Finite Automata

CSE 105 Week 2 Discussion

Deadlines and Logistics

- Schedule your tests asap on PrairieTest !
- Do review quizzes on PrairieLearn
- HW2 due Thur 1/30/25 at 5pm (late submission open until 8am next morning)

DFA Definition

DEFINITION 1.5

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

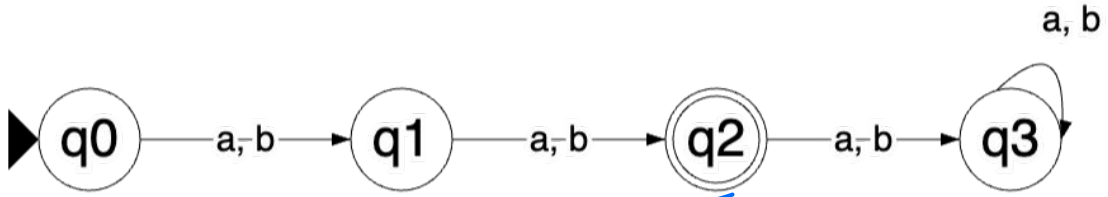
1. Q is a finite set called the ***states***,
2. Σ is a finite set called the ***alphabet***,
3. $\delta : Q \times \Sigma \longrightarrow Q$ is the ***transition function***,[1]
4. $q_0 \in Q$ is the ***start state***, and
5. $F \subseteq Q$ is the ***set of accept states***.[2]

each state of DFA has exactly one outgoing arrow for each symbol in alphabet.

1 and only 1 possible start state

0 or more accept states

DFA State Diagram to Formal Definition



flap.js link

$$\delta((q_i, x)) = \begin{cases} q_{i+1} & \text{if } 0 \le i < 3, x \in \Sigma \\ q_i & \text{if } i = 3, x \in \Sigma \end{cases}$$

Write out the formal definition of the above DFA

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

1. Q is a finite set called the ***states***,
2. Σ is a finite set called the ***alphabet***,
3. $\delta : Q \times \Sigma \longrightarrow Q$ is the ***transition function***,[1]
4. $q_0 \in Q$ is the ***start state***, and
5. $F \subseteq Q$ is the ***set of accept states***.[2]

$Q = \{q_0, q_1, q_2, q_3\}$

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

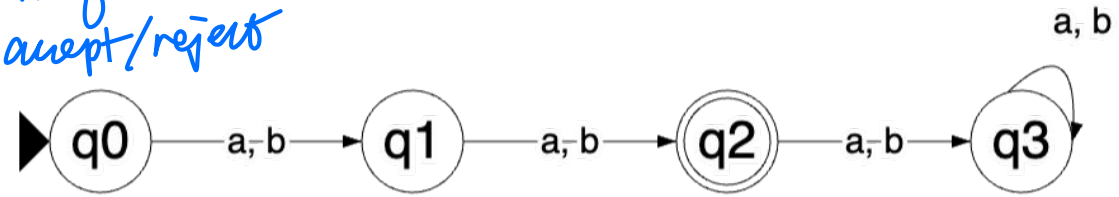
$q_0 = q_0$

$F = \{q_2\}$

δ	a	b
q_0	q_1	q_1
q_1	q_2	q_2
q_2	q_3	q_3
q_3	q_3	q_3

DFA Computation

input : string
output : accept/reject



flap.js link

any string of length 2 over {a,b}

- What are some example strings accepted by this DFA?
- What is the language recognized by this DFA? {aa, ab, ba, bb}
- What is a regular expression that describes this language?

