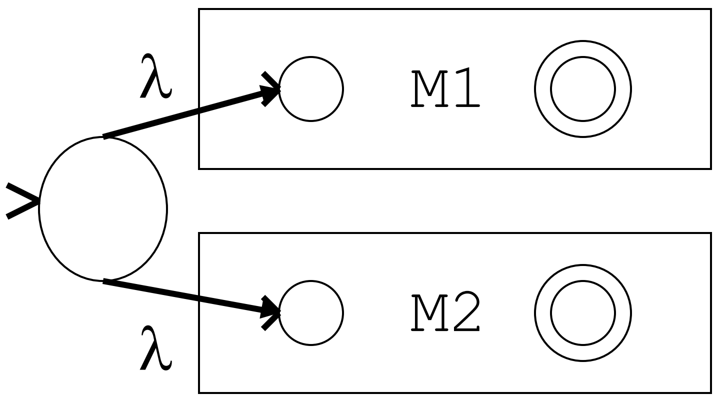
NFA vs. DFA

NFAs vs. DFAs

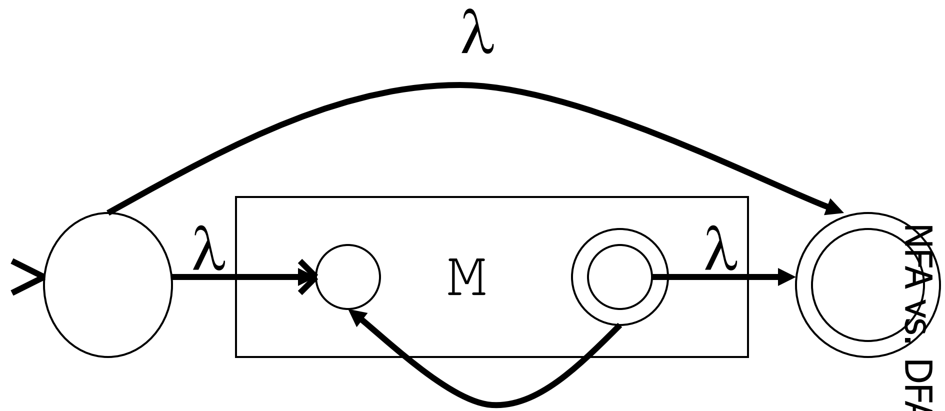
- NFAs can be constructed from DFAs using transitions:
  - Called NFA-λ

    $\in$
  - Suppose M_1 accepts L_1, M_2 accepts L_2
    - Then an NFA can be constructed that accepts:
      - L_1 U L_2 (union)
      - L_1L_2 (concatenation)
      - L_1^* (Kleene star)

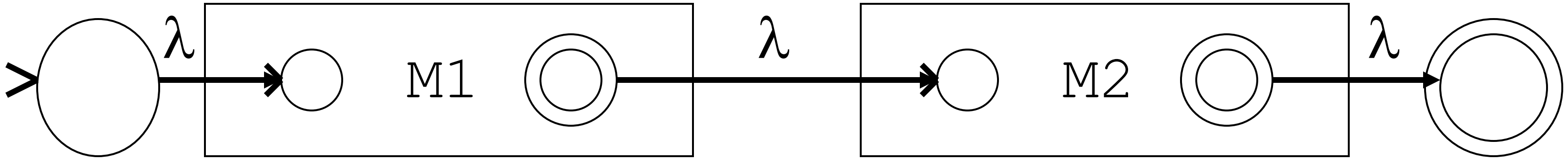
Closure Properties of NFA-Λs



L(M1) ∪ L(M2)



L(M)*



L(M1) L(M2)

