Recall Definitions

- DFA $M = (Q, \Sigma, \delta, q_0, F)$
- Language accepted by DFA:

  $L(M) = \{ w \mid \delta(q_0, w) \in F \}$

In terms of transition graph, there is a path labeled w from start state to a final state.

Given a language L and a DFA M, to show that our DFA design is correct we need to prove that L = L(M)

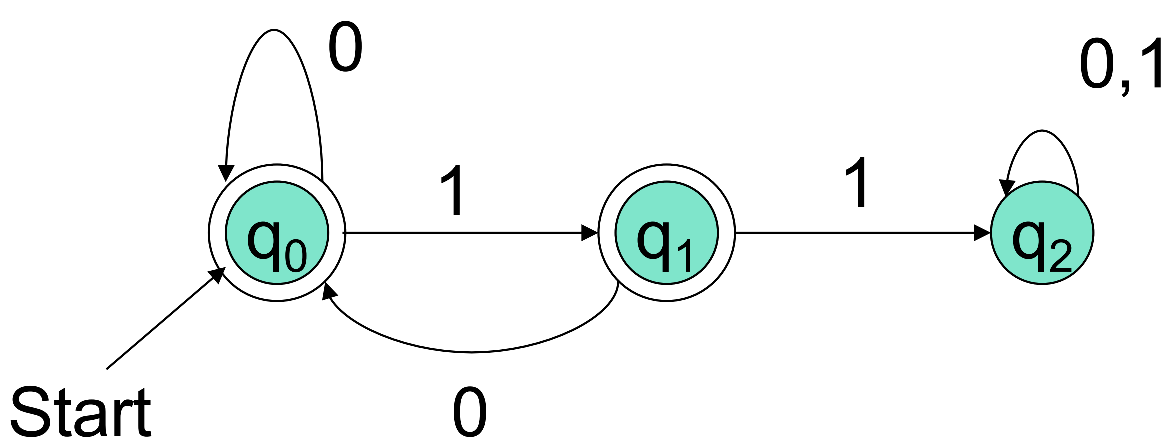
Example – Proving Property of language accepted by DFA

- The language of this example DFA is:

{w | w is in {0,1}* and w does not have

two consecutive 1's}

These conditions about w are true.



Proofs of Set Equivalence

- Often, we need to prove that two descriptions of sets are in fact the same set.
- Here, one set is “the language of this DFA,” and the other is “the set of strings of 0’s and 1’s with no consecutive 1’s.”

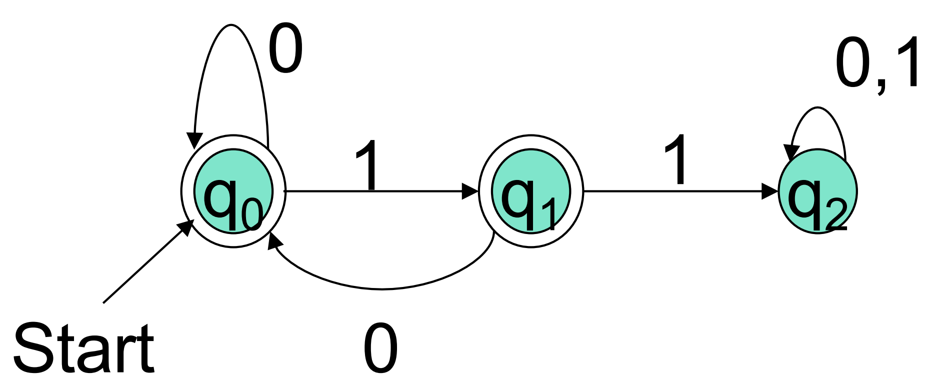
Proofs

- In general, to prove S = T, we need to prove two parts: S ⊆ T and T ⊆ S. That is:
  1. If w is in S, then w is in T.
  2. If w is in T, then w is in S.
- Here, S = the language of our running DFA, and T = "no consecutive 1's."

Part 1: $S \subseteq T$

- To prove: if w is accepted by

  then w has no consecutive 1's.
- Proof is an induction on length of w.
- *Important trick*: Expand the inductive hypothesis to be more detailed than the statement you are trying to prove.



