NFA/DFA Equivalence

CSCI 338

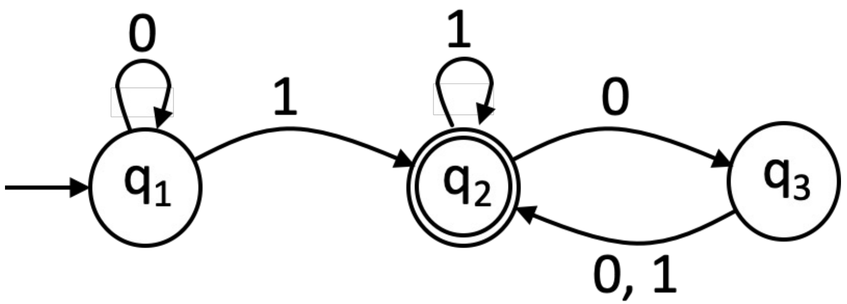
Definitions

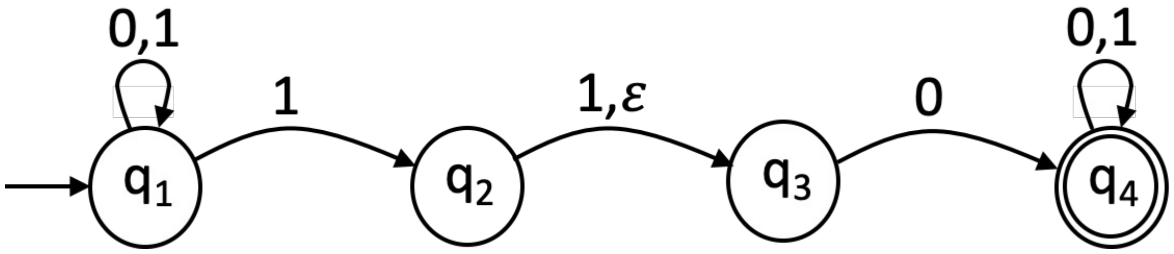
DFAs consist of:

1. Finite set of states, Q.
2. Finite alphabet, Σ.
3. Transition function, $\delta: Q \times \Sigma \to Q$.
4. Start state, $q_0 \in Q$.
5. Set of accept states, $F \subseteq Q$.

NFAs consist of:

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

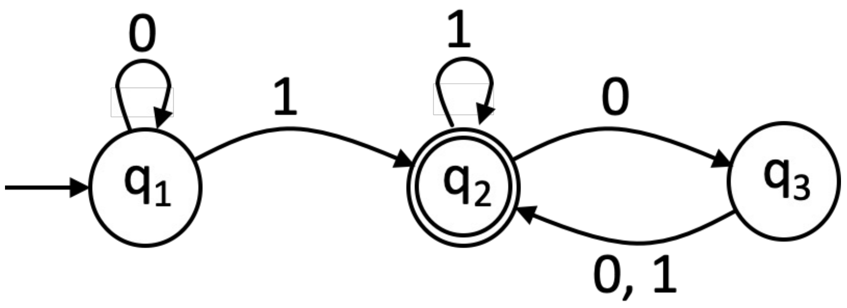




Definitions

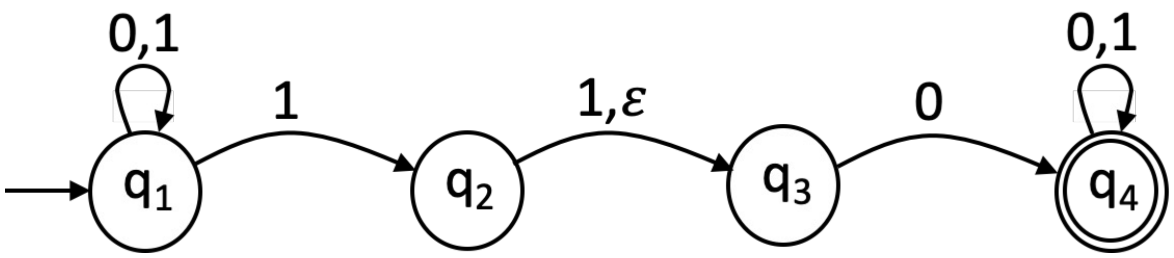
DFAs consist of:

1. Finite set of states, Q.
2. Finite alphabet, Σ.
3. Transition function, $\delta: Q \times \Sigma \to Q$.
4. Start state, $q_0 \in Q$.
5. Set of accept states, $F \subseteq Q$.



NFAs consist of:

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



DFA vs NFA

NFA's have a lot of shiny features, but do they actually get us any new capability?

How would we prove that NFA's **do** provide new capability?

DFA vs NFA

NFA's have a lot of shiny features, but do they actually get us any new capability?

How would we prove that NFA's **<u>do</u>** provide new capability?

> Find some language that can be recognized by an NFA, but not a DFA.

DFA vs NFA

NFA’s have a lot of shiny features, but do they actually get us any new capability?

How would we prove that NFA’s **<u>do</u>** provide new capability?

Find some language that can be recognized by an NFA, but not a DFA.

How would we prove that NFA’s **<u>do not</u>** provide new capability?

DFA vs NFA

NFA’s have a lot of shiny features, but do they actually get us any new capability?

How would we prove that NFA’s **do** provide new capability?

Find some language that can be recognized by an NFA, but not a DFA.

How would we prove that NFA’s **do not** provide new capability?

Show that every language recognized by an NFA can be recognized by a DFA.

DFA vs NFA

NFA's have a lot of shiny features, but do they actually get us any new capability?

How would we prove that NFA's <u>**do**</u> provide new capability?

Find some language that can be recognized by an NFA, but not a DFA.

How would we prove that NFA's <u>do not</u> provide new capability?

Show that every language recognized by an NFA can be recognized by a DFA.

DFA vs NFA

Claim: Every NFA has an equivalent DFA.

Proof Approach:

For any NFA, turn it into a DFA.

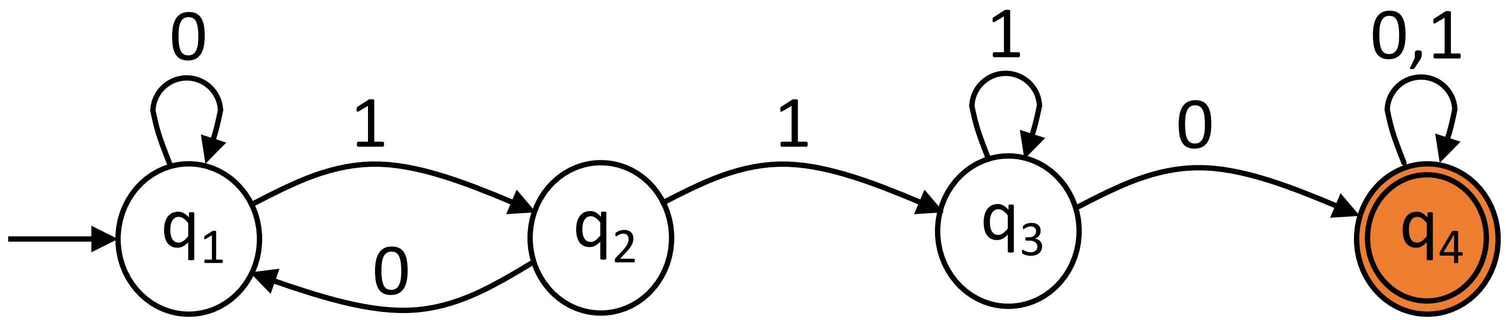
DFA vs NFA

Claim: Every NFA has an equivalent DFA.

Proof Approach:

How did we keep track of our location in a DFA?

$$q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_3 \xrightarrow{1} q_3 \xrightarrow{0} q_4$$



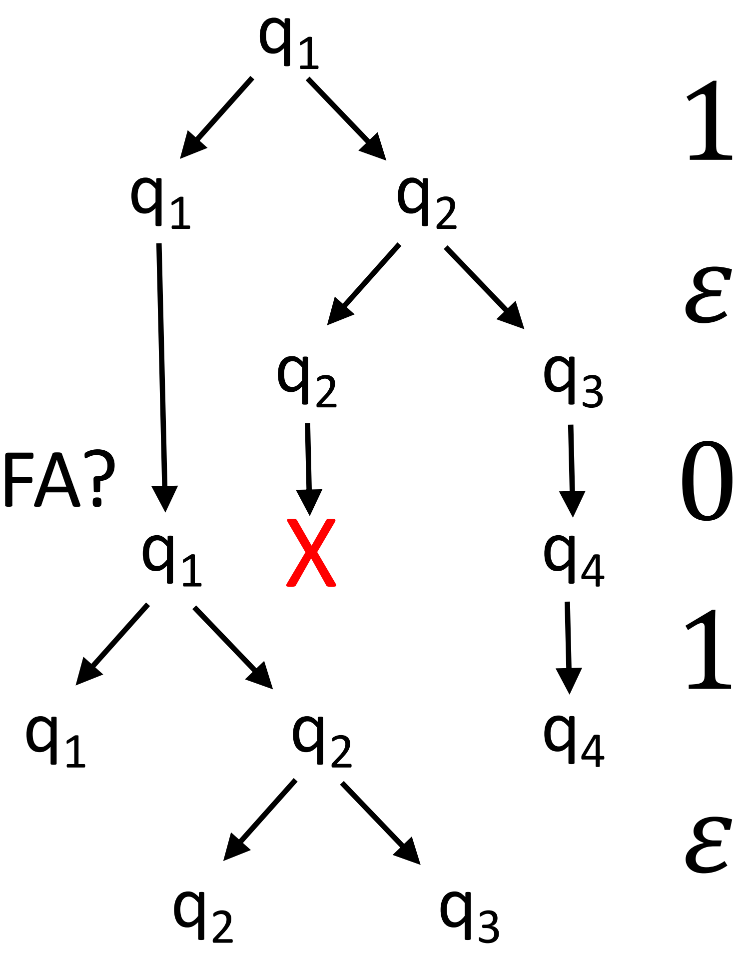
$$\omega = 1110$$

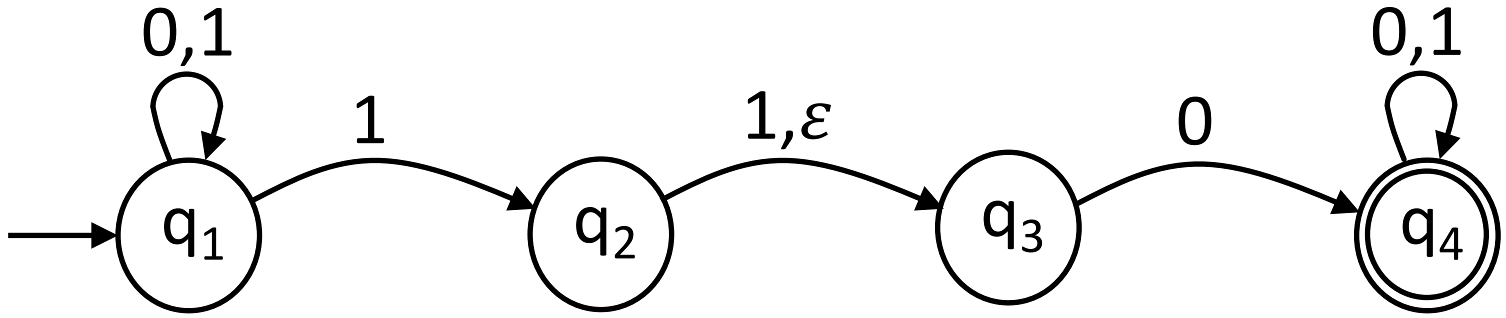
DFA vs NFA

Claim: Every NFA has an equivalent DFA.

Proof Approach:

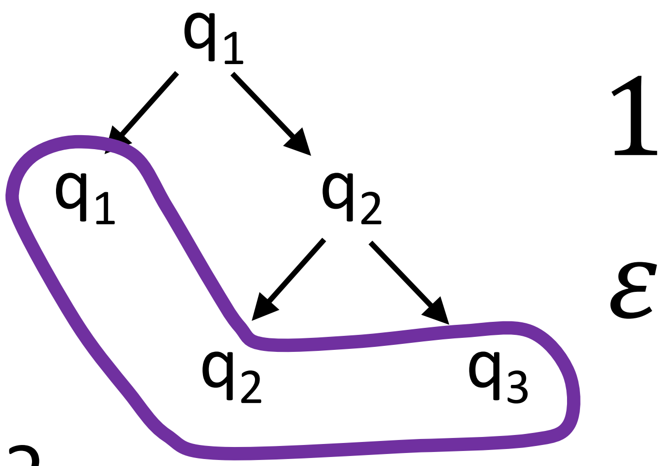
How did we keep track of our location in an NFA?





$$\omega = 101$$

DFA vs NFA

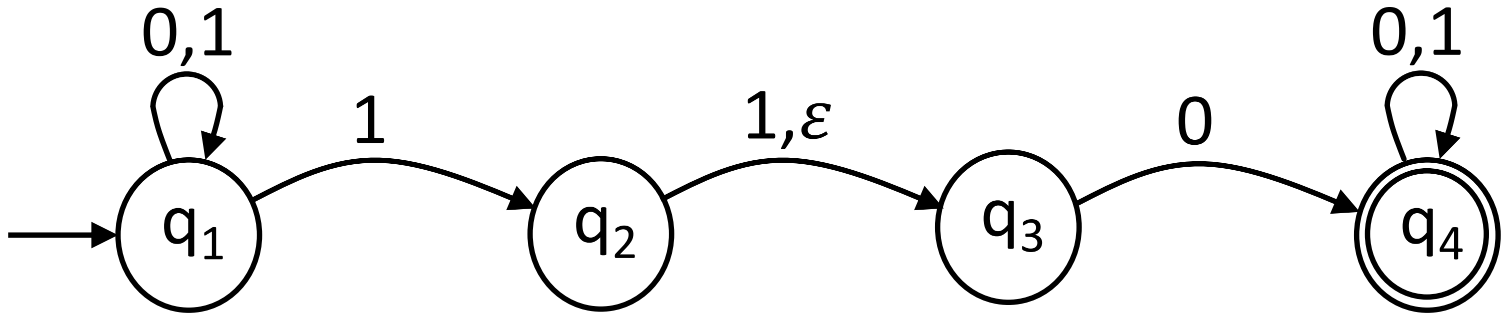


Claim: Every NFA has an equivalent DFA.

Proof Approach:

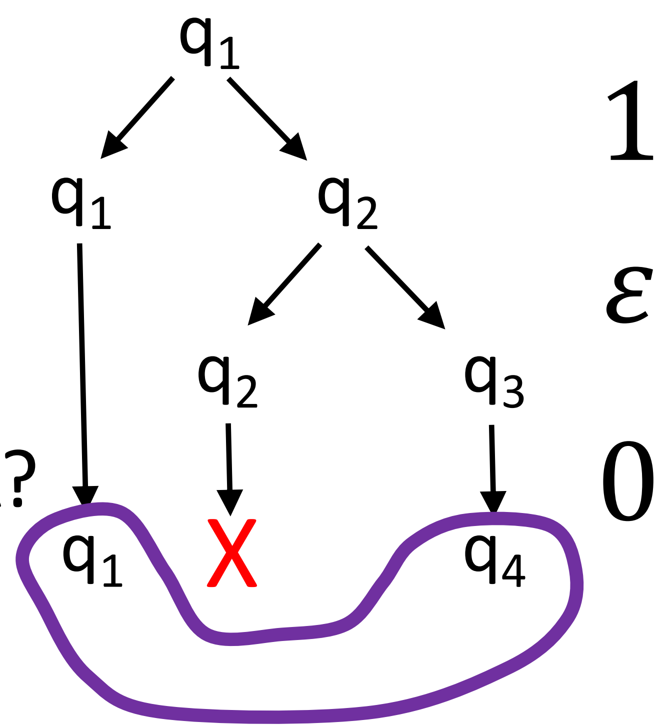
How did we keep track of our location in an NFA?

Set of all states we could possibly be in.



$\omega = 101$

DFA vs NFA

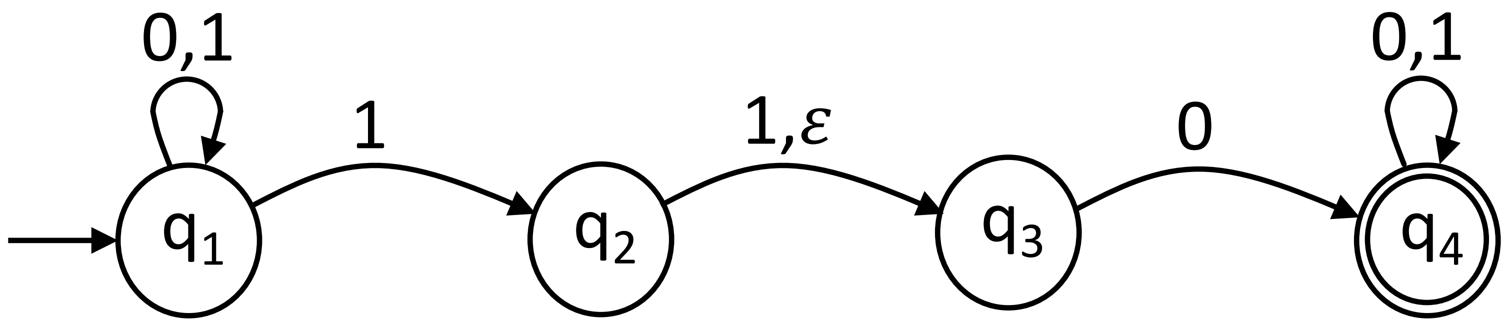


Claim: Every NFA has an equivalent DFA.

Proof Approach:

How did we keep track of our location in an NFA?

Set of all states we could possibly be in.



$\omega = 101$

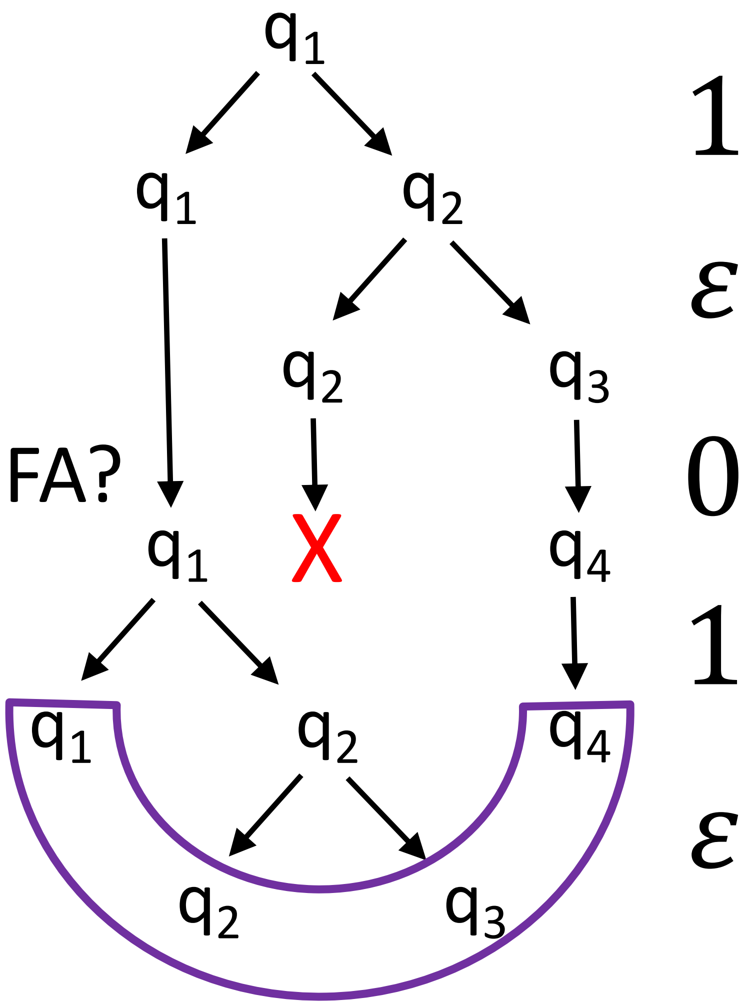
DFA vs NFA

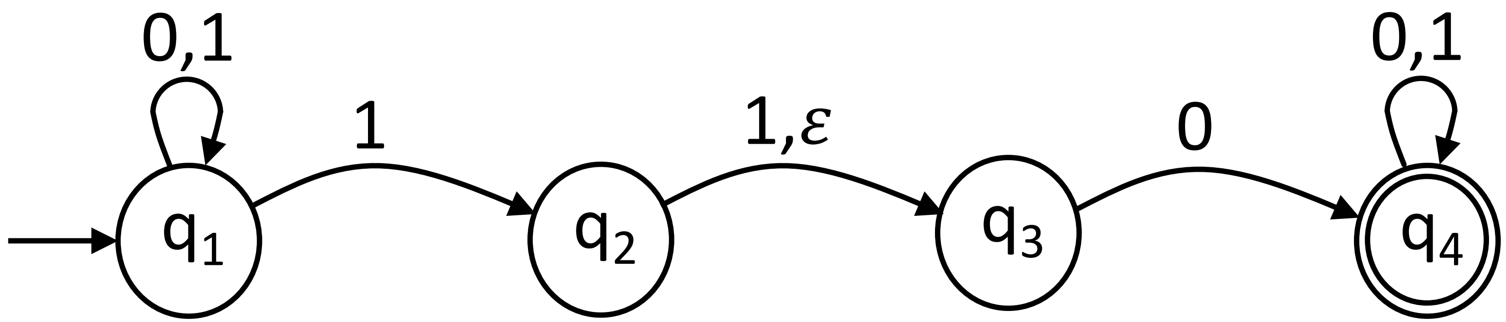
Claim: Every NFA has an equivalent DFA.

