WI24 CSE 105 DI #2

DFAs and NFAs

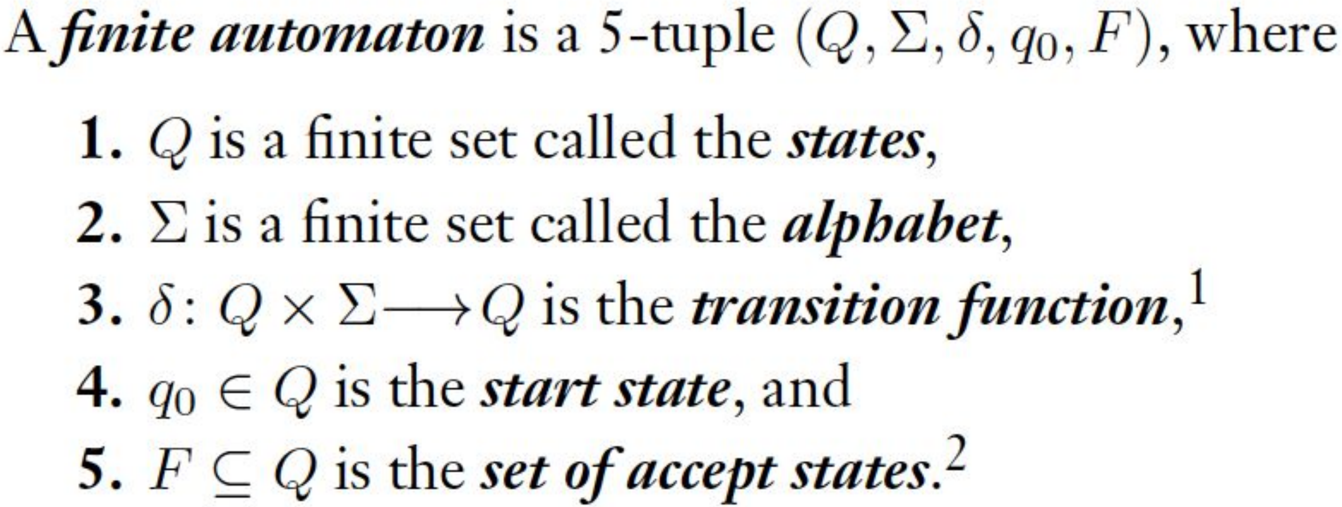
Announcements

- Good job finishing your first assignment! Next one comes out soon and due in ~2 weeks
- Don’t forget to submit your review quizzes due today. (late deadline is Monday 8am after which you won’t be able to submit anymore)

Agenda

- DFA
  - Definition
  - Computation
  - Deriving its language
  - Designing DFA to recognize a certain language
- NFA
  - Definition
  - Compare and contrast with DFA
  - Computation
  - Designing NFA

DFA Definition



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. $\delta \colon Q \times \Sigma \longrightarrow Q$ is the ***transition function***,[1]

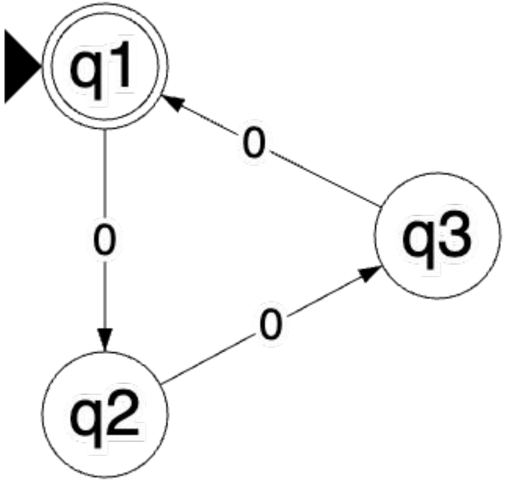
4. $q_0 \in Q$ is the ***start state***, and

5. $F \subseteq Q$ is the ***set of accept states***.[2]

