COMP30026 Models of Computation

Finite-State Automata

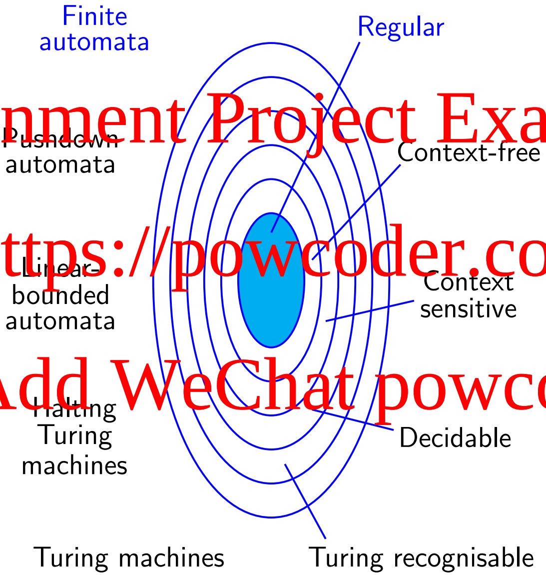
Bach Le / Anna Kalenkova

Lecture Week 7 Part 1

Semester 2, 2021

This Lecture is Being Recorded

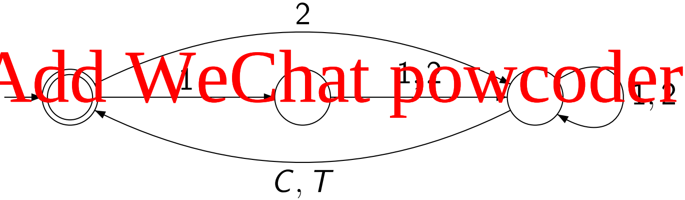
Machines vs Languages



An Example Automaton

Imagine a vending machine selling tea or coffee for \$2. It accepts 1- and 2-dollar coins.

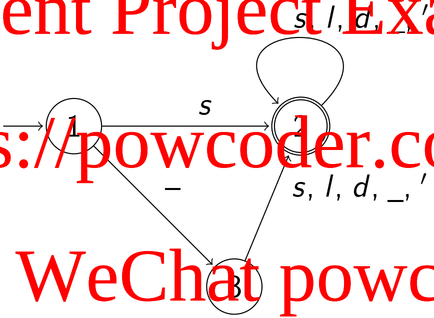
If we let '1' ('2') stand for the event that a 1-dollar (2-dollar) coin enters the coin slot, and C (T) stand for the push of button 'C' ('T') and subsequent delivery of a cup of coffee (tea), then the following automaton describes the acceptable event sequences:



That's "acceptable" from a greedy vending machine owner's point of view, for example, $2T11C22C$ is accepted, but $111C1T$ is not.

Example 2

Here is an automaton for recognising Haskell variable identifier:



s is an abbreviation for $a, \ldots, z$ (the small or lower-case letters)
l is an abbreviation for $A, \ldots, Z$ (the large or upper-case letters)
d is an abbreviation for $0, \ldots, 9$ (the digits)

Formal Definition

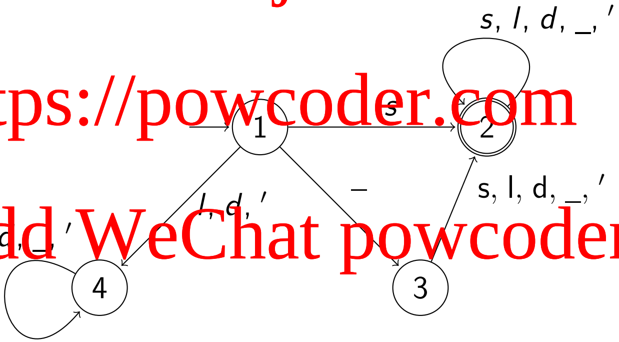
A finite automaton is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

- Q is a finite set of states,
- Σ is a finite alphabet,
- $\delta : Q \times \Sigma \to Q$ is the transition function,
- $q_0 \in Q$ is the start state, and
- $F \subseteq Q$ are the accept states.

Here δ is a total function, that is, δ must be defined for all possible inputs.

Back to Example 2

To make it clear that the transition function is total, we should add a new state 4 and arcs to state 4 from state 1:



Strings and Languages

An alphabet Σ can be any non-empty finite set.

The elements of Σ are the symbols of the alphabet. Usually we choose symbols such as a, b, c, 1, 2, 3, ...