Proof Approach:

How did we keep track of our location in an NFA?

Set of all states we could possibly be in.





$\omega = 101$

DFA vs NFA

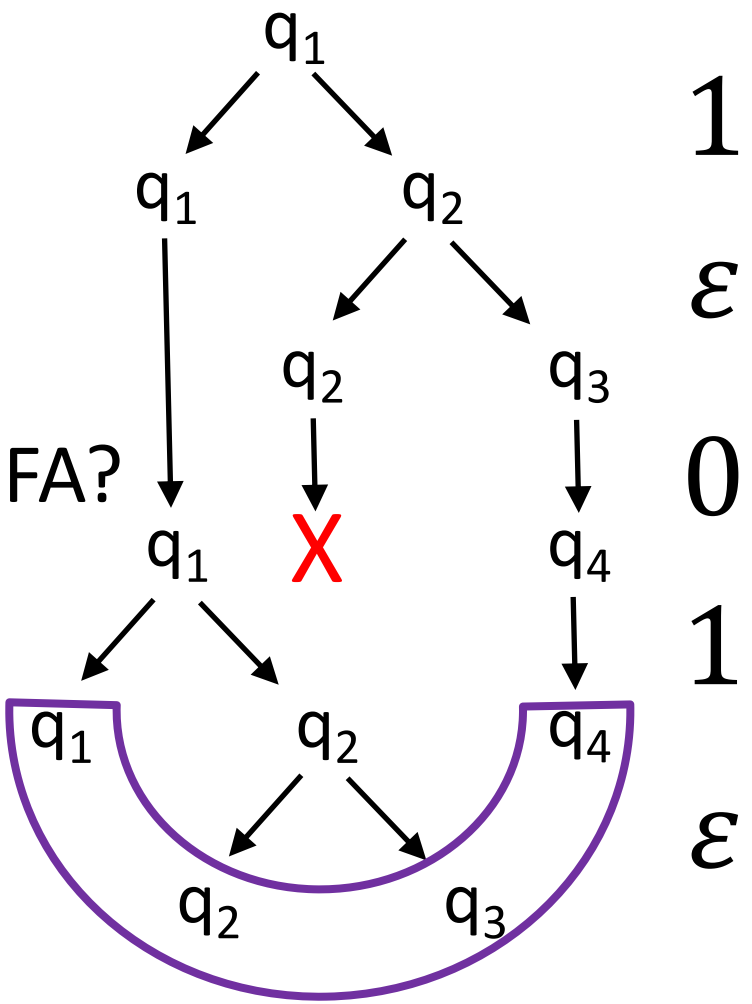
Claim: Every NFA has an equivalent DFA.

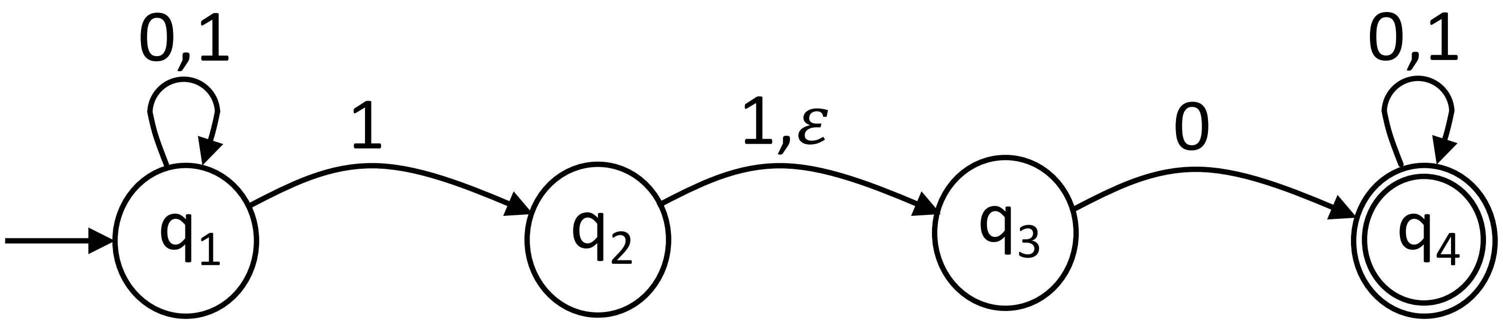
Proof Approach:

How did we keep track of our location in an NFA?

Set of all states we could possibly be in.

What is the set of all possible locations?





$\omega = 101$

DFA vs NFA

Claim: Every NFA has an equivalent DFA.

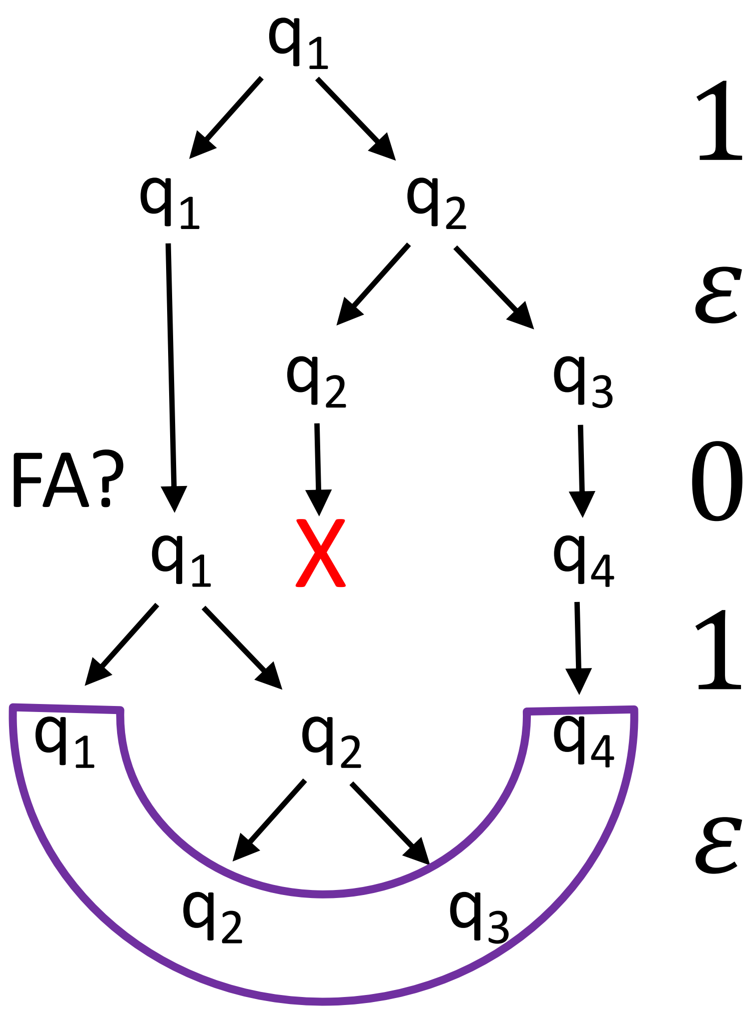
Proof Approach:

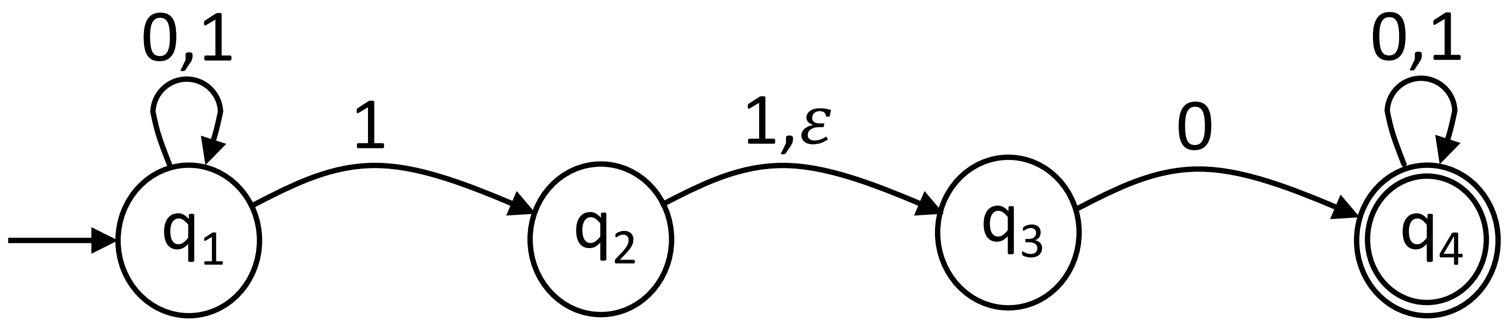
How did we keep track of our location in an NFA?

Set of all states we could possibly be in.

What is the set of all possible locations?

Power set! (set of all subsets)

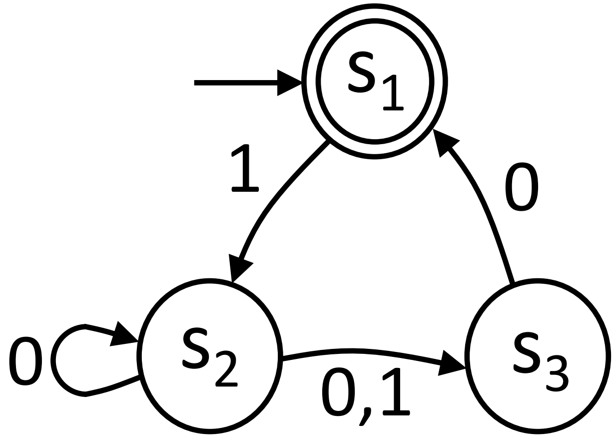




$\omega = 101$

DFA vs NFA

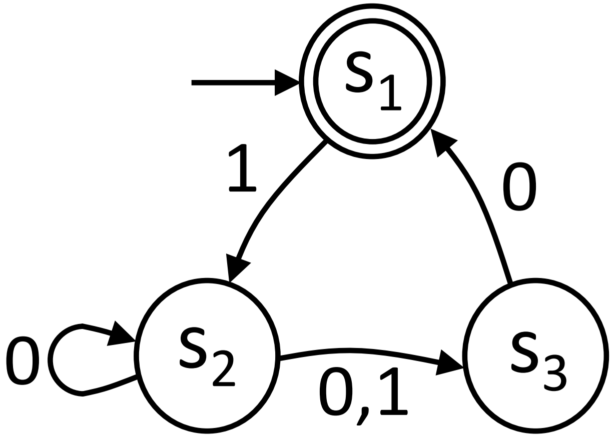
NFA



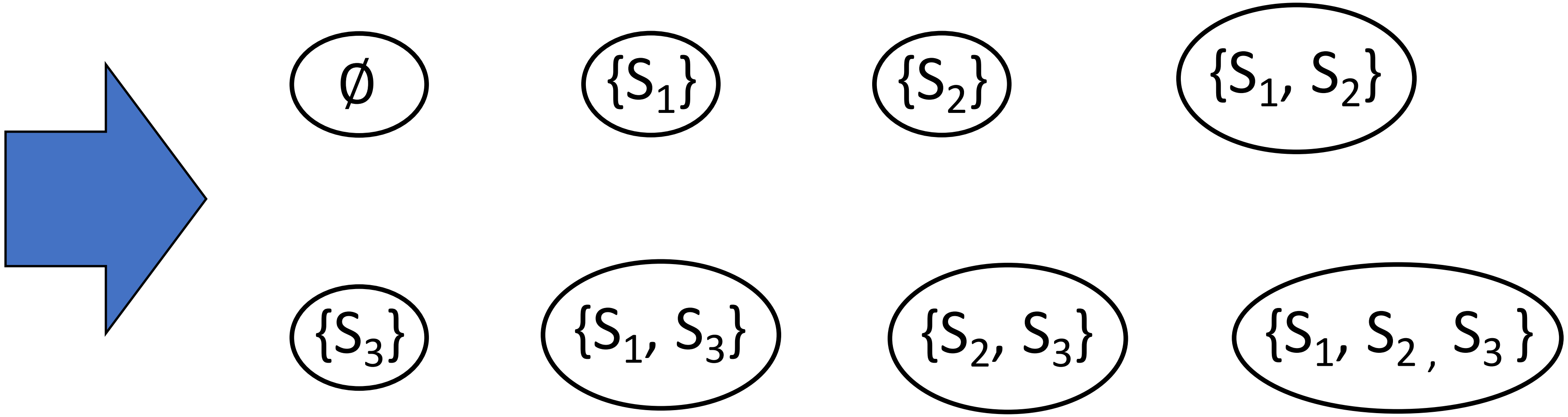
DFA

DFA vs NFA

NFA



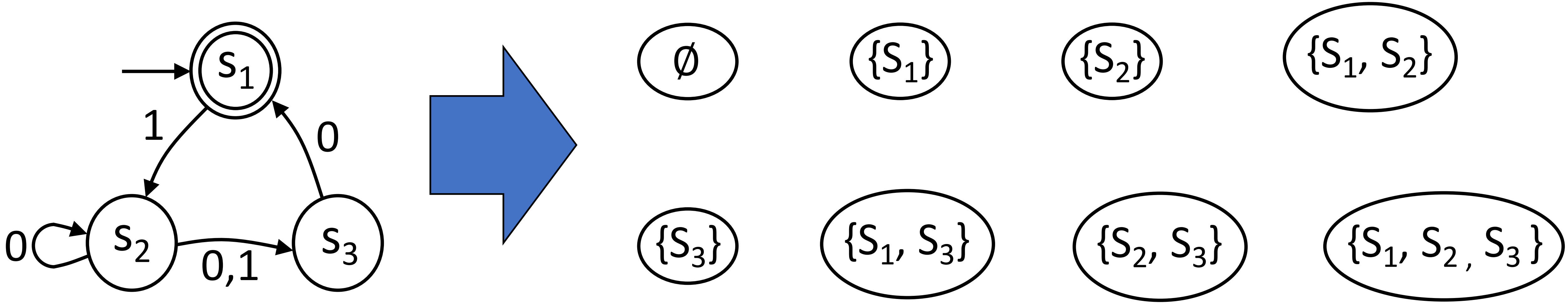
DFA



DFA states = Power set of NFA states.

DFA vs NFA

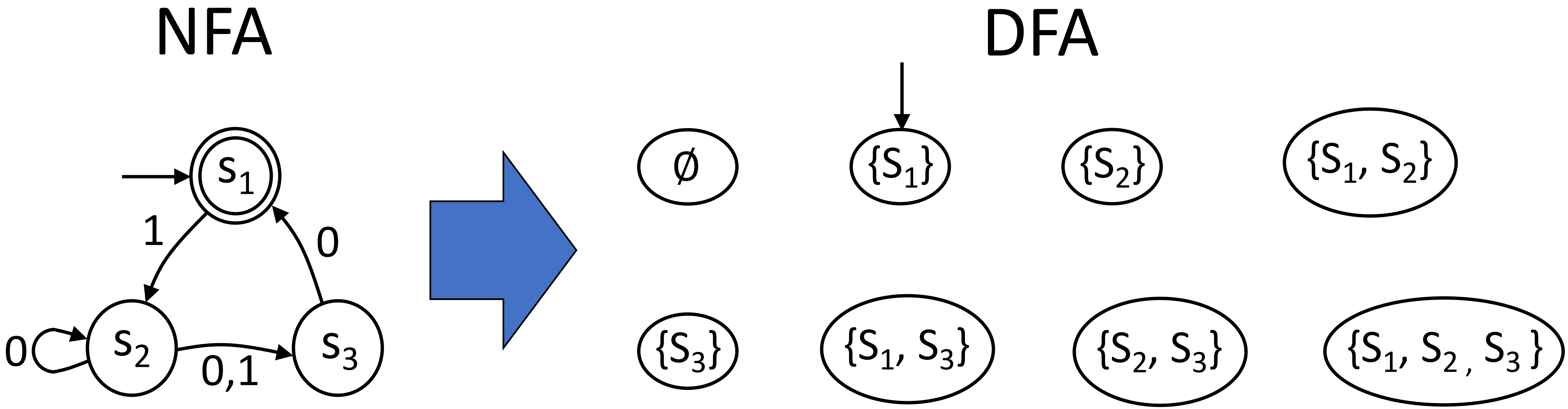
NFA DFA



DFA states = Power set of NFA states.

Start state = ?

DFA vs NFA

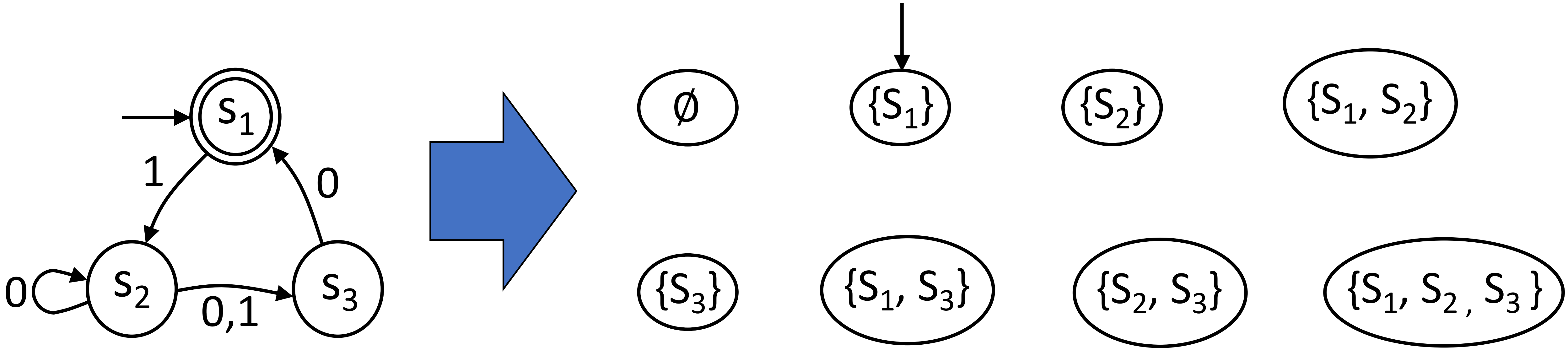


DFA states = Power set of NFA states.

Start state = NFA's start state.