The Inductive Hypothesis

1. If $\delta(q_0, w) = q_0$, then w has no consecutive 1's and does not end in 1.
2. If $\delta(q_0, w) = q_1$, then w has no consecutive 1's and ends in a single 1.

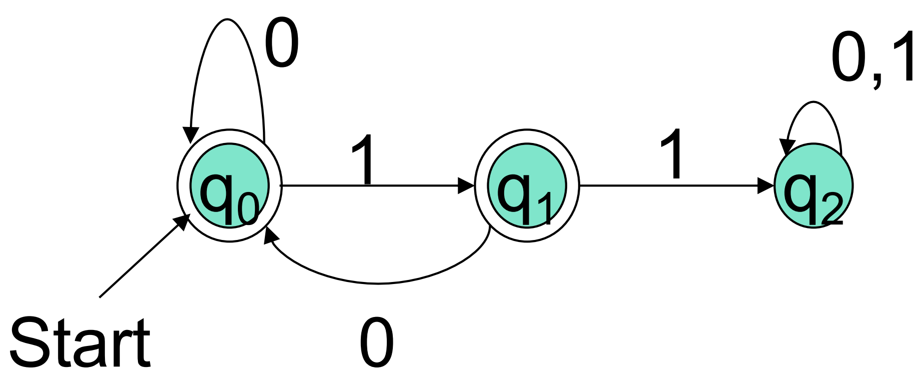
- Basis: $|w| = 0$; i.e., $w = \epsilon$.
  - (1) holds since ϵ has no 1's at all.
  - (2) holds *vacuously*, since $\delta(q_0, \epsilon)$ is not q_1.

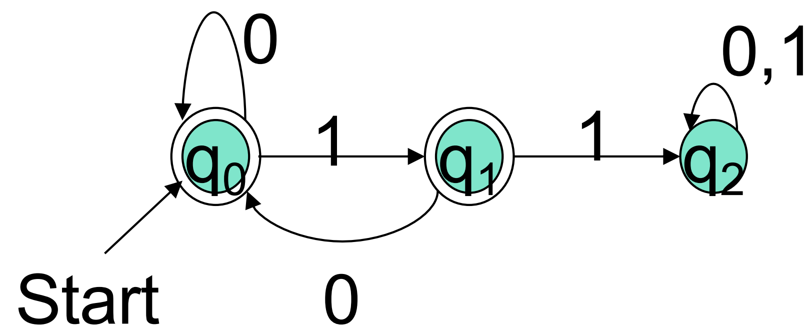
"length of"

Important concept:
If the "if" part of "if..then" is false, the statement is true.

Inductive Step

- Assume (1) and (2) are true for strings shorter than w, where |w| is at least 1.
- Because w is not empty, we can write w = xa, where *a* is the last symbol of w, and x is the string that precedes.
- IH is true for x.

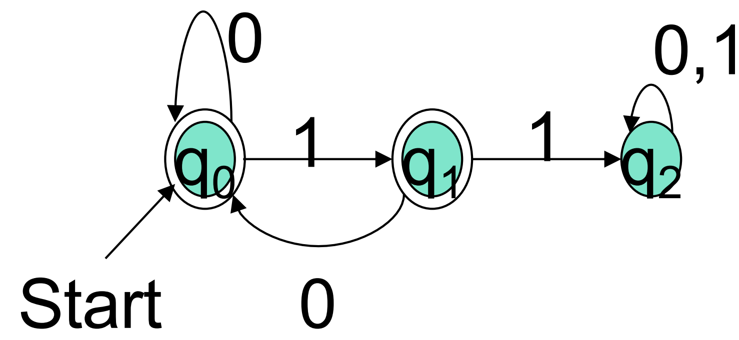




Inductive Step – (2)

- Need to prove (1) and (2) for w = xa.
- (1) for w is: If $\delta(q_0, w) = q_0$, then w has no consecutive 1's and does not end in 1.
- Since $\delta(q_0, w) = q_0$, $\delta(q_0, x)$ must be q_0 or q_1, and *a* must be 0 (look at the DFA).
- By the IH, x has no 11's.
- Thus, w has no 11's and does not end in 1.

Inductive Step – (3)



- Now, prove (2) for w = xa: If $\delta(q_0, w) = q_1$, then w has no 11's and ends in 1.
- Since $\delta(q_0, w) = q_1$, $\delta(q_0, x)$ must be q_0, and *a* must be 1 (look at the DFA).
- By the IH, x has no 11's and does not end in 1.
- Thus, w has no 11's and ends in 1.

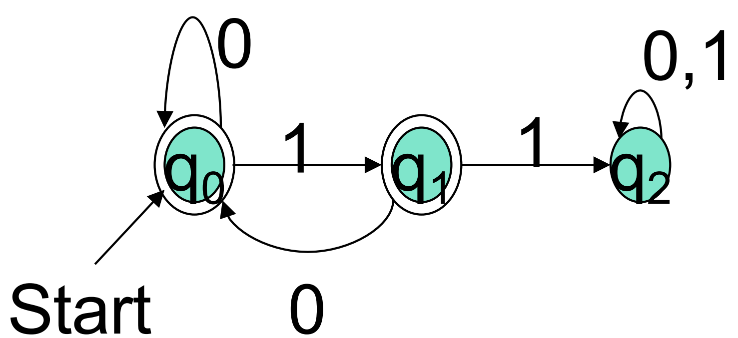
Part 2: T ⊆ S

X

- Now, we must prove: if w has no 11's,
  then w is accepted by DFA M

Y

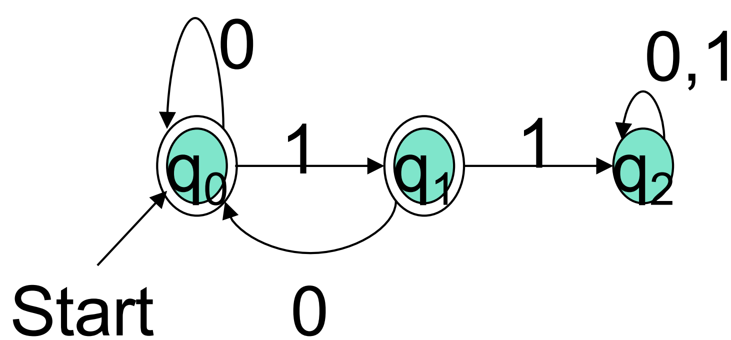
- *Contrapositive* : If w is **not** accepted by M then
  string w has 11



Key idea: contrapositive of "if X then Y" is the equivalent statement "if not Y then not X."

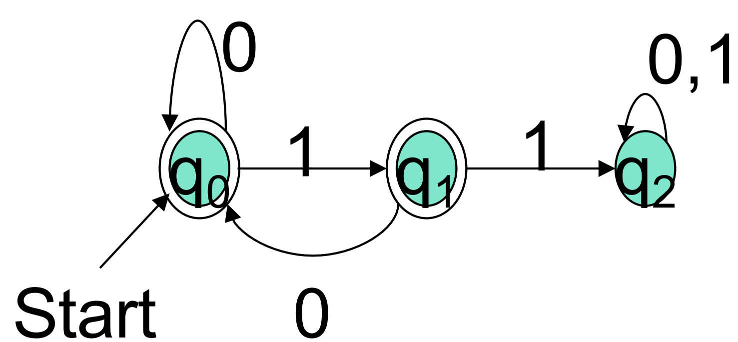
Using the Contrapositive

- Because there is a unique transition from every state on every input symbol, each w gets the DFA to exactly one state.
- The only way w is not accepted is if it gets to C.



Using the Contrapositive – (2)

- The only way to get to C [formally: δ(A,w) = C] is if w = x1y, x gets to B, and y is the tail of w that follows what gets to C for the first time.
- If δ(A,x) = B then surely x = z1 for some z.
- Thus, w = z11y and has 11.



Questions ?

