Nondeterminism
CSCI 338

DFA vs NFA

Deterministic Finite Automaton (DFA):

Nondeterministic Finite Automaton (NFA):

DFA vs NFA

Deterministic Finite Automaton (DFA):

- Every state has exactly one transition for every character ($e \in \Sigma$).

Nondeterministic Finite Automaton (NFA):

DFA vs NFA

Deterministic Finite Automaton (DFA):

- Every state has exactly one transition for every character ($e \in \Sigma$).

Nondeterministic Finite Automaton (NFA):

- Allowed to have multiple (or 0) transitions for each $e \in \Sigma$.

DFA vs NFA

Deterministic Finite Automaton (DFA):

- Every state has exactly one transition for every character ($e \in \Sigma$).

Nondeterministic Finite Automaton (NFA):

- Allowed to have multiple (or 0) transitions for each $e \in \Sigma$.
- Allowed to have transitions that happen without input (ε-transition).

DFA vs NFA

Deterministic Finite Automaton (DFA):

- Every state has exactly one transition for every character ($e \in \Sigma$).
- For the same input, everyone takes the same path to the same final state.

Nondeterministic Finite Automaton (NFA):

- Allowed to have multiple (or 0) transitions for each $e \in \Sigma$.
- Allowed to have transitions that happen without input (ε-transition).

DFA vs NFA

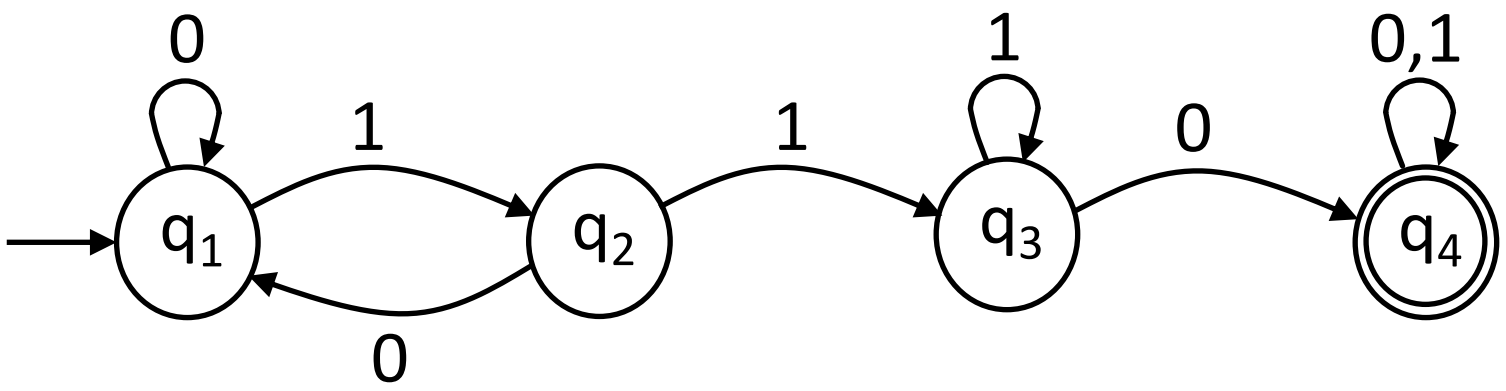
Deterministic Finite Automaton (DFA):

- Every state has exactly one transition for every character ($e \in \Sigma$).
- For the same input, everyone takes the same path to the same final state.

Nondeterministic Finite Automaton (NFA):

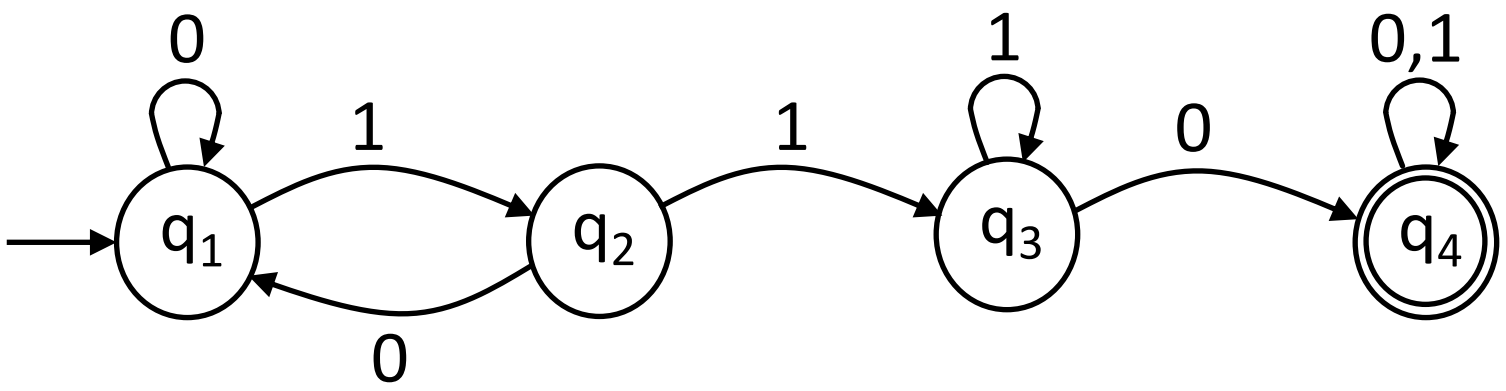
- Allowed to have multiple (or 0) transitions for each $e \in \Sigma$.
- Allowed to have transitions that happen without input (ε-transition).
- Processing strings is different.

DFA Processing



$\{\omega\colon\ \omega$ contains the substring $110\}$

DFA Processing

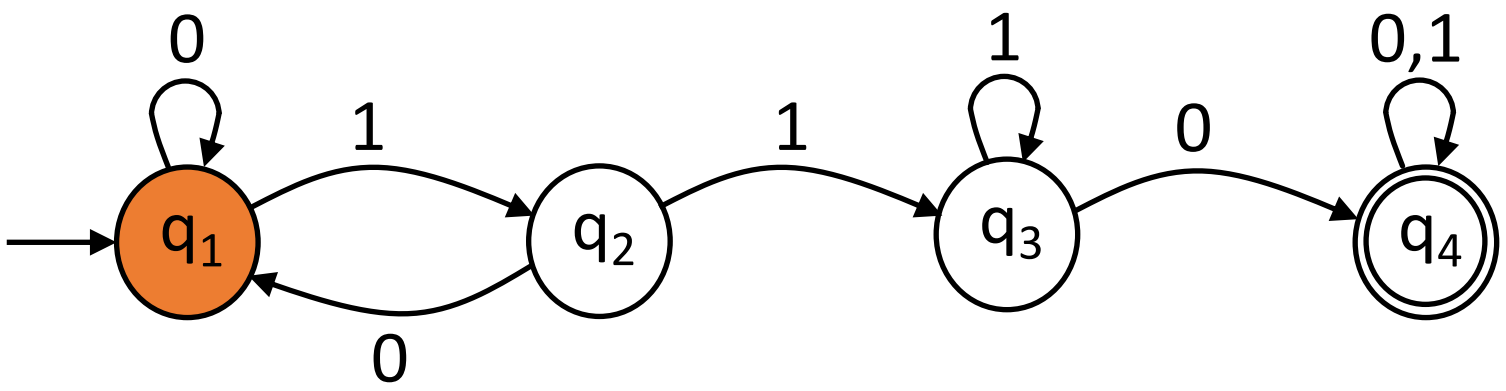


$\{\omega$: ω contains the substring $110\}$

$\omega = 1110$

DFA Processing

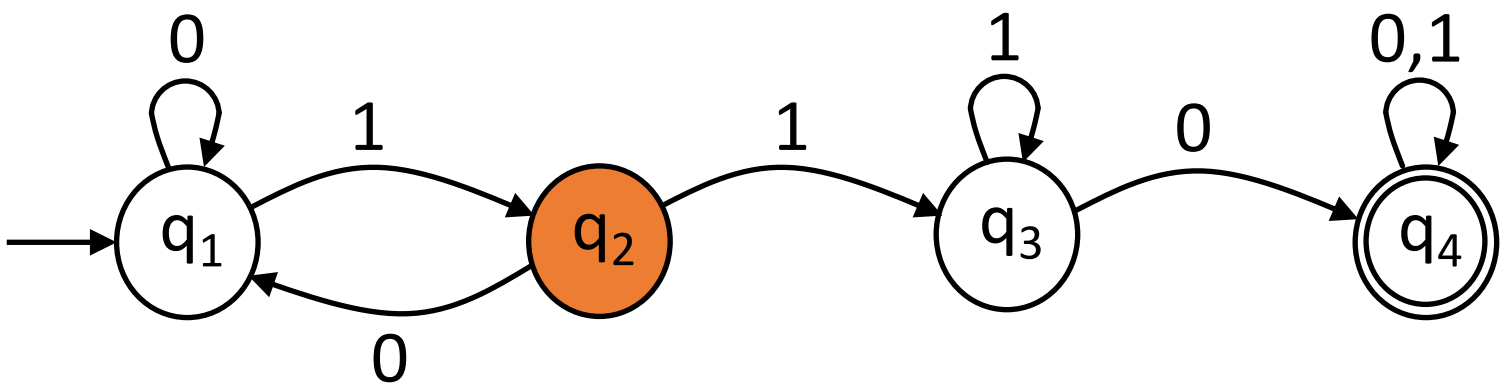
q_1



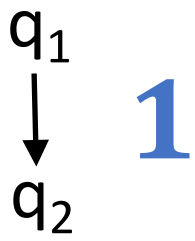
$\{\omega\colon\ \omega$ contains the substring $110\}$

