Computabilità, Complessità e Logica

Lezione 2

Automi a stati finiti

Un **automa a stati finiti** è una quintupla $(Q, \Sigma, \delta, q_0, F)$ dove:

1. Q è un insieme finito chiamato insieme degli **stati**,

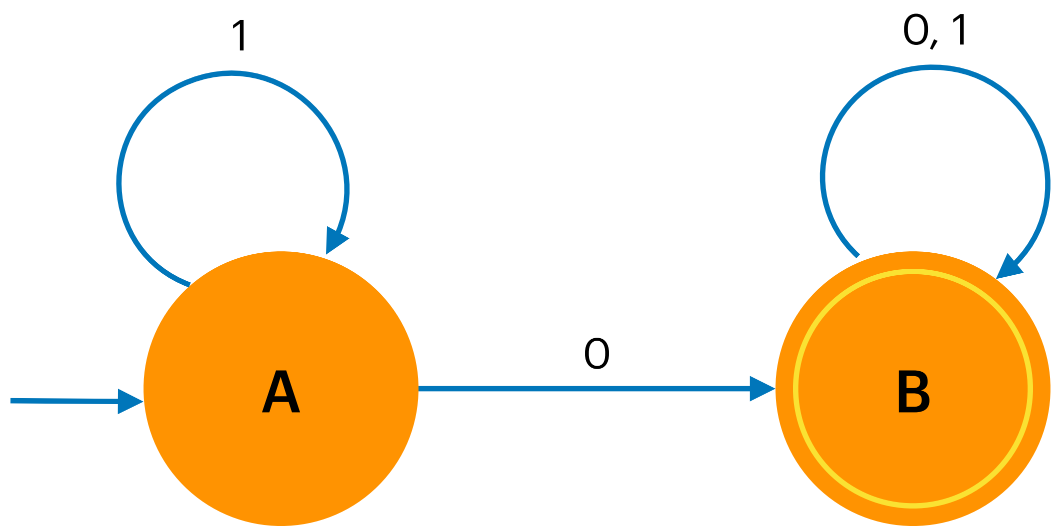
2. Σ è un insieme finito chiamato **alfabeto**,

3. $\delta\colon Q \times \Sigma \to Q$ è la **funzione di transizione**,

4. $q_0 \in Q$ è lo **stato iniziale**,

5. $F \subseteq Q$ è l'insieme degli **stati accettanti** (o stati finali).

Un esempio
Dal diagramma degli stati alla definizione



- La funzione di transizione δ
  può essere definita da una tabella:

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

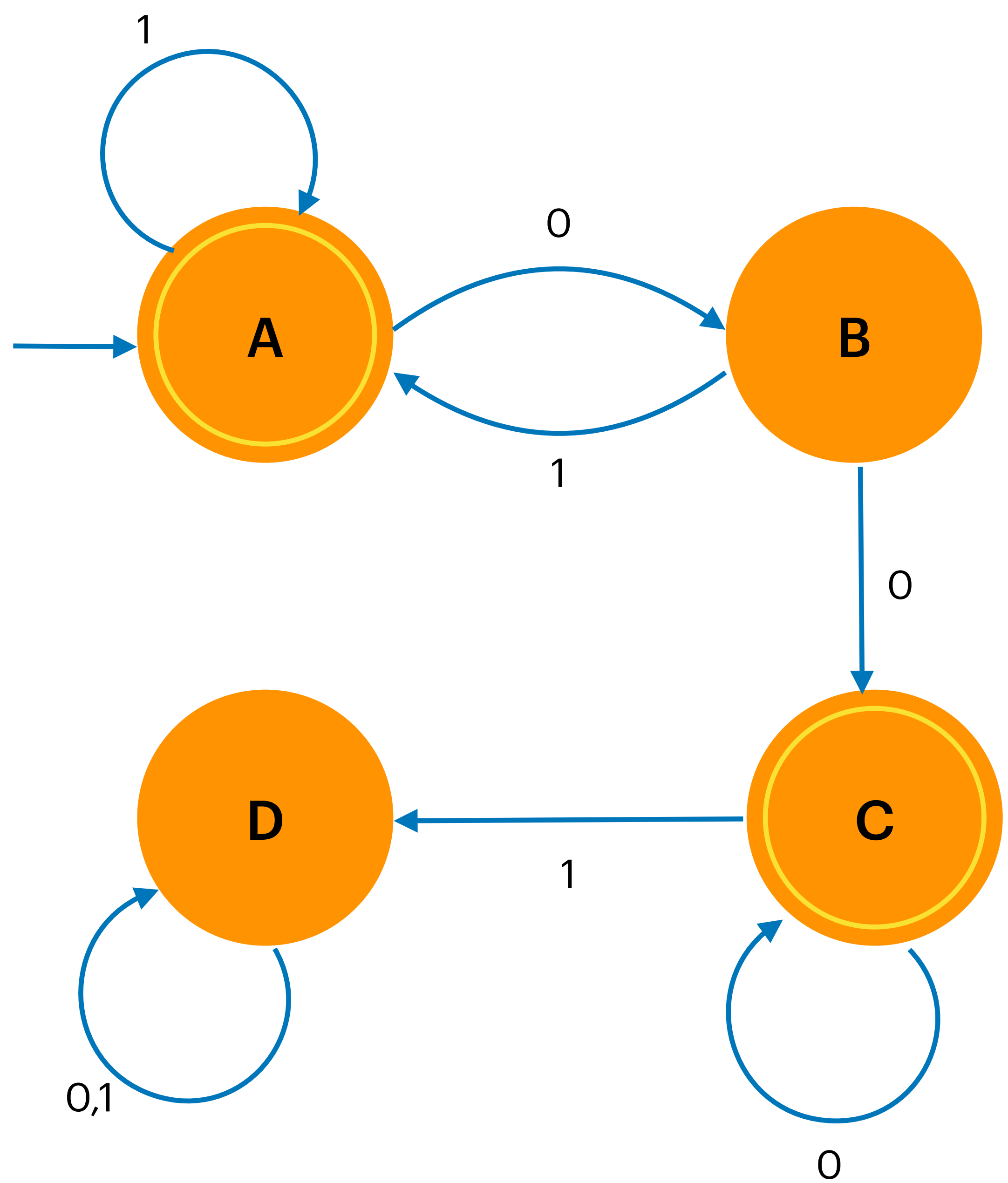
- Ha come insieme degli stati $Q = \{A, B\}$
- L'alfabeto consiste di due soli simboli: $\Sigma = \{0,1\}$
- Lo stato iniziale è indicato da una freccia entrante, quindi $q_0 = A$
- Gli stati finali sono tutti quelli indicati da un "cerchio" aggiuntivo, quindi $F = \{B\}$

Un esempio

Dalla definizione al diagramma degli stati

- $Q = \{A, B, C, D\}$
- $\Sigma = \{0,1\}$
- $q_0 = A$
- $F = \{A, C\}$

δ	**0**	**1**
A	B	A
B	C	A
C	C	D
D	D	D



Computazione

Per automi a stati finiti

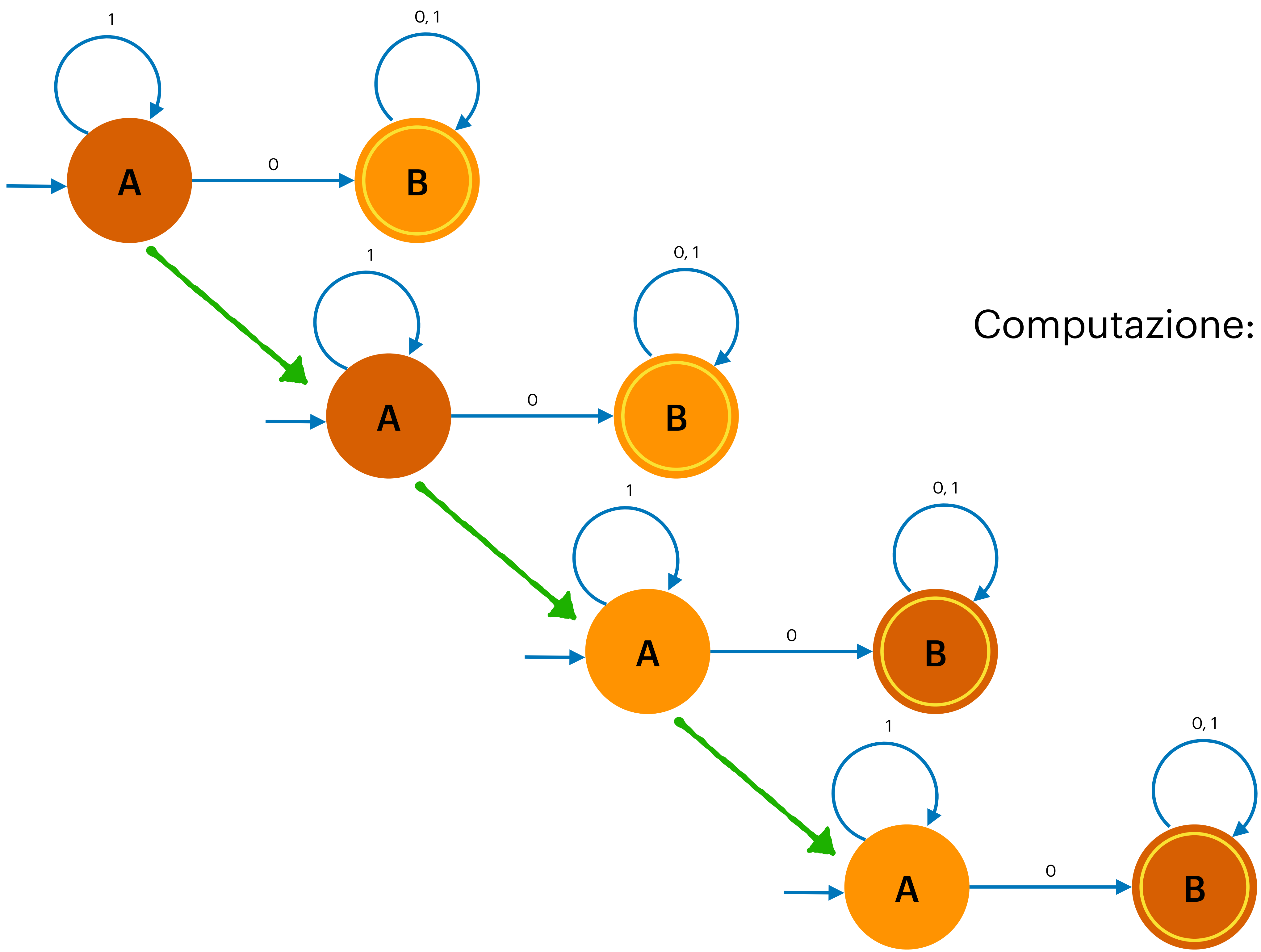
Sia $M = (Q, \Sigma, \delta, q_0, F)$ un automa a stati finiti e sia $w = w_1 w_2 \cdots w_n$ una stringa in $\Sigma^\star$.

