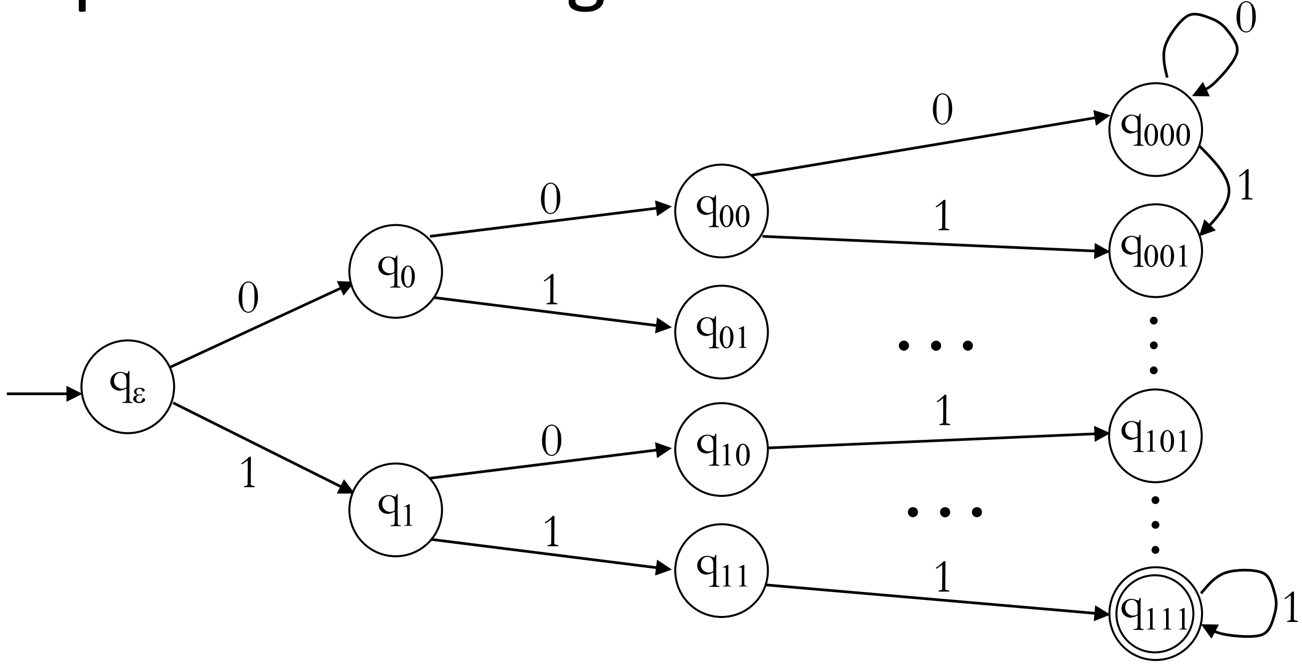
Lesson 5: DFA minimization algorithm

Marc Gaetano

Edition 2018

Example

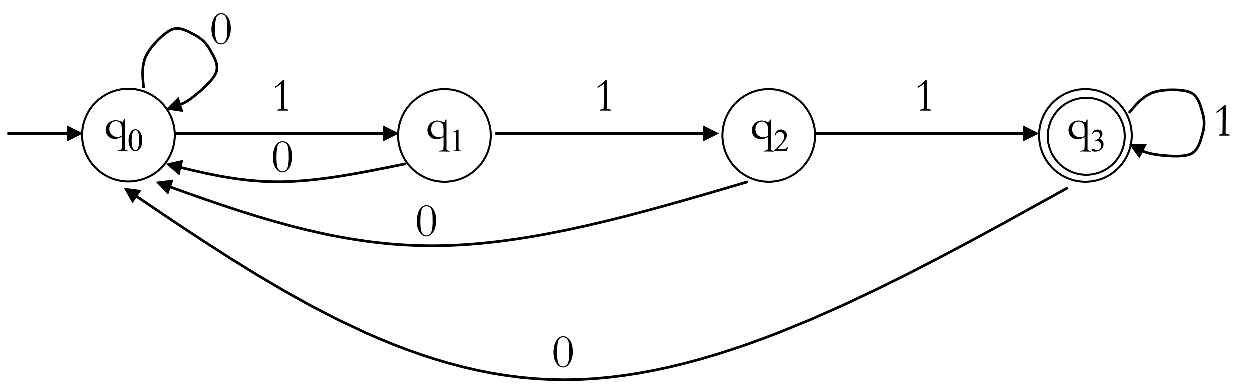
- Construct a DFA over alphabet $\{0, 1\}$ that accepts those strings that end in 111



- This is big, isn't there a smaller DFA for this?

Smaller DFA

- Yes, we can do it with 4 states:



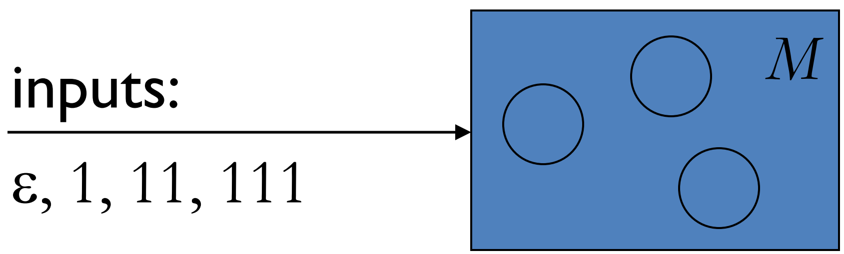
  – The state remembers the number of consecutive 1s at the end of the string (up to 3)

- Can we do it with 3 states?

Even smaller DFA?

- Suppose we had a 3 state DFA M for L
- We do not know what this M looks like

… but let's imagine what happens when:



- By the pigeonhole principle, on two of these inputs M ends in the same state

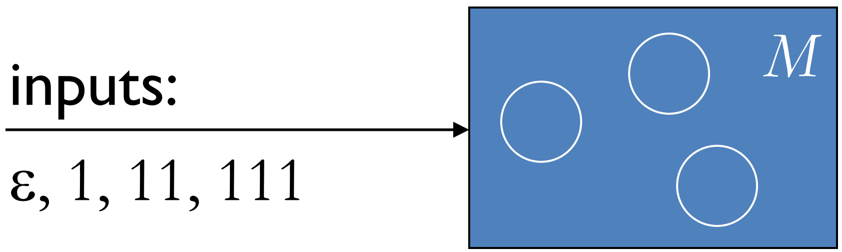
Pigeonhole principle

Suppose you are tossing m balls into n bins, and $m > n$. Then two balls end up in the same bin.

