Computational Models - Lecture 1[1]

Handout Mode

Iftach Haitner and Yishay Mansour.

Tel Aviv University.

February 17/February 19, 2014

[1] Based on frames by Benny Chor, Tel Aviv University, modifying frames by Maurice Herlihy, Brown University.

Talk Outline

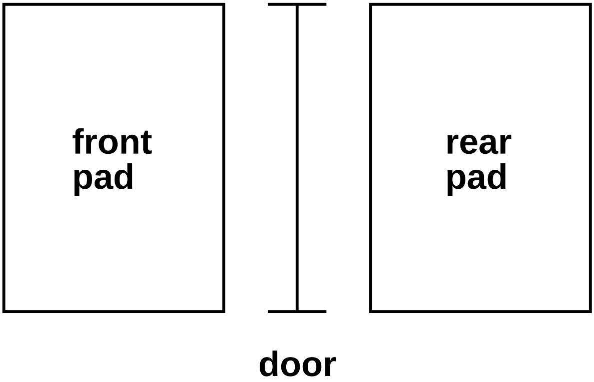
- Finite automata and regular languages
- Regular operations

- Sipser's book, chapter 1.1

Part I

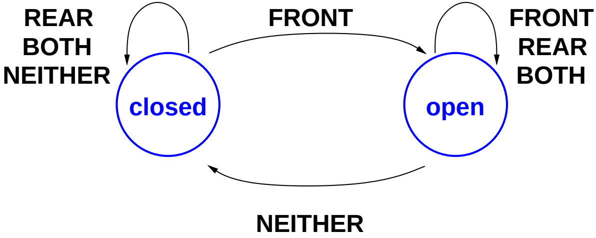
Finite Automata

Example: A One-Way Automatic Door



- open when person approaches
- hold open until person clears
- don't open when someone standing behind door

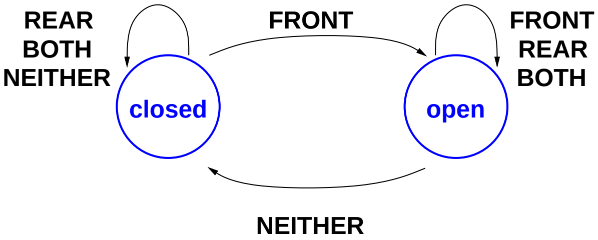
The Automatic Door as DFA



- States:
  - OPEN
  - CLOSED
- Sensor:
  - FRONT: someone on front pad
  - REAR: someone on rear pad
  - BOTH: someone(s) on both pads
  - NEITHER no one on either pad.

The Automatic Door as DFA

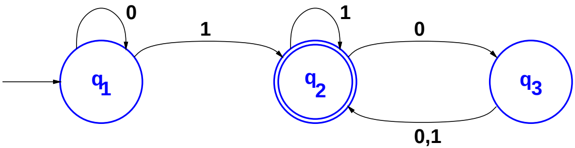
A DFA is Deterministic Finite Automata



	neither	front	rear	both
closed	closed	open	closed	closed
open	closed	open	open	open

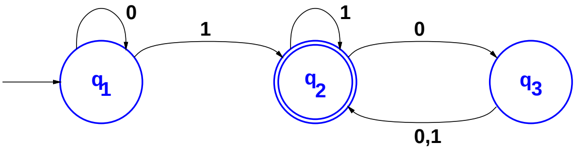
DFA: Informal definition

The machine M_1:



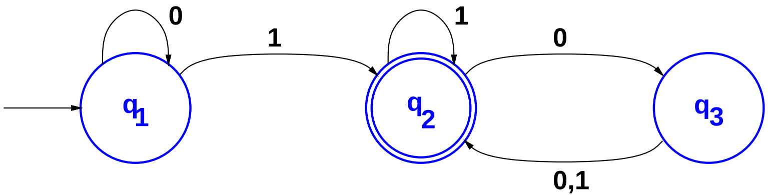
- States: q_1, q_2, and q_3.
- Start state: q_1 (arrow from "outside").
- Accept state: q_2 (double circle).
- State transitions: arrows tagged with letters.

