Finite Automata

CSE 105 Week 2 Discussion

Deadlines and Logistics

- Schedule your tests asap on PrairieTest !
- Do review quizzes on PrairieLearn
- HW2 due 10/15/24 (Tue) at 5pm (late submission open until 8am next morning)

DFA Definition

DEFINITION 1.5

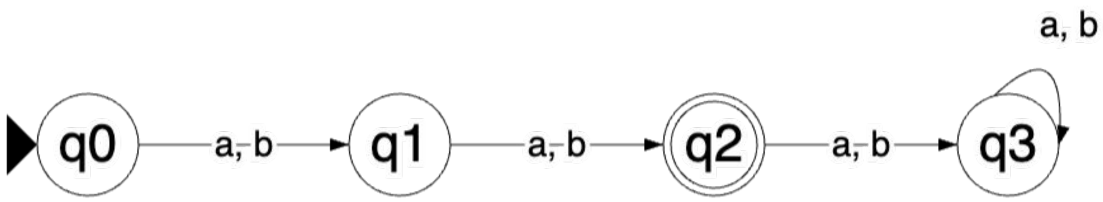
A ***finite automaton*** is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

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]

Sipser Definition 1.5, Pg 35

DFA State Diagram to Formal Definition

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



flap.js link

Write out the formal definition of the above DFA

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

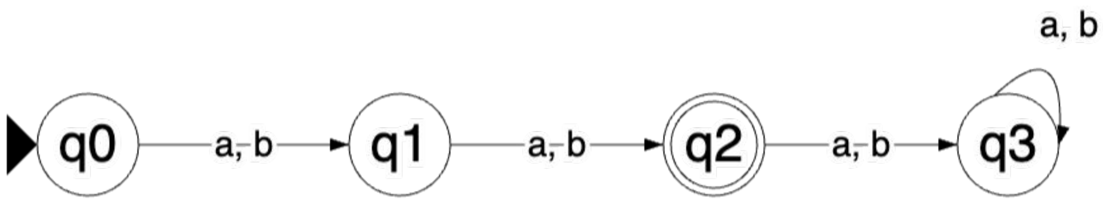
$(\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \{q_2\})$

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]

where

$$\delta((q_i, x)) = \begin{cases} q_{i+1} & \text{where } i < 3 \\ q_i & \text{where } i = 3 \end{cases}$$

DFA Computation



flap.js link

- 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?

~~$((a \cup b)(a \cup b))^*$~~

~~$((a \cup b)(a \cup b))^+$~~

$(a \cup b)^2$

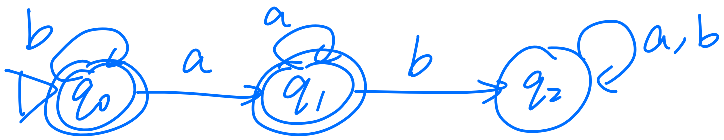
DFA Design

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

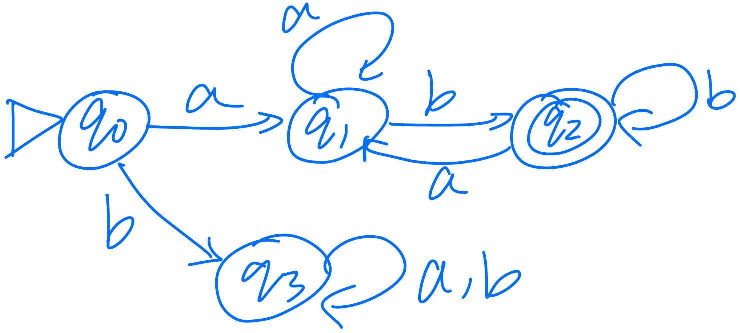
$\{w \mid w$ does not contain the substring **ab**$\}$

Complement:

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



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



NFA Definition

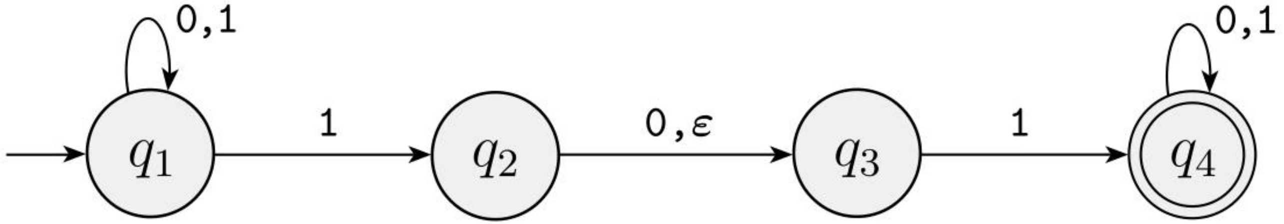
DEFINITION 1.37

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. $\delta \colon Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ is the transition function,
4. $q_0 \in Q$ is the start state, and
5. $F \subseteq Q$ is the set of accept states.

