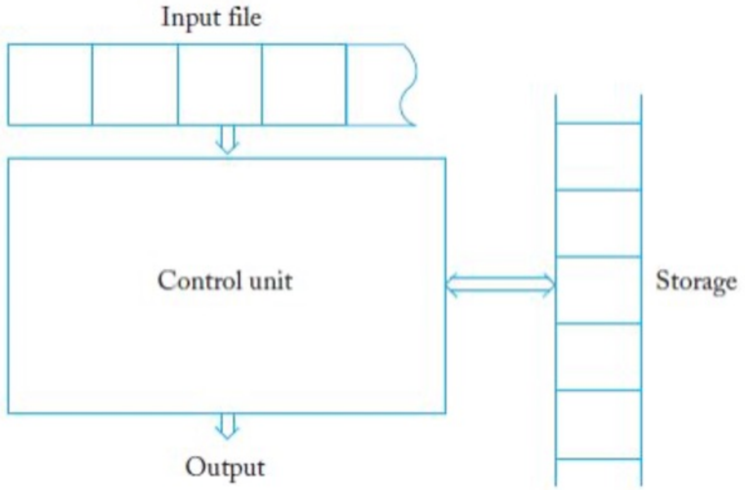
Foundations of Computing

Lecture 1

Arkady Yerukhimovich

January 14, 2025

Modeling Computation



Outline

Strings

- Alphabet Σ: Set of symbols
  - Ex: $\Sigma = \{a, b\}$, $\Sigma = \{0, 1\}$

Strings

- Alphabet Σ: Set of symbols
  - Ex: $\Sigma = \{a, b\}$, $\Sigma = \{0, 1\}$
- String: sequence of symbols from Σ
  - ex: $v = aba$, $w = abaaa$
  - ex: $v = 001$, $w = 11001$
  - λ, ϵ – empty string
  - Length of a string: $|v| = 3$ and $|\lambda| = 0$

Strings

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

Strings

- Alphabet Σ: Set of symbols
  - Ex: $\Sigma = \{a, b\}$, $\Sigma = \{0, 1\}$
- String: sequence of symbols from Σ
  - ex: $v = aba$, $w = abaaa$
  - ex: $v = 001$, $w = 11001$
  - λ, ϵ – empty string
  - Length of a string: $|v| = 3$ and $|\lambda| = 0$
- Operations on Strings
  - $v = aba$, $w = abaaa$
  - Concatenation: $vw = abaabaaa$
  - Reverse: $w^R = aaaba$
  - Repeat: $v^2 = abaaba$ and $v^0 = \epsilon$
- Kleene Closure
  - For an alphabet Σ, Σ^* is the set of all strings formed by concatenating zero or more symbols from Σ
  - Ex: If $\Sigma = \{0, 1\}$ then Σ^* is the set of all binary strings, including ϵ

Languages

- Language L: Set of strings
  - We say that any $s \in L$ is in the language

Languages

- Language L: Set of strings
  - We say that any $s \in L$ is in the language
- Examples:
  - $L_1 = \{ab, aa\}$

Languages

- Language L: Set of strings
  - We say that any $s \in L$ is in the language
- Examples:
  - $L_1 = \{ab, aa\}$
  - $L_2 = \{a^n b^n : n \geq 0\}$

Languages

- Language L: Set of strings
  - We say that any $s \in L$ is in the language
- Examples:
  - $L_1 = \{ab, aa\}$
  - $L_2 = \{a^n b^n : n \geq 0\}$
  - The language of all English sentences

Languages

- Language L: Set of strings
  - We say that any $s \in L$ is in the language
- Examples:
  - $L_1 = \{ab, aa\}$
  - $L_2 = \{a^n b^n : n \geq 0\}$
  - The language of all English sentences
  - For any alphabet Σ, Σ^* is a language

Languages

- Language L: Set of strings
  - We say that any $s \in L$ is in the language
- Examples:
  - $L_1 = \{ab, aa\}$
  - $L_2 = \{a^n b^n : n \geq 0\}$
  - The language of all English sentences
  - For any alphabet Σ, Σ^* is a language
- Size or a language:
  - A language L has size $|L|$

Languages

- Language L: Set of strings
  - We say that any $s \in L$ is in the language
- Examples:
  - $L_1 = \{ab, aa\}$
  - $L_2 = \{a^n b^n : n \geq 0\}$
  - The language of all English sentences
  - For any alphabet Σ, Σ^* is a language
- Size or a language:
  - A language L has size $|L|$
  - $|L|$ can be finite – e.g. $|L_1| = 2$

Languages

- Language L: Set of strings
  - We say that any $s \in L$ is in the language
- Examples:
  - $L_1 = \{ab, aa\}$
  - $L_2 = \{a^n b^n : n \geq 0\}$
  - The language of all English sentences
  - For any alphabet Σ, Σ^* is a language
- Size or a language:
  - A language L has size $|L|$
  - $|L|$ can be finite – e.g. $|L_1| = 2$
  - $|L|$ can be infinite – e.g., $L_2, L_3, L_4 = \infty$

We will often be interested in languages recognized by a particular "computer".

Deciding Languages vs. Computing Functions

- Deciding Languages:
  - We will often want to "decide" a language L.

Deciding Languages vs. Computing Functions

- Deciding Languages:
  - We will often want to "decide" a language L.
  - Given a string x, output whether $x \in L$ or not
- Computing Functions:
  - Given alphabet Σ

Deciding Languages vs. Computing Functions

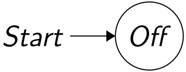
- Deciding Languages:
  - We will often want to "decide" a language L.
  - Given a string x, output whether $x \in L$ or not
- Computing Functions:
  - Given alphabet Σ
  - Define a function $f_L : \Sigma^* \to \{0, 1\}$ s.t. $f_L(x) = 1$ iff $x \in L$

Remember

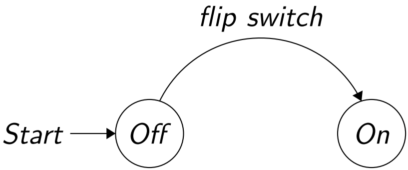
Deciding the language L is the same as computing F_L

Outline

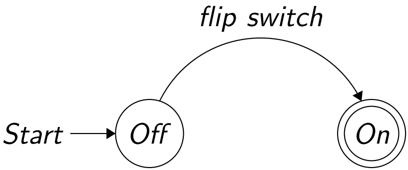
A Simple Example: A Light Switch



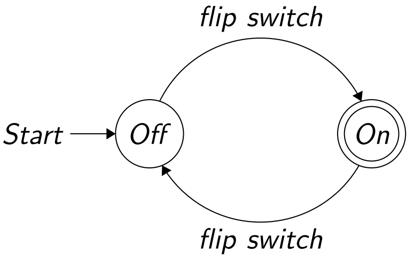
A Simple Example: A Light Switch



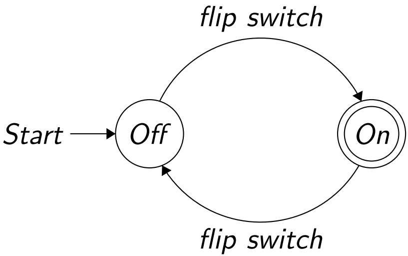
A Simple Example: A Light Switch



A Simple Example: A Light Switch



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} = \{$1 flip, 3 flips, 5 flips, ...$\}$

Finite Automata

- An automaton is an abstract model of a computing device

Finite Automata

- An automaton is an abstract model of a computing device
- An automaton consists of:
    - An input mechanism

Finite Automata

- An automaton is an abstract model of a computing device
- An automaton consists of:
  - An input mechanism
  - A control unit

Finite 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

Finite 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

Finite 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 transitions between internal states, as determined by a next-state or transition function
- There are a finite number of states

Finite 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 transitions between internal states, as determined by a next-state or transition function
- There are a finite number of states

A note on input size

- An automaton must be able to accept input of arbitrary length.

Finite 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 transitions between internal states, as determined by a next-state or transition function
- There are a finite number of states

A note on input size

- An automaton must be able to accept input of arbitrary length.
- The input may be much larger than the number of states.

Finite 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 transitions between internal states, as determined by a next-state or transition function
- There are a finite number of states

A note on input size

- An automaton must be able to accept input of arbitrary length.
- The input may be much larger than the number of states.
- Our goal is a single machine that works for all inputs

Automata we will study

- Finite Automata (Deterministic and Non-deterministic)
    - These model Finite State Machines with no external memory

Automata we will study

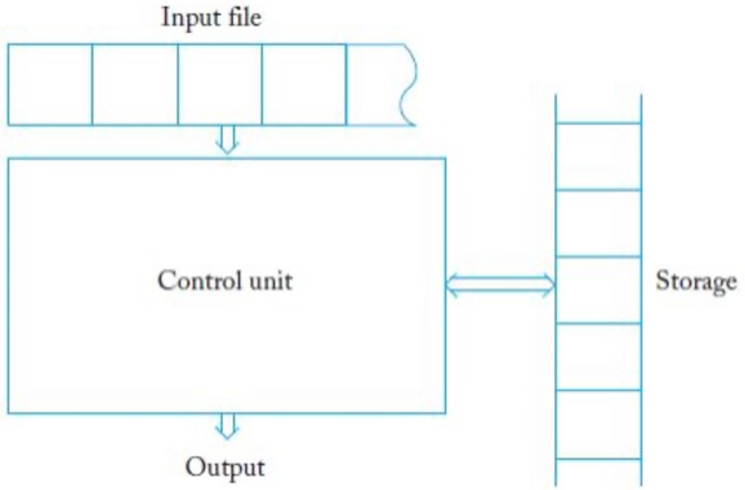
- Finite Automata (Deterministic and Non-deterministic)
  - These model Finite State Machines with no external memory
- Pushdown automata
  - Add the simplest form of memory to a Finite State Machine

Automata we will study

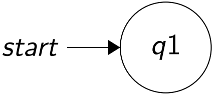
- Finite Automata (Deterministic and Non-deterministic)
    - These model Finite State Machines with no external memory
- Pushdown automata
    - Add the simplest form of memory to a Finite State Machine
- Turing Machines
    - Add unrestricted memory to a Finite State Machine

Outline

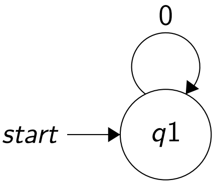
What is an Automaton



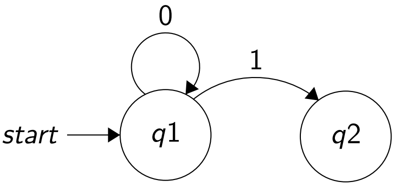
Finite Automata by Picture



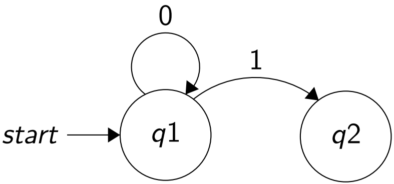
Finite Automata by Picture



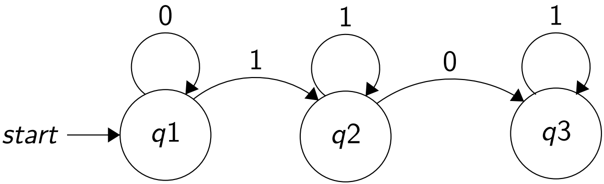
Finite Automata by Picture



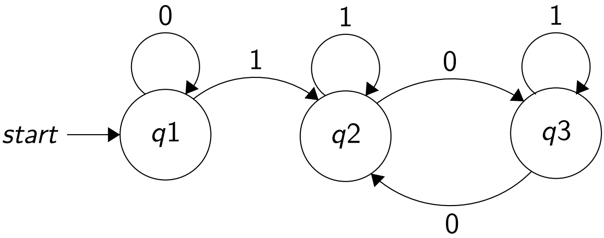
Finite Automata by Picture



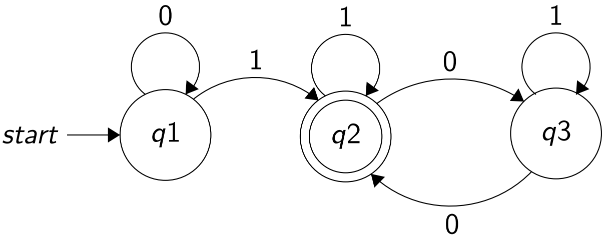
Finite Automata by Picture



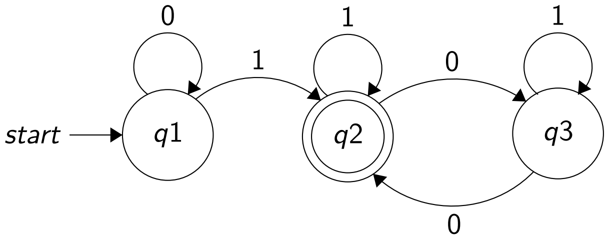
Finite Automata by Picture



Finite Automata by Picture

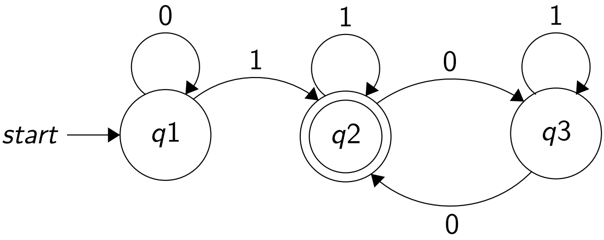


Finite Automata by Picture



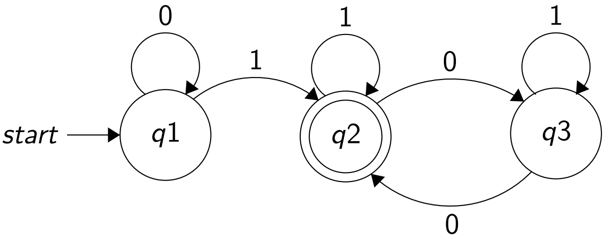
Computation on string $x = 1101$

Finite Automata by Picture



Computation on string $x = 1101$

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 $q3$
6. "reject" (output 0) because $q3$ is not an accept state

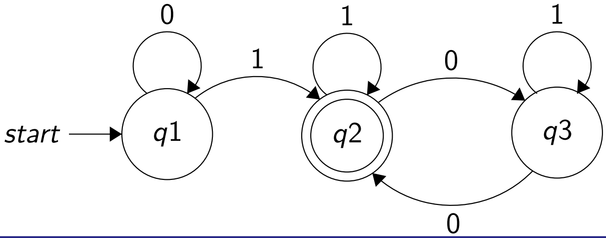
Finite Automaton – Formal Definition

Finite Automaton

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

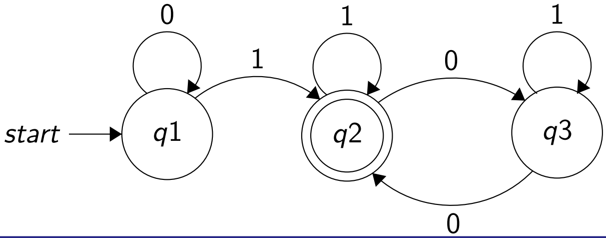
- Q is a finite set of states
- Σ is a finite input alphabet
- $\delta : Q \times \Sigma \to Q$ is the transition function
- $q_0 \in Q$ is the start state
- $F \subseteq Q$ is the set of accept states

Example Automaton



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

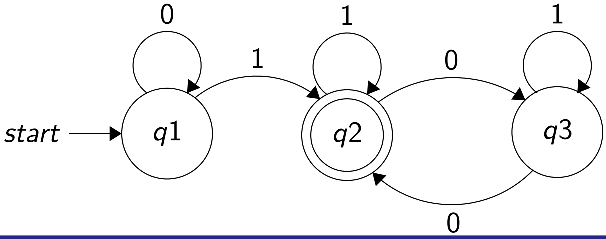
Example Automaton



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

- $Q = \{q1, q2, q3\}$

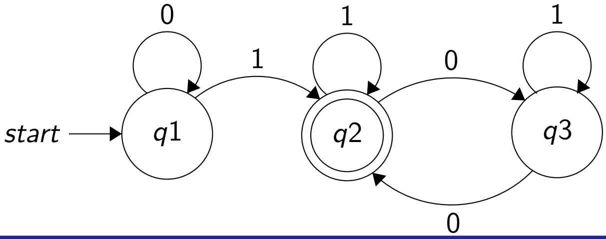
Example Automaton



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

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

Example Automaton

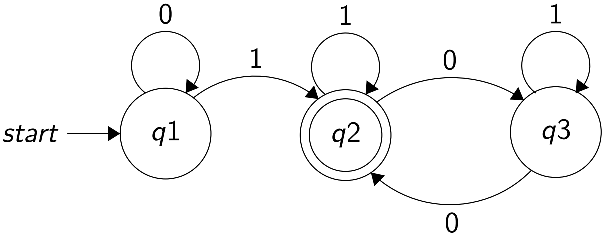


Defining this formally: $M = (Q, \Sigma, \delta, q1, F)$

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

	0	1
q1	q1	q2

Example Automaton

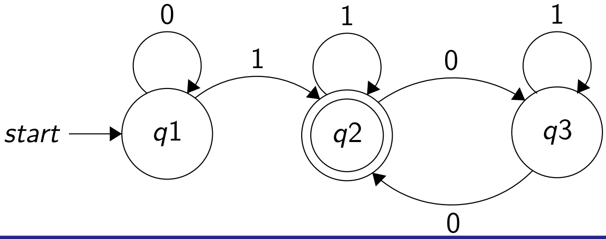


Defining this formally: $M = (Q, \Sigma, \delta, q1, F)$

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

	0	1
q1	q1	q2
q2	q3	q2
q3	q2	q3

Example Automaton



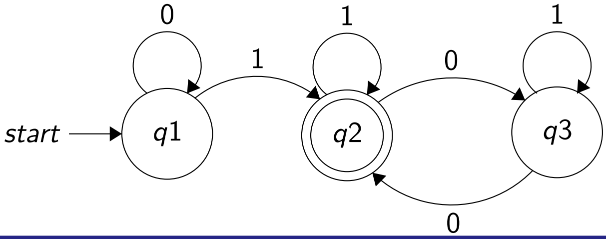
Defining this formally: $M = (Q, \Sigma, \delta, q1, F)$

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

	0	1
q1	q1	q2
q2	q3	q2
q3	q2	q3

- $q1$ is the start state

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	q3

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