$\omega = 1110$

DFA Processing

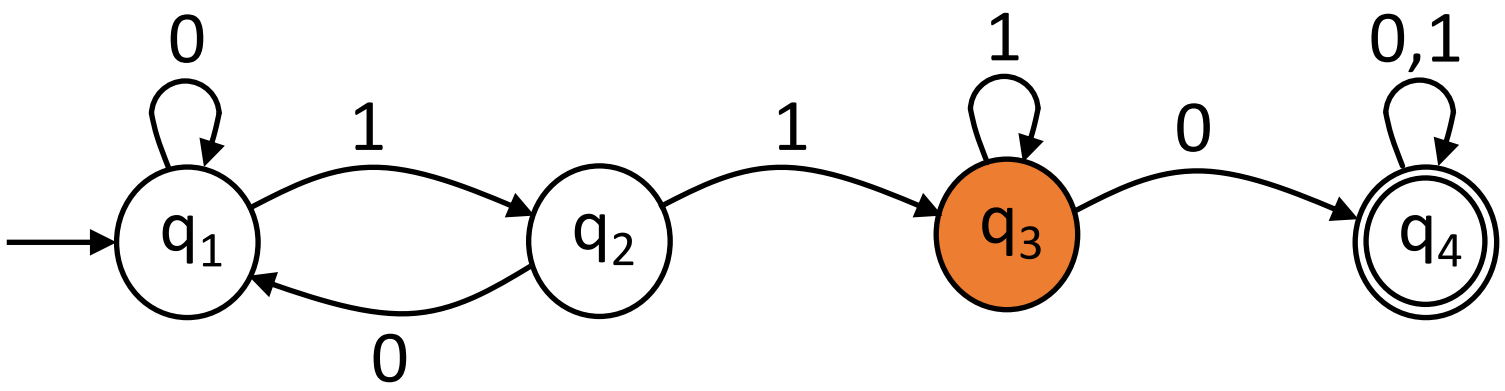


$\{\omega\colon\ \omega$ contains the substring $110\}$

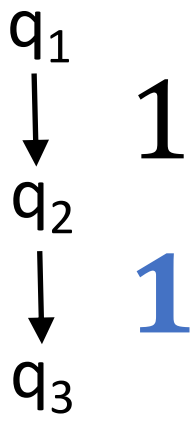


$\omega = \mathbf{1}110$

DFA Processing

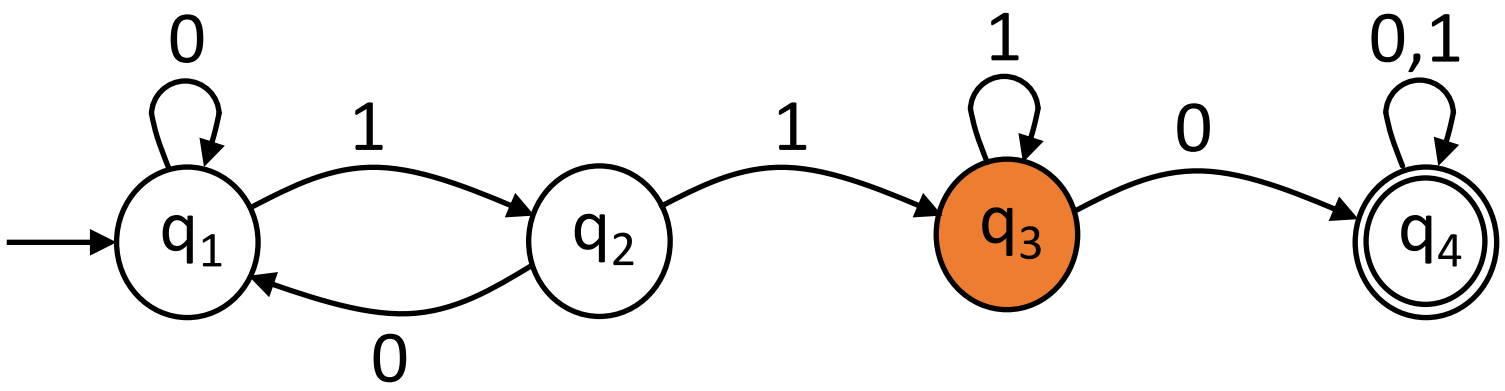


$\{\omega\colon\ \omega$ contains the substring $110\}$



$\omega = 1110$

DFA Processing

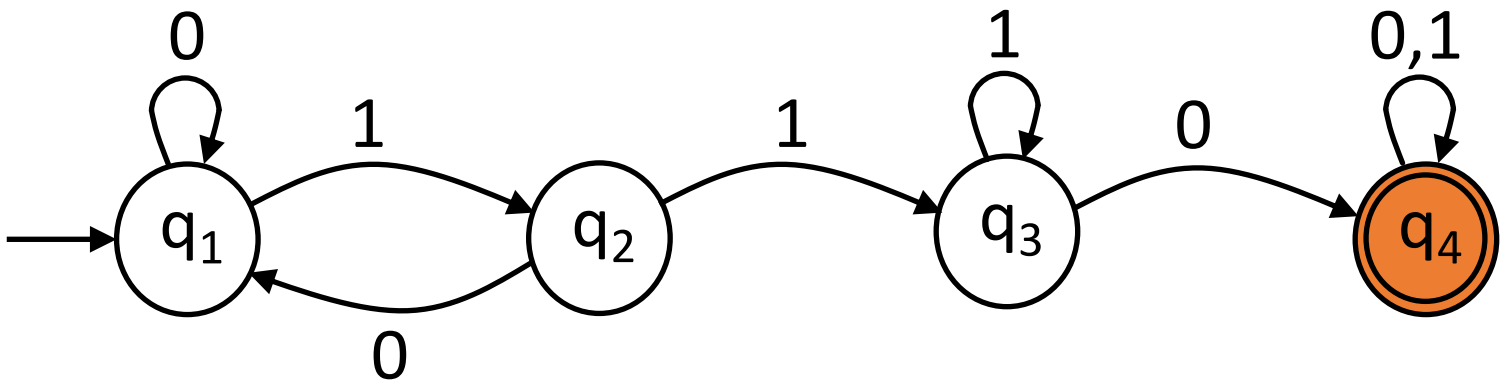


$\{\omega$: ω contains the substring $110\}$

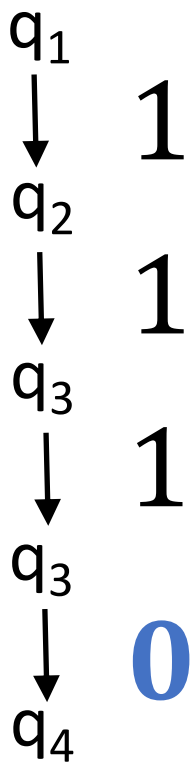
q_1
↓ 1
q_2
↓ 1
q_3
↓ **1**
q_3

$\omega = 11\mathbf{1}0$

DFA Processing

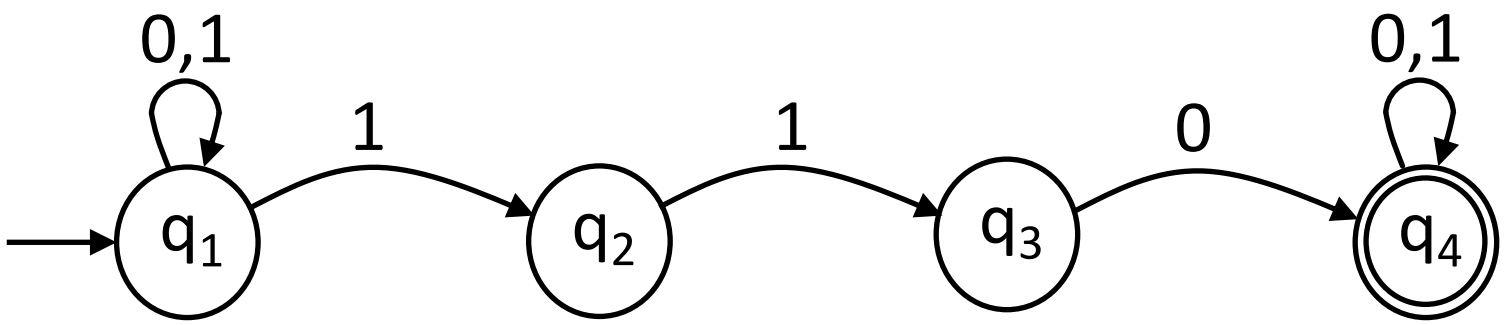


$\{\omega\colon\ \omega$ contains the substring $110\}$



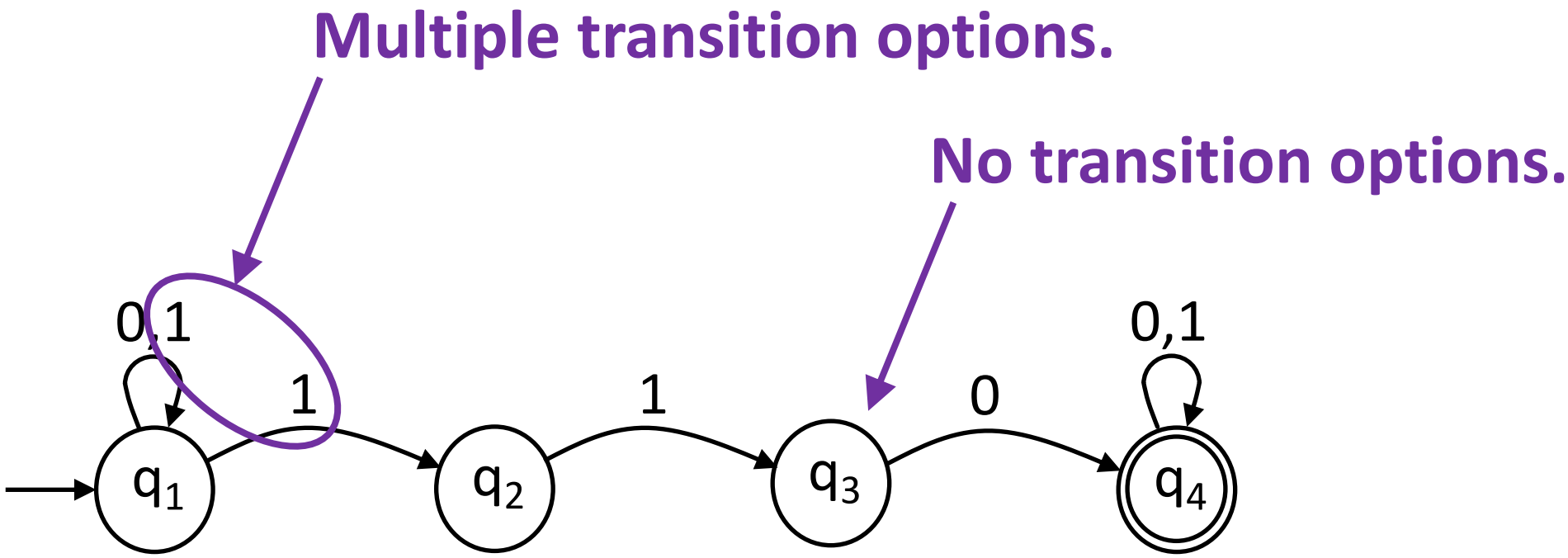
$\omega = 1110$

NFA Processing



$\{\omega\colon\ \omega$ contains the substring $110\}$

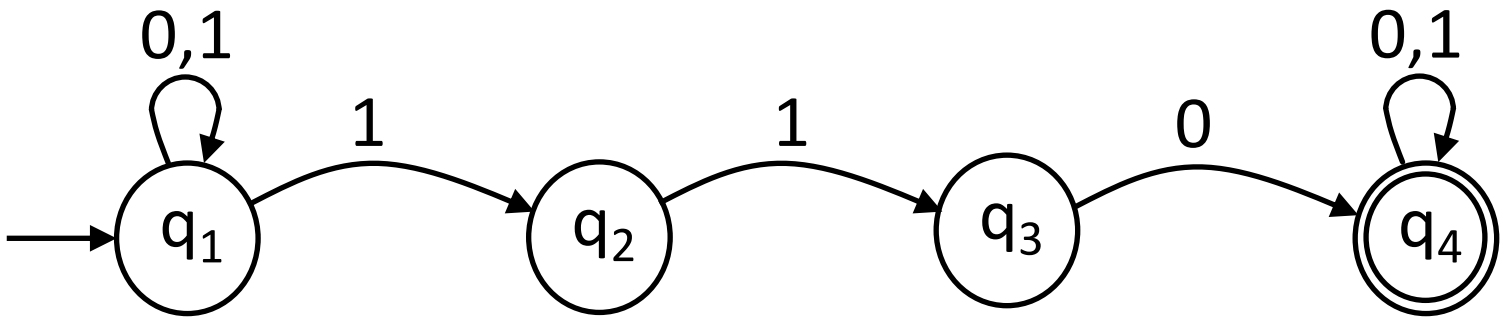
NFA Processing



$\{\omega\colon\ \omega$ contains the substring $110\}$

NFA Processing

1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.