NFA to DFA Conversion

DFA vs NFA

- Deterministic vs nondeterministic
  - For every nondeterministic automata, there is an equivalent deterministic automata
  - Finite acceptors are equivalent iff they both accept the same language

$$L(M_1) = L(M_2)$$

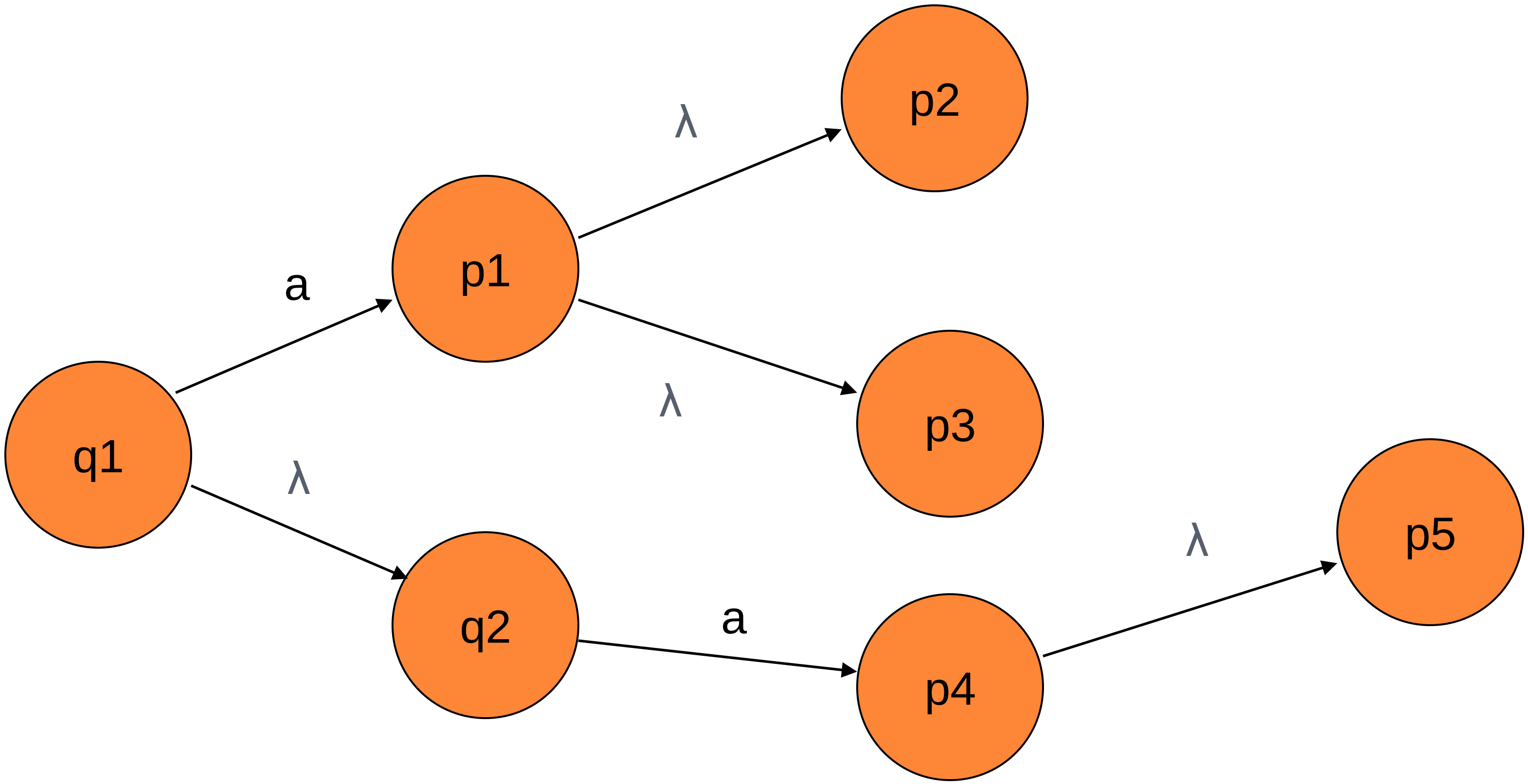
DFA VS NFA

- Deterministic vs nondeterministic
  - In DFA, label resultant state as a set of states
    - {q1, q2, q3,…}
  - For a set of |Q| states, there are exactly 2^Q subsets
    - Finite number of states

REMOVING NONDETERMINISM

- By simulating all moves of an NFA-λ *in parallel* using a DFA.
- *λ-closure* of a state is the set of states reachable using only the λ-transitions.

