11 Jan 2025 - Theory of Computation - Week 02

Doubts

* Computability theory and Gödel's incompleteness theorem.

13 Jan 2025

Regular Expressions → regex

Basic (atomic) expressions

(i) Σ finite alphabet

→ every symbol $a \in \Sigma$ is a regular expression.

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

(ii) ε, empty string (iii) ϕ, empty language

algebraic expressions

$xy + 2y^2z$

operations on variables

regular expressions

operations on regular sets

Regular expression		language
r	$\longleftrightarrow$	$L(r)$ = set of all strings that 'matches' this expression r
$a \in \Sigma$	$\longleftrightarrow$	$L(a) = \{a\}$
ε	$\longleftrightarrow$	$L(\varepsilon) = \{\varepsilon\}$
ϕ	$\longleftrightarrow$	$L(\phi) = \phi$

$\phi \rightarrow$ empty set

$\phi \rightarrow$ regular expression

Regular Operations : (i) Let r and s be two regular expressions. Then, $r + s$ is a regular expression that denotes $L(r) \cup L(s)$

Union

$$\underbrace{0 + 1}_{\text{regular expression}} \equiv \underbrace{\{0, 1\}}_{L(0) \cup L(1)}$$

(ii) Concatenation $rs = r \cdot s \equiv L(r) \cdot L(s)$

$A, B \subseteq \Sigma^*$

$A \cdot B = \{xy \mid x \in A \text{ and } y \in B\}$

$A = \{a, b\}$, $B = \{0, 1\}$

$A \cdot B = \{a0, a1, b0, b1\}$

$\phi \cdot A = \phi = A \cdot \phi$

(iii) Kleene closure (star) *

this is what makes it fundamental

$$A^* \equiv \bigcup_{n=0}^{\infty} A^n$$

$$= \{ x = x_1 x_2 \dots x_k \mid \exists\ k \geq 0 \text{ such that each } x_i \in A \}$$

$$\varepsilon \in A^*$$

$(0+1)^*$ $\equiv$ set of all finite binary strings

$(0+1)^* \cdot 1$ $\equiv$ all primary strings that end with 1.

$(0+1)^*\ 1\ 1\ (0+1)^* \equiv$ all binary strings with consecutive ones.

same

Automaton — M — $L(M)$

regular expression — α — $L(\alpha)$

language represented by α

e.g.:

$(0+1)^*\ 0^5\ 1^{10}\ 1^*\ (0+1)$

$*$

$.$

$+$

↑ precedence

If A is a regular language, then does there exist a regular expression α such that $A = L(\alpha)$

- Any computation → is identifying a language.

- these three are universal operations : offer operations like intersection, complementation can be implemented using these.

- there's no 'non-regular' expression. 'Regular expression' ← single entity.

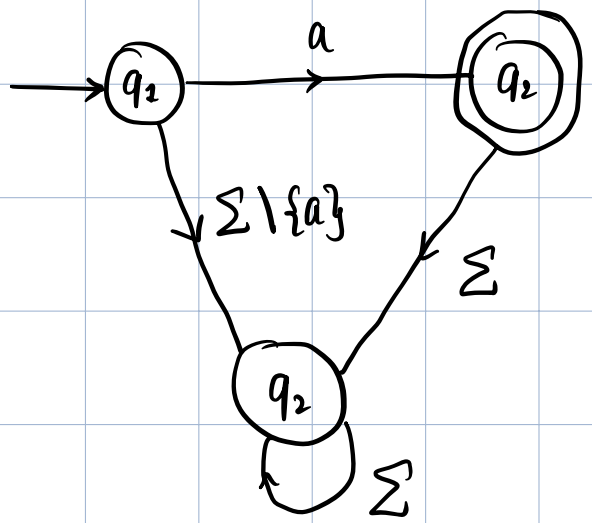
* Given a regular expression α, show that $L(\alpha)$ is regular.

Base cases

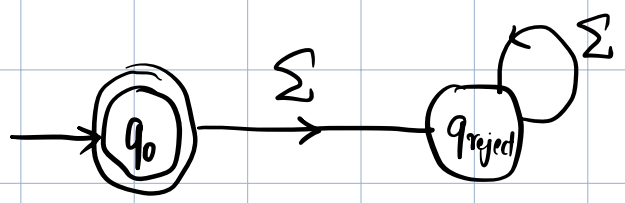
Case 1: $a \in \Sigma$

M_a

$L(M_a) = \{a\}$

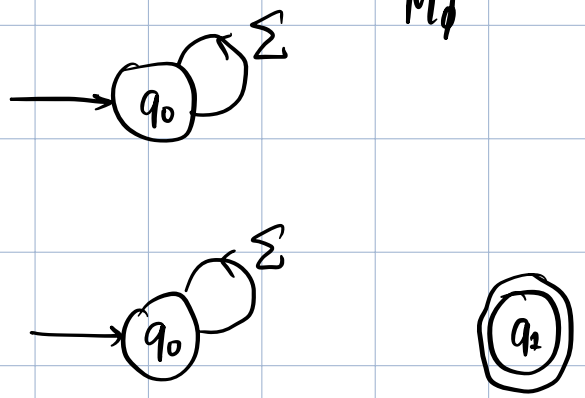


Case 2:



Case 3: $L(M_\phi) = \phi$

M_ϕ



- Now, we need to prove that regular expressions are closed under the 3 fundamental operations.

$L = \{0^k 1^k \mid k \geq 0\}$
↓
not a regular language

15 Jan 2025

Regular languages

set of languages accepted by finite automata (A)

$\stackrel{?}{=}$

Regular expressions

set of languages recognized by regular expressions (B)

Claim 1 : $A \subseteq B$

Claim 2 : $B \subseteq A$

last class

Proof of claim 2

α is reg exp $\Sigma = \{0, 1\}$

$\alpha = \underbrace{(0+1)^* \cdot (1)^*}_{\text{inductively build automata}} + 1$

combine

Claim 3: If L_1 and L_2 are regular (languages), then $L_1 \cup L_2$ is regular.

$L \subseteq \Sigma^*$
↳ always defined in relation to alphabet

Proof :- Let M_i be a finite automata recognizing L_i $i \in \{1, 2\}$

$$M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$$

$$M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$$

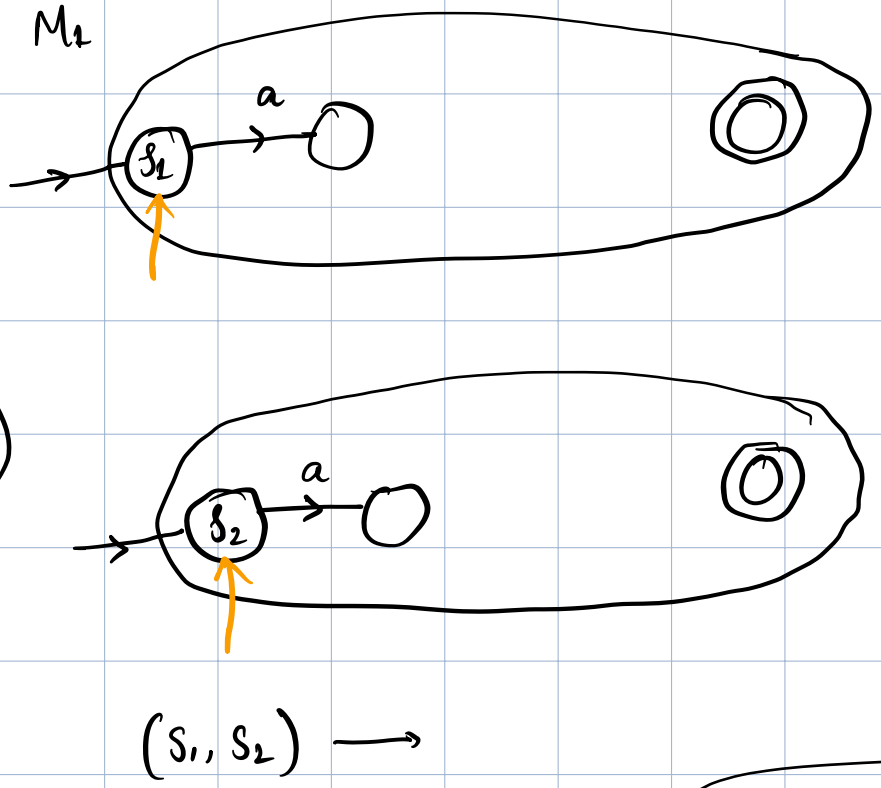
Create M_3 that runs M_1 and M_2 parallelly

$$M_3 = (Q_3, \Sigma, \delta_3, (s_1, s_2), F_3)$$

$$Q_3 = \overbrace{Q_1 \times Q_2}^{\text{finite}}$$

$$\delta_3((q_1, q_2), a)$$

$$= (\delta_1(q_1, a), \delta_2(q_2, a))$$



$$F_3 = F_1 \times Q_2 \cup Q_1 \times F_2$$

If alphabets are different, add base cases to reject exceptional symbols for each machine

Correctness of M_3 $\quad L(M_3) = L_1 \cup L_2$ ⟵ prove by induction on the length of M_3

Take $x \in L(M_3)$, then

$x \in L(M_3) \Rightarrow x \in L_1 \cup L_2$

$x \in L_1$ or $L_2 \Rightarrow x \in L(M_3)$

→ Use $\hat{\delta}$

$F_3' = F_1 \times F_2$ ⟵ intersection of regular languages

Complement of L $\quad L \subseteq \Sigma^*$: $\bar{L} / \sim L = \Sigma^* \setminus L$

regular languages are closed under complementation

union

intersection

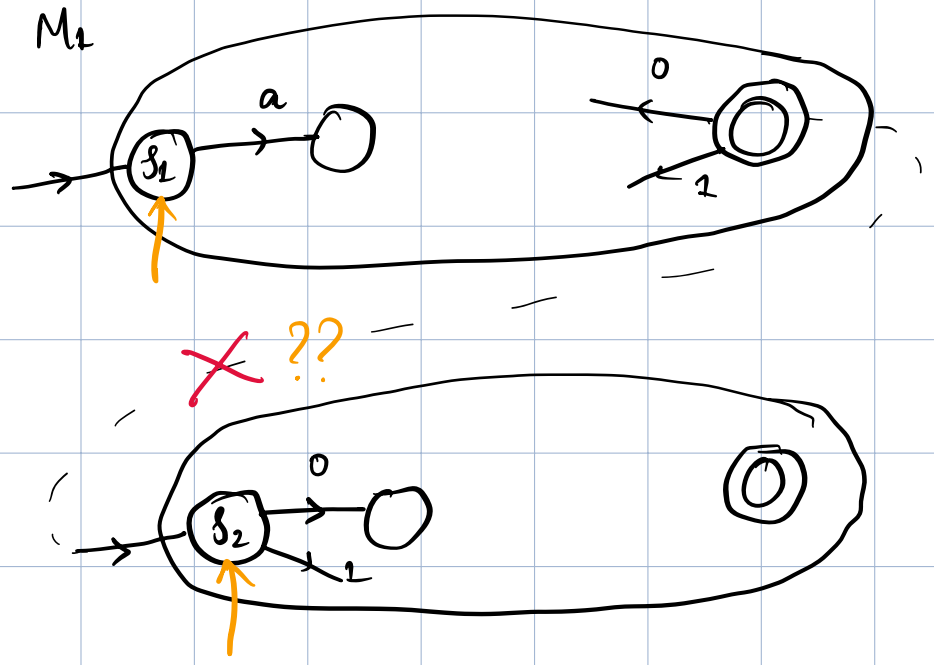
Claim 4: If L_1 and L_2 are regular then $L_1 \cdot L_2$ is regular

$x = y \cdot z$

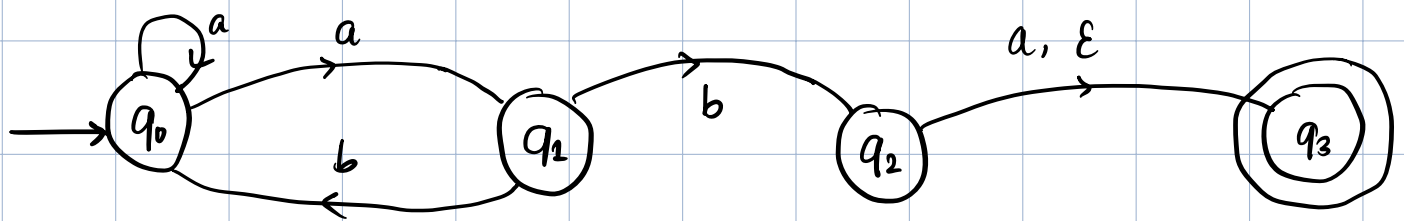
$x_1 \cdots x_n$

$y \in L_1$

$z \in L_2$



Non-determinism Non-deterministic finite automata

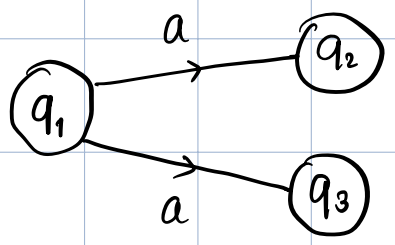


M_2

$q_0 \xrightarrow{b} \phi$

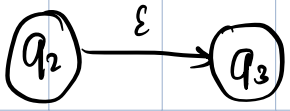
(1) move to a subset of states

At each step, you have many options to take, including ϕ



(2) ε transitions

not required for non-determinism but useful

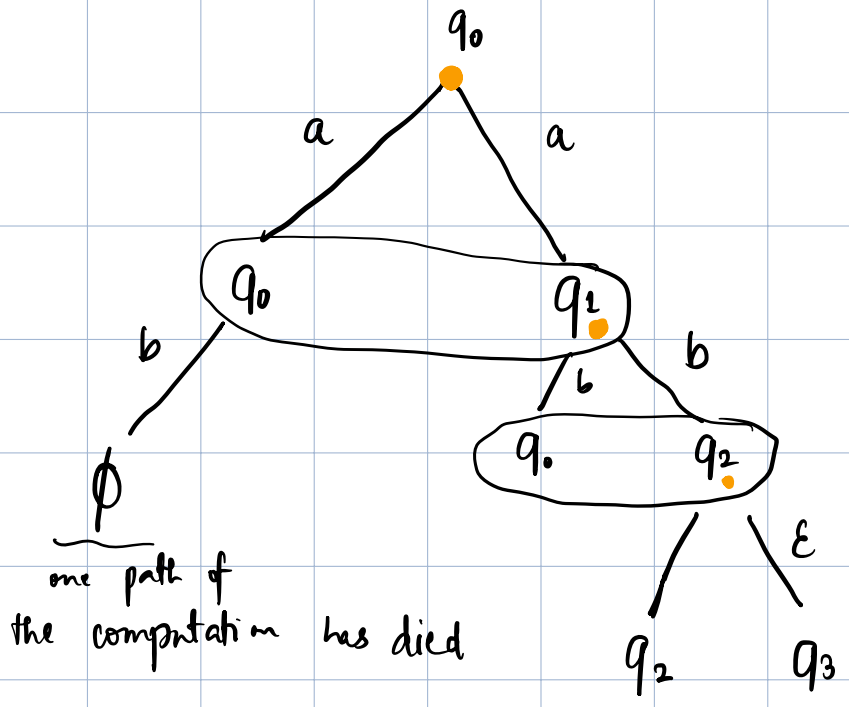


moving to a state without reading an input

non-deterministic does not mean random

16 Jan 2025

Input : ab



one path of the computation has died

• → non-deterministic choices

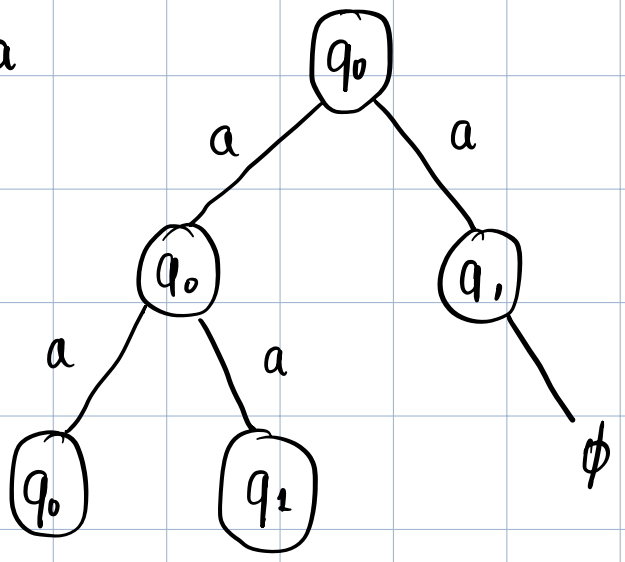
M_1 accepts ab

$\varepsilon a = a\varepsilon = a$

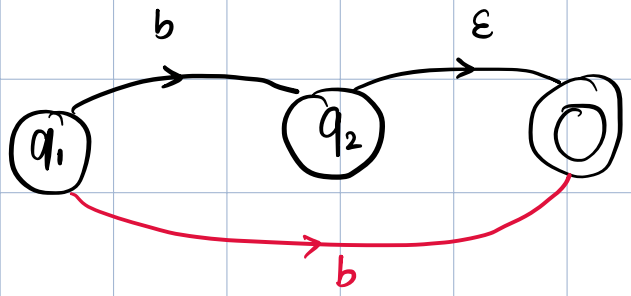
$a\,\varepsilon\,b = ab = a\,\varepsilon^2\,b\,\varepsilon \ldots$

$\varepsilon\,a\,\varepsilon\,b\,\varepsilon$

aa



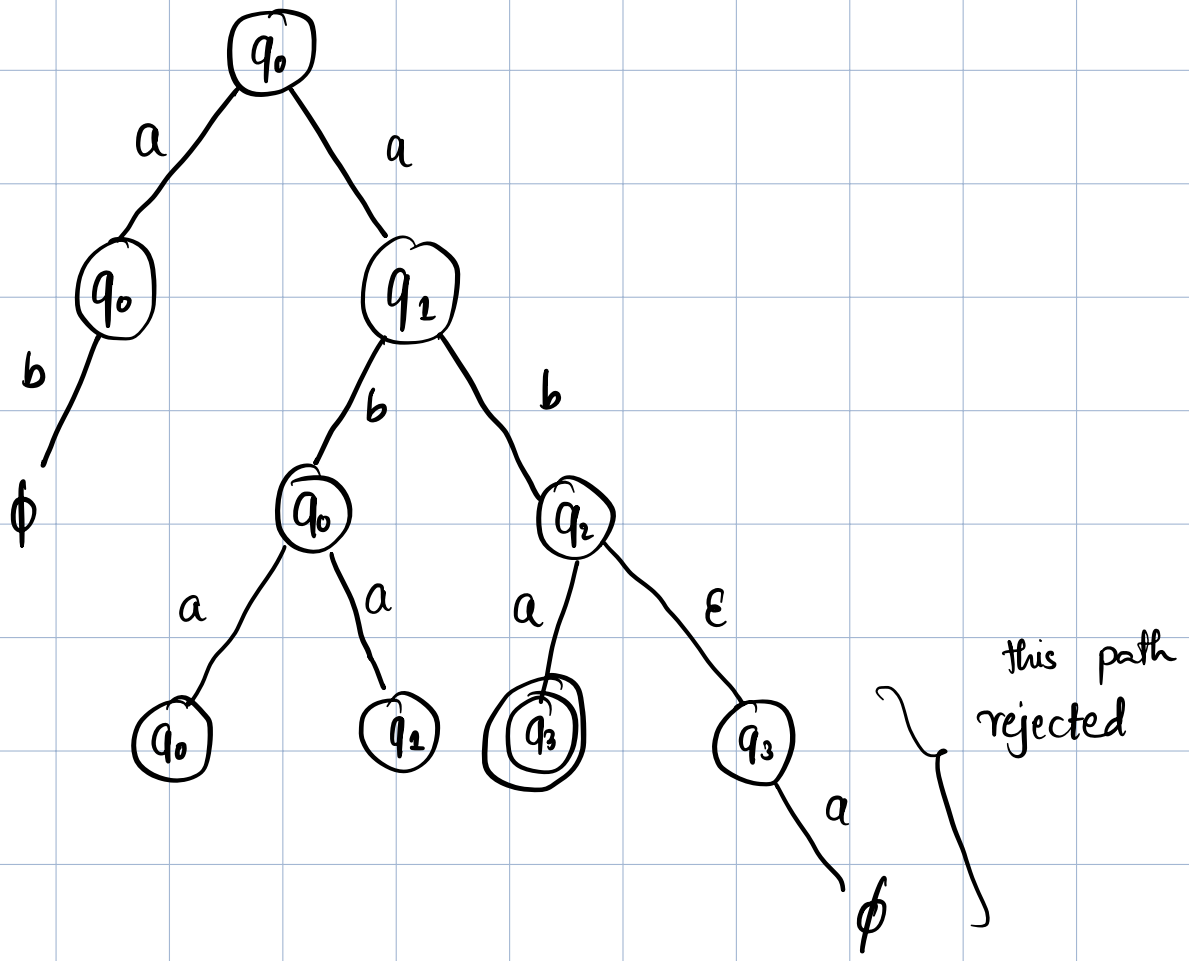
aa → rejected



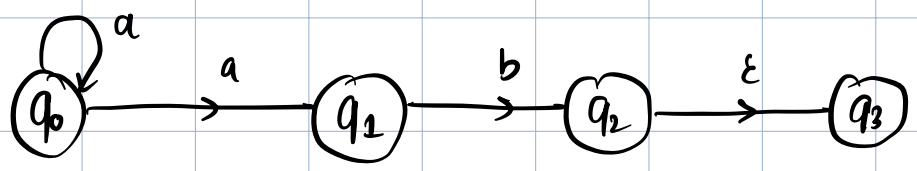
Why not ?

→ ε transitions are helpful in constructing automata, not necessary for non-DFA

a ba


q_0
a
a
q_0
q_1
b
b
b
ϕ
q_0
q_2
a
a
a
ε
q_0
q_1
q_3
q_3
a
ϕ
this path
rejected

aab



abb

all computations die

$$(\ Q\ ,\ \Sigma\ ,\ \delta\ ,\ q_0\ ,\ F\ )$$

Q: states, finite

Σ: alphabet

δ:

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \longrightarrow \mathcal{P}(Q)$$

power set: set of all states

all subsets of q

- all DFAs are (by def^n) non-DFAs

$\mathcal{P}(Q)$ → replace singleton set

DFA $\subseteq$ NFA

$\downarrow$

languages accepted by DFA can be accepted NFA

NFA seems to have more power than DFA, but it does not

later: NFA (equivalent) DFA

Theorem: Let A be a language recognized by a NFA M.

Then A is regular.

Proof to-do