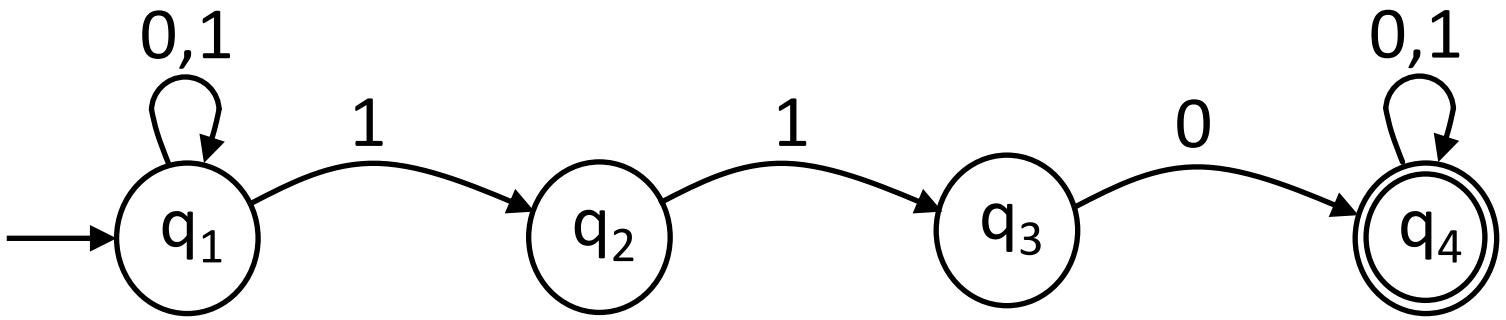


$\{\omega\colon\ \omega$ contains the substring $110\}$

NFA Processing

1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.

q_1

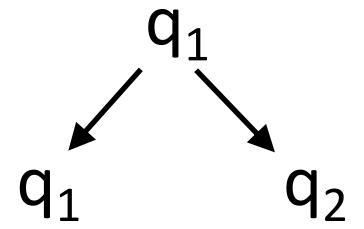


$\{\omega\colon\ \omega$ contains the substring $110\}$

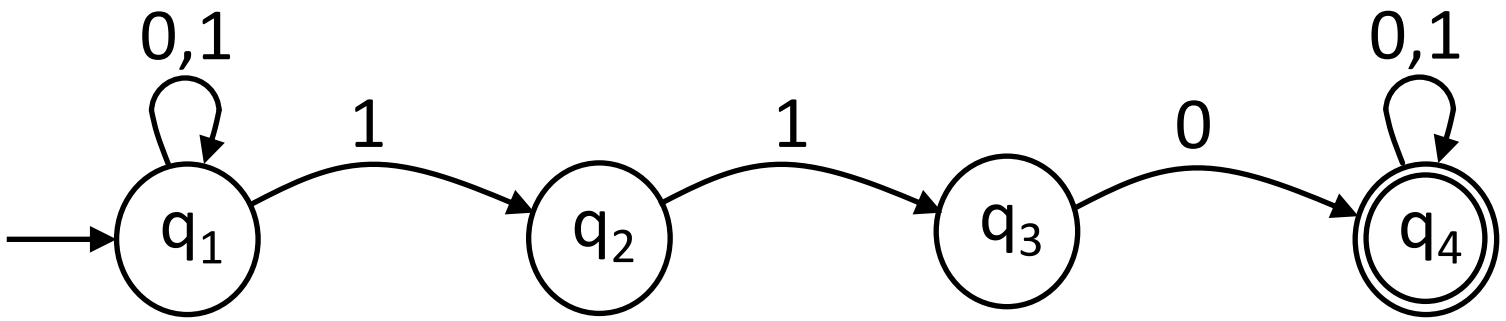
$\omega = 1110$

NFA Processing

1. **If a "decision" is encountered, split and take all options.**
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.



1

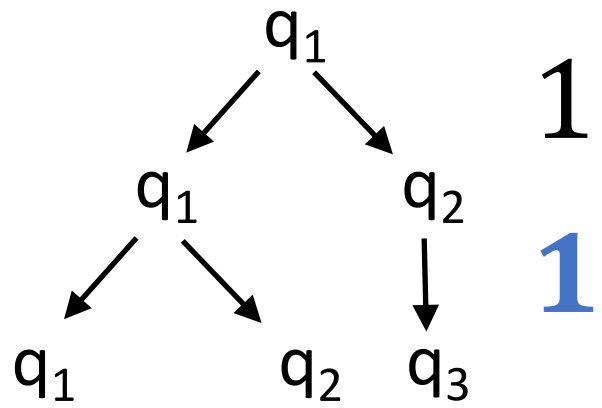


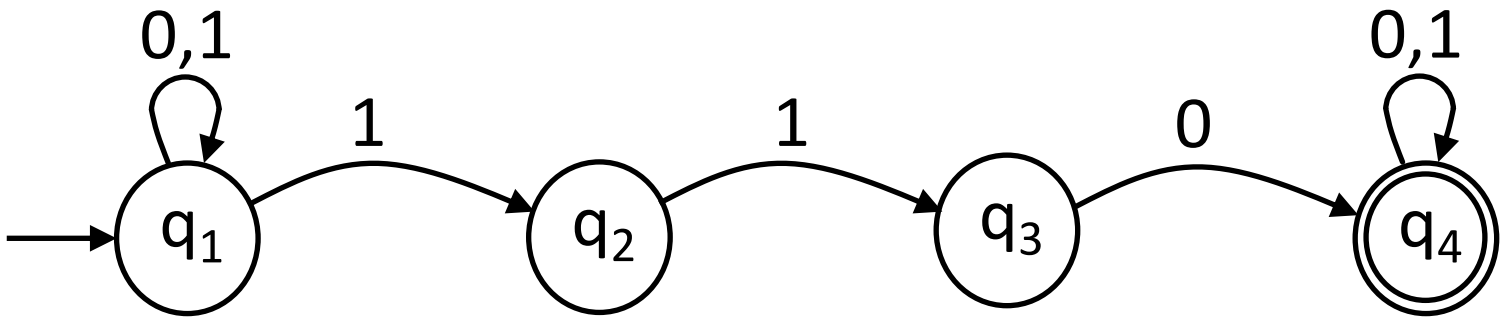
$\{\omega\colon\ \omega$ contains the substring $110\}$

$\omega = \mathbf{1}110$

NFA Processing

1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.



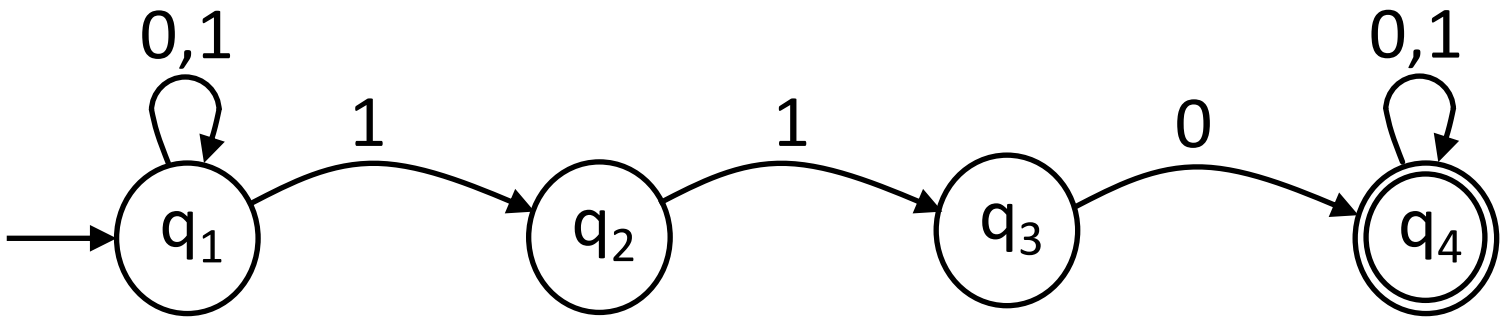


$\{\omega\colon\ \omega$ contains the substring $110\}$

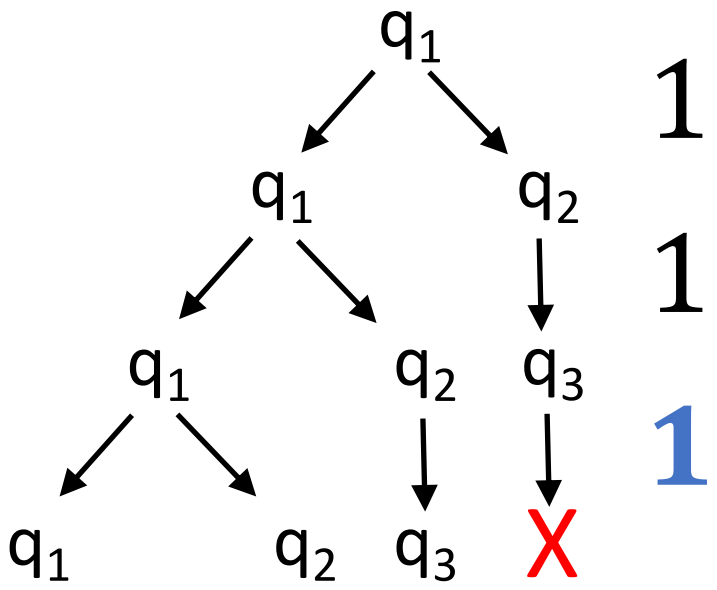
$\omega = 1110$

NFA Processing

1. If a “decision” is encountered, split and take all options.
2. **If input symbol does not match any outgoing transitions, that branch dies.**
3. If any branch ends in an accept state, accept. If not, reject.