DFA vs NFA

NFA DFA

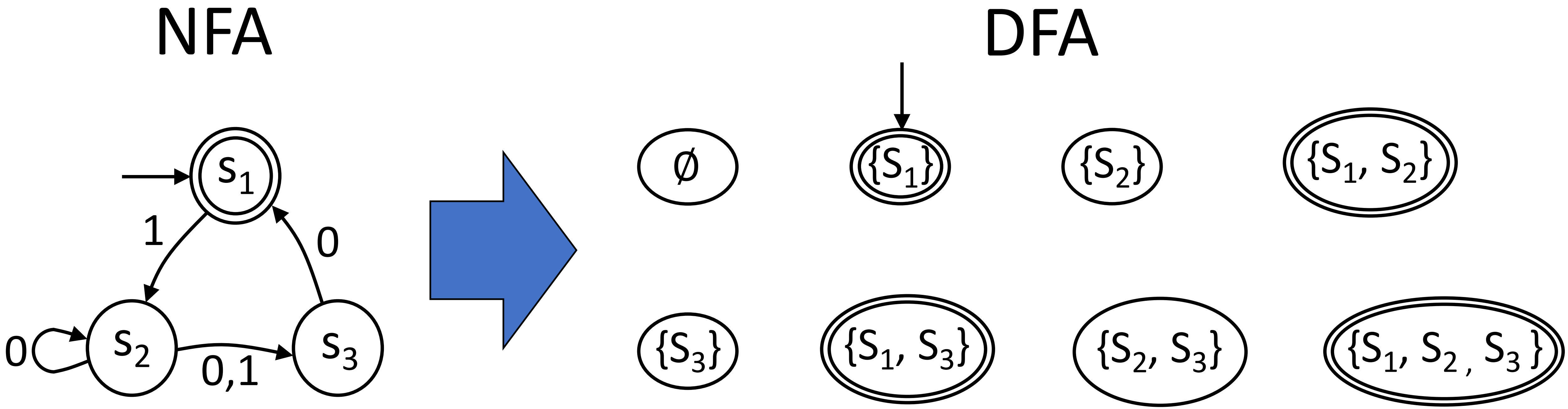


DFA states = Power set of NFA states.

Start state = NFA's start state.

Accept states = ?

DFA vs NFA

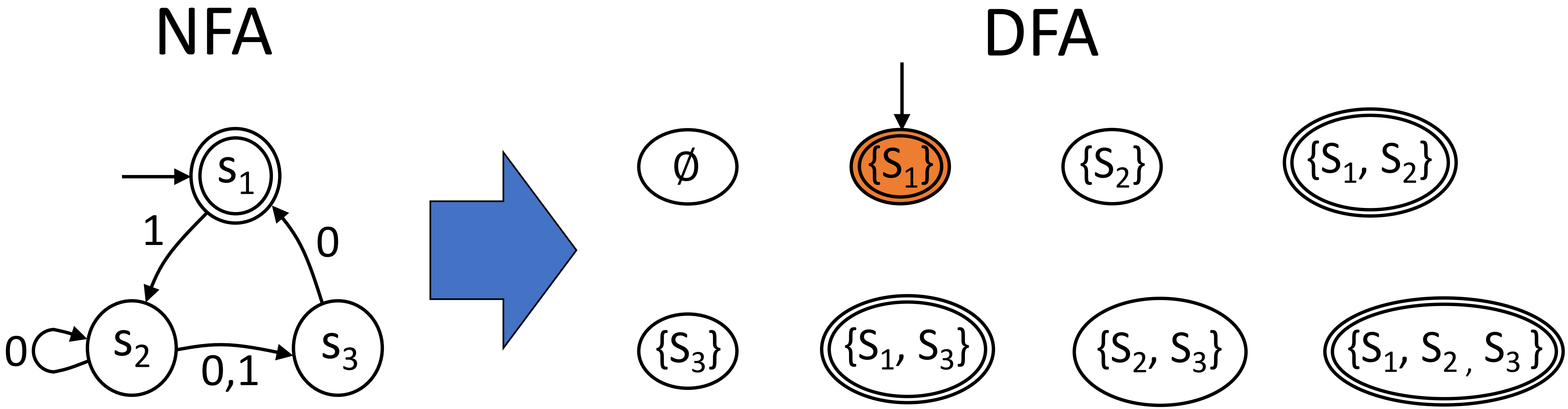


DFA states = Power set of NFA states.

Start state = NFA's start state.

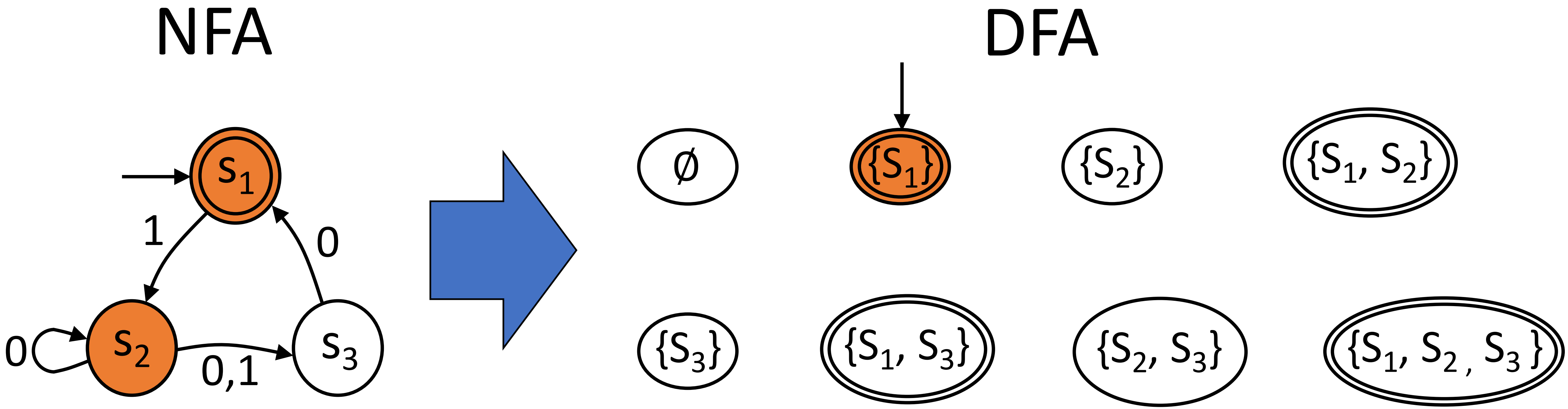
Accept states = Any state that includes accept state from NFA.

DFA vs NFA



Where should transition out of $\{S_1\}$ with character 1 go?

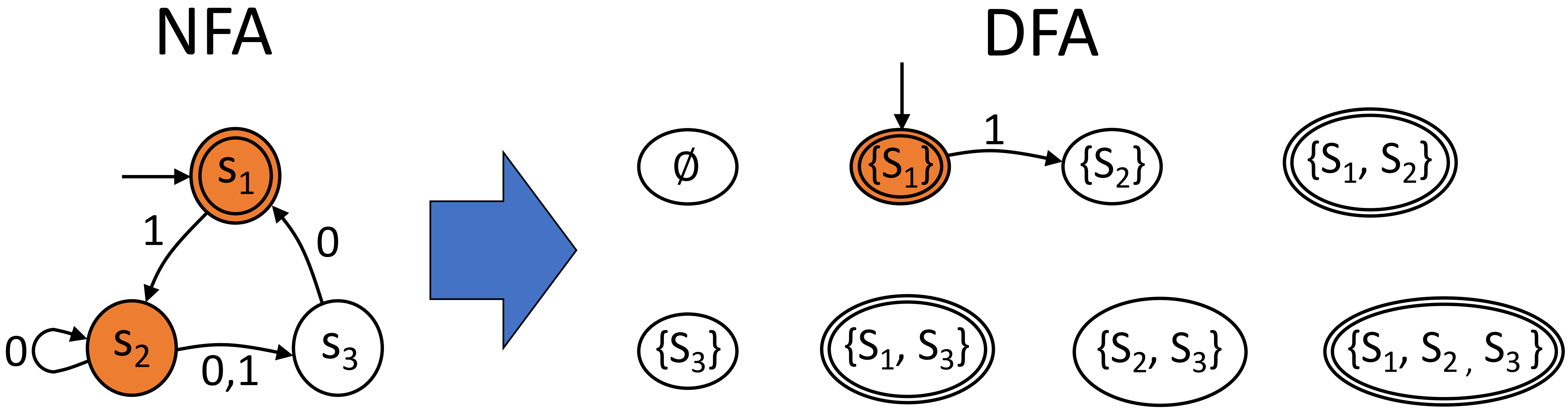
DFA vs NFA



Where should transition out of $\{S_1\}$ with character 1 go?

Wherever S_1 goes with 1 in the NFA.

DFA vs NFA

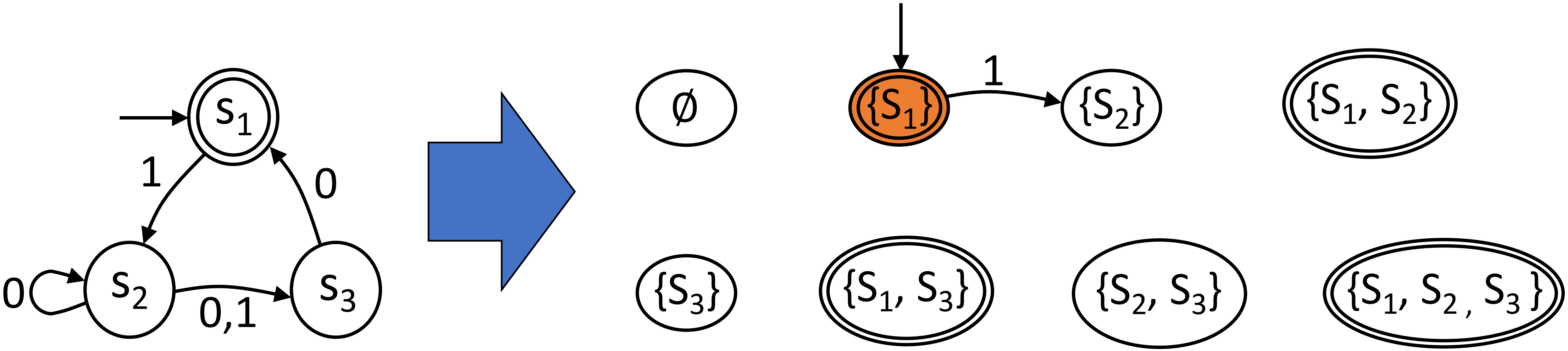


Where should transition out of $\{S_1\}$ with character 1 go?

Wherever S_1 goes with 1 in the NFA.

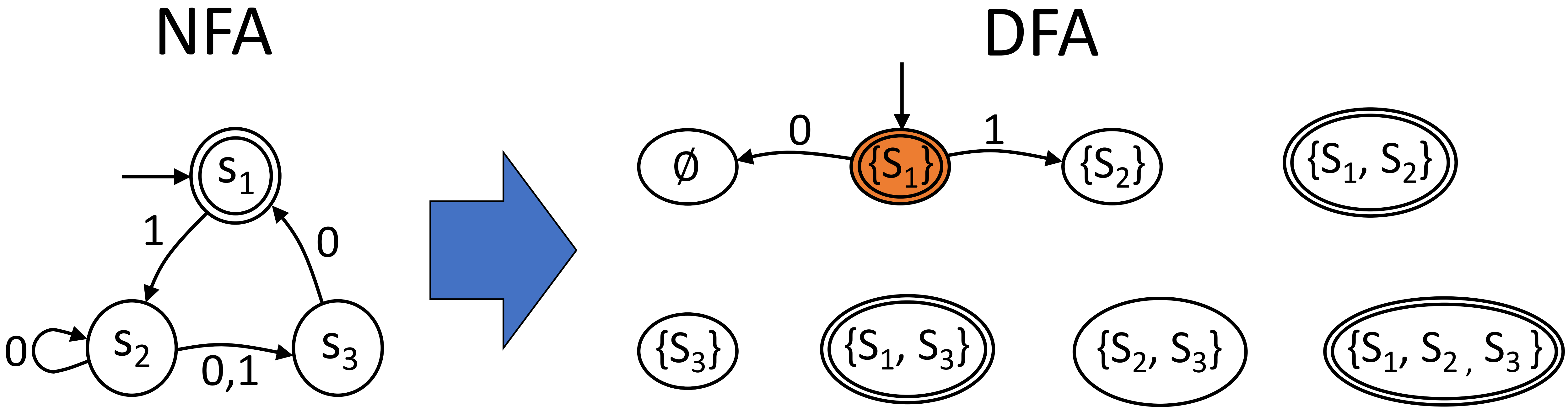
DFA vs NFA

NFA DFA



Where should transition out of $\{S_1\}$ with character 0 go?

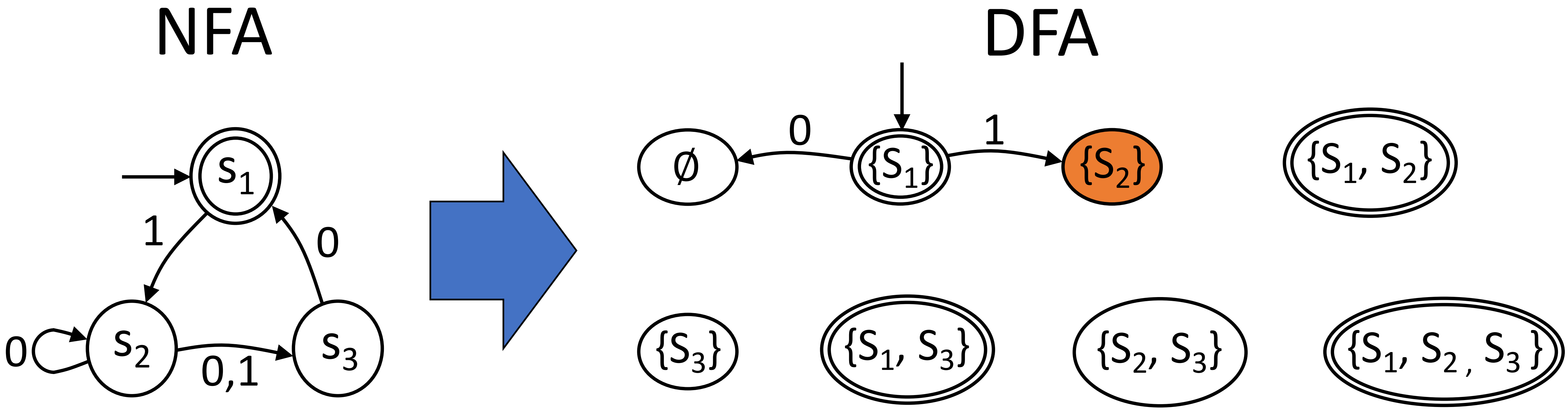
DFA vs NFA



Where should transition out of $\{S_1\}$ with character 0 go?

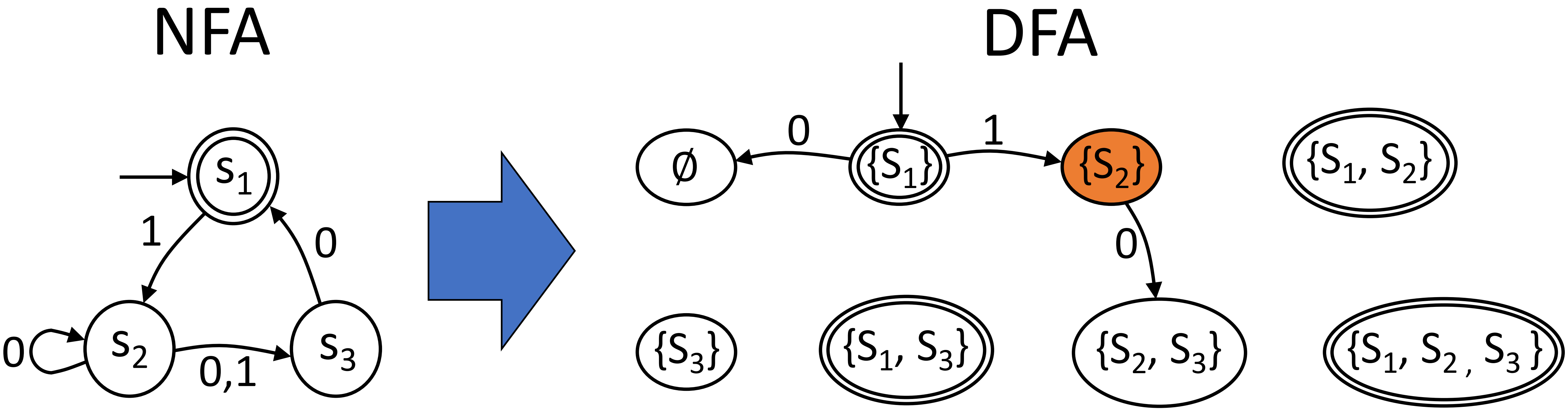
If transition is not handled by NFA, send it to Ø (junk state).

DFA vs NFA



Where should transition out of $\{S_2\}$ with character 0 go?

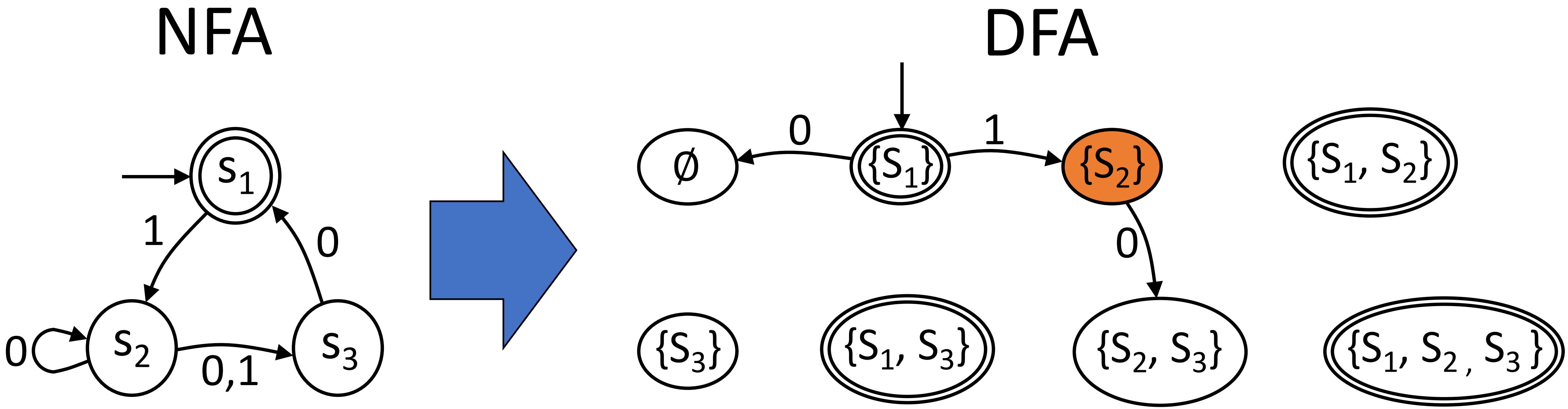
DFA vs NFA



Where should transition out of $\{S_2\}$ with character 0 go?