A string over Σ is a finite sequence of symbols from Σ.

We write the concatenation of a string y to a string x as xy.

The empty string is denoted by ϵ.

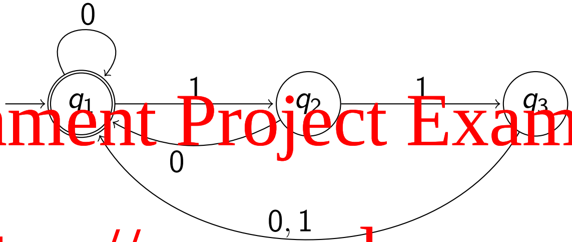
A language (over alphabet Σ) is a (finite or infinite) set of finite strings over Σ.

Σ^* denotes the set of all finite strings over Σ.

Examples of Languages over Alphabet $\Sigma = \{0, 1\}$

- $\emptyset$
- $\{\epsilon\}$
- $\{\epsilon, 0, 1\}$
- $\{00, 01, 10, 11\}$
- $\{\epsilon, 0, 00, 000, \dots\}$
- $\{\epsilon, 0, 1, 00, 11, 000, 111, \dots\}$
- $\{\epsilon, 01, 0011, 000111, \dots\}$
- $\{w \mid w$ contains odd number of $0\}$
- $\{w \mid$ the length of w is a multiple of $3\}$
- $\{w \mid w$ is not empty string$\}$
- $\{w \mid w$ does not contain $001\}$
- Σ^*

Example 3



The automaton M_1 (above) can be described precisely as

$$M_1 = (\{q_1, q_2, q_3\}, \{0, 1\}, \delta, q_1, \{q_1\}) \quad \text{with}$$

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

$$L(M_1) = \left\{ w \;\middle|\; \begin{array}{l} w \text{ is } \epsilon, \textit{ or } \text{ends with '0', } \textit{or } \text{the number of} \\ \text{'1' symbols ending } w \text{ is a multiple of 3} \end{array} \right\}$$

is the language recognised by M_1.

What does it mean for an automaton to accept a string?

Let $M = (Q, \Sigma, \delta, q_0, F)$ and let $w = v_1 v_2 \cdots v_n$ be a string from Σ^*.

M accepts w iff there is a sequence of states $r_0, r_1, \ldots, r_n$, with each $r_i \in Q$, such that

1. $r_0 = q_0$
2. $\delta(r_i, v_{i+1}) = r_{i+1}$ for $i = 0, \ldots, n-1$
3. $r_n \in F$

M recognises language A iff $A = \{w \mid M \text{ accepts } w\}$.

Exercise

Consider the alphabet $\Sigma = \{0, 1\}$. Build an automaton over Σ that recognises a language that contains all strings of odd length and only them.

A language is regular iff there is a finite automaton that recognises it.

We shall soon see that there are languages which are not regular.

Regular Operations

Remember that, to us, a language is simply a set of strings.

Let A and B be languages. The regular operations are:

- **Union:** $A \cup B$
- **Concatenation:** $A \circ B = \{xy \mid x \in A, y \in B\}$
- **Kleene star:** $A^* = \{x_1 x_2 \cdots x_k \mid k \geq 0, \text{ each } x_i \in A\}$

Note that the empty string, ϵ, is always in A^*.

Let $A = \{\mathtt{aa}, \mathtt{abba}\}$ and $B = \{\mathtt{a}, \mathtt{ba}, \mathtt{bba}, \mathtt{bbba}, \ldots\}$.

$A \cup B = \{\mathtt{a}, \mathtt{aa}, \mathtt{abba}, \mathtt{ba}, \mathtt{bba}, \mathtt{bbba}, \ldots\}$.

$A \circ B = \{\mathtt{aaa}, \mathtt{abbaa}, \mathtt{aaba}, \mathtt{abbaba}, \mathtt{aabba}, \mathtt{abbabba}, \ldots\}$.

$$A^* = \left\{ \begin{array}{l} \epsilon, \mathtt{aa}, \mathtt{abba}, \mathtt{aaaa}, \mathtt{aaabba}, \mathtt{abbaaa}, \mathtt{abbaabba}, \\ \mathtt{aaaaaa}, \mathtt{aaaaabba}, \mathtt{aaabbaaa}, \mathtt{aaabbaabba}, \ldots \end{array} \right\}.$$

The regular languages are closed under the regular operations.

It will be easier to show this after we have considered non-deterministic automata.