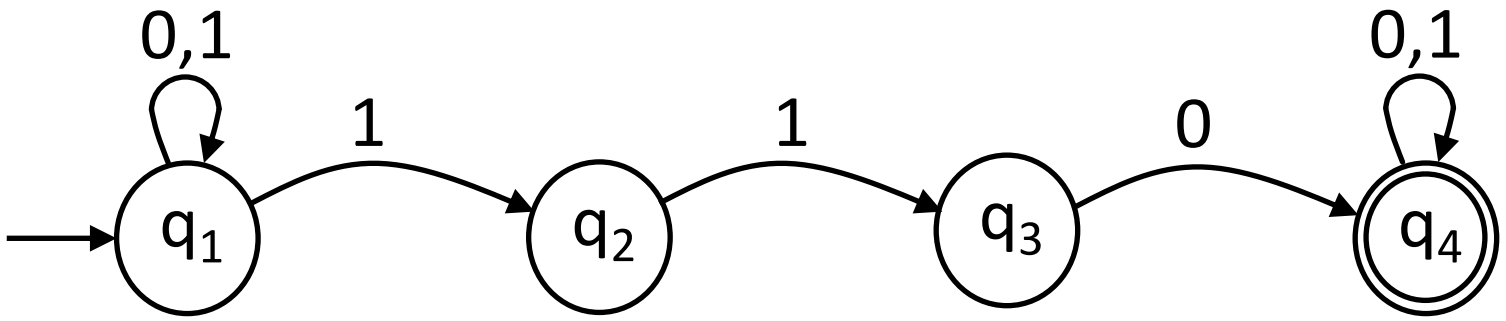
$\{\omega\colon\ \omega$ contains the substring $110\}$



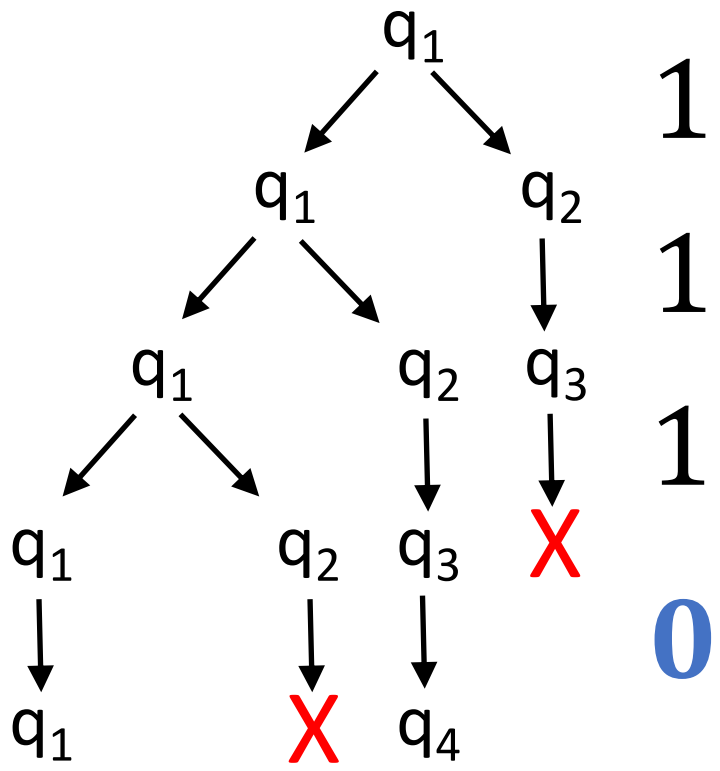
$\omega = 1110$

NFA Processing

1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.



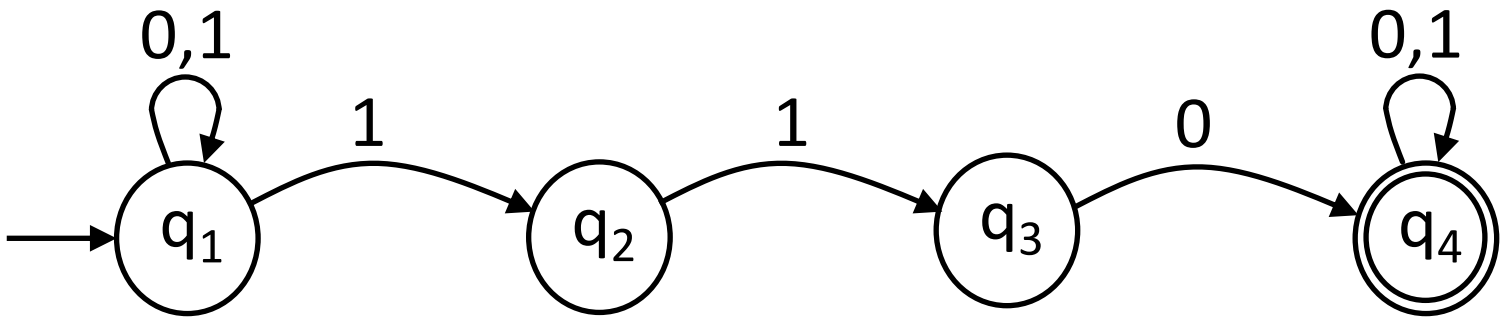
$\{\omega\colon\ \omega$ contains the substring $110\}$



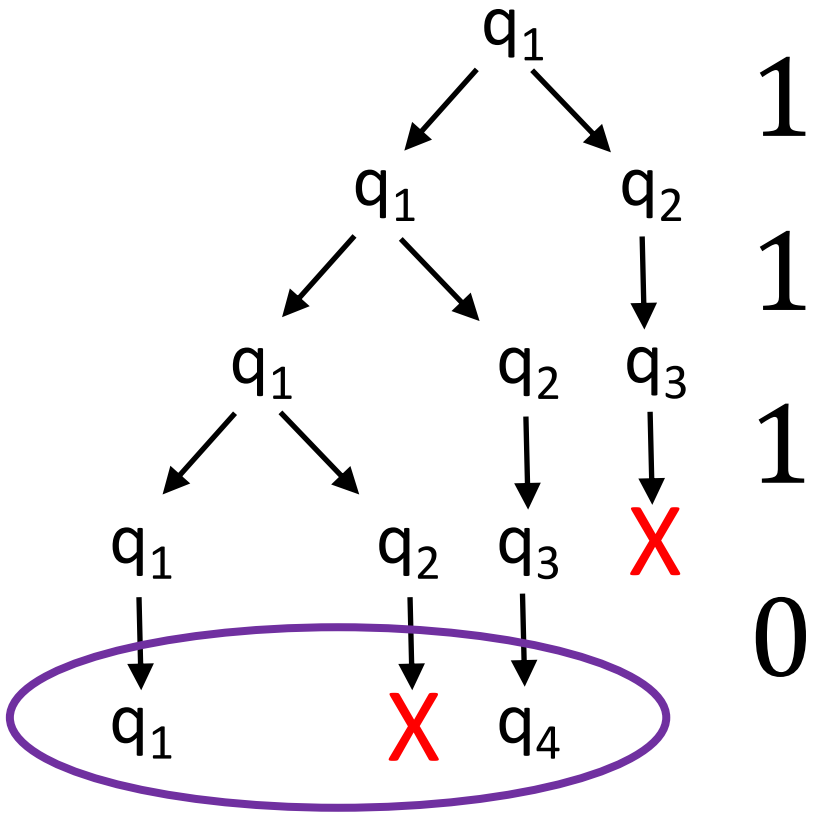
$\omega = 1110$

NFA Processing

1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. **If any branch ends in an accept state, accept. If not, reject.**



$\{\omega\colon\ \omega$ contains the substring $110\}$

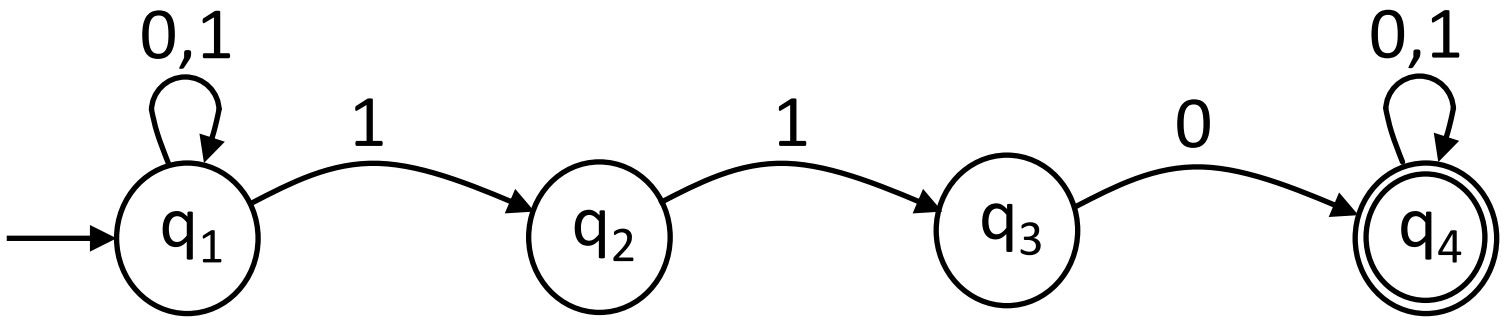


$\omega = 1110$

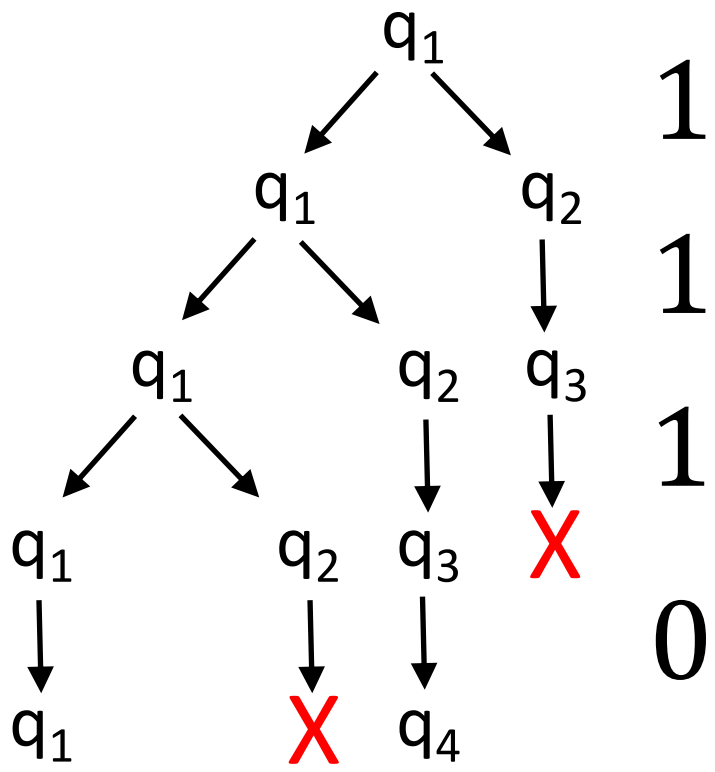
NFA Processing

1. If a "decision" is encountered, split
   ar
2. If
   ou
3. If
   accept. If not, reject.

Alternate approach: If it is possible to end on an accept state, accept.



$\{\omega\colon\ \omega$ contains the substring $110\}$

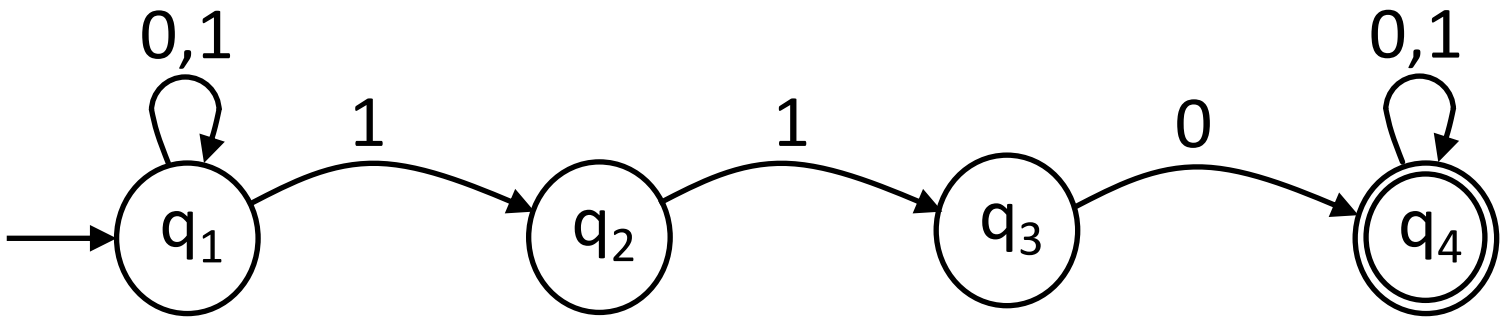


$\omega = 1110$

NFA Processing

1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.

