Theory of Computation
Chapter 1

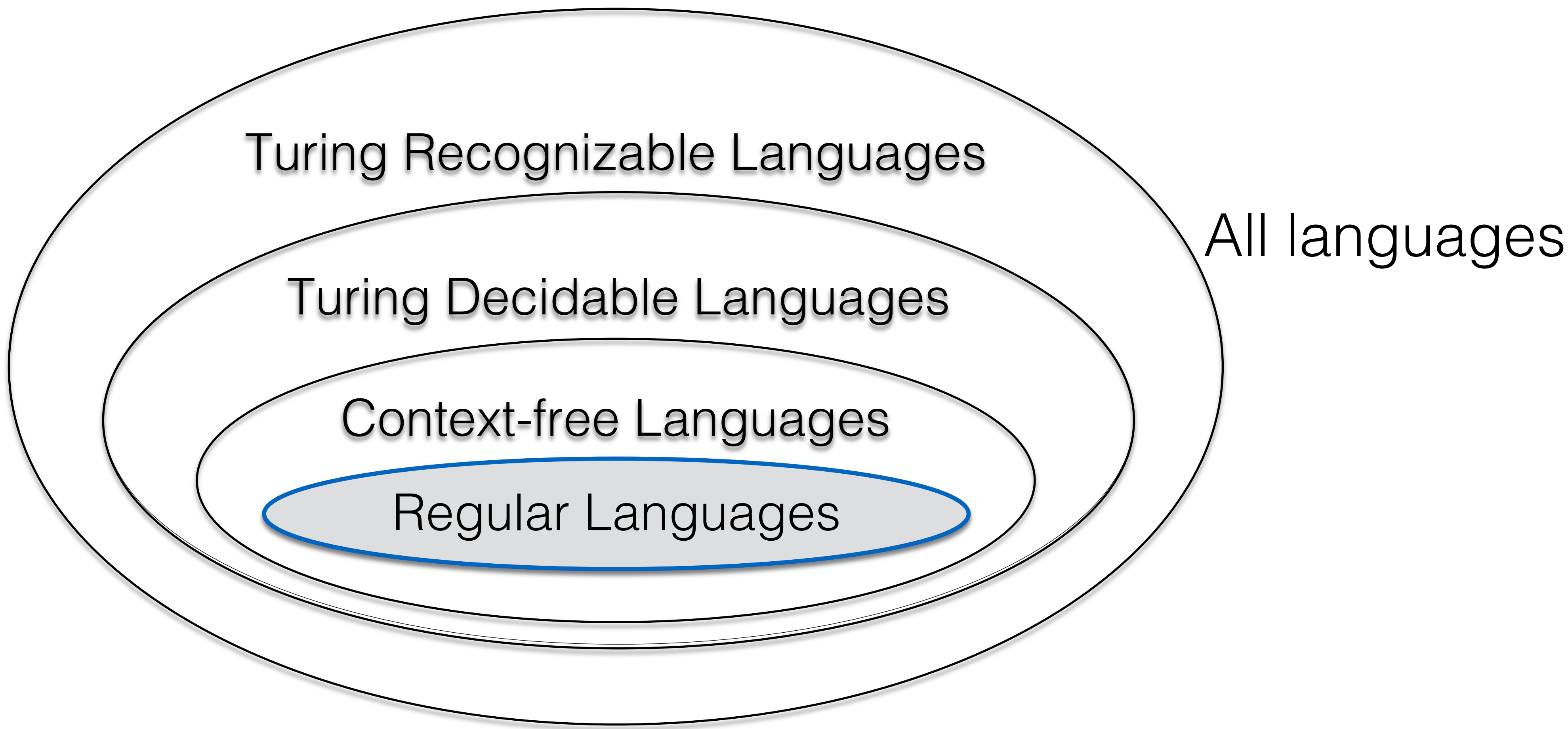
Regular Languages Part 2

School of Engineering | Computer Science

Gordon Moore 1929-2023

- The “Moore” of “Moore’s Law”
- Co-founder (1968) and longtime chairman and CEO of Intel Corp.
- Endowed Gordon and Betty Moore Foundation (with wife)
- One of original 8 founders (1957) of Fairchild Semiconductor, manufacturer of first cost-effective silicon integrated circuit

