Computing Theory

COMP 147 (4 units)

Chapter 1: Regular Languages

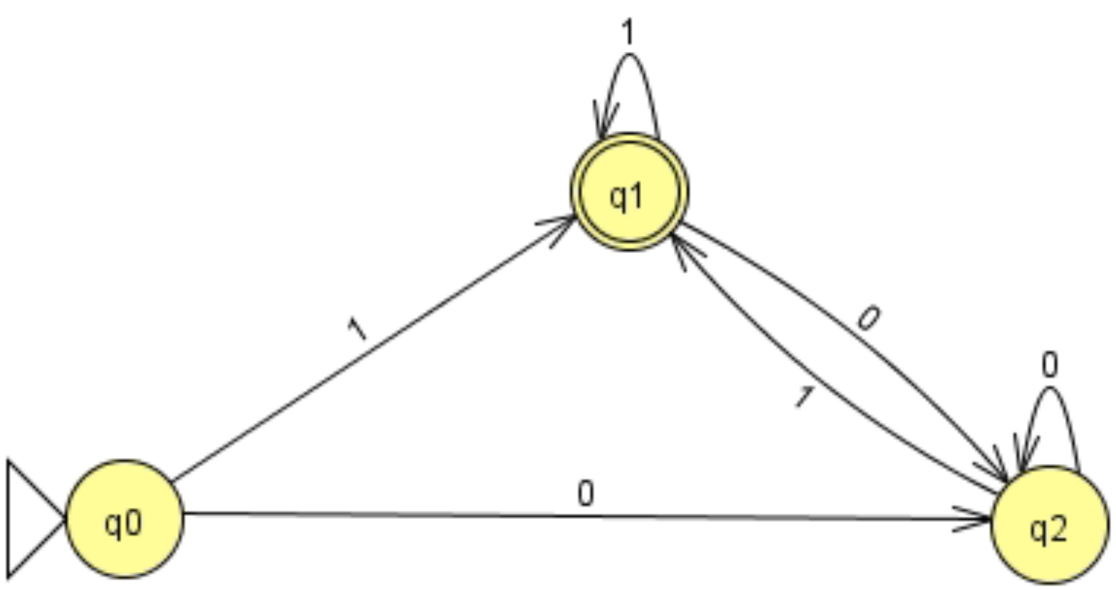
Closure Properties of Regular Languages

Last Time

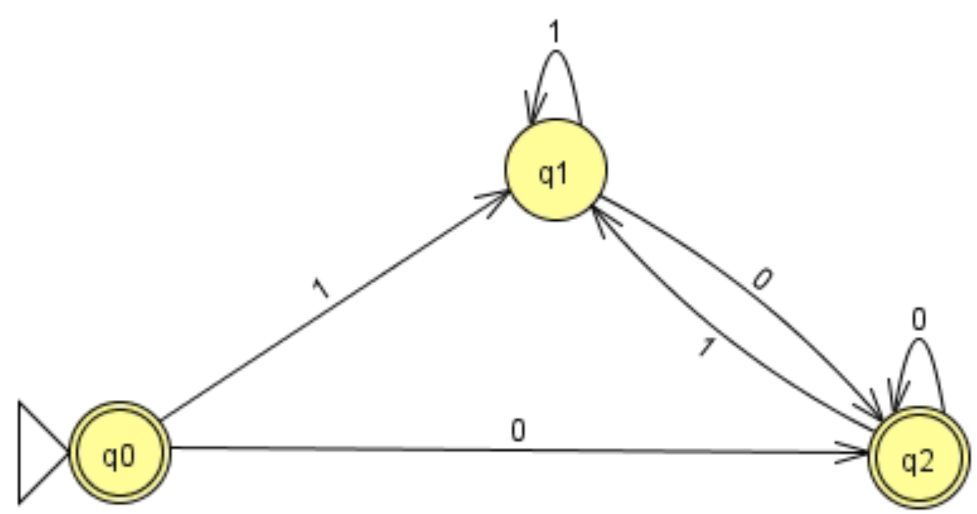
- NFAs
  - Nondeterminism
  - NFA and DFA equivalence
  - Convert NFA to DFA
- Regular Languages

Complement of the Language Recognized by a DFA

- Given a machine that recognizes L,
  we can construct a machine that recognizes
  the complement of L, $\overline{L}$, by reversing the accept
  states and non-accept states **Important: does not work for NFA!**



Recognizes L



Recognizes $\overline{L}$

Complement DFA

Let $M = (Q,\Sigma,\delta,q_0,F)$ be a DFA that recognizes $L(M)$.