? $L(a \cup b) = \{a, b\}$

$L((a \cup b) \cdot (a \cup b))$

a or b a or b

regex

$(a \cup b) \cdot (a \cup b)$

$(a \cup b)^2$

Σ^2

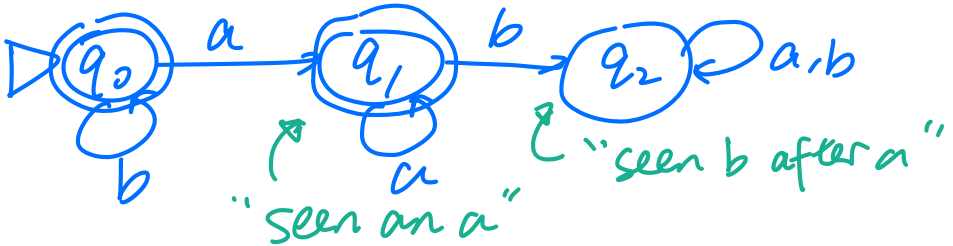
Σ as regex
= shorthand for $(a \cup b)$

DFA Design

deterministic

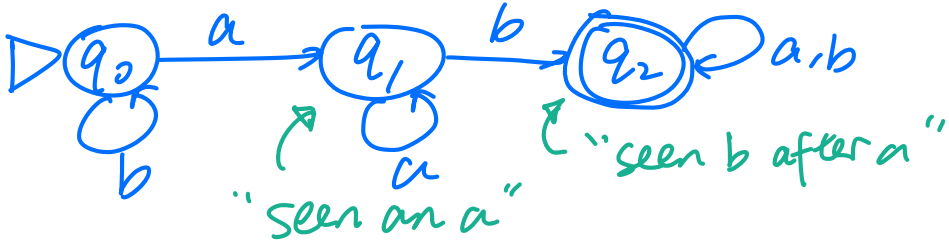
Consider the alphabet $\Sigma = \{a, b\}$, design DFA that recognizes:

$\{w \mid w$ does not contain the substring **ab**$\}$



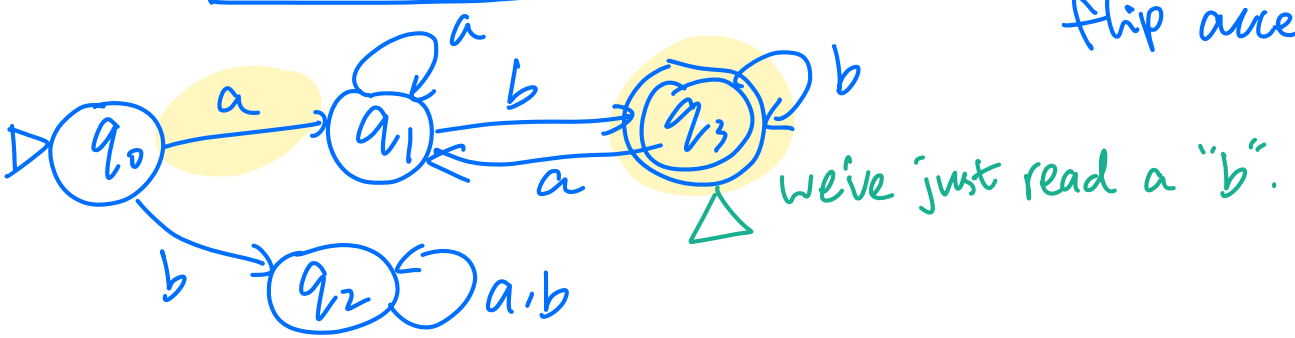
Complement:

$\{w \mid w$ contains substring ab$\}$



flip accept/non-accept states

$\{w \mid w$ begins with **a** and ends with **b**$\}$



NFA Definition

DEFINITION 1.37

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

1. Q is a finite set of states,
2. Σ is a finite alphabet,
3. $\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ is the transition function,
4. $q_0 \in Q$ is the start state, and
5. $F \subseteq Q$ is the set of accept states.

power set. output would be a sets of states

one single state as output

DFA: $\delta : Q \times \Sigma \to Q$

overload meaning of symbol:

ε : as a string represents empty string.

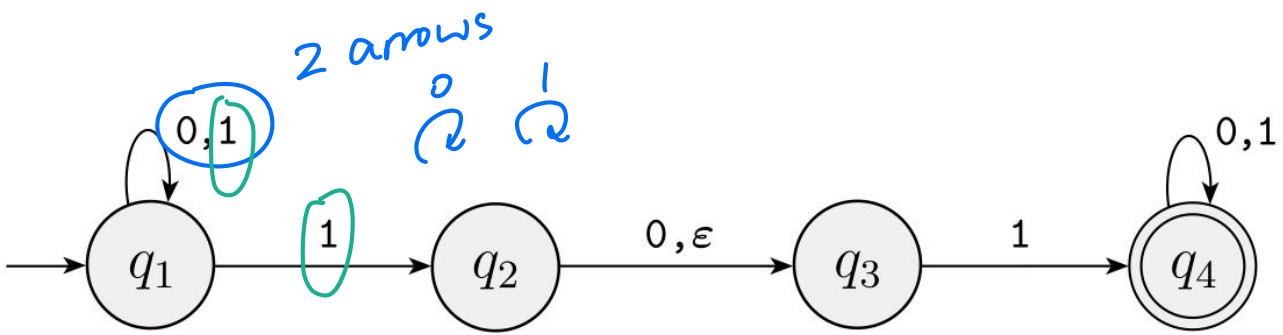
here is an arrow label

$\{\varepsilon\} \cup \Sigma$

arrow labels: symbols from alphabet or ε $\xrightarrow{\varepsilon}$

Spontaneous moves

NFA Transition Function



Consider alphabet $\Sigma = \{0, 1\}$

Complete the transition function of the NFA

	0	1	ε
q_1	$\{q_1\}$	$\{q_1, q_2\}$	$\emptyset$
q_2	$\{q_3\}$	$\emptyset$	$\{q_3\}$
q_3	$\emptyset$	$\{q_4\}$	$\emptyset$
q_4	$\{q_4\}$	$\{q_4\}$	$\emptyset$,

Sipser Example 1.38, Pg 54

NFA vs. DFA

- **Nondeterministic**: when an NFA is in a given state and reads the next input symbol, there is a set of possible next states, i.e. several choices (or no choice) may exist for the next state
- **ε-transitions**: spontaneously moving without reading any input symbols
- **Acceptance condition**: there is a computation of the machine on the string that processes the whole string and ends in an accept state

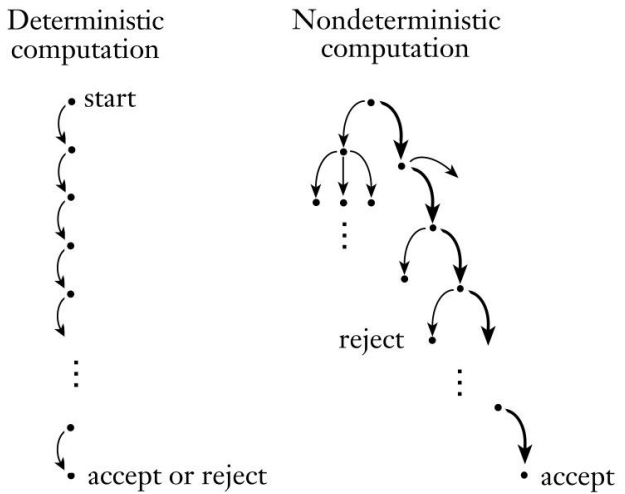
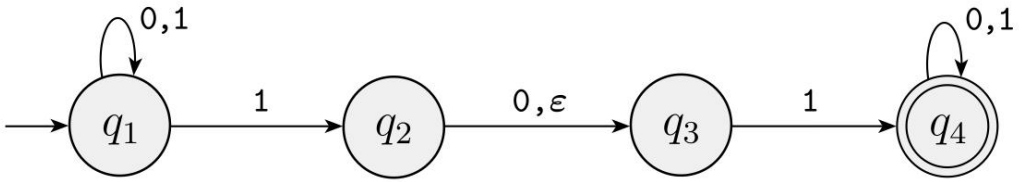


FIGURE **1.28**
Deterministic and nondeterministic computations with an accepting branch

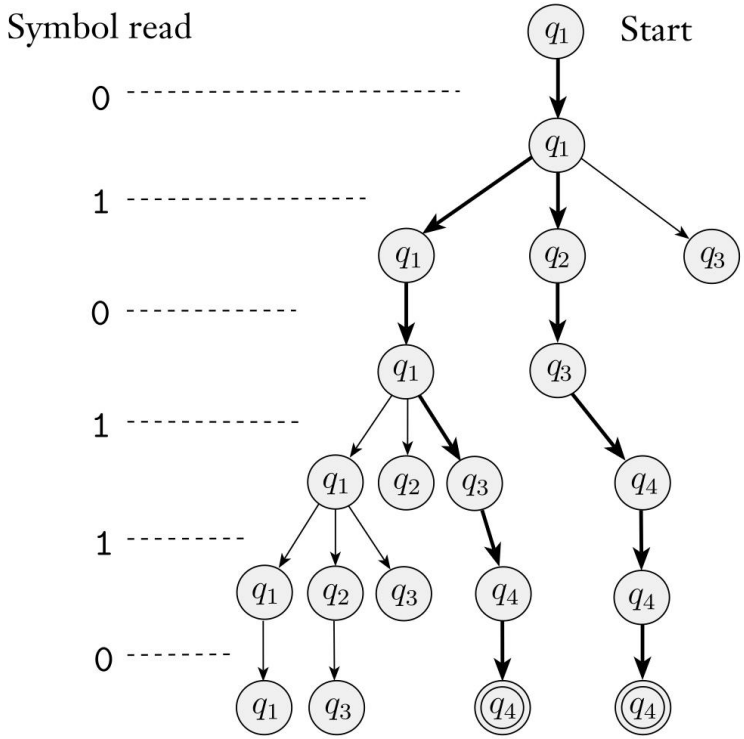
Sipser Figure 1.28, Pg 49

NFA Computation

Computation of the NFA on string 010110



Sipser Figure 1.27, Pg 48



Sipser Figure 1.29, Pg 49

Questions?

Good luck for HW 2 !