Computability Theory

Sunday, October 18, 2020 10:34 PM

Computability Theory.

Motivation. What kind of problems can we solve with a computer?

Automata An automaton is a mathematical model of a computer.

Why build models?

-> Mathematical simplicity

-> Intellectual robustness (i.e., generalization)

Types of models

-> Finite automata (finite resources, actually buildable machines)

-> Turing machines (upper bounds for what <u>could</u> be built)

Formal Language Theory

Alphabet Σ, set of possible characters that are valid

String Finite sequence of characters drawn from Σ

The empty string ε has no characters.

Language A language is a set of strings

We say L is a language over Σ iff it is a set of strings over Σ

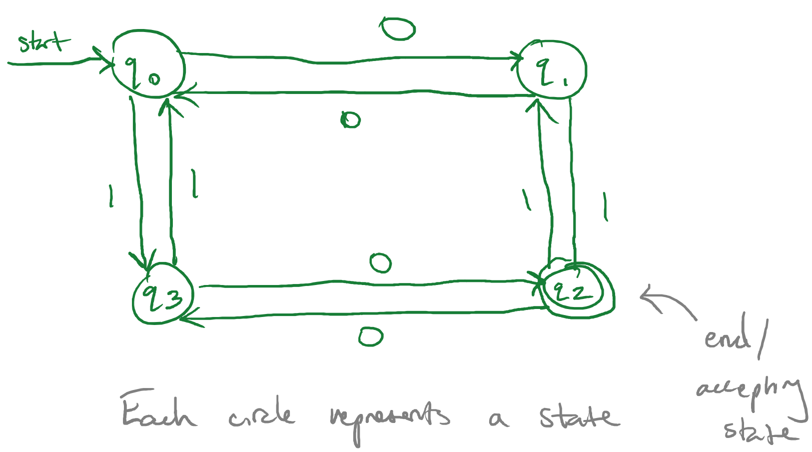
The set of all strings composed from Σ is Σ^*.

$$L \subseteq \Sigma^*$$

Finite Automata

Defn. A finite automaton is a simple type of mathematical machine for determining whether a string is contained within some language.

Ex.



Each circle represents a state

Each arrow is a transition

Automaton operates on string and returns yes or no

(reading chars from string sequentially)

Outputs yes if <u>last</u> char lands on accepting state

0 1 0 1 1 0

$q_0 \to q_1 \to q_2 \to q_3 \to q_0 \to q_3 \to q_2$

Language The language of an automaton is the strings it accepts.

If D is an automaton over the alphabet Σ, then $\mathcal{L}(D)$ is defined as

$$\mathcal{L}(D) = \{ w \in \Sigma^* \mid D \text{ accepts } w \}$$

DFA A DFA is a Deterministic Finite Automaton

For each state in the DFA, there must be exactly one transition defined for each symbol in the alphabet

There is a unique start state

There are zero more accepting states

At each point in execution, a DFA can only remember what state it is in.