Diciamo che M accetta w se esiste una sequenza di stati $r_0, r_1, \ldots, r_n$ tale per cui:

1. $r_0 = q_0$ Il primo stato della sequenza è lo stato iniziale
2. $\delta(r_i, w_{i+1}) = r_{i+1}$ per $0 \leq i < n$ Se siamo nello stato r_i e leggiamo il simbolo w_{i+1} lo stato successivo è dato dalla funzione di transizione
3. $r_n \in F$ L'ultimo stato della sequenza è accettante

Esempio di computazione

Parola in ingresso: 101



Computazione: A, A, B, B

Linguaggi riconosciuti da automi

Se abbiamo un automa a stati finiti $M = (Q, \Sigma, \delta, q_0, F)$ possiamo costruire il linguaggio

$L(M) = \{w \in \Sigma^\star : w$ è accettata da $M\}$

$L(M)$ è il linguaggio riconosciuto dall'automa M

Un linguaggio L è un **linguaggio regolare** se esiste un automa M tale per cui $L = L(M)$, ovvero M riconosce il linguaggio L

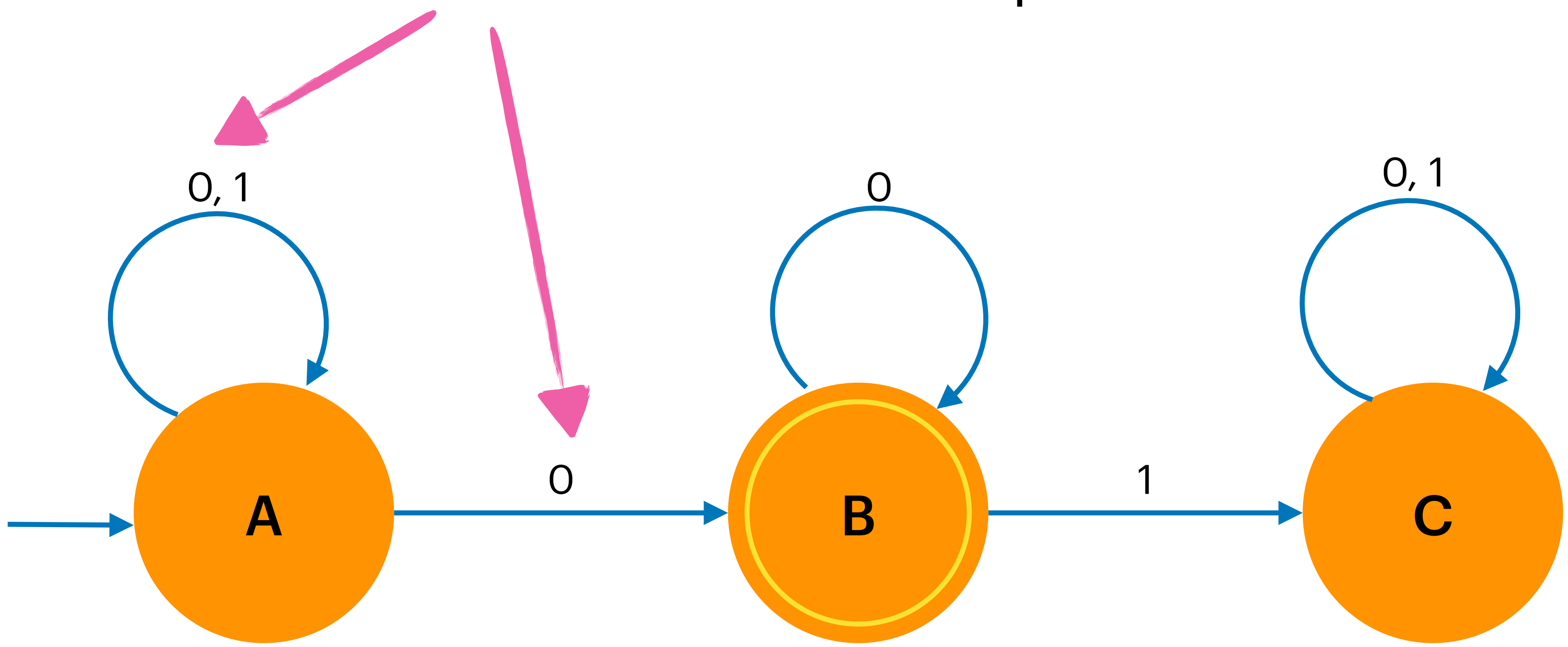
Automi non deterministici

Quando sono possibili più scelte

- Negli automi visti fino ad ora (automi *deterministici*) esiste una sola scelta possibile per ogni combinazione di stato e simbolo dell'alfabeto
- Ma se fossero possibili più scelte?
- E.g., Se siamo nello stato q_2 e leggiamo il simbolo b possiamo scegliere di cambiare stato in q_3 oppure q_4
- In quel caso avremmo degli automi **non deterministici**

Automi non deterministici

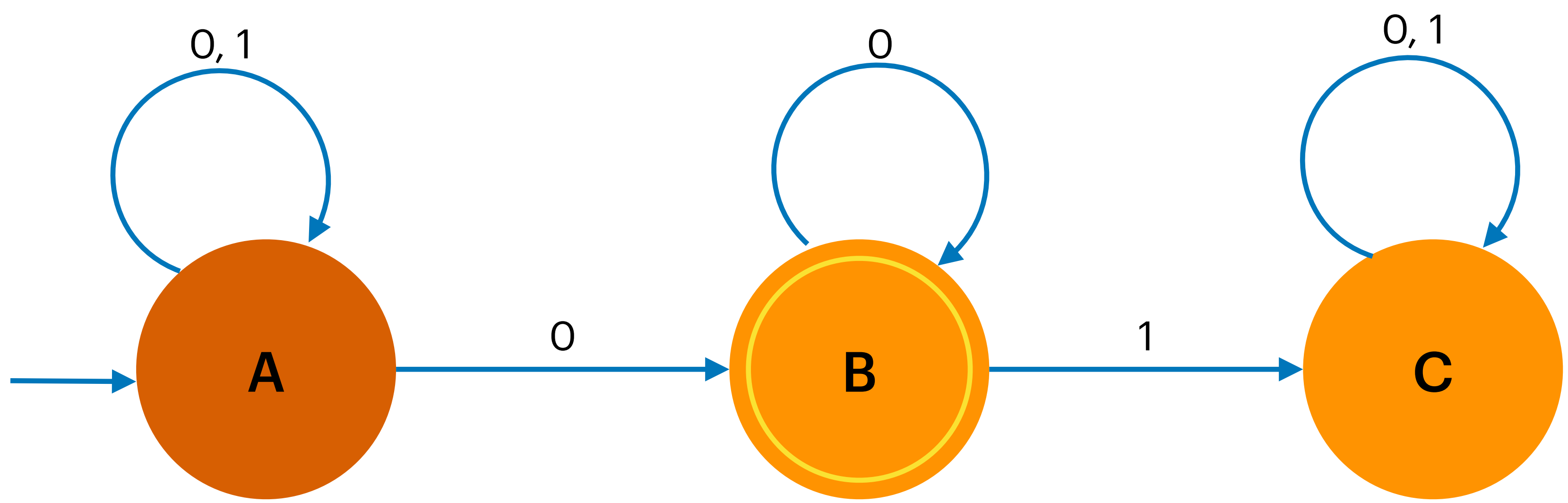
Notare come leggendo 0
vi siano **due** scelte possibili



Quale scelta fare?
Come possiamo capire se una parola viene accettata o rifiutata?

Automi non deterministici

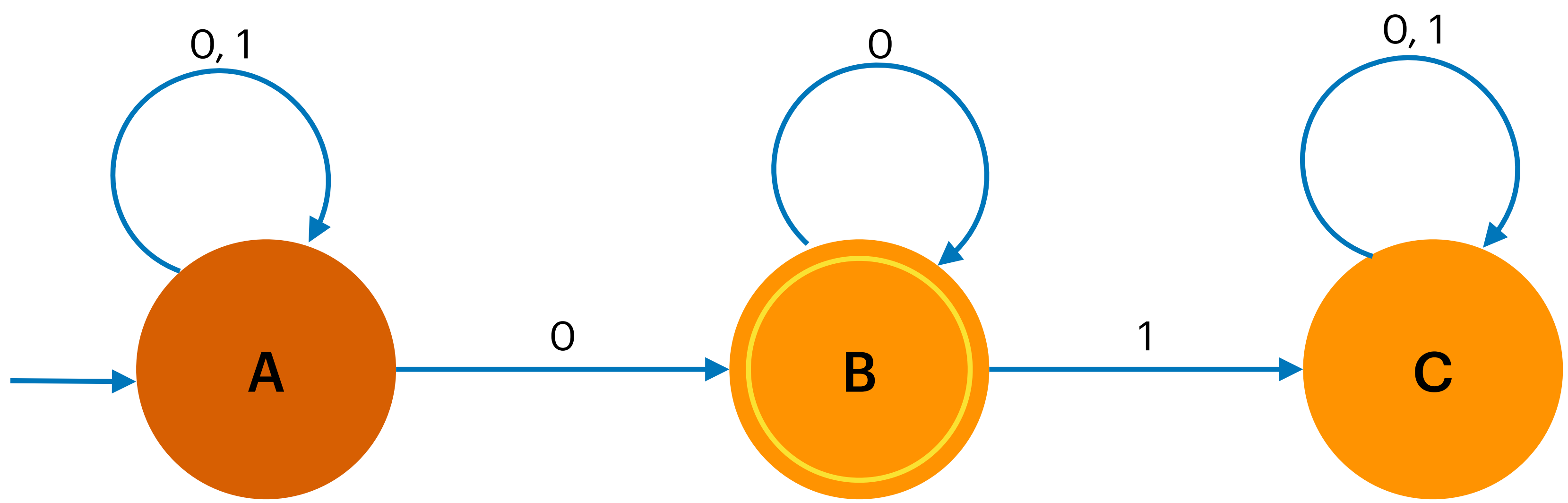
Parola in ingresso: 01010



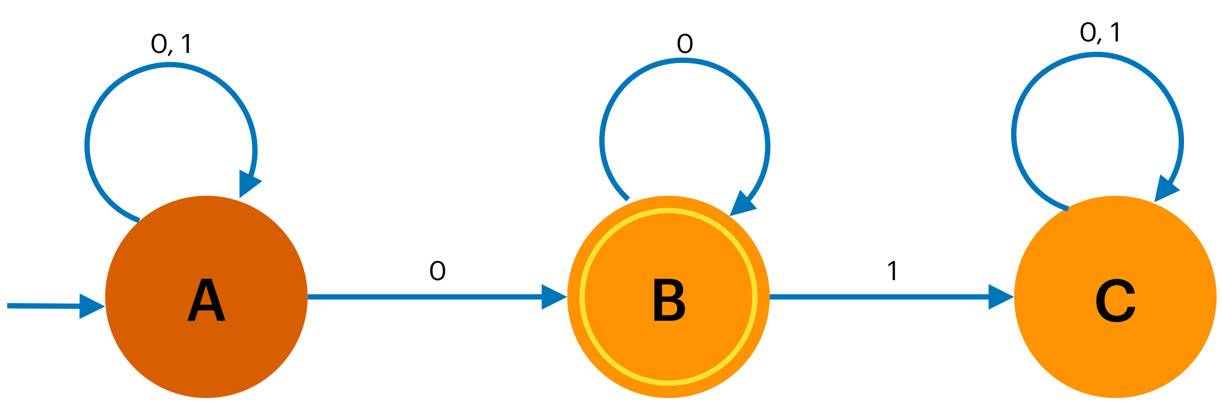
Già al primo simbolo ci sono due scelte possibili:
Possiamo rimanere nello stato **A** oppure muoverci nello stato **B**