Give a formal definition of a DFA M' that recognizes $\overline{L(M)}$.

$$M' = (Q,\Sigma,\delta,q_0,F'), \text{ where } F' = Q - F.$$

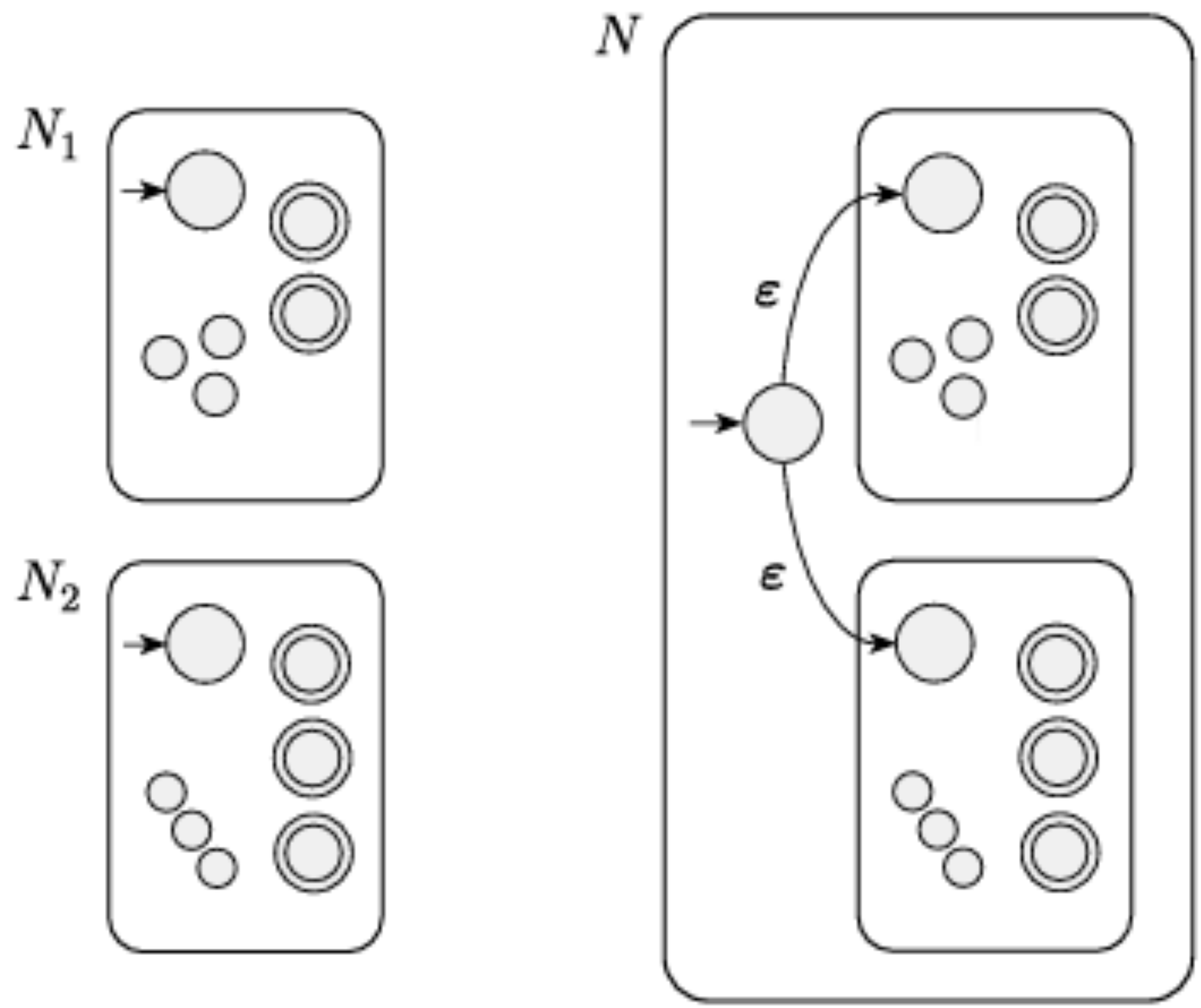
Closure under Complement Operation

Closure Under Union

- L1 = {w | w has odd number of 0’s}
- L2 = {w | w contains substring 01}
- L1 and L2 are regular languages (We constructed FA’s for them)
- is L1 ∪ L2 regular?

Closure Under Union

- Theorem: Regular languages are closed under union.
    - Proof by NFA construction.



Closure Under Union

PROOF

Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognize A_1, and
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognize A_2.

Construct $N = (Q, \Sigma, \delta, q_0, F)$ to recognize $A_1 \cup A_2$.