?

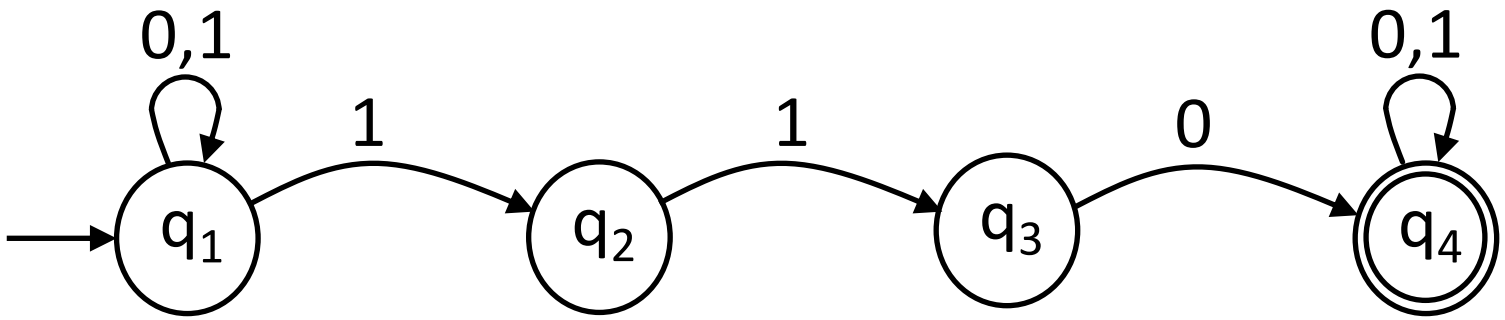


$\{\omega\colon\ \omega$ contains the substring $110\}$

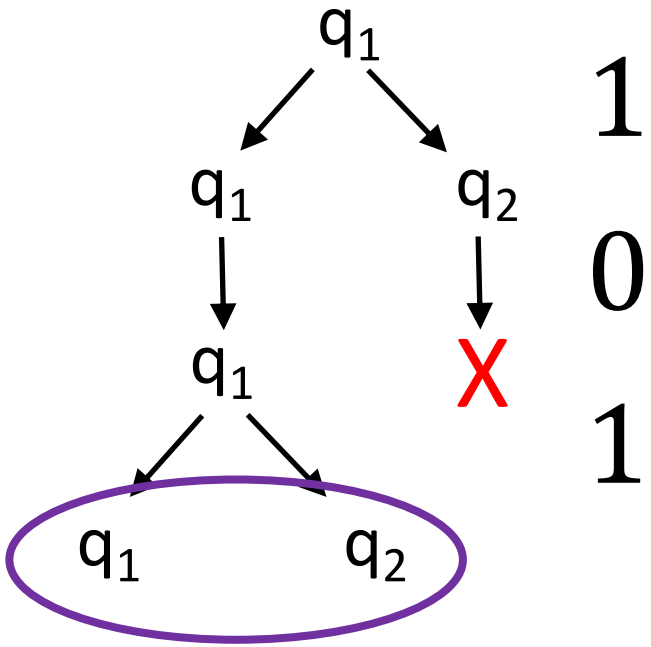
$\omega = 101$

NFA Processing

1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.



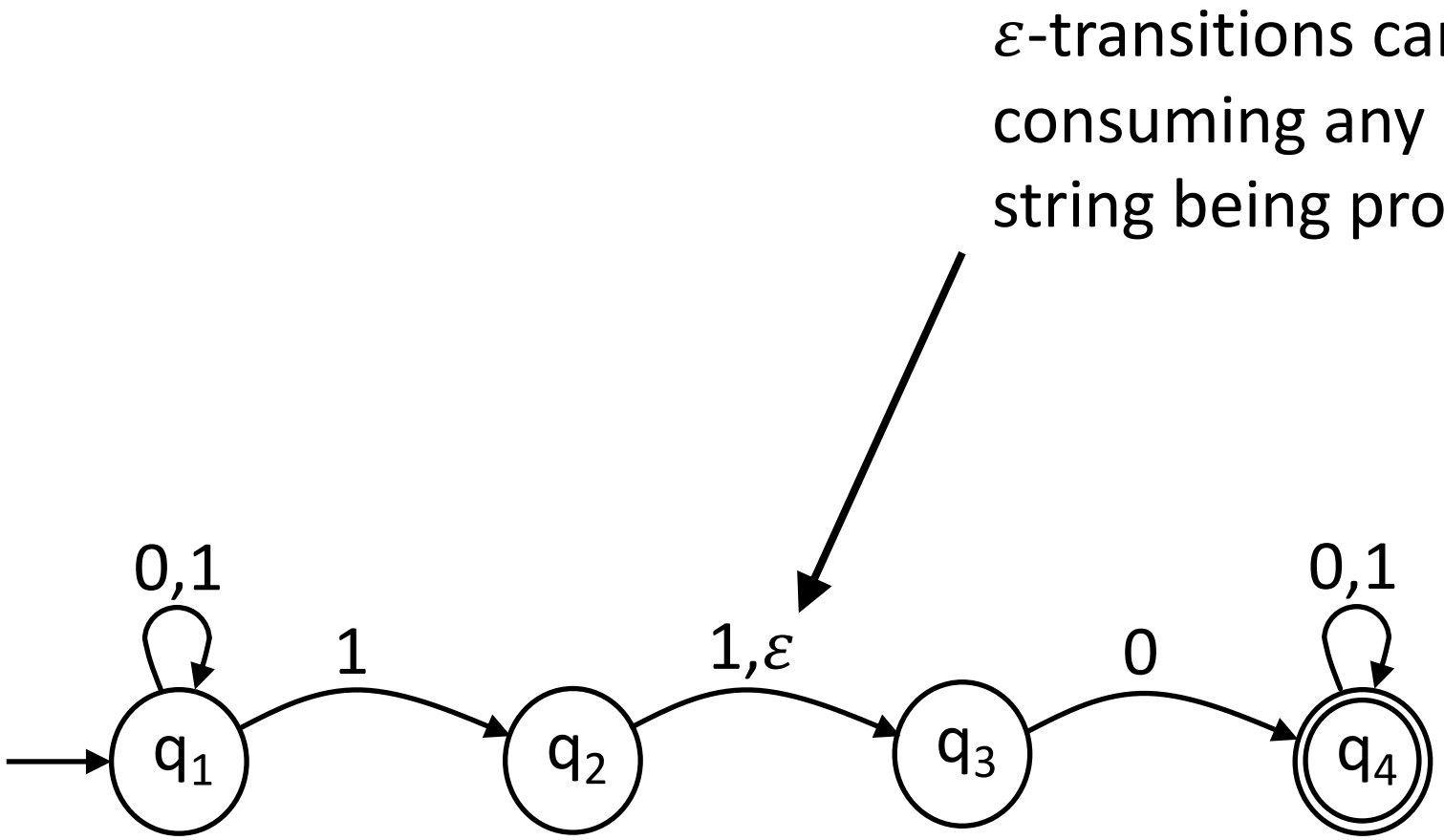
$\{\omega\colon\ \omega$ contains the substring $110\}$



$\omega = 101$

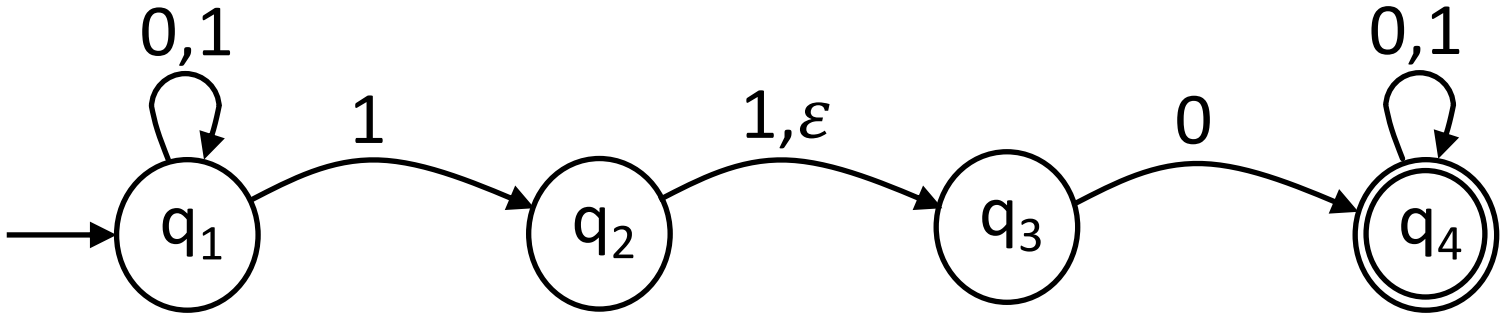
ε-Transitions

ε-transitions can be taken without consuming any characters from the string being processed.



ε-Transitions

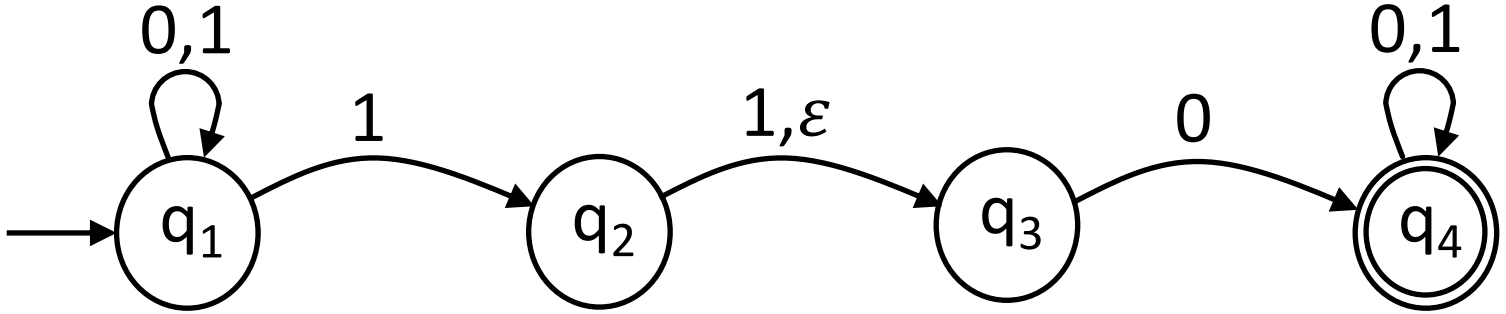
1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.
4. **If ε is encountered, split and take all options without consuming a character from string.**



ε-Transitions

q_1

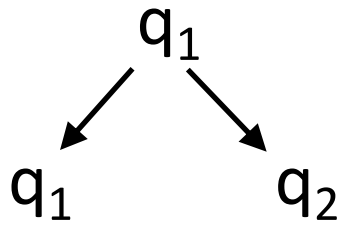
1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.
4. If ε is encountered, split and take all options without consuming a character from string.