Regular Languages



Regular Languages

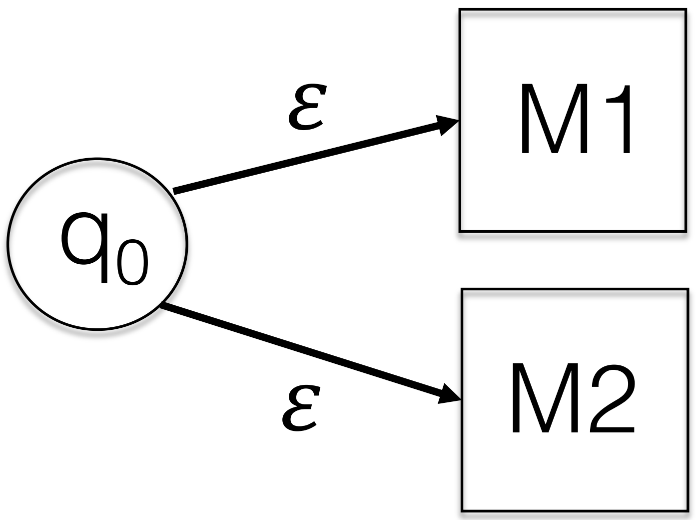
DFA = NFA = RE

Closed under union, ∪, concatenation, ∘, and star, $*$.

VCU School of Engineering | Computer Science

Closure Under Union

- Closure is much easier to show with NFA's verses DFA's
  - Union: $A \cup B = \{ x \mid x \in A \text{ or } x \in B\}$
    - M1 is a DFA that recognizes A and M2 is a DFA that recognizes B,
    - The union is an NFA, recognizing both of the languages.



Closure Under Union

- Union Formally Defined:
  - Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$
  - Construct $N_3 = (Q_3, \Sigma, \delta_3, q_3, F_3)$ as follows:

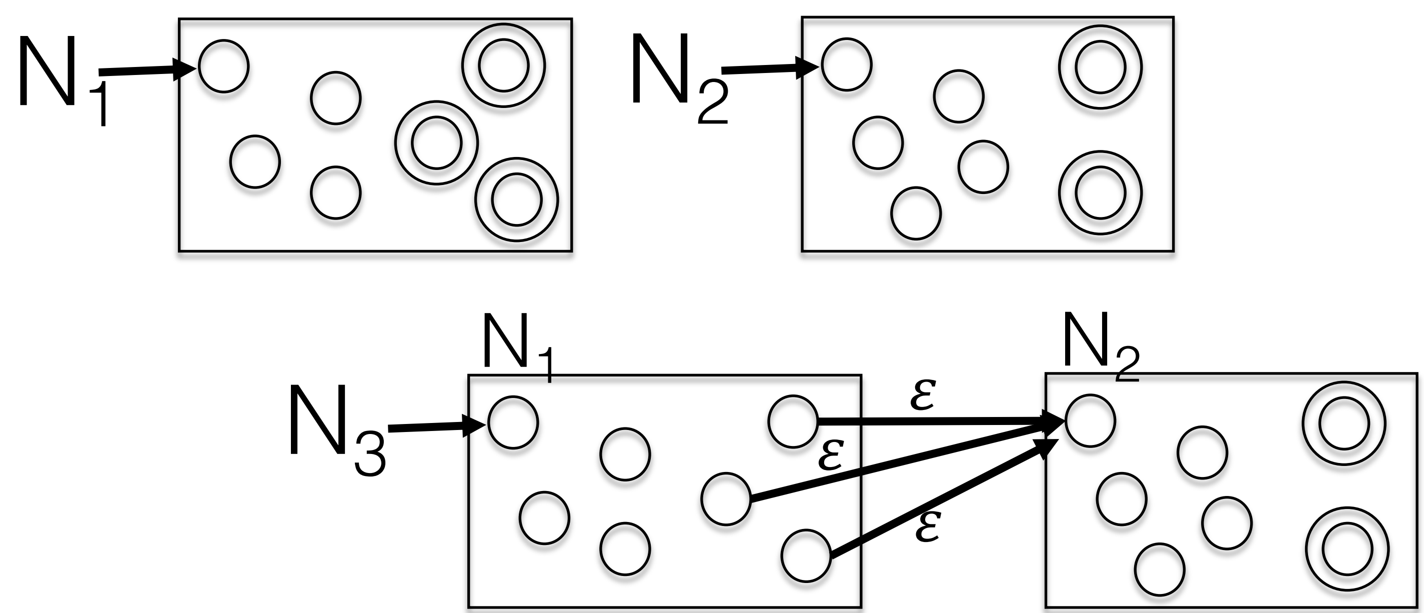
1. $Q_3 = Q_1 \cup Q_2 \cup \{q_0\}$
2. $q_3 = q_0$
3. $F_3 = F_1 \cup F_2$
4. δ_3 is defined such that for any $q \in Q$ and any $a \in \Sigma_\varepsilon$:

