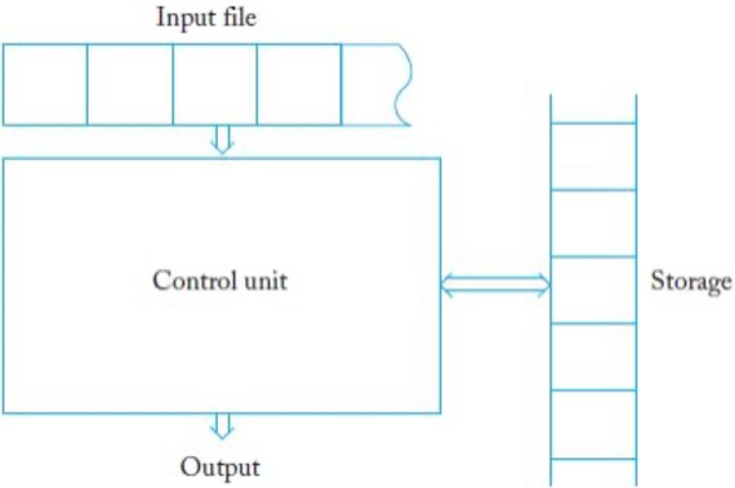
Foundations of Computing

Lecture 1

Arkady Yerukhimovich

January 17, 2023

Modeling Computation



Outline

1. Strings, Languages, and Automata
2. Deterministic Finite Automata (DFA)

Strings

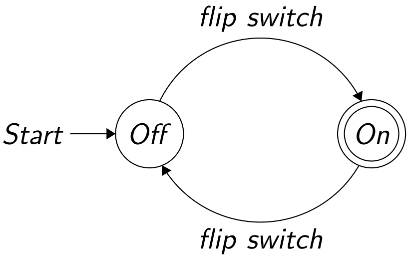
- Alphabet Σ: Set of symbols
  - Ex: $\Sigma = \{a, b\}$, $\Sigma = \{0, 1\}$
- String: finite sequence of symbols from Σ
  - ex: $v = aba$, $w = abaaa$
  - ex: $v = 001$, $w = 11001$
  - λ – empty string
  - Length of a string: $|v| = 3$ and $|\lambda| = 0$
- Operations on Strings
  - Concatenation: $vw = abaabaaa$
  - Reverse: $w^R = aaaba$
  - Repeat: $v^2 = abaaba$ and $v^0 = \lambda$

Languages

- Language L: Set of strings
  - Usually meant to capture strings that satisfy some property
- More formally:
  Σ^* = set of all strings formed by concatenating zero or more symbols from Σ
  - Ex: If $\Sigma = \{0, 1\}$ then $\Sigma* = \{$ all binary strings, including empty string$\}$
- Examples: $L_1 = \{ab, aa\}$ and $L2 = \{a^n b^n : n \geq 0\}$

We will define computation as deciding membership in a language.

A Simple Example: A Light Switch



Viewing this as a language

$L_{light} = \{$ set of all flip sequences resulting in the light being on$\}$
$L_{light} = \{$1 flip, 3 flips, 5 flips, ...$\}$

Automata

- An automaton is an abstract model of a computing device
- An automaton consists of:
  - An input mechanism
  - A control unit
  - Possibly, a storage mechanism
  - Possibly, an output mechanism
- Control unit can be in any number of internal states, as determined by a next-state or transition function
- There are a finite number of states

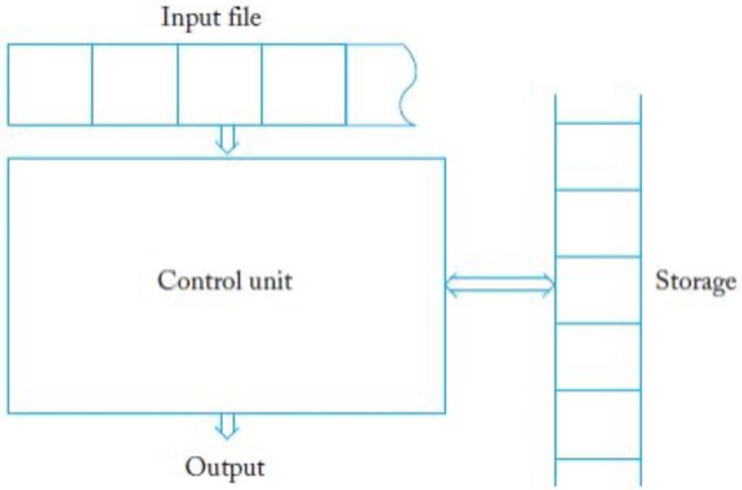
Automata we will study

- Finite Automata (Deterministic and Non-deterministic)
  - These model Finite State Machines with no memory
- Pushdown automata
  - Add the simplest form of memory to a Finite state machine
- Turing Machines
  - These model today's computers in terms of computational ability
  - This will be the main model of computation used in computability and complexity theory