- Here, balls are inputs, bins are states:

If you have a DFA with n states and you run it on m inputs, and $m > n$, then two inputs end up in same state.

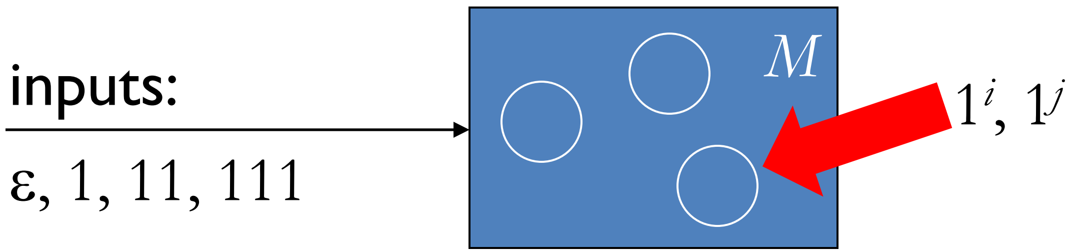
A smaller DFA?



- What goes wrong if…
  - M ends up in same state on input 1 and input 111?
  - M ends up in same state on input ε and input 11?

A smaller DFA

- Suppose M ends up in the same state after reading inputs $x = 1^i$ and $y = 1^j$, where $0 \leq i < j \leq 3$



- Then after reading the continuation $z = 1^{3-j}$
  – The state of yz should be accepting ($yz = 111$)
  – The state of xz should be rejecting ($xz = \varepsilon$, 1, or 11)

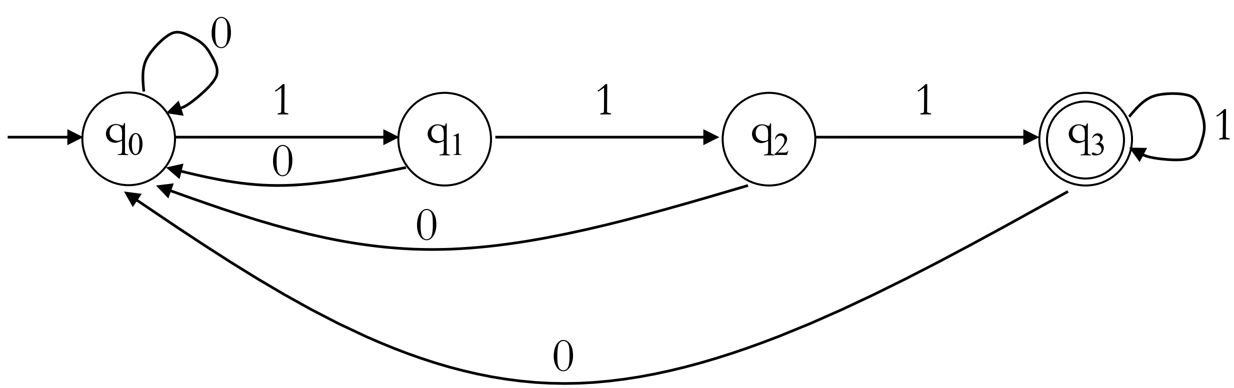
… but they are both in the same state!

No smaller DFA!

- Conclusion

  There is no DFA with 3 states for L

- So, this DFA is minimal

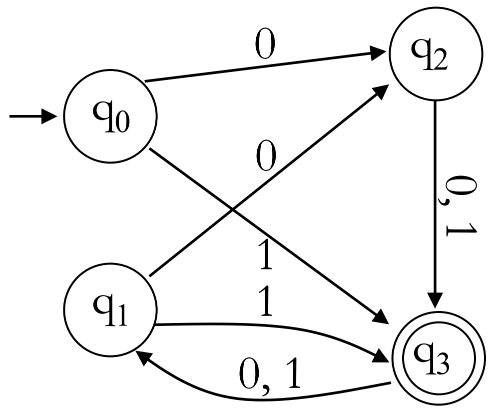


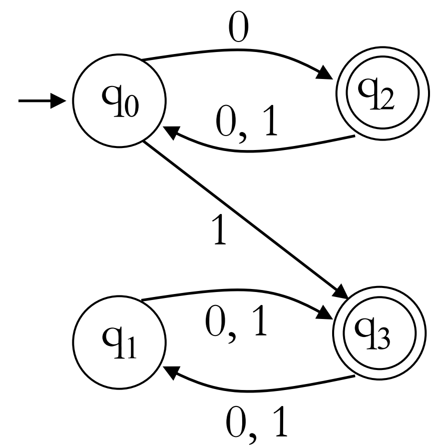
  – In fact, it is the unique minimal DFA for L

DFA minimization

- There is an algorithm to start with any DFA and reduce it to the smallest possible DFA
- The algorithm attempts to identify classes of equivalent states
- These are states that can be merged together without affecting the answer of the computation

Examples of equivalent states





q_0, q_1 equivalent

q_0, q_1 equivalent

q_2, q_3 also equivalent

Equivalent and distinguishable states

- Two states q, q' are equivalent if

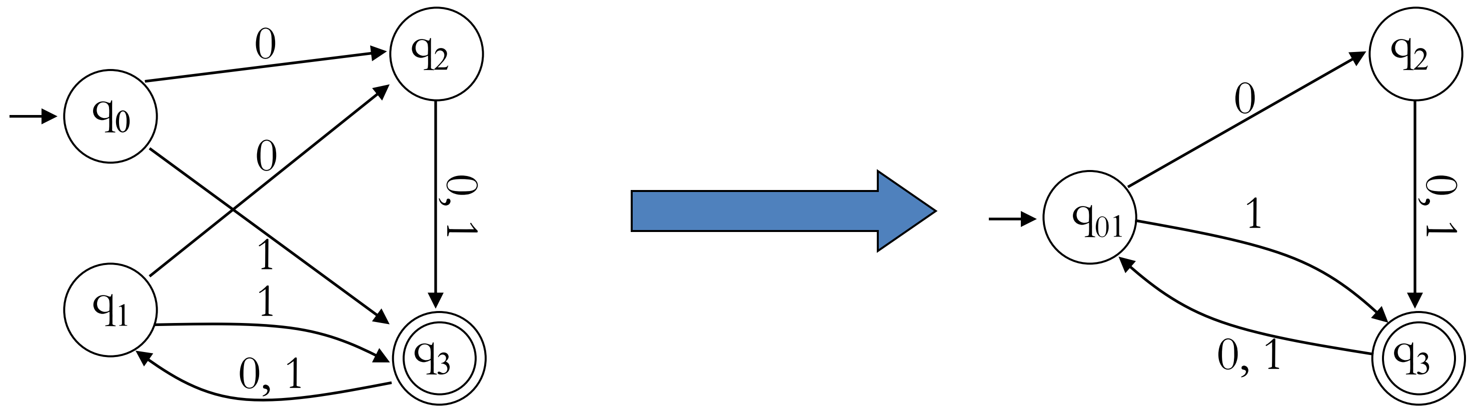
  For every string w, the states $\hat{\delta}(q, w)$ and $\hat{\delta}(q', w)$ are either both accepting or both rejecting