$$\delta_3(q, a) = \begin{cases} \delta_1(q, a) & q \in Q_1 \\ \delta_2(q, a) & q \in Q_2 \\ \{q_1, q_2\} & q = q_0 \text{ and } a = \varepsilon \\ \emptyset & q = q_0 \text{ and } a \neq \varepsilon \end{cases}$$

Closure Under Concatenation

- Theorem 1.47: Regular languages are closed under concatenation.
  - Let A and B be regular languages. Then $A \circ B$ is a regular language $A \circ B = \{xy \mid x \in A \text{ and } y \in B\}$
  - Proof Idea: Let N_1 and N_2 be NFAs for A and B respectively

N_1 no longer has its accept states. They instead have ε transitions to N_2's start state.



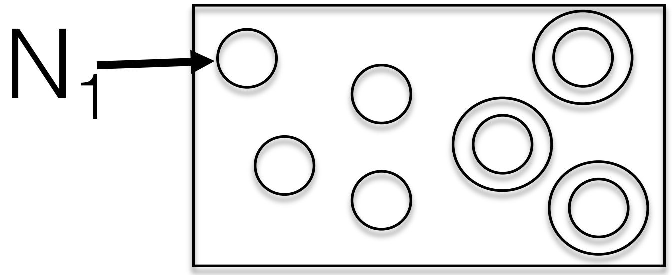
Closure Under Concatenation

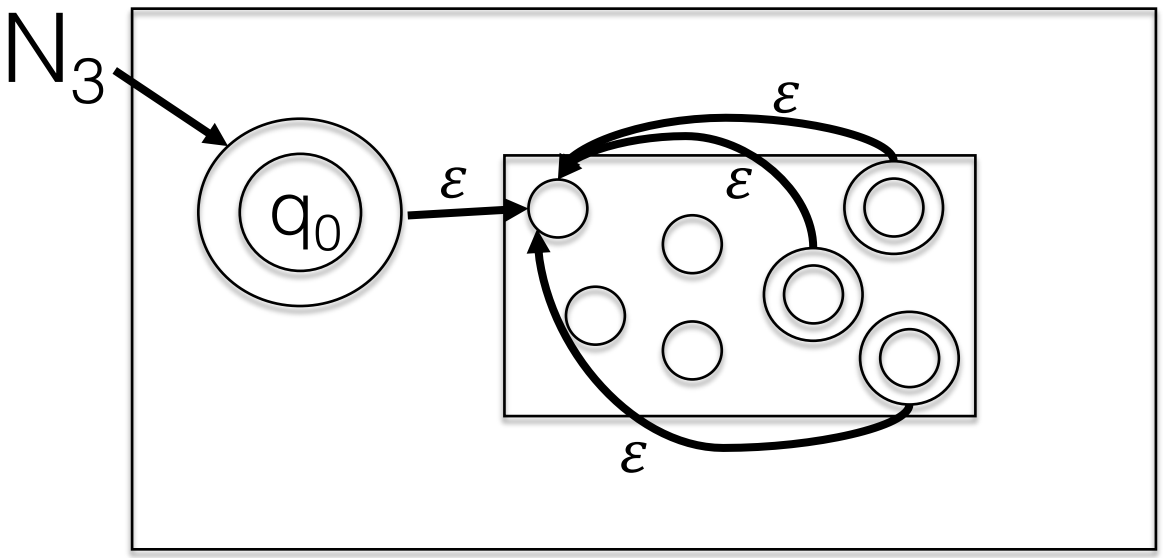
- <u>Theorem 1.47</u>: Regular languages are closed under concatenation.
  - Formally: Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$. Construct $N_3 = (Q_3, \Sigma, \delta_3, q_3, F_3)$ as follows:
    1. $Q_3 = Q_1 \cup Q_2$
    2. $q_3 = q_1$
    3. $F_3 = F_2$
    4. Define δ_3 such that for any $q \in Q_3$ and any $a \in \Sigma_\varepsilon$