NFA-Λ



$$t(q1, a) = \{p1, p2, p3, p4, p5\}$$

Λ – Closure

- Selected λ closures

q_1: $\{q_1, q_2\}$

p_1: $\{p_1, p_2, p_3\}$

q_2: $\{q_2\}$

EQUIVALENCE CONSTRUCTION

- Given an NFA-λ M_1, construct a DFA M_2 such that $\mathcal{L}(M) = \mathcal{L}(DM)$.
- Observe that
  - A node of the DFA = Set of nodes of NFA-λ
  - Transition of the DFA =

    Transition among set of nodes of NFA- λ

Special States to Identify

Start state of DFA =

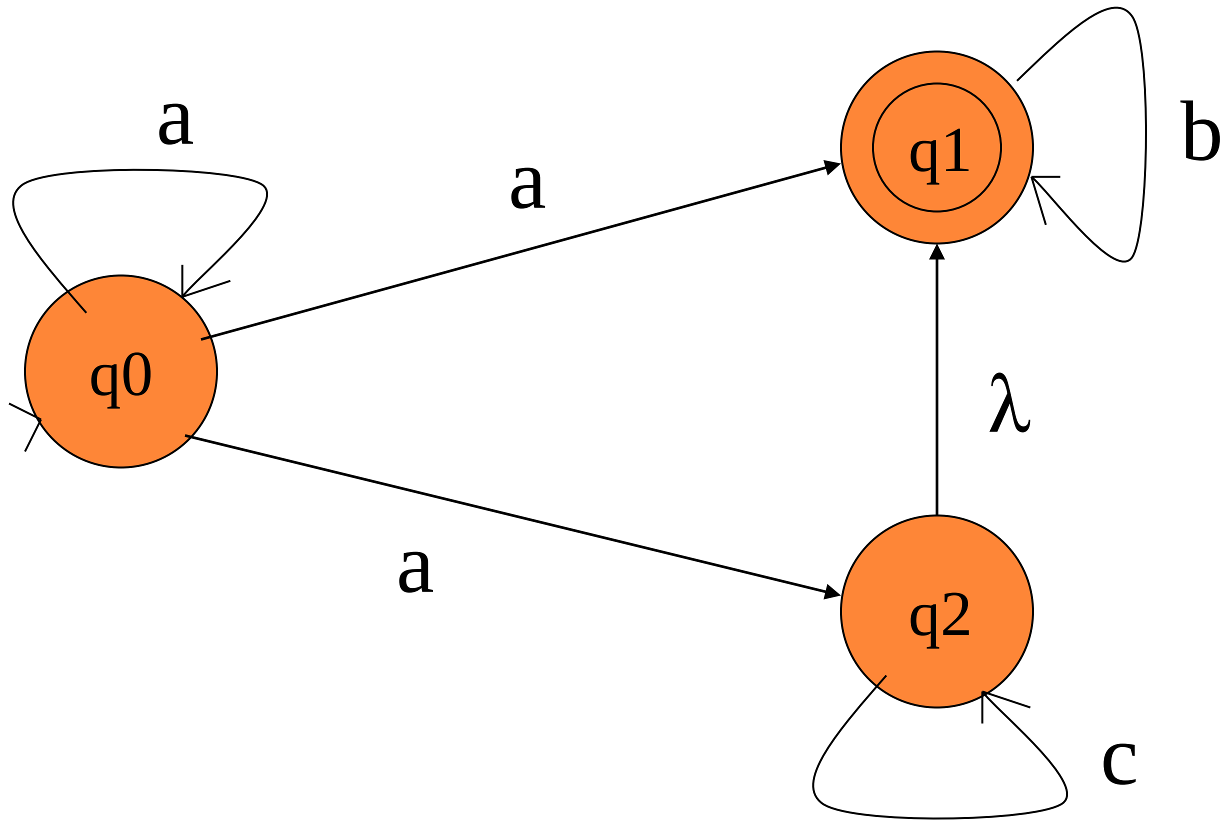
$$\lambda\text{-}closure(\{q_0\})$$

Final/Accepting state of DFA =

All subsets of states of NFA-λ that contain an accepting state of the NFA-λ

Dead state of DFA = ϕ

EXAMPLE



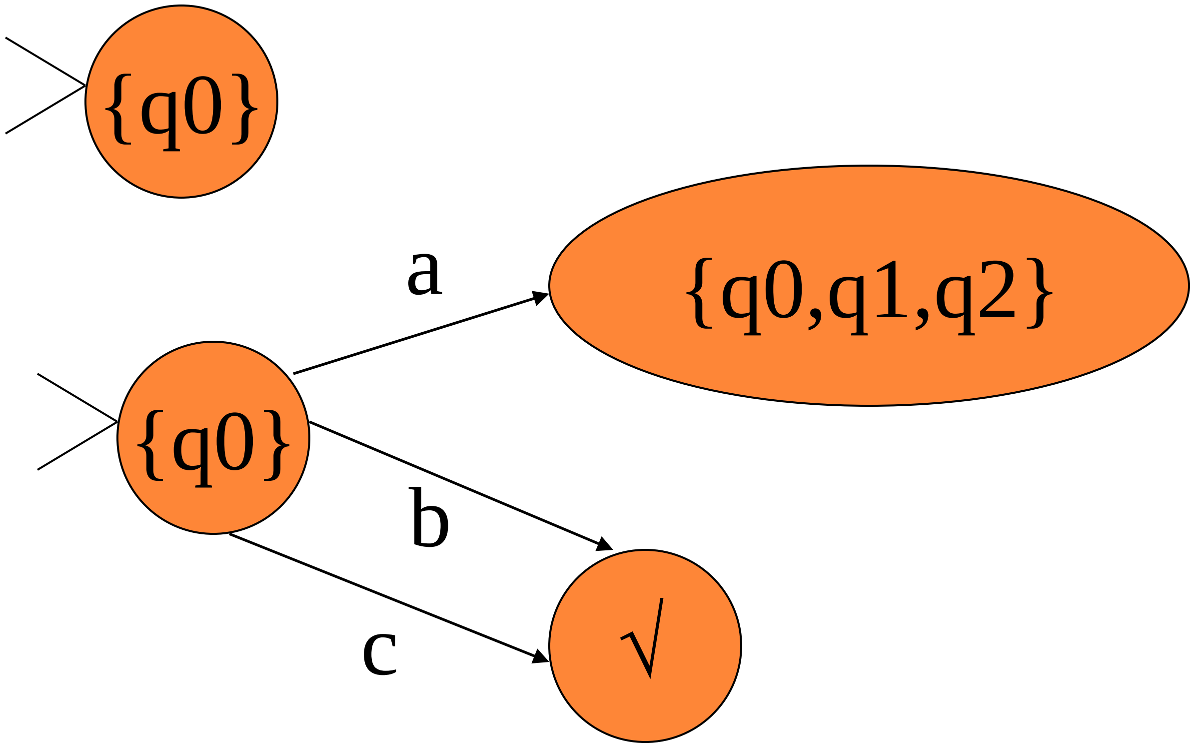
EXAMPLE

- Identify λ-closures
  - q_0: $\{q_0\}$
  - q_1: $\{q_1\}$
  - q_2: $\{q_{1,}q_2\}$