  – Here, $\hat{\delta}(q, w)$ is the state that the machine is in if it starts at q and reads the string w

- q, q' are distinguishable if they are not equivalent:

  For some string w, one of the states $\delta(q, w)$, $\delta(q', w)$ is accepting and the other is rejecting

Examples of distinguishable states



q_3 distinguishable from q_0, q_1, q_2

q_3 is accepting, others are rejecting

(q_0, q_2) and (q_1, q_2) distinguishable

they behave differently on input 0

q_0, q_1 equivalent

DFA minimization algorithm

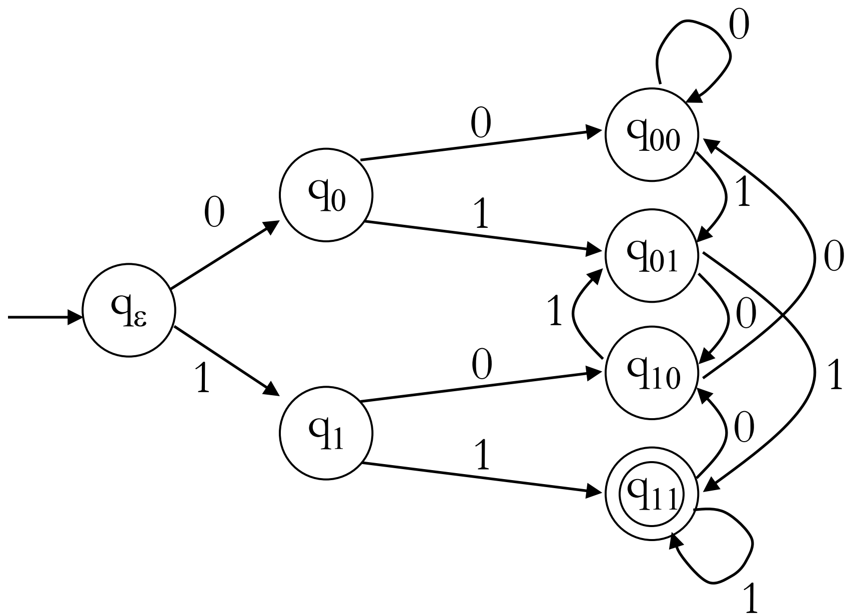
- Find all pairs of distinguishable states as follows:

① For any pair of states q, q':
    If q is accepting and q' is rejecting
        Mark (q, q') as distinguishable

② Repeat until nothing is marked:
    For any pair of states (q, q'):
        For every alphabet symbol a:
            If (δ(q, a), δ(q', a)) are marked as distinguishable
                Mark (q, q') as distinguishable

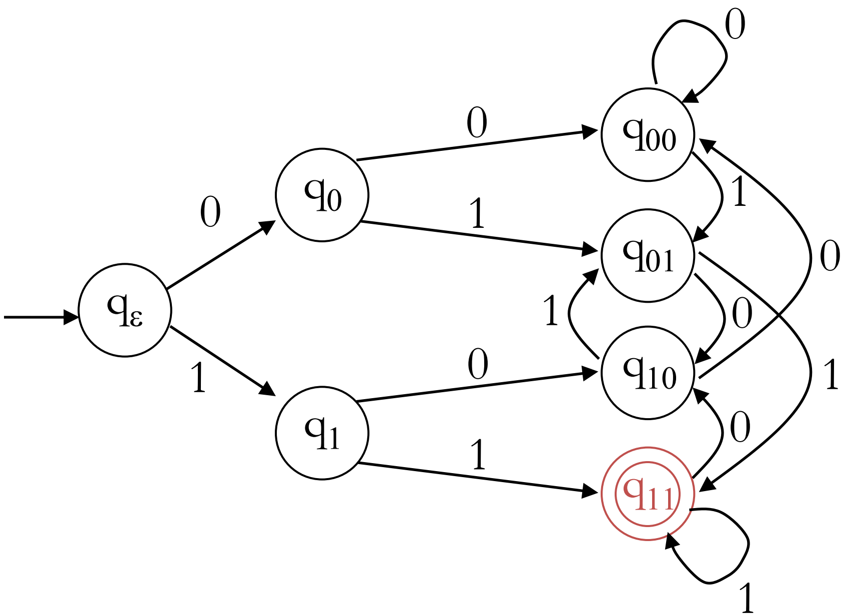
③ For any pair of states (q, q'):
    If (q, q') is not marked as distinguishable
        Merge q and q' into a single state

Example of DFA minimization



q_0						
q_1						
q_{00}						
q_{01}						
q_{10}						
q_{11}						
	q_ε	q_0	q_1	q_{00}	q_{01}	q_{10}

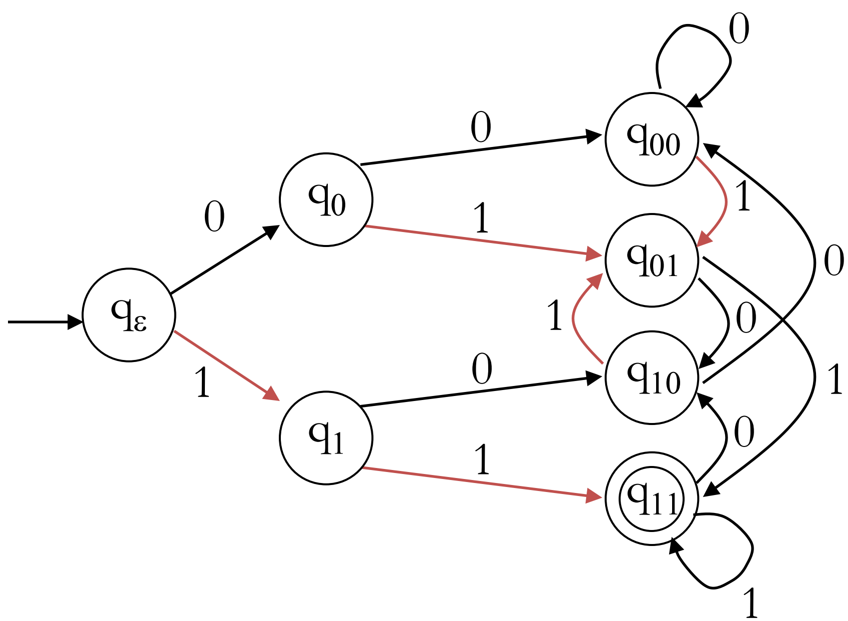
Example of DFA minimization



q_0						
q_1						
q_{00}						
q_{01}						
q_{10}						
q_{11}	x	x	x	x	x	x
	q_ε	q_0	q_1	q_{00}	q_{01}	q_{10}