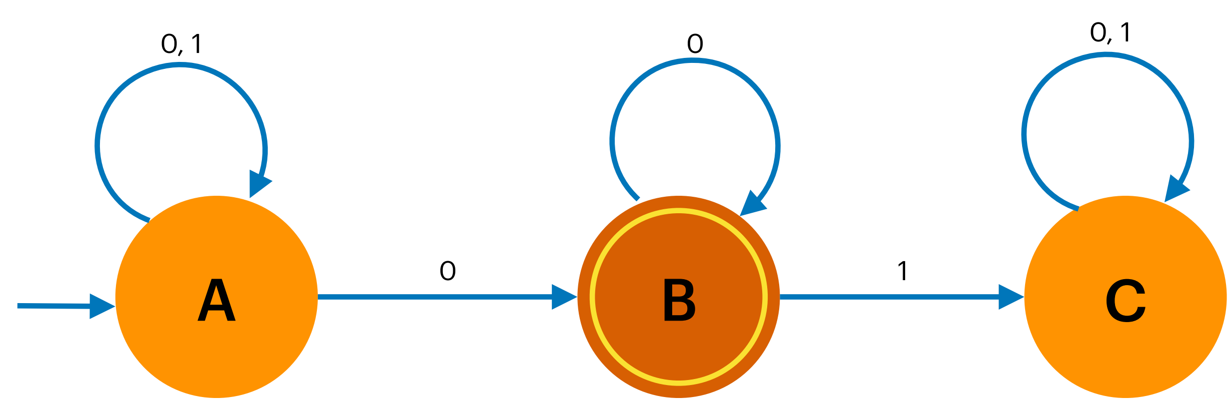
Automi non deterministici

Parola in ingresso: 01010



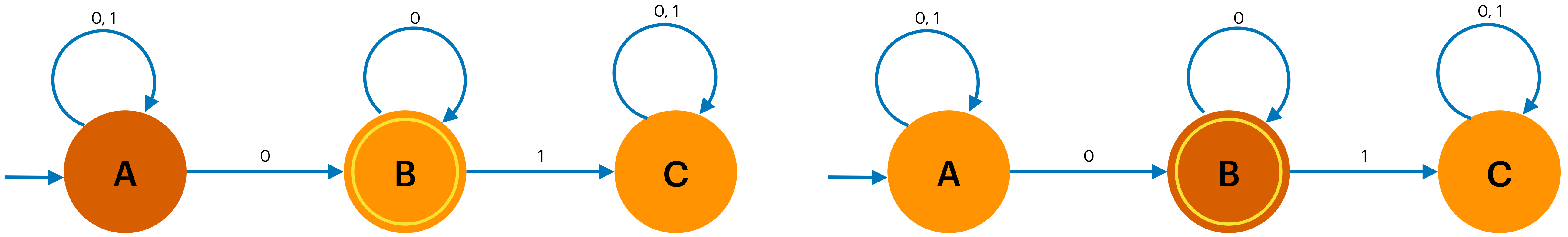
Abbiamo due computazioni diverse a seconda della scelta effettuata



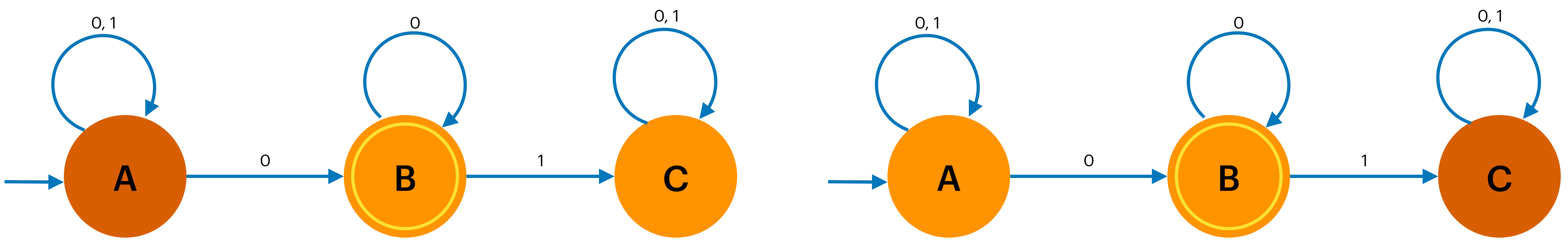


Automi non deterministici

Parola in ingresso: 01010

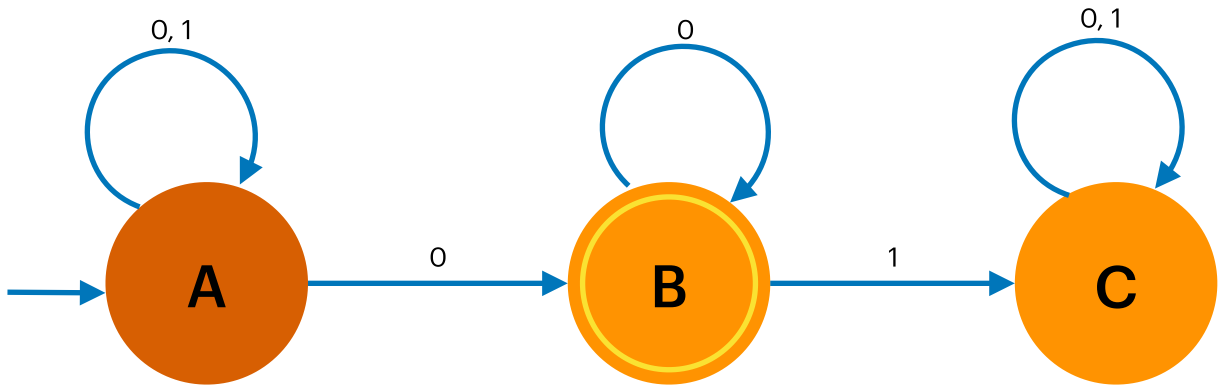


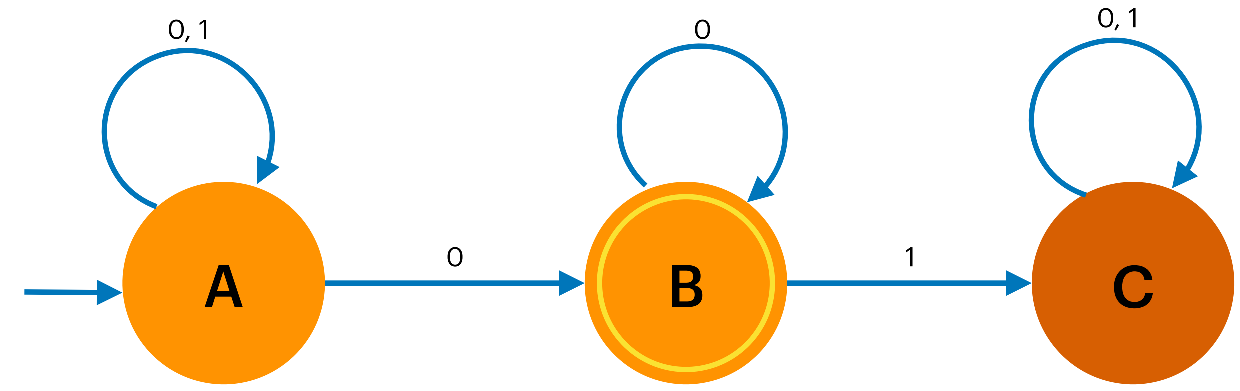
Portiamo avanti
le due computazioni
“in parallelo”