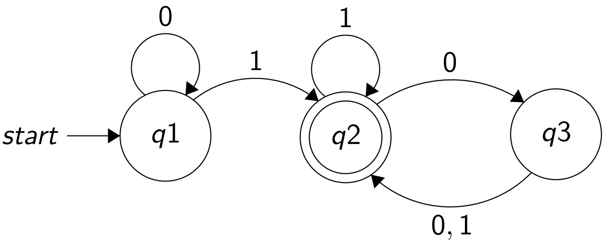
Outline

1. Strings, Languages, and Automata
2. Deterministic Finite Automata (DFA)

Modeling Computation



Finite Automata by Picture



Computation on string $x = 1101$

1. Start in state $q1$
2. read 1, follow transition to $q2$
3. read 1, follow transition to $q2$
4. read 0, follow transition to $q3$
5. read 1, follow transition to $q2$
6. "accept" (output 1) because $q2$ is an accept state

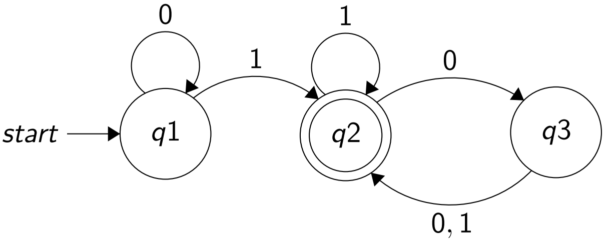
Finite Automaton – Formal Definition

Finite Automaton

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 input alphabet
- $\delta : Q \times \Sigma \to Q$ is the transition function
- $q_0 \in Q$ is the start state
- $F \subseteq Q$ is the set of accept states

Example Automaton



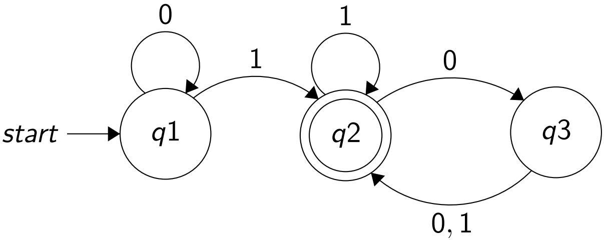
Defining this formally: $M = (Q, \Sigma, \delta, q1, F)$

- $Q = \{q1, q2, q3\}$
- $\Sigma = \{0, 1\}$
- $\delta =$

	0	1
q1	q1	q2
q2	q3	q2
q3	q2	q2

- $q1$ is the start state
- $F = \{q2\}$

Language accepted by M



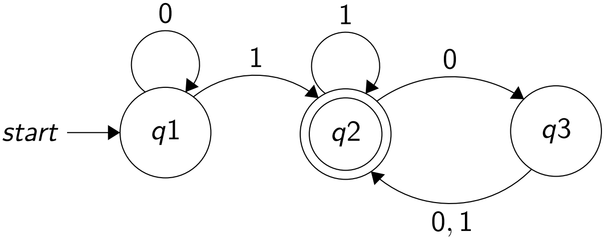
Accepting a string

- M accepts a string x (over Σ) if $M(x)$ stops in an accept state
- We already saw that this M accepts 1101
- What other strings does M accept?

Accepting a language

- M accepts a language L if it accepts ALL strings in L and NO strings not in L
- Every M accepts exactly one language $L(M)$

What language does M accept?



$L(M)$:

- String must contain at least one 1
- After the last 1, there must be an even number of 0's

Why study this?

- Finite Automata are one of the most basic models of computation
- Turns out they capture some very useful functionalities
  - We will see next week, that finite automata correspond to regular expressions

Next...

- Labs this week:
  - Review of proof techniques
  - Review languages/strings/graphs
  - In-class exercises
- Thursday lecture:
  - More about finite automata and their properties
- Your to do list:
  - Sign up for Piazza
  - (optional) Download and install JFLAP (check tutorial on course webpage)