Logica e Modelli Computazionali

Automi a Stati Finiti Non Deterministici

Marco Console

Ingegneria Informatica e Automatica, Sapienza Università di Roma

Macchine Deterministiche e Non-Deterministiche

- Gli ASFD sono caratterizzati dalla seguente proprietà: **dato uno stato e un simbolo** dell'input, c'è **solo un possibile stato interno** dell'automa risultante dalla computazione
  - Chiamiamo questa proprietà **determinismo**

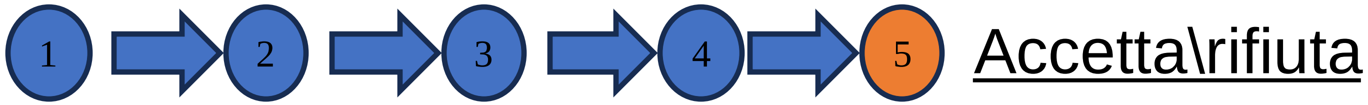
- La proprietà di **determinismo** è cruciale nelle nostre definizioni dei linguaggi regolari
  - **Deriva** dal fatto che le transazioni sono definite da funzioni
  - **Ci assicura** che esiste una unica computazione possibile per ogni stringa di input

- Possiamo rilassare questa assunzione e definire **modelli non deterministici**?
  - Modelli computazionali il cui prossimo stato non è più determinato dall'input e lo stato corrente
  - Tale stato potrebbe essere uno qualunque di quelli presi da uno specifico insieme

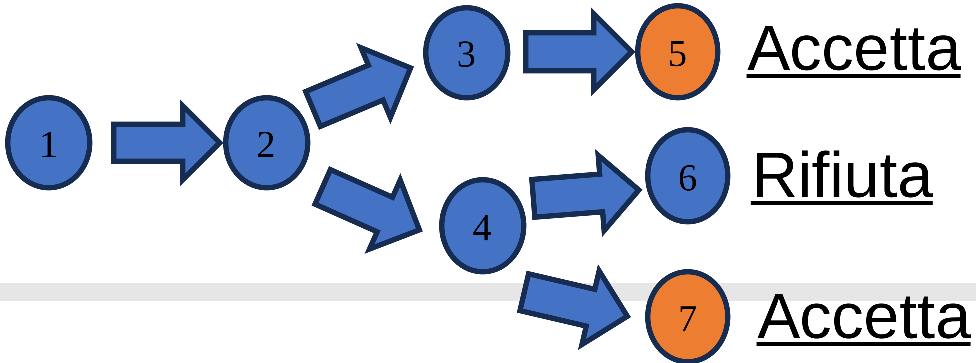
- Le **macchine reali** sono **intrinsecamente deterministiche**
  - Dato un input e uno stato interno, possiamo sempre determinare il risultato della computazione

Computazioni Deterministiche e Non-Deterministiche

- Possiamo immaginare una **computazione deterministica** come una **sequenza di configurazioni** della macchina che stiamo esaminando
  - Nel caso degli automi, tali configurazioni sono rappresentate dagli stati interni della macchina



- Possiamo immaginare una **computazione non-deterministica** come una **albero di configurazioni** della macchina che stiamo esaminando
  - La prossima configurazione **non è determinata a priori** dallo stato corrente e il simbolo dell'input
  - Nel caso degli automi, tali configurazioni possono ancora essere rappresentate dagli stati ma …
  - Una domanda importante rimane aperta: **quando accettiamo**?



Computazioni Non-Deterministiche: Intuizione

- Un modello non-deterministico potrebbe **rappresentare** macchine che
  - **Eseguono computazioni in parallelo** (una per ogni possibile ramo dell'albero di computazione) **oppure**
  - **"Indovinano" alcune proprietà dell'input** (alcuni problemi hanno una naturale soluzione in questi termini)

- **Esempio**: Riconoscere le stringhe $\boldsymbol{S}$ **per cui** $|\boldsymbol{S}|$ **è multiplo di due oppure multiplo di tre**
  1. Sotto l'assunzione che l'input **non sia multiplo di tre**
     - Esegui un ASFD che riconosce stringhe $\boldsymbol{S}$ **per cui** $|\boldsymbol{S}|$ **è multiplo di due**
  2. Sotto l'assunzione che l'input **non sia multiplo di due**
     - Esegui un ASFD che riconosce stringhe $\boldsymbol{S}$ **per cui** $|\boldsymbol{S}|$ **è multiplo di tre**

- Una modello computazionale in grado di eseguire una computazione non deterministica sembra contro-intuitivo se l'obbiettivo è definire il modello di una macchina reale.
  - Una domanda fondamentale è come **passare da non-determinismo a determinismo** ovvero
  - Possiamo costruire una **macchina deterministica equivalente a una non-deterministica**?
  - **A che prezzo**?

Automa a Stati Finiti Non-Deterministico – Definizione

- **Notazione aggiuntiva**
  - $\mathrm{P}(\mathrm{Q})$ denota **l'insieme delle parti di $\mathbf{Q}$** cioè **la famiglia di tutti i sottoinsiemi di $\mathbf{Q}$**
  - Esempio. $Q = \{A, B, C\}; P(Q) = \{\emptyset, \{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}, \{A, B, C\}\}$

- Un **automa a stati finiti non-deterministico** (**ASFND**) è una **quintupla** $< \Sigma, Q, \delta, I, F >$

1. $\Sigma = \{a_1, \dots, a_n\}$ è l'***alfabeto*** di input;
2. Q è un insieme finito detto ***insieme degli stati*** ;
3. $I \in Q$ è lo ***stato iniziale*** ;
4. $F \subseteq Q$ è un ***insieme degli stati finali***;
5. $\delta: Q \times \Sigma \rightarrow P(Q)$ è una funzione da $Q \times \Sigma$ in $P(Q)$, chiamata ***funzione di transizione***

ASFND – Intuizione

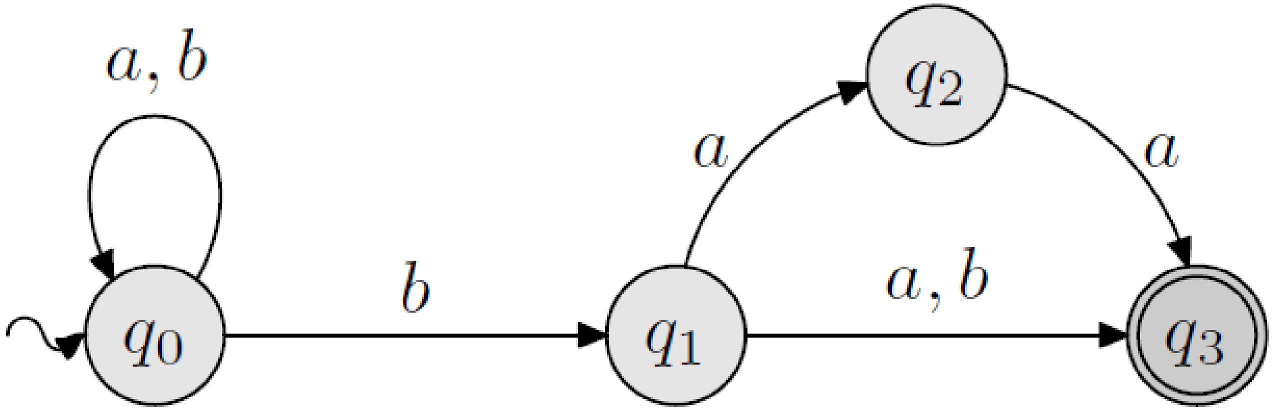
- **Intuitivamente**, le cinque componenti di un ASFND rappresentano
  1. L'insieme dei possibili simboli utilizzati nelle stringhe di input (***alfabeto*** $\boldsymbol{\Sigma}$)
  2. L'insieme dei possibili stati interni della macchina (***insieme degli stati*** $\mathbf{Q}$)
  3. Lo stato interno della macchina da cui parte la computazione (***stato iniziale*** $\boldsymbol{I}$)
  4. L'insieme degli stati in cui la macchina ritorna 1 (***insieme degli stati finali*** $\mathbf{F}$)
  5. La funzione che determina il passaggio di stato **non-deterministico** (***funzione di transizione*** $\boldsymbol{\delta}$)

- L'unica differenza fra ASFND e ASFD è la funzione di transizione infatti
  1. Per un ASFND i possibili stati successivi sono un insieme e
  2. Potremmo non consumare alcun simbolo della stringa iniziale durante la transizione

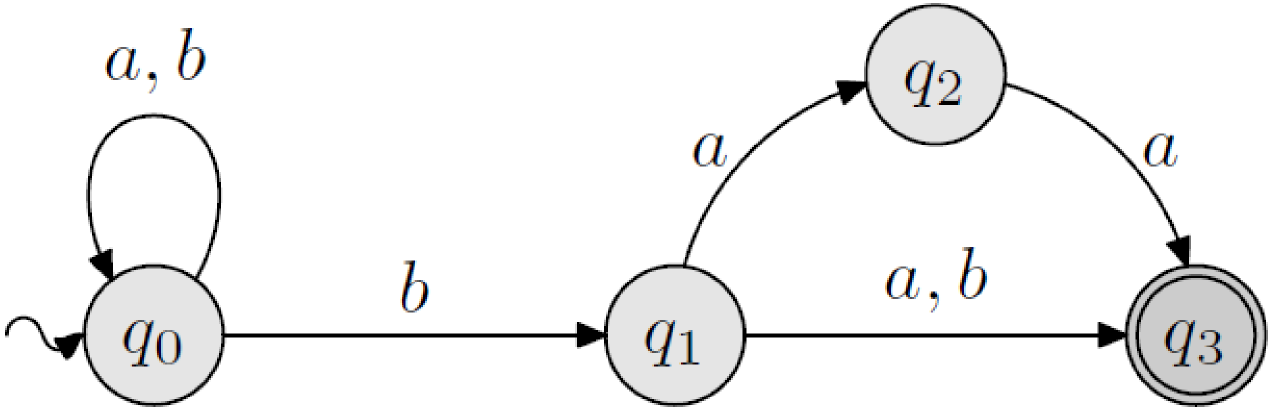
Diagramma degli Stati

- Come per gli ASFD, possiamo rappresentare un ASFND tramite il **diagramma degli stati** (noto anche come **grafo di transizione**)

- I **nodi** rappresentano gli stati e gli **archi** le transizioni, che sono quindi **etichettati** con il carattere la cui lettura determina la transizione (**più archi uscenti dal medesimo nodo etichettati con il medesimo carattere e potremmo avere ϵ**). Lo stato iniziale è rappresentato tramite una freccia, mentre gli stati finali sono rappresentati con un doppio cerchio



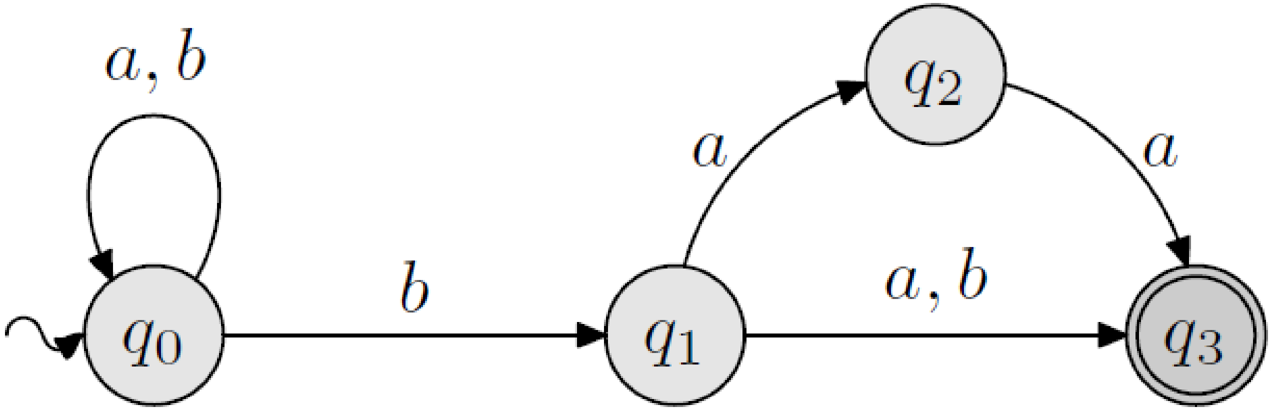
Esempio di Computazione di un ASFND



INPUT

A	B	A	B

Esempio di Computazione di un ASFND



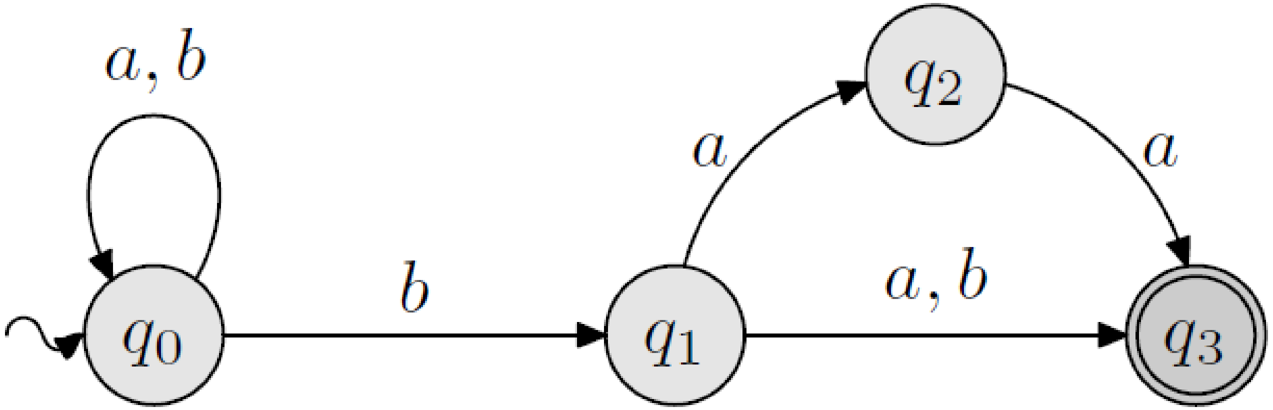
INP

A	B	A	B

STATO CORRENTE

q_0

Esempio di Computazione di un ASFND



INPUT

A	B	A	B

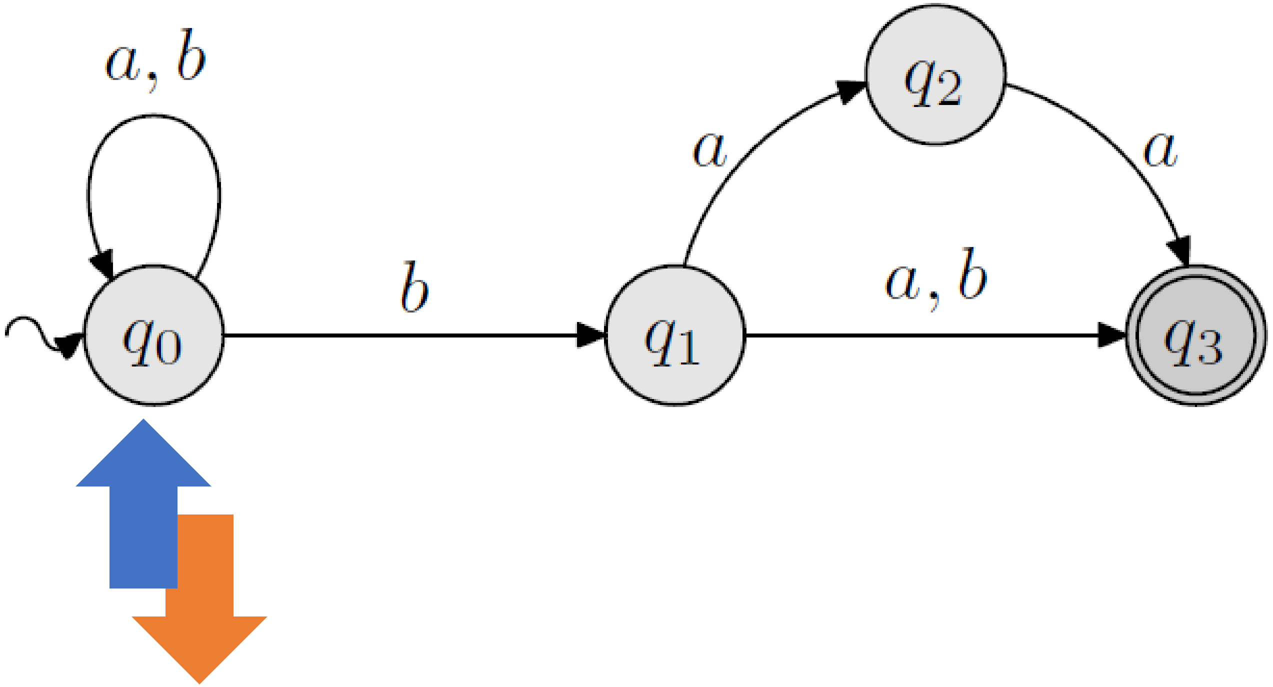
STATO CORRENTE

q_0

<u>**Due scelte possibili per il prossimo stato!**</u>

<u>**Scegliamo q_0**</u>

Esempio di Computazione di un ASFND



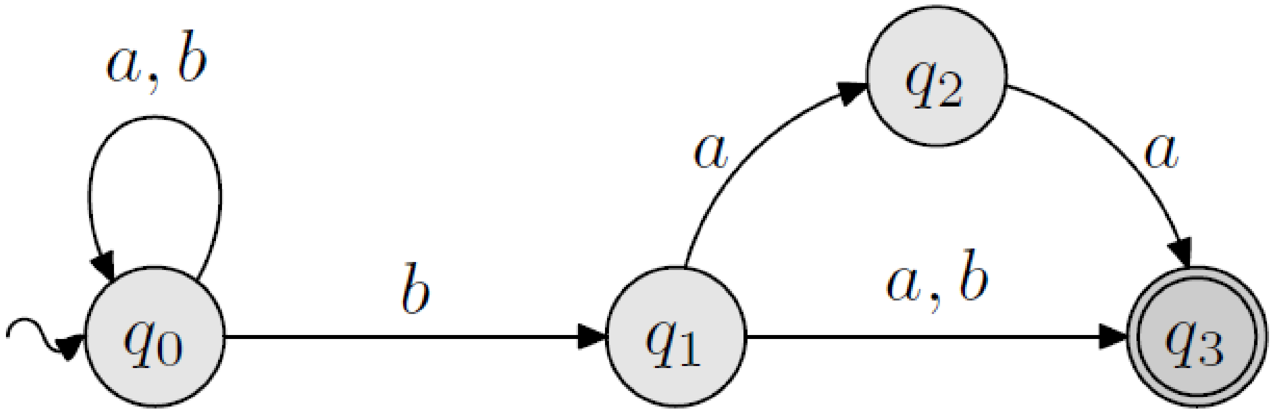
INPUT

A	B	A	B

STATO CORRENTE

q_0

Esempio di Computazione di un ASFND



INPUT

A	B	A	B

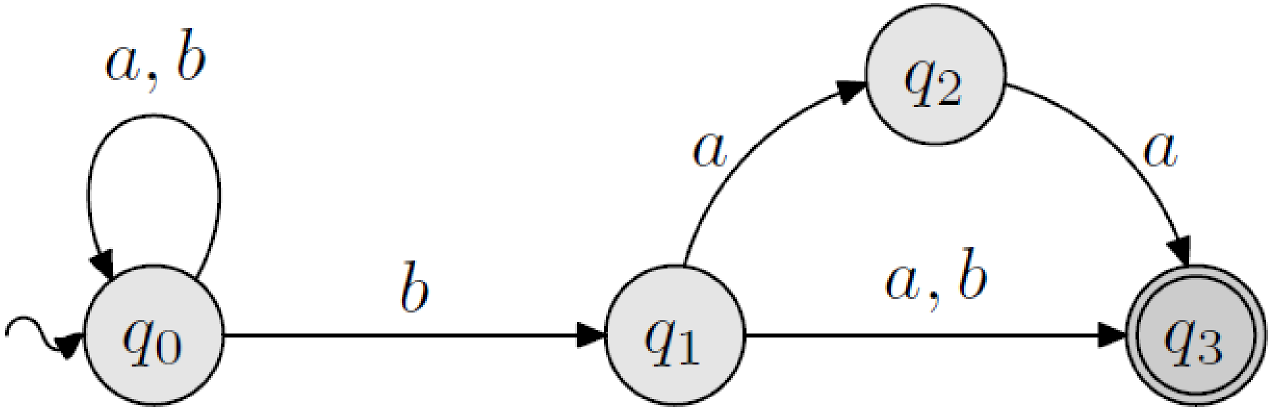
STATO CORRENTE

q_0

<u>**Due scelte possibili per il prossimo stato!**</u>

<u>**Scegliamo q_0**</u>

Esempio di Computazione di un ASFND



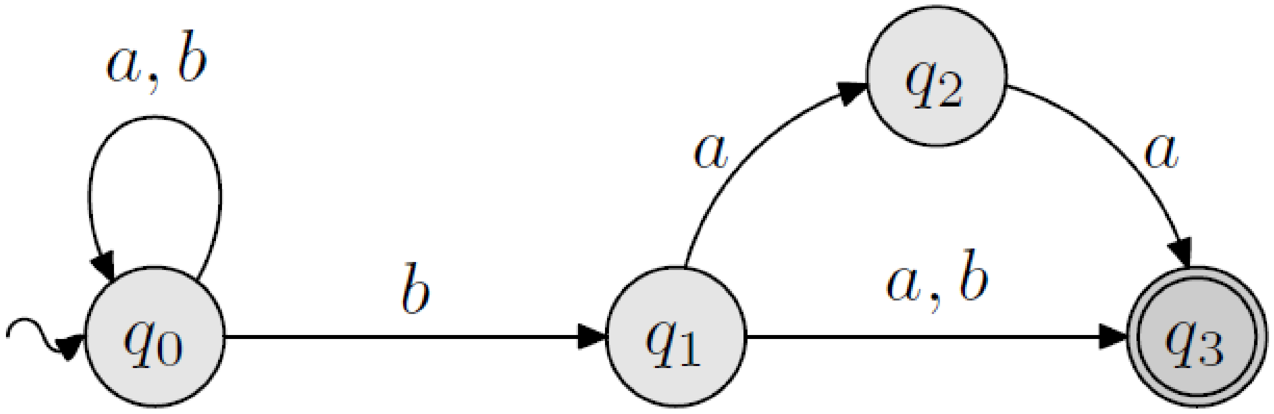
INPUT

A	B	A	B

STATO CORRENTE

q_0

Esempio di Computazione di un ASFND



INPUT

A	B	A	B

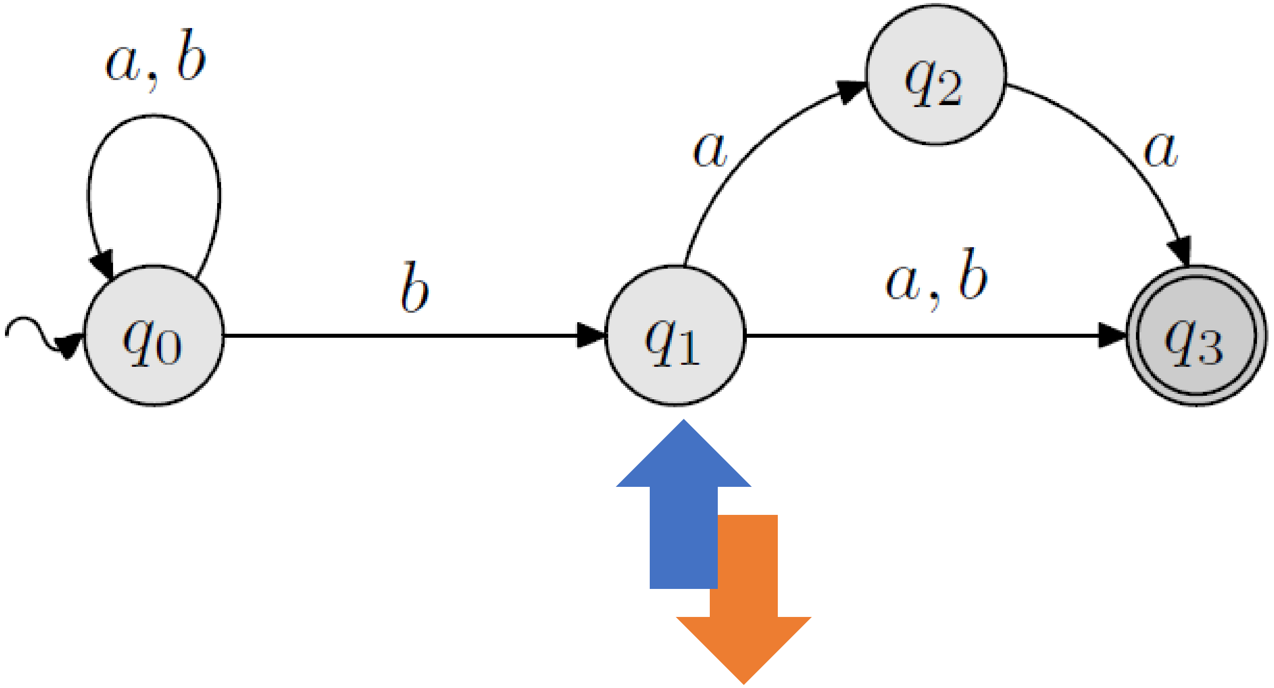
STATO CORRENTE

q_0

<u>**Due scelte possibili per il prossimo stato!**</u>

<u>**Scegliamo q_1**</u>

Esempio di Computazione di un ASFND



INPUT

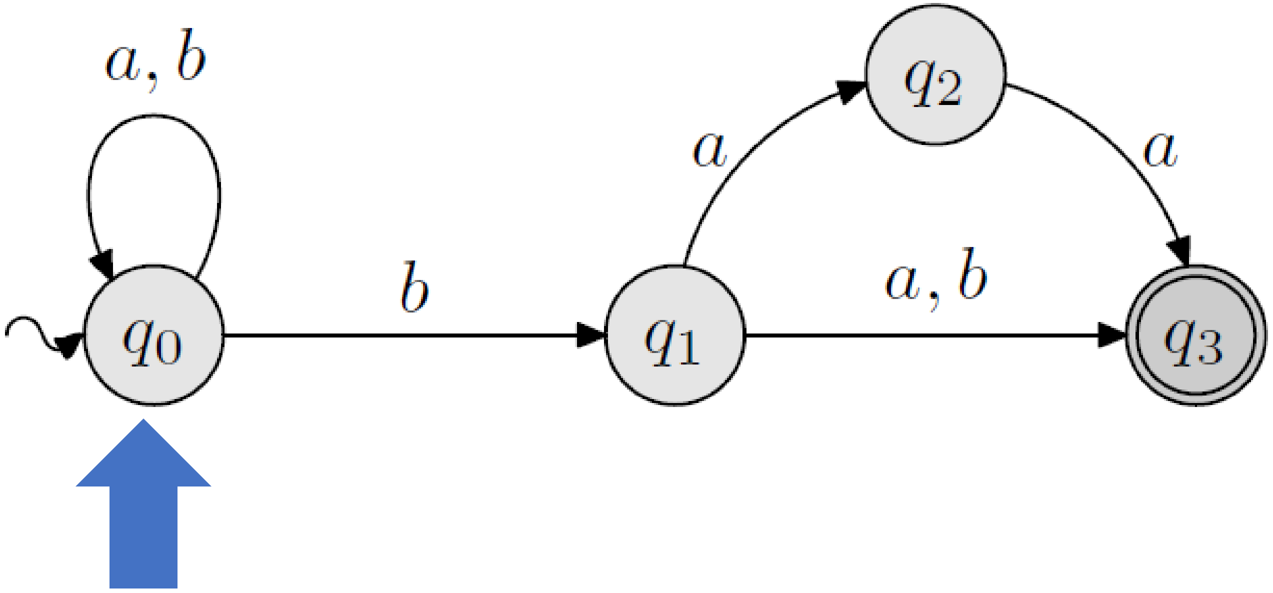
A	B	A	B

STATO CORRENTE

q_1

Computazione Terminata in uno Stato Non Finale!

Esempio di Computazione di un ASFND



INPUT

A	B	A	B

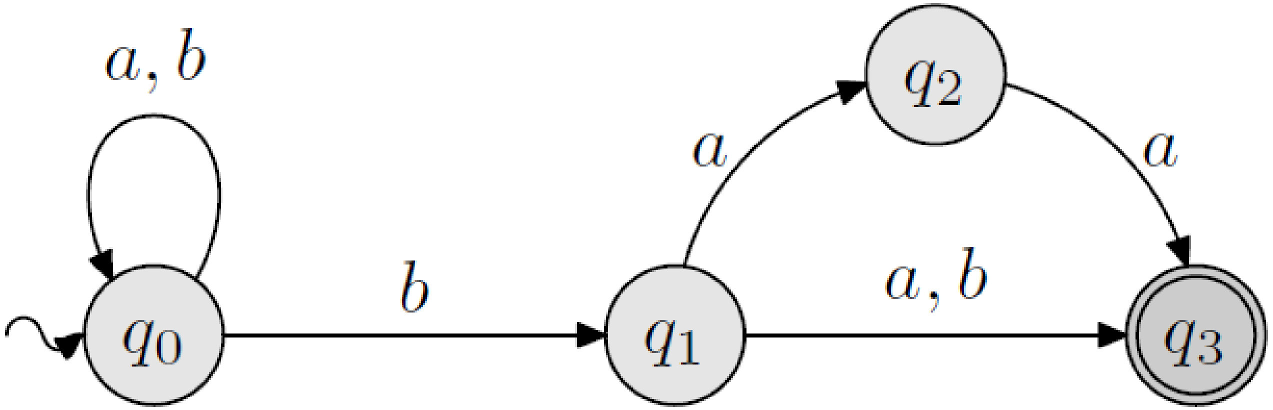
STATO CORRENTE

q_0

<u>**Due scelte possibili per il prossimo stato!**</u>

<u>**Scegliamo q_1**</u>

Esempio di Computazione di un ASFND



INPUT

A	B	A	B

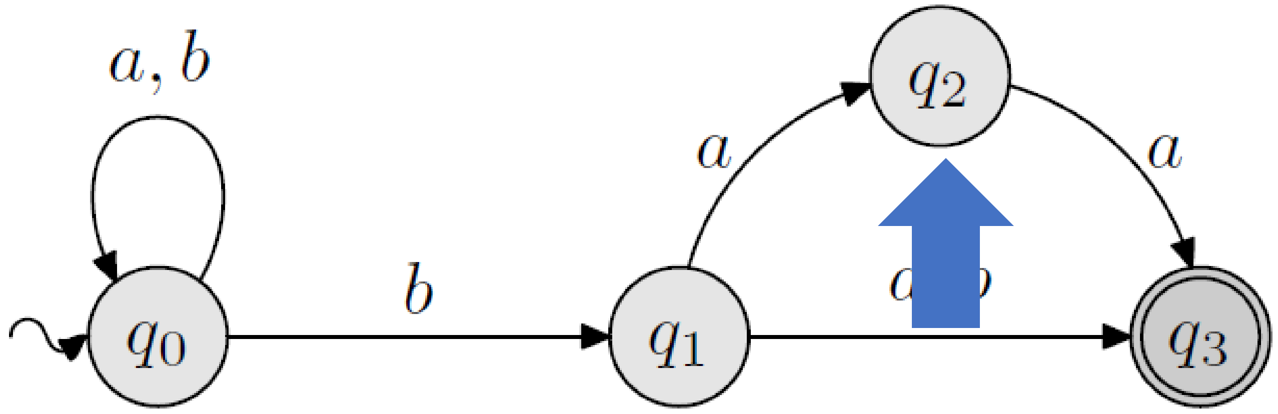
STATO CORRENTE

q_1

<u>**Due scelte possibili per il prossimo stato!**</u>

<u>**Scegliamo q_2**</u>

Esempio di Computazione di un ASFND



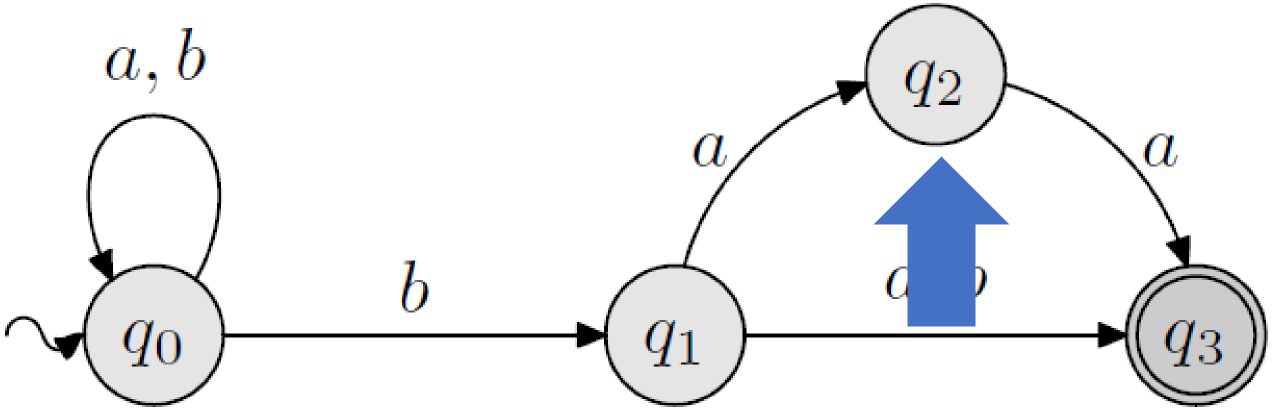
INPUT

A	**B**	**A**	**B**

STATO CORRENTE

q_2

Esempio di Computazione di un ASFND



INPUT

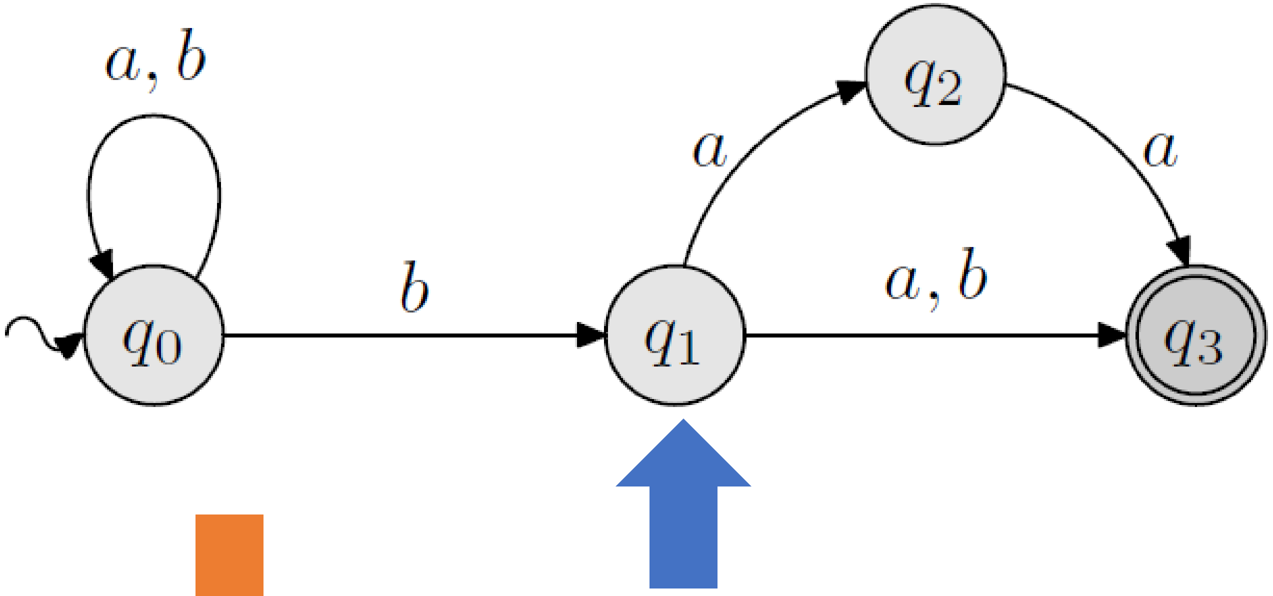
A	B	A	B

STATO CORRENTE

q_2

<u>**Computazione Terminata in uno Stato Non Finale!**</u>

Esempio di Computazione di un ASFND



INPUT

A	B	A	B

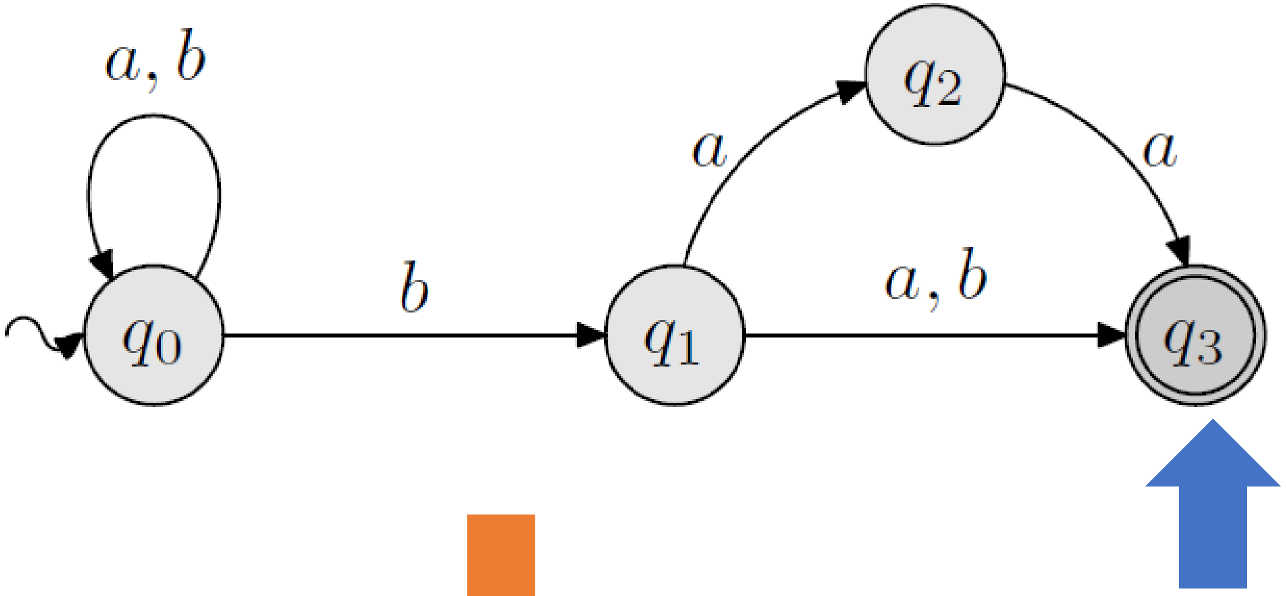
STATO CORRENTE

q_1

<u>**Due scelte possibili per il prossimo stato!**</u>

<u>**Scegliamo q_3**</u>

Esempio di Computazione di un ASFND



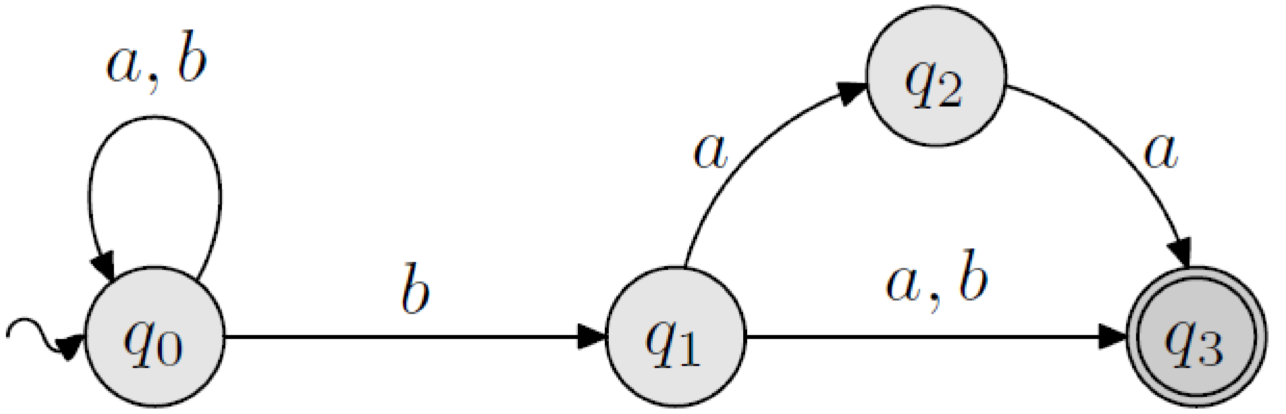
INPUT

A	B	A	B

STATO CORRENTE

q_3

Esempio di Computazione di un ASFND



INPUT

A	B	A	B

STATO CORRENTE

q_3

Computazione Terminata in uno Stato Finale!

Esecuzioni di un ASFND

- Sia dato un ASFND $A =< \Sigma, Q, \delta, I, F >$ e una stringa $S = "c_1 c_2 \dots c_n" \in \Sigma^*$ con $|S| = n$

- **Definizione**. Una **esecuzione di $\boldsymbol{A}$ su S** è una sequenza $(q_1, \dots, q_{n+1}) \in Q^{n+1}$ di $n + 1$ elementi di Q tale che
  - $\boldsymbol{q_1 = I}$ (**intuitivamente**, il primo stato è quello iniziale)
  - $\boldsymbol{q_{i+1} \in \delta(q_i, c_i)}$ **per** $\boldsymbol{i = 1, \dots, n}$ (**intuitivamente**, ogni stato appartiene all'insieme definito dal precedente applicando δ al simbolo corrente)

- **Definizione**. Lo stato finale di una esecuzione $(q_0, \dots, q_n)$ di A su S è q_n
  - **L'ultimo della sequenza**

- **Definizione**. Una esecuzione di $(q_0, \dots, q_n)$ di A su S è **accettante** se il suo stato finale $\boldsymbol{q_n}$ **è in F**

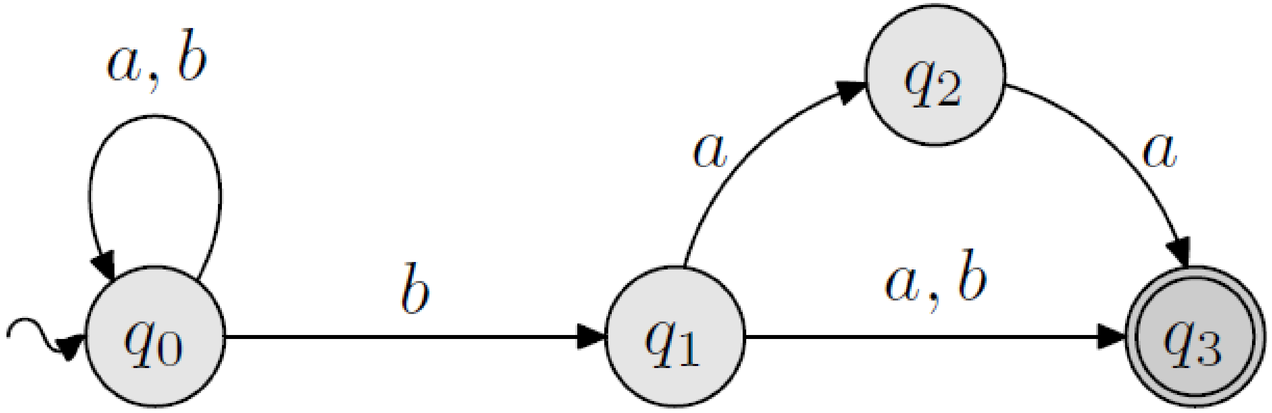
Esecuzioni di un ASFND

- **Proposizione**. Dato una ASFD A e una stringa S sull’alfabeto di A, potrebbe esistere **più di una esecuzione** di A su S.
- Dimostrazione. L’esempio precedente.

- Un ASFND può eseguire diverse configurazioni per lo stesso input a differenza di ASFD

- **Domanda**. Quando dovremmo accettare?

Esempio di Esecuzione



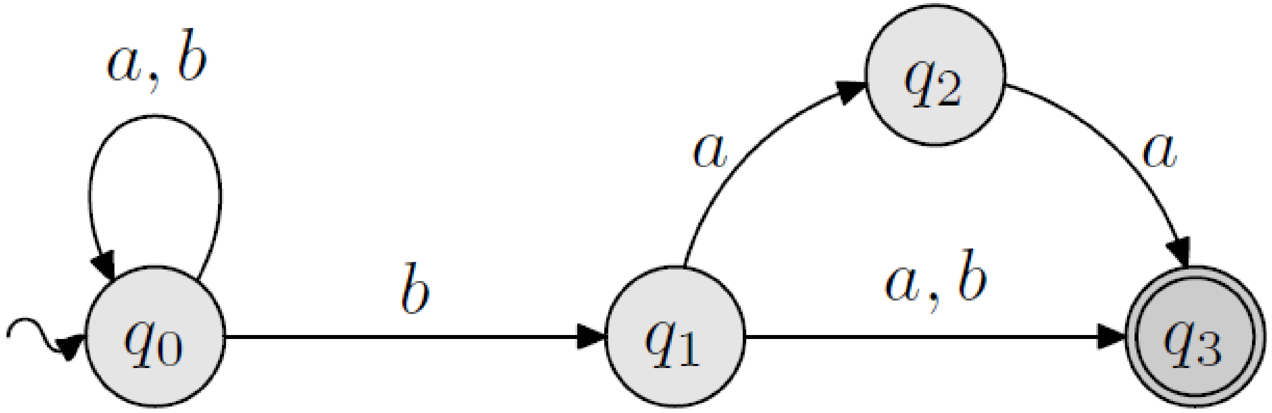
INPUT

A	**B**	**A**	**B**

ESECUZIONE ACCETTANTE

q_0	q_0	q_1	q_3	q_3

Esempio di Esecuzione



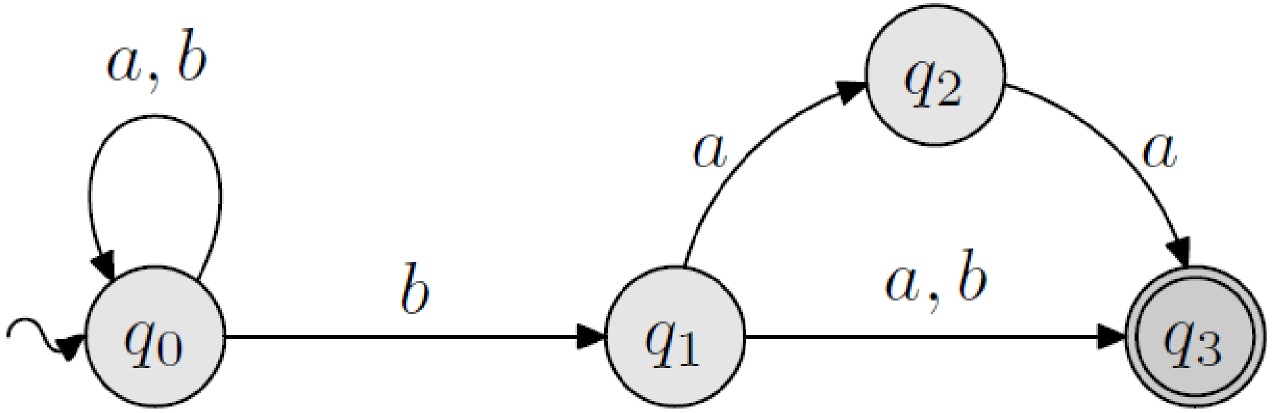
INPUT

A	B	A	B

ESECUZIONE NON ACCETTANTE

q_0	q_0	q_0	q_0	q_1

Esempio di Esecuzione



INPUT

A	**B**	**A**	**B**

Non è una esecuzione!

q_1	q_3	q_3	q_3	q_3

Linguaggio Riconosciuto da un ASFD

- **Definizione**. Dato un **ASFND** $A =< \Sigma, Q, \delta, q_0, F >$ e una stringa $x \in \Sigma^*$
  - x è **accettata** da A se **esiste almeno una** esecuzione accettante di A su x
  - Altrimenti, x è **rifiutata**

- **Definizione**. Sia $A =< \Sigma, Q, \delta, q_0, F >$ un **ASFND**. **Il linguaggio riconosciuto da $\boldsymbol{A}$** è il linguaggio $L(A)$ sull'alfabeto Σ tale che

$$L(A) = \{x \in \Sigma^* \mid x \text{ è } accettata\ da\ A\}$$

- **Definizione**. Un linguaggio $\boldsymbol{\mathcal{L}}$ **è detto regolare** se esiste un **ASFD $\boldsymbol{A}$ tale che $\boldsymbol{L(A) = \mathcal{L}}$**

- **Domanda**. È vero che ogni linguaggio riconosciuto da un **ASFND** è regolare?
  - Ovvero, possiamo sempre definire un **ASFD** equivalente a un **ASFND**?

Equivalenza tra ASFND e ASFD

ASFND e ASFD Sono Modelli Computazionali Equivalenti

- Dimostreremo ora che gli ASFND e gli ASFD hanno lo **stesso potere espressivo**, ovvero riconoscono esattamente gli stessi linguaggi

- In termini più precisi dimostreremo le seguenti due affermazioni

Teorema 1. Le seguenti due affermazioni sono vere.

1. **Per ogni ASFD $\boldsymbol{A}$ esiste un ASFND $\boldsymbol{A_N}$** tale che $L(A) = L(A_N)$
   – Gli ASFND possono esprimere ogni ASFD
2. **Per ogni ASFND $\boldsymbol{A_N}$ esiste un ASFD $\boldsymbol{A}$** tale che $L(A_N) = L(A)$
   – Gli ASFD possono esprimere ogni ASFND

Gli ASFND Possono Esprimere gli ASFD (Punto 1)

- Questa affermazione è semplice da dimostrare.

- Dato un ASFD $A = < \Sigma, Q, \delta, q_0, F >$, sia $A_N = < \Sigma, Q, \delta_N, q_0, F >$ l'ASFND tale che $\delta_N(q, a) = \{\delta(q, a)\}$ per ogni coppia $(q, a) \in Q \times \Sigma$

- E' ovvio che $L(A) = L(A_N)$ inquanto, per ogni stringa $S \in \Sigma^*$ esiste una sola computazione di A_N su S che coincide proprio con $A(S)$

Gli ASFD Possono Esprimere gli ASFND (Punto 2)

- Questa affermazione è tutt'altro che semplice da dimostrare ☺

- **Intuizione 1**. Dato un ASFND $A_N = <\Sigma, Q, \delta_N, q_0, F>$, costruiamo un ASFD il cui insieme degli stati è $P(Q)$ (insieme delle parti di Q)
  - Utilizzeremo questo (enorme) insieme di stati per simulare deterministicamente tutte le possibili computazioni dell'ASFND A_N

- **Intuizione 2.** Uno stato di A è finale se contiene almeno uno stato finale di A_N
  - Una esecuzione che termina in tale stato rappresenta un insieme di esecuzioni tra cui ce ne è una che termina proprio nello stato finale

- **Intuizione 2**. La funzione di transizione $\delta_A(q_A, x) = q_A'$ se l'insieme ottenuto applicando δ a (q, x) per ogni stato $q \in q_A$ definisce proprio q_A'
  - Ogni passaggio di stato rappresenta un insieme di possibili passaggi di stato nelle computazioni di A_N

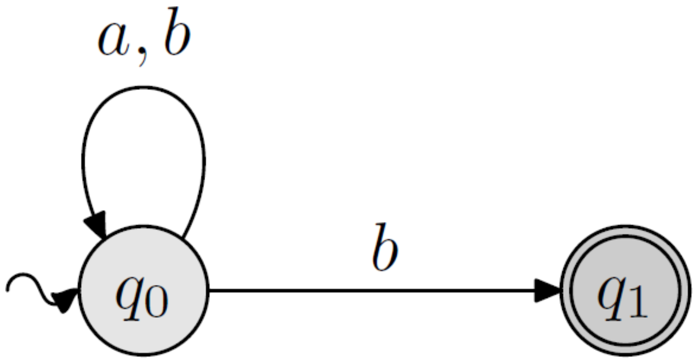
Punto 2 – Costruzione

- Dato un ASFND $A_N = < \Sigma, Q, \delta_N, q_0, F >$, sia $A' = < \Sigma, Q', \delta', q_0', F' >$ l'ASFD tale che:
  - $Q' = P(Q)$
  - $q_0' = \{q_0\}$,
  - $F' = \{X \in P(Q) \mid X \cap F \neq \emptyset\}$
  - $\delta'(\{\}, a) = \{\}$, per ogni $a \in \Sigma$
  - $\delta'\left(\{q_i^1, \dots, q_i^k\}, a\right) = \delta_N\left(q_i^1, a\right) \cup \cdots \cup \delta_N\left(q_i^k, a\right)$, per ogni $\{q_i^1, \dots, q_i^k\} \in (P(Q) \setminus \emptyset)$ e $a \in \Sigma$

Punto 2 – Costruzione – Esercizio

- Definire A' per l'ASFND seguente.
  - **Nota**: se un nodo ha un arco mancante assumiamo un cappio
  - $\Sigma = \{a, b\}$
  - $Q = \{q_0, q_1\}$
  - $I = q_0$
  - $F = \{q_1\}$
  - δ *definito dalla tabella sottostante*

Stato q	Simbolo s	$\delta'(q, s)$
q_0	a	$\{q_0\}$
q_0	b	$\{q_0, q_1\}$
q_1	a	$\{q_1\}$
q_1	b	$\{q_1\}$

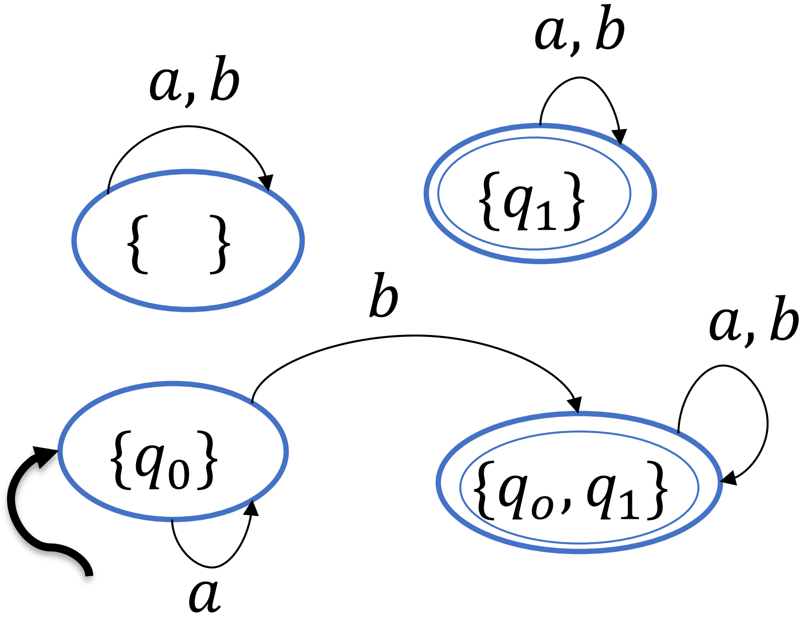


Punto 2 – Costruzione – Esercizio – Soluzione

- $A' = < \Sigma, Q', \delta', q'_0, F' >$ **definito come segue**
  - $\Sigma = \{a, b\}$
  - $Q' = \{\{\ \ \}, \{q_0\}, \{q_1\}, \{q_0, q_1\}\}$
  - $q'_0 = \{\{q_0\}\}$
  - $F' = \{\{q_1\}, \{q_0, q_1\}\}$

Stato q	Simbolo s	$\delta'(q, s)$
$\{\ \ \}$	a	$\{\ \ \}$
$\{\ \ \}$	b	$\{\ \ \}$
$\{q_0\}$	a	$\{q_0\}$
$\{q_0\}$	b	$\{q_0, q_1\}$
$\{q_1\}$	a	$\{q_1\}$
$\{q_1\}$	b	$\{q_1\}$
$\{q_0, q_1\}$	a	$\{q_0, q_1\}$
$\{q_0, q_1\}$	b	$\{q_0, q_1\}$

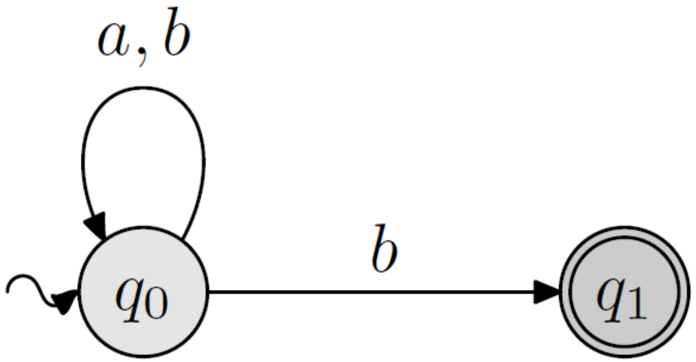
Punto 2 – Costruzione – Esercizio 1 – Soluzione



Stato q	Simbolo s	$\delta'(q, s)$
$\{\ \}$	a	$\{\ \}$
$\{\ \}$	b	$\{\ \}$
$\{q_0\}$	a	$\{q_0\}$
$\{q_0\}$	b	$\{q_0, q_1\}$
$\{q_1\}$	a	$\{q_1\}$
$\{q_1\}$	b	$\{q_1\}$
$\{q_0, q_1\}$	a	$\{q_0, q_1\}$
$\{q_0, q_1\}$	b	$\{q_0, q_1\}$

Punto 2 – Costruzione – Esercizio 2

- $A' = < \Sigma, Q', \delta', q_0', F' >$ =
  - $\Sigma = \{a, b\}$
  - $Q' = \{\{\ \ \}, \{q_0\}, \{q_1\}, \{q_0, q_1\}\}$
  - $q_0' = \{\{q_0\}\}$
  - $F' = \{\{q_1\}, \{q_0, q_1\}\}$

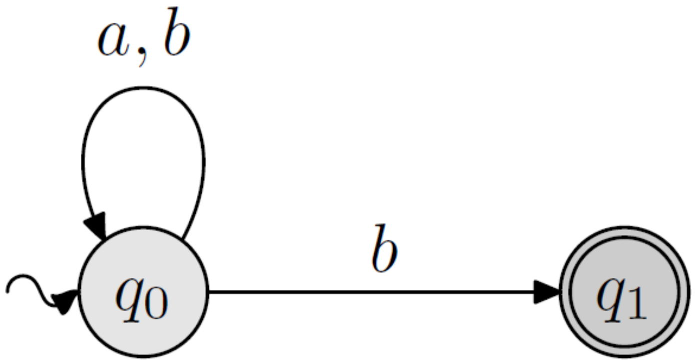


Stato q	Simbolo s	$\delta'(q, s)$
$\{\ \ \}$	a	$\{\ \ \}$
$\{\ \ \}$	b	$\{\ \ \}$
$\{q_0\}$	a	$\{q_0\}$
$\{q_0\}$	b	$\{q_0, q_1\}$
$\{q_1\}$	a	$\{q_1\}$
$\{q_1\}$	b	$\{q_1\}$
$\{q_0, q_1\}$	a	$\{q_0, q_1\}$
$\{q_0, q_1\}$	b	$\{q_0, q_1\}$

- *Qual è il linguaggio accettato dai due automi?*

Punto 2 – Costruzione – Esercizio 2 – Soluzione

- $A' = < \Sigma, Q', \delta', q_0', F' >$ =
  - $\Sigma = \{a, b\}$
  - $Q' = \{\{\ \ \}, \{q_0\}, \{q_1\}, \{q_0, q_1\}\}$
  - $q_0' = \{\{q_0\}\}$
  - $F' = \{\{q_1\}, \{q_0, q_1\}\}$



Stato q	Simbolo s	$\delta'(q, s)$
$\{\ \ \}$	a	$\{\ \ \}$
$\{\ \ \}$	b	$\{\ \ \}$
$\{q_0\}$	a	$\{q_0\}$
$\{q_0\}$	b	$\{q_0, q_1\}$
$\{q_1\}$	a	$\{q_1\}$
$\{q_1\}$	b	$\{q_1\}$
$\{q_0, q_1\}$	a	$\{q_0, q_1\}$
$\{q_0, q_1\}$	b	$\{q_0, q_1\}$

- Qual è il linguaggio accettato dai due automi?
  - Stringhe che contengono almeno una b

Punto 2 – Prova

- Dimostriamo che $s \in L(A_N)$ se e e solo se $s \in L(A')$.

- Per farlo, proveremo prima una proprietà più forte. Specificatamente, dimostreremo il seguente argomento.

- **Lemma 1**. Lo stato finale della computazione $A'(s)$ di A' su s **coincide con** l'insieme degli stati finali di tutte le computazione di A_N su s.

- La prova è **per induzione** sulla lunghezza della stringa di input x

- **Passo base** ($|x| = 0$). Sia $s = \epsilon$. A_N ha un'unica computazione possibile su ϵ, cioè (q_0); e $A'(\epsilon) = (\{q_0\})$. La proposizione segue banalmente.

Punto 2 – Prova Lemma 1 (1/2)

- **Passo induttivo** ($|x| > n$). Assumiamo **l'ipotesi induttiva**, e cioè che la proposizione sia vera per ogni stringa di input con $|x| \leq n$. In altre parole, assumiamo la seguente

- **Ipotesi Induttiva**. Per ogni stringa di input s di dimensione $|s| \leq n$ vale la seguente proprietà (**Lemma 1**): lo stato finale della computazione $A'(s)$ di A' su s **coincide con** l'insieme degli stati finali di tutte le computazione di A_N su s.

- **Dimostriamo** che lo stato finale della computazione $A'(s)$ di A' su s **coincide con** l'insieme degli stati finali di tutte le computazione di A_N su s per ogni stringa x tale che $|x| = n + 1$

Punto 2 – Prova Lemma 1 (1/2)

- Consideriamo ora il caso $|x| = n + 1$ e assumiamo $x = sa$ con $|s| = n$ e $a \in \Sigma$.

- Una **esecuzione di $\boldsymbol{A'}$ su $\boldsymbol{x}$** è una sequenza $A'(s) \circ (\delta'(\mathrm{Y}, \mathrm{a}))$
  - $A'(s) \circ (\delta'(\mathrm{Y}, \mathrm{a}))$ è la concatenazione di $A'(s)$ e la sequenza $(\delta'(\mathrm{Y}, \mathrm{a}))$ formata dall' unico elemento $(\delta'(\mathrm{Y}, \mathrm{a}))$

- **Lo stato finale della computazione di $\boldsymbol{A'}$ su $\boldsymbol{x}$ è $\boldsymbol{\delta'(\mathrm{Y}, \mathrm{a}) = \delta_N(q_i^1, a) \cup \cdots \cup \delta_N(q_i^k, a)}$**

- **Dimostriamo** che $\boldsymbol{\delta'(\mathrm{Y}, \mathrm{a})}$ è lo stato desiderato utilizzando l'Ipotesi Induttiva
  - È l'insieme degli stati finali di tutte le computazioni di A_N sulla stringa di input s

Punto 2 – Prova Lemma 1 (2/2)

- Esaminiamo $\boldsymbol{\delta}'(\mathbf{Y}, \mathbf{a}) = \boldsymbol{\delta_N}\left(\boldsymbol{q_i^1}, \boldsymbol{a}\right) \cup \cdots \cup \boldsymbol{\delta_N}\left(\boldsymbol{q_i^k}, \boldsymbol{a}\right)$

- Sia $\boldsymbol{T} \in \boldsymbol{\delta}'(\mathbf{Y}, \mathbf{a})$ **uno stato di** $\boldsymbol{A_N}$. Dimostriamo che esiste una computazione di A_N sul'input $x = s \circ a$ il cui stato finale è T
  - Se $\boldsymbol{T} \in \boldsymbol{\delta}'(\mathbf{Y}, \mathbf{a})$ allora esiste $T' \in Y$ tale che $\mathrm{T} \in \delta(T', a)$ (in A_N) (costruzione di A')
  - Per **Ipotesi Induttiva** lo **stato finale** $\boldsymbol{Y}$ di $A'(s)$ è l'insieme degli stati finali di tutte le computazioni di A_N su s
  - Possiamo concludere che T è lo stato finale di una computazione di A_N su x

- Sia C una computazione di A_N sul'input $x = s \circ a$ il cui stato finale è T. Dimostriamo che $\boldsymbol{T} \in \boldsymbol{\delta}'(\mathbf{Y}, \mathbf{a})$
  - Esiste una computazione C' di A_N su s con stato finale T' e $\mathrm{T} \in \delta(T', a)$ (definizione di ASFND)
  - Per **Ipotesi Induttiva** lo **stato finale** $\boldsymbol{Y}$ di $A'(s)$ è l'insieme degli stati finali di tutte le computazioni di A_N su s
  - Possiamo concludete che $\boldsymbol{T} \in \boldsymbol{\delta}'(\mathbf{Y}, \mathbf{a})$

Punto 2 – Prova

- **Lemma 2.** $s \in L(A_N)$ se e e solo se $s \in L(A')$.

- **Prova**. Proviamo le due proposizioni separatamente

- **Se** $\boldsymbol{s \in L(A_N)}$ allora esiste una esecuzione accettante di A_N su s. Sia X tale esecuzione e Y il suo stato finale. Allora $Y \in F$ e $Y \in A'(s)$ (**Lemma 1**) che, per costruzione, implica $A'(s) \in F'$ e quindi $s \in L(A')$.

- **Se** $\boldsymbol{s \in L(A')}$ allora $A'(s) \in F'$ che, per costruzione, implica che esiste $Y \in A'(s)$ tale che $Y \in F$. Possiamo concludere che esiste una computazione di A_N su s il cui stato finale è proprio Y (**Lemma 1**) che $s \in L(A_N)$.