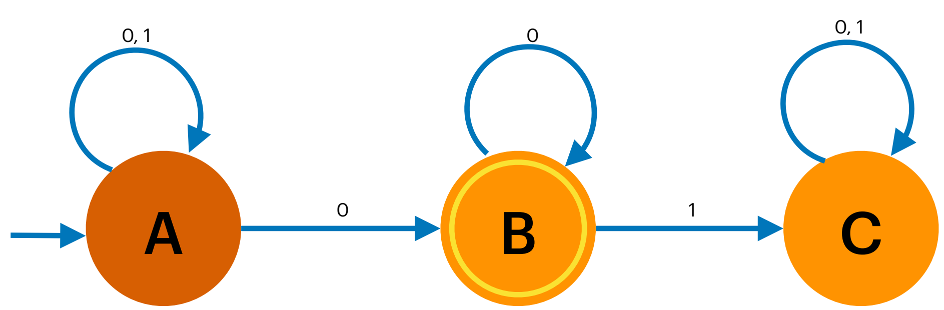
Automi non deterministici

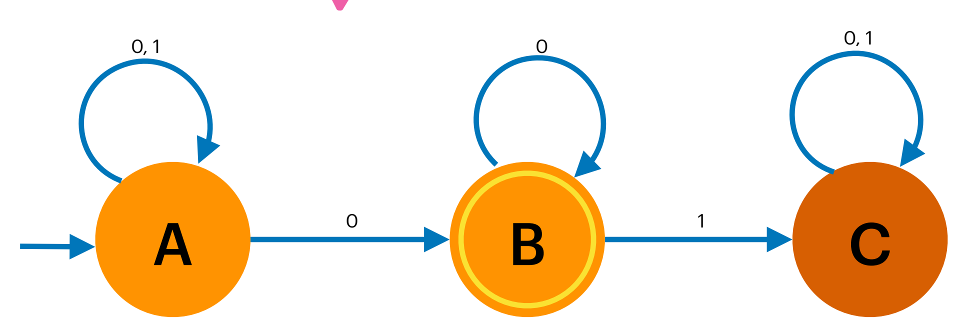
Parola in ingresso: 01010

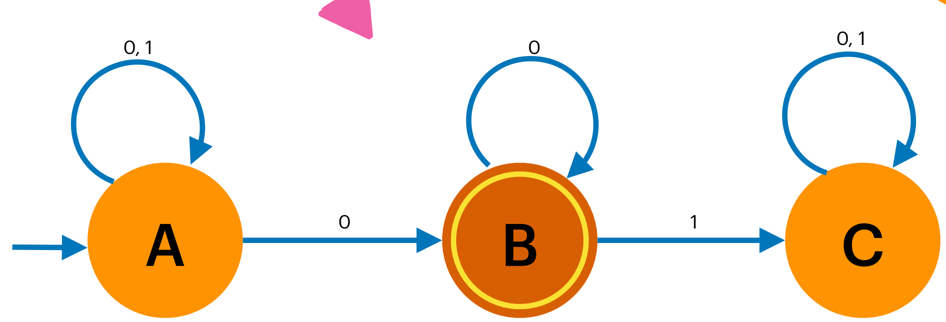




Abbiamo ancora una scelta non deterministica

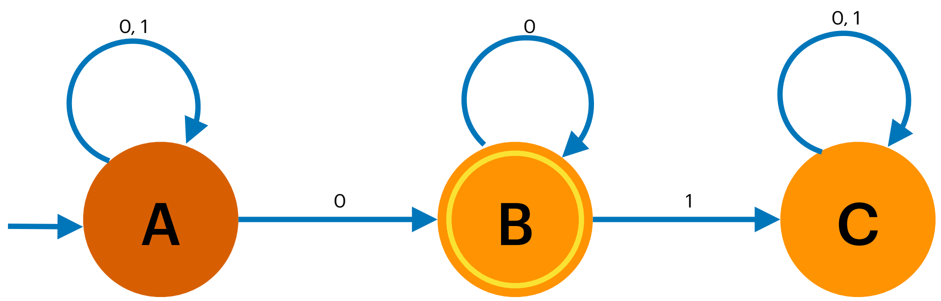


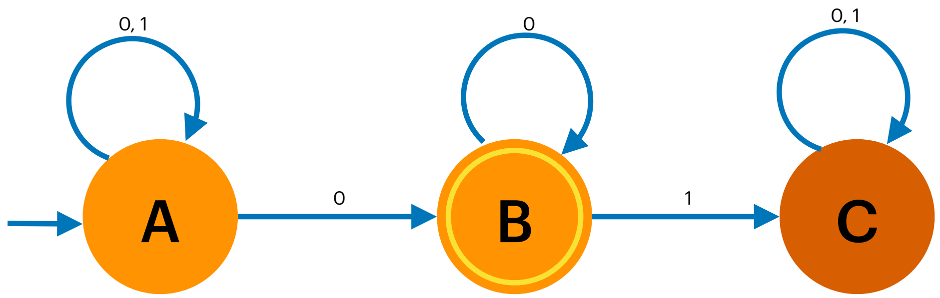


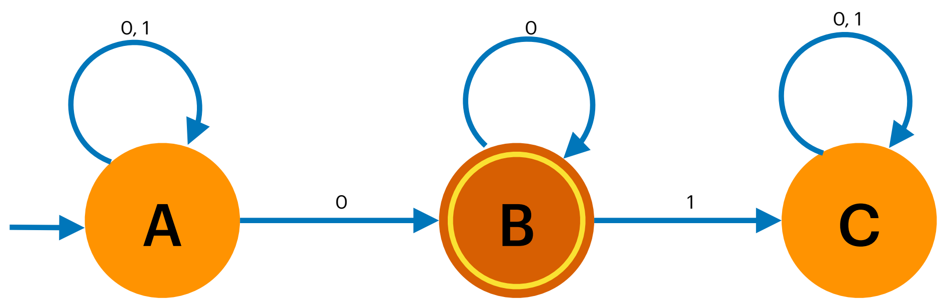


Automi non deterministici

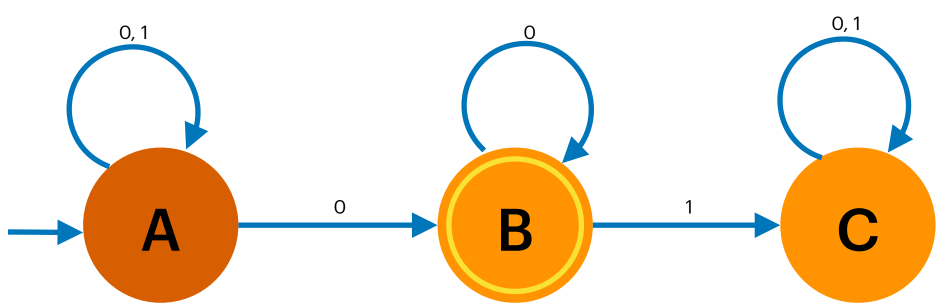
Parola in ingresso: 01010

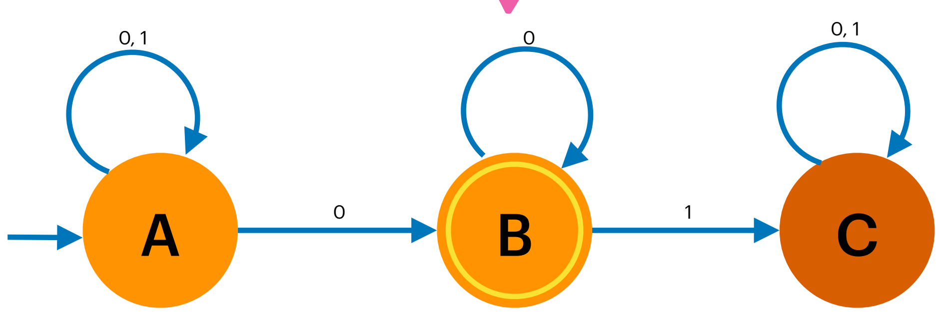


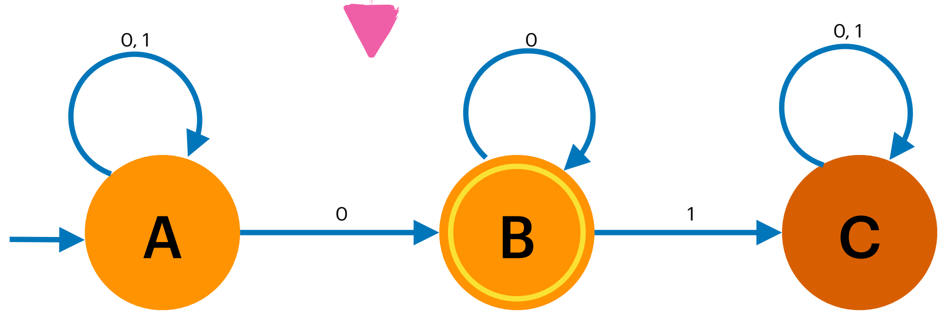




Continuiamo “in parallelo”

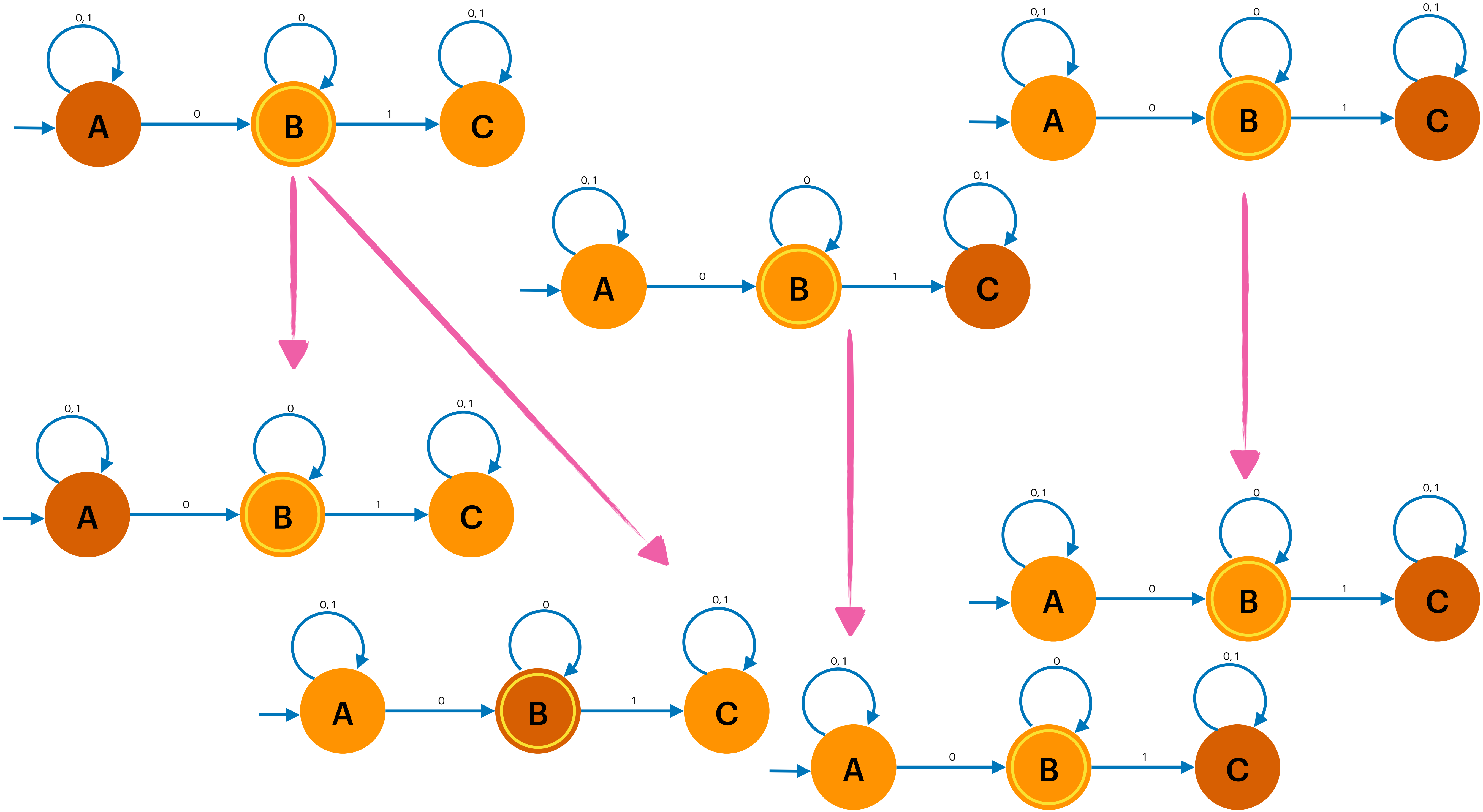






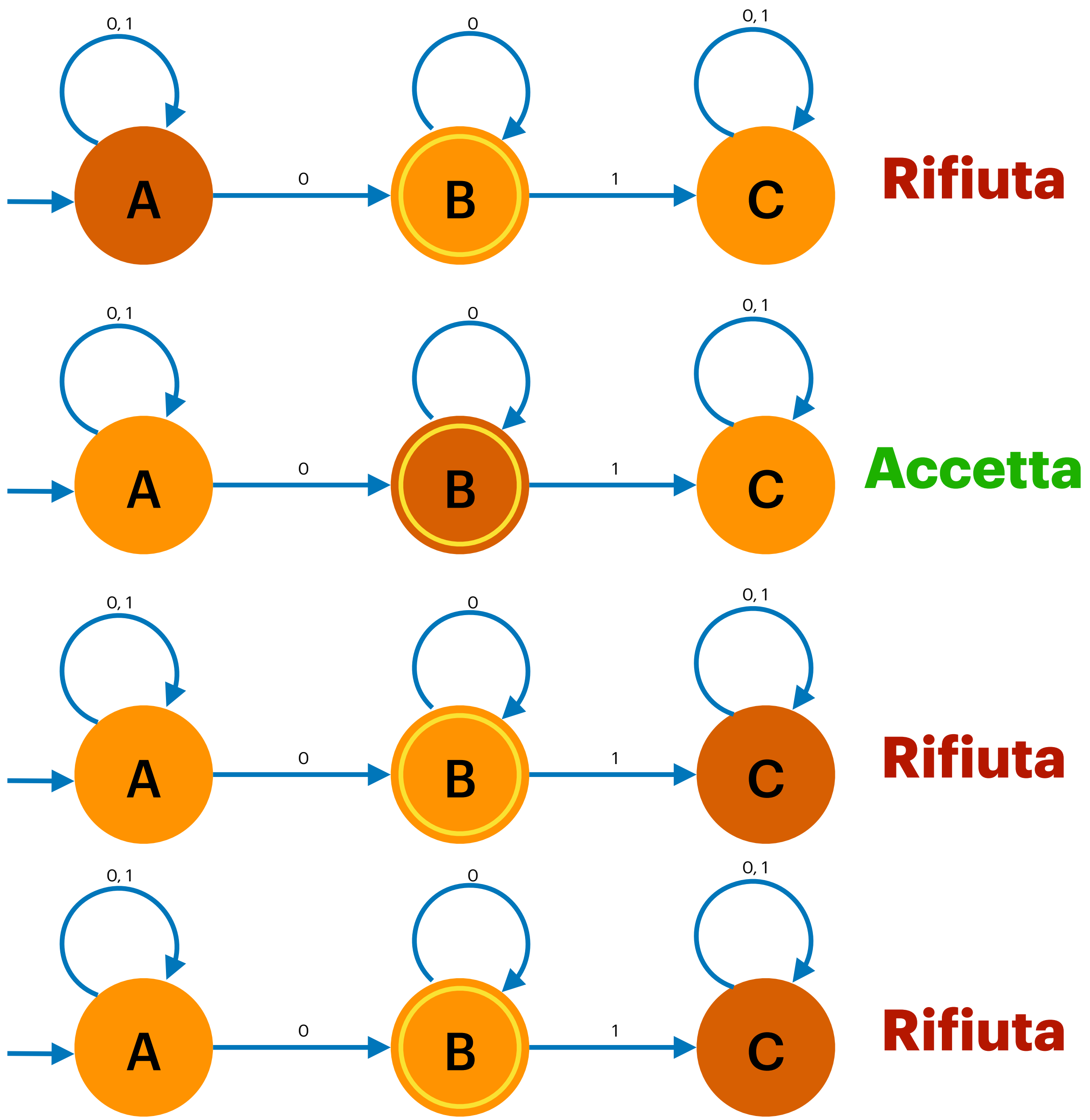
Automi non deterministici

Parola in ingresso: 01010



Automi non deterministici

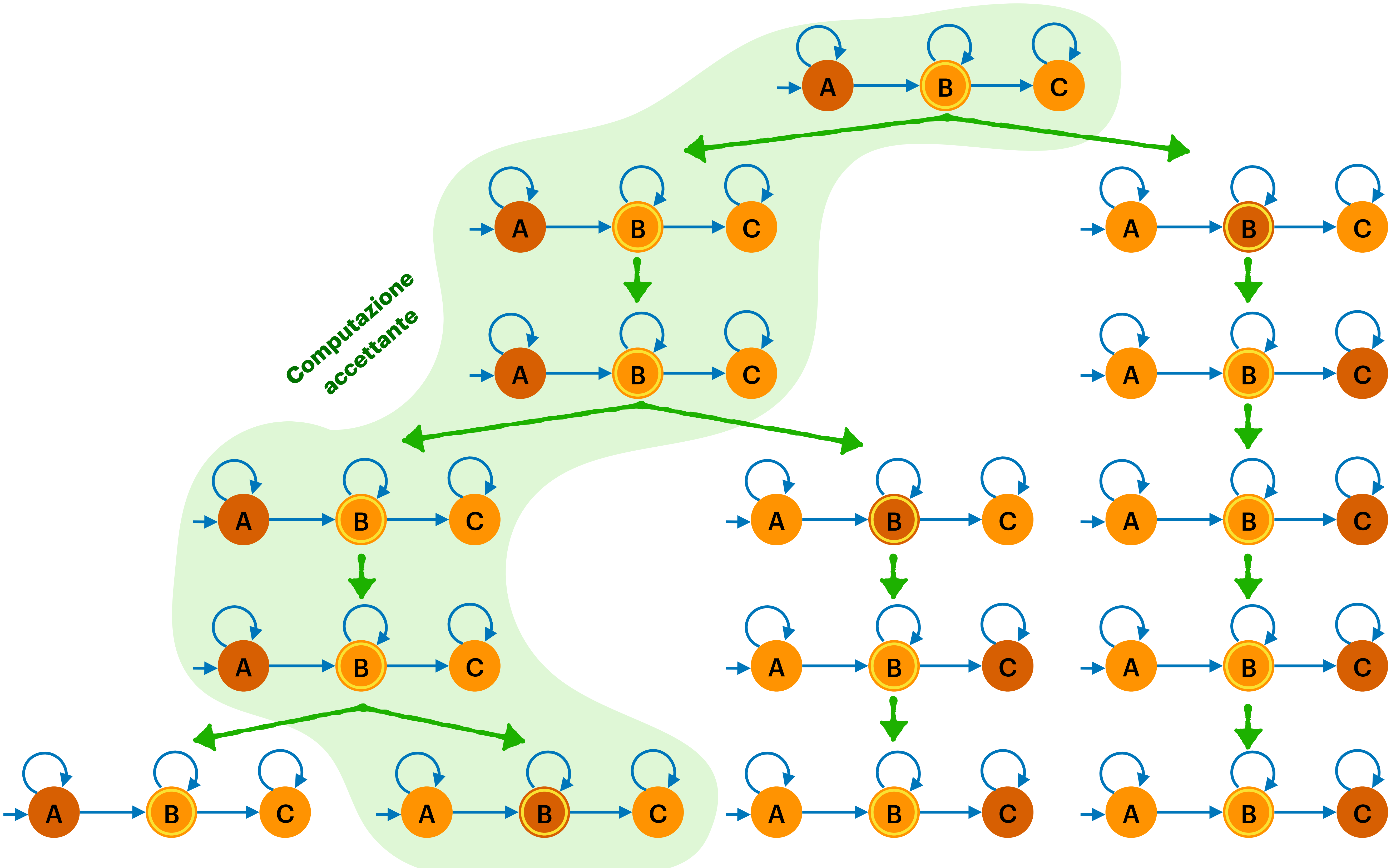
Parola in ingresso: 01010



La parola è accettata o rifiutata?

Se **esiste** una computazione accettante allora la parola è **accettata**

Albero delle computazioni

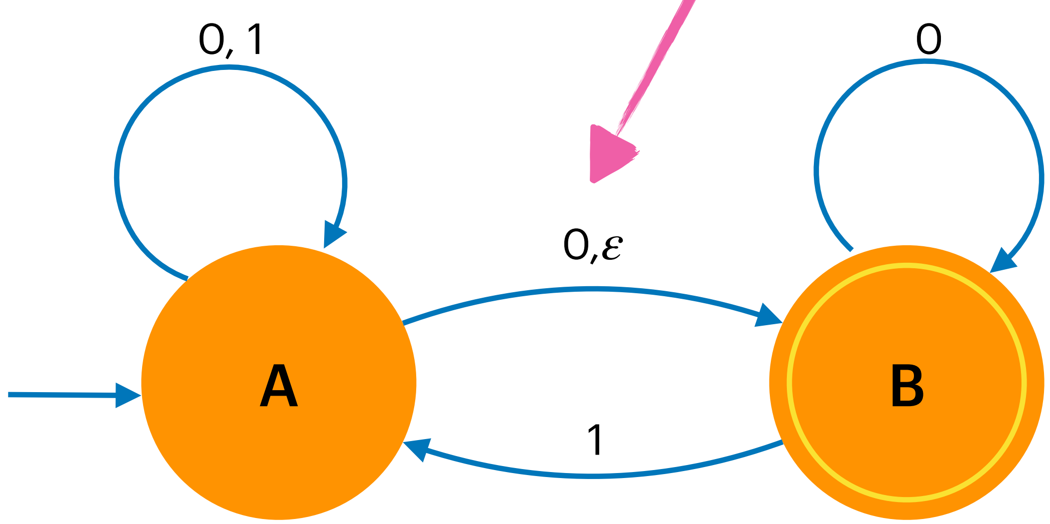


ε-transizioni
Muoversi senza leggere

- Solitamente gli automi non deterministici hanno anche le "ε-transizioni"
- Quando un arco è etichettato con ε allora è possibile percorrerlo **senza** leggere un simbolo
- E.g., siamo nello stato q_2 che è collegato allo stato q_3 con una ε-transizione: possiamo passare a q_3 senza proseguire nella lettura della parola

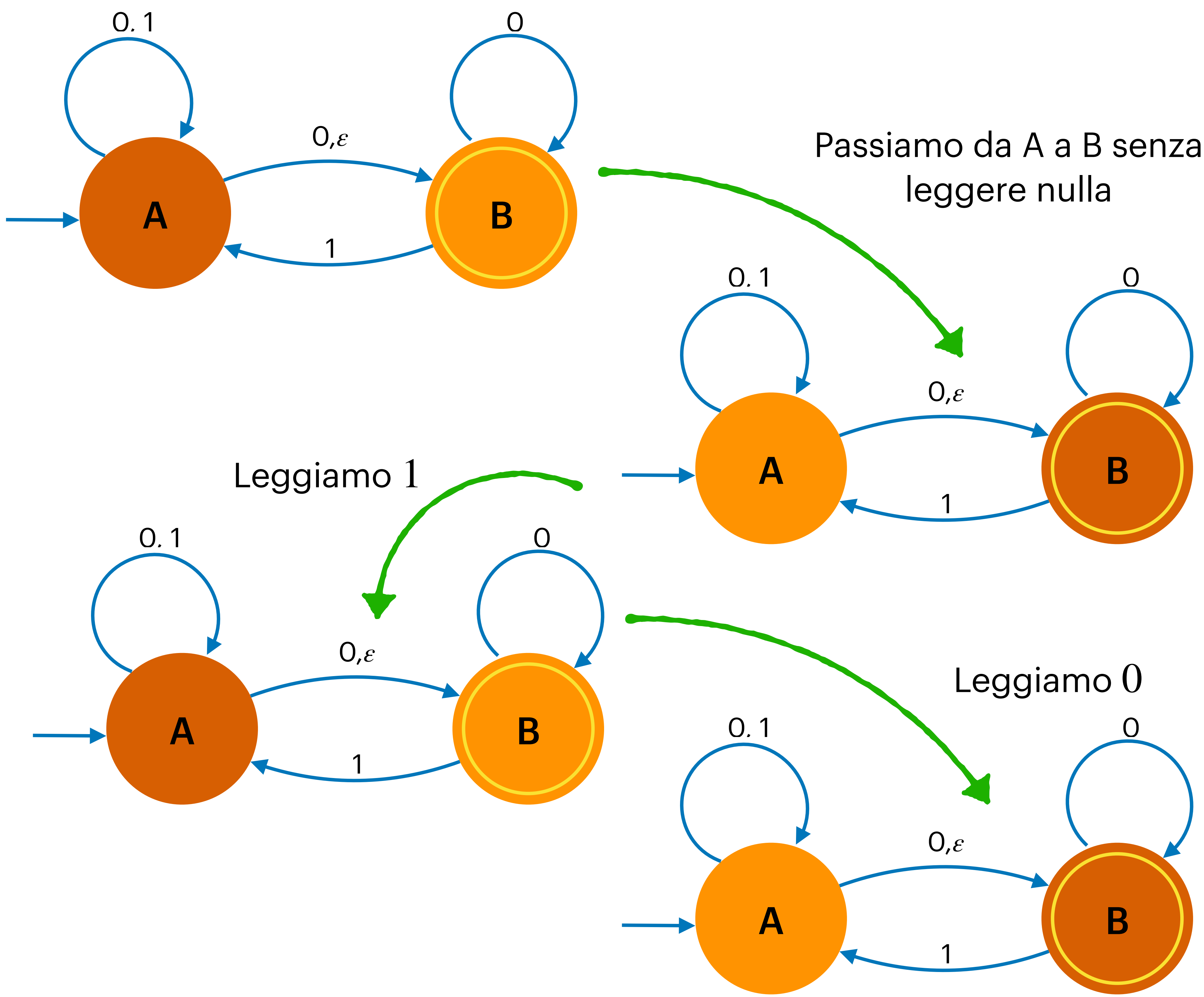
ε-transizioni

Notare come sia possibile passare nello stato **B** prima ancora di aver letto un qualsiasi simbolo



Vediamo un esempio di computazione con $w = 10$

ε-transizioni



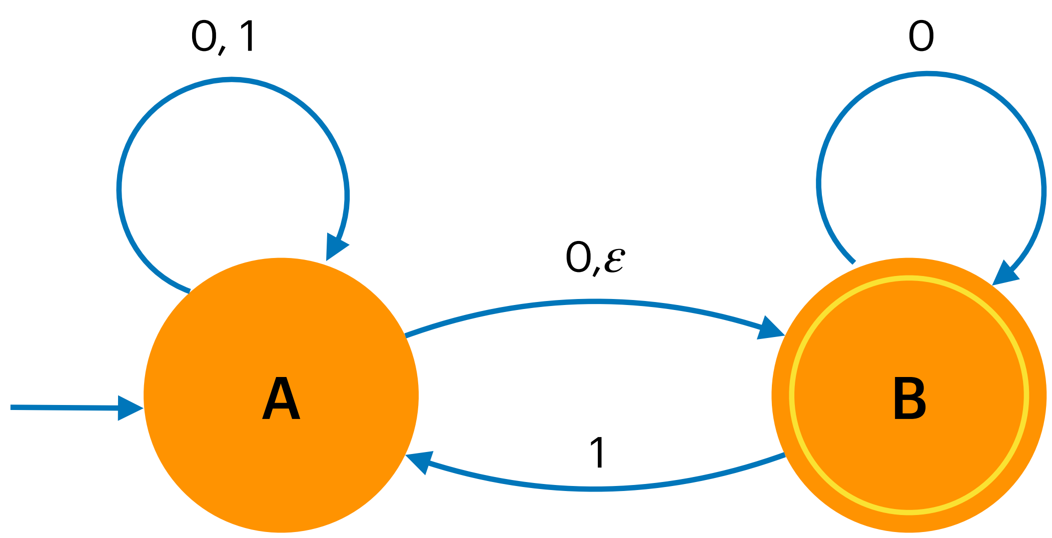
Automi a stati finiti non deterministici

Un **automa a stati finiti non deterministico** è una quintupla $(Q, \Sigma, \delta, q_0, F)$ dove:

1. Q è un insieme finito chiamato insieme degli **stati**,
2. Σ è un insieme finito chiamato **alfabeto**,
3. $\delta\colon\ Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ è la **funzione di transizione**,
4. $q_0 \in Q$ è lo **stato iniziale**,
5. $F \subseteq Q$ è l'insieme degli **stati accettanti** (o stati finali).

Un esempio

Dal diagramma degli stati alla definizione



- La funzione di transizione δ può essere definita da una tabella:

	0	**1**	ε
A	{A, B}	{A}	{B}
B	{B}	{A}	{}

- Ha come insieme degli stati $Q = \{A, B\}$
- L'alfabeto consiste di due soli simboli: $\Sigma = \{0,1\}$
- Lo stato iniziale è indicato da una freccia entrante, quindi $q_0 = A$
- Gli stati finali sono tutti quelli indicati da un "cerchio" aggiuntivo, quindi $F = \{B\}$

Computazione

Per automi a stati finiti non deterministici

Sia $M = (Q, \Sigma, \delta, q_0, F)$ un automa a stati finiti e sia $w = w_1 w_2 \cdots w_n$ una stringa in $\Sigma^\star$.

Diciamo che M accetta w se esiste un modo di scrivere w come $w = y_1 y_2 \cdots y_m$ con $y_i \in \Sigma \cup \{\varepsilon\}$ per $1 \leq i \leq m$ e una sequenza di stati $r_0, r_1, \ldots, r_m$ tale per cui:

Inserimento di ε per le ε-transizioni

1. $r_0 = q_0$ — Il primo stato della sequenza è lo stato iniziale
2. $r_{i+1} \in \delta(r_i, y_{i+1})$ per $0 \leq i < m$ — Se siamo nello stato r_i e leggiamo il simbolo y_{i+1} lo stato successivo è uno dei possibili indicati dalla funzione di transizione
3. $r_m \in F$ — L'ultimo stato della sequenza è accettante

Glossario

E una domanda

- **DFA** (Deterministic Finite Automaton)
  automa a stati finiti deterministico

- **NFA** (Nondeterministic Finite Automaton)
  automa a stati finiti non deterministico

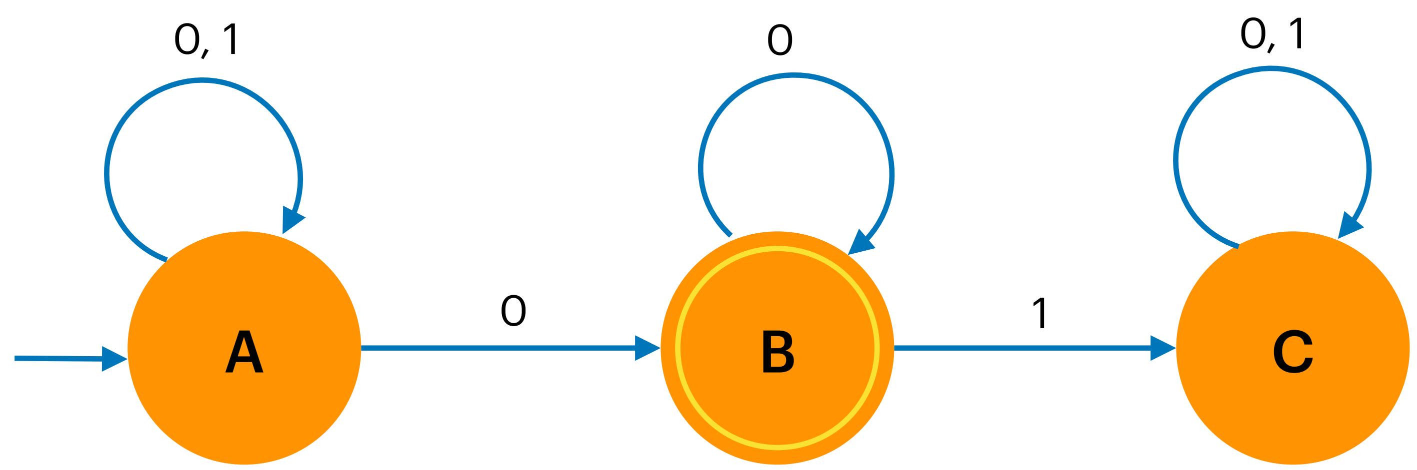
- **Linguaggio regolare** (più raramente “razionale”)
  un linguaggio per il quale esiste un DFA che lo riconosce

- E i linguaggi riconosciuti dai NFA? Sono di più?

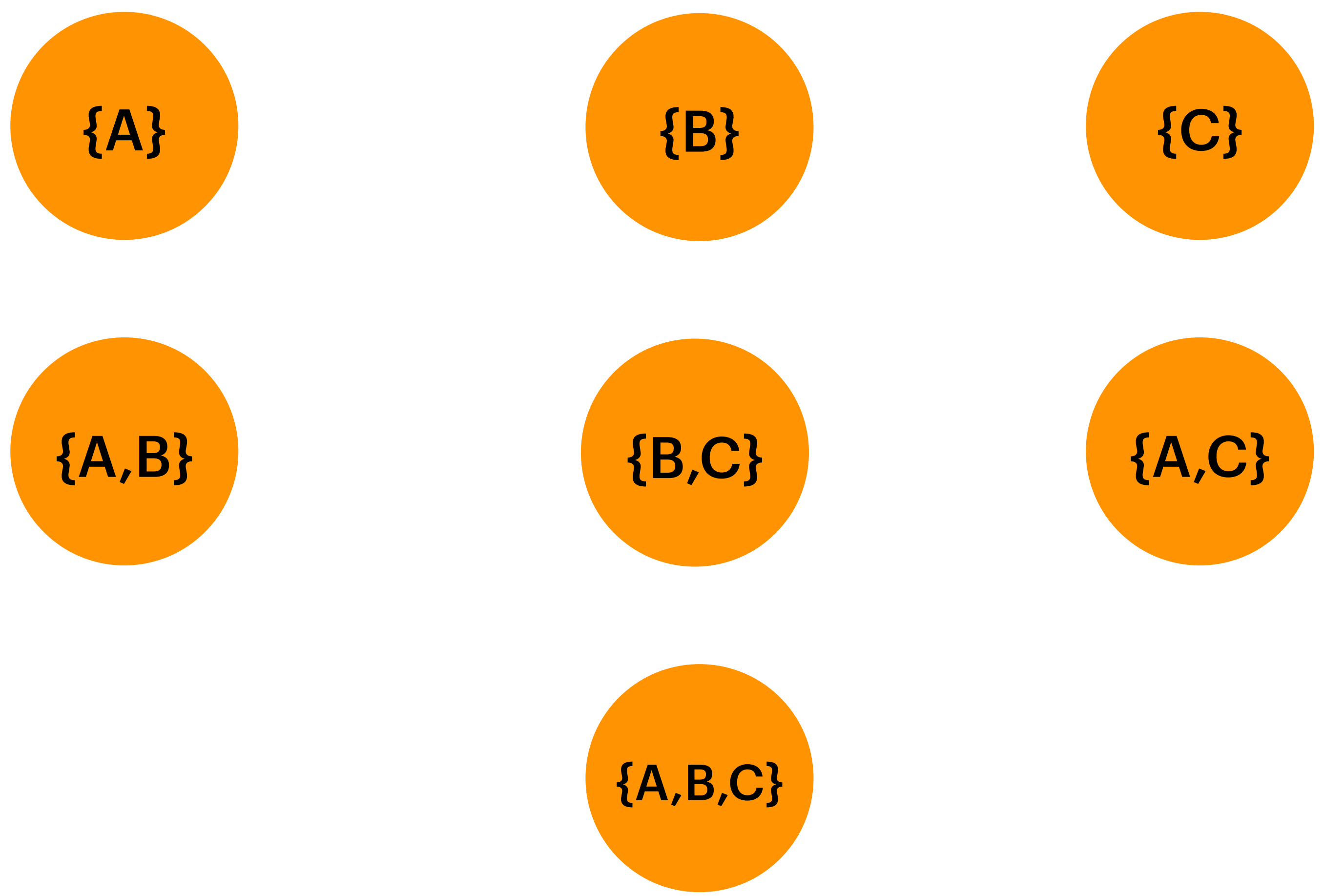
Equivalenza di NFA e DFA

- Si mostra che DFA e NFA riconoscono gli stessi linguaggi
- Chiaramente per ogni linguaggio riconosciuto da un DFA esiste anche un NFA che lo riconosce (lo stesso!)
- E quando abbiamo un NFA?
- L'idea della dimostrazione è che se un NFA N ha insieme di stati Q allora possiamo costruire un DFA M che usa 2^Q come insieme degli stati
- Uno stato di M rappresenta l'insieme degli stati in cui si può trovare N

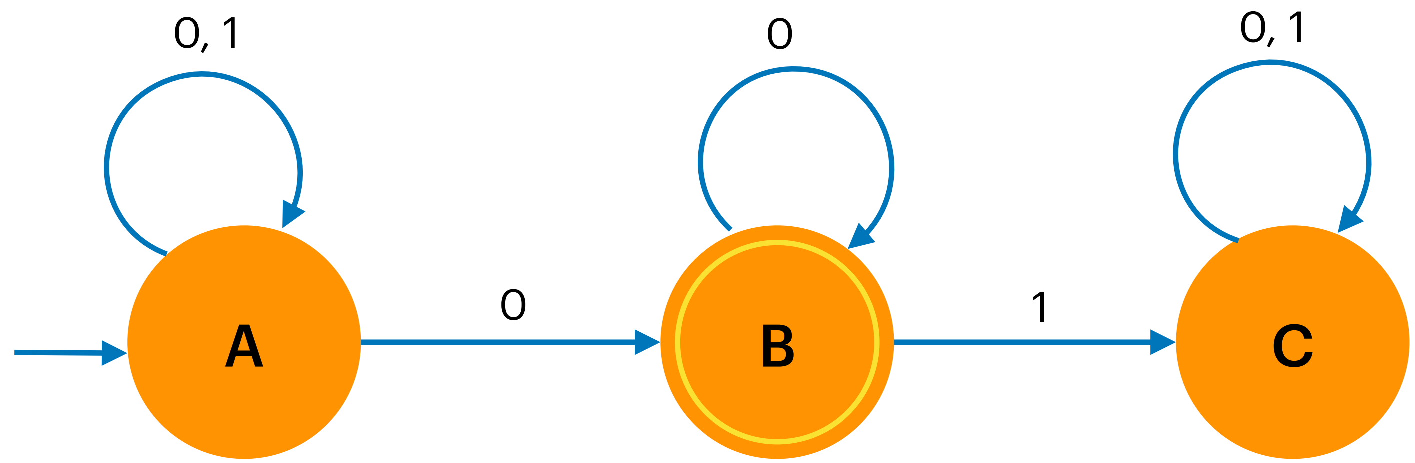
Passaggio da NFA a DFA



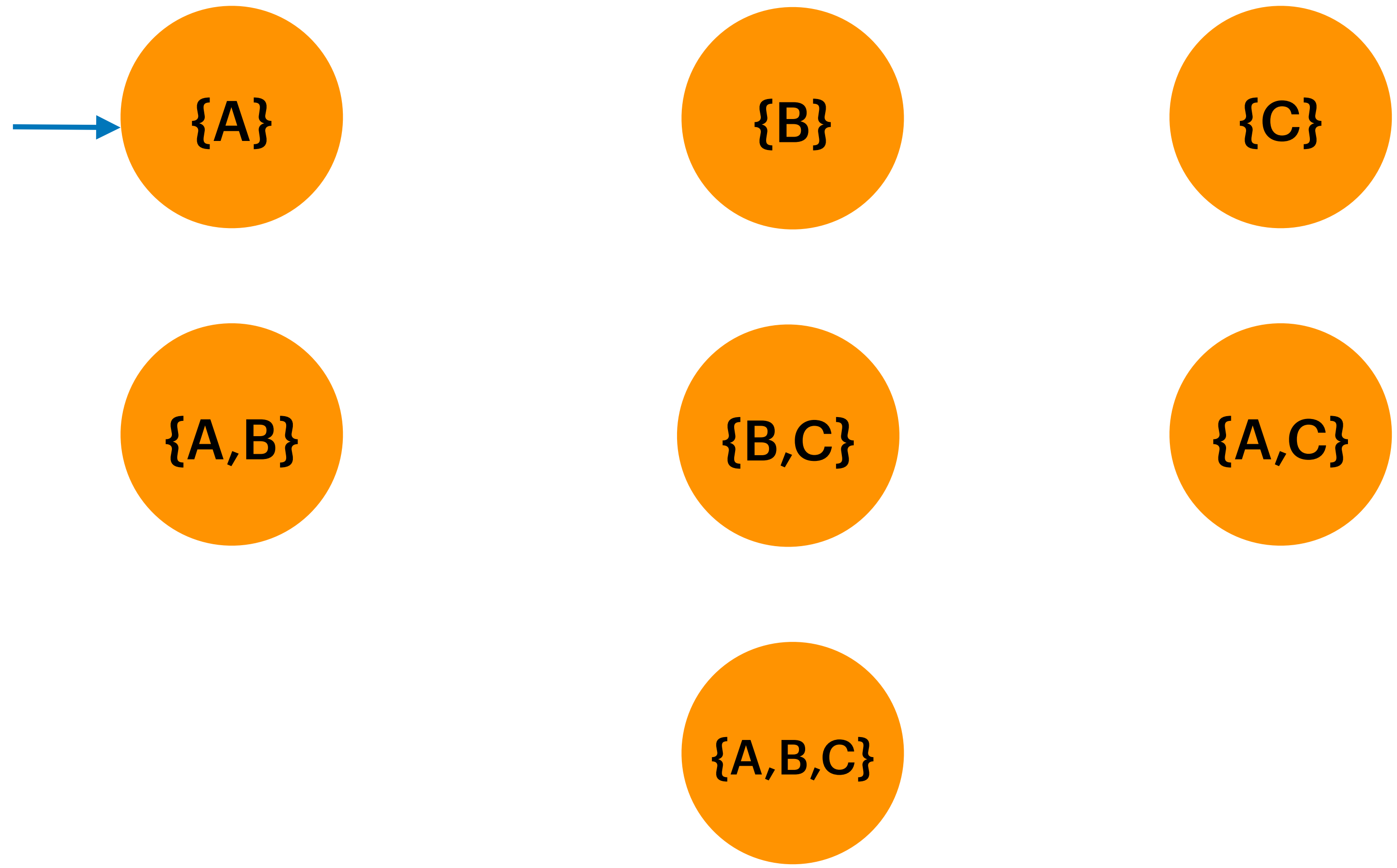
Iniziamo creando uno stato per ogni sottoinsieme non vuoto di $Q = \{A, B, C\}$



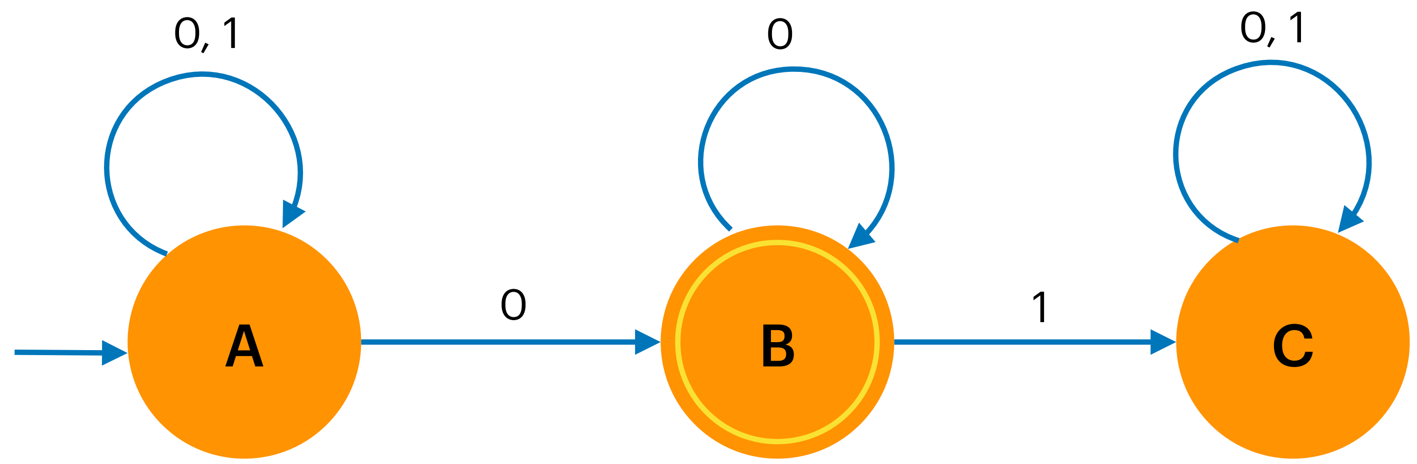
Passaggio da NFA a DFA



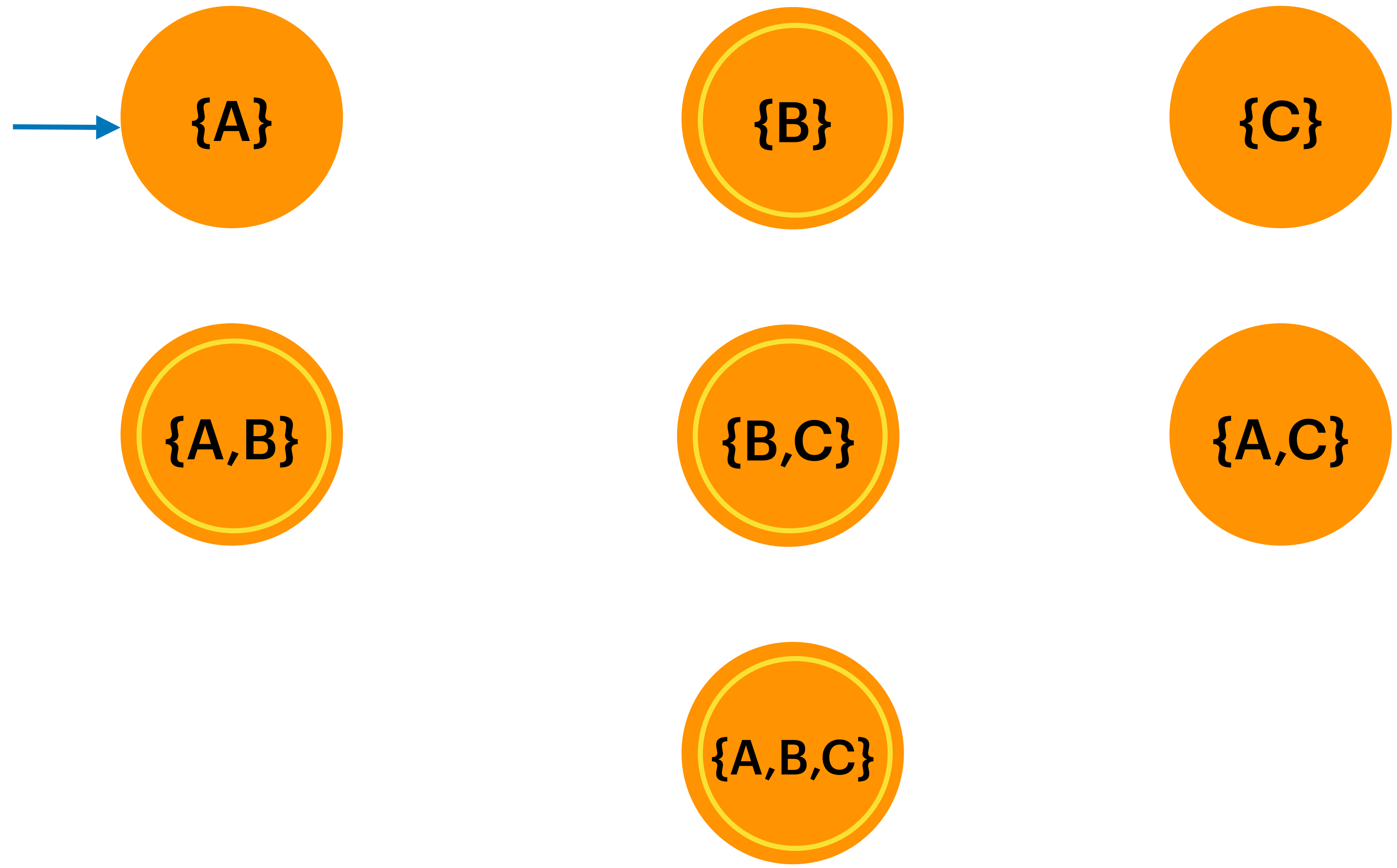
Se A è lo stato iniziale nel NFA allora $\{A\}$ lo è nel DFA



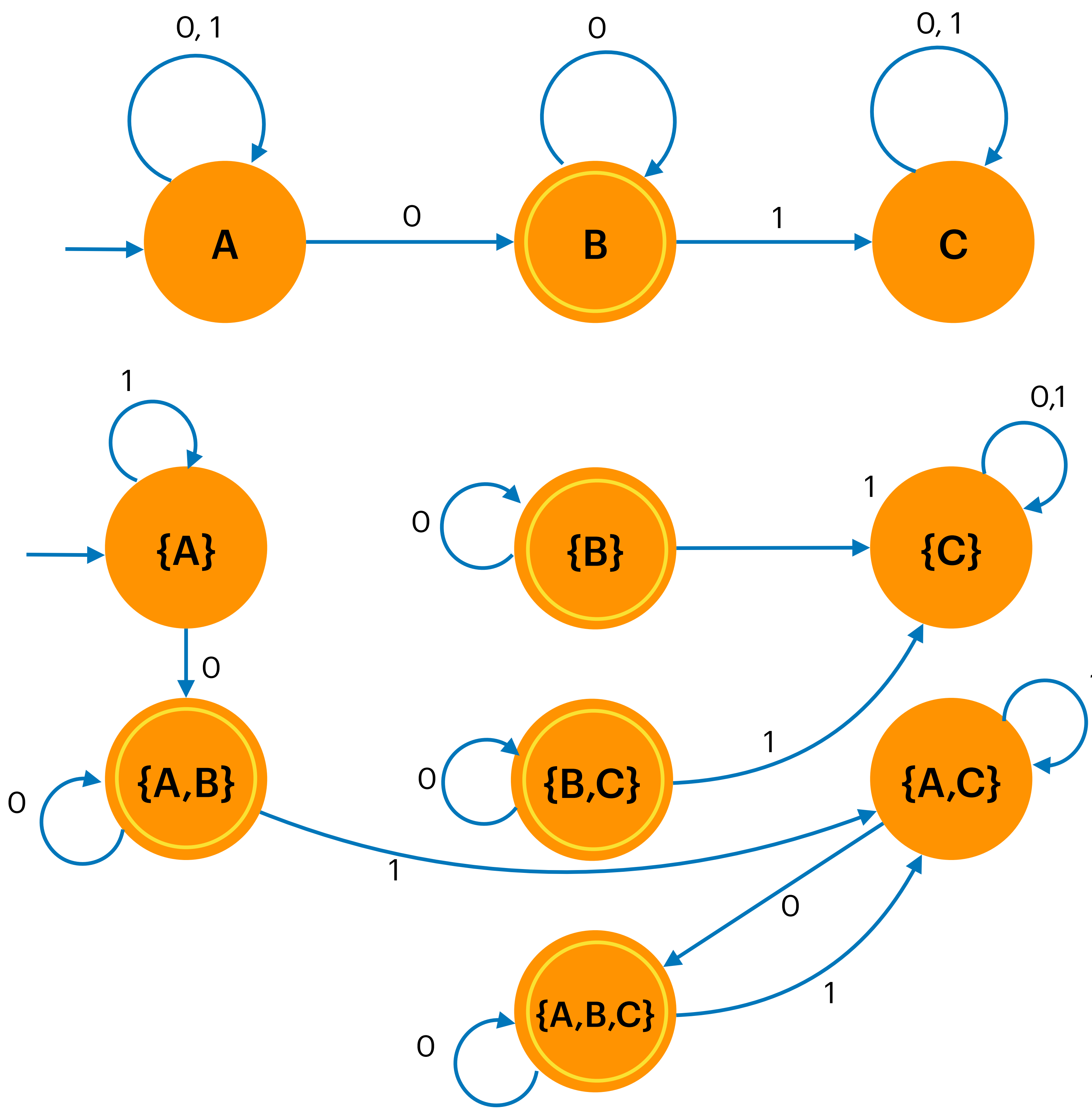
Passaggio da NFA a DFA



Gli stati accettanti del DFA sono quelli che contengono uno stato accettante del NFA



Passaggio da NFA a DFA



Le transizioni si possono ottenere vedendo per ogni insieme di stati del DFA in quali altri stati possiamo andare dopo aver letto un simbolo

Equivalenza di NFA e DFA

Teorema

Per ogni NFA N esiste un corrispondente DFA M tale per cui $L(M) = L(N)$

i.e., M e N riconoscono lo stesso linguaggio

Equivalenza di NFA e DFA

Sia $N = (Q, \Sigma, \delta, q_0, F)$ un NFA che riconosce un linguaggio L

Costruiamo un DFA $M = (Q', \Sigma, \delta', q_0', F')$ che riconosce lo stesso linguaggio L

Iniziamo dal caso più semplice in cui N **non** ha ε-transizioni.

Sia $Q' = 2^Q$, ovvero l'insieme dei sottoinsiemi di Q.

Sia $q_0' = \{q_0\}$ ovvero l'insieme contenente solo lo stato iniziale di N

Sia $F' = \{R \in Q' \colon F \cap R \neq \emptyset\}$, ovvero l'insieme dei sottoinsiemi di Q che contengono almeno uno stato accettante di N

Equivalenza di NFA e DFA

Dato $R \in Q'$ e $a \in \Sigma$ si definisce $\delta'(R, a)$ come

$\delta'(R, a) = \{q \in Q \colon q \in \delta(r, a)$ per qualche $r \in R\}$

Ovvero l'insieme di tutti gli stati raggiungibili a partire da qualche stato in R tramite una transazioni definite da δ.

In modo equivalente avremmo potuto scrivere $\delta'(R, a) = \bigcup_{r \in R} \delta(r, a)$

L'automa deterministico risultante riconosce lo stesso linguaggio di N se N non ha ε-transizioni

Consideriamo ora la presenza di ε-transizioni

Equivalenza di NFA e DFA

Si definisca $E(R)$ per ogni stato $R \in Q'$ come gli stati raggiungibili con zero o più ε-transizioni a partire da stati in R.

Formalmente:
$E(R) = \{q\colon q$ può essere raggiunto da uno stato in R seguendo 0 o più archi etichettati con $\varepsilon\}$

Usando la definizione di $E(R)$ modifichiamo la funzione di transizione come:

$$\delta'(R, a) = \{q \in Q\colon q \in E(\delta(r, a)) \text{ per qualche } r \in R\}$$

Aggiorniamo inoltre lo stato iniziale in modo da contenere non solo q_0 ma anche tutti gli stati raggiungibili con ε-transizioni da esso:
$q_0' = E(\{q_0\})$

Equivalenza di NFA e DFA

La costruzione di M funziona correttamente.

A ogni passo della computazione di M su un input lo stato di M corrisponde al sottoinsieme degli stati in cui N si può trovare in quel momento.

Questo completa la dimostrazione.