① q_{11} is distinguishable from all other states

Example of DFA minimization

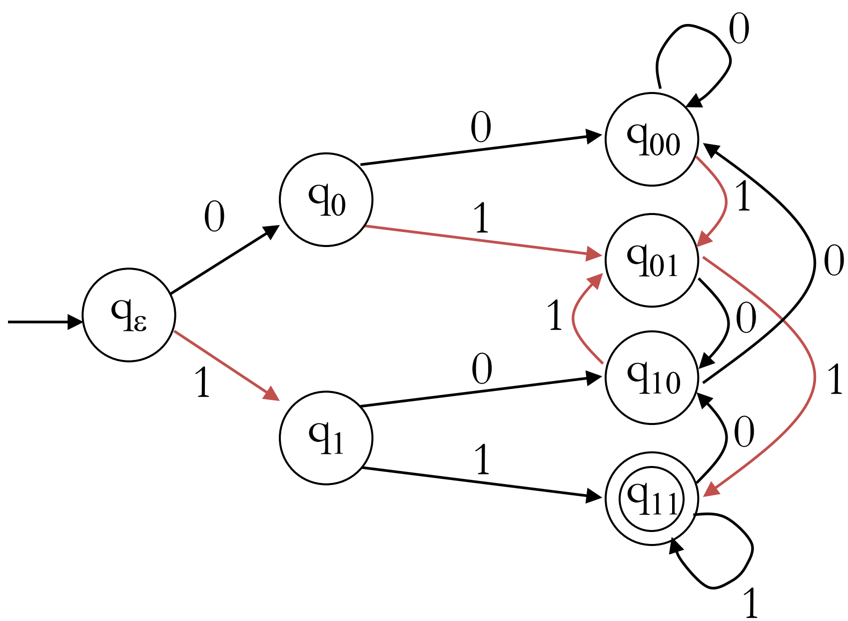


	q_ε	q_0	q_1	q_{00}	q_{01}	q_{10}
q_0						
q_1	x	x				
q_{00}			x			
q_{01}						
q_{10}			x			
q_{11}	x	x	x	x	x	x

② q_1 is distinguishable from q_ε, q_0, q_{00}, q_{10}

On transition 1, they go to distinguishable states

Example of DFA minimization

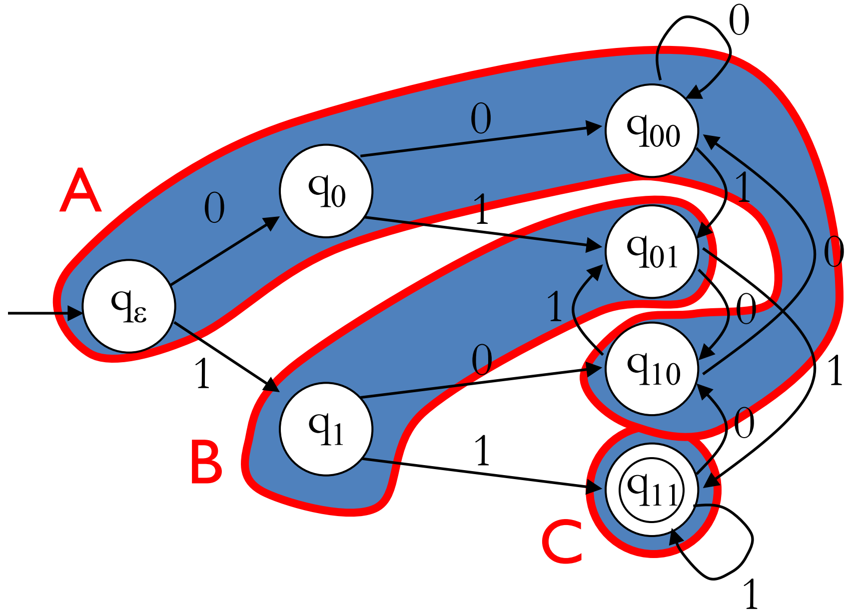


q_0						
q_1	x	x				
q_{00}			x			
q_{01}	x	x		x		
q_{10}			x		x	
q_{11}	x	x	x	x	x	x
	q_ε	q_0	q_1	q_{00}	q_{01}	q_{10}

② q_{01} is distinguishable from $q_\varepsilon, q_0, q_{00}, q_{10}$

On transition 1, they go to distinguishable states

Example of DFA minimization



q_0	A					
q_1	x	x				
q_{00}	A	A	x			
q_{01}	x	x	B	x		
q_{10}	A	A	x	A	x	
q_{11}	x	x	x	x	x	x
	q_ε	q_0	q_1	q_{00}	q_{01}	q_{10}

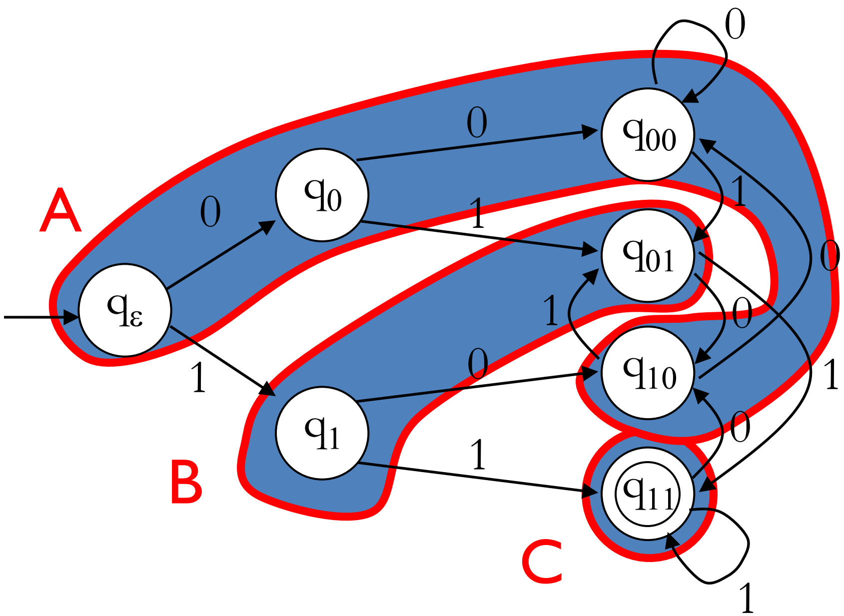
③ Merge states not marked distinguishable

