Espressioni regolari

Ricordo: un linguaggio è regolare se è riconosciuto da un NFA (=DFA)

Definizione sintattica

L'espressioni regolari di base : $\emptyset, \quad \lambda, \quad \alpha$

Date le espressioi regolari r_1 e r_2

$$\left.\begin{array}{l} r_1 + r_2 \\ r_1 \cdot r_2 \\ r_1 * \\ (r_1) \end{array}\right\} \text{Sono espressioni regolari}$$

Una semantica: Linguaggi associati alle espressioni regolari

Per le espressioni regolari di base:

$$L(\emptyset)=\emptyset$$

$$L(\lambda)=\{\lambda\}$$

$$L(a)=\{a\}$$

passo

per le espressioni regolari r_1 e r_2

$$L(r_1 + r_2) = L(r_1) \cup L(r_2)$$

$$L(r_1 \cdot r_2) = L(r_1)\, L(r_2)$$

$$L(r_1 *) = (L(r_1))*$$

$$L((r_1)) = L(r_1)$$

Linguaggi associati alle espressioni regolari

$L(r)$: linguaggio associato all'espressione r

esempio

$$L((a+b\cdot c)^*)=\{\lambda,a,bc,aa,abc,bca,...\}$$

Espressioni regolari definiscono solo e soltanto i linguaggi regolari?

proprietà dei linguaggi regolari e automi

per linguaggi regolari L_1 e L_2
dimostreremo che:

Unione: $L_1 \cup L_2$

Concatenatione: $L_1 L_2$

Star: $L_1{}^*$

Reversal: $L_1{}^R$

Complemento: $\overline{L_1}$

Intersezione: $L_1 \cap L_2$

sono linguaggi regolari

diremo: linguaggi regolari sono **chiusi sotto**

Unione: $L_1 \cup L_2$

Concatenatione: $L_1 L_2$

Star: $L_1 *$

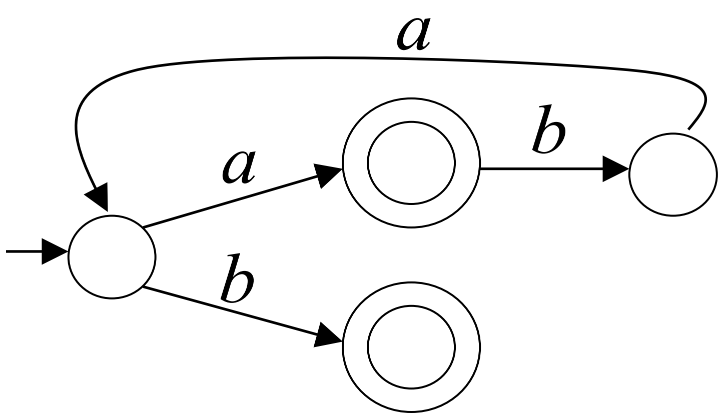
Reversal: $L_1{}^R$

Complemento: $\overline{L_1}$

Intersezione: $L_1 \cap L_2$

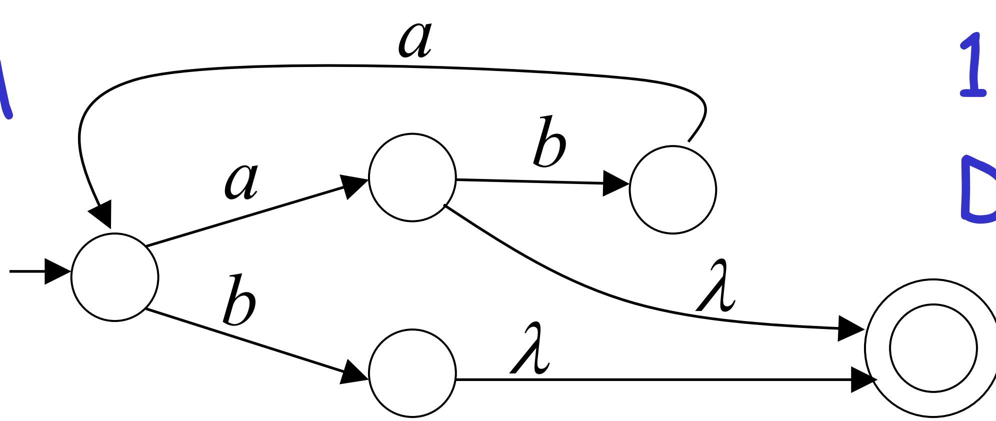
Useremo nfa con un solo stato finale

NFA



2 stati di accettazione

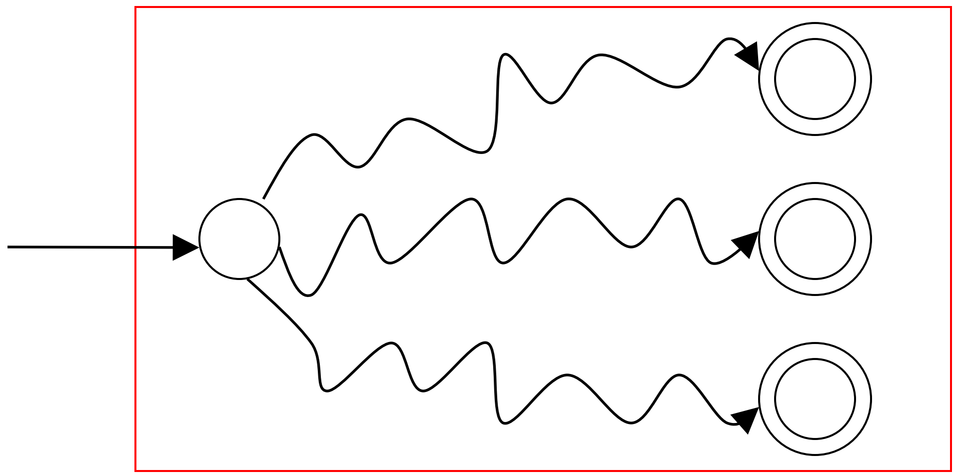
Equivalente NFA



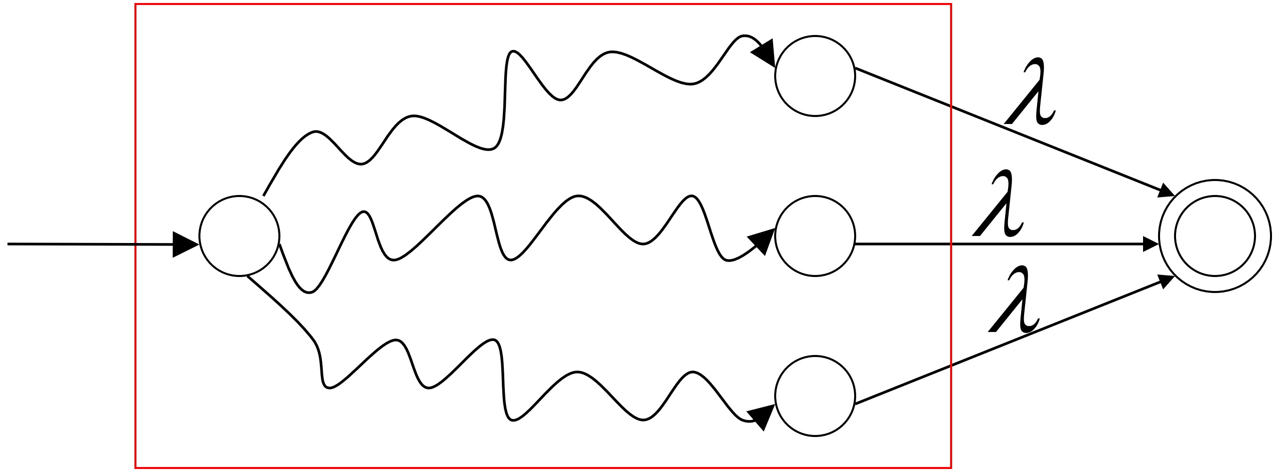