1. $Q = \{q_0\} \cup Q_1 \cup Q_2$.
   The states of N are all the states of N_1 and N_2, with the addition of a new start state q_0.
2. The state q_0 is the start state of N.
3. The set of accept states $F = F_1 \cup F_2$.
   The accept states of N are all the accept states of N_1 and N_2. That way, N accepts if either N_1 accepts or N_2 accepts.
4. Define δ so that for any $q \in Q$ and any $a \in \Sigma_\varepsilon$,

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \\ \delta_2(q,a) & q \in Q_2 \\ \{q_1, q_2\} & q = q_0 \text{ and } a = \boldsymbol{\varepsilon} \\ \emptyset & q = q_0 \text{ and } a \neq \boldsymbol{\varepsilon}. \end{cases}$$

Closure Under Intersection

- L1 = {w | w has even number of 0's}
- L2 = {w | w contains substring 01}
- L1 and L2 are regular languages (We constructed FA's for them)
- is L1 ∩ L2 regular?

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

beware of complementing an NFA

Less Fussy: Product of Machines

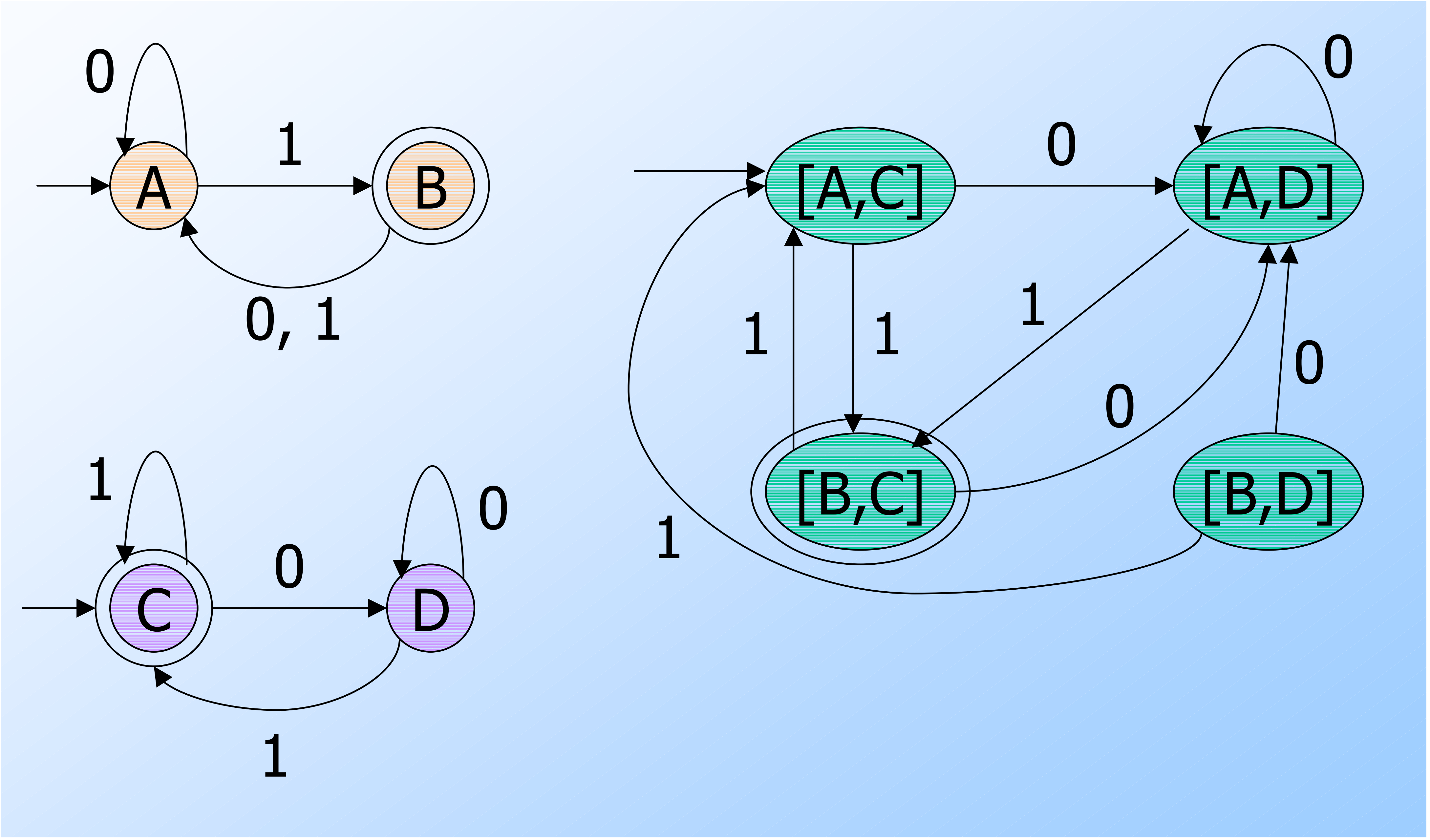
Closure Under Intersection

- is L1 ∩ L2 regular?
- Solution1: $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$

  beware of complementing an NFA

- Solution2
  - Product of Machines (next slide)