q_ε, q_0, q_{00}, q_{10} are equivalent → group A

q_1, q_{01} are equivalent → group B

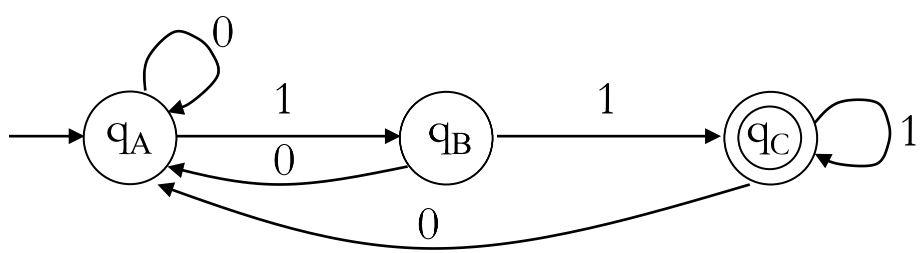
q_{11} cannot be merged → group C

Example of DFA minimization



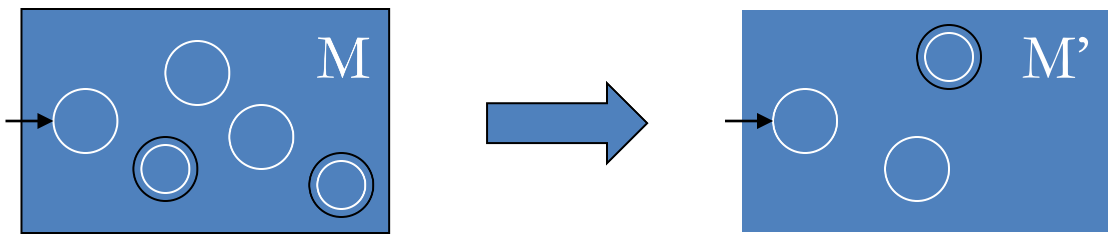
	q_ε	q_0	q_1	q_{00}	q_{01}	q_{10}
q_0	A					
q_1	x	x				
q_{00}	A	A	x			
q_{01}	x	x	B	x		
q_{10}	A	A	x	A	x	
q_{11}	x	x	x	x	x	x

minimized DFA:



Why does DFA minimization work?

- We need to convince ourselves of three properties:



① Consistency
  - The new DFA M' is well-defined

② Correctness
  - The new DFA M' is equivalent to the original DFA

③ Minimality
  - The new DFA M' is the smallest DFA possible for L

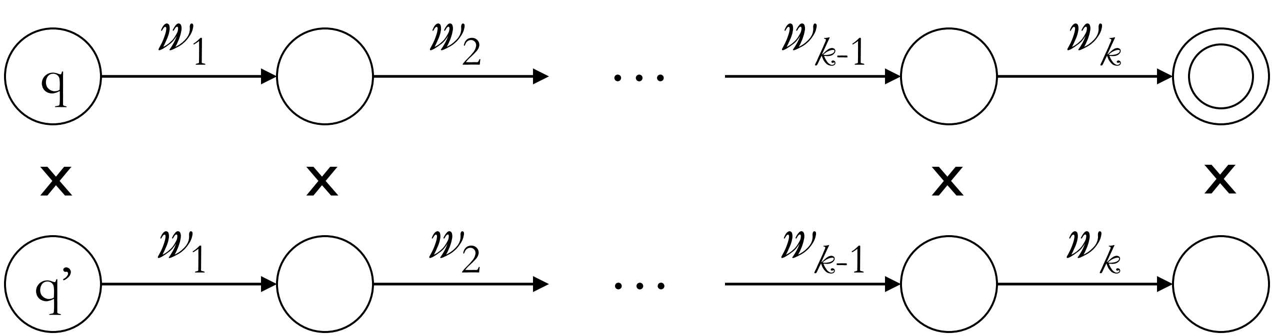
Main claim

Any two states q and q' in M are distinguishable if and only if the algorithm marks them as distinguishable

- Proof outline:
  - First, we assume q, q' are marked distinguishable, and show they must be distinguishable
  - Then, we assume q, q' are not marked distinguishable, and show they must be equivalent

Proof of part 1

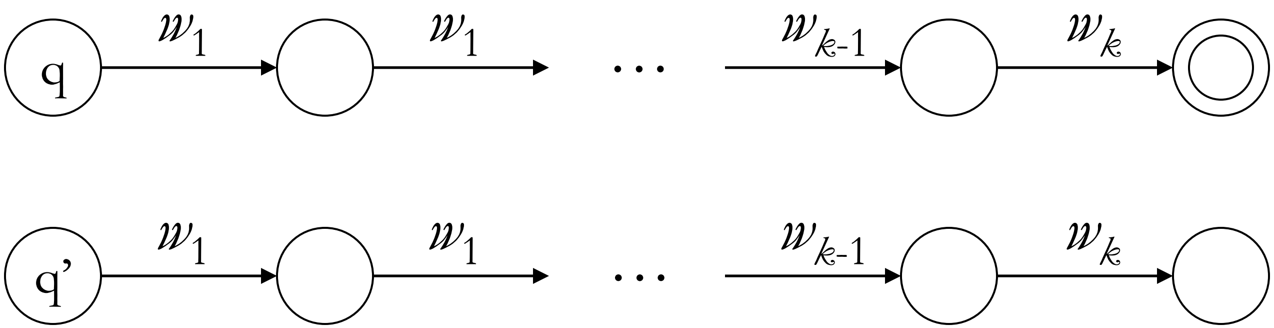
- First, suppose q, q' are marked as distinguishable
- Could it be that they are equivalent?
  - No, because recall how the algorithm works:



Proof idea for part 2

- Now suppose q, q' are not marked as distinguishable
- Could it be that they are distinguishable? Suppose so
- Then for some string w, $\hat{\delta}(q, w)$ accepts, but $\hat{\delta}(q', w)$ rejects
- Working backwards, the algorithm will mark q and q' as distinguishable at some point

