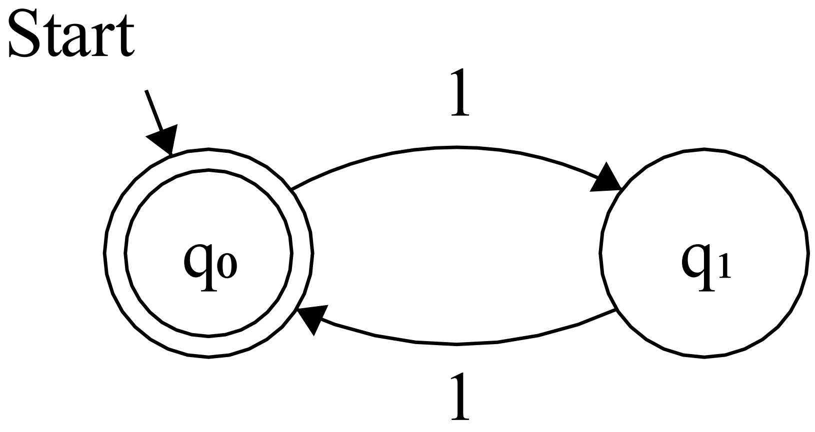
Outline for Proving Automata Correct

1. Create the automaton.
2. Give a definition for each state.
   - "If D is in state q_0, then the string must …"
3. Show that your automaton is correct in the base case.
   - This will usually be only one or two characters
4. Assuming that the automaton is in the correct state, prove that no matter what symbol is read next, it will continue to be in the next state.
5. Ensure that your final state(s) are defined correctly.

Example: Even number of 1s

This is automaton D:

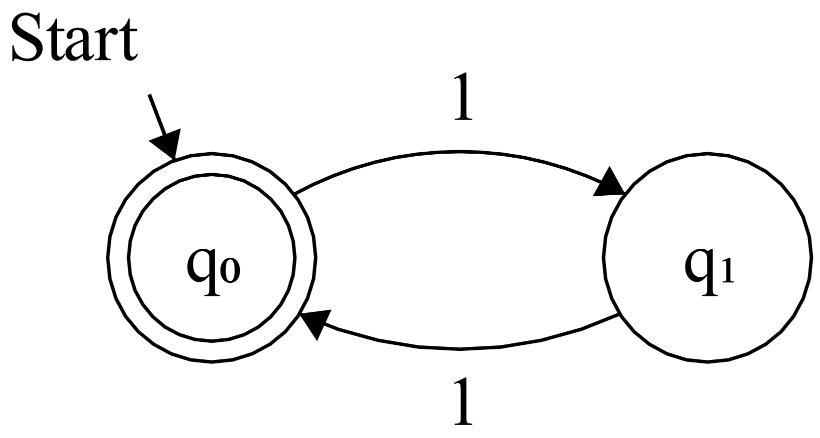


Claim: D accepts strings if and only if they contain an even number of 1s.

Proof:

We start by claiming that D is in state q_0 if and only if an even number of 1s have been read, and q_1 if and only if an odd number of 1s have been read.

Example: Even number of 1s

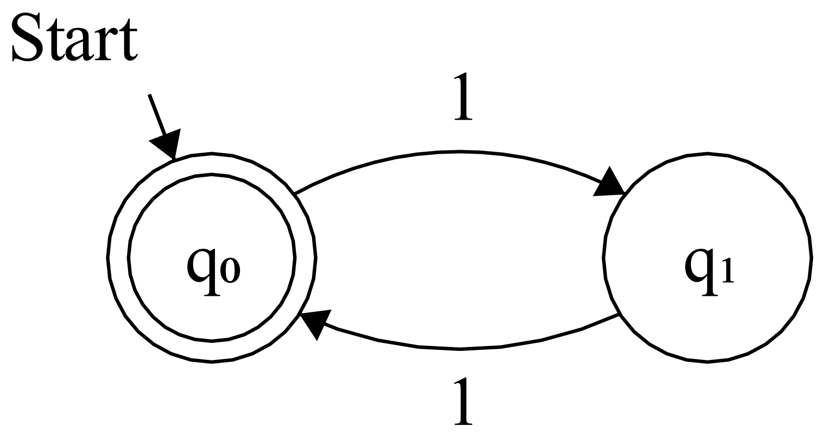


We will prove this by induction on a string x.

Base case: $|x| = 0$

The automaton starts in state q_0. Since a string of length 0 contains an even number of 1s, our claim about q_0 and q_1 is valid.

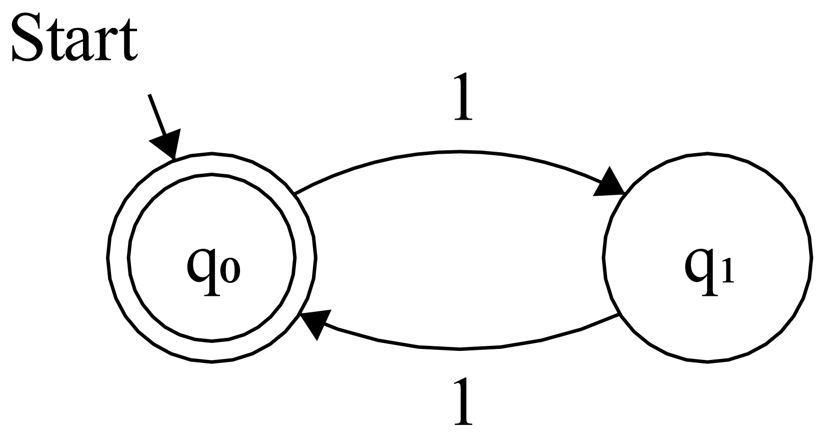
Example: Even number of 1s



Inductive step:

Assume D is in the correct state after reading string x. We will now show that it will be in the correct state after reading string xa, for a symbol a. (In this proof, a can only be a 1, since $\Sigma = \{1\}$.)

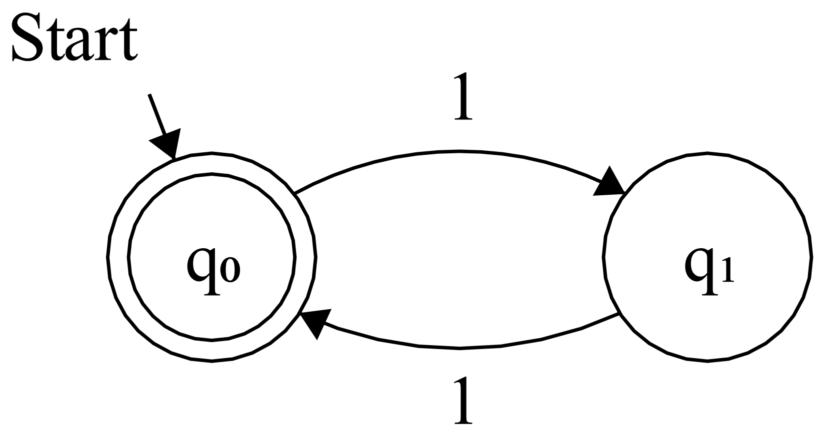
Example: Even number of 1s



Inductive step (continued):

Assume D is in state q_0 after reading x. By the inductive hypothesis, this means that x contained an even number of 1s. Reading another symbol means that the string will contain an odd number of 1s and be in state q_1, so our claims about q_0 and q_1 are valid.

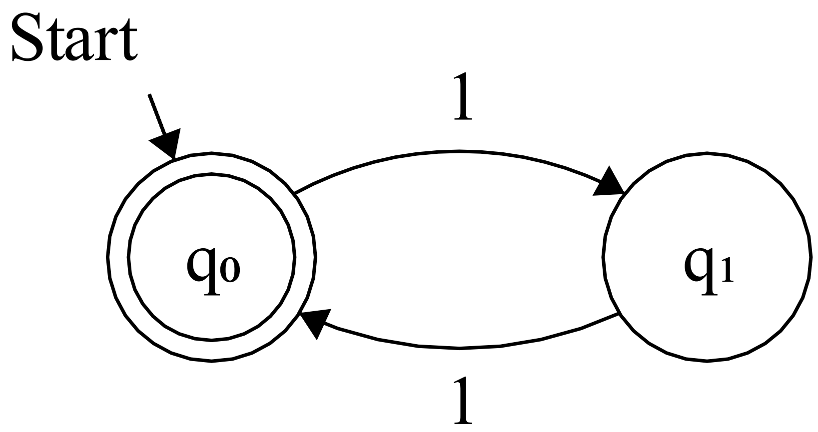
Example: Even number of 1s



Inductive step (continued):

Assume D is in state q_1 after reading x. By the inductive hypothesis, this means that x contained an odd number of 1s. Reading another symbol means that the string will contain an even number of 1s and be in state q_0, so our claims about q_0 and q_1 are valid.

Example: Even number of 1s



Finishing up:

D's only accepting state is q_0, and we just proved by induction that D is in q_0 if and only if it D has read an even number of 1s.

Example: Binary Number Divisible by 7

Objective: Write a DFA D where $\Sigma = \{0, 1\}$ and $L(D) = \{x \mid \text{when interpreted as a binary number, } x \text{ is evenly divisible by } 7\}$.

How should we write this automaton?

Example: Binary Number Divisible by 7

Objective: Write a DFA D where $\Sigma = \{0, 1\}$ and $L(D) = \{x \mid$ when interpreted as a binary number, x is evenly divisible by $7\}$.

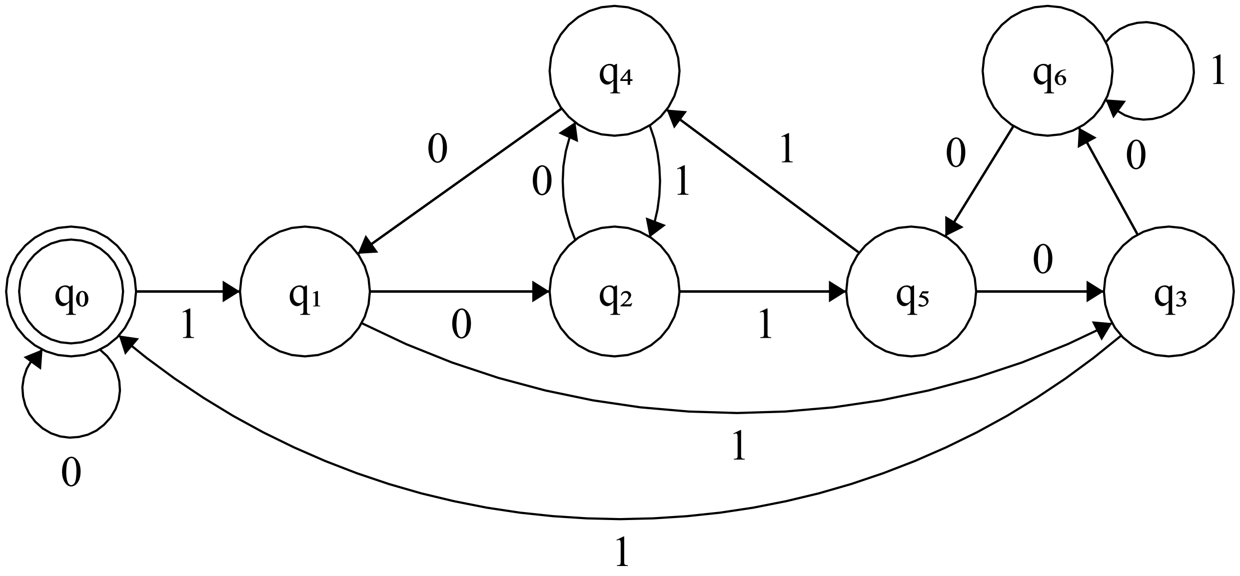
- We will use modular arithmetic to construct D.
- Define states q_0 through q_6 where D is in q_i if and only if the string read so far is equal to i mod 7.
- Any number divisible by 7 is equal to 0 mod 7.
- Exploit modular arithmetic to define δ
  - $2x$ mod 7 = 2(x mod 7) mod 7
  - $(2x + 1)$ mod 7 = (2(x mod 7) + 1) mod 7

Example: Binary Number Divisible by 7

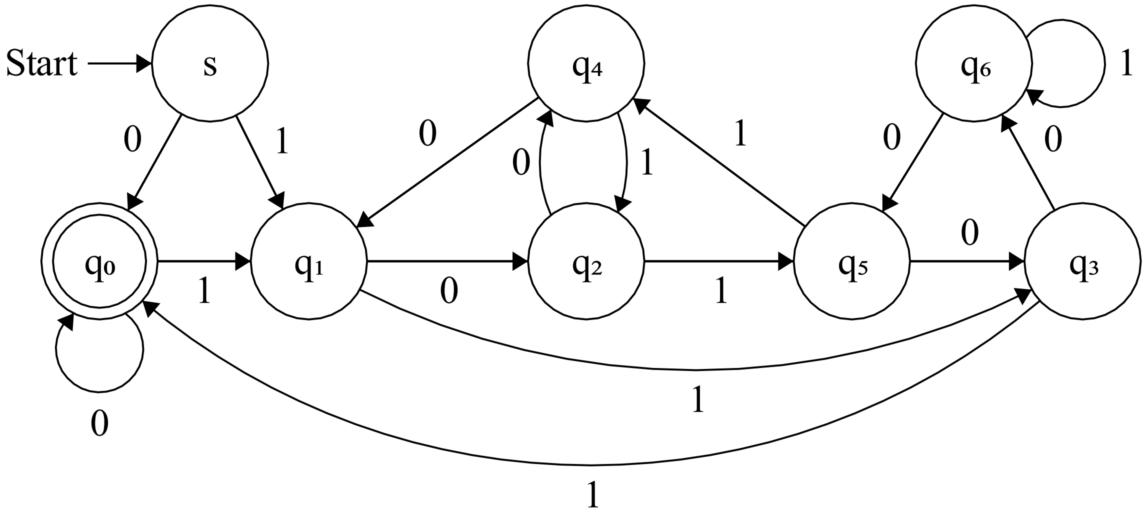
- Exploit modular arithmetic to define δ
  - $2x$ mod 7 = 2(x mod 7) mod 7
  - $(2x + 1)$ mod 7 = (2(x mod 7) + 1) mod 7
- By the definition of base-2 numbers:
  - Reading a 0 doubles the value we've read so far
  - Reading a 1 doubles the value read so far and adds 1
- $\delta(q_i, 0) = q_{2i \bmod 7}$
- $\delta(q_i, 1) = q_{(2i+1) \bmod 7}$

- $\delta(q_i, 0) = q_{2i \bmod 7}$
- $\delta(q_i, 1) = q_{(2i+1) \bmod 7}$
- $2v$ mod 7 = 2(v mod 7) mod 7
- $(2v + 1)$ mod 7 = (2(v mod 7) + 1) mod 7

Example: Binary Number Divisible by 7

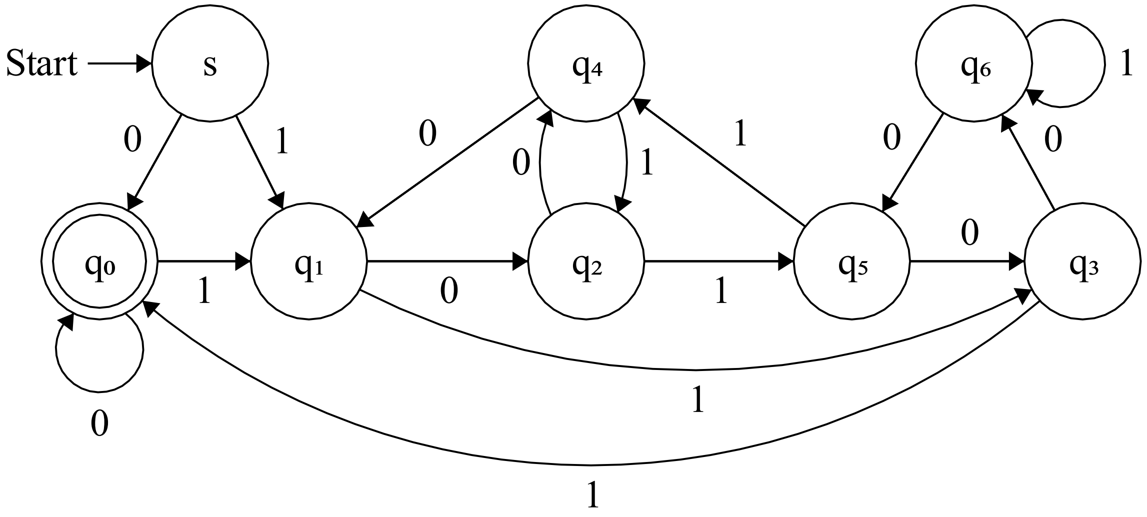


Example: Binary Number Divisible by 7



We don't want to accept an empty string, so we'll create an additional start state.

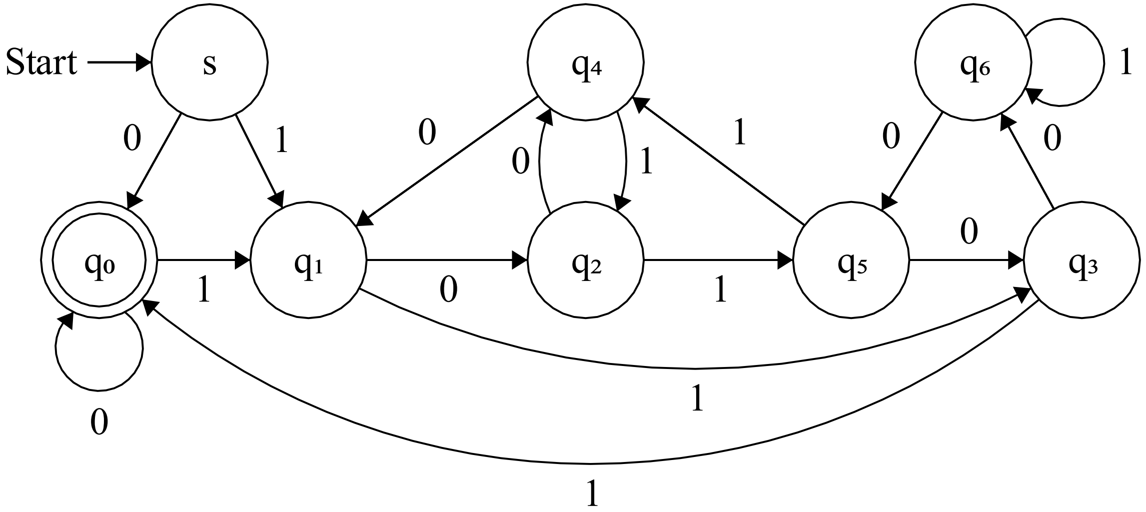
Example: Binary Number Divisible by 7



Let x be the string read so far, and v be the value of x when interpreted as a binary number.

Claim: D is in state q_i if and only if $v \bmod 7 = i$. We will prove this by induction on x.

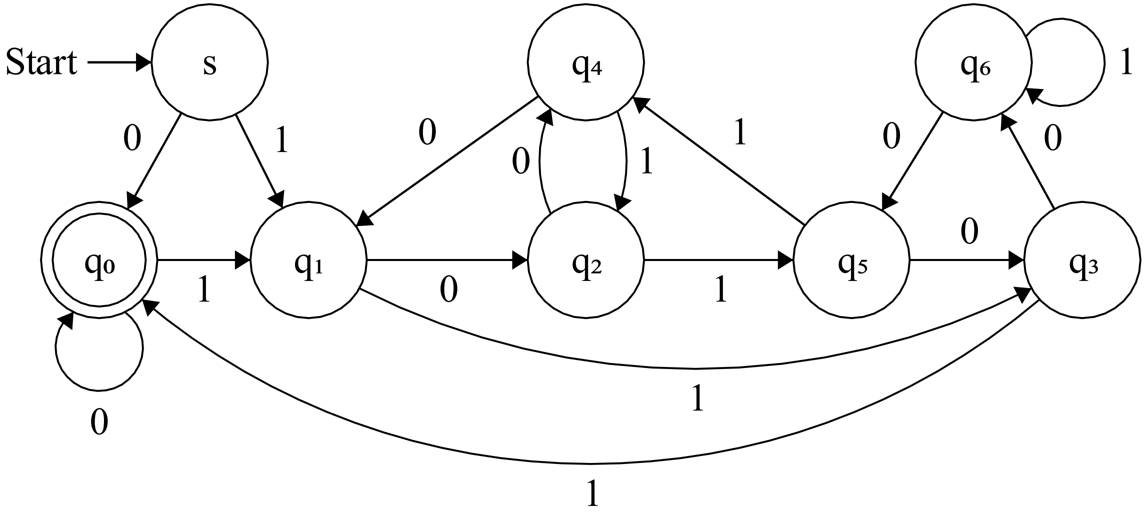
Example: Binary Number Divisible by 7



Base case: $|x| = 1$. (Note that the claim *doesn't* hold for $|x| = 0$.

v is either 0 or 1, if x was 0 or 1, respectively. If $v = 0$, then x was 0, so D is in q_0. Otherwise, x was 1 and D is in q_1. In either case, the claim holds.

Example: Binary Number Divisible by 7

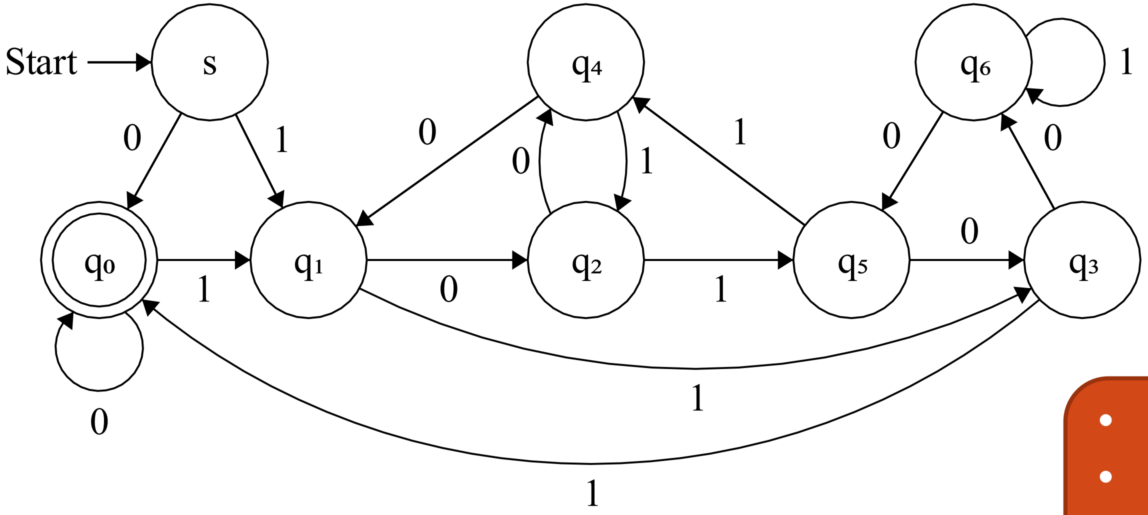


Inductive step:

Assume that D is in the correct state after reading string x, and v is the correct value. After reading a character a, the new value of v will be:

- $2v$, if $a = 0$
- $2v + 1$, if $a = 1$

Example: Binary Number Divisible by 7



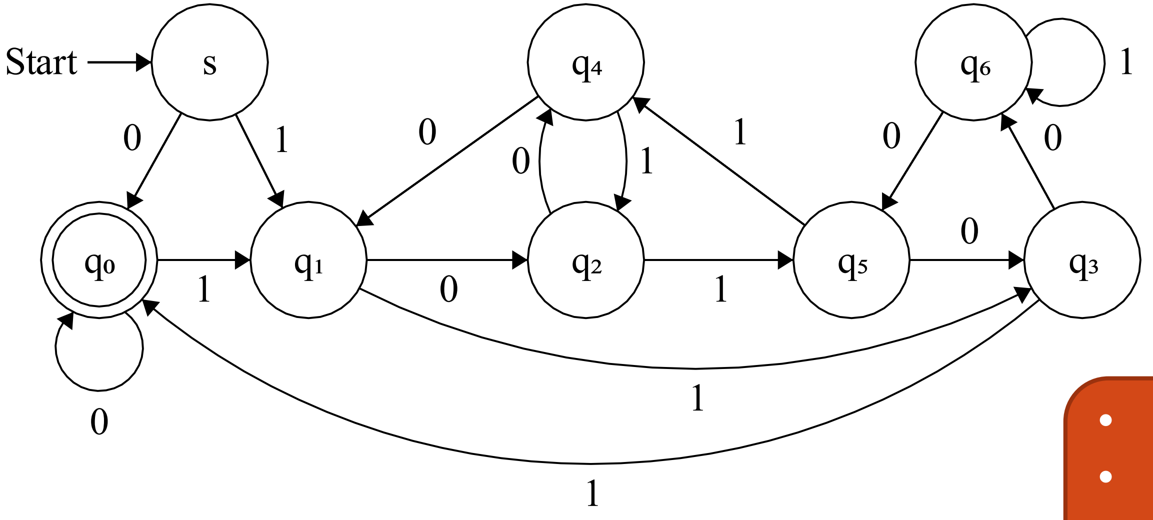
- $\delta(q_i, 0) = q_{2i \bmod 7}$
- $\delta(q_i, 1) = q_{(2i+1) \bmod 7}$
- $2v$ mod 7 = $2(v$ mod 7) mod 7
- $(2v + 1)$ mod 7 = $(2(v$ mod 7) + 1) mod 7

Inductive step, continued:

Recall that D is in state q_i if and only if $v \bmod 7 = i$.

This property holds if a is either a 0 or a 1, by the definitions of δ, v, and properties of modular arithmetic.

Example: Binary Number Divisible by 7



- $\delta(q_i, 0) = q_{2i \bmod 7}$
- $\delta(q_i, 1) = q_{(2i+1) \bmod 7}$
- $2v \bmod 7 = 2(v \bmod 7) \bmod 7$
- $(2v + 1) \bmod 7 = (2(v \bmod 7) + 1) \bmod 7$

Finishing up:

D accepts x if and only if it ends in state q_0. Since D is in q_0 if and only if $v \bmod 7 = 0$, and v is equal to x interpreted as a binary number, D accepts x if and only if, when interpreted as a binary number, x is divisible by 7.

Working With Lots of States

- The previous can be extended to any positive integer.
- For example, we could describe an automaton that detects divisibility by 10,007:
  - $Q = \{q_i \mid 0 \leq i < 10{,}007\} \cup \{s\}$ (s is the start state)
  - $\Sigma = \{0, 1\}$
  - δ:
    - $\delta(s, 0) = q_0$
    - $\delta(s, 1) = q_1$
    - $\delta(q_i, 0) = q_{2i \bmod 10007}$
    - $\delta(q_i, 1) = q_{(2i+1) \bmod 10007}$
  - $F = \{q_0\}$
- Note that we can fully describe this automaton in terms of i even if we can't draw a diagram for it.