Closure Under Intersection
Product construction of DFA



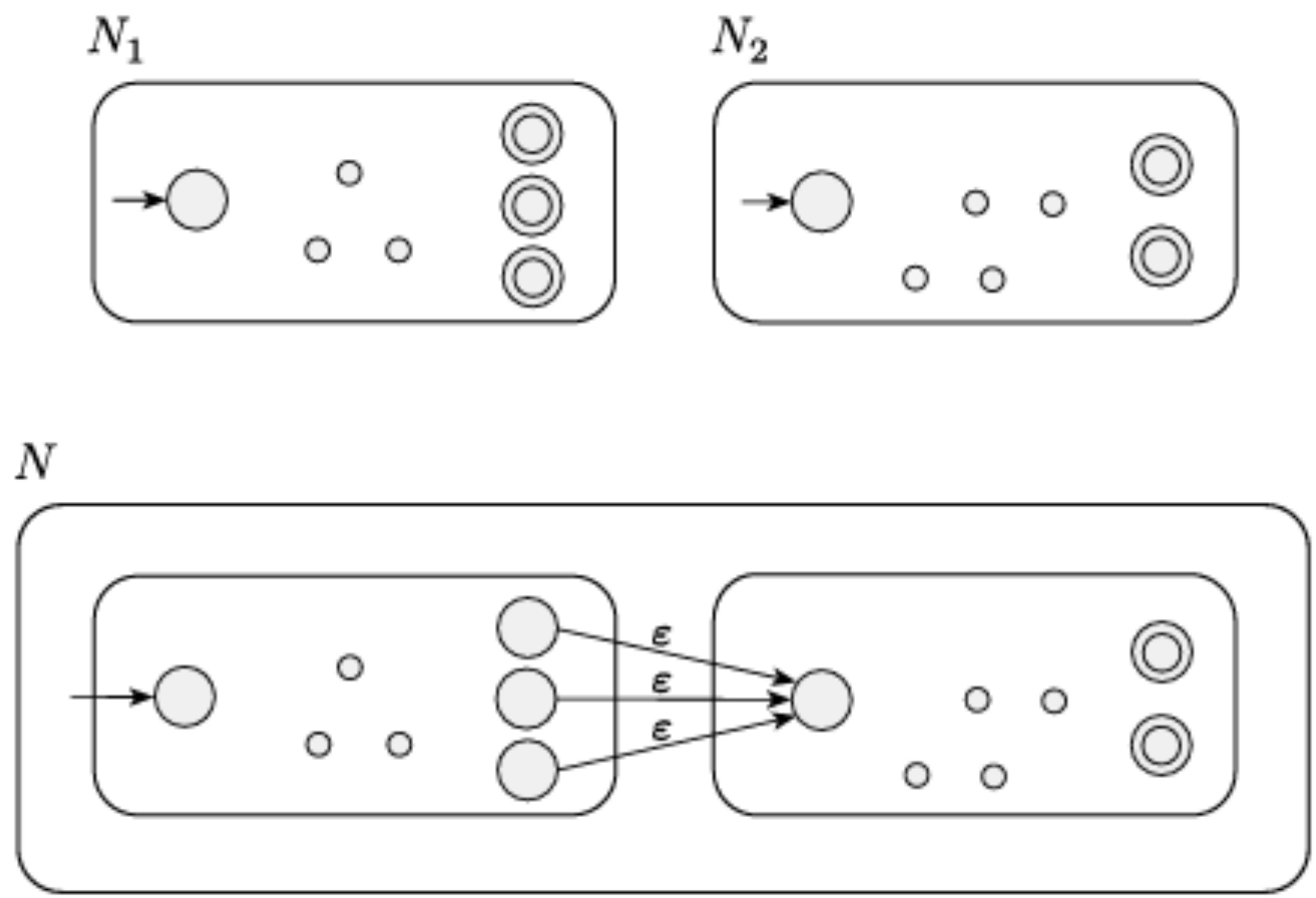
Can be used for other set operations

Closure Under Concatenation

- L1 = {w | w has even number of 1's}
- L2 = {w | w contains substring 101}
- L1 and L2 are regular languages (We constructed FA's for them)
- is L1 $\circ$ L2 regular?

Closure Under Concatenation

- Theorem: Regular languages are closed under concatenation.
  - Proof by NFA construction.



Closure Under Concatenation

PROOF

Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognize A_1, and
$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognize A_2.

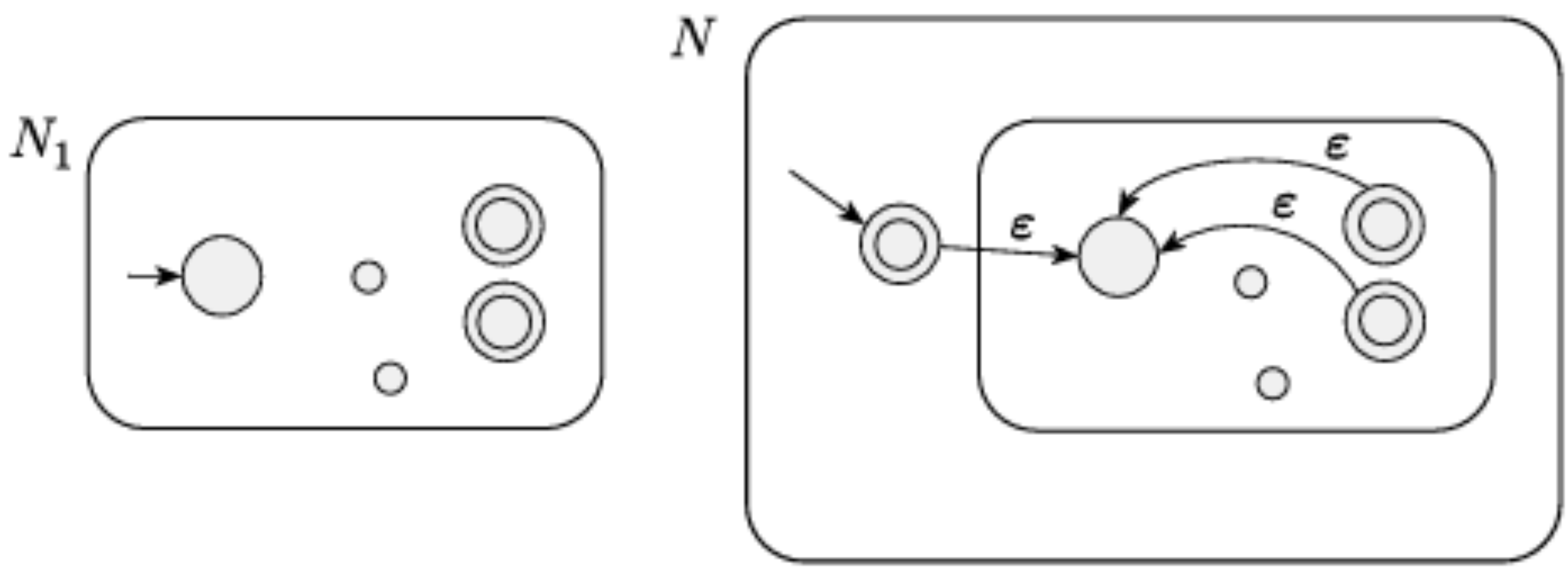
Construct $N = (Q, \Sigma, \delta, q_1, F_2)$ to recognize $A_1 \circ A_2$.

1. $Q = Q_1 \cup Q_2$.
   The states of N are all the states of N_1 and N_2.
2. The state q_1 is the same as the start state of N_1.
3. The accept states F_2 are the same as the accept states of N_2.
4. Define δ so that for any $q \in Q$ and any $a \in \Sigma_\varepsilon$,

$$\delta(q, a) = \begin{cases} \delta_1(q, a) & q \in Q_1 \text{ and } q \notin F_1 \\ \delta_1(q, a) & q \in F_1 \text{ and } a \neq \varepsilon \\ \delta_1(q, a) \cup \{q_2\} & q \in F_1 \text{ and } a = \varepsilon \\ \delta_2(q, a) & q \in Q_2. \end{cases}$$

Closure Under Star

- Theorem: Regular languages are closed under the star operation.
  - Proof by NFA construction.



Closure Properties

- Regular Languages are closed under a variety of operations
  - Union, Intersection, Complement
  - String Reversal
  - Set Difference
- Regular Languages are not closed under subset or superset