Next: Using JFLAP to build and test your automata

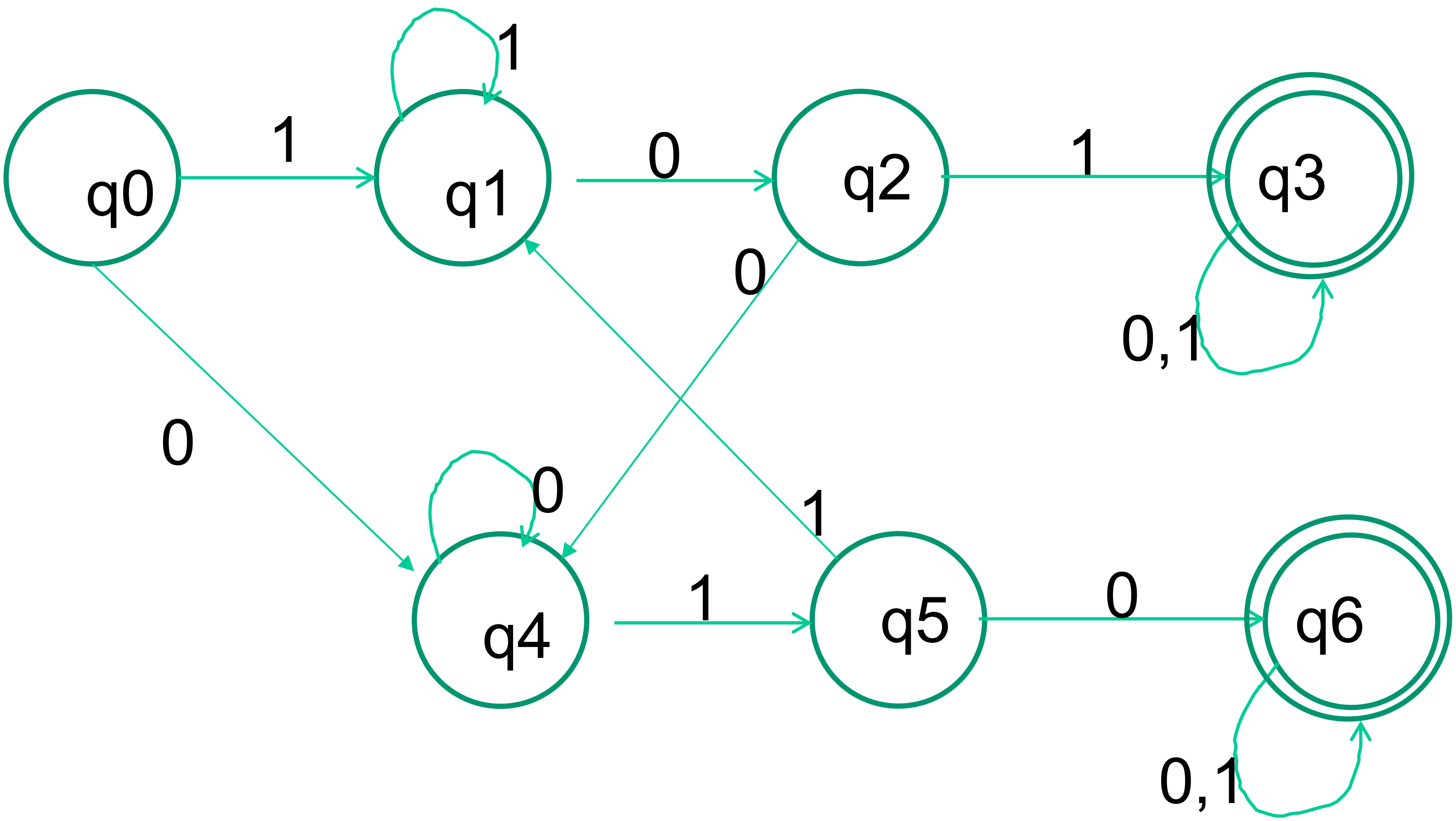
- JFLAP is a simulation tool
  - Specify your automaton
  - Test behavior of automaton on test input
  - Visualization of automaton

- Why use JFLAP
  - To test/debug your design
  - This means you need to come up with interesting test cases (including edge cases)

- JFLAP is not really a substitute for proving your automaton is correct !!

Example: DFA in JFLAP

- Provide a DFA in JFLAP for L = { w | w is a string in $\{0,1\}^*$ and w contains (a) the substring 101 or (b) substring 010 }



JFLAP Exercise: Work in breakout groups and submit individually

- Ques 1: (Required) Provide a DFA for L = { w | w is a string in $\{0,1\}^*$ and w contains (a) the substring 101 or (b) substring 100 or (c) substring 001. }

- Ques 2: (work and submit as optional question). Provide a DFA in JFLAP for L = { w | w is a string in $\{0,1\}^*$ and w contains the substring 101 with at most 1-bit of mis-match. }