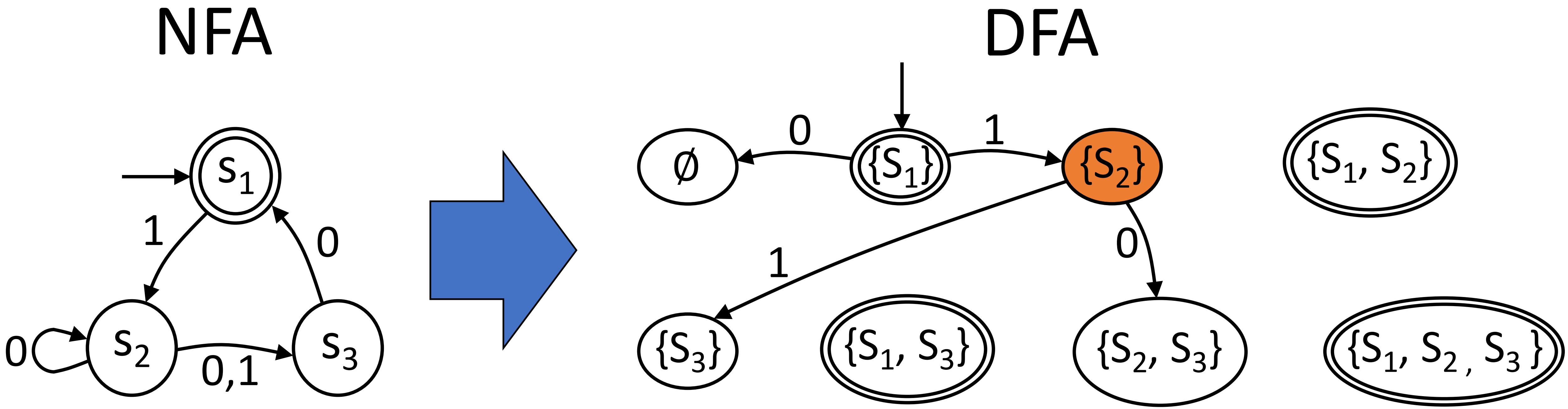
NFA could stay in S_2 or go to S_3, so $\{S_2, S_3\}$

DFA vs NFA



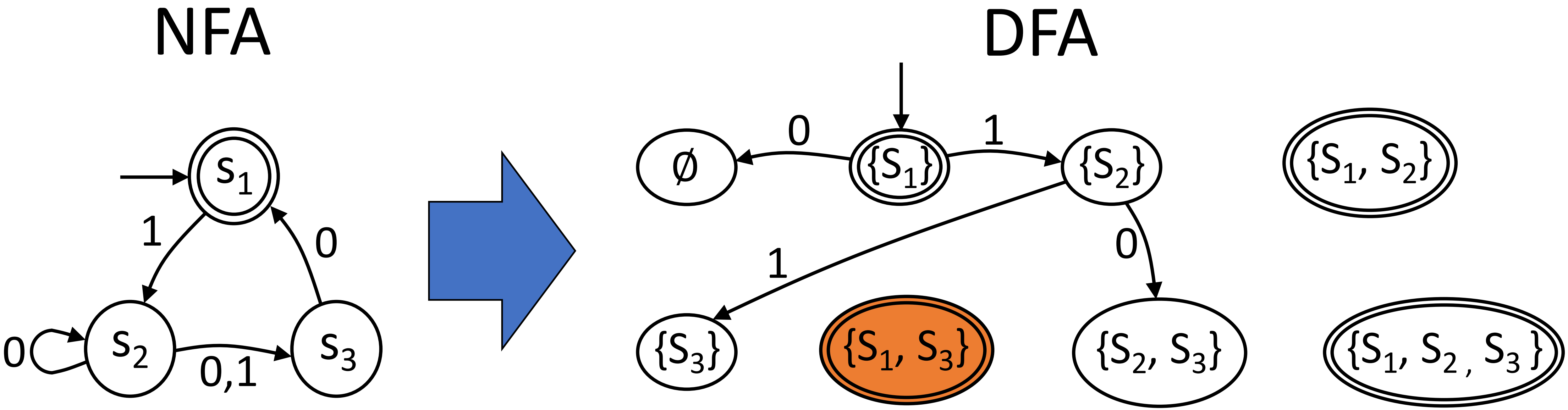
Where should transition out of $\{S_2\}$ with character 1 go?

DFA vs NFA



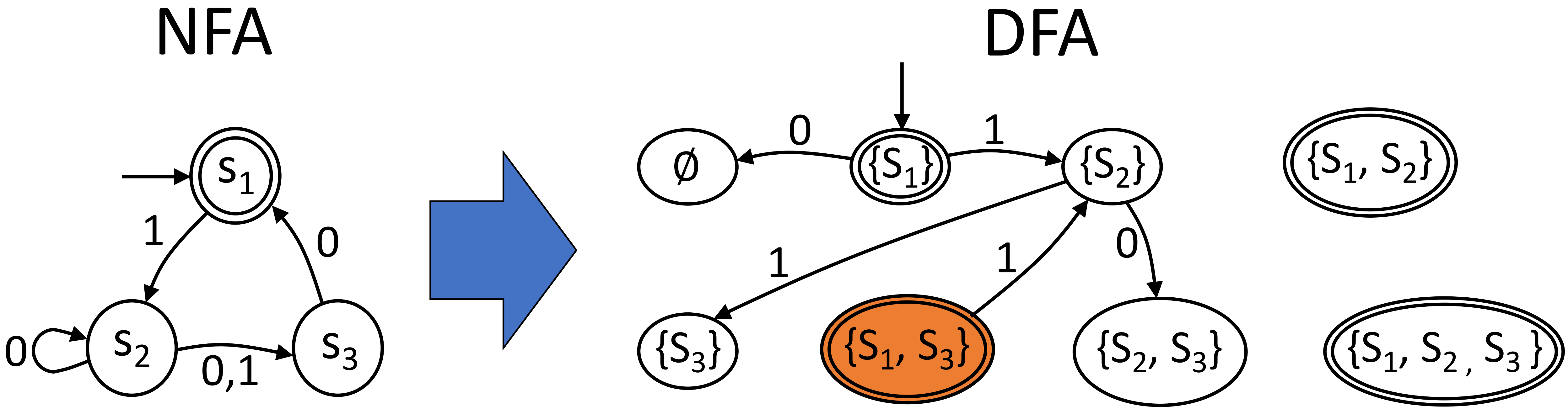
Where should transition out of {S_2} with character 1 go?

DFA vs NFA



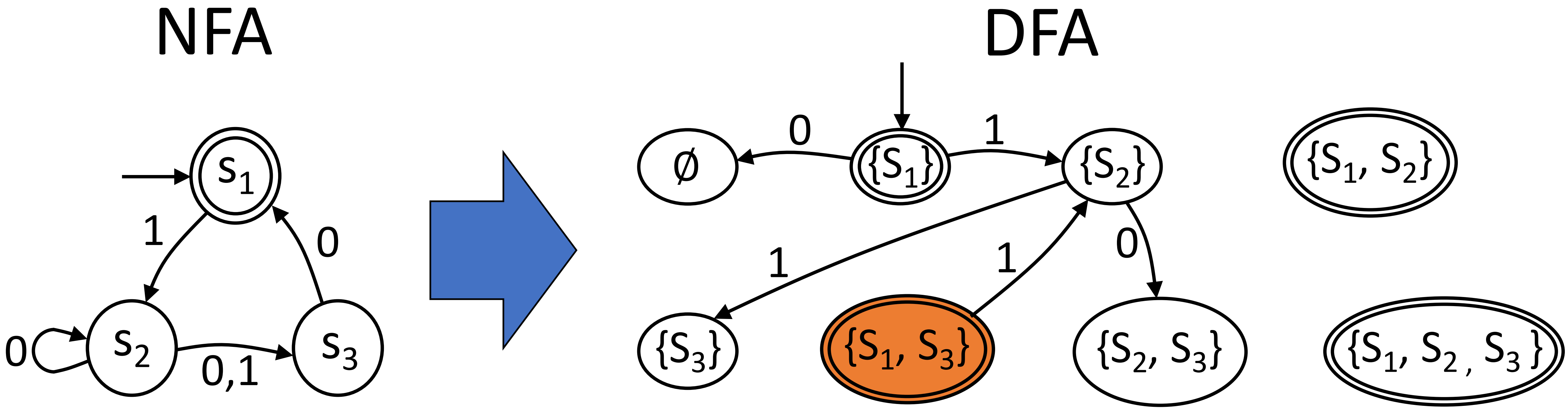
Where should transition out of $\{S_{1}, S_{3}\}$ with character 1 go?

DFA vs NFA



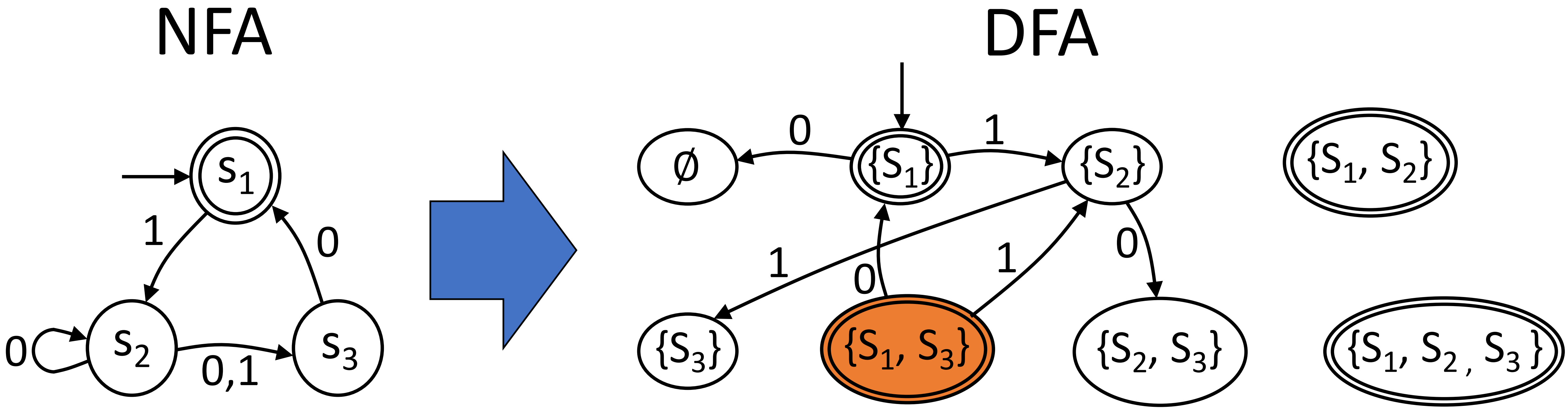
Where should transition out of $\{S_1, S_3\}$ with character 1 go?

DFA vs NFA



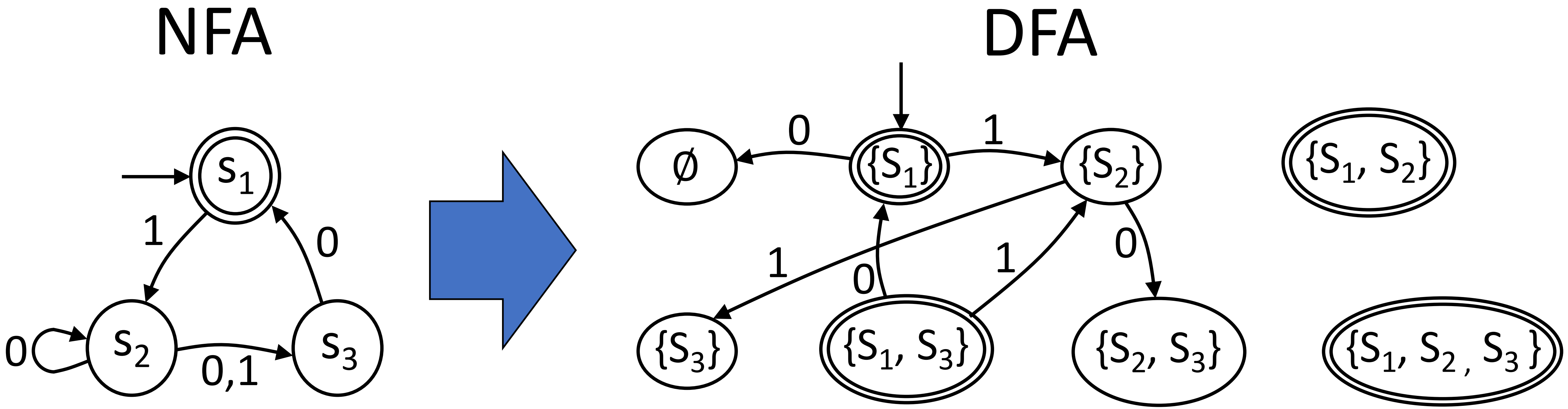
Where should transition out of $\{S_1, S_3\}$ with character 0 go?

DFA vs NFA



Where should transition out of $\{S_1, S_3\}$ with character 0 go?

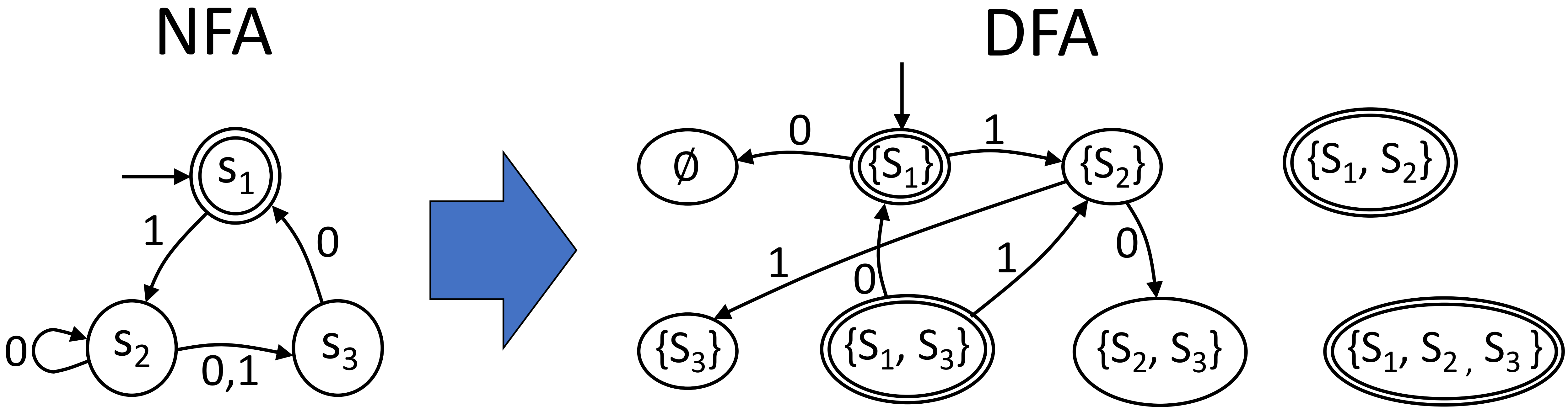
DFA vs NFA



Rule?

DFA state transitions to DFA state consisting of all states it's NFA states transition to.

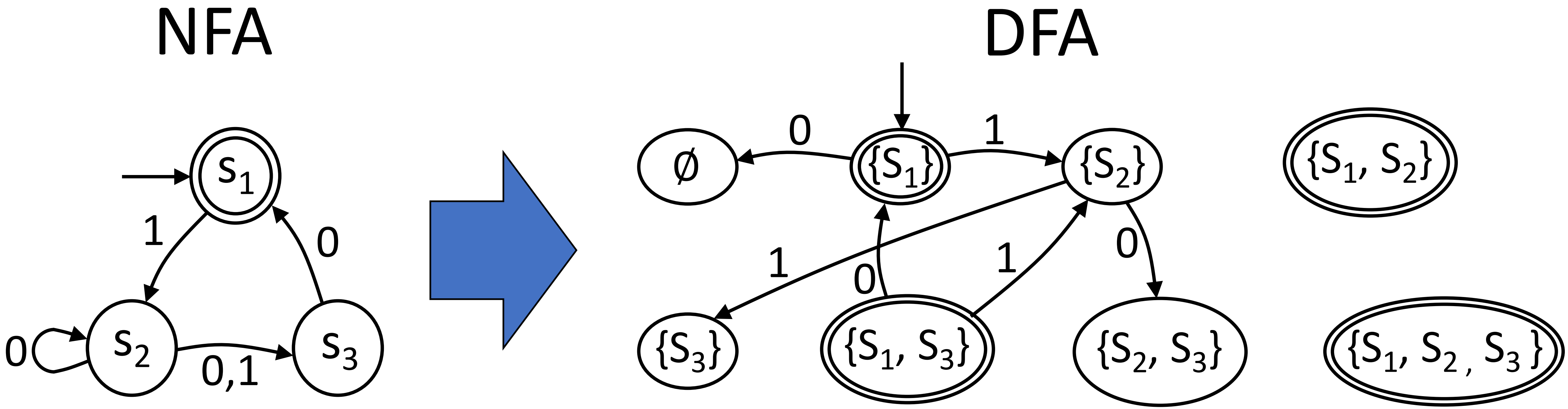
DFA vs NFA



Rule? For each DFA state R and $e \in \Sigma$,

$\text{transition}(R, e) = \{q \in \text{NFA}: q \in \text{transition}(r, e) \text{ for some } r \in R\}$

DFA vs NFA

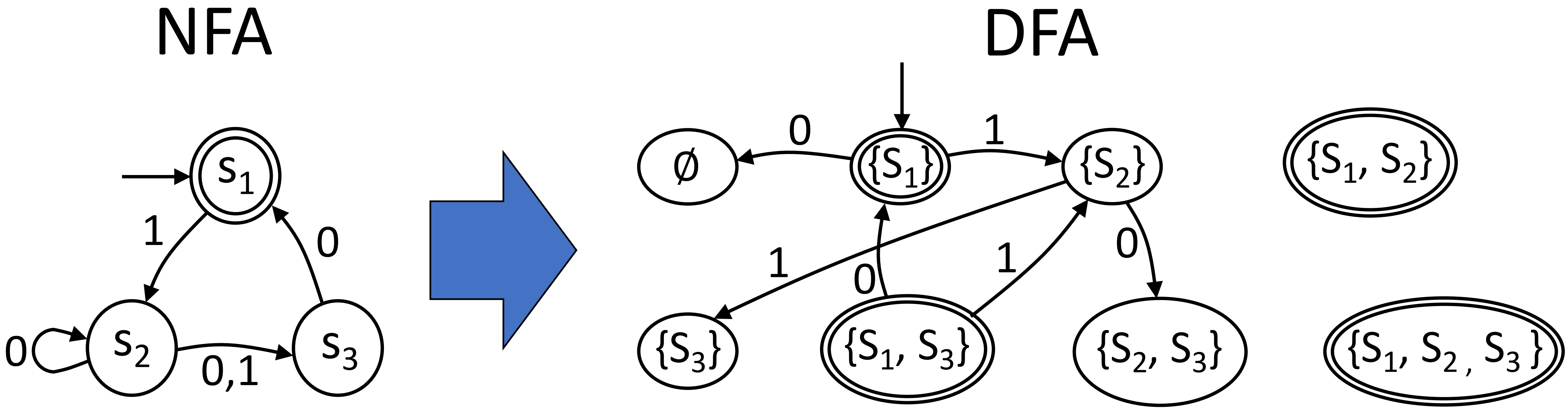


Rule? For each DFA state R and $e \in \Sigma$,

transition$(R, e) = \{q \in \text{NFA}: q \in \text{transition}(r, e)$ for some $r \in R\}$

transition$(\{S_2\}, 0) = \{S_2, S_3\}$

DFA vs NFA

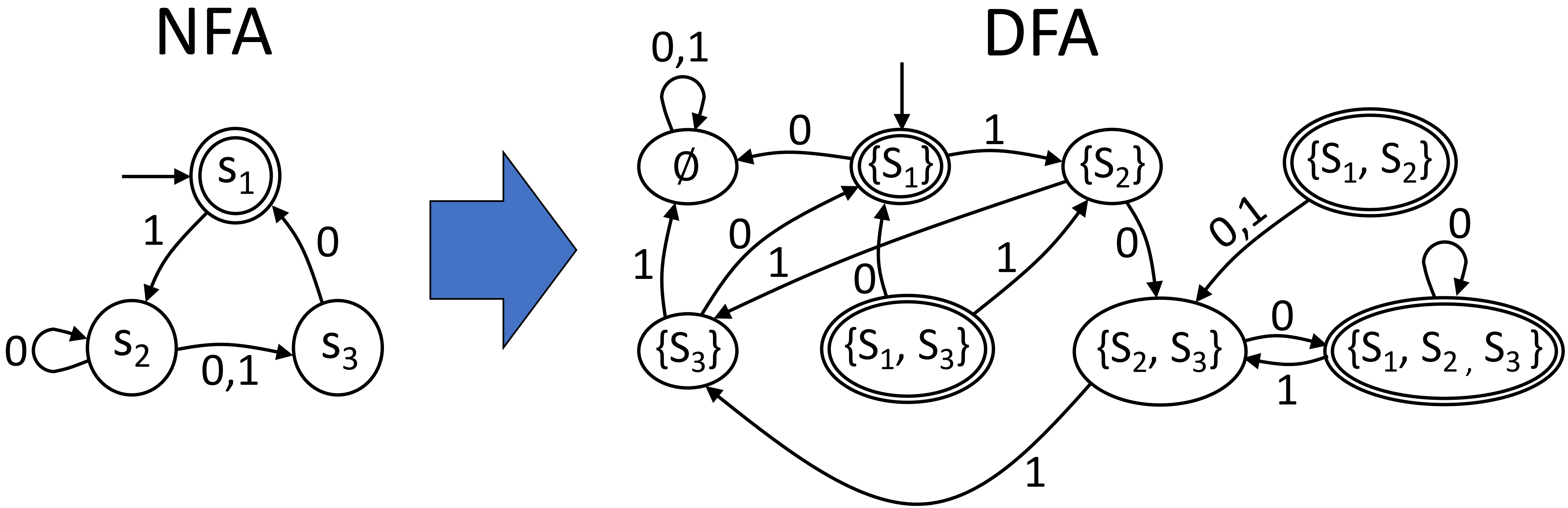


Rule? For each DFA state R and $e \in \Sigma$,

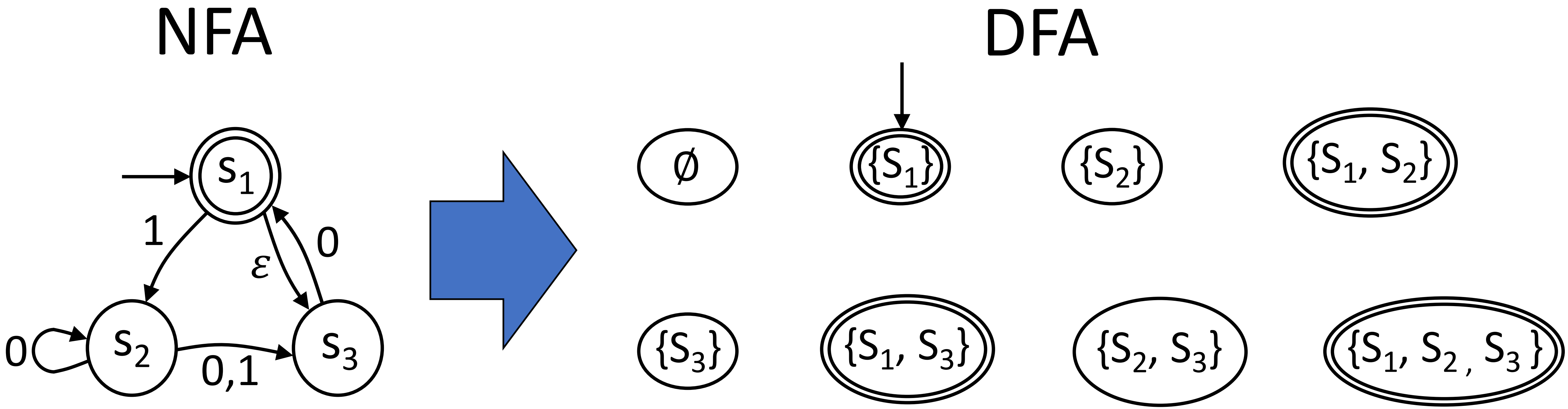
$\text{transition}(R, e) = \{q \in \text{NFA}\colon q \in \text{transition}(r, e) \text{ for some } r \in R\}$