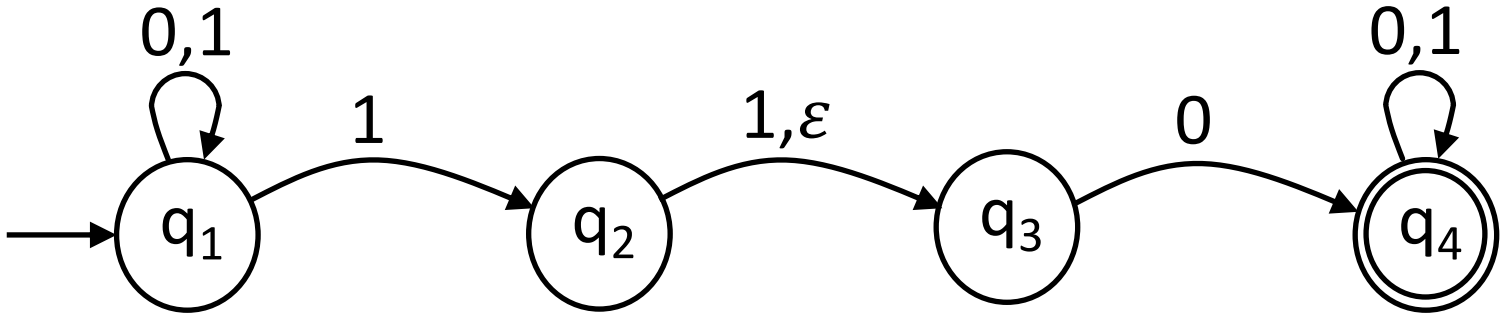
$\omega = 101$

ε-Transitions

1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.
4. If ε is encountered, split and take all options without consuming a character from string.



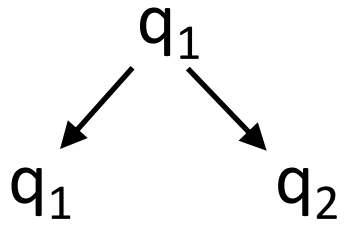
1



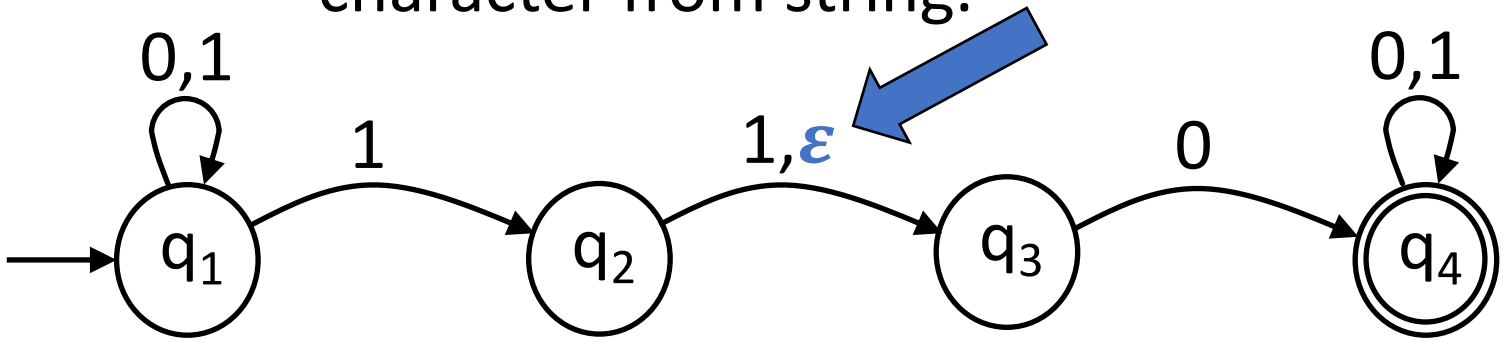
$\omega = \mathbf{1}01$

ε-Transitions

1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.
4. If ε is encountered, split and take all options without consuming a character from string.



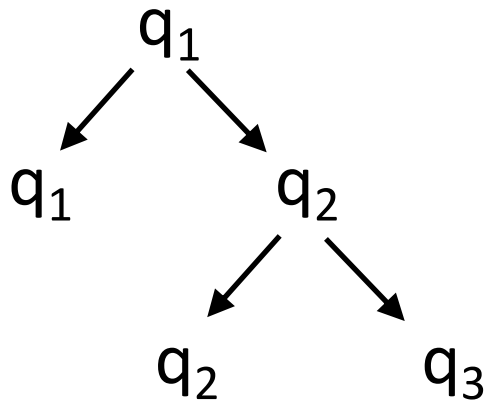
1



$\omega = 101$

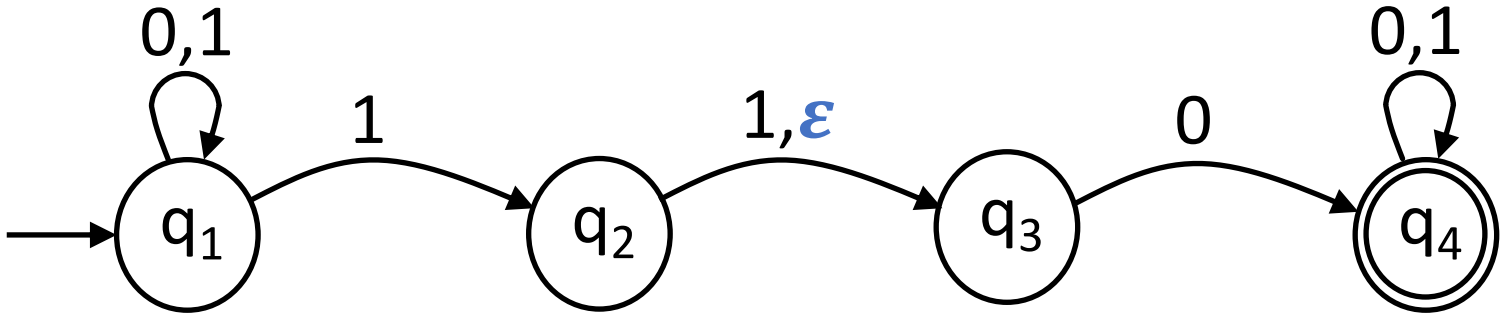
ε-Transitions

1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.
4. **If ε is encountered, split and take all options without consuming a character from string.**



1

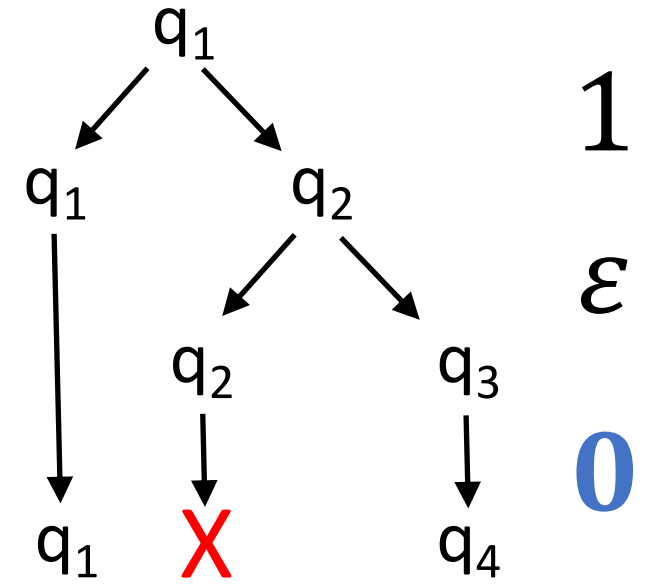
ε

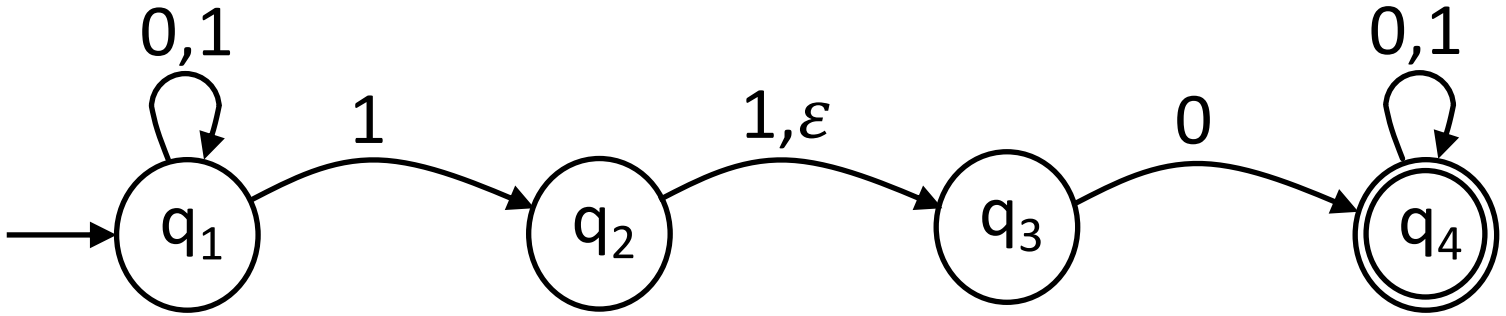


$\omega = 101$

ε-Transitions

1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.
4. If ε is encountered, split and take all options without consuming a character from string.