Inferring Formal Specification

flapjs link

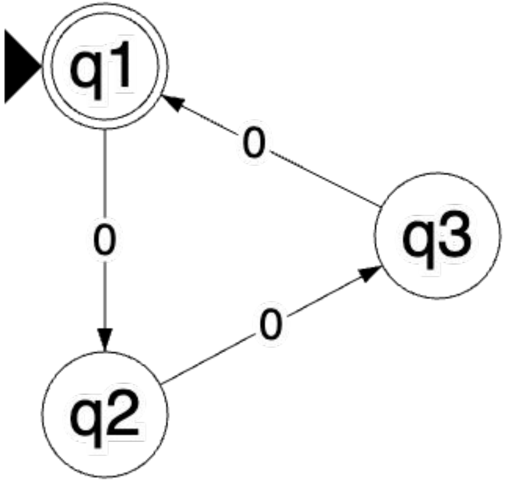
- Q = {...} ?
- Σ = {...} ? How do you know that’s everything
- Transition function
- What is the start state?
- What are the accept states?



Inferring Formal Specification

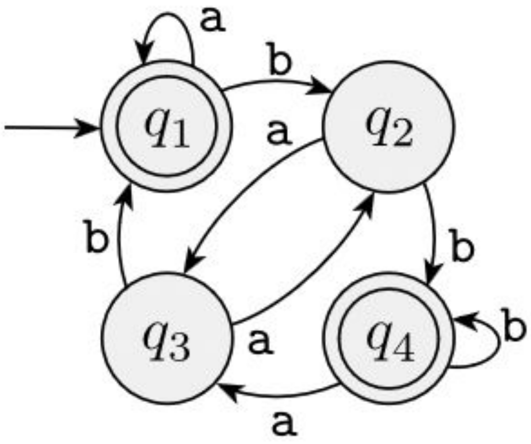
flapjs link

- $Q = \{q1, q2, q3\}$
- $\Sigma = \{0\}$ size same as # transitions from each state
- $\delta(qi,0) = q\{i+1\}$ for $i < 3$ else q1
- q1 is the start
- {q1} is the accept



DFA Practice 1

Write out the transition function for the DFA below:



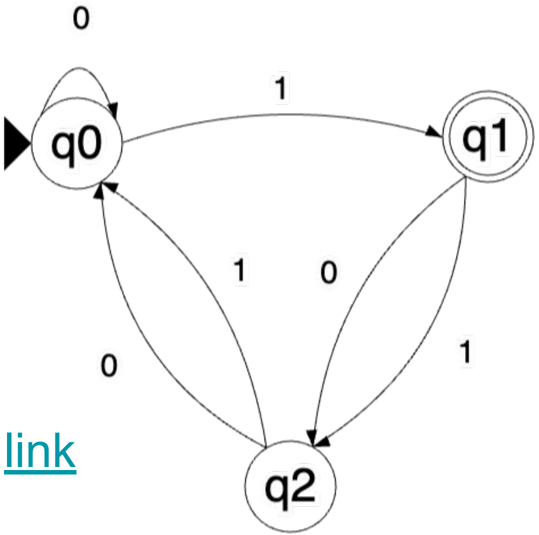
state	a	b
q1	q1	q2
q2	q3	q4
q3	q2	q1
q4	q3	q4

DFA Computation

- DFA traverses its input symbol by symbol, switching states in accordance with a transition function.
- DFA accepts computation if **finishes** reading string and is in an accept state
- Rejects otherwise

What happens when 01101 is fed to the machine?

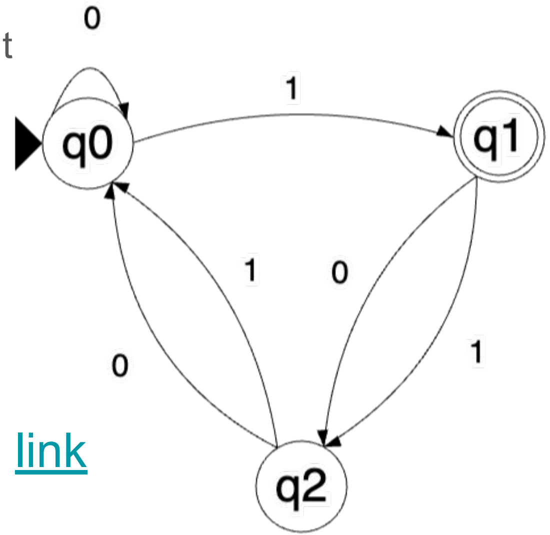
What about 1110



link

Deriving the Language of a DFA (not super important)