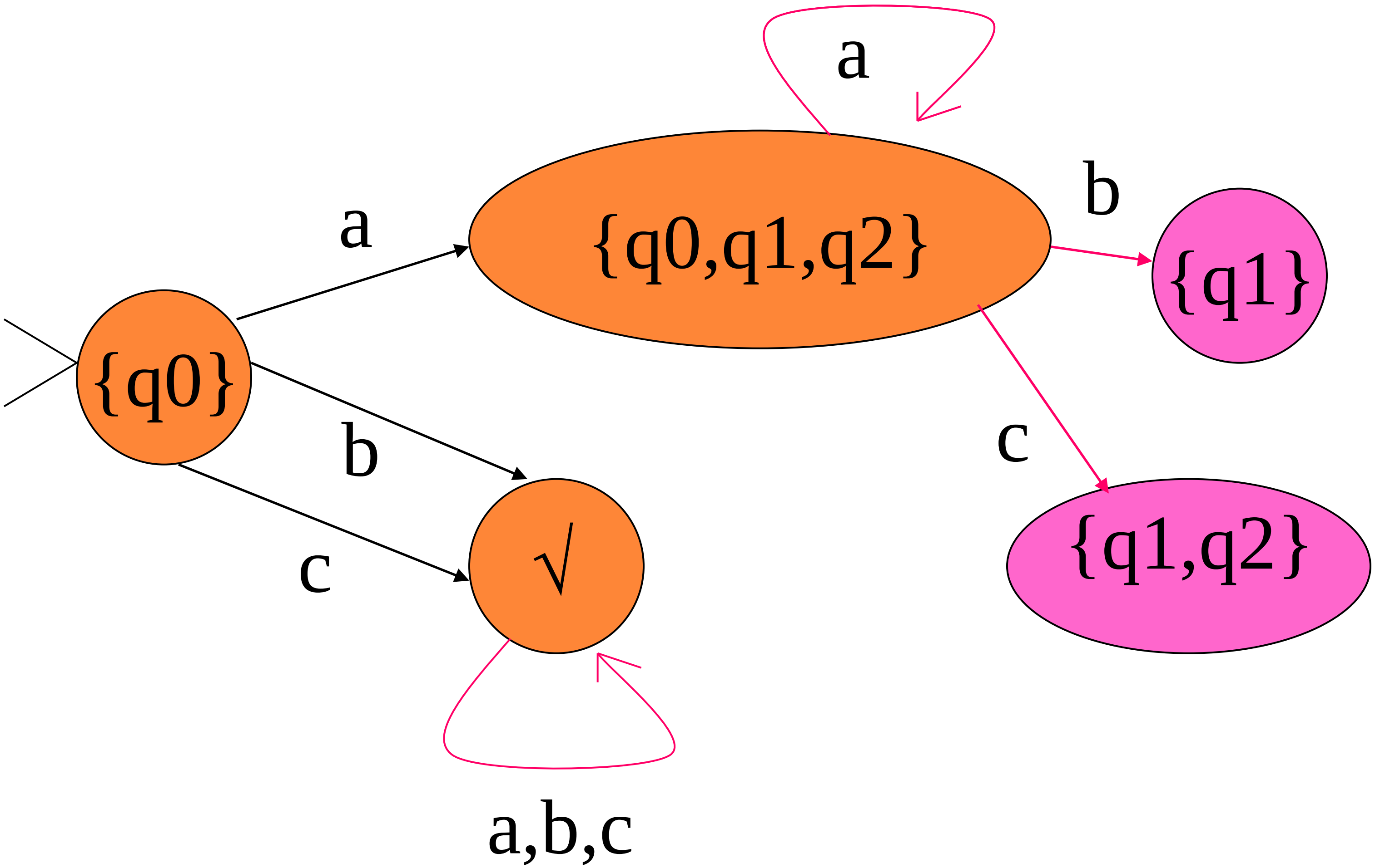
EXAMPLE

- Identify transitions
  - Start with λ-closure of start state
  - $\{q_0\}$: Where can you go on each input?
    - a: $\{q_0,q_1,q_2\}$
      - So, $\{q_0,q_1,q_2\}$ is a state in the DFA
    - b, c: Nowhere, so {Φ} is in the DFA
    - Next slide…
  - Next, do the same for $\{q_0,q_1,q_2\}$ and {Φ}
    - Find destinations from any node in the set for each of the three alphabet symbols
    - Subsequent slide…