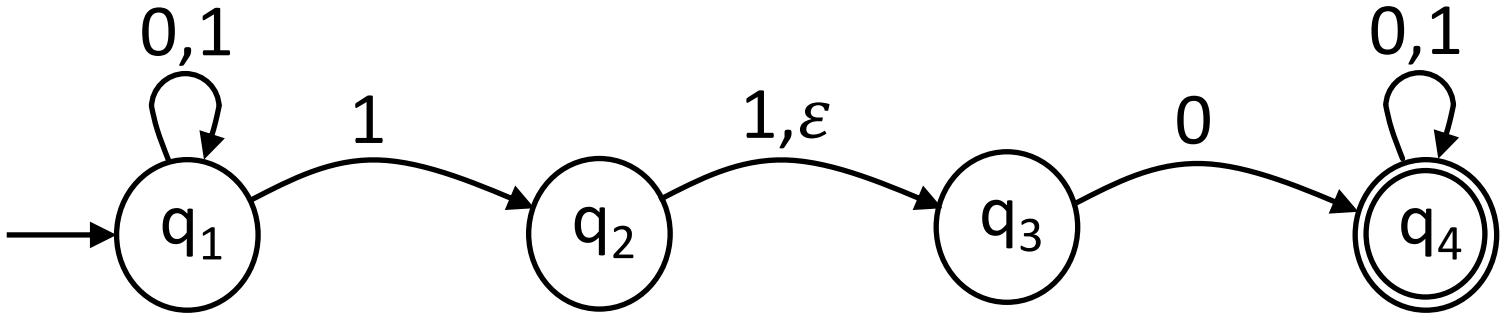


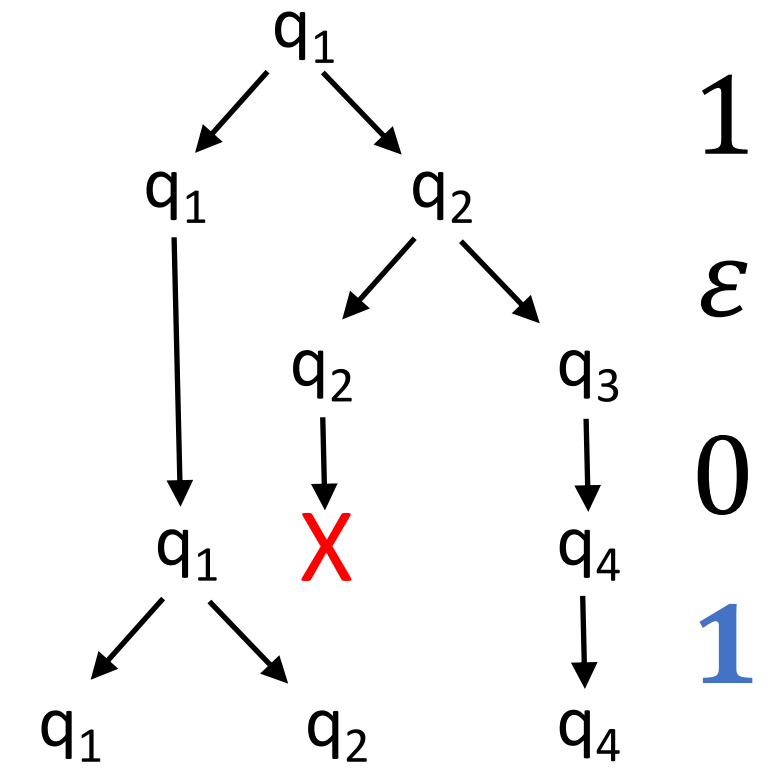


$\omega = 101$

ε-Transitions

1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.
4. If ε is encountered, split and take all options without consuming a character from string.

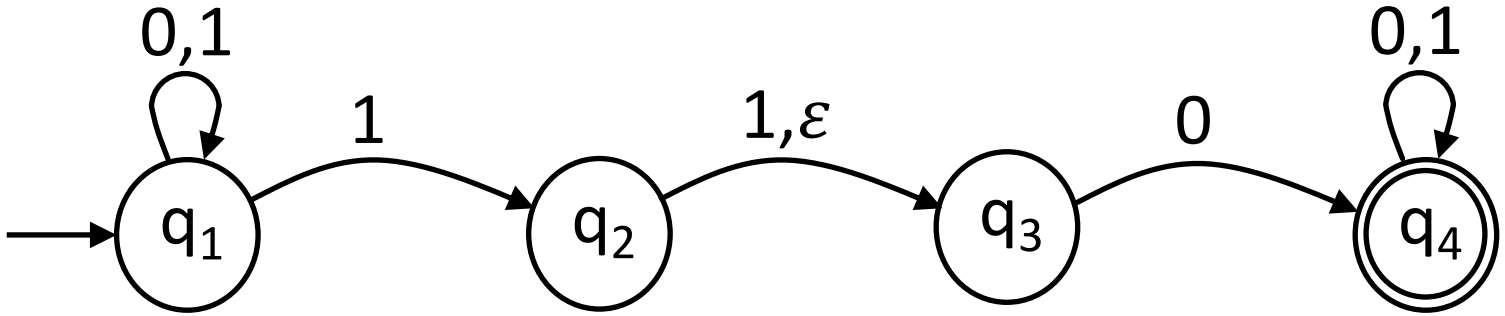


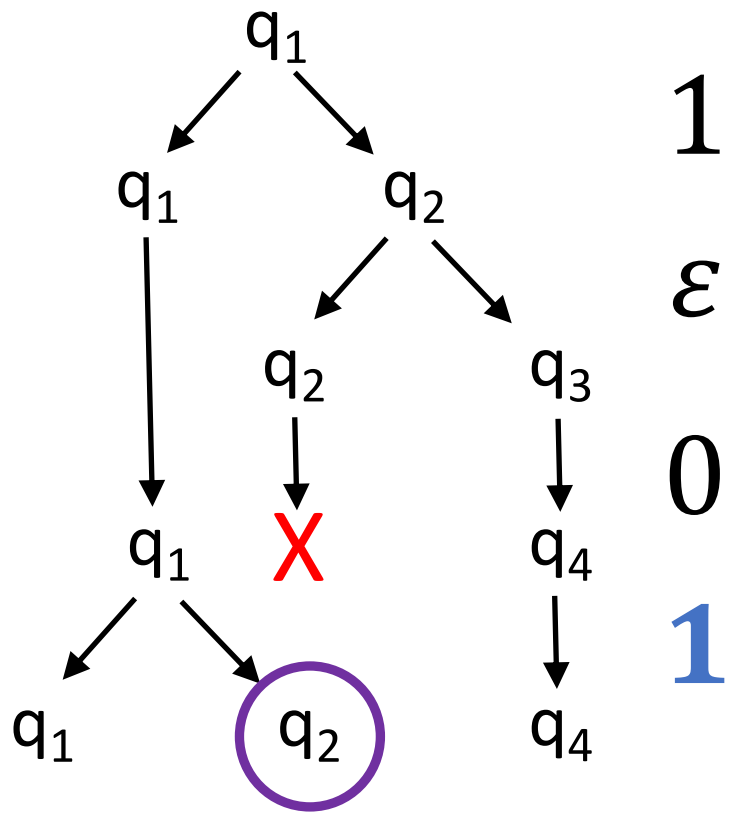


$\omega = 101$

ε-Transitions

1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.
4. If ε is encountered, split and take all options without consuming a character from string.

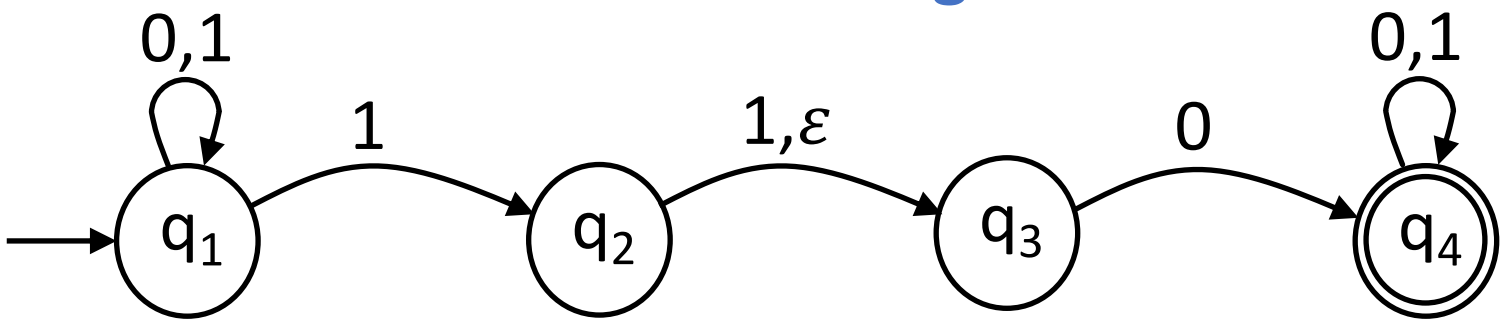


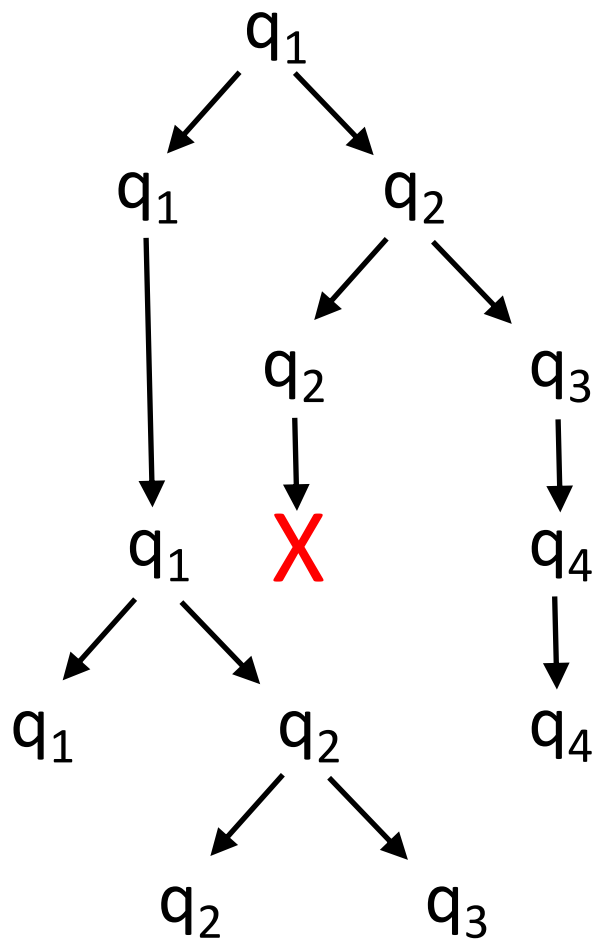


$\omega = 101$

ε-Transitions

1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.
4. **If ε is encountered, split and take all options without consuming a character from string.**

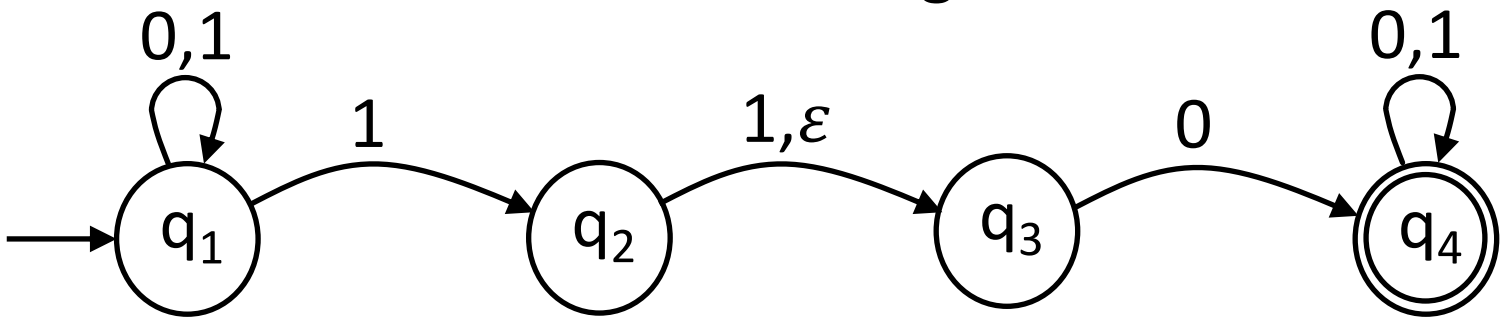


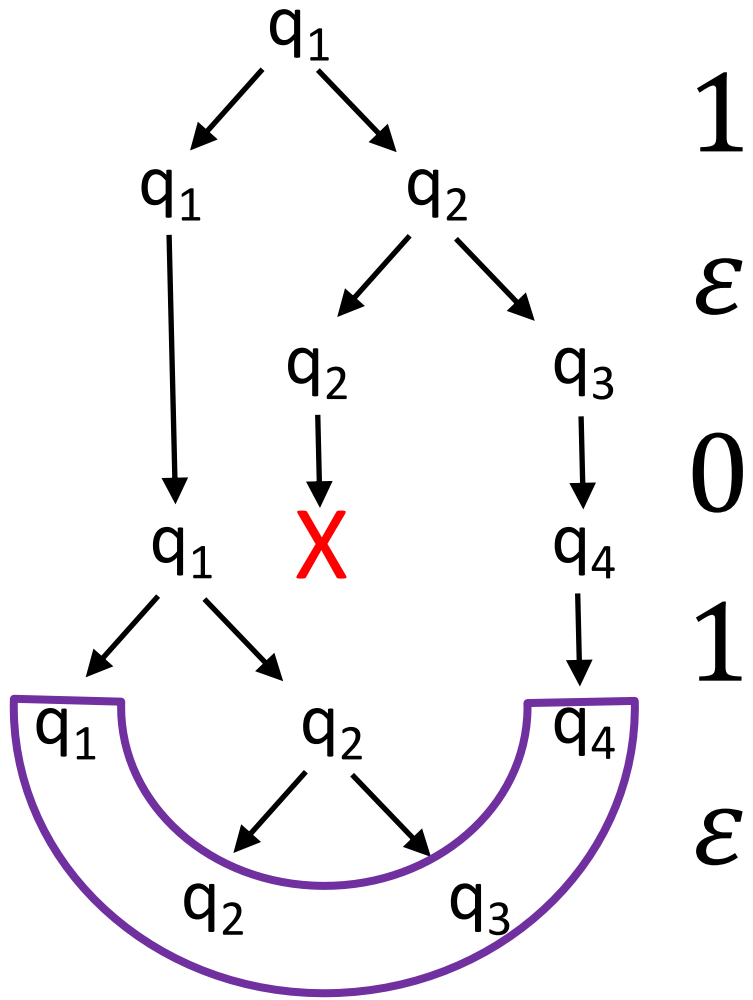


$\omega = 101$

ε-Transitions

1. If a “decision” is encountered, split and take all options.
2. If input symbol does not match any outgoing transitions, that branch dies.
3. If any branch ends in an accept state, accept. If not, reject.
4. If ε is encountered, split and take all options without consuming a character from string.



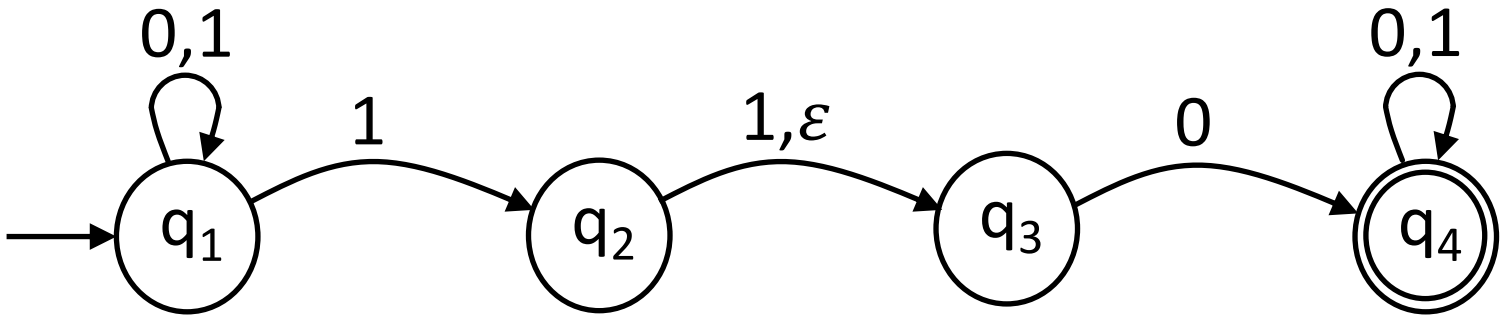


$\omega = 101$

NFA Formal Definition

NFAs consist of:

1. Finite set of states, Q.
2. Finite alphabet, Σ.
3. Transition function, $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$.
4. Start state, $q_0 \in Q$.
5. Set of accept states, $F \subseteq Q$.



NFA Formal Definition

Power set of Q.

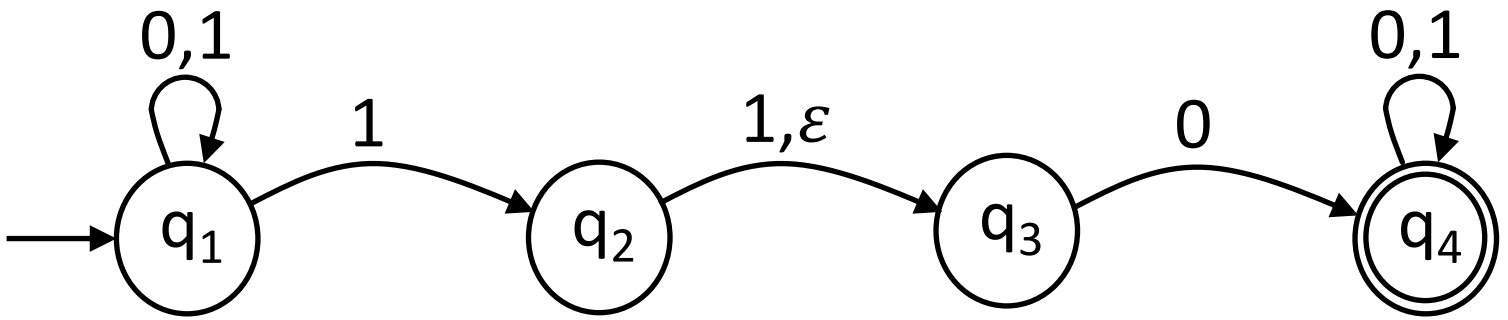
I.e. set of all subsets.

E.g. $Q = \{q_1, q_2\}$

$\Rightarrow \mathcal{P}(Q) = \{\emptyset, \{q_1\}, \{q_2\}, \{q_1, q_2\}\}$

NFAs consist of:

1. Finite set of states, Q.
2. Finite alphabet, Σ.
3. Transition function, $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$.
4. Start state, $q_0 \in Q$.
5. Set of accept states, $F \subseteq Q$.



NFA Formal Definition

Power set of Q.

I.e. set of all subsets.

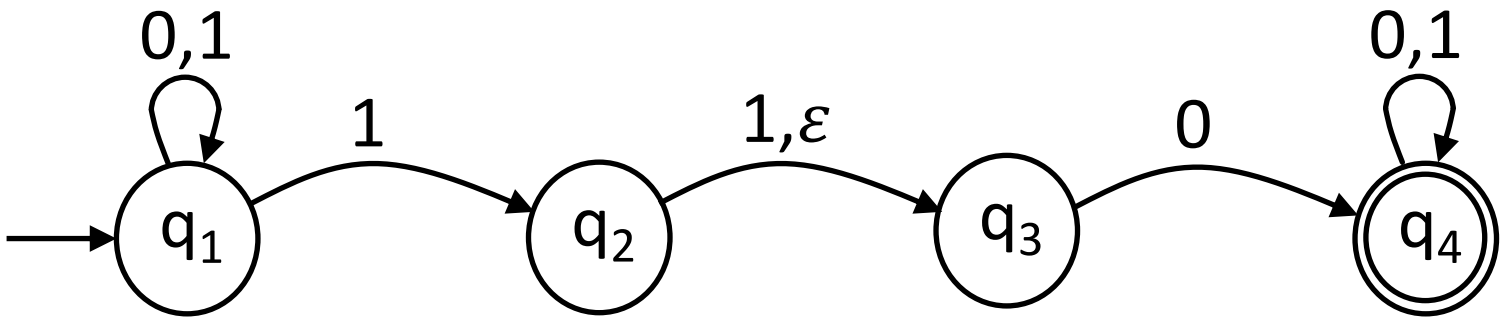
E.g. $Q = \{q_1, q_2\}$

$\Rightarrow \mathcal{P}(Q) = \{\emptyset, \{q_1\}, \{q_2\}, \{q_1, q_2\}\}$

I.e. $\exists$ 0 or more transitions for each $e \in \Sigma \cup \{\varepsilon\}$ at each state

NFAs consist of:

1. Finite set of states, Q.
2. Finite alphabet, Σ.
3. Transition function, $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$.
4. Start state, $q_0 \in Q$.
5. Set of accept states, $F \subseteq Q$.

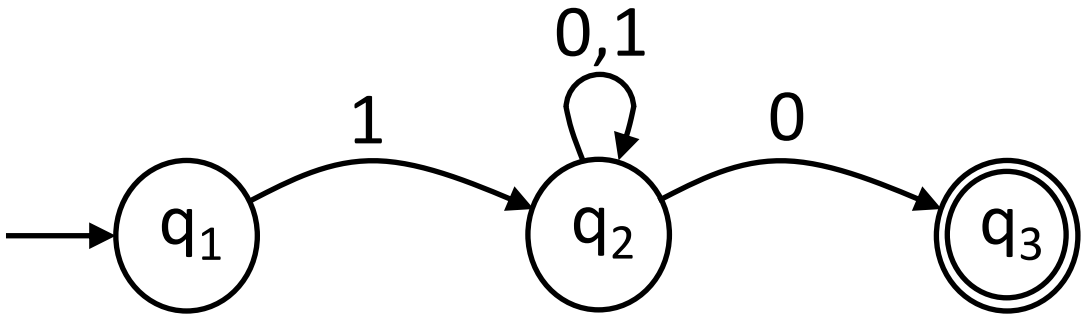


NFA Practice

What is the NFA that accepts $\{\omega:\ \omega$ starts with 1 **and** ends with $0\}$**?**

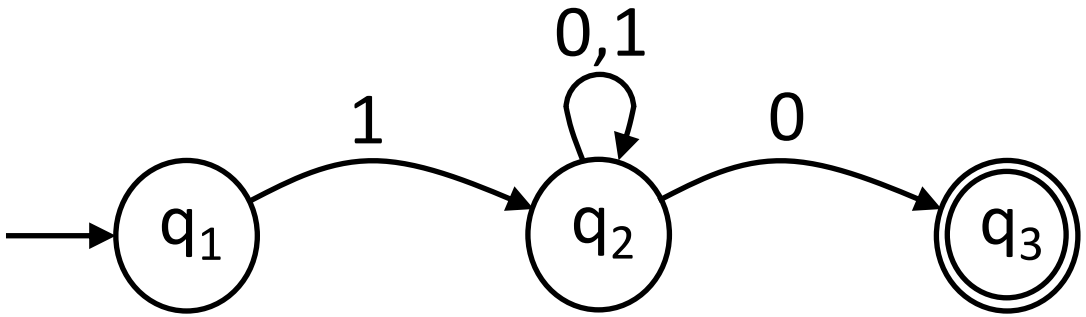
NFA Practice

What is the NFA that accepts $\{\omega:\ \omega$ starts with 1 **and** ends with $0\}$?



NFA Practice

What is the NFA that accepts $\{\omega:\ \omega$ starts with 1 **and** ends with $0\}$?



Only ω's that start with 1 get to q_2.
Any string that gets to q_2, can get to q_3 and terminate, if it ends with 0.