- There is a principled way to convert DFA to a regular expression
- But we are just doing intuitively to get a hang of DFAs
- If there is only one accept state, one trick that sometimes work is this:
  - Find a path to the accept state
  - Starting from the accept state, find ways to get back to it



- 0*1(ΣΣ0*1)* is the regex for it
- Hard to come up with a verbal description

link

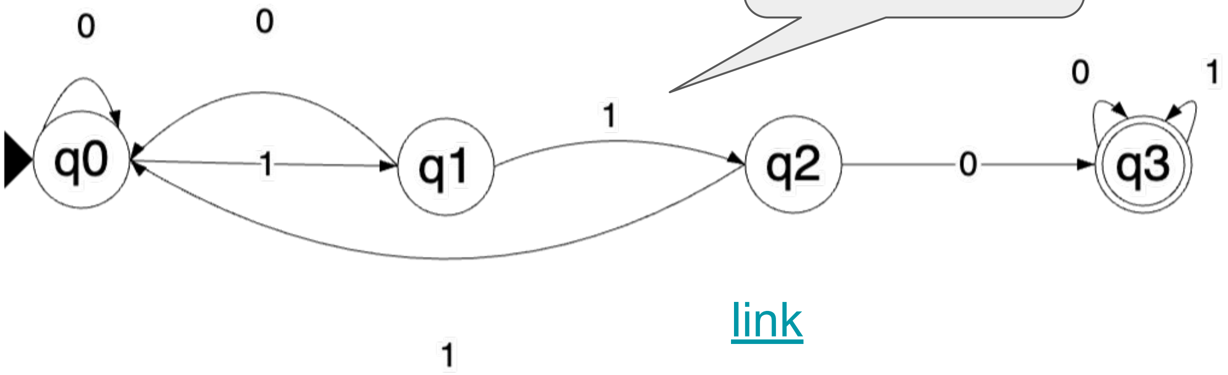
Designing a DFA

Design a DFA for the following language:

L = {w | w contains the substring 110}

- Examples 01101, 110…
- Non-examples 01, 1010…

First attempt



Does this accept 1110?

link

Designing a DFA

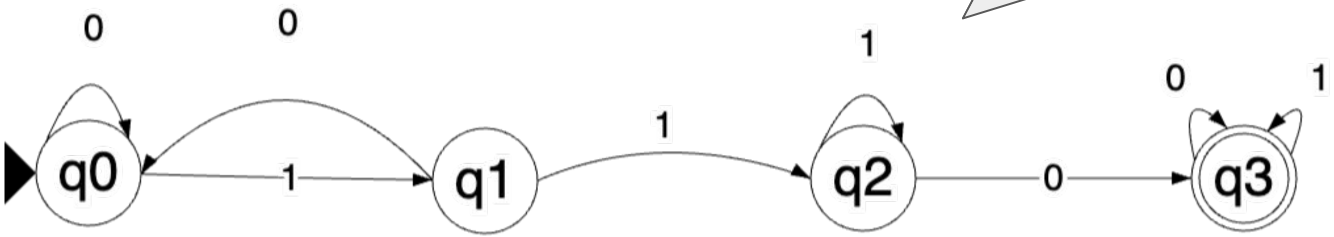
Design a DFA for the following language:

L = {w | w contains the substring 110}

- Examples 01101, 110…
- Non-examples 01, 1010…

Second attempt

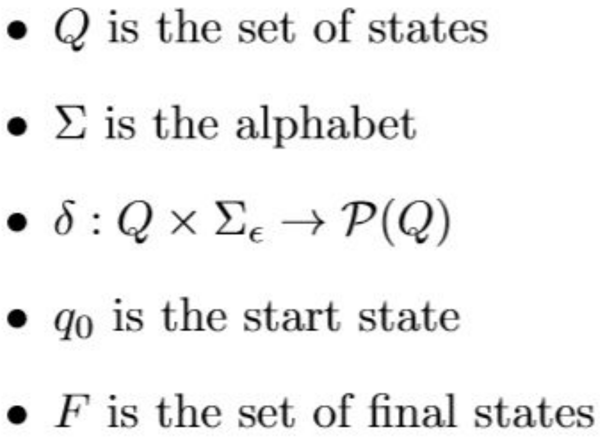
It's often useful to assign your states a "role" to reason about some a DFA. Here q1 means a single 1 has been seen. q2 means two 1s has been seen, and q3 means pattern found!



link

NFA Definition

NFA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$



- Q is the set of states
- Σ is the alphabet
- $\delta : Q \times \Sigma_\epsilon \to \mathcal{P}(Q)$
- q_0 is the start state
- F is the set of final states

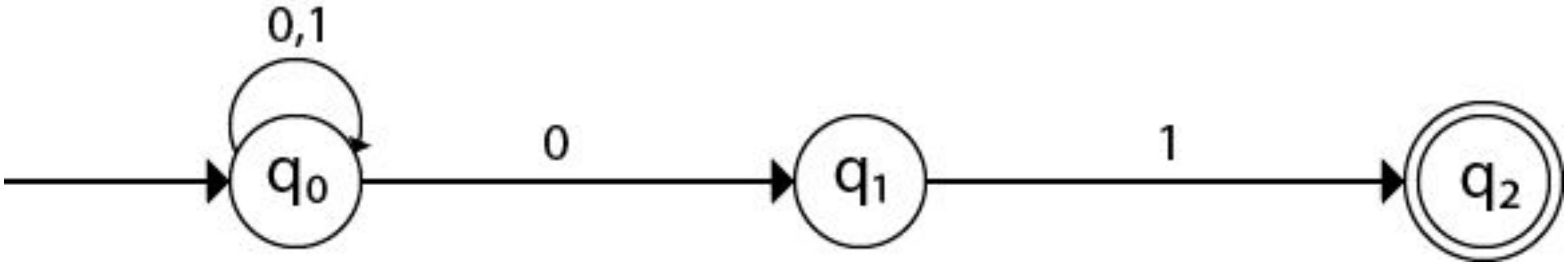
NFA vs DFA

- Nondeterminism manifests as the option of having multiple “next states” when consuming an input symbol in a given state
- ε-transitions: taken without consuming any input symbols i.e. spontaneously
- Acceptance condition: at least one branch of computation must end in an accept state

NFA Practice

What gives away the fact that this is an NFA?

Is 001 accepted? What about 010?

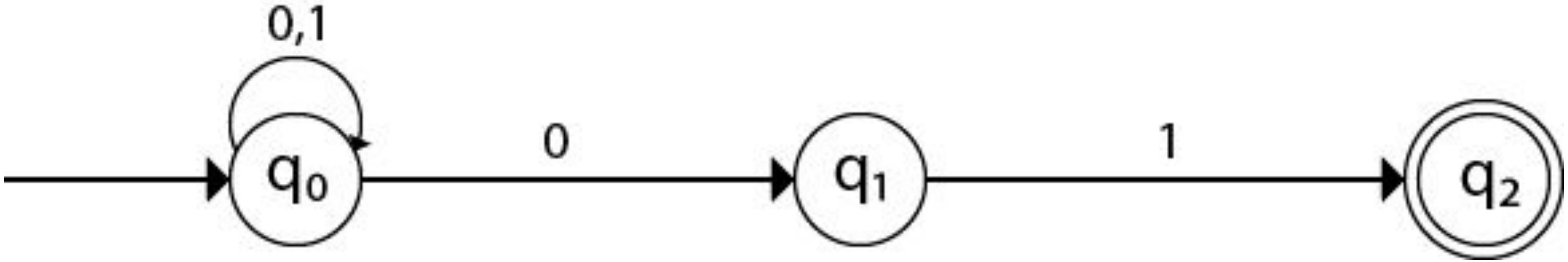


What is the language of this NFA? Give a plain English description and a regex.

NFA Practice

What gives away the fact that this is an NFA? q0 has two outgoing 0-edges

Is 001 accepted? Yes. What about 010? No, the computation gets “stuck.”

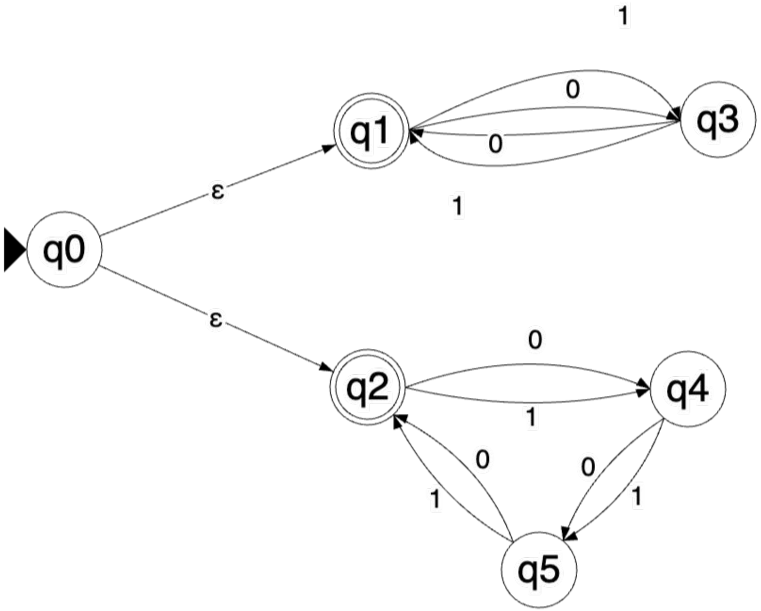


What is the language of this NFA? Give a plain English description and a regex.

Answer: Σ*01 or all strings ending with 01

Designing an NFA

Design an NFA that accepts strings of length that is either a multiple of 2 or a multiple of 3

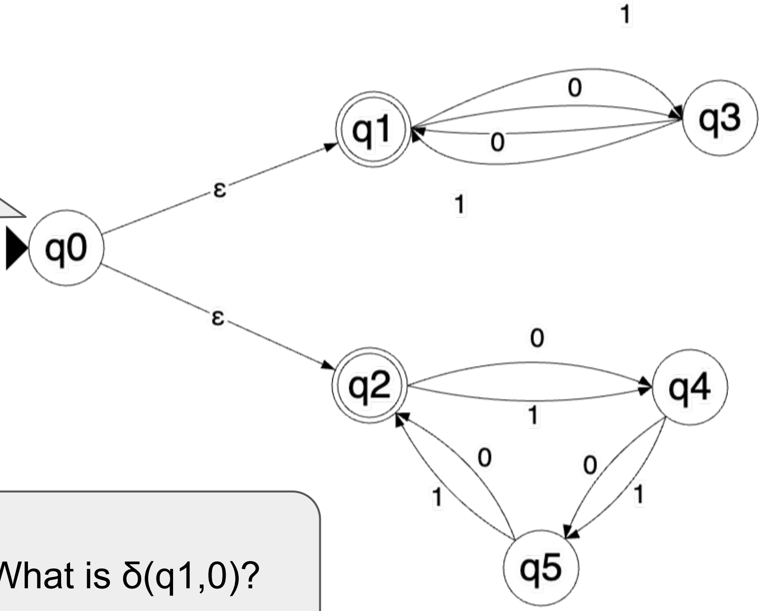


Do you notice anything? Is it easier than designing a DFA

link

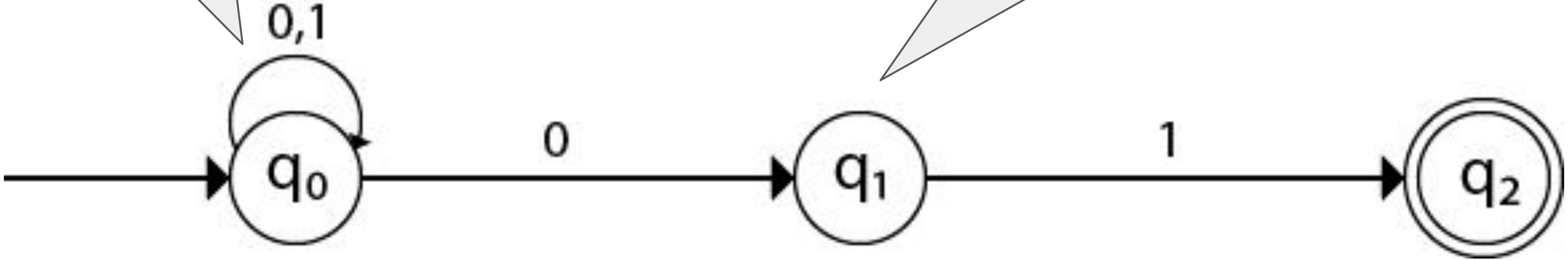
NFA Transition

What is δ(q0,ε)?



What is δ(q0,0)?

What is δ(q1,0)?


0,1
q_0
0
q_1
1
q_2