Proof of part 2

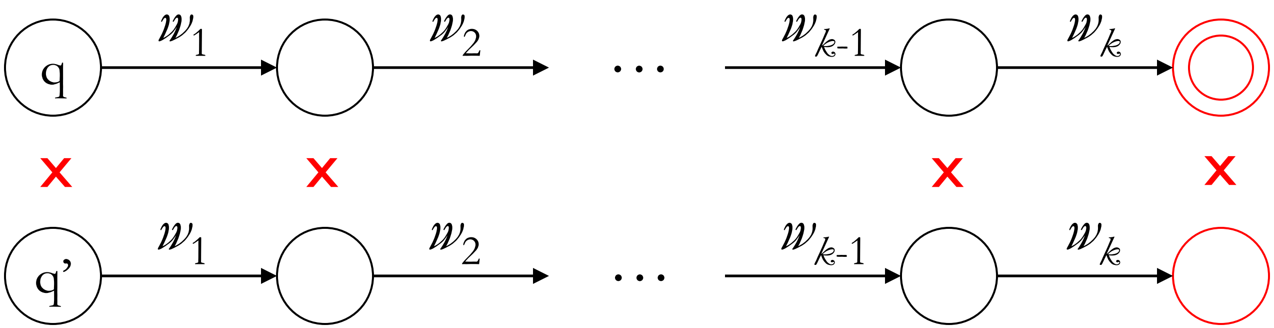


① For any pair of states q, q':

If q is accepting and q' is rejecting

Mark (q, q') as distinguishable

Proof of part 2



② Repeat until nothing is marked:

For any pair of states (q, q’):

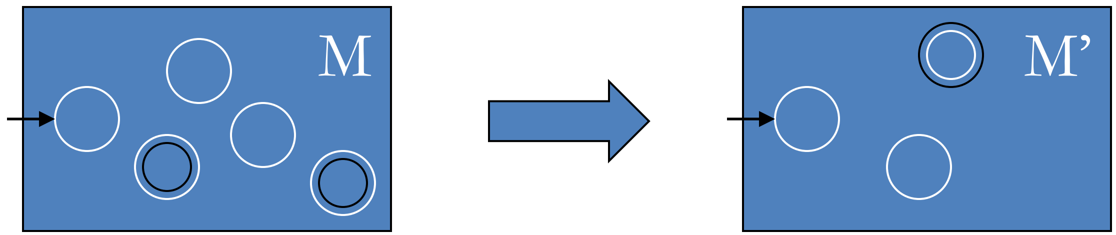
For every alphabet symbol a:

If (δ(q, a), δ(q’, a)) are marked as distinguishable

Mark (q, q’) as distinguishable

Why does DFA minimization work?

- We need to convince ourselves of three properties:



① Consistency

- The new DFA $\mathrm{M'}$ is well-defined

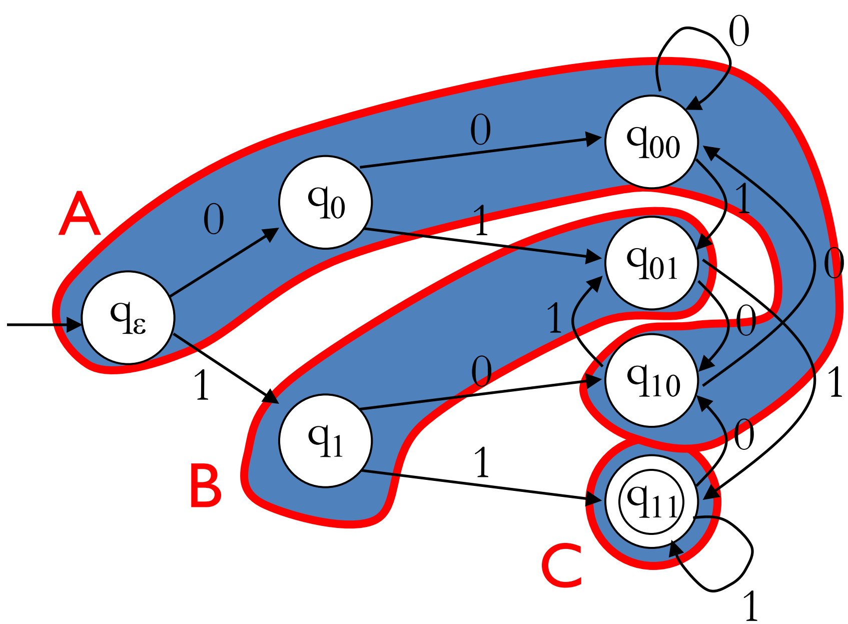
② Correctness

- The new DFA $\mathrm{M'}$ is equivalent to the original DFA

③ Minimality

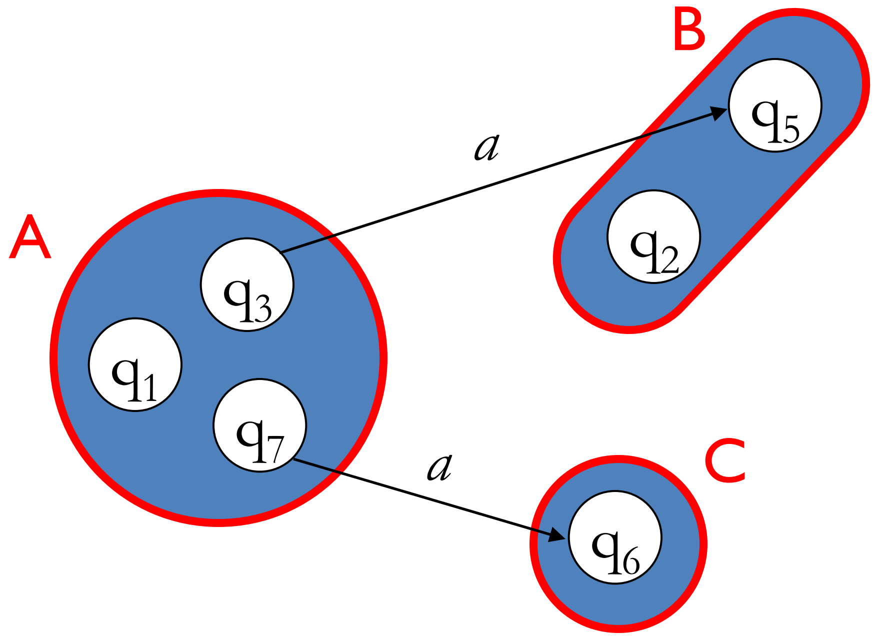
- The new DFA $\mathrm{M'}$ is the smallest DFA possible for L

Consistency of transitions



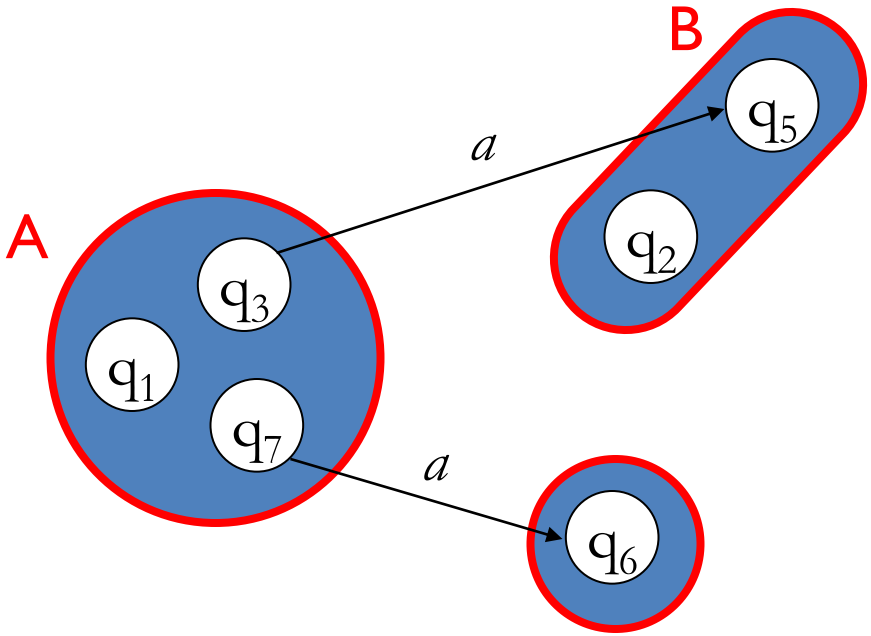
- Why are the transitions between the merged states consistent?

Proof of consistency



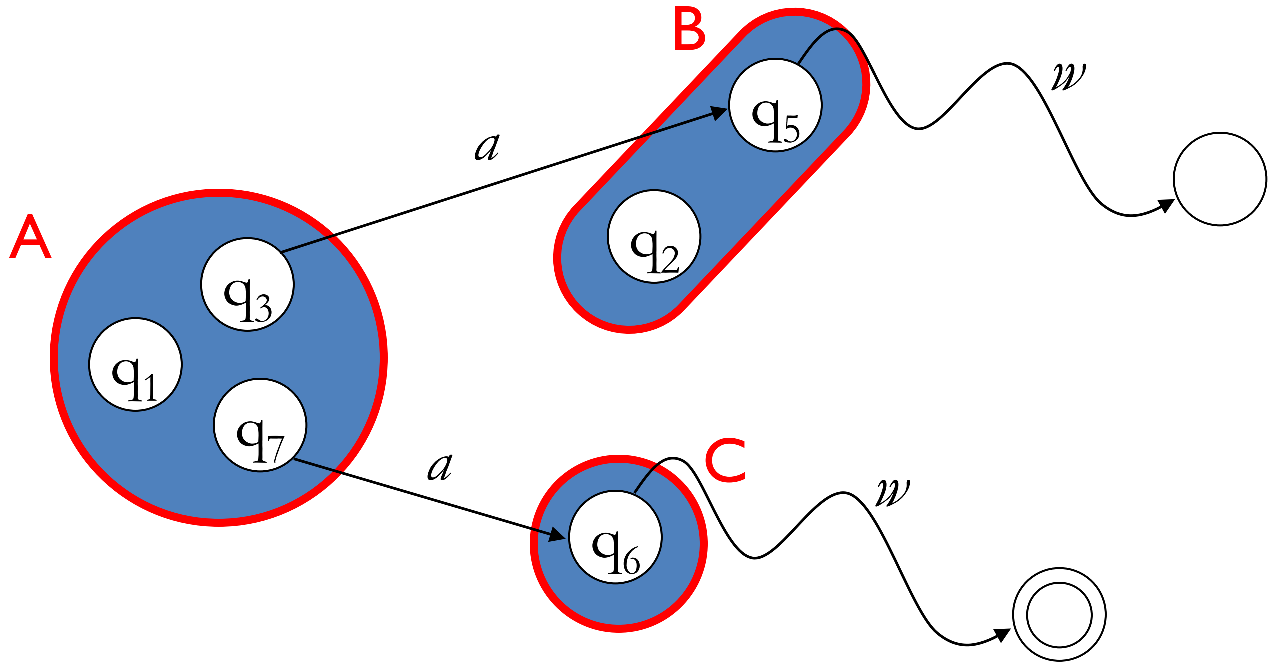
- Suppose there were two inconsistent transitions in M' labeled a out of merged state q_A

Proof of consistency



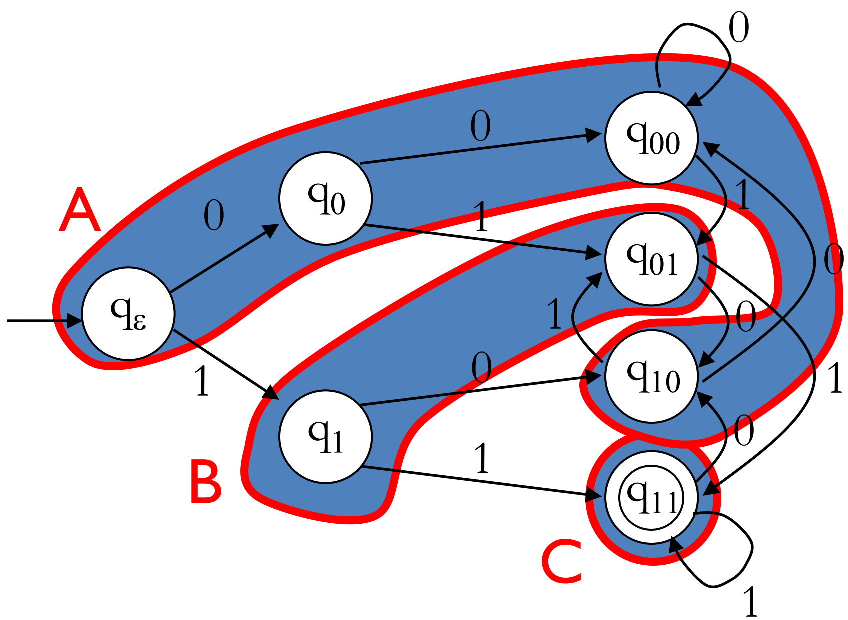
- States q_5 and q_6 must be distinguishable because they were not merged together

Proof of consistency



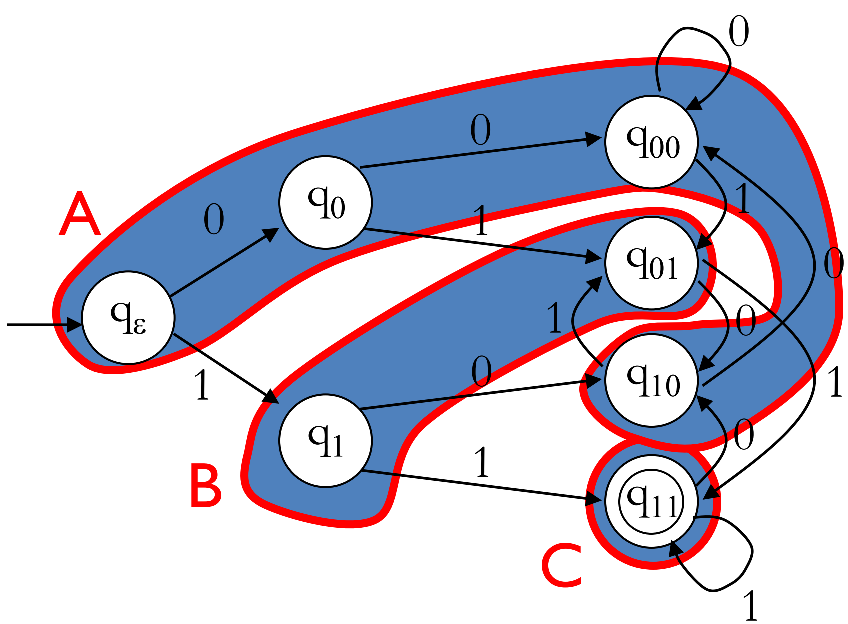
- Then, q_3 and q_7 must also be distinguishable, and this is impossible

Consistency of states



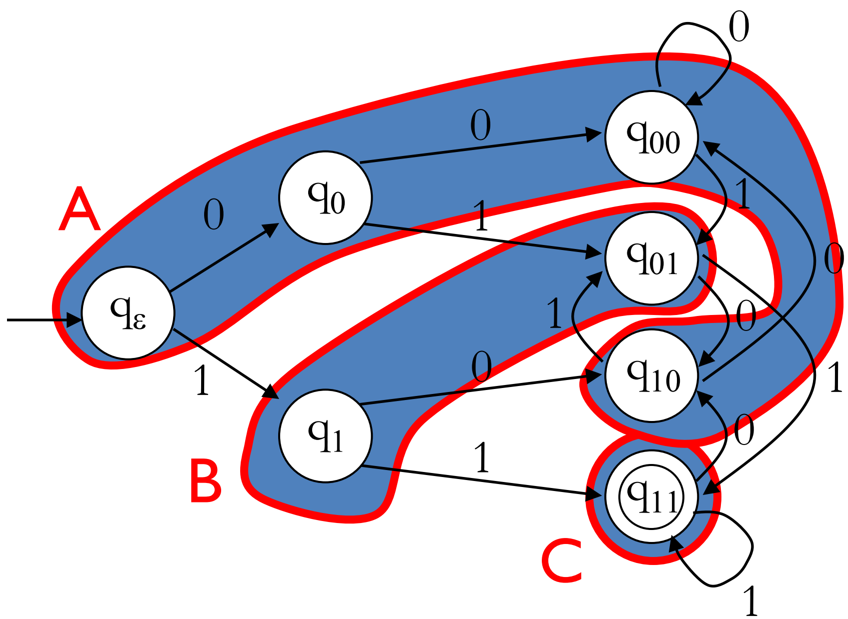
- Why are the merged states either all accepting or all rejecting?

Consistency of states



- Because merged states are not distinguishable, so they are mutually equivalent

Correctness



- Each state of M’ corresponds to a class of mutually equivalent states in M

Proof of correctness

- Claim: After reading input w,

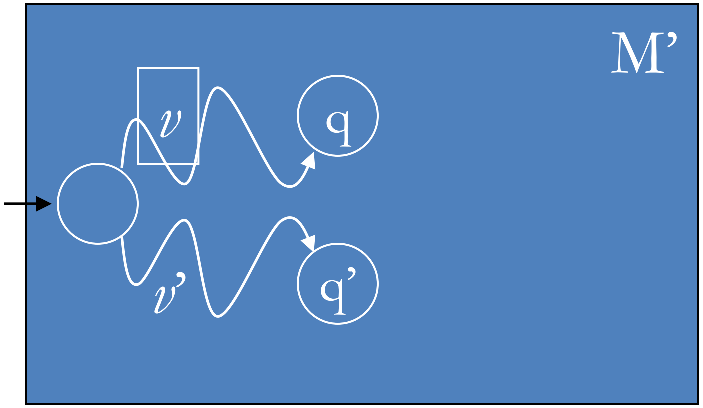
> M' is in state q_A if and only if M is in one of the states represented by A

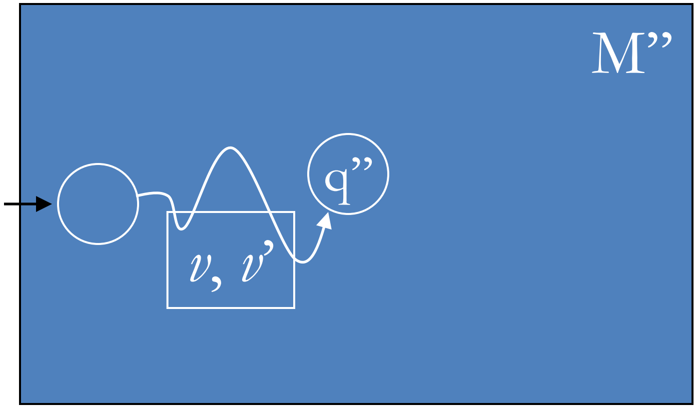
- Proof: True in start state, and stays true after, because transitions are consistent
- At the end, M accepts w in state q_i if and only if M' on input w lands in the state q_A that represents q_i
  - q_A must be accepting by consistency of states

Proof of minimality

All pairs of states in M' are distinguishable

- Now suppose there is some smaller M'' for L
- By the pigeonhole principle, there is a state q'' of M'' such that





Proof of minimality

All pairs of states in M' are distinguishable

- Since q and q' are distinguishable, for some w…
- But in M'', $\delta(q'', w)$ cannot both accept and reject!
- So, M'' cannot be smaller than M'