COMP30026 Models of Computation

Finite-State Automata

Lecture Week 7 Part

Semester 2, 2021

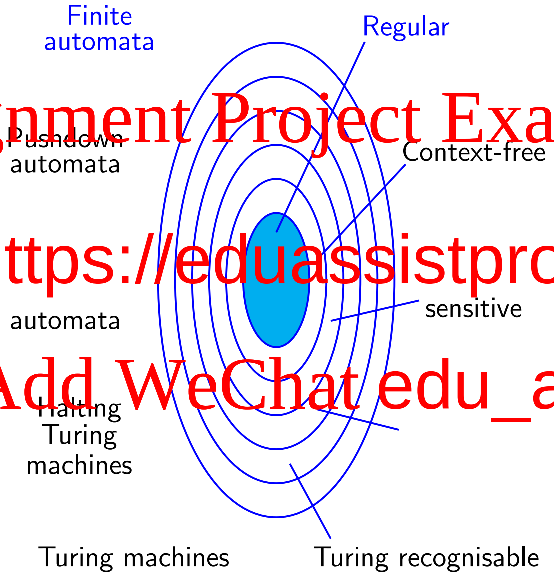
This Lecture is Being Recorded

Assignment Project Exam Help

https://eduassistpro.github.i

Add WeChat edu_assist_pr

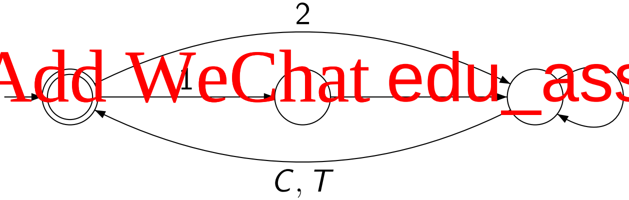
Machines vs Languages



An Example Automaton

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

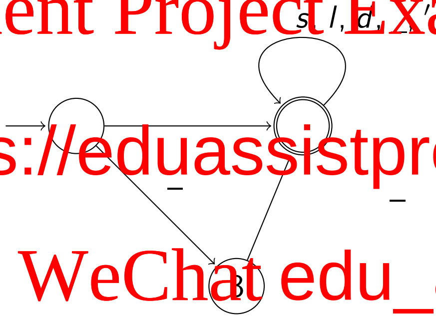
If we let '1' ('2') stand for the event that a 1-dollar (2-dollar) coin enters the coin slot, and C (T) stand for the push of button 'C' ('T') and subs owing automa



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)

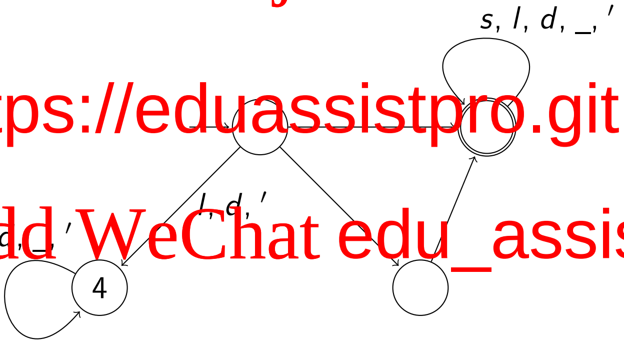
Formal Definition

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

- Q is a finite set of states,
- Σ is a fi
- δ :
- $q_0 \in Q$ is the start state, and
- $F \subseteq Q$ are the accept states.

Here δ is a total function, that is, δ mu
inputs.

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

We write t xy.

The empty string is denoted by ϵ.

A language (over alphabet Σ) is a (finite or infinit strings over Σ.

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

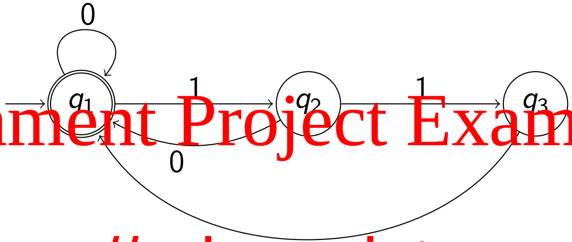
Examples of Languages over Alphabet $\Sigma = \{0, 1\}$

- $\emptyset$
- $\{\epsilon\}$

- $\{\epsilon, 0, 1\}$
- $\{0$
- $\{\epsilon,$

- $\{\epsilon,$
- $\{\epsilon, 01, 0011, 000111, \dots\}$
- $\{w \mid w$ 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^*



The auto

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

		q_2
		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.

Acceptance and Recognition, Formally

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 acce ... ith each

$r_i \in Q$, s

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

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

Exercise

Conside
recognis n and only
them.

Regular Languages

A langua

We shall s ar.

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

Let A a

- **Un**

- **Concatenation:** $A \circ B = \{xy \mid x$
- **Kleene star:** $A^* = \{x_1 x_2 \cdots x_k \mid k$

Note that the empty string, ϵ, is always in

Regular Operations: Example

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

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

$A \circ B$

$A^* = \{$

$\mathtt{aaaaaa}, \mathtt{aaaaabba}, \mathtt{aaabbaaa}$

The regular languages are closed under the regular

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