Sipser Definition 1.37, Pg 53

NFA Transition Function



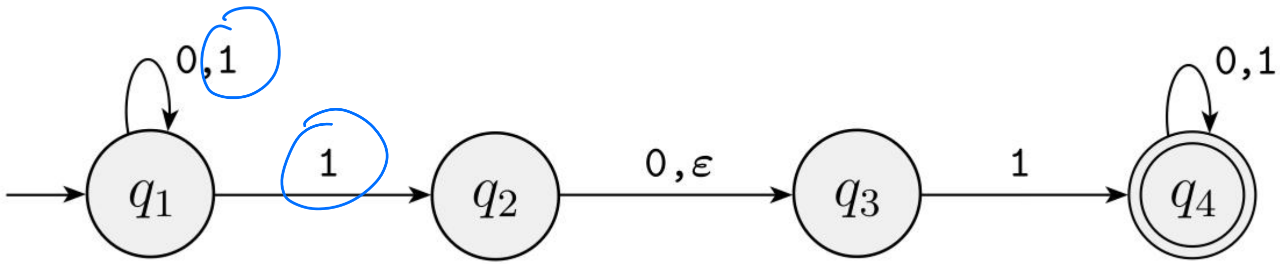
Consider alphabet $\Sigma = \{0, 1\}$

Complete the transition function of the NFA

q_1	
q_2	
q_3	
q_4	

Sipser Example 1.38, Pg 54

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

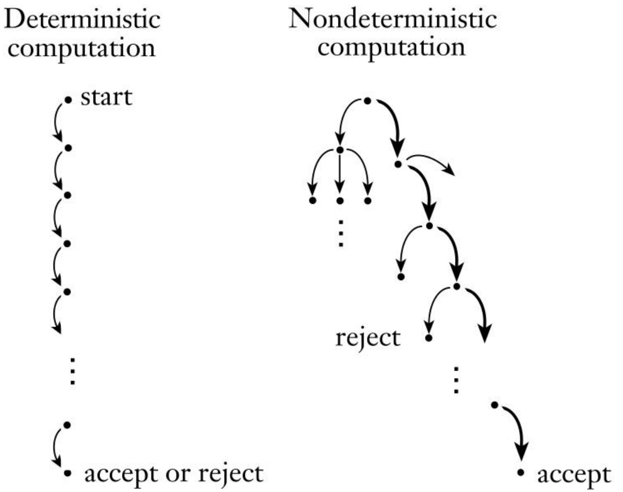
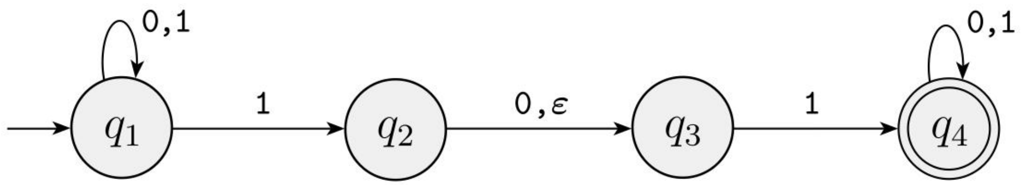


FIGURE 1.28
Deterministic and nondeterministic computations with an accepting branch

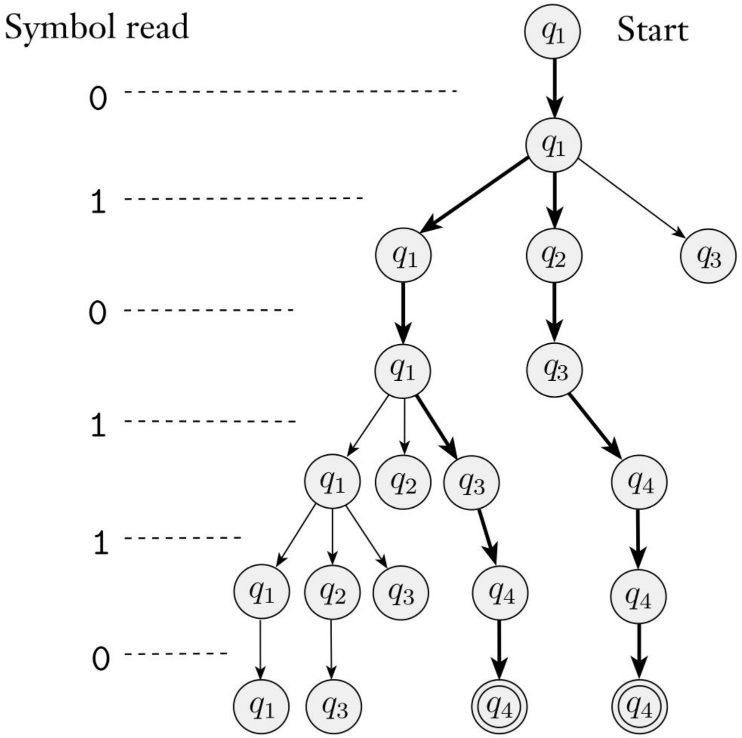
Sipser Figure 1.28, Pg 49

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 !