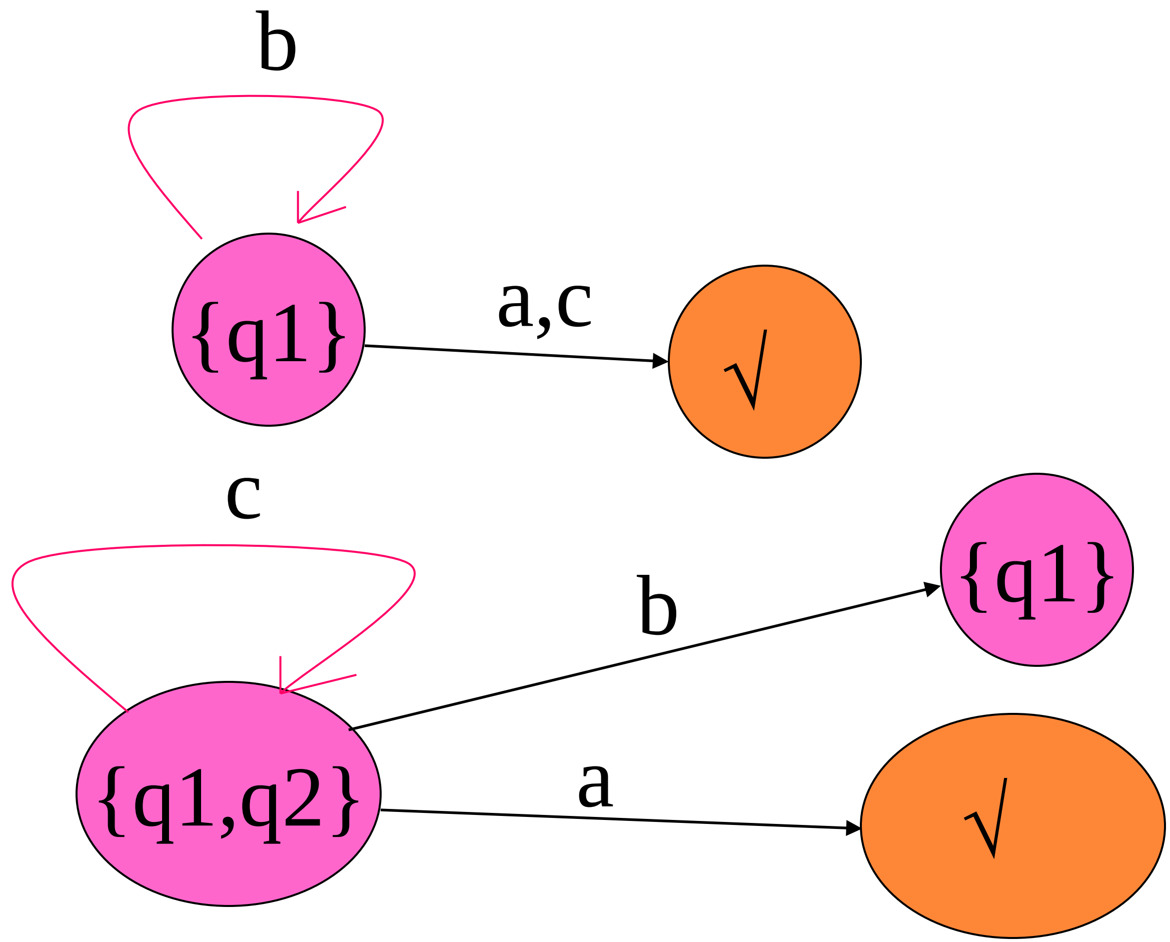
All steps from $\{q_0\}$



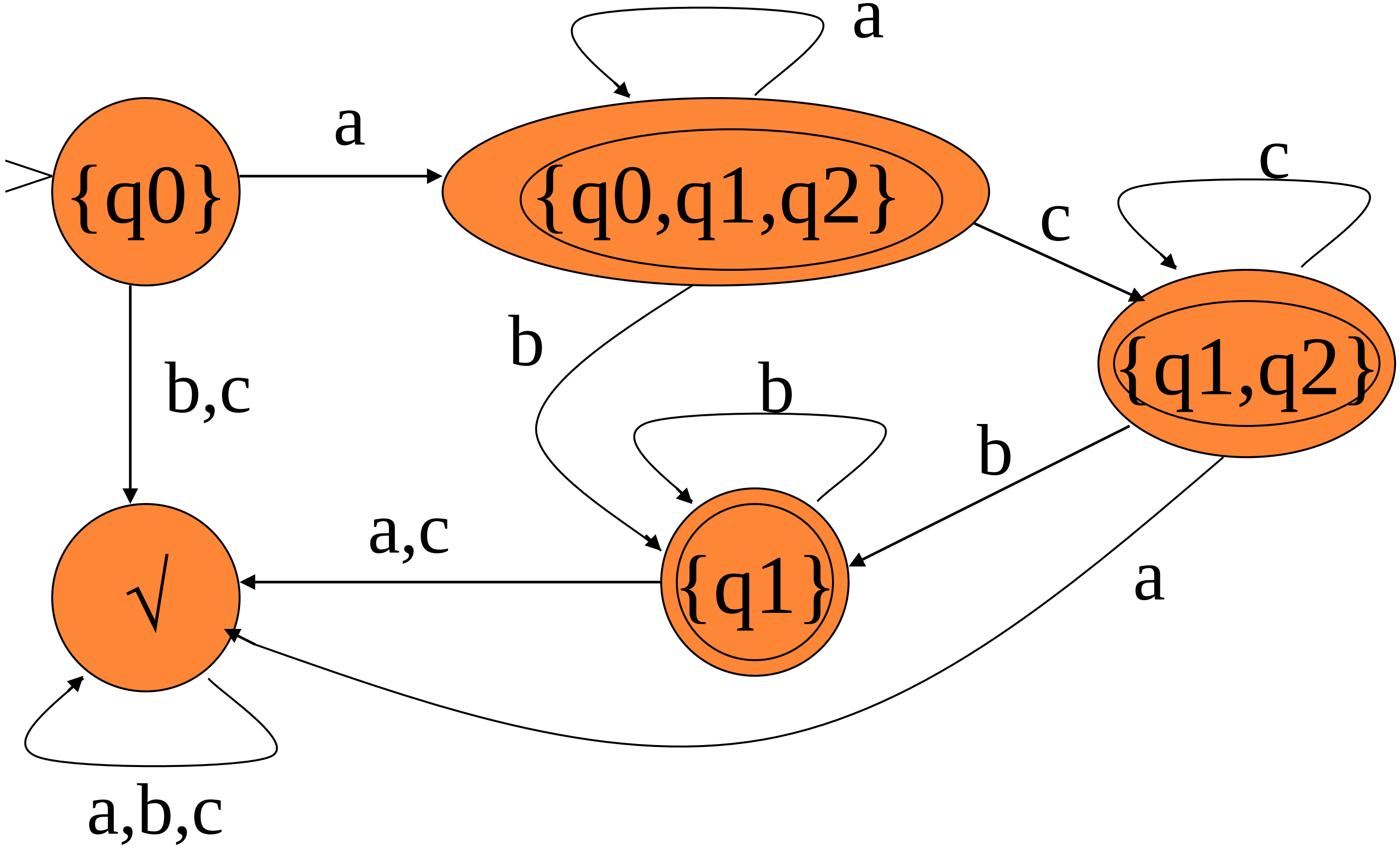
All steps from $\{q_0,q_1,q_2\}$



All steps from $\{q_1\}$ and $\{q_1, q_2\}$



Equivalent DFA



NFA VS. DFA

Theorem: Given any NFA N, then there exists a DFA D such that N is equivalent to D

- Proven by constructing a general NFA and showing that the closure exists among the possible DFA states P(Q)
  - Every possible transition goes to an element of P(Q)

LIMITATIONS OF FINITE AUTOMATA

- Obvious: Can only accept languages that can be represented in finite memory!
- Can this language be represented with a FA?
  - L(M)=($a^i b^i$ | i $\leq$ n)
- How about this one?
  - L(M)=($a^i b^i$ | i > 0)

EXERCISE: CONVERT THIS NFA