2\}$.

Given the description of an automaton, we can ask: which strings will lead to an accept state when fed into the automaton? As we have seen in the example computations with M_1 before, some strings do and others don't. The set of strings that do lead to an accept state form a language over Σ, the language of M_1.

Formal description of M_1



The above state diagram corresponds to the following formal description:
$M_1 = (Q, \Sigma, \delta, q_1, F)$, where

$Q = \{q_1, q_2, q_3\}$

$\Sigma = \{0, 1\}$

q_1 is starting state

$F = \{q_2\}$

δ	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

Language accepted by an automaton

Let Σ be the alphabe[illegible] automaton M. Then we let

$$L(M) = \{w \in \Sigma^k \mid k \in \mathbb{N} \text{ and } w \text{ is accepted by } M\}$$

denote the language of machine M. The is $L(M)$ is the set of all words over Σ that are accepted by machine M.

For the language $A = L(M)$ we also say machine M recognizes (or accepts) A.

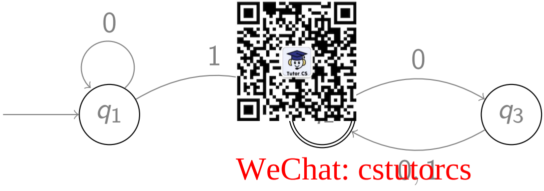
Formal definition of acceptance

Definition

Let $M = (Q, \Sigma, \delta, q_0, F)$ a finite automaton and $\mathbf{w} = w_1 w_2 \ldots w_n$ a string over Σ. We say that M **accepts** $\mathbf{w}$ if there exists a sequence $s_0 s_1 s_2 \ldots s_n$ of states such that

1. $s_0 = q_0$,
2. $\delta(s_i, w_{i+1}) = s_{i+1}$ for $i = 0, 1, \ldots, n-1$, and
3. $s_n \in F$.

Language accepted by M_1

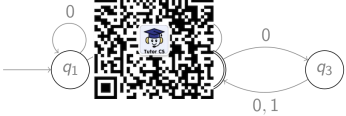


For the machine M_1 we get

$$L(M_1) = \{\mathbf{w} \mid \mathbf{w} \text{ contains at least one 1 and the number of 0s after the last 1 is even}\}$$

Task for you: convince yourself that this is exactly the set of words accepted by this automaton.

Language accepted by M_1



What is the language of M_1 ?

- all words that end with 1 are accepted

$\{ w = w_1 w_2 \dots w_n \in \Sigma^* \mid w_n = 1 \} \subseteq L(M_1)$

- but there are other words (e.g. 100) that don't end with 1 and are also accepted.

Language accepted by M_1

0100 gets accepted

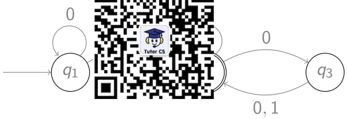
For the machine M_1 we get

$L(M_1) =$

Task for you: figure out what exactly is the set of words accepted by this automaton.

Talking: Ruth Urner

Language accepted by M_1

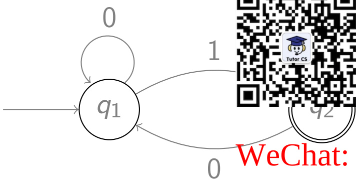


For the machine M_1 we get

$L(M_1) = \{ w \in \{0,1\}^* \mid w$ contains at least one 1 and the number of 0's after the last 1 is even $\}$

Task for you: figure out what exactly is the set of words accepted by this automaton.

Examples automaton M_2



For the machine M_2 we get

$$L(M_2) = \{w \in \{0,1\}^* \mid w \text{ ends with letter } 1\}$$

Task for you: convince yourself that this is exactly the set of words accepted by this automaton.

Examples automaton M_2

Words accepted

001 ✓

0101 ✓

01111

Words not accepted

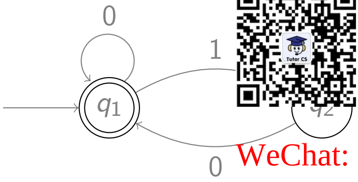
000 X

0100 X

0110 X

$L(M_2) = \{ w \in \{0,1\}^* \mid w \text{ ends with } 1 \}$

Examples automaton M_3



For the machine M_3 we get

$$L(M_3) = \{w \mid w \in \{0,1\}^* \text{ is the empty string or ends with letter } 0\}$$

Task for you: convince yourself that this is exactly the set of words accepted by this automaton.

Examples automaton M_3

Words accepted

0 1 0 ✓

0 1 1 0 ✓

Words not accepted

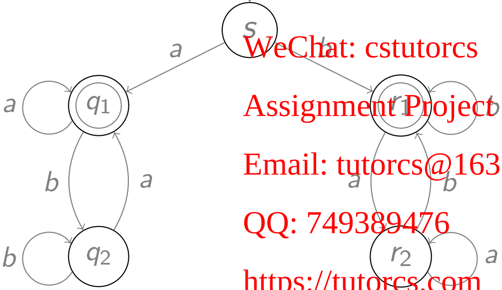
001 ✗

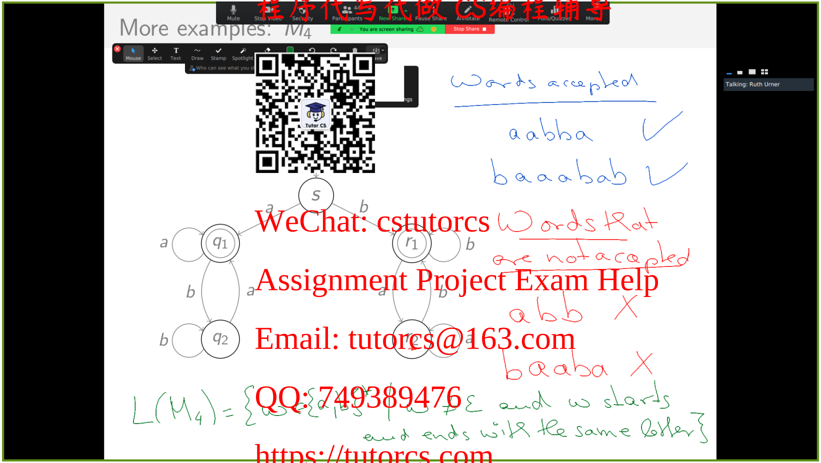
0101 ✗

$L(M_3) = \{\varepsilon\} \cup \{w \in \{0,1\}^* \mid w \text{ ends with } 0\}$

↑ set that contains only the empty word ε

More examples: M_4




More examples: M_4
Words accepted
aabba
baaabab
Words that are not accepted
abb
baba
s
a
b
q_1
q_2
r_1
r_2
$L(M_4) = \{w \in \{a,b\}^* \mid w \neq \varepsilon$ and w starts and ends with the same letter$\}$
Talking: Ruth Urner
You are screen sharing
Stop Share

Design an automaton that verifies if a string contains an odd number of 1s

Design an automaton that verifies if a string contains 001 as a substring

Regular Languages

Reading: ITC Section 1.1

Definition

A language L over some alphabet Σ is called a regular language if there exists a finite automaton M such that $L = L(M)$, that is, if there exists a finite automaton that recognizes it.

Regular language

Σ

Definition

A language L is called a regular language if there exists a finite automaton M such that $L = L(M)$, that is, if there exists a finite automaton that recognizes it.

Examples of regular languages

1. $\{ w \in \{0,1\}^* \mid w$ contains 001 as a substring $\}$

2. $\{ w \in \{0,1\}^* \mid$ the number of 1 is odd $\}$

Talking: Anmoldeep Gadri

Definition:
Let A and B be languages. Then we define the following operations that each form a new language:

- **Union:** $A \cup B = \{\mathbf{w} \mid \mathbf{w} \in A \text{ or } \mathbf{w} \in B\}$.

- **Concatenation:** $A \circ B = \{\mathbf{wv} \mid \mathbf{w} \in A \text{ and } \mathbf{v} \in B\}$.

- **Star:**
  $A^* = \{\mathbf{w}_1\mathbf{w}_2 \ldots \mathbf{w}_k \mid k \geq 0 \text{ and } \mathbf{w}_i \in A \ \forall\, i \in \{0, 1, \ldots, k\}\}$

in particular for an alphabet Σ, the set Σ^* is the set of all finite words over Σ.

Regular o

Ex: $A = \{01, \ldots 1\}$
$B = \{1, \ldots$

$A \times B \ni (01, 110)$ not $\in \{0,1\}^*$

Definition:
Let A and B … Then we define the following operations that each form a new language:

- **Union:** $A \cup B = \{w \mid w \in A \text{ or } w \in B\}$.

Ex $A \cup B = \{01, 001, 0001, 1, 11, 110, 10\}$

- **Concatenation:** $A \circ B = \{wv \mid w \in A \text{ and } v \in B\}$

Ex $A \circ B = \{011, 0011, 00011, 0111, 00111, 000111, 01110, 001110, 0001110, 010, 0010, 00010\}$

- **Star:**
$A^* = \{w_1 w_2 \ldots w_k \mid k \geq 0 \text{ and } w_i \in A \ \forall i \in \{0, 1, \ldots, k\}\}$

Ex $A^* = \{\varepsilon, 01, 001, 0001, 0101, 00101, 01001, 001001, 0010001, 0001001, 010101, 0100101, \ldots\}$

infinite set of words...

Talking: Ruth Urner

Regular operations

In particular for an alphabet Σ, the set Σ^* is the set of all finite words over Σ.

$\Sigma = \{0,1\}$

Ex: $A = \{01, 001, 0001\}$

$B = \{1, 11, 110, 10\}$

$A \times B \ni (01, 110)$ not $\in \{0,1\}^*$

Definition:
Let A and B be languages. Then we define the following operations that each form a new language:

- **Union:** $A \cup B = \{w \mid w \in A \text{ or } w \in B\}$.

  Ex $A \cup B = \{01, 001, 0001, 1, 11, 110, 10\}$

- **Concatenation:** $A \circ B = \{wv \mid w \in A \text{ and } v \in B\}$

  Ex $A \circ B = \{011, 0011, 00011, 0111, 00111, 000111, 01110, 001110, 0001110, 010, 0010, 00010\}$

- **Star:**
  $A^* = \{w_1 w_2 \dots w_k \mid k \geq 0 \text{ and } w_i \in A \ \forall i \in \{0, 1, \dots, k\}\}$

  Ex $A^* = \{\varepsilon, 01, 001, 0001, 0101, 00101, 010001, 00101, 0010001, 0001001, 010101, 0100101, \dots\}$

  infinite set of words...

$B \notin A^*$

$A \notin B^*$

$C = \{0, 1\}$

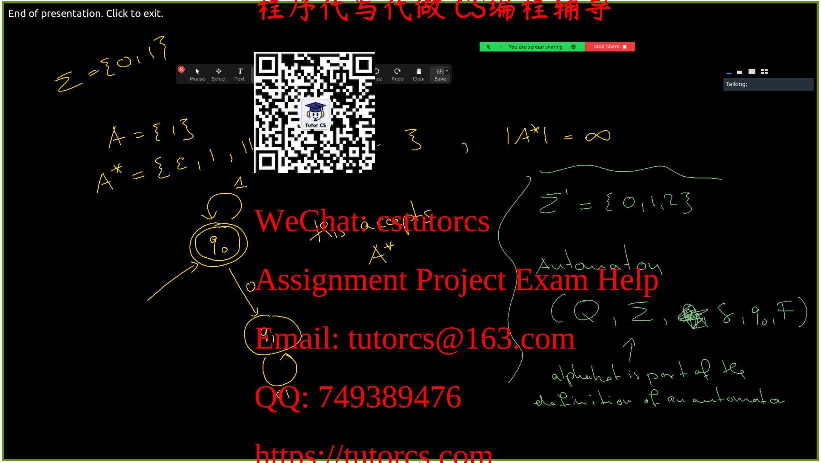
$B \subsetneq C^*$

$A \subsetneq C^*$

$D = \{0, 1, 00, 11\}$

$B \subsetneq D^*$

$A \subsetneq D^*$


End of presentation. Click to exit.
You are screen sharing
Stop Share
Mouse
Select
Text
Redo
Clear
Save
Talking:
Σ = {0,1}
A = {1}
A* = {ε, 1, 11, ...}
|A*| = ∞
Σ' = {0,1,2}
Automaton
(Q, Σ, δ, q0, F)
alphabet is part of the definition of an automaton