1 solo stato Di accettazione

In Generale

NFA



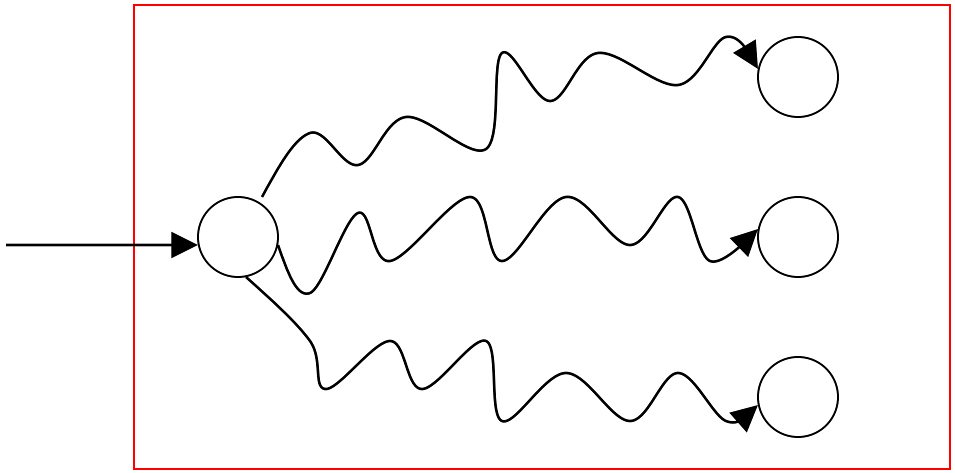
Equivalent NFA

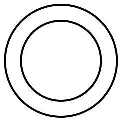


Un solo Stato di accettazione

Caso estremo

NFA senza stato di accettazione





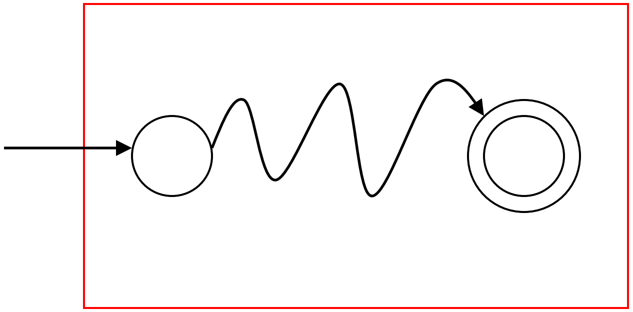
Addizioniamo
Uno stato di
Accettazione
Senza transizione

Prendiamo due linguaggi

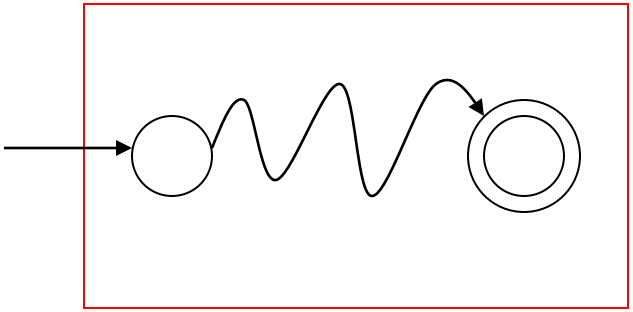
Linguaggio regolare L_1 linguaggio regolare L_2

$$L(M_1) = L_1 \qquad L(M_2) = L_2$$

NFA M_1



NFA M_2



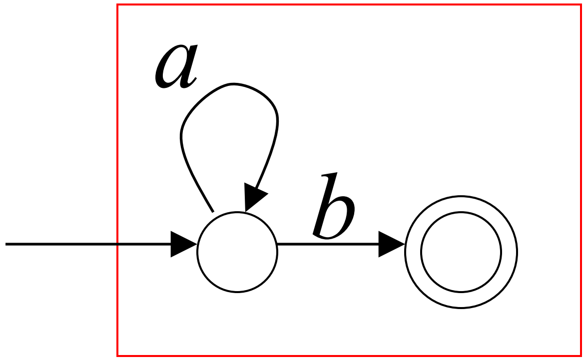
Un solo stato di accettazione Un solo stato di accettazione

Esempio

M_1

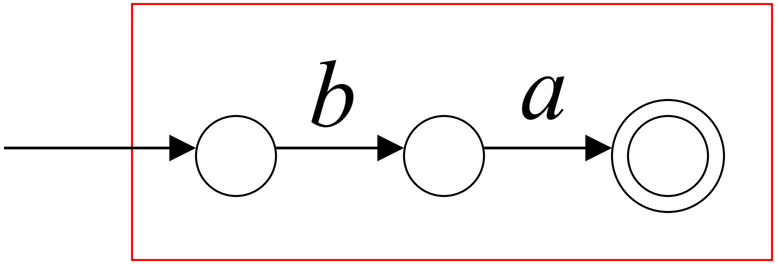
$n \geq 0$

$L_1 = \{a^n b\}$



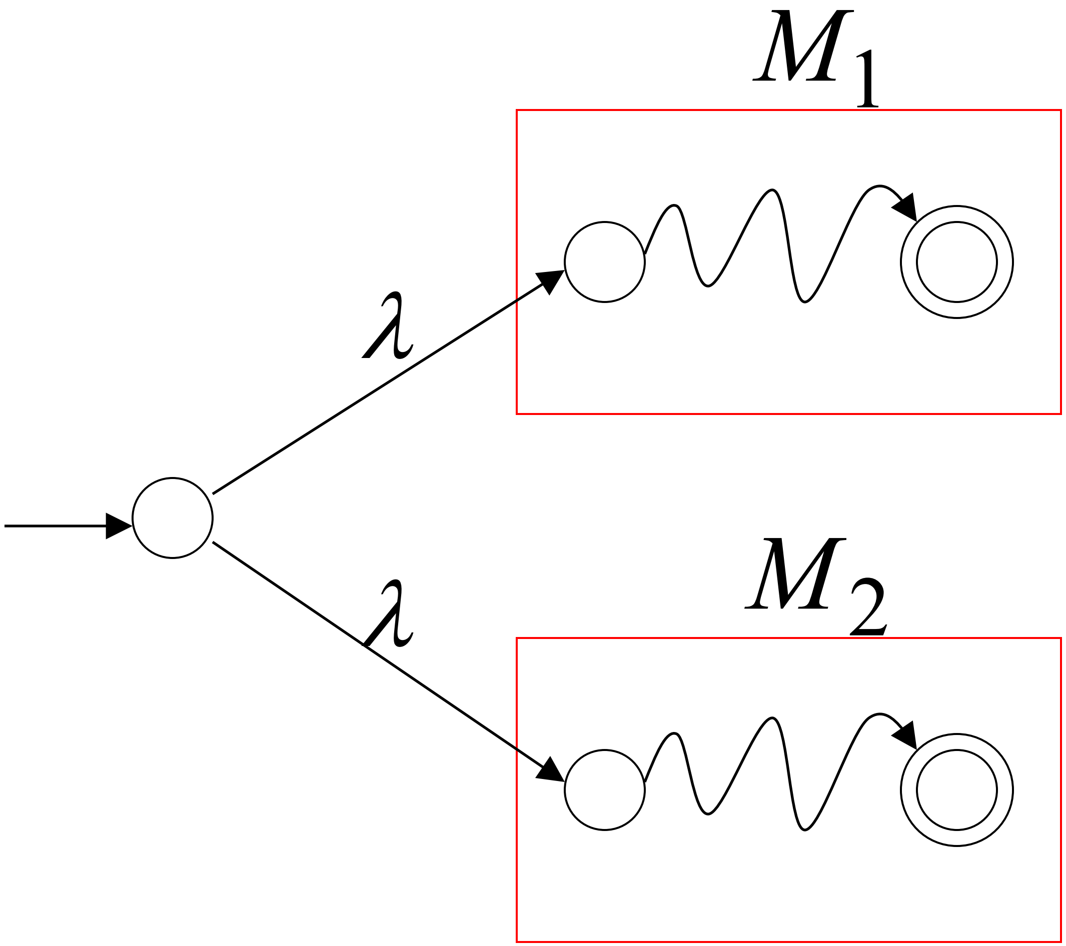
M_2

$L_2 = \{ba\}$



Unione

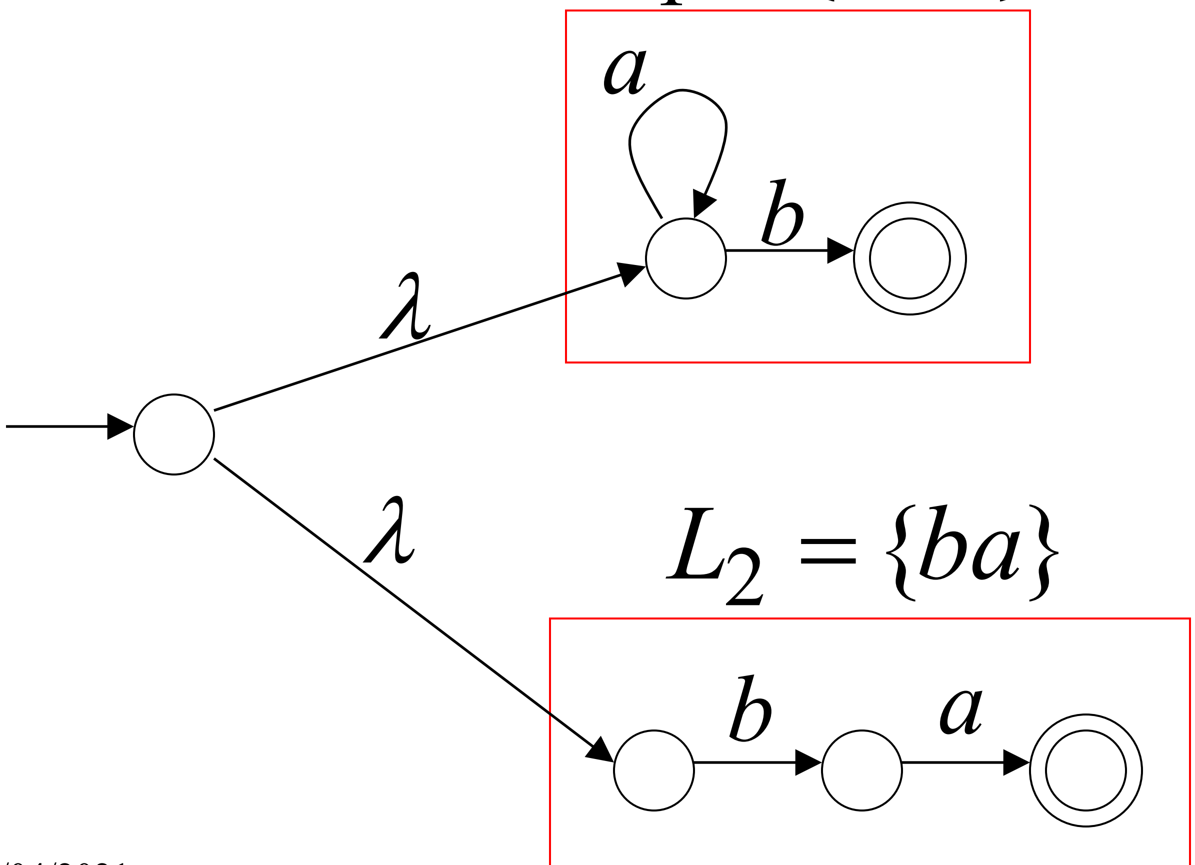
NFA per $L_1 \cup L_2$



Esempio

NFA per $L_1 \cup L_2 = \{a^n b\} \cup \{ba\}$

$$L_1 = \{a^n b\}$$



Evitiamo le transizioni con le lambda transizioni.
Mostriamo che a partire da due automi (N_1,N_2) , si può costruire l'automa unione dei due linguaggi definiti dagli automi precedenti.

Gli stati del nuovo automa sono l'unione degli stati precedenti, K_1 e K_2, più un nuovo stato iniziale q'_0.

Funzione transizione dell'automa unione, N, a partire dalle delta di N_1 e N_2.

$$\delta_N(q,a) = \delta_{N_1}(q,a), q \in K_1, a \in \Sigma_1$$

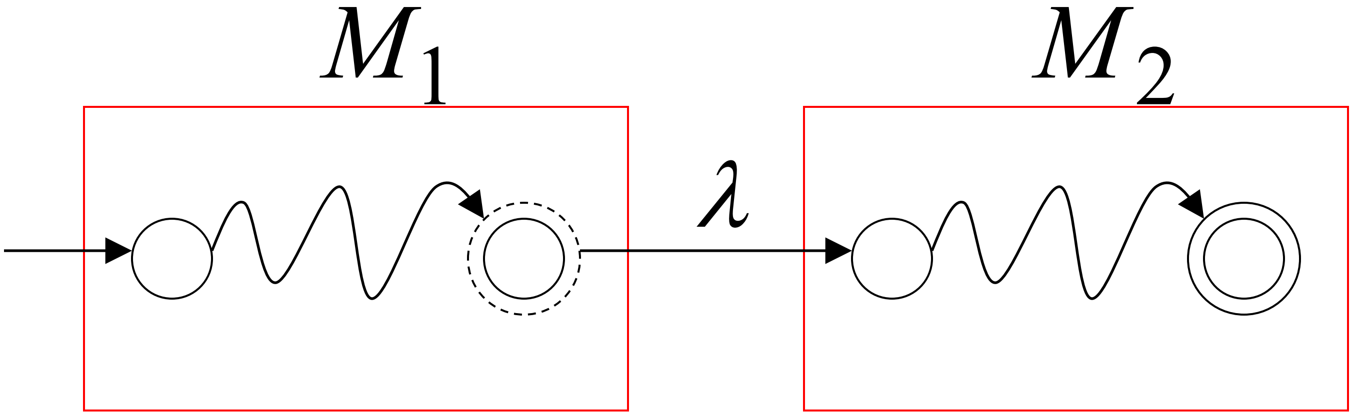
$$\delta_N(q,a) = \delta_{N_2}(q,a), q \in K_2, a \in \Sigma_2$$

$$\delta_N(q'_0,a) = \delta_{N_1}(q_{0_1},a) \bigcup \delta_{N_2}(q_{0_2},a), a \in \Sigma$$

Provare che la definizione precedente definisce l'unione di due automi.

Concatenazione

NFA per $L_1 L_2$

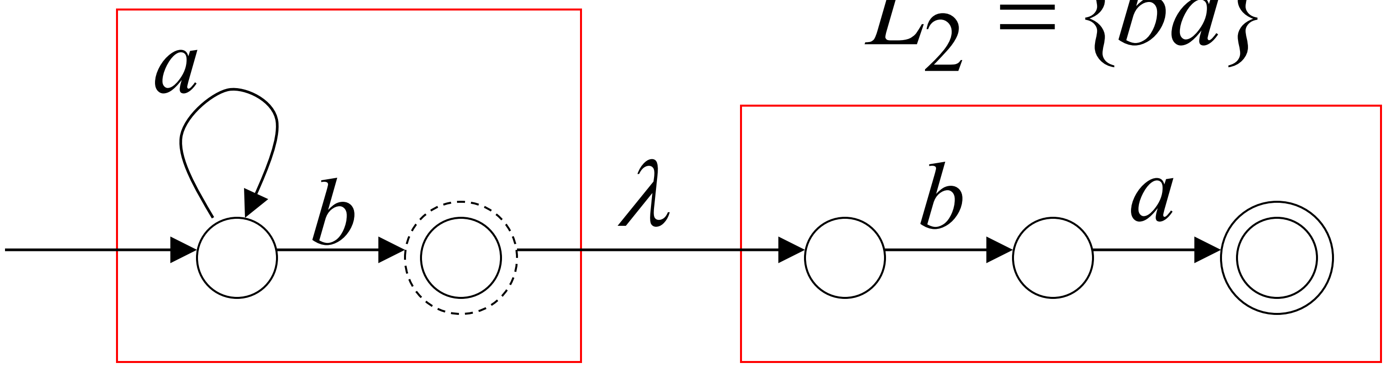


esempio

NFA per $L_1 L_2 = \{a^n b\}\{ba\} = \{a^n bba\}$

$L_1 = \{a^n b\}$

$L_2 = \{ba\}$

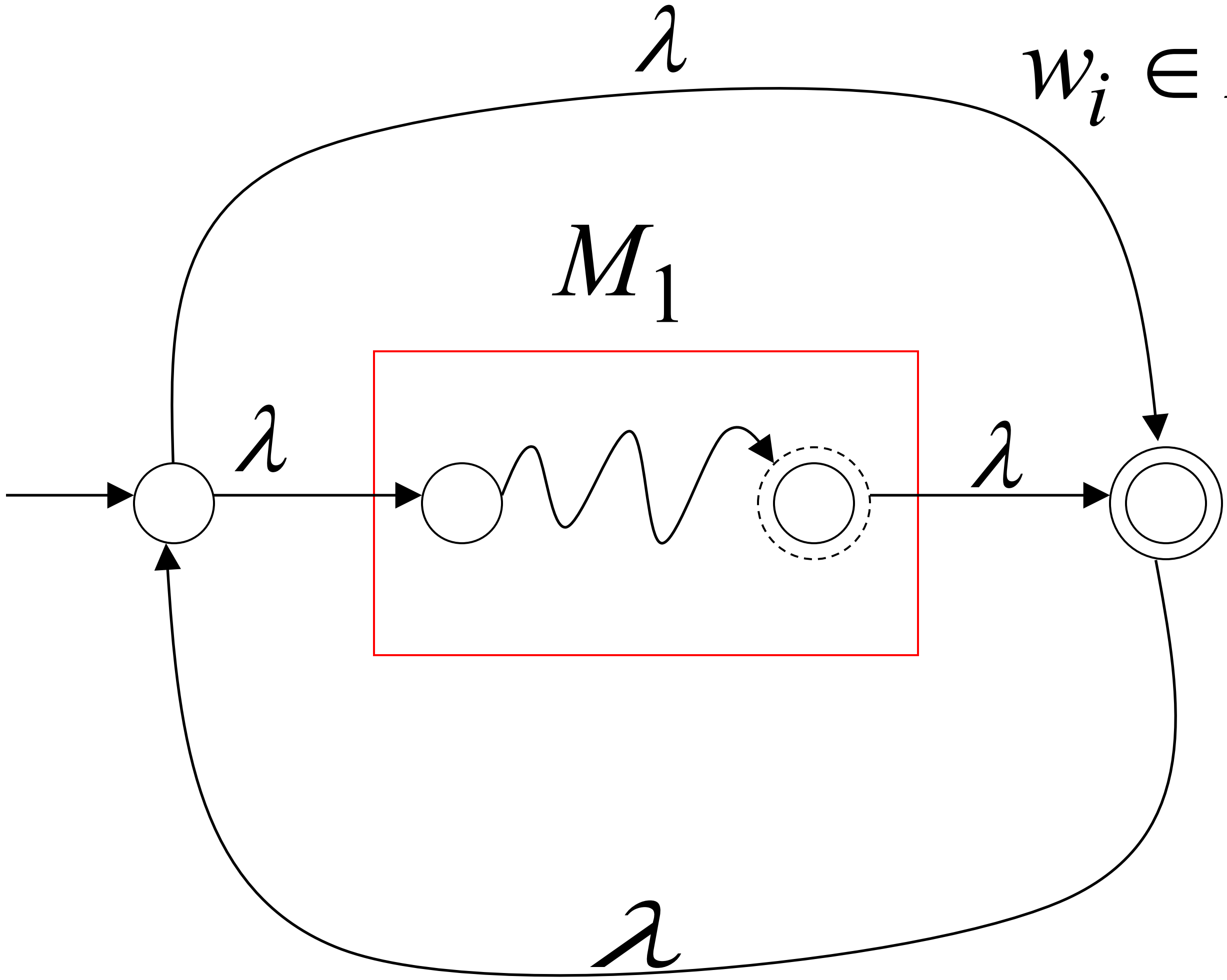


Star

NFA per $L_1{}^*$

$w = w_1 w_2 \cdots w_k$