DFA: Informal definition (cont.)



- On an input string
  - DFA begins in start state q_1
  - after reading each symbol, DFA makes state transition with matching label.
- After reading last symbol, DFA produces output:
  - accept if DFA is an accepting state.
  - reject otherwise.

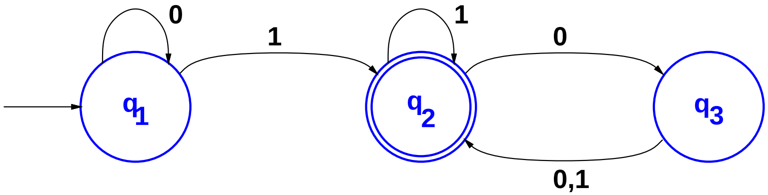
DFA: Informal definition (cont..)



What happens on the following input strings:

- 1101
- 0010
- 01100
- In general?!

DFA: Informal definition (cont...)



This DFA accepts

- All input strings that end with a 1
- All input strings that contain at least one 1, and end with an even number of 0's
- No other strings

Proof: ?

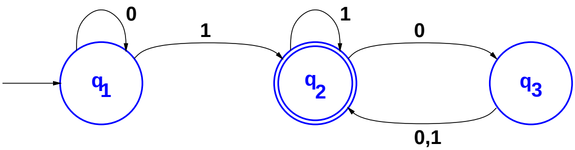
DFA - Formal Definition

Definition 1

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

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

Back to M_1



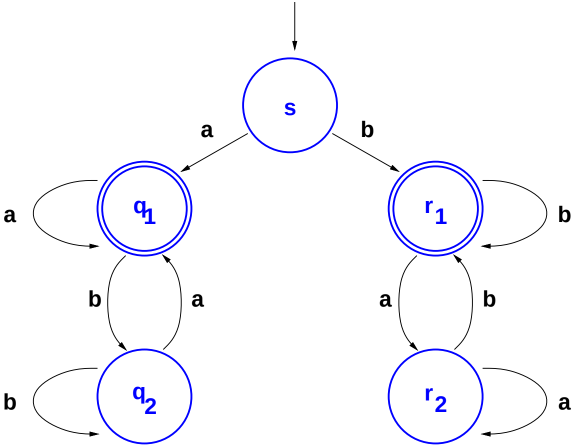
$M_1 = (Q, \Sigma, \delta, q_1, F)$ where

- $Q = \{q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}$,
- the transition function δ is

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

- q_1 is the start state
- $F = \{q_2\}$.

Another Example



A Formal Model of Computation

Definition 2

$M = (Q, \Sigma, \delta, q_0, F)$ accepts $w \in \Sigma^*$ if $\widehat{\delta}(q_0, w) \in F$.

Definition 3 ($\widehat{\delta}$)

For DFA $M = (Q, \Sigma, \delta, q_0, F)$, define $\widehat{\delta}\colon Q \times \Sigma^* \mapsto Q$ by

$$\widehat{\delta}(q, w) = \begin{cases} \delta(q, w), & |w| = 1 \\ \delta(\widehat{\delta}(q, w_{1,\ldots,n-1}), w_n), & n = |w| > 1 \\ q, & w = \varepsilon. \end{cases}.$$

Definition 4 (Alternative (and equivalent) definition)

$M = (Q, \Sigma, \delta, q_0, F)$ accepts $w = w_1 w_2 \ldots w_n$, if exists $r_0, \ldots, r_n \in Q$, such that

- $r_0 = q_0$.
- $\delta(r_i, w_{i+1}) = r_{i+1}$, for all $0 \leq i < n$.
- $r_n \in F$.

Languages, words and alphabets

Definition 5

An alphabet Σ is a finite set of letters.