$\text{transition}(\{S_1\}, 0) = \{\,\}$

DFA vs NFA


NFA
s1
s2
s3
1
0
0
0,1
DFA
0,1
∅
{S1}
{S2}
{S3}
{S1, S2}
{S1, S3}
{S2, S3}
{S1, S2 , S3 }
0
1
1
0
1
0
1
0
0,1
0
0
1
1

DFA vs NFA

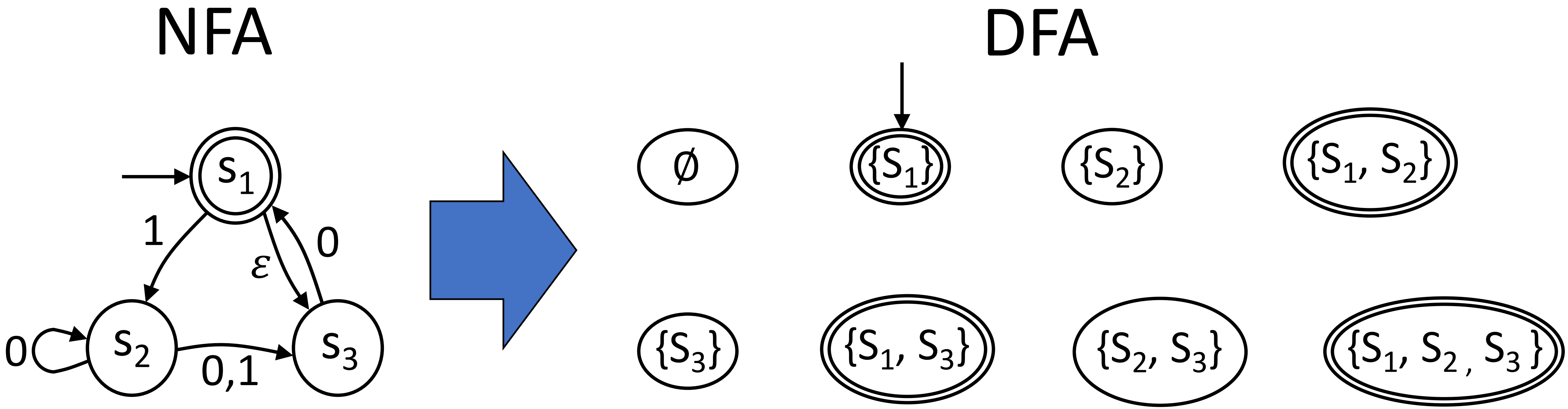


What about ε-transitions?

Define extension of DFA state R:

$E(R) = \{q \in \text{NFA}: q \text{ reachable from } r \in R \text{ with } \geq 0 \text{ ε–transitions}\}$

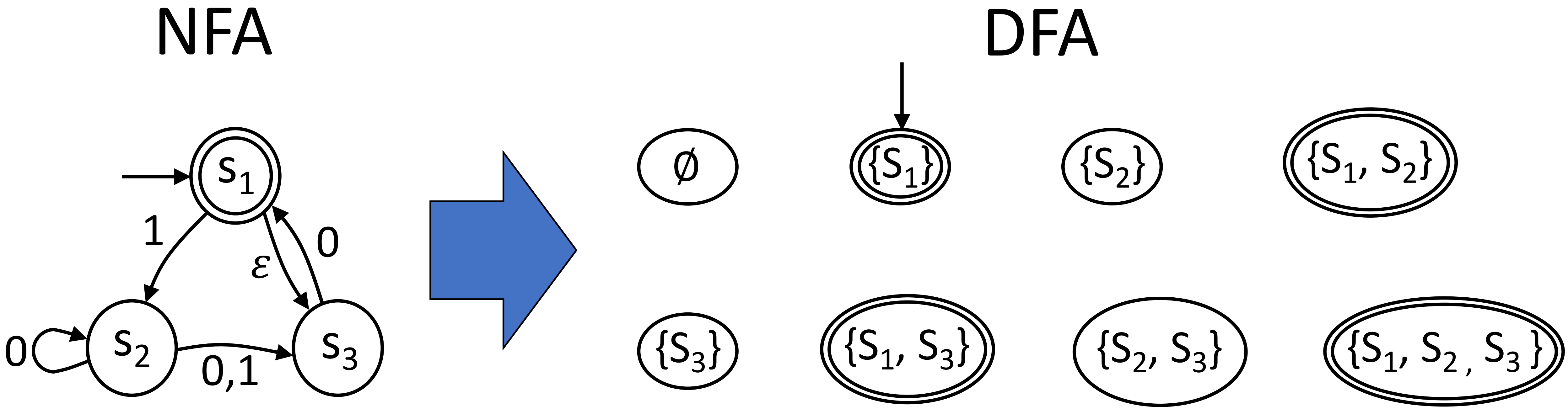
DFA vs NFA



$E(R) = \{q \in \text{NFA}: q \text{ reachable from } r \in R \text{ with } \geq 0\ \varepsilon\text{–transitions}\}$

Example: $E(\{S_2, S_3\}) = ?$

DFA vs NFA



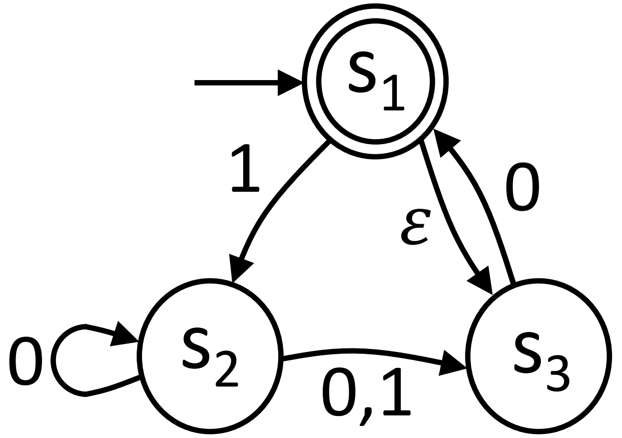
$E(R) = \{q \in \text{NFA}: q \text{ reachable from } r \in R \text{ with } \geq 0 \text{ ε–transitions}\}$

Example: $E(\{S_2, S_3\}) = \{S_2, S_3\}$

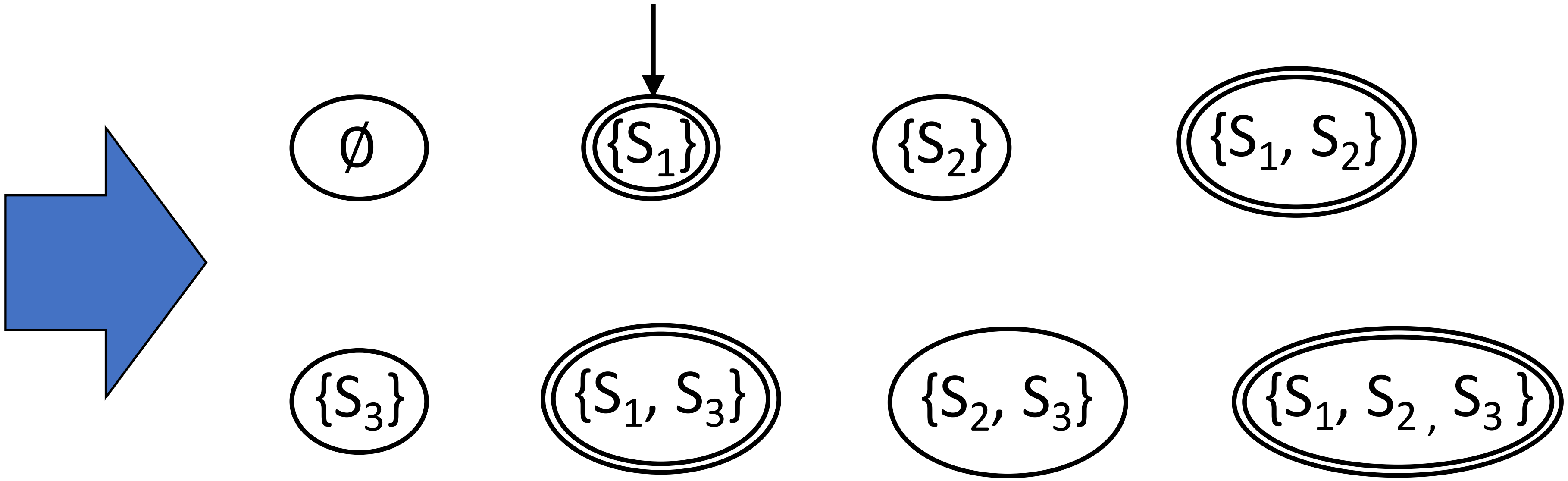
$E(\{S_1, S_2\}) = ?$

DFA vs NFA

NFA



DFA



$E(R) = \{q \in \text{NFA}: q$ reachable from $r \in R$ with ≥ 0 ε–transitions}

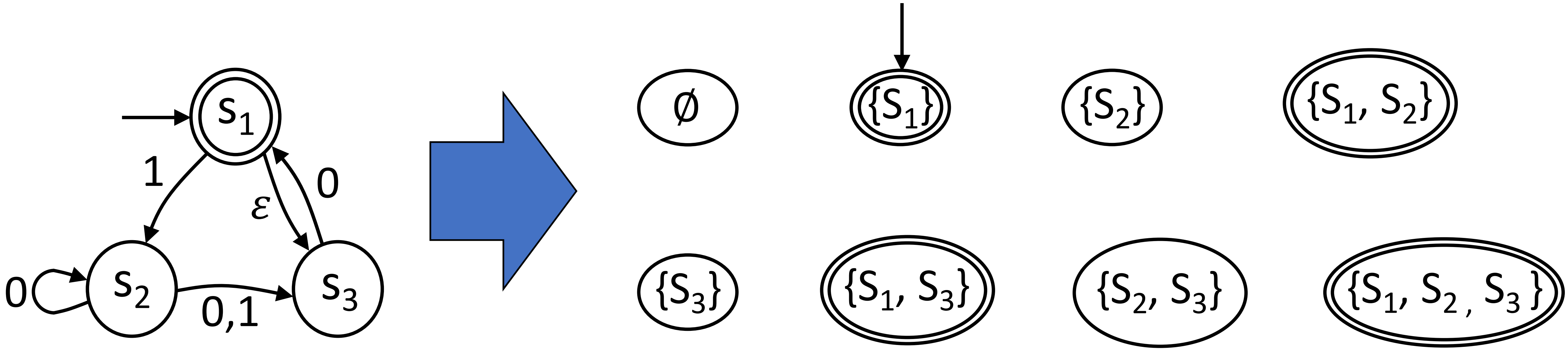
Example: $E(\{S_2, S_3\}) = \{S_2, S_3\}$

$E(\{S_1, S_2\}) = \{S_1, S_2, S_3\}$

DFA vs NFA

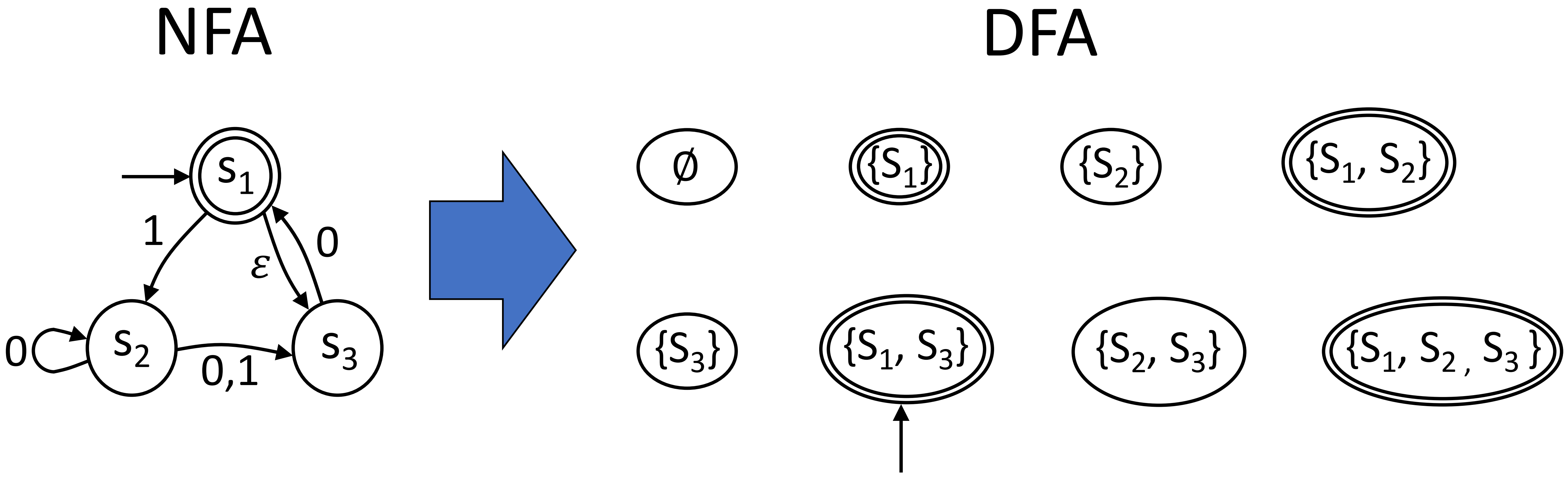
NFA

DFA



Make start state = ?

DFA vs NFA

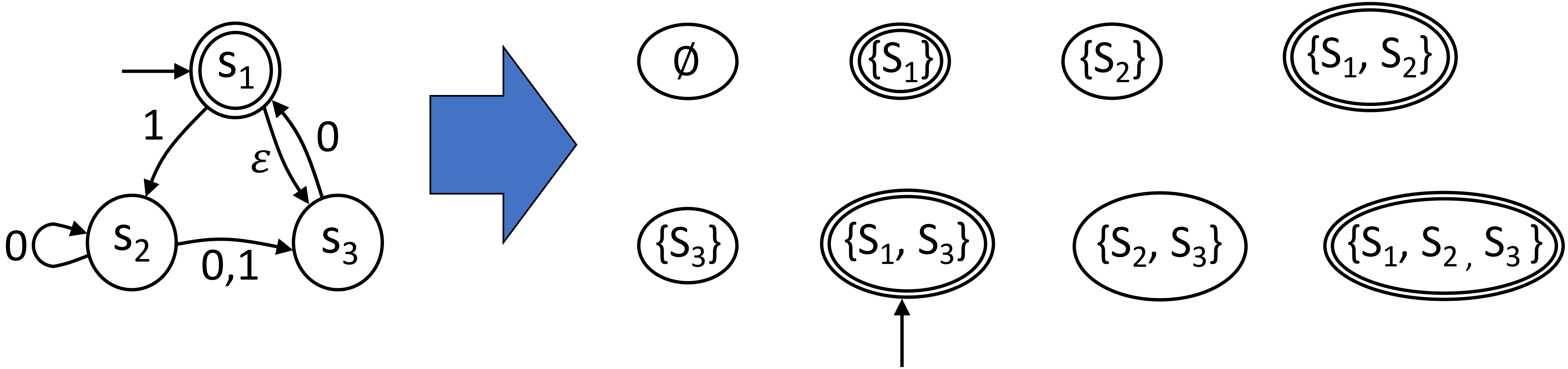


Make start state $= E(\{S_1\}) = \{S_1, S_3\}$

DFA vs NFA

NFA

DFA



$E\big(\text{transition}(r, e)\big)$

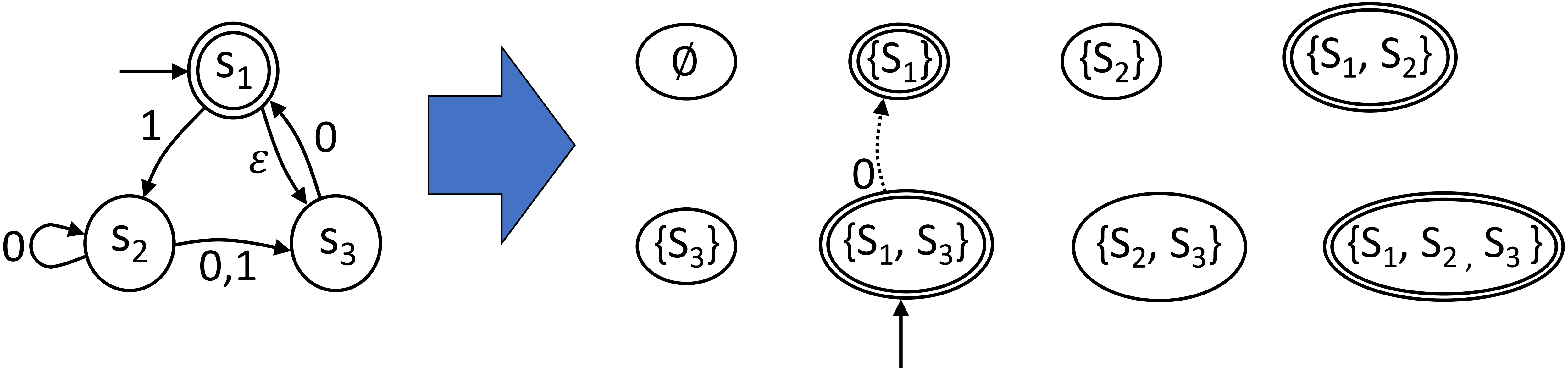
Make transitions:

$\text{transition}(R, e) = \{q \in \text{NFA}: q \in \cancel{\text{transition}(r, e)} \text{ for some } r \in R\}$

