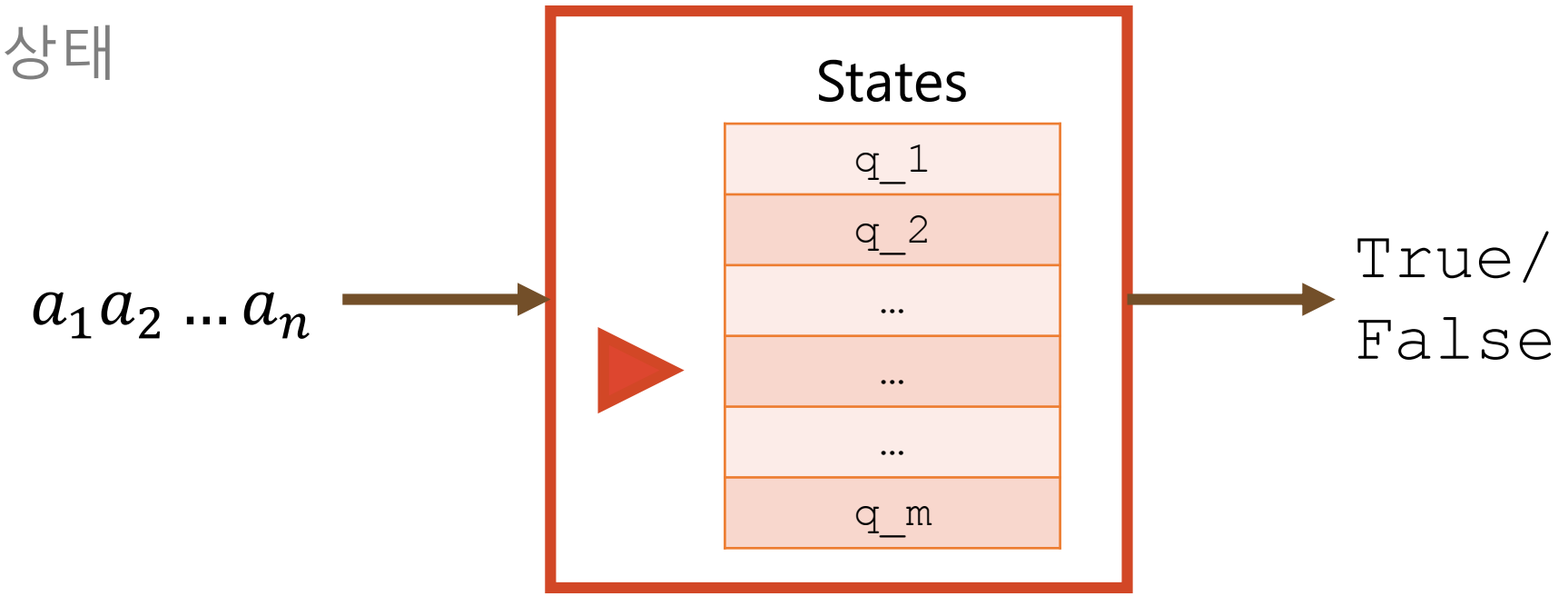
Finite Automata

2015.10.07

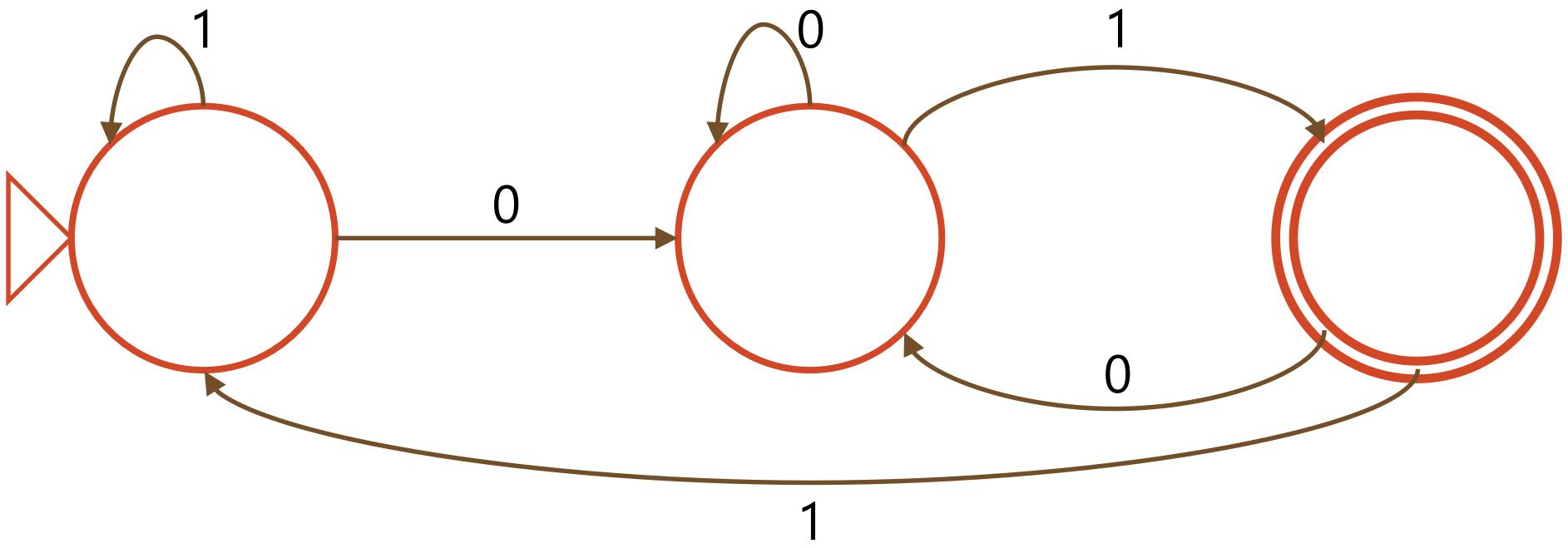
Finite Automata

- 연산 작업을 수행하는 수학적 모델
  - 정해진 입력
  - 한정된 상태



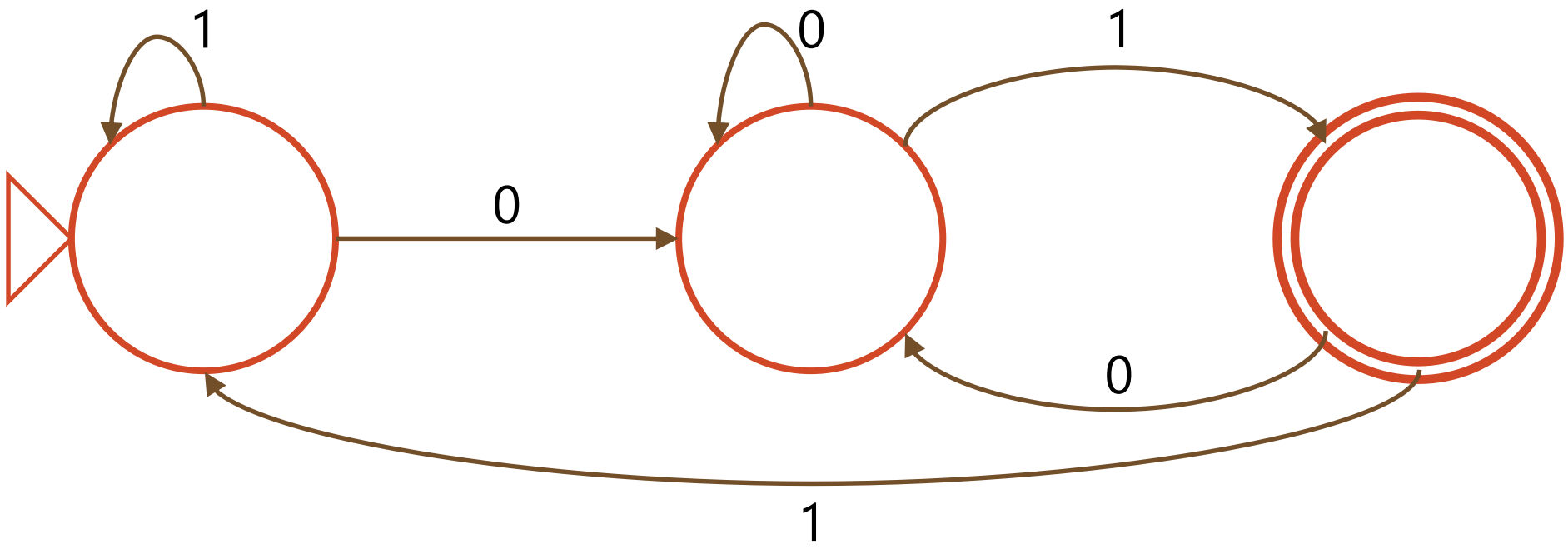
Finite Automata

맨 뒤 두 개의 문자열이 01인 문장을 받아들이는 오토마타



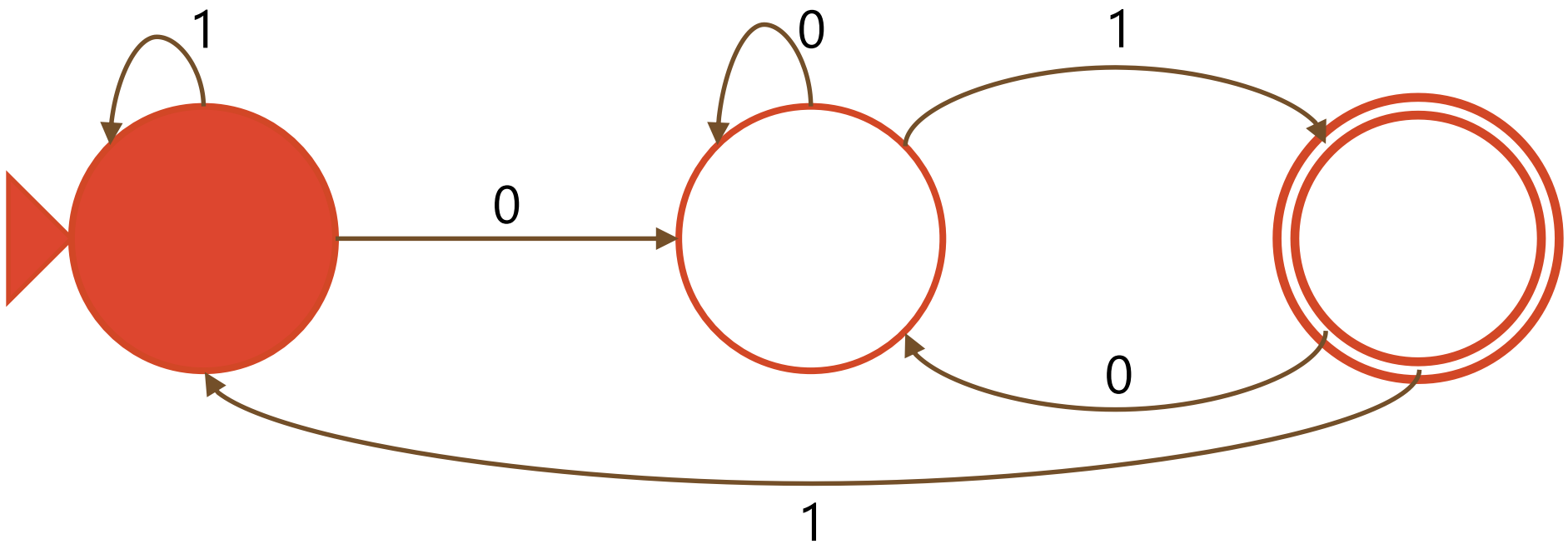
Finite Automata

```
>> 001101
```



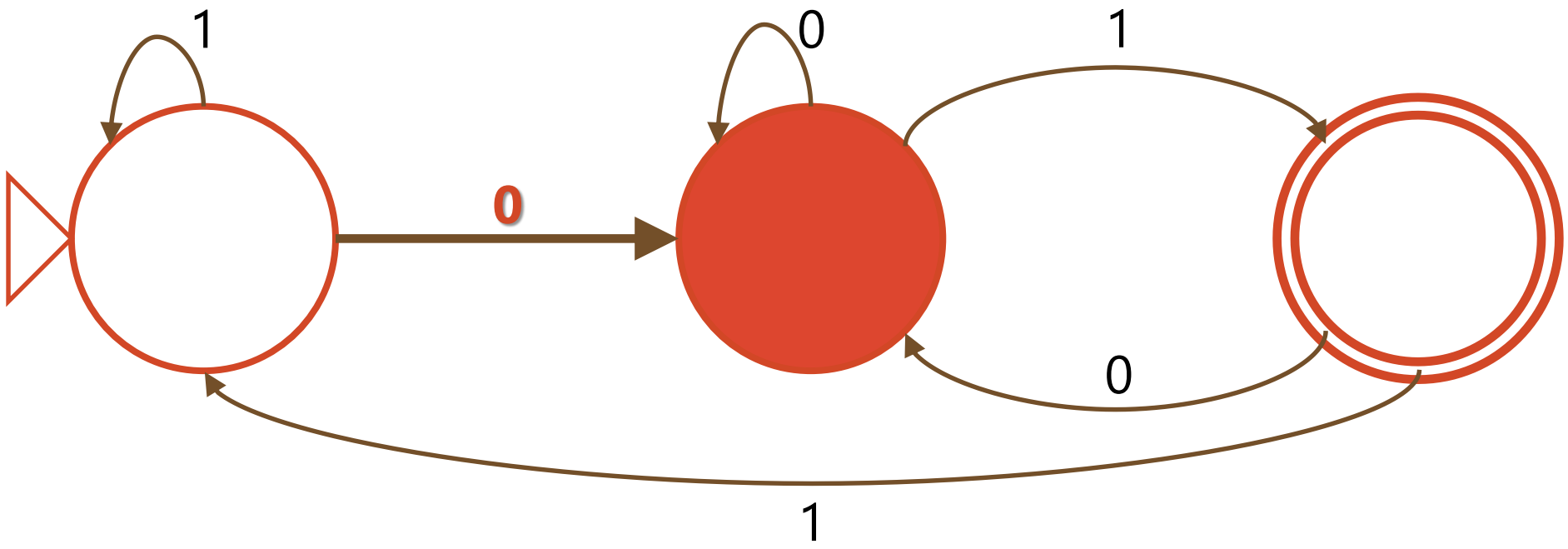
Finite Automata

```
>> 001101
```



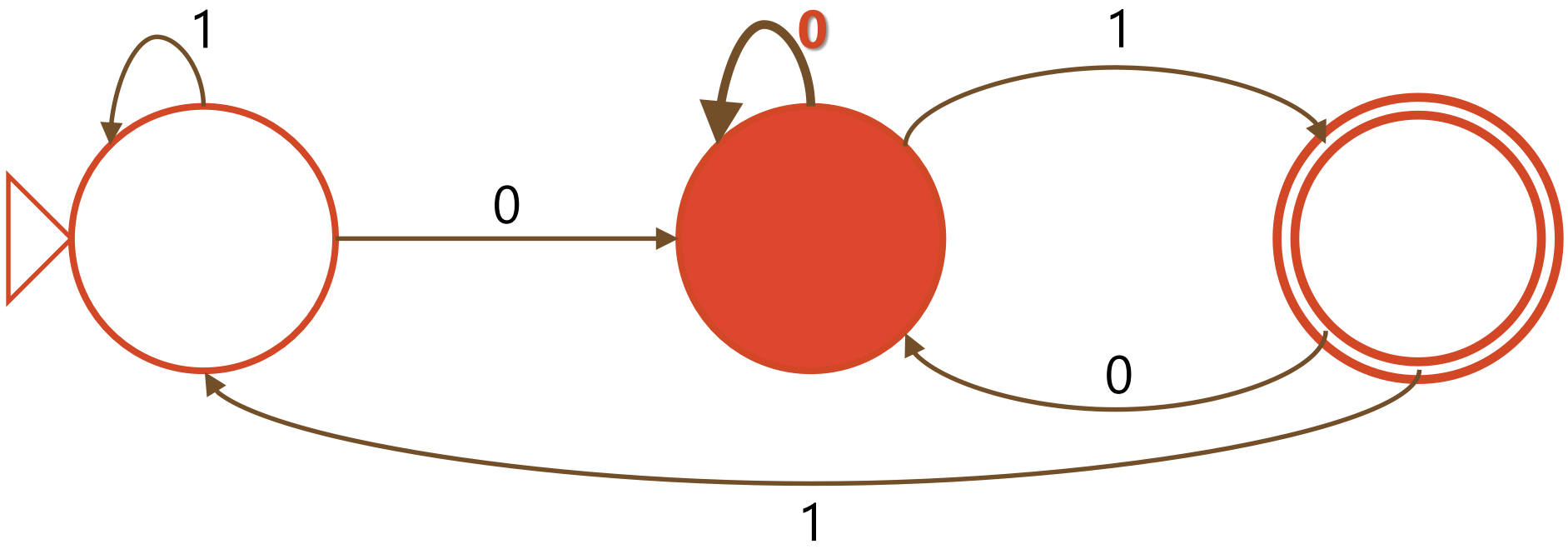
Finite Automata

>> **0**01101



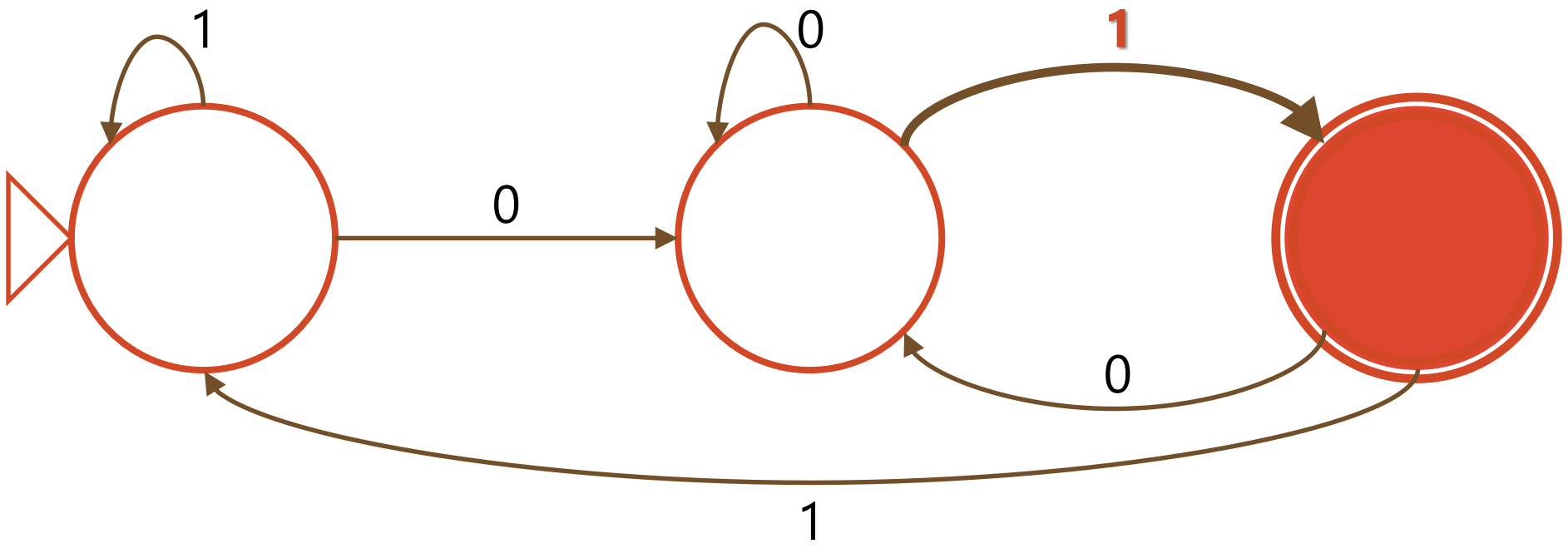
Finite Automata

>> 0**0**1101



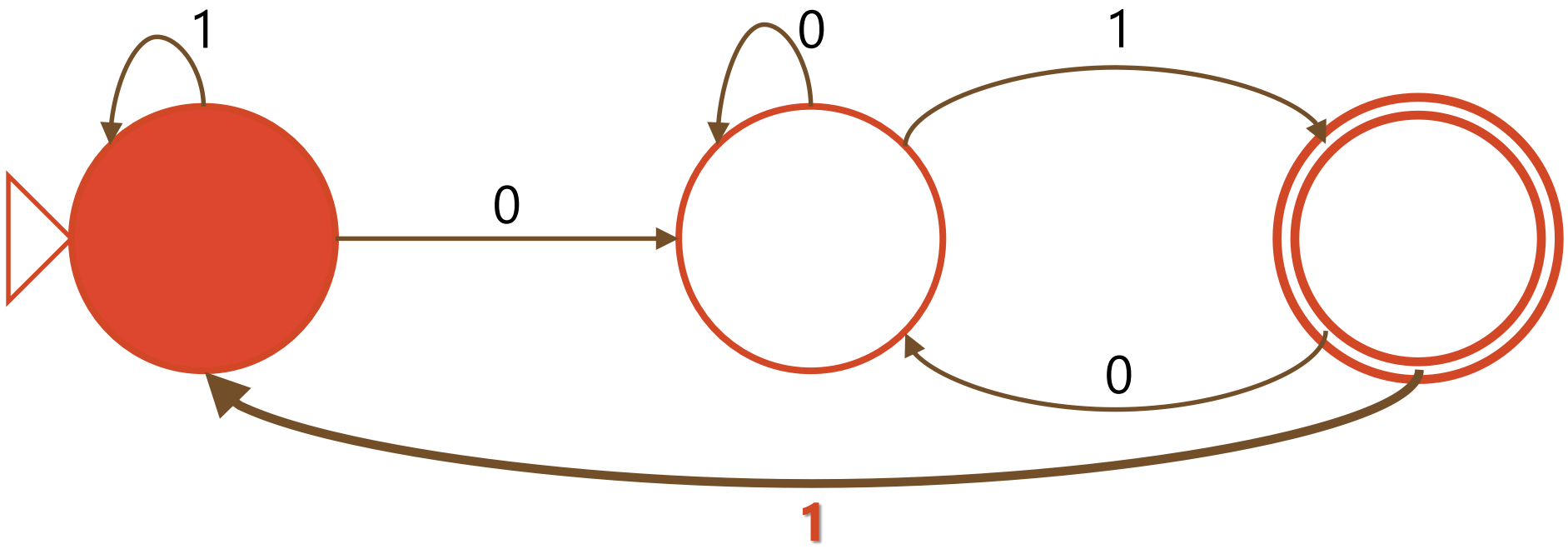
Finite Automata

>> 0011101



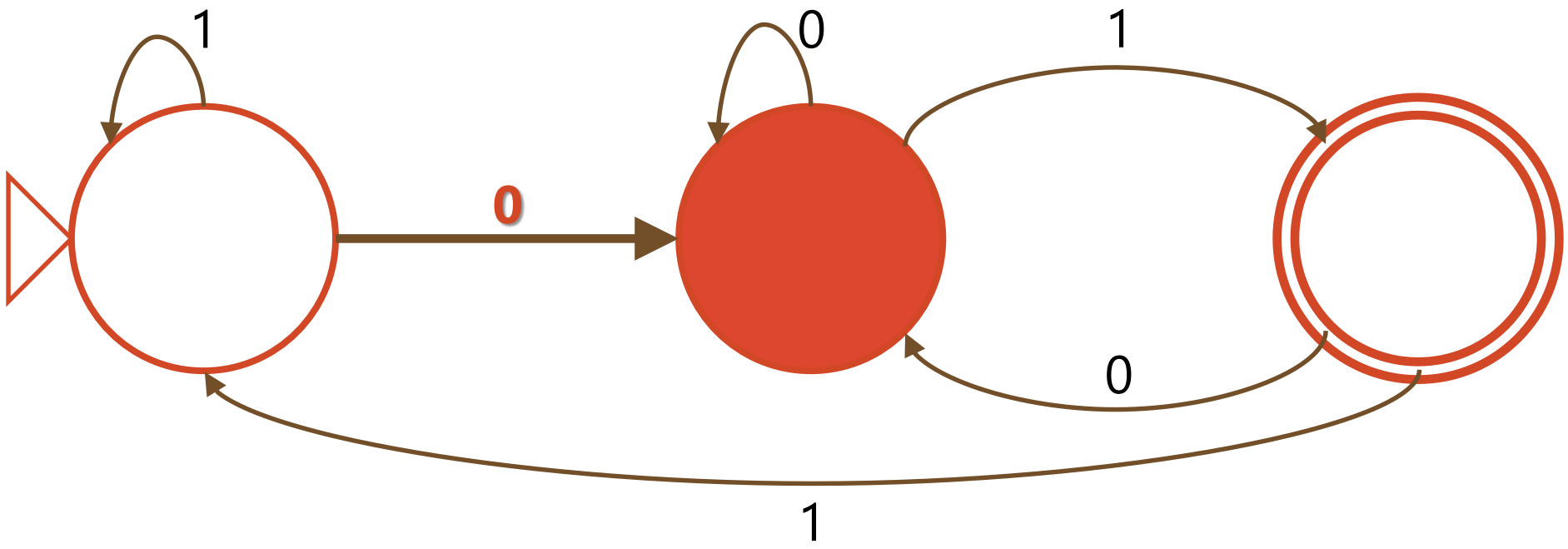
Finite Automata

>> 001101



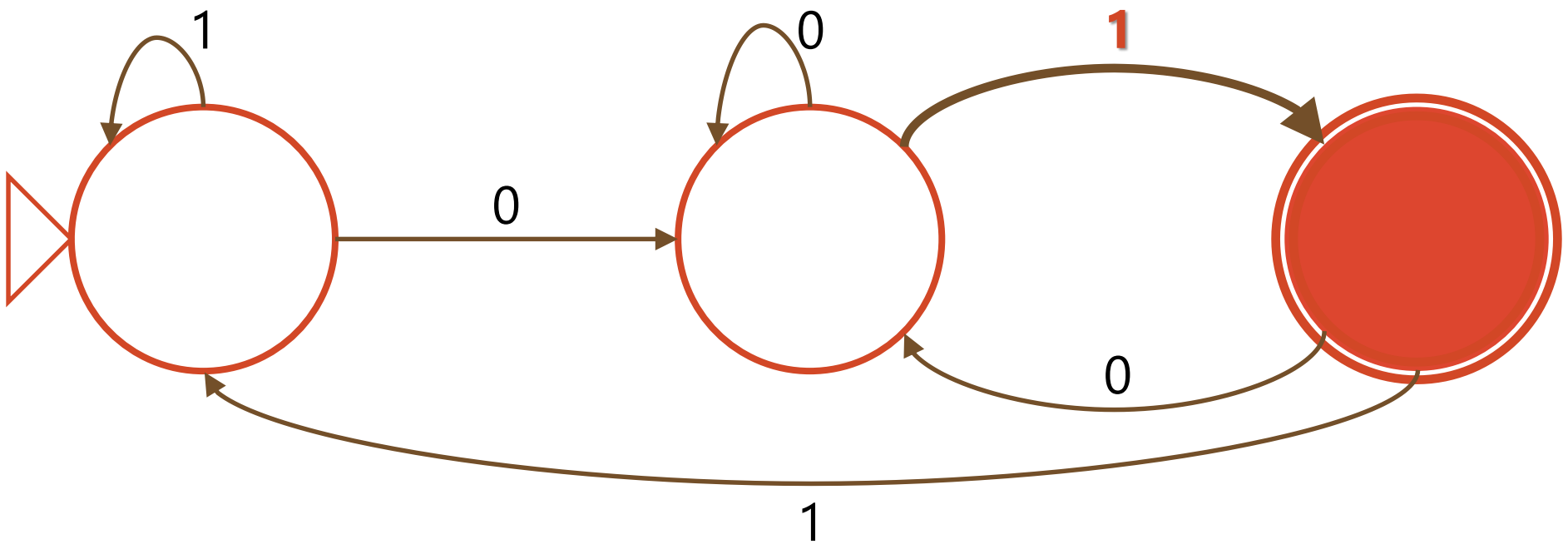
Finite Automata

>> 001101



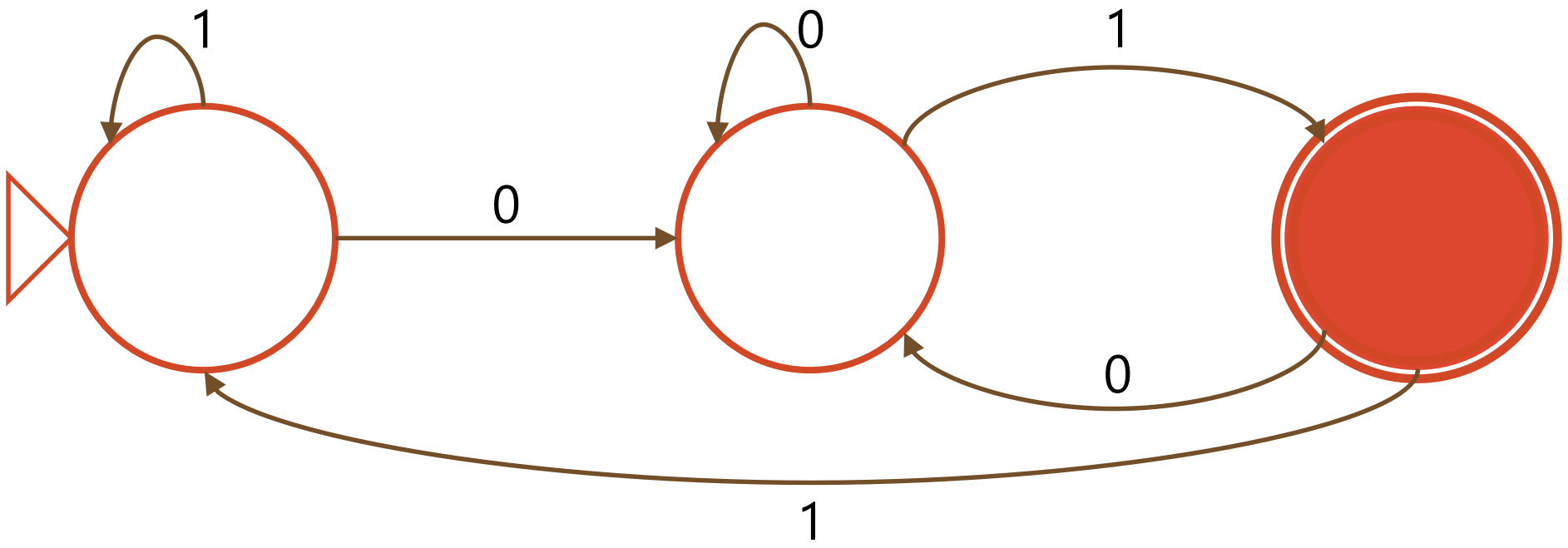
Finite Automata

```
>> 001101
```



Finite Automata

```
>> 001101
True
```



DFA

DFA $D = (Q, \Sigma, \delta, q_0, F)$

- Q : 상태들의 집합
- Σ : 입력 문자열의 집합
- $\delta : Q \times \Sigma \to Q$: 상태 이동 함수
- $q_0 \in Q$: 초기 상태
- $F \sqsubseteq Q$: 최종 상태들

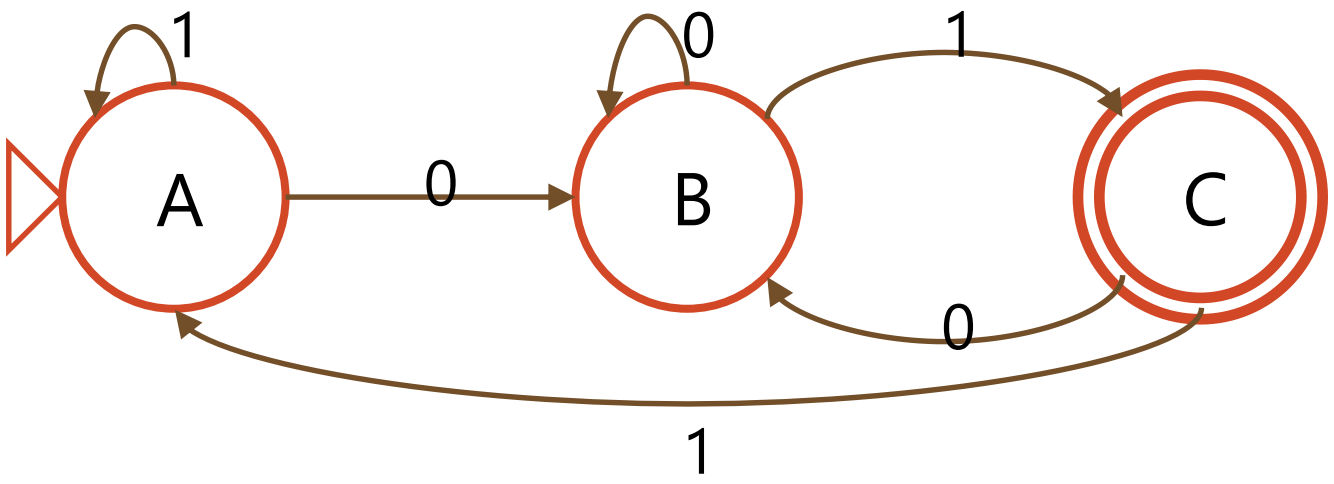
DFA

```
def get_dfa (Q, Sigma, delta, q0, F):
    def dfa (str):
        state = q0
        for ch in str:
            assert(ch in Sigma)
            state = delta[state][ch]
        return state in F
    return dfa
```

DFA

- $Q = \{A, B, C\}$, $\Sigma = \{0,1\}$
- $q_0 = A$, $F = \{C\}$
- $\delta : Q \times \Sigma \rightarrow Q$

	0	**1**
A	B	A
B	B	C
C	B	A



NFA

NFA $N = (Q, \Sigma, \delta, q_0, F)$

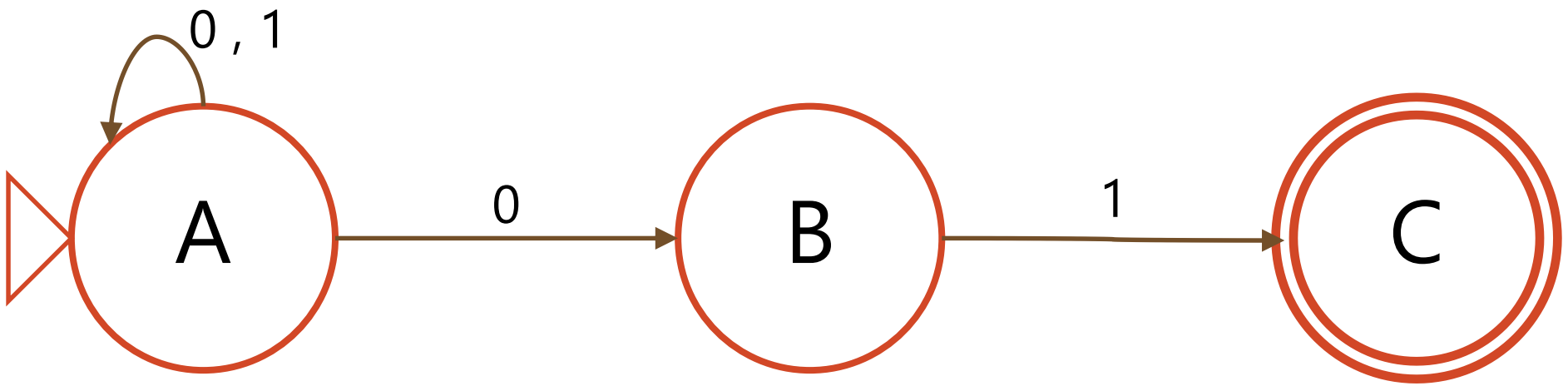
- Q : 상태들의 집합
- Σ : 입력 문자열의 집합
- $\boldsymbol{\delta : Q \times \Sigma \rightarrow 2^Q}$: 상태 이동 함수
- $q_0 \in Q$: 초기 상태
- $F \subseteq Q$: 최종 상태들

NFA

- DFA : $\delta : Q \times \Sigma \rightarrow Q$
  - 각 상태와 입력에 대해 다음 상태가 유일하게 정해져 있다
- NFA : $\delta : Q \times \Sigma \rightarrow 2^Q$
  - 다음 상태가 여러 가지 경우일 수도 있다. $(|\delta(q, a)| > 1)$
  - 다음 상태가 존재하지 않을 수도 있다. $(\delta(q, a) = \emptyset)$

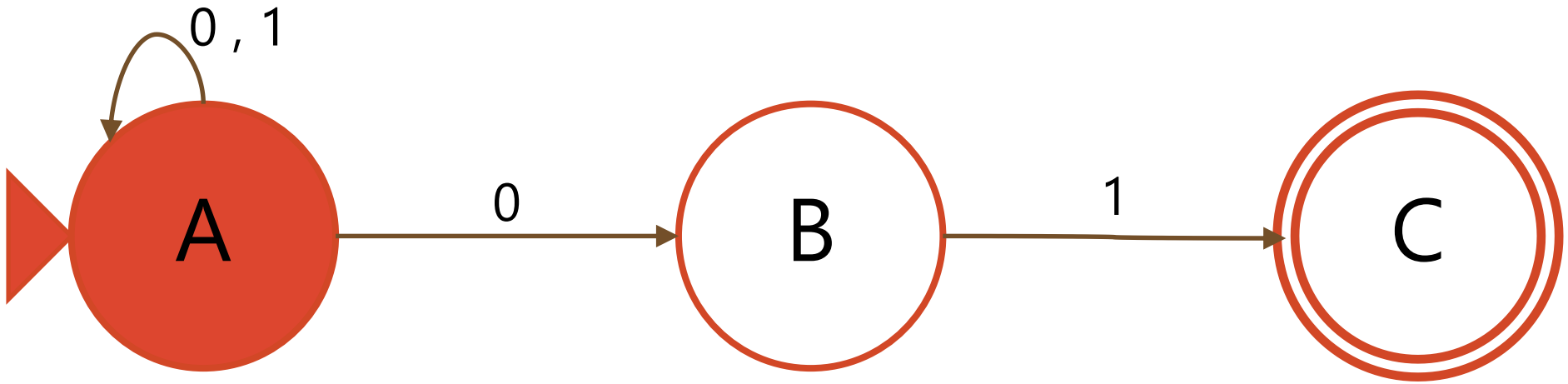
NFA

맨 뒤 두 개의 문자열이 01인 문장을 받아들이는 오토마타



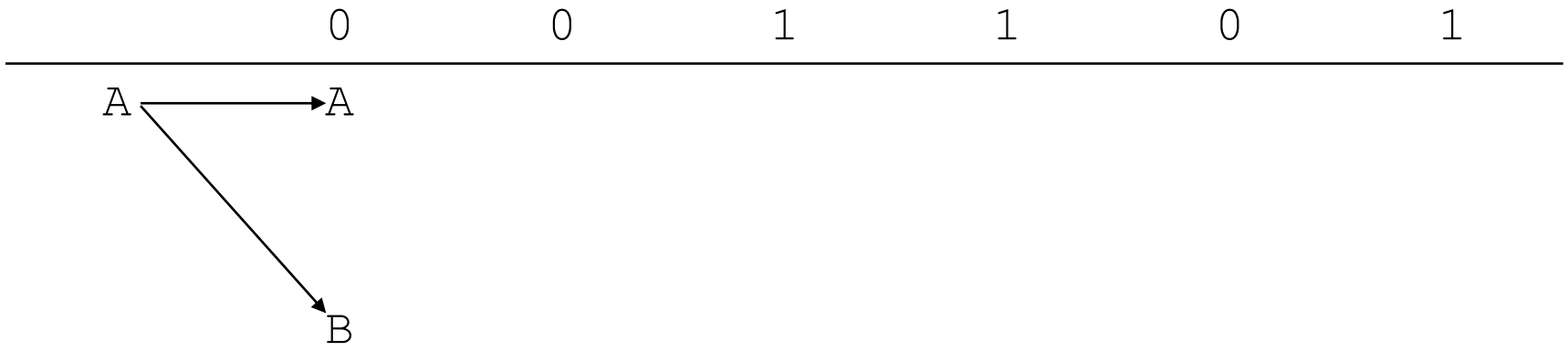
NFA

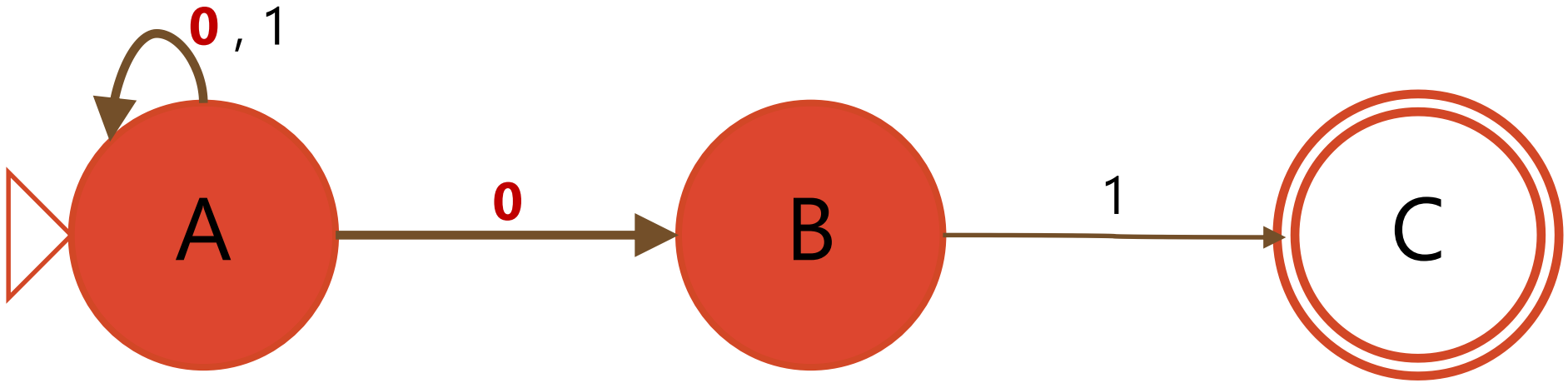
```
>> 001101

                0     0     1     1     0     1
  ------------------------------------------------
  A
```



NFA

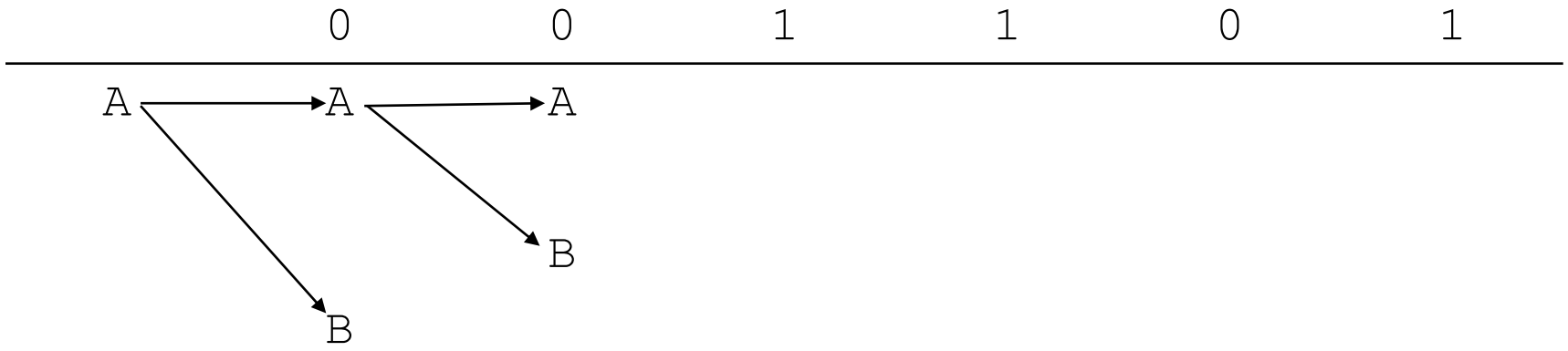
>> 0 0 1 1 0 1

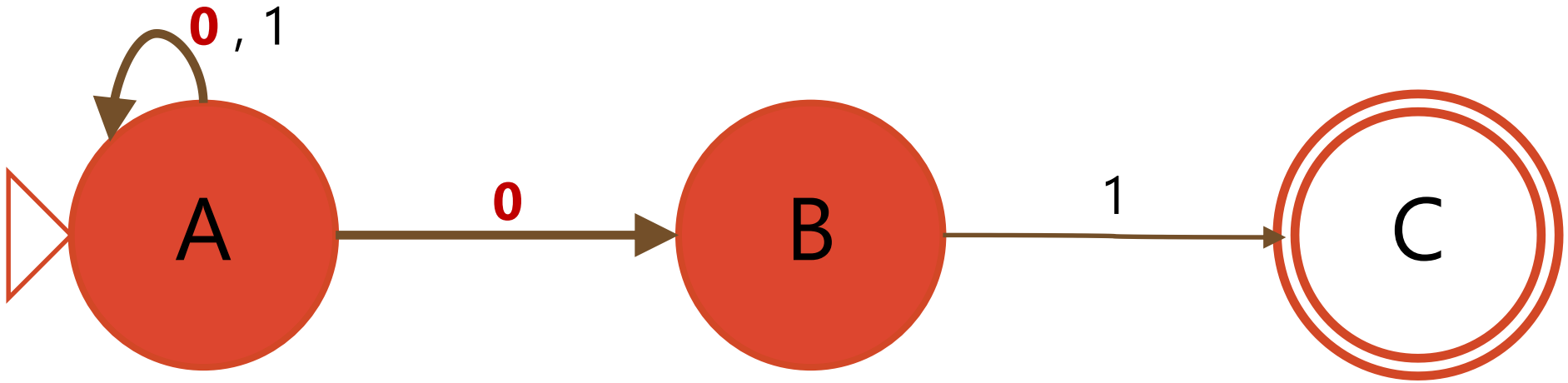




NFA

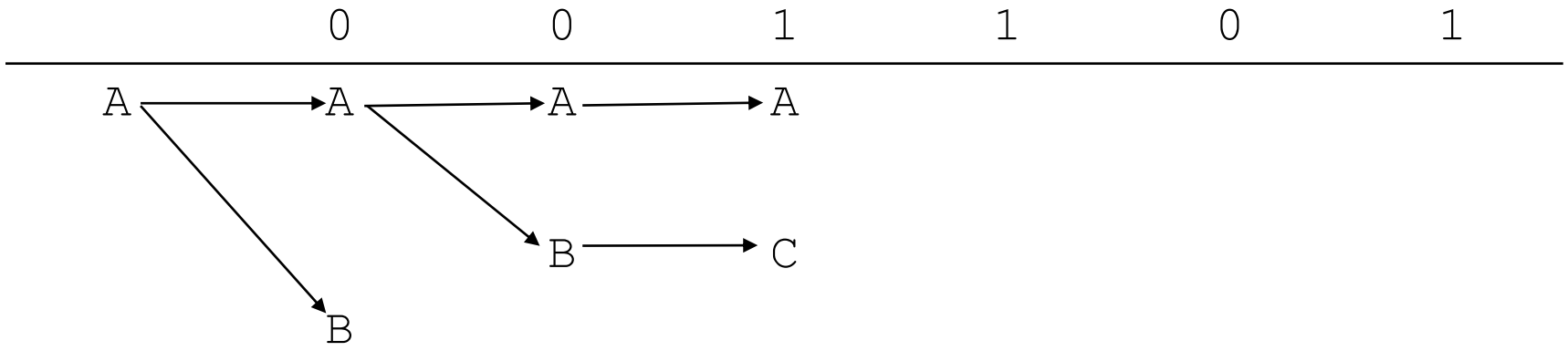
>> 0**0**1101

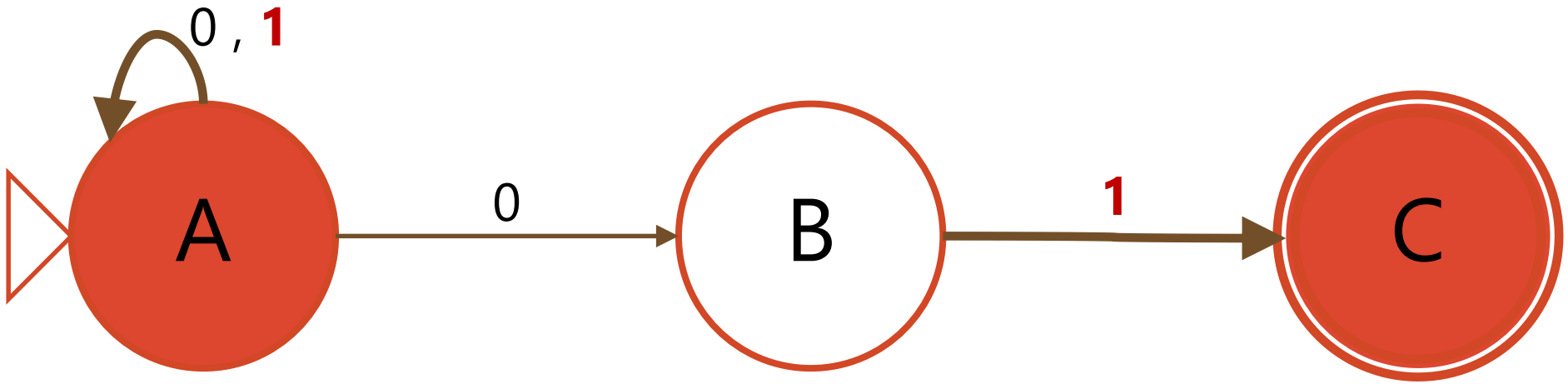




NFA

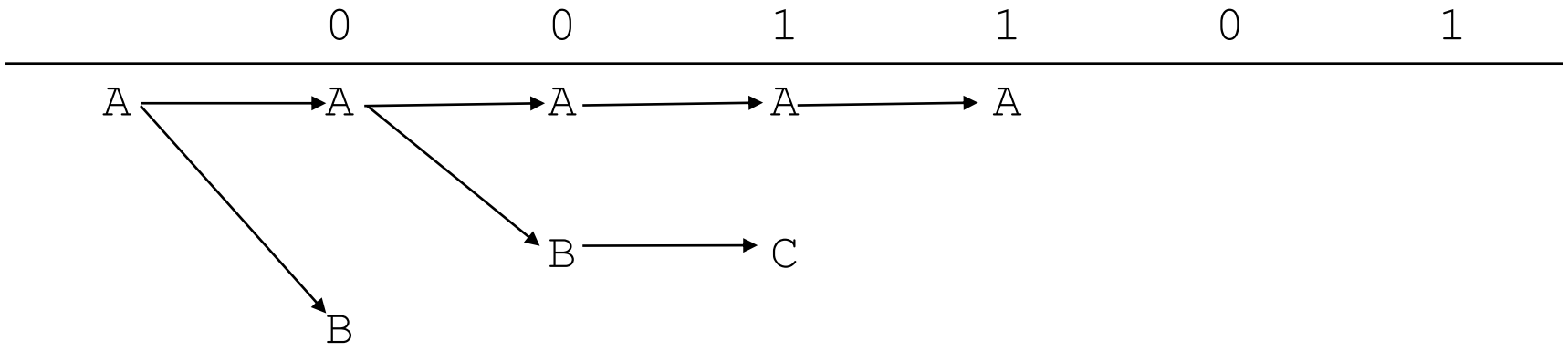
>> 0011101

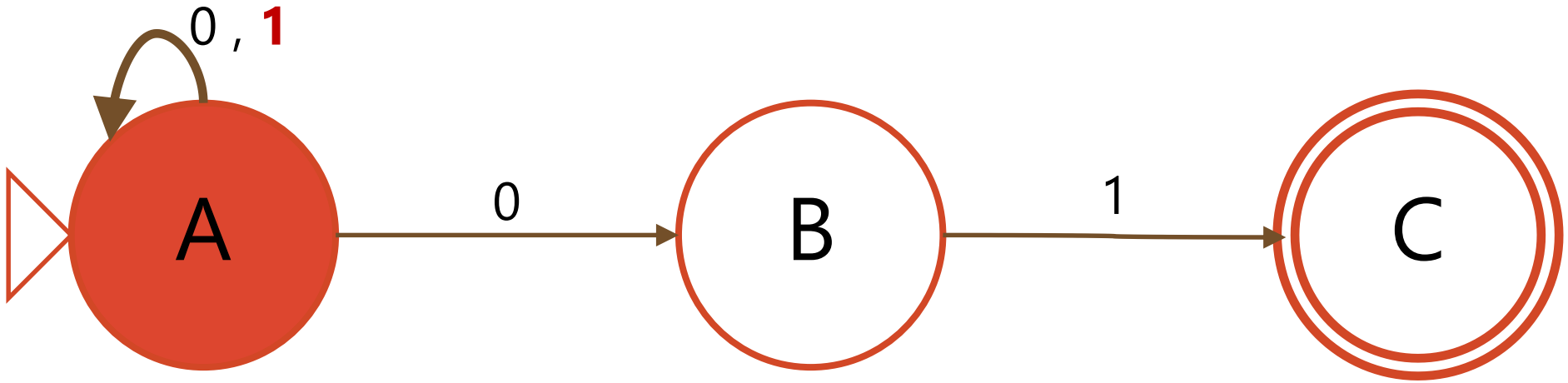




NFA

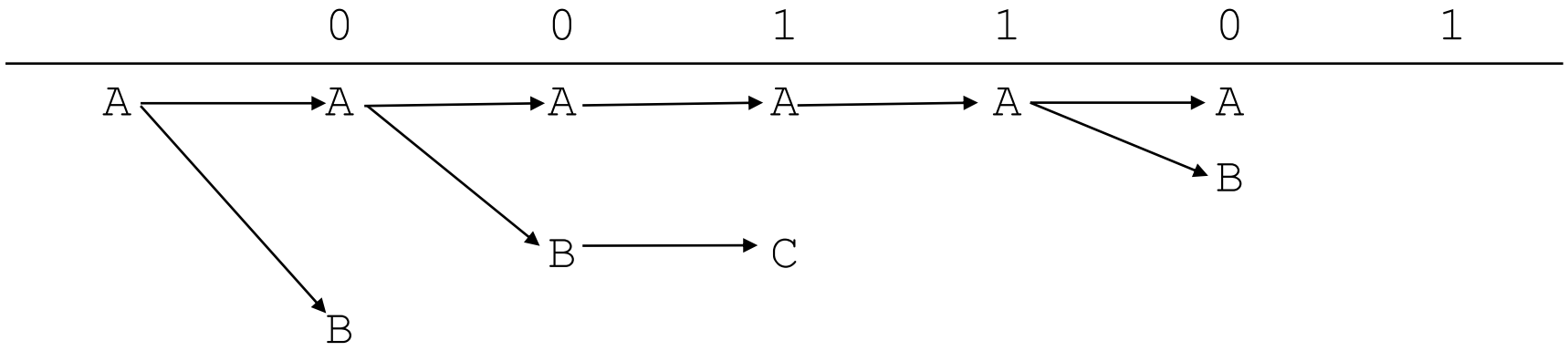
>> 001101

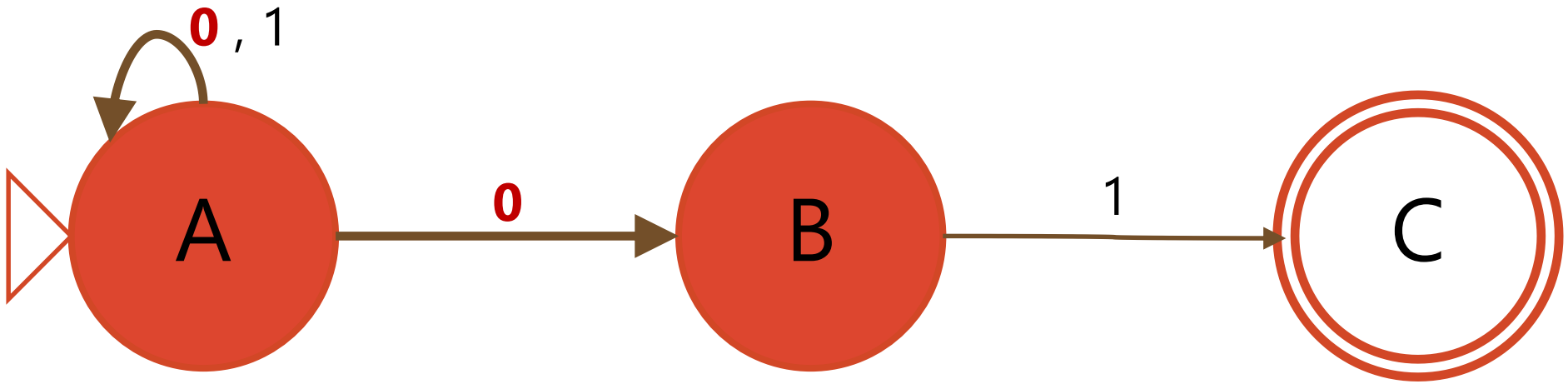




NFA

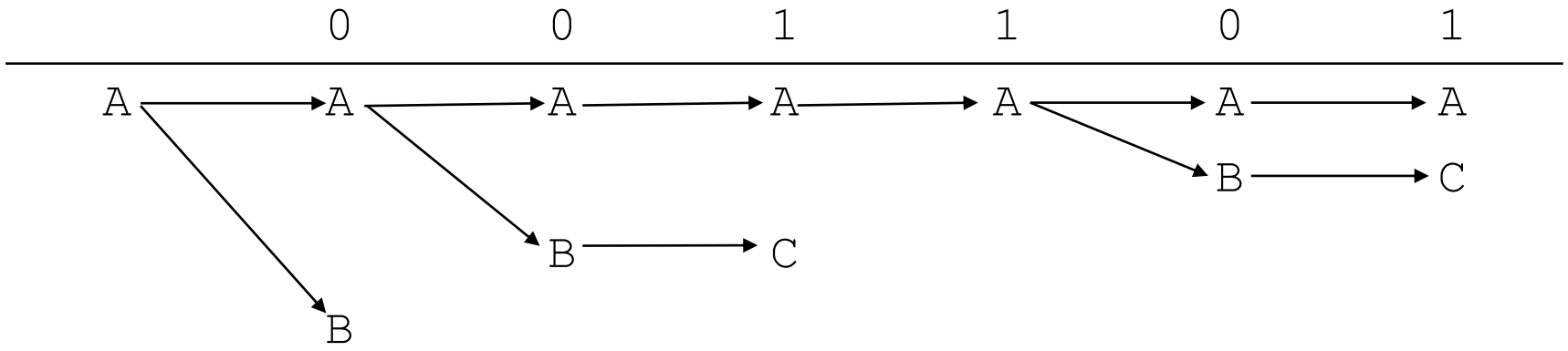
>> 001101

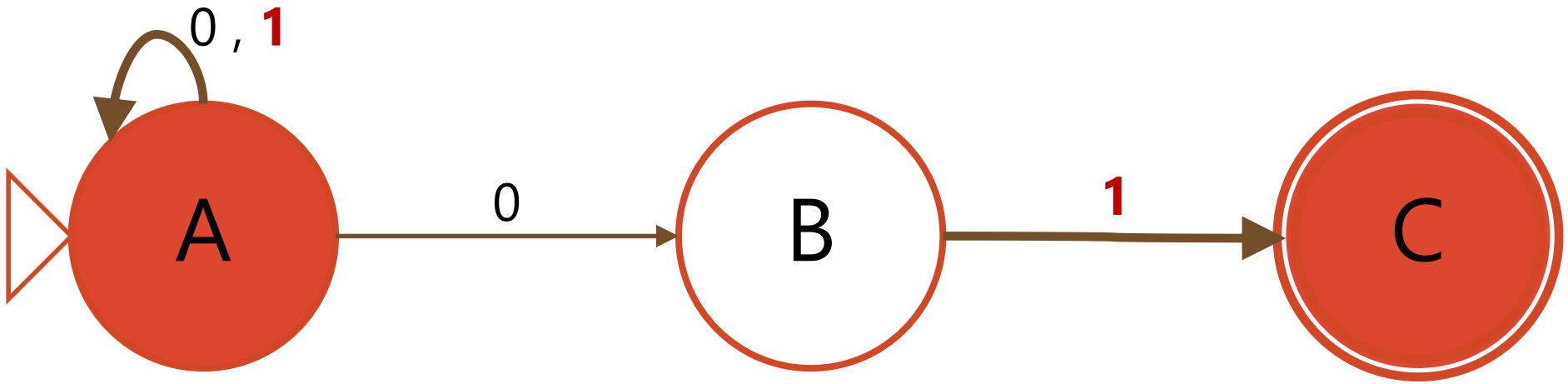




NFA

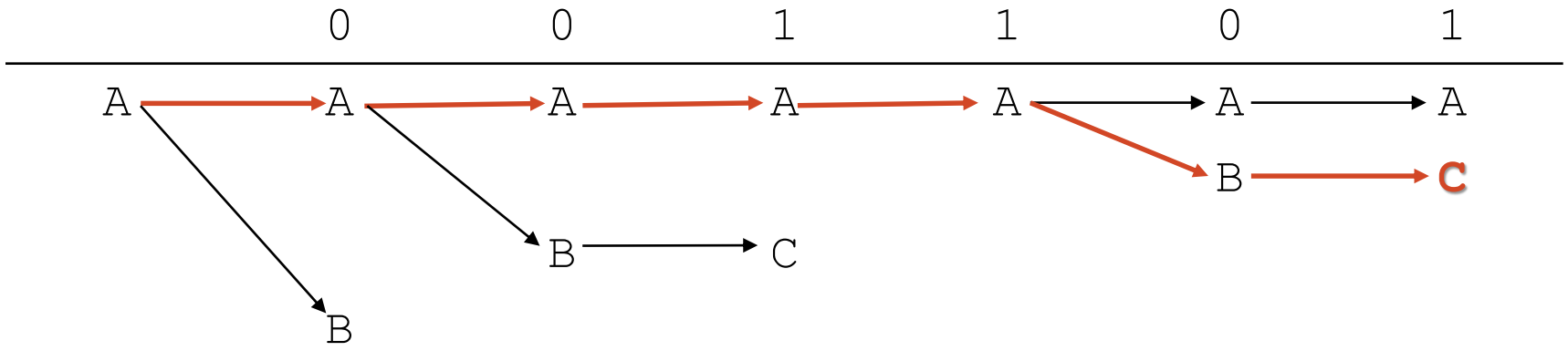
>> 001101

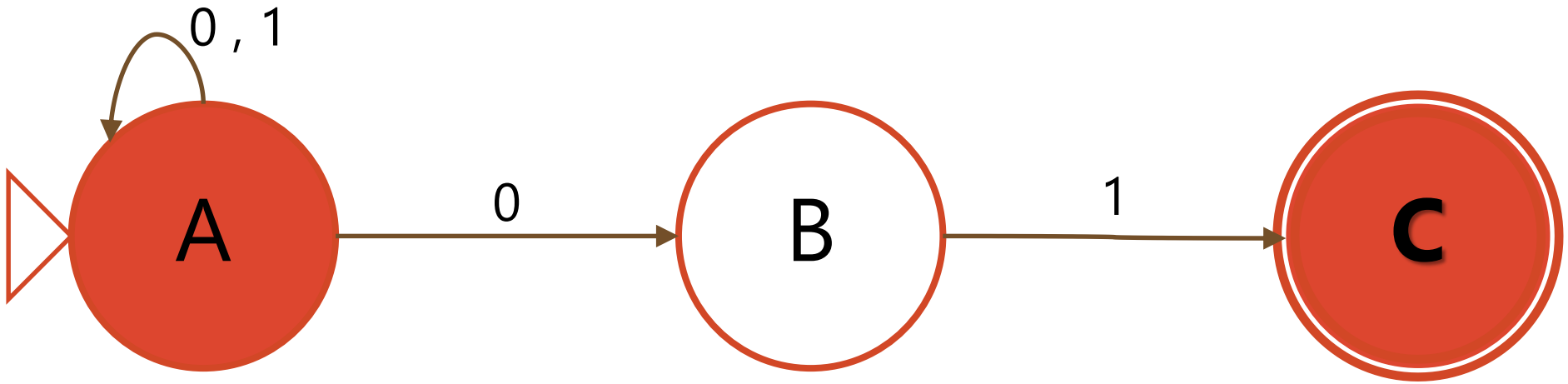




NFA

```
>> 001101
True
```

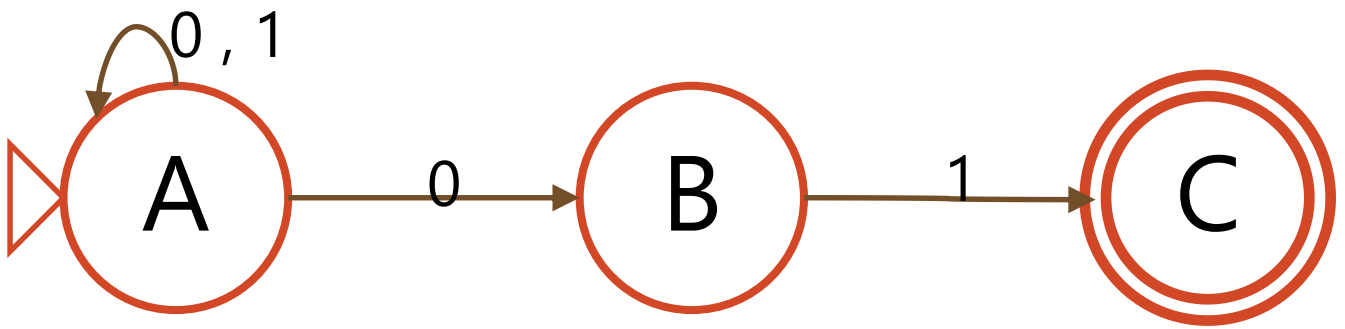




NFA

- $Q = \{A, B, C\}$, $\Sigma = \{0,1\}$
- $q_0 = A$, $F = \{C\}$
- $\delta : Q \times \Sigma \rightarrow 2^Q$

	0	1
A	{A,B}	{A}
B	∅	{C}
C	∅	∅



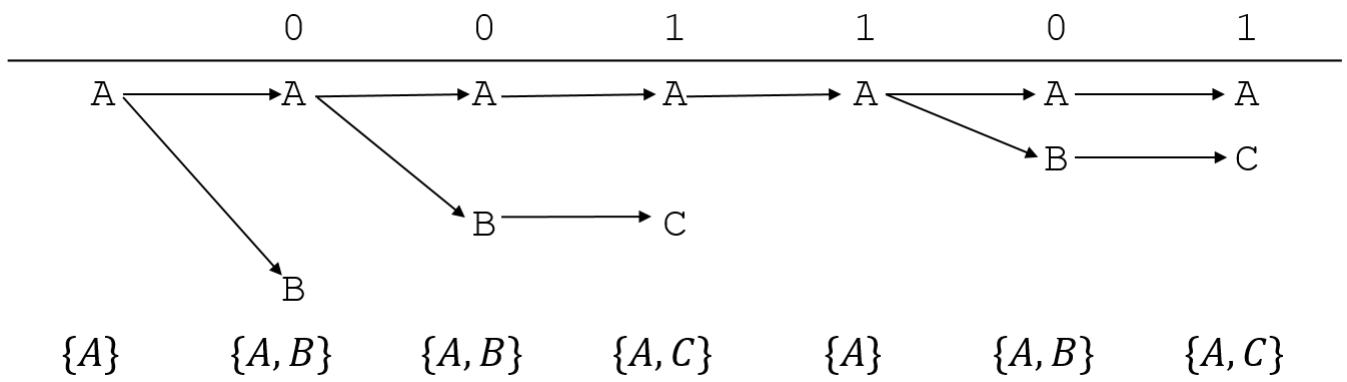
DFA to NFA

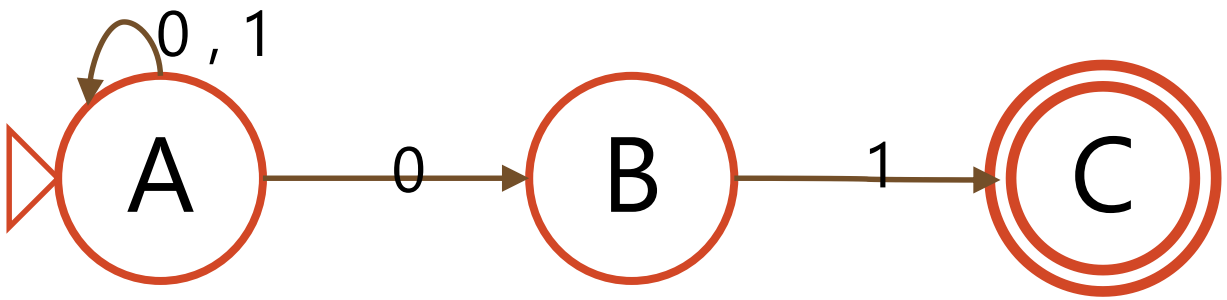
주어진 DFA $D = (Q_D, \Sigma, \delta_D, q_{0D}, F_D)$에 대해

- $Q_N = Q_D$, $q_{0N} = q_{0D}$, $F_N = F_D$
- $\delta_N : Q_N \times \Sigma \to 2^{Q_N}$, $\delta_N(q, a) = \{\delta_D(q, a)\} \subset Q_D = Q_N$
  - 가능한 다음 상태가 유일하므로 DFA와 똑같이 작동하게 된다.

와 같이 지정하면 NFA $N = (Q_N, \Sigma, \delta_N, q_{0N}, F_N)$은 DFA D와 똑같이 작동한다.

NFA to DFA





- $\{A\}, 0 \rightarrow \{A, B\}$
- $\{A, B\}, 0 \rightarrow \{A, B\}$
- $\{A, B\}, 1 \rightarrow \{A, C\}$
- $\{A, C\}, 1 \rightarrow \{A\}$
- Etc…

NFA to DFA

주어진 NFA $N = (Q_N, \Sigma, \delta_N, q_{0N}, F_N)$에 대해서

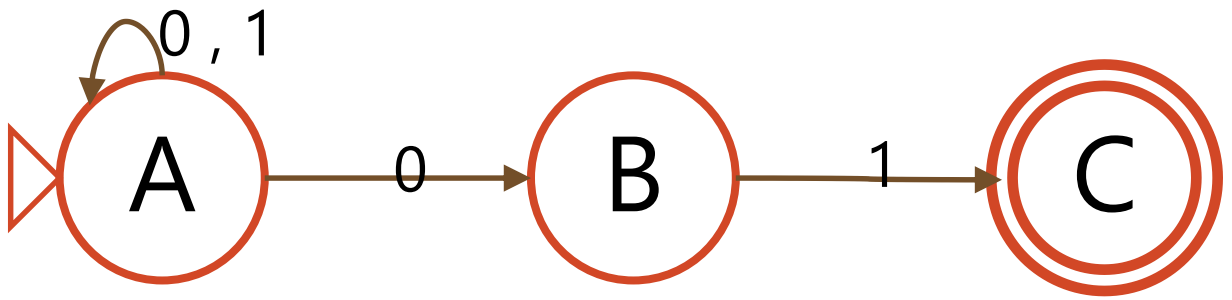
- $Q_D = 2^{Q_N}$
- $\delta_D : Q_D \times \Sigma \to Q_D$, $\delta(P, a) = \bigcup_{q \in P} \delta(q, a) \subseteq Q_N$ $(P \subseteq Q_N)$
  - 이전까지의 입력으로 도달할 수 있는 상태의 집합이 P일 때 a를 추가로 입력함으로써 도달할 수 있는 상태의 집합을 구한다.
- $q_{0D} = \{q_{0N}\}$, $F_D = \{P \subseteq Q_N \mid P \cap F_N \neq \emptyset\}$
  - 초기에는 q_{0N}에만 도달할 수 있다.
  - 도달할 수 있는 상태들 중 하나라도 최종 상태이면 된다.

와 같이 지정하면 DFA $D = (Q_D, \Sigma, \delta_D, q_{0D}, F_D)$는 NFA N과 동치이다.

NFA to DFA

$q_{0D} = \{q_{0N}\} = \{A\}$

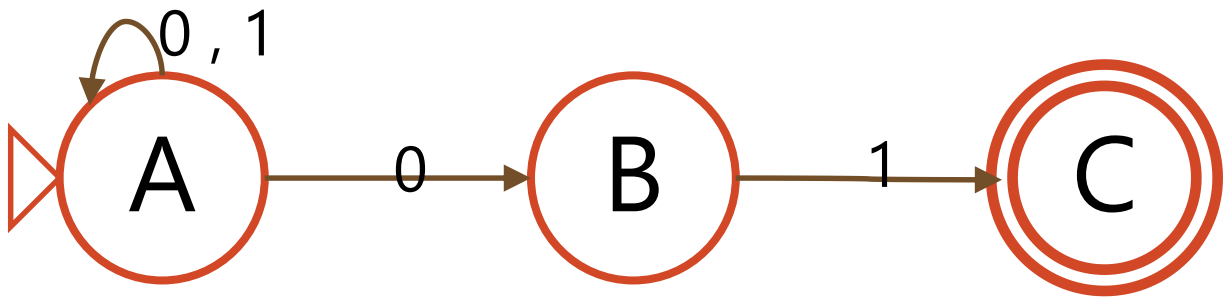
	0	**1**
$\{A\}$		



NFA to DFA

$q_{0D} = \{q_{0N}\} = \{A\}$

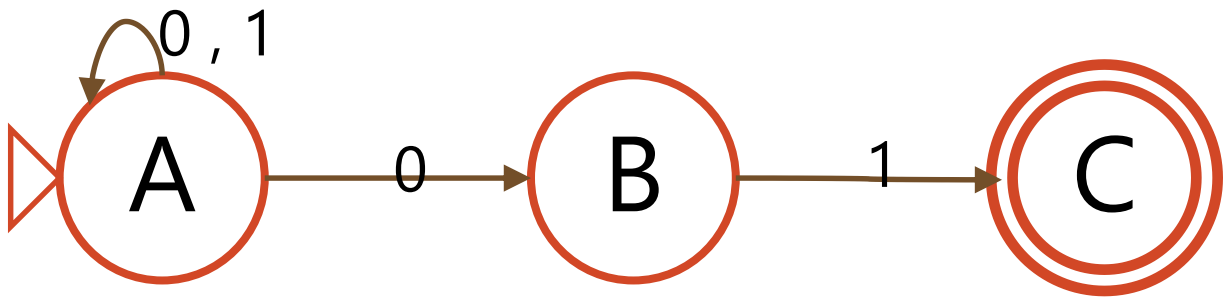
	0	**1**
$\{A\}$	$\{A, B\}$	$\{A\}$
$\{A, B\}$		



NFA to DFA

$q_{0D} = \{q_{0N}\} = \{A\}$

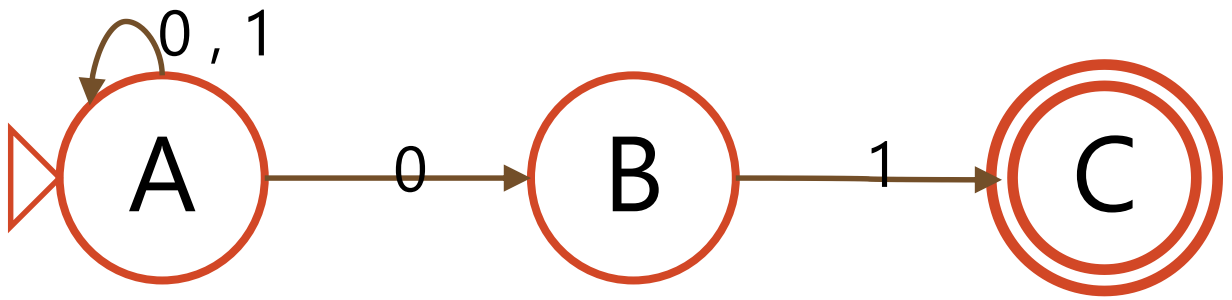
	0	**1**
$\{A\}$	$\{A, B\}$	$\{A\}$
$\{A, B\}$	$\{A, B\}$	$\{A, C\}$
$\{A, C\}$		



NFA to DFA

$q_{0D} = \{q_{0N}\} = \{A\}$

	0	**1**
$\{A\}$	$\{A, B\}$	$\{A\}$
$\{A, B\}$	$\{A, B\}$	$\{A, C\}$
$\{A, C\}$	$\{A, B\}$	$\{A\}$



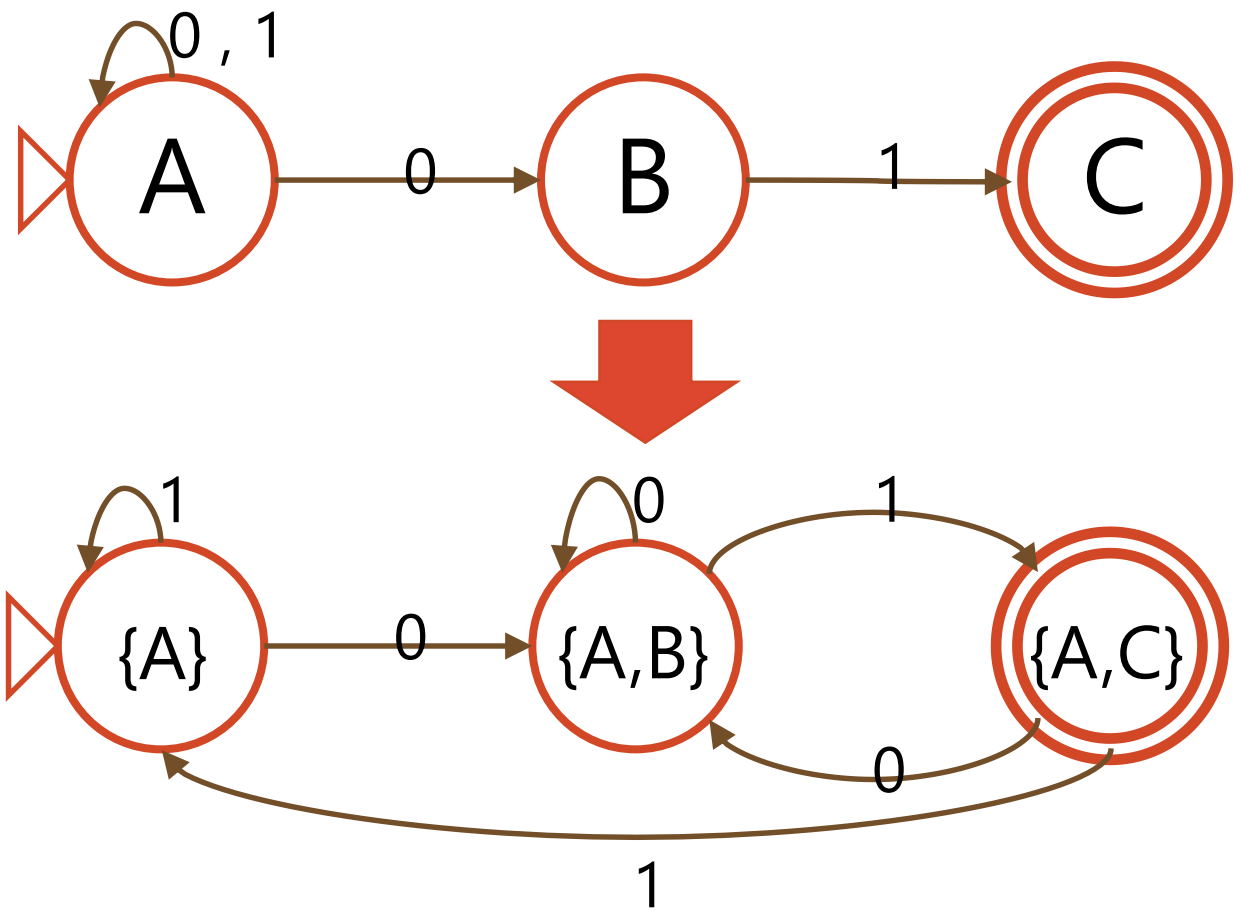
NFA to DFA

$q_{0D} = \{q_{0N}\} = \{A\}$

	0	**1**
$\{A\}$	$\{A, B\}$	$\{A\}$
$\{A, B\}$	$\{A, B\}$	$\{A, C\}$
$\{A, C\}$	$\{A, B\}$	$\{A\}$

$Q_D = \{\{A\}, \{A, B\}, \{A, C\}\}$

$F = \{\{A, C\}\}$



JFLAP

http://www.jflap.org/