- $\Sigma = \{a, b, c, \ldots, z\}$ – the English alphabet.
- $\Sigma = \{\alpha, \beta, \gamma, \ldots, \zeta\}$ – the Greek alphabet.
- $\Sigma = \{0, 1\}$ – the binary alphabet.
- $\Sigma = \{0, 1, \ldots, 9\}$ – the digital alphabet.

Definition 6

A word (i.e., string) over Σ, is a finite sequence of letters from Σ.

The collection of all strings over Σ is denoted by Σ^*.

For the binary alphabet, ε, 1, 0, 000000000, 1111111000 are all members of Σ^*.

Definition 7

A language over Σ is a (possibly infinite) subset of Σ^*.

Language Examples

- Modern English.
- Ancient Greek.
- All prime numbers, written using digits.
- $\mathcal{A} = \{w \in \{0,1\}^* \colon w$ has at most seventeen 0's$\}$.
- $\mathcal{B} = \{0^n 1^n \colon n \geq 0\}$.
- $\mathcal{C} = \{w \in \{0,1\}^* \colon w$ has an equal number of 0's and 1's$\}$.

The language of a DFA

Definition 8

$\mathcal{L}(M)$, the language of a DFA M, is the set of strings that M accepts.

- M may accept many strings
- M accepts only one language.

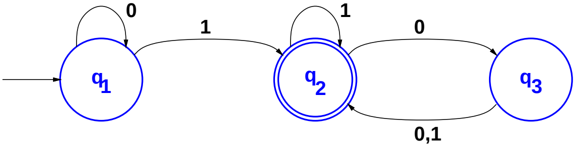
What language does M accept if it accepts no strings?

Definition 9

A language is called regular, if some deterministic finite automaton accepts it.

Proving a language of a DFA

M_1



Theorem 10

$\mathcal{L}(M_1) = \{w10^{2k} : k \geq 0, w \in \Sigma^*\}$

Proof:

Claim 11 (implies the theorem)

- $\mathcal{L}_1 = \{0^k : k \geq 0\}$
- $\mathcal{L}_2 = \{w10^{2k} : k \geq 0, w \in \Sigma^*\}$
- $\mathcal{L}_3 = \{w10^{2k+1} : k \geq 0, w \in \Sigma^*\}$

$x \in \mathcal{L}_i$ iff $\widehat{\delta}(q_1, x) = q_i$.

Proving Claim 11

Proof by induction on word length.

- Induction basis (length 0): Easy to see that hypothesis holds for ϵ (which is the only word of this length).
- Induction step: Assume hypothesis holds for words of length $j \geq 0$.

  Let $x = y\sigma$, where $y \in \Sigma^*$ and $\sigma \in \Sigma$, be a word of length $j + 1$ (hence, $|y| = j$)

  We prove the hypothesis for x, separately for each $i \in \{1, 2, 3\}$

$x \in \mathcal{L}_1$ iff $\widehat{\delta}(q_1, x) = q_1$

- $x \in \mathcal{L}_1 \implies \widehat{\delta}(q_1, x) = q_1$.
  - Assume $x \in \mathcal{L}_1$.
  - $x = 0^{j+1}$, $y = 0^j$ and $\sigma = 0$.
  - Since $y \in \mathcal{L}_1$, by induction hypothesis $\widehat{\delta}(q_1, y) = q_1$
  - By definition, $\widehat{\delta}(q_1, 0) = q_1$.
    Therefore, $\widehat{\delta}(q_1, x) = \widehat{\delta}(\widehat{\delta}(q, y), \sigma) = \widehat{\delta}(q_1, 0) = q_1$.
- $\widehat{\delta}(q_1, x) = q_1 \implies x \in \mathcal{L}_1$.
  - Assume $\widehat{\delta}(q_1, x) = q_1$.
  - Let $q_y = \widehat{\delta}(q_1, y)$ (hence, $\widehat{\delta}(q_1, x) = \widehat{\delta}(q_y, \sigma) = q_1$)
  - It holds that (?) $q_y = q_1$ and $\sigma = 0$
  - By i.h. $y = 0^j$,
    Hence, $x = y\sigma = 0^j 0 = 0^{j+1} \in \mathcal{L}_1$

$x \in \mathcal{L}_2$ iff $\widehat{\delta}(q_1, x) = q_2$

- $x \in \mathcal{L}_2 \implies \widehat{\delta}(q_1, x) = q_2$.
  - Assume $x \in \mathcal{L}_2$.
  - $x = w10^{2k}$ for $k \geq 1$, or $x = w1$.
  - If $x = w1$, then $\sigma = 1$.
    Since $\widehat{\delta}(q_i, 1) = q_2$, for any i, then $\widehat{\delta}(q_1, x) = q_2$.
  - If $x = w10^{2k}$, then $y = w10^{2k-1}$ and $\sigma = 0$
    Hence, $y \in \mathcal{L}_3$.
    By i.h. $\widehat{\delta}(q_1, y) = q_3$
    Thus, $\widehat{\delta}(q_1, x) = \widehat{\delta}(q_3, 0) = q_2$.
- $\widehat{\delta}(q_1, x) = q_2 \implies x \in \mathcal{L}_2$.
  - Assume $\widehat{\delta}(q_1, x) = q_2$
  - Let $q_y = \widehat{\delta}(q_1, y)$ (hence, $\widehat{\delta}(q_y, \sigma) = \widehat{\delta}(q_1, x) = q_2$).
  - If $\sigma = 1$, then $x \in \mathcal{L}_2$ (?)
  - $\sigma = 0 \implies q_y = q_3$
    By i.h. $y = w10^{2k+1}$
    Therefore $x = y\sigma = w10^{2k+1}0 \in \mathcal{L}_2$

$x \in \mathcal{L}_3$ iff $\widehat{\delta}(q_1, x) = q_3$

- $x \in \mathcal{L}_3 \implies \widehat{\delta}(q_1, x) = q_3$.
  - Assume $x \in \mathcal{L}_3$.
  - $x = w10^{2k+1}$, $y = w10^{2k}$ and $\sigma = 0$
  - Then $y \in \mathcal{L}_2$, and by i.h. $\widehat{\delta}(q_1, y) = q_2$. Therefore, $\widehat{\delta}(q_1, x) = \widehat{\delta}(q_2, 0) = q_3$.
- $\widehat{\delta}(q_1, x) = q_3 \implies x \in \mathcal{L}_3$.
  - Assume $\widehat{\delta}(q_1, x) = q_3$
  - Let $q_y = \widehat{\delta}(q_1, y)$ (hence, $\widehat{\delta}(q_y, \sigma) = \widehat{\delta}(q_1, x) = q_3$)
  - Hence, $q_y = q_2$ and $\sigma = 0$ (?)
  - By i.h. $y = w10^{2k}$
  - Therefore, $x = y\sigma = w10^{2k}0 \in \mathcal{L}_3$

Part II

Regular Operations

Additional examples of regular languages

Let $\Sigma = \{0, 1\}$.

- Odd number of 1's: $\{w \in \{0,1\}^* \colon \#_1(w) \pmod 2 = 1\}$.
- Sequence of 0 followed by sequence of 1, i.e., $\{0^m 1^n \colon m, n \geq 0\}$.
- Any finite language.

All the above languages are regular Is there a simple proof?

The regular operations

Let $\mathcal{A}$ and $\mathcal{B}$ be languages.

The union operation:

$$\mathcal{A} \cup \mathcal{B} = \{x : x \in \mathcal{A} \text{ or } x \in \mathcal{B}\}$$

The concatenation operation:

$$\mathcal{A} \| \mathcal{B} = \{xy : x \in \mathcal{A} \text{ and } y \in \mathcal{B}\}$$

The star operation:

$$\mathcal{A}^* = \{x_1 x_2 \ldots x_k : k \geq 0 \text{ and each } x_i \in \mathcal{A}\}$$

The regular operations – Examples

Let $\mathcal{A} = \{\text{good}, \text{bad}\}$ and $\mathcal{B} = \{\text{boy}, \text{girl}\}$.

Union

$$\mathcal{A} \cup \mathcal{B} = \{\text{good}, \text{bad}, \text{boy}, \text{girl}\}$$

Concatenation

$$\mathcal{A} \| \mathcal{B} = \{\text{goodboy}, \text{goodgirl}, \text{badboy}, \text{badgirl}\}$$

Star

$$\mathcal{A}^* = \{\varepsilon, \text{good}, \text{bad}, \text{goodgood}, \text{goodbad}, \text{badbad}, \text{badgood}, \ldots\}$$

Closure under union

Theorem 12

If $\mathcal{A}_1$ and $\mathcal{A}_2$ are regular languages, then so is $\mathcal{A}_1 \cup \mathcal{A}_2$.

Approach to Proof:

- Some DFA M_1 accepts $\mathcal{A}_1$
- Some DFA M_2 accepts $\mathcal{A}_2$
- Construct DFA M that accepts $\mathcal{A}_1 \cup \mathcal{A}_2$.

Attempted Proof Idea:

- first simulate M_1, and
- if M_1 doesn't accept, then simulate M_2.

What's wrong with this?
Fix: Simulate both machines simultaneously.

Closure Under Union: Correct Proof

Suppose

- $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ accepts $\mathcal{L}_1$,
- $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ accepts $\mathcal{L}_2$.

Define M as follows (M will accept $\mathcal{L}_1 \cup \mathcal{L}_2$):

- $Q = Q_1 \times Q_2$.
- Σ is the same.
- For each $(r_1, r_2) \in Q$ and $a \in \Sigma$,
  $\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$
- $q_0 = (q_1, q_2)$
- $F = \{(r_1, r_2) \colon r_1 \in F_1 \text{ or } r_2 \in F_2\}$.
- Formal proof (next slide) ♣

(hey, why not choose $F = F_1 \times F_2$?)

Correctness of the construction

Claim 13

$\mathcal{L}(M) = \mathcal{L}(M_1) \cup \mathcal{L}(M_2)$.

Proof:

Induction hypothesis: $\widehat{\delta}((q_1,q_2),x) = (\widehat{\delta}_1(q_1,x),\widehat{\delta}_2(q_2,x))$.

Induction Basis: $x = \epsilon$, $|x| = 0$. Trivial

Induction Step: $x = y\sigma$, $|x| = i+1$, $|y| = i$ and $\sigma \in \Sigma$.

Follows from the definition of δ.

Completing the proof:

$x \in \mathcal{L}(M_1)$ implies $\widehat{\delta}_1(q_1,x) = r_1 \in F_1$.
Hence, $\widehat{\delta}((q_1,q_2),x) = (r_1,r_2) \in F$.
(similar if $x \in \mathcal{L}(M_2)$.)

$x \in \mathcal{L}(M)$ implies $\widehat{\delta}((q_1,q_2),x) = (r_1,r_2) \in F$.
Hence, $\widehat{\delta}_i(q_i,x) = r_i$, $i \in \{1,2\}$.
Since $(r_1,r_2) \in F$ either $r_1 \in F_1$ or $r_2 \in F_2$.

What about concatenation?

Theorem 14

If $\mathcal{L}_1$, $\mathcal{L}_2$ are regular languages, then so is $\mathcal{L}_1 \| \mathcal{L}_2$.

Example: $\mathcal{L}_1 = \{\text{good}, \text{bad}\}$ and $\mathcal{L}_2 = \{\text{boy}, \text{girl}\}$.

$$\mathcal{L}_1 \| \mathcal{L}_2 = \{\text{goodboy}, \text{goodgirl}, \text{badboy}, \text{badgirl}\}$$

This is much harder to prove.

Idea: Simulate M_1 for a while, then switch to M_2.

Problem: But when do you switch?

This leads us into non-determinism, wait for next class...