DFA vs NFA

NFA

DFA

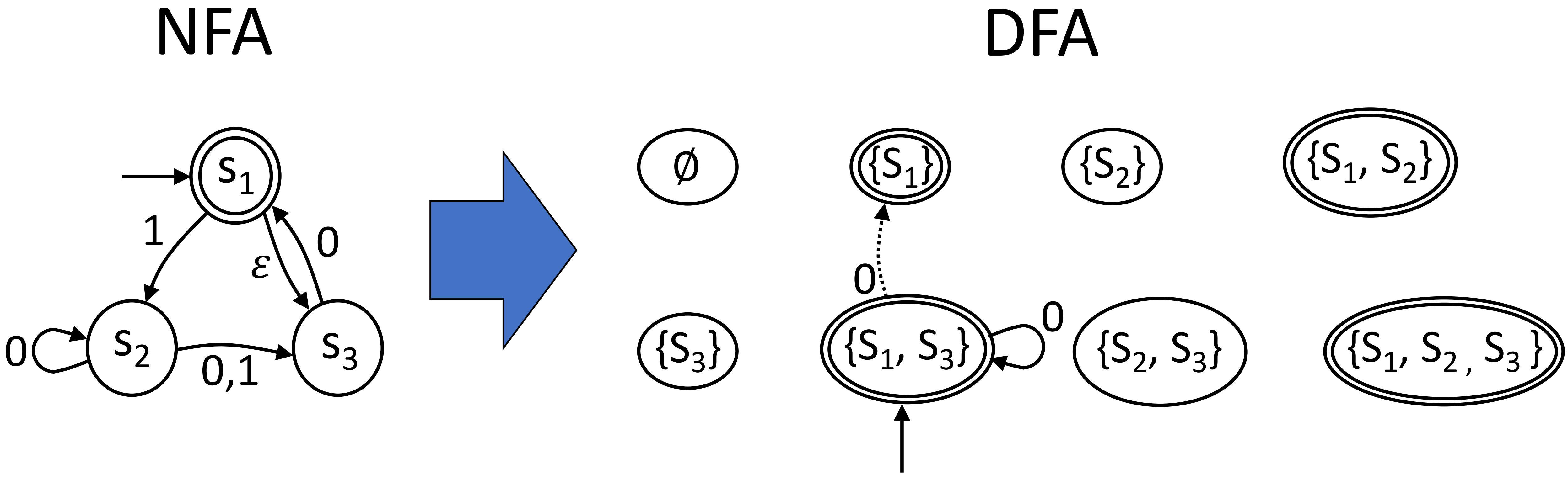


$E\big(\text{transition}(r, e)\big)$

Make transitions:

$\text{transition}(R, e) = \{q \in \text{NFA}: q \in \cancel{\text{transition}(r, e)} \text{ for some } r \in R\}$

DFA vs NFA



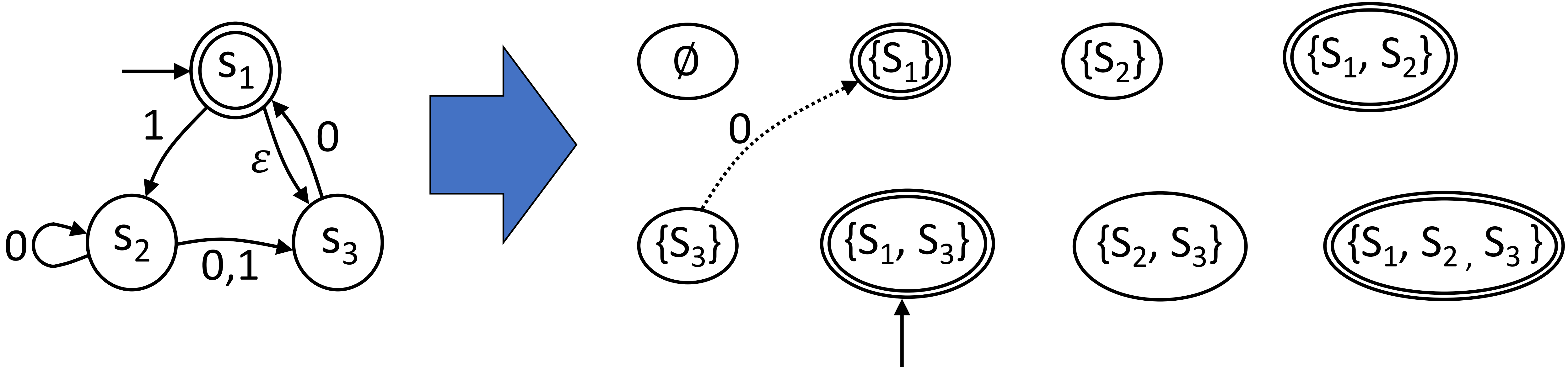
$E\big(\text{transition}(r, e)\big)$

Make transitions:

$\text{transition}(R, e) = \{q \in \text{NFA}: q \in \cancel{\text{transition}(r, e)} \text{ for some } r \in R\}$

DFA vs NFA

NFA　　　　　　　　　　DFA

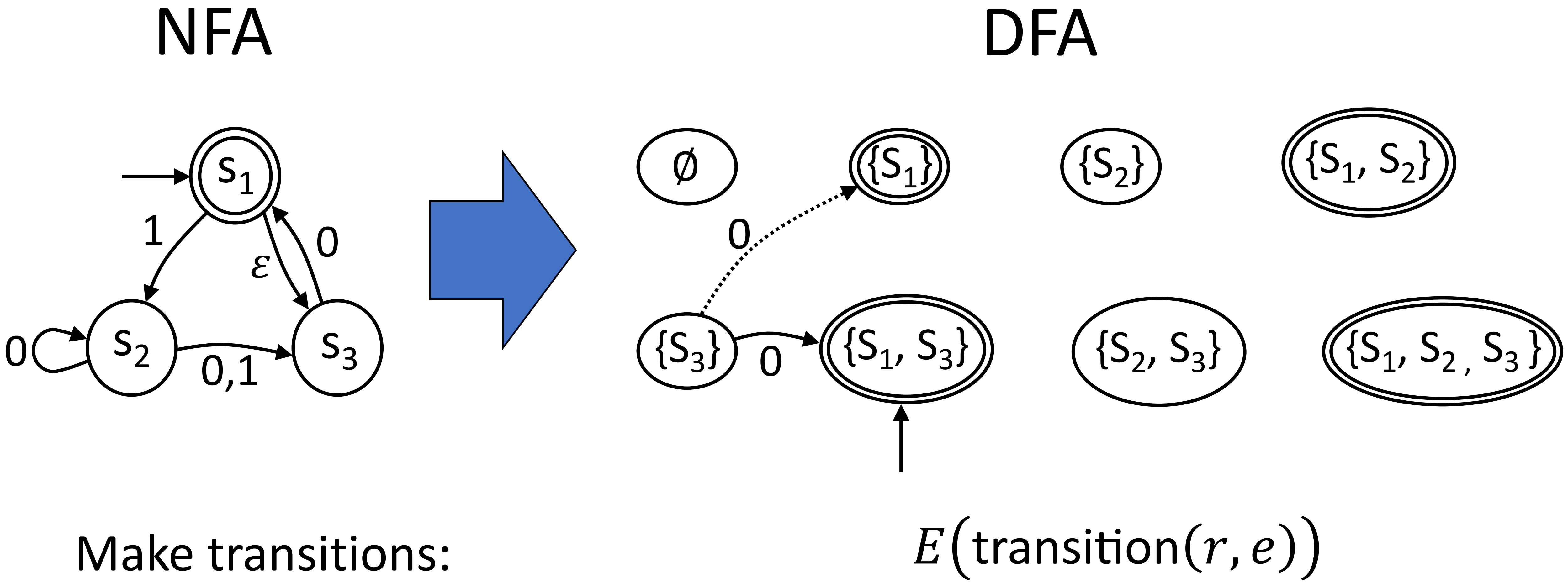


$E\big(\text{transition}(r, e)\big)$

Make transitions:

$\text{transition}(R, e) = \{q \in \text{NFA}: q \in \cancel{\text{transition}(r, e)} \text{ for some } r \in R\}$

DFA vs NFA

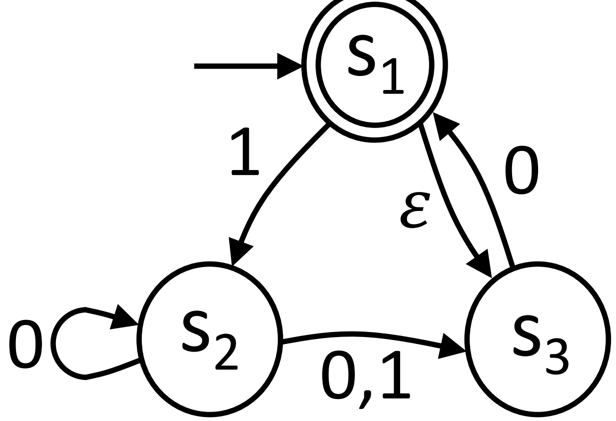


Make transitions:

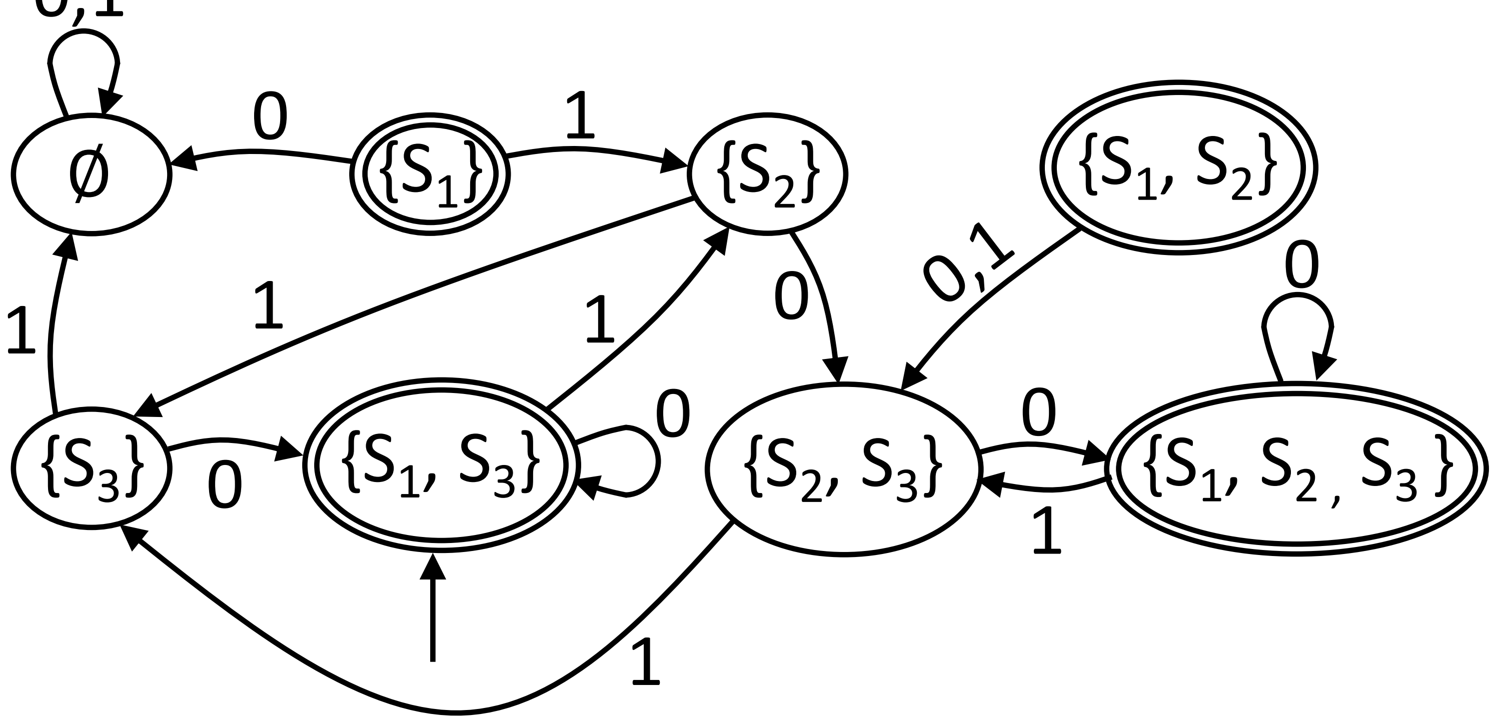
transition$(R, e) = \{q \in$ NFA: $q \in \cancel{\text{transition}(r, e)}$ for some $r \in R\}$

DFA vs NFA

NFA



DFA



Finite set of states?

Finite Alphabet?

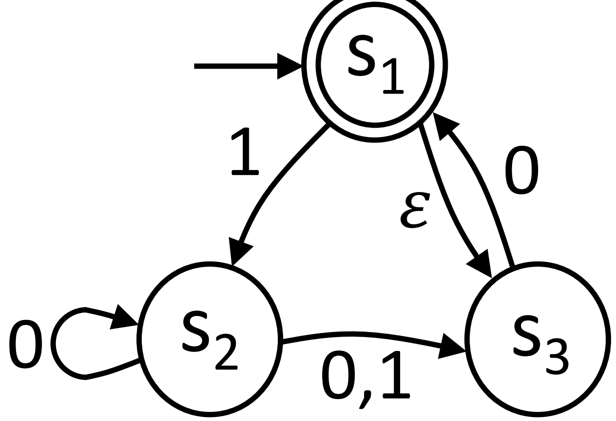
Transition function for every state/character pair?

Single start state?

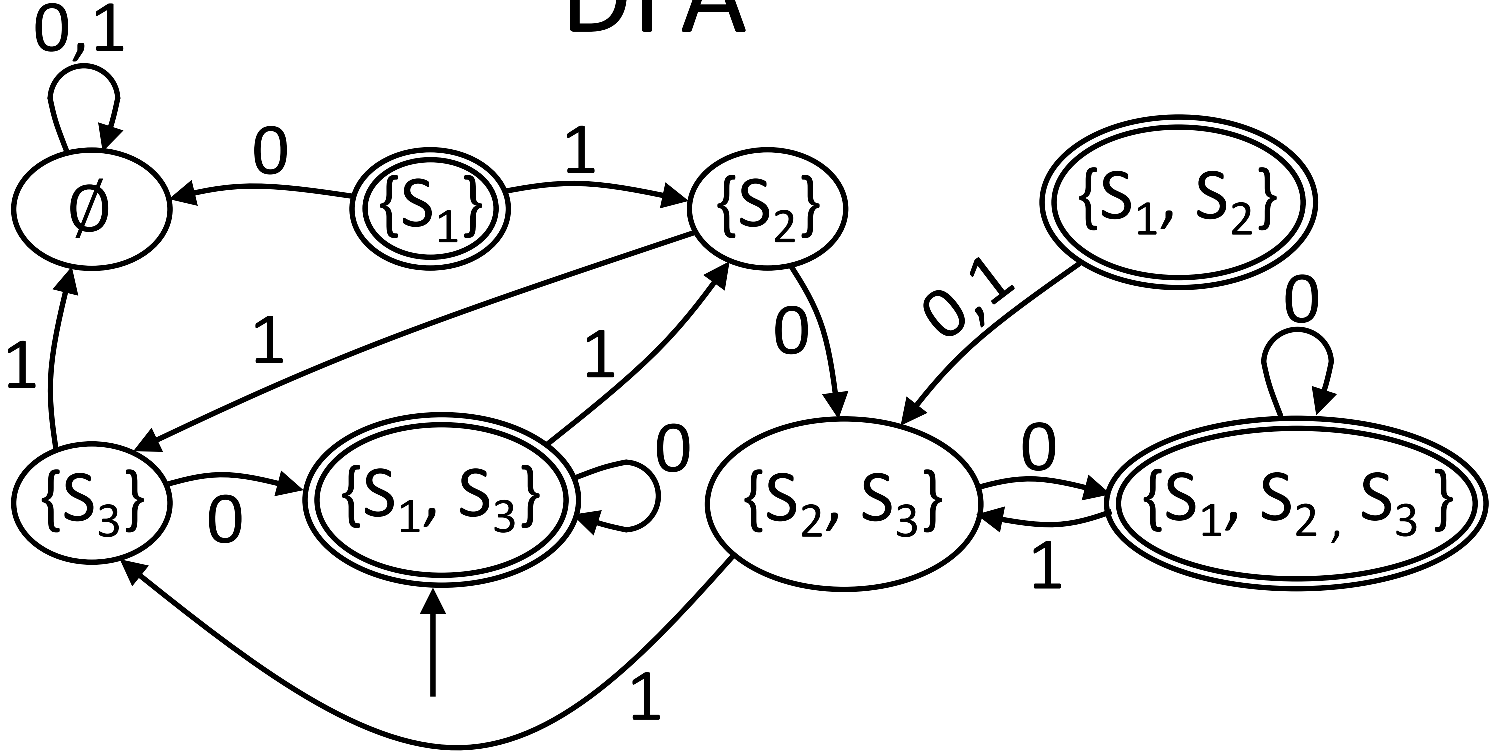
Set of accept states?

DFA vs NFA

NFA



DFA

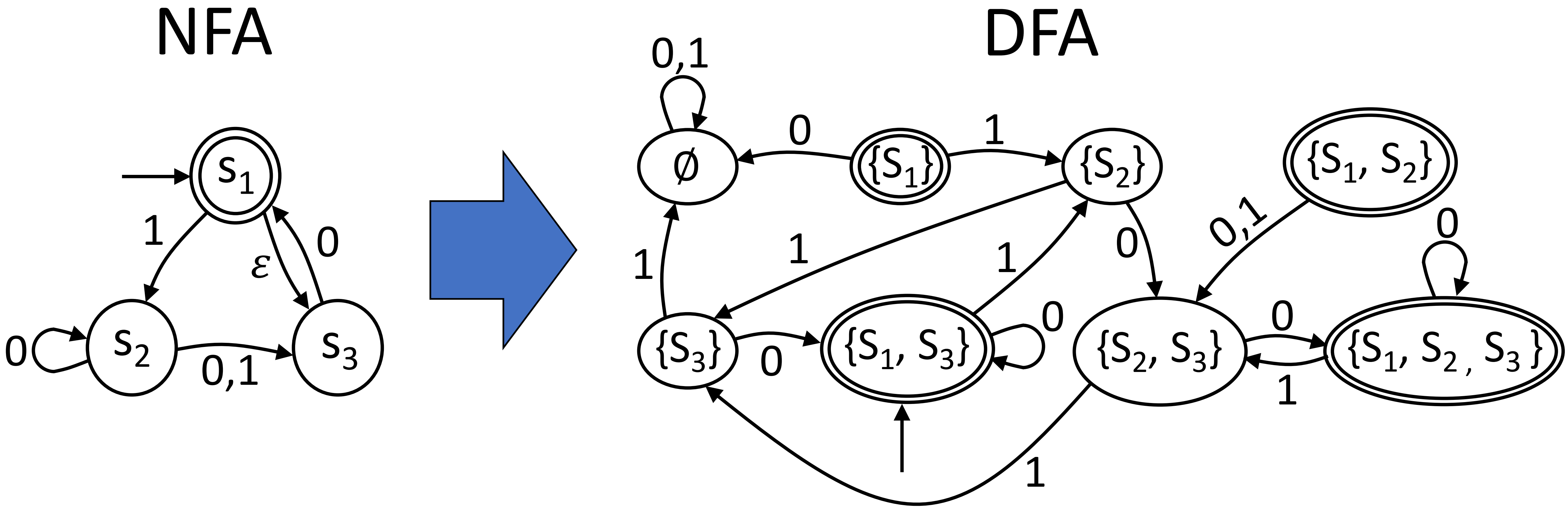


Finite set of states?   Yes   Finite Alphabet?

Transition function for every state/character pair?

Single start state?   Set of accept states?

