Comp 330 - Theory of Computation

Lecture 3 - Fall 2023

Claude Crépeau, Cesare Spinoso-Di Piano

September 7th 2023

McGill University - School of Computer Science

Plan for today

1. Intro to Automata Theory
2. Deterministic Finite Automata

Intro to Automata Theory

What's an automaton?

Automaton: Theoretical computing machine

- From Greek, *automatos* which means "acting of itself".

A bit of history

- 1936/1937: Alan Turing introduces the "a-machines" ("a" for automatic) and shows that "Entscheindungsproblem" is undecidable.
- 1942: Warren McCulloh (MD) and Walter Pitts (Logician) introduce the *neural network* to model the brain in their paper "A Logical Calculus of the Ideas Immanent in Nervous Activity".
- 1951: John von Neumann publishes "The General and Logical Theory of Automata".
- 1967: Marvin Minsky publishes the book "Finite and Infinite Machines".

Finite memory ← $\boxed{\text{finite}}$ Deterministic automata

Deterministic Finite Automata

Deterministic Finite Automaton / Automata

$\Sigma \to$ alphabet

Def (Informal) A DFA, M, is a computational model which, given some input $x \in \Sigma^*$, either accepts or rejects x. DFA are known as language acceptors / recognizers.

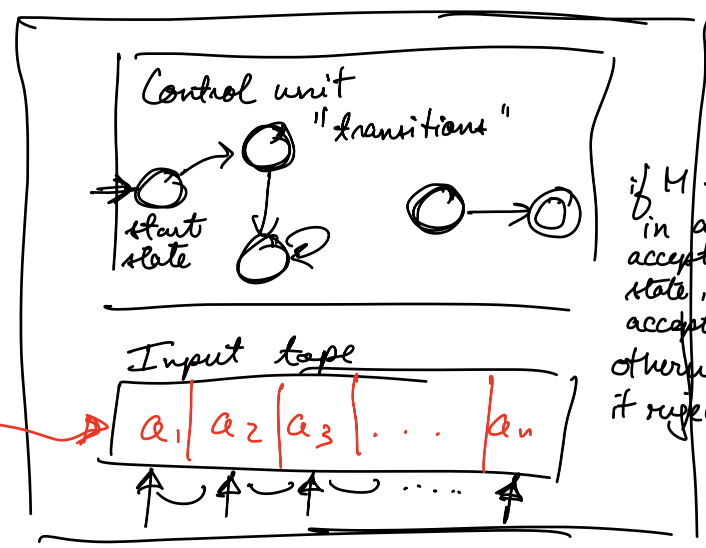
Illustration

DFA M

Input:

$x = \underbrace{a_1 a_2 \ldots a_n}$

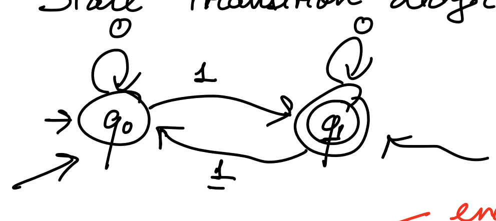
$a_i \in \Sigma$
$1 \le i \le n$



if M ends in an accept state, it accepts x otherwise it rejects x

Ex $\Sigma = \{0,1\}$. Following DFA M

State transition diagram



$x = 11$
$x = 1001$
encodes odd # 1

encodes even # of 1s

end of string marker.

$x = 0 \; 1 \; 1 \; \cancel{\vdash}$ $\to$ reject x because q_0 is not an accept state

$q_0 \quad q_0 \quad q_1 \quad q_0$

end of string marker

$x = 1 \quad 0 \quad 0 \quad \cancel{1} \rightarrow$ accept x because q_1 is an accept state

$\;\;\; q_0 \rightarrow q_1 \rightarrow q_1 \rightarrow q_1$

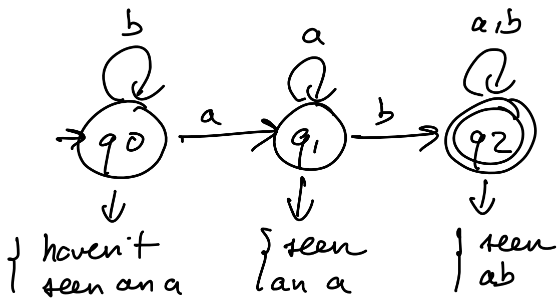
$L(M) = \{x \in \{0,1\}^* : x \text{ has an odd \# of 1s}\}$

$L(M)$ ← language accepted by M

Ex Design a DFA M s.t. $L(M) = \{ w \in \{a,b\}^* :$ w contains substring $ab \}$

$x = aaa$ ✗

$x = a\underbrace{ab}$ ✓

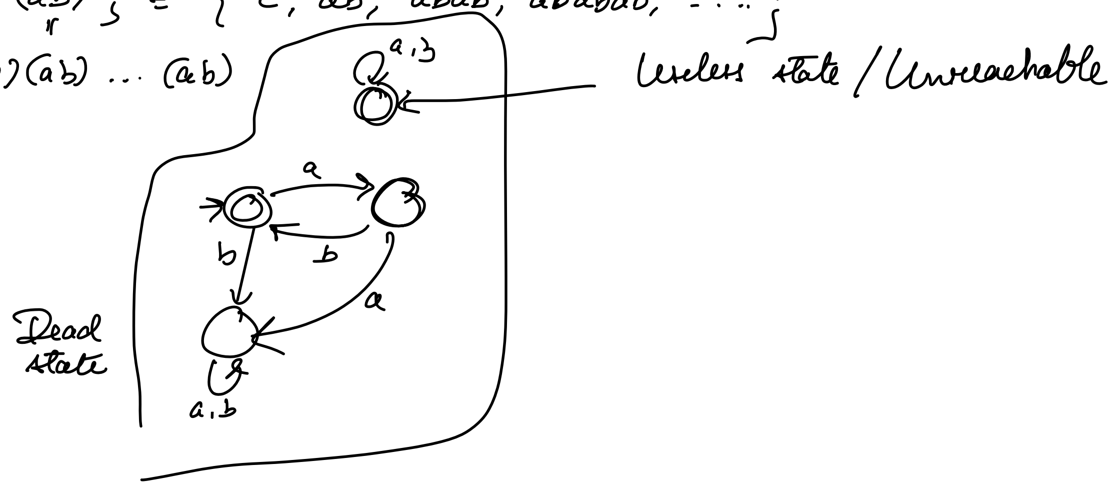


For each state, There are exactly $|\Sigma|$ transitions, one for each $\sigma \in \Sigma$

Ex Design a DFA M s.t. $L(M) = \{ (ab)^n : n \geq 0 \}$

$\{ \underbrace{(ab)^n}_{} \} = \{ \varepsilon, ab, abab, ababab, \ldots \}$

$(ab)(ab) \ldots (ab)$



Formal definition of DFA

Def A DFA M is a 5-tuple

$(Q, \Sigma, \delta, s_0, F)$

Q : set of states

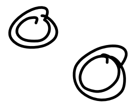
Σ : input alphabet

δ : transition function $\quad \delta : Q \times \Sigma \to Q$

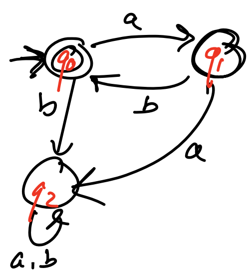
$$q \xrightarrow{a} p \iff \delta(q,a) = p$$

s_0 : start state, $s_0 \in Q$

F : set of accept states, $F \subseteq Q$



M



$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{a, b\}$

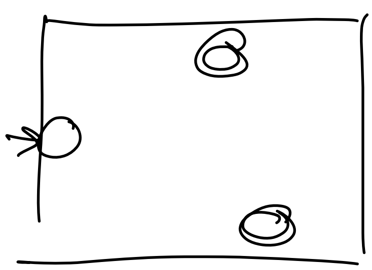
$\delta(q_0, a) = q_1$

$s_0 := q_0$

$F = \{q_0\}$

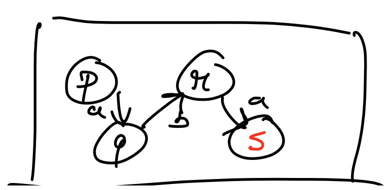
In general, given DFA M

x



Formally : Extended transition function δ^*

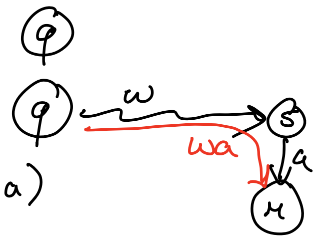
$$\delta^* : Q \times \Sigma^* \to Q$$



$\delta^*(p, aba) = s$

Definition δ^*

1. $\delta^*(q, \varepsilon) = q$



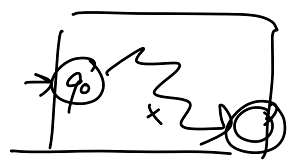
2. $w \in \Sigma^*,\ a \in \Sigma$

$$\delta^*(q, wa) = \delta(\underbrace{\delta^*(q, w)}, a)$$

Given a string $x \in \Sigma^*$ & a DFA M:

We say M accepts x $\iff$

$$\delta^*(s_0, x) \in F$$

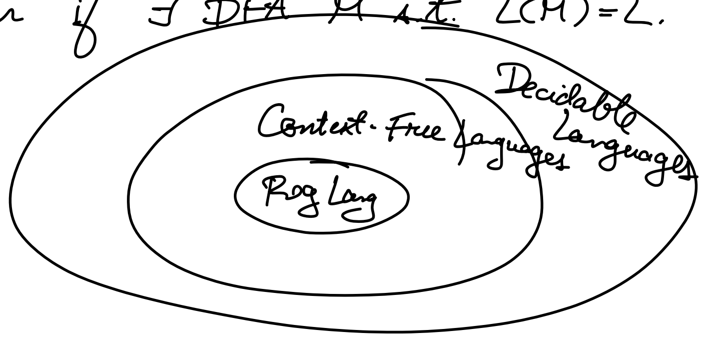


Def Given a DFA $M = (Q, \Sigma, \delta, s_0, F)$

$\rightarrow L(M) = \{ x \in \Sigma^* : \delta^*(s_0, x) \in F \}$

Language accepted by M

Def (Regular language) Given Σ, $L \subseteq \Sigma^*$, L is regular if $\exists$ DFA M s.t. $L(M) = L$.



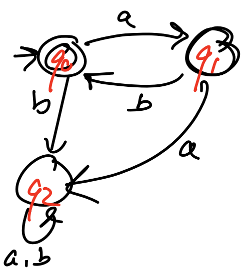
Proof Technique How do we prove that a language L is regular?

1. Construct a DFA M s.t. $L(M) = L$

2. Prove correctness of M : $\forall x \in \Sigma^*$, $x \in L(M) \Leftrightarrow x \in L$

<u>Ex</u> Prove that language $L = \{(ab)^n : n \geq 0\}$ is regular.

1. M



2. Prove that $L(M) = L$

   1. Make a claim about δ^* and prove that claim

   2. Show using ① that $\forall x \in \Sigma^*$, $x \in L(M) \Leftrightarrow x \in L$

① What is each state doing?

$P(M, w)$: " if $\delta^*(q_0, w) = q_0$ then $w = (ab)^n$, $n \geq 0$
AND if $\delta^*(q_0, w) = q_1$ then $w = (ab)^n a$, $n \geq 0$
AND if $\delta^*(q_0, w) = q_2$ then $\forall n \geq 0$ $w \neq (ab)^n$ and $w \neq (ab)^n a$ "

<u>Proof</u> By induction on $|w|$

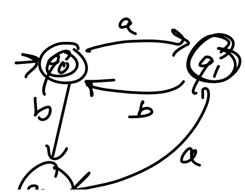
$M = (Q, \Sigma, \delta, S_0, F)$ $S_0 := q_0$ → M is the machine we built

② W.T.S. Given DFA M, $\forall x \in \Sigma^*$, $x \in L(M) \Leftrightarrow x \in L$

⇒ Directly

⇐ Contrapositive

⇒ Pick some $x \in \Sigma^*$, $x \in L(M)$.



$\Rightarrow \quad \delta^*(q_0, x) \in F$

$\Rightarrow \quad \delta^*(q_0, x) \in \{q_0\}$

$\Rightarrow \quad \delta^*(q_0, x) = q_0$

$\Rightarrow \quad x = (ab)^n, \; n \geq 0$ by $P(M,w)$

$\Rightarrow \quad x \in L$

$\boxed{L = \{(ab)^n : n \geq 0\}}$

$\Leftarrow \quad (x \in L \Rightarrow x \in L(M), \; \underline{x \notin L(M) \Rightarrow x \notin L})$

$x \in \Sigma^*, \; x \notin L(M)$

$\Rightarrow \quad \delta^*(q_0, x) \notin F$

$\Rightarrow \quad \delta^*(q_0, x) \in \{q_1, q_2\}$

$\Rightarrow \quad \delta^*(q_0, x) = q_1$ or $\delta^*(s_0, x) = q_2$

$\Rightarrow \quad x = (ab)^n a$ or $[x \neq (ab)^n$ $\boxed{\text{and}}$ $x \neq (ab)^n a]$ by $P(M,w)$

$\Rightarrow \quad x \notin L$

↳ Shows $x \notin L(M) \Rightarrow x \notin L$

$\therefore \quad \forall x \in \Sigma^*. \; x \in L(M) \Leftrightarrow x \in L.$ $\boxed{\text{A1Q4}}$

Proof of $P(M,w)$ (We skipped this during lecture)

I will post this in a separate handout under Lecture 3.