$$\delta_3 \begin{cases} \delta_1(q, a) & q \in Q_1 \text{ and } q \notin F_1 \\ \delta_2(q, a) & q \in Q_2 \\ \delta_1(q, a) \cup \{q_2\} & q \in F_1 \text{ and } a = \varepsilon \\ \delta_1(q, a) & q \in F_1 \text{ and } a \neq \varepsilon \end{cases}$$

Closure Under Star

- Theorem 1.49: Regular languages are closed under star.
  - Let A be a regular language. Then A^* is a regular language
  - $A^* = \{x_1x_2\ldots x_k \mid k \geq 0 \text{ and } x_i \in A \text{ for all } 1 \leq i \leq k\}$
  - Proof Idea: Let N_1 be a NFA for A





There is a new start state that is an accept state. Each accept state transitions to the old start state.

Closure Under Star

- Theorem 1.49: Regular languages are closed under star.
  - Formally: Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$. Construct $N_3 = (Q_3, \Sigma, \delta_3, q_3, F_3)$ as follows:
    1. $Q_3 = Q_1 \cup \{q_0\}$
    2. $q_3 = q_0$
    3. $F_3 = F_1 \cup \{q_0\}$
    4. Define δ_3 such that for any $q \in Q_3$ and any $a \in \Sigma_\varepsilon$

$$\delta_3 \begin{cases} \delta_1(q, a) & q \in Q_1 \text{ and } q \notin F_1 \\ \delta_1(q, a) & q \in F_1 \text{ and } a \neq \varepsilon \\ \delta_1(q, a) \cup \{q_1\} & q \in F_1 \text{ and } a = \varepsilon \\ \{q_1\} & q = q_0 \text{ and } a = \varepsilon \\ \emptyset & q = q_0 \text{ and } a \neq \varepsilon \end{cases}$$

Regular Expressions

- We can use regular operations to build up expressions describing languages.
- They are a sequence of characters and are used to describe patterns in text
- Ex: list all words that start with the letters "pre"
  - Use "pre"Σ^* (where Σ is the English alphabet)

Regular Expressions

- R is a regular expression (RE) if R is one of the following:
  1. a for some $a \in \Sigma$
  2. ε
  3. $\emptyset$
  4. $R_1 \cup R_2$ for regular expressions R_1 and R_2
  5. $R_1 \circ R_2$ for regular expressions R_1 and R_2
  6. $R_1{}^*$ for regular expression R_1
- These are the possible cases of regular expressions. You can build the regular expressions for this course from just these 6 cases.

Regular Expressions

- Given the possible cases of regular expressions, the language of these regular expressions corresponds to:

1. $L(R) = \{a\}$
2. $L(R) = \{\varepsilon\}$
3. $L(R) = \emptyset$
4. $L(R_1 \cup R_2) = L(R_1) \cup L(R_2)$
5. $L(R_1 \circ R_2) = L(R_1) \circ L(R_2)$
6. $L(R_1{}^*) = (L(R_1))^*$

Regular Expressions

- Precedence (Order of Operations)
  1. Parentheses
  2. Star (exponent)
  3. Concatenation (multiplication)
  4. Union (addition)

- Convention: $R^+ = R^1R^*$ (At least one copy of R then * copies of R afterwards)

Regular Expressions

- Examples: Describe the language elaborated by the following regular expressions. Let $\Sigma = \{0, 1\}$

1. $0^*10^* =$
2. $\Sigma^*1\Sigma^* =$
3. $1^*(01^+)^* =$
4. $(\Sigma\Sigma\Sigma)^* =$
5. $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1 =$
6. $(0 \cup \varepsilon) \circ (1 \cup \varepsilon) =$

Regular Expressions

- Examples: : Describe the language elaborated by the following regular expressions. Let $\Sigma = \{0, 1\}$

1. $0^*10^* = \{w \mid w \text{ contains a single } 1\}$
2. $\Sigma^*1\Sigma^* = \{w \mid w \text{ contains at least one } 1\}$
3. $1^*(01^+)^* = \{w \mid \text{every } 0 \text{ in } w \text{ is followed by at least one } 1\}$
4. $(\Sigma\Sigma\Sigma)^* = \{w \mid \text{length of } w \text{ is a multiple of } 3\}$
5. $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1 = \{w \mid w \text{ starts and ends with the same symbol, with length at least } 1\}$
6. $(0 \cup \varepsilon) \circ (1 \cup \varepsilon) = \{\varepsilon, 0, 1, 01\}$

Regular Expressions

- Examples: Build a regular expression for the following. Let $\Sigma = \{0, 1\}$
  1. {w | w ends with a 1}
  2. {w | w contains 1001}
  3. {w | every 1 in w is followed by at least one 0}
  4. {w | length of w is a multiple of 2}
  5. {w | w has length at least 3 and starts and ends with a different symbol}
  6. {w | w contains at least one 0 and exactly two 1s}

Regular Expressions

- Examples: Build a regular expression for the following. Let $\Sigma = \{0, 1\}$
  1. $\{w \mid w$ ends with a 1$\}$ Σ^*1
  2. $\{w \mid w$ contains 1001$\}$ $\Sigma^*1001\Sigma^*$
  3. $\{w \mid$ every 1 in w is followed by at least 1 0$\}$ $0^*(10^+)^*$
  4. $\{w \mid$ length of w is a multiple of 2$\}$ $(\Sigma\Sigma)^*$
  5. $\{w \mid w$ has length at least 3 and starts and ends with a different symbol$\}$ $0\Sigma^+1 \cup 1\Sigma^+0$
  6. $\{w \mid w$ contains at least one 0 and exactly two 1s$\}$
     $0^*10^+10^* \cup 0^*10^*10^+ \cup 0^+10^*10^*$

VCU School of Engineering | Computer Science

Regular Expressions

- <u>Boundary Cases</u>
  1. $1^*\emptyset =$
  2. $\emptyset^* =$
  3. $R \cup \emptyset =$
  4. $R \cup \varepsilon =$
  5. $R \circ \varepsilon =$
  6. $R \circ \emptyset =$

Regular Expressions

- Boundary Cases
    1. $1^*\emptyset = \emptyset$ ($1^* \circ \emptyset$ - any set concatenated with the empty set gives the empty set)
    2. $\emptyset^* = \{\varepsilon\}$ (can only generate 0 strings from an empty language)
    3. $R \cup \emptyset = R$ (adding the empty set does not change R)
    4. $R \cup \varepsilon \neq R$ (cannot assume ε is in R)
    5. $R \circ \varepsilon = R$ (joining the empty string to any string will not change the string)
    6. $R \circ \emptyset = \emptyset$ (ex: $R = 0$, $L(R) = \{0\}$, $L(R \circ \emptyset) = \emptyset$, can also use case 1 above)

Regular Expressions

- Regular Languages are closed under
  - Union
  - Concatenation
  - Star
- Regular expressions are equivalent to deterministic finite automata which are equivalent to non-deterministic finite automata
  - RE = DFA = NFA

Try It

- Write a regular expression for the language of strings (with $\Sigma = \{0, 1\}$) for the following:
  - $\{w \mid w$ either ends in 01 or has a length divisible by 4 (or both)$\}$
  - $\{w \mid w$ contains exactly one 0s and at least three 1s$\}$

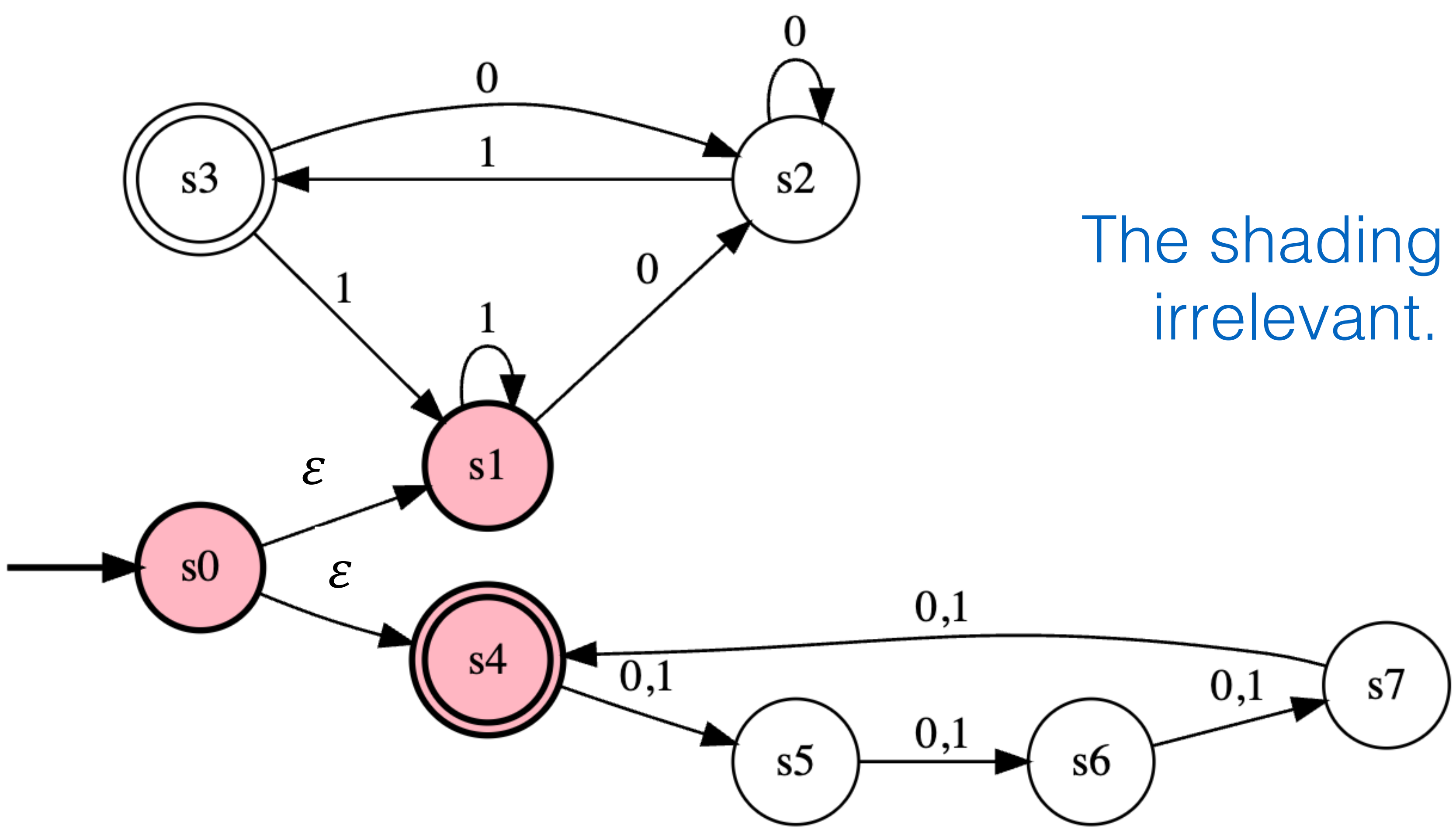
- Construct two separate DFAs (with $\Sigma = \{0, 1\}$) that accept (i) strings that end in 01 and (ii) strings with length divisible by 4. Combine them to construct an NFA that accepts the union of both.

Try It

- Write a regular expression for the language of strings (with $\Sigma = \{0, 1\}$) for the following:
  - {w | w either ends in 01 or has a length divisible by 4 (or both)} $\Sigma^*01 \cup (\Sigma\Sigma\Sigma\Sigma)^*$
  - {w | w contains exactly one 0s and at least three 1s} $1^+1^+1^+0 \cup 1^+1^+01^+ \cup 1^+01^+1^+ \cup 01^+1^+1^+$

Try It

- Construct two separate DFAs (with $\Sigma = \{0, 1\}$) that accept (i) strings that end in 01 and (ii) strings with length divisible by 4. Combine them to construct an NFA that accepts the union of both.



The shading is irrelevant.