DFA vs NFA



Finite set of states? Yes

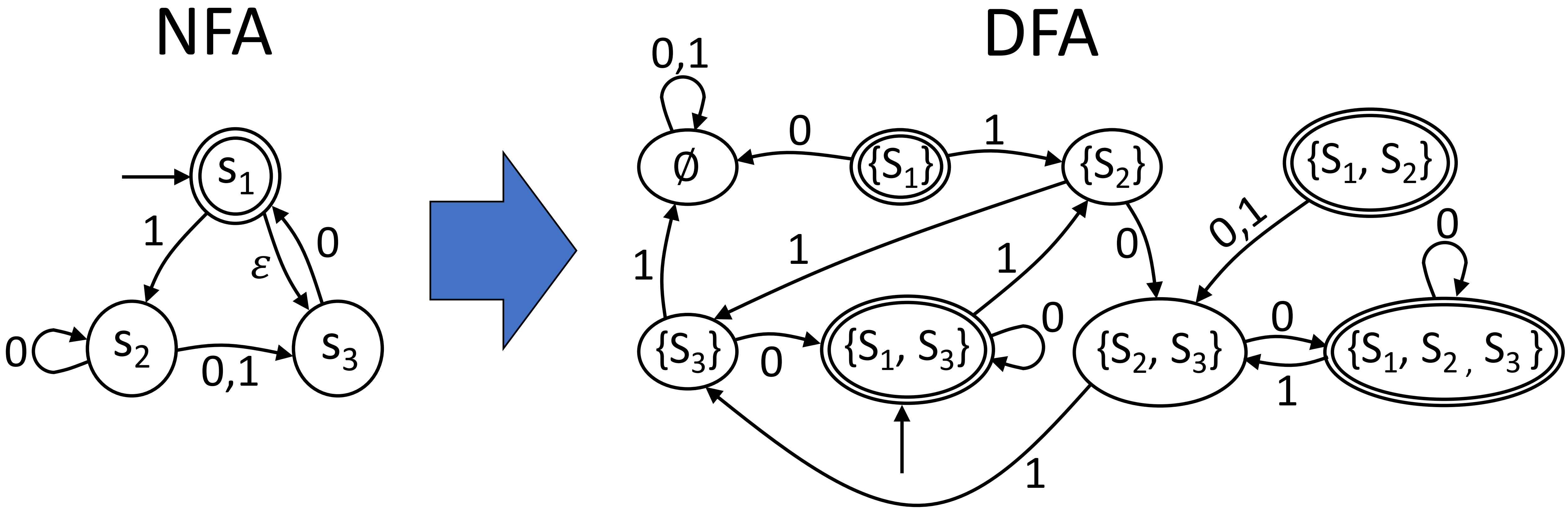
Finite Alphabet? Yes

Transition function for every state/character pair?

Single start state?

Set of accept states?

DFA vs NFA



Finite set of states? Yes

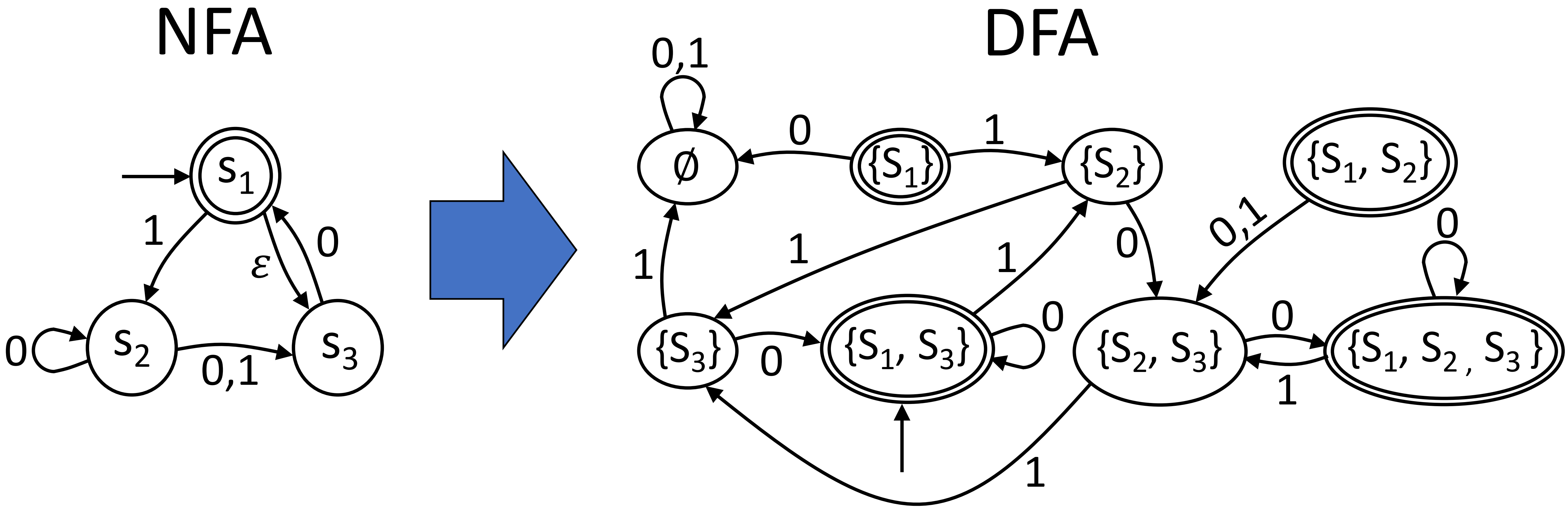
Finite Alphabet? Yes

Transition function for every state/character pair? Yes

Single start state?

Set of accept states?

DFA vs NFA

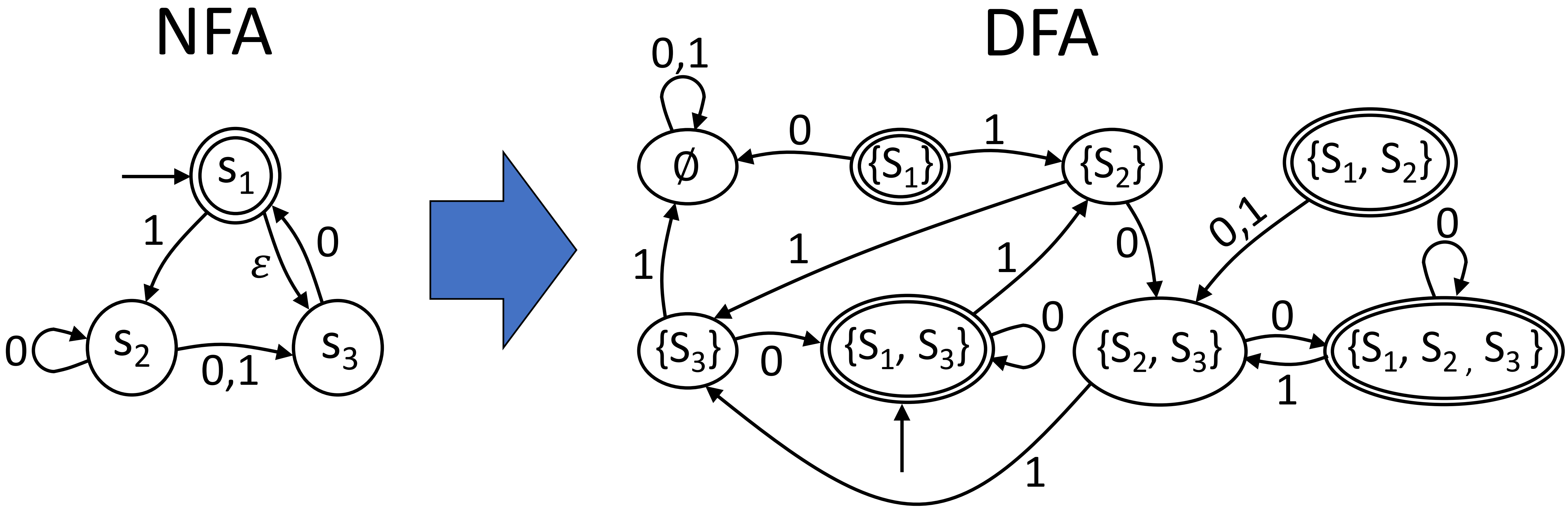


Finite set of states? Yes Finite Alphabet? Yes

Transition function for every state/character pair? Yes

Single start state? Yes Set of accept states?

DFA vs NFA



Finite set of states? Yes

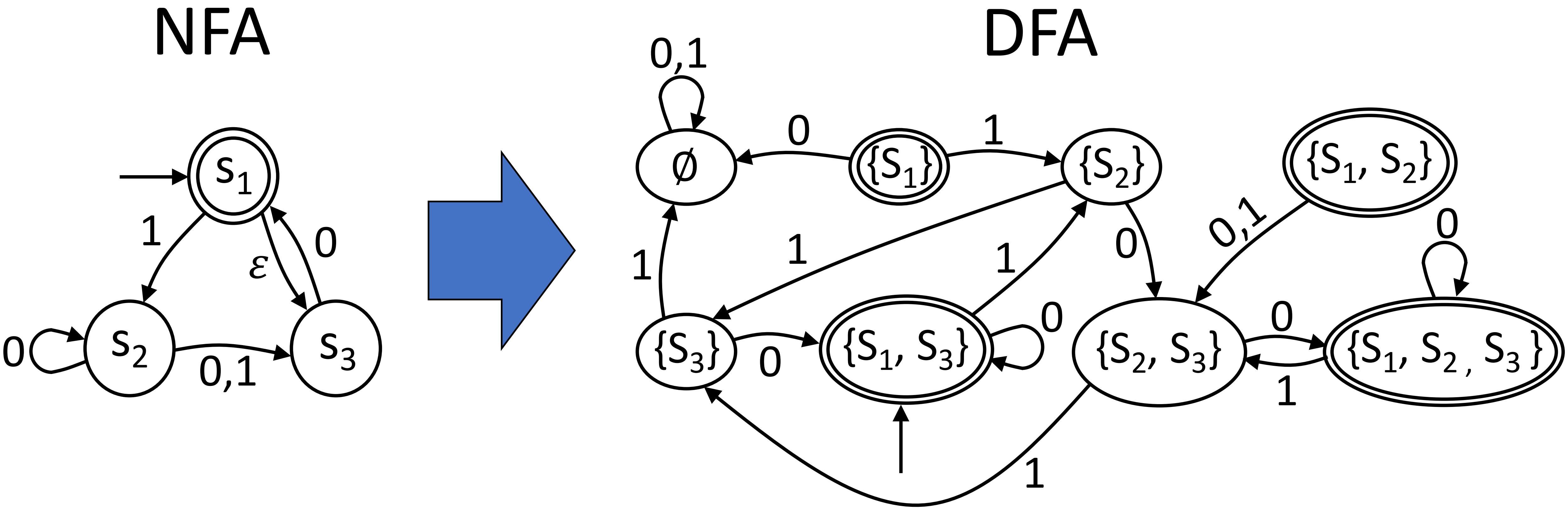
Finite Alphabet? Yes

Transition function for every state/character pair? Yes

Single start state? Yes

Set of accept states? Yes

DFA vs NFA



Finite set of s lphabet? Yes

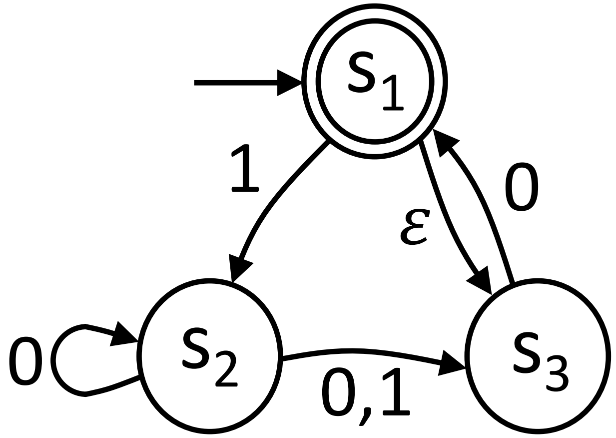
Transition fu cter pair? Yes

Single start st ccept states? Yes

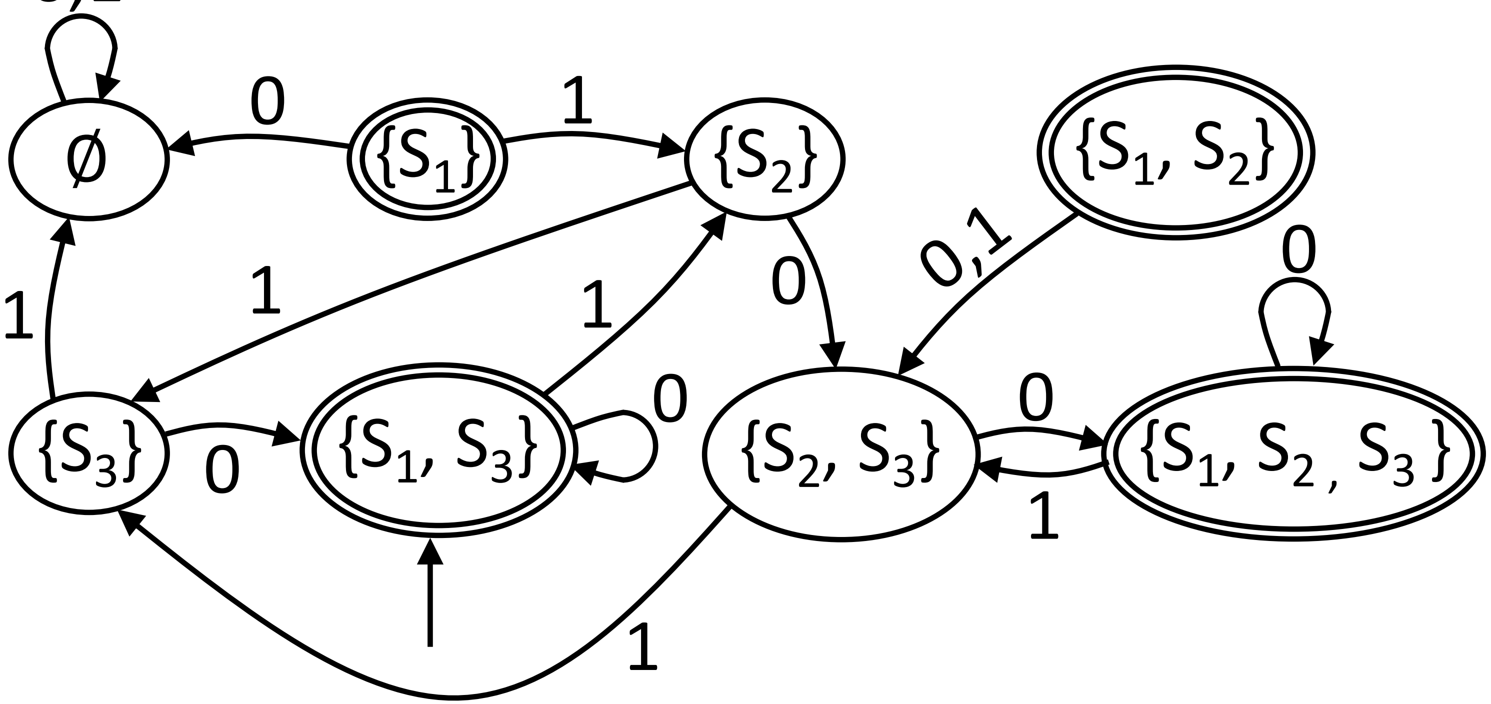
It's a DFA

DFA vs NFA

NFA

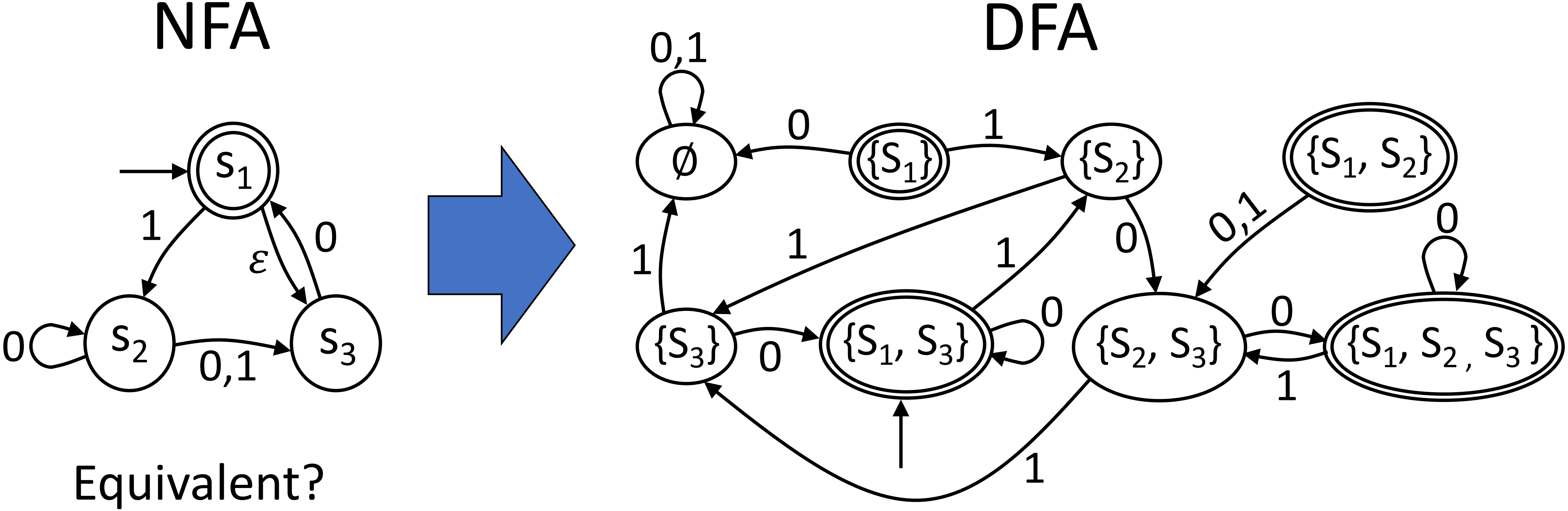


DFA



Equivalent?

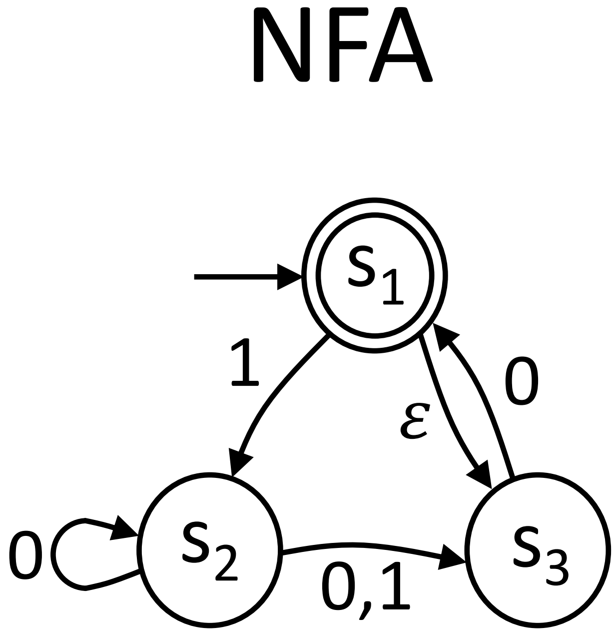
DFA vs NFA

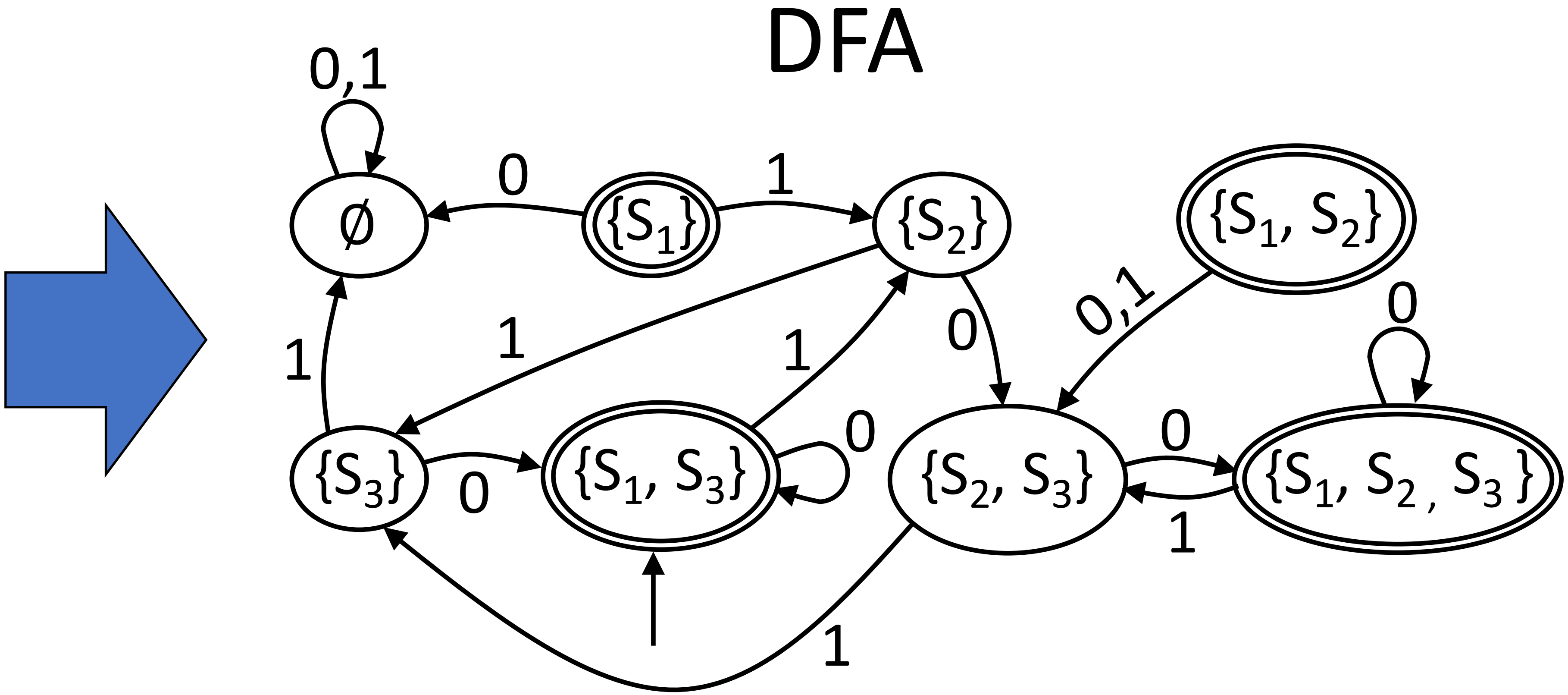


Equivalent?

Suppose w accepted by NFA.

DFA vs NFA

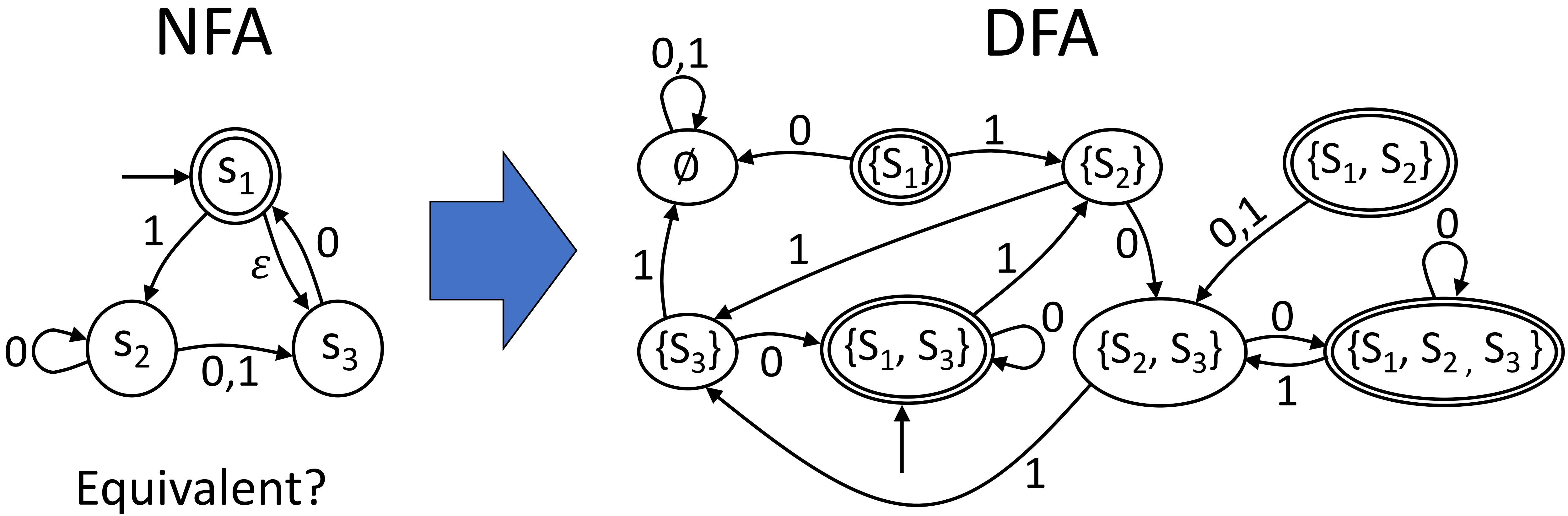




Equivalent?

Suppose w accepted by NFA. At each step of its processing, DFA will be in state that corresponds to all possible NFA states.

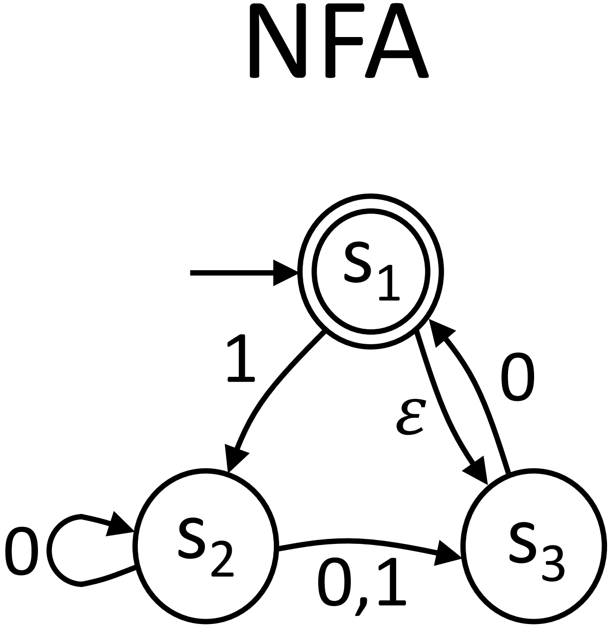
DFA vs NFA

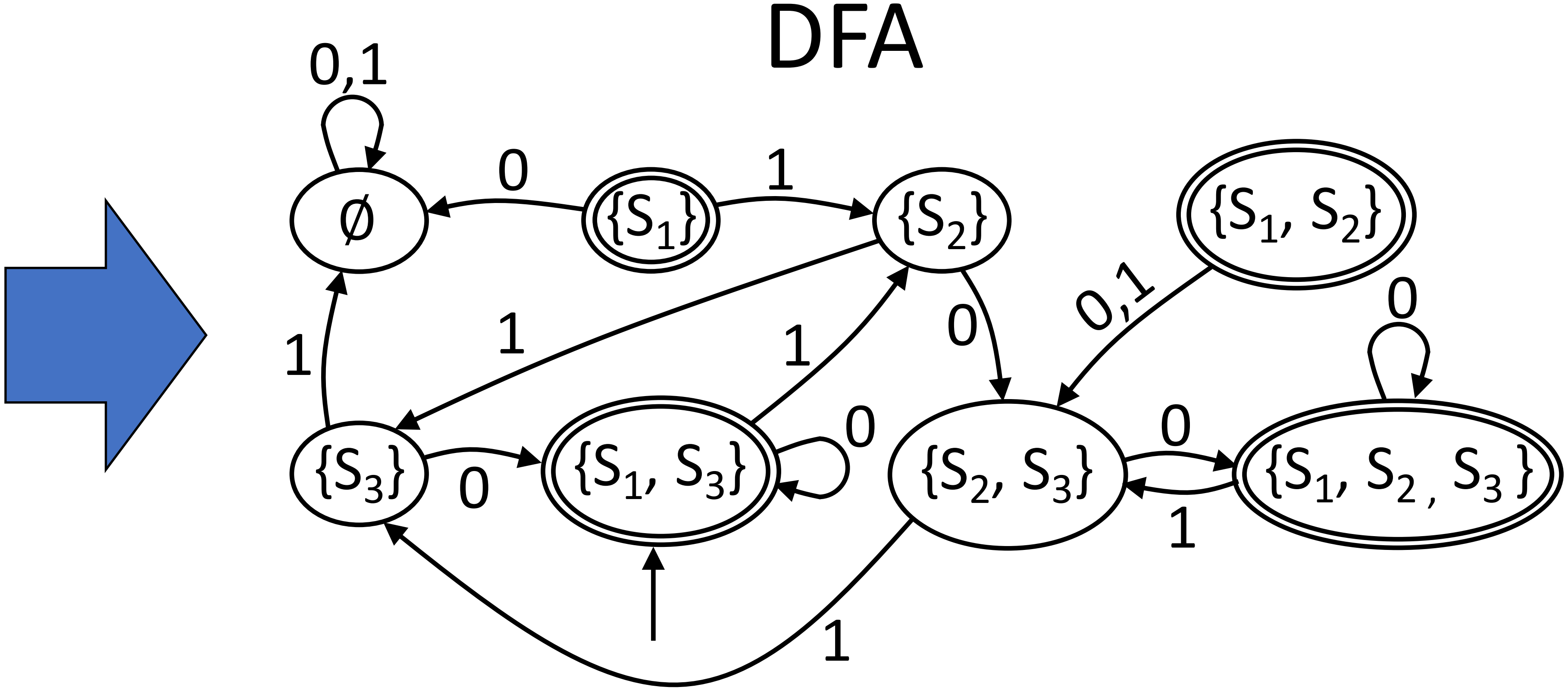


Equivalent?

Suppose w accepted by NFA. At each step of its processing, DFA will be in state that corresponds to all possible NFA states. If NFA ends on accept state, corresponding DFA state will accept too.

DFA vs NFA

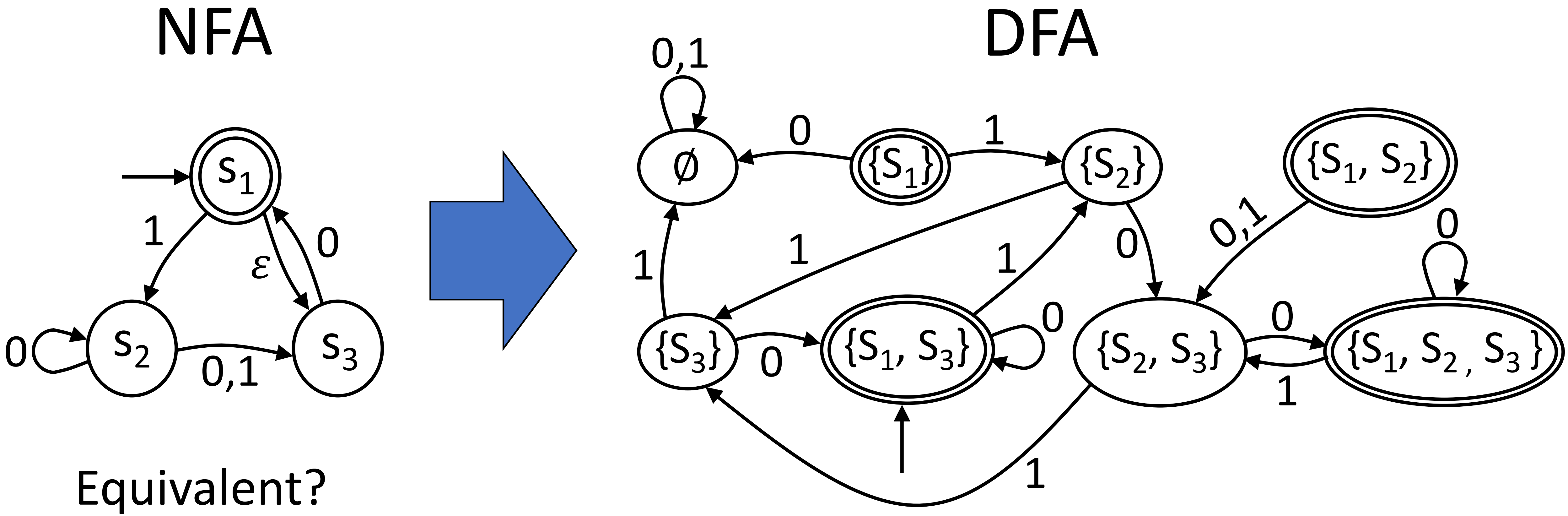




Equivalent?

Suppose w accepted by DFA.

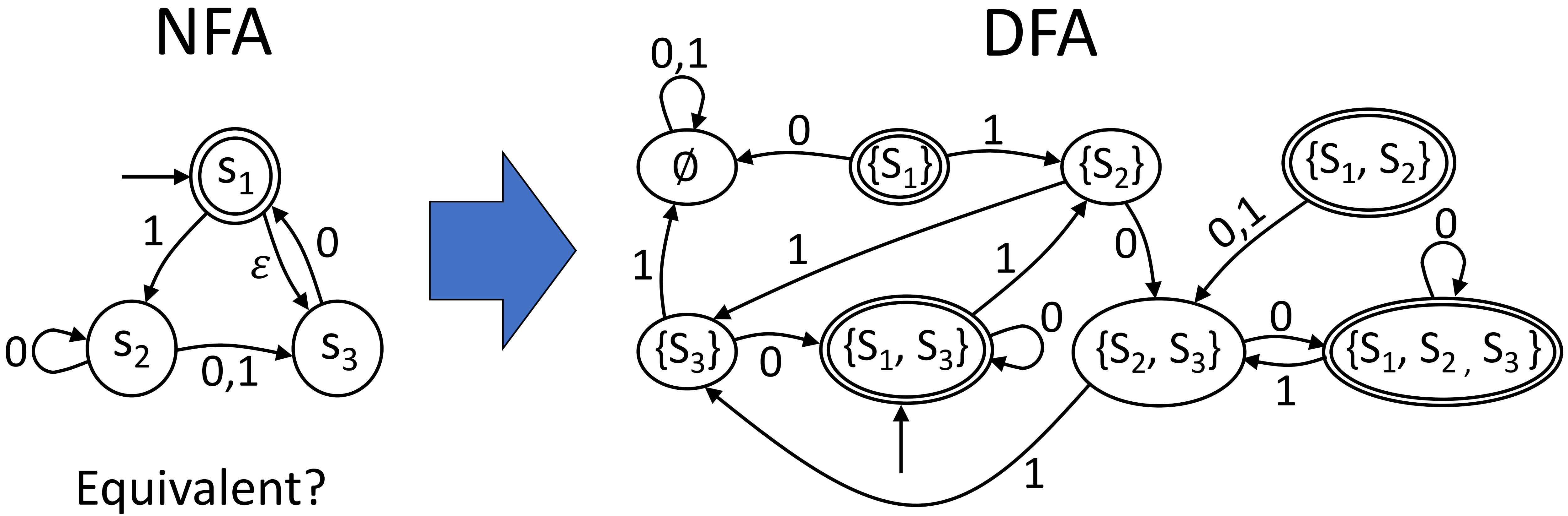
DFA vs NFA



Equivalent?

Suppose w accepted by DFA. At each step of its processing, DFA will be in state that corresponds to all possible NFA states.

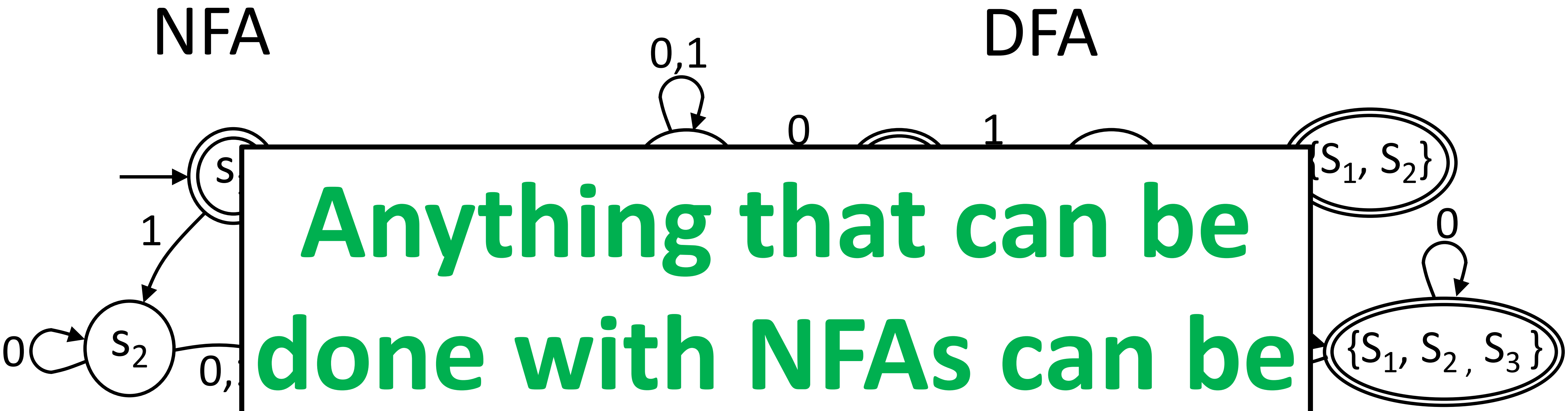
DFA vs NFA



Equivalent?

Suppose w accepted by DFA. At each step of its processing, DFA will be in state that corresponds to all possible NFA states. If DFA ends on accept state, it includes an NFA accept state.

DFA vs NFA



Anything that can be done with NFAs can be done with DFAs!

Equivale

Su cessing,

DFA will be in state that corresponds to all possible NFA states. If DFA ends on accept state, it includes an NFA accept state.

Definitions

A language is called a <u>regular language</u> if some DFA recognizes it.

Definitions

A language is called a <u>regular language</u> if some DFA **or NFA** recognizes it.

Definitions

A language is called a regular language if some DFA **or** **NFA** recognizes it.

How do you prove a language is regular?

Make a DFA that recognizes it.

Definitions

A language is called a regular language if some DFA **or NFA** recognizes it.

How do you prove a language is regular?

Make a DFA **or NFA** that recognizes it.

Make an NFA with three states for:

$\{\omega : \omega \text{ has the form } 0^*1^*0^+.\}$

Proof:

Additional string notation:

0^*: Zero or more 0s (e.g. $0, 0000, \varepsilon$)

0^+: One or more 0s (e.g. $0, 0000$)

Make an NFA with three states for:

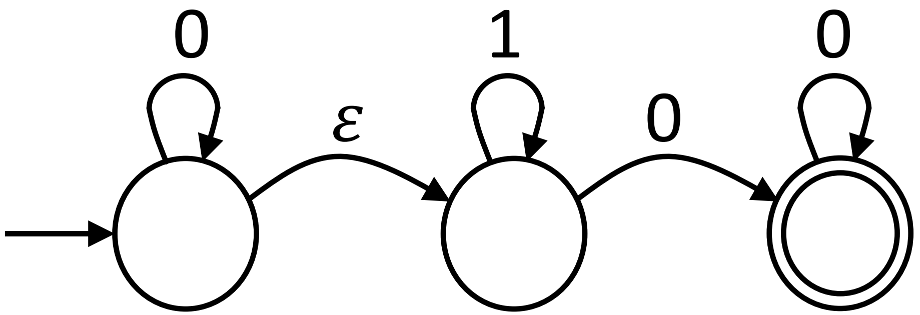
$\{\omega : \omega$ has the form $0^*1^*0^+.\}$

Proof:

Additional string notation:

0^*: Zero or more 0s (e.g. $0, 0000, \varepsilon$)

0^+: One or more 0s (e.g. $0, 0000$)



Make an NFA with three states for:

$\{\omega: \omega \text{ has the form } 1^*(001^+)^*.\}$

e.g. 0010011111, 11100111001

Proof:

Additional string notation:

0^*: Zero or more 0s (e.g. $0, 0000, \varepsilon$)

0^+: One or more 0s (e.g. $0, 0000$)

Make an NFA with three states for:

$\{\omega : \omega \text{ has the form } 1^*(001^+)^*.\}$

e.g. 0010011111, 11100111001

Additional string notation:

0^*: Zero or more 0s (e.g. $0, 0000, \varepsilon$)

0^+: One or more 0s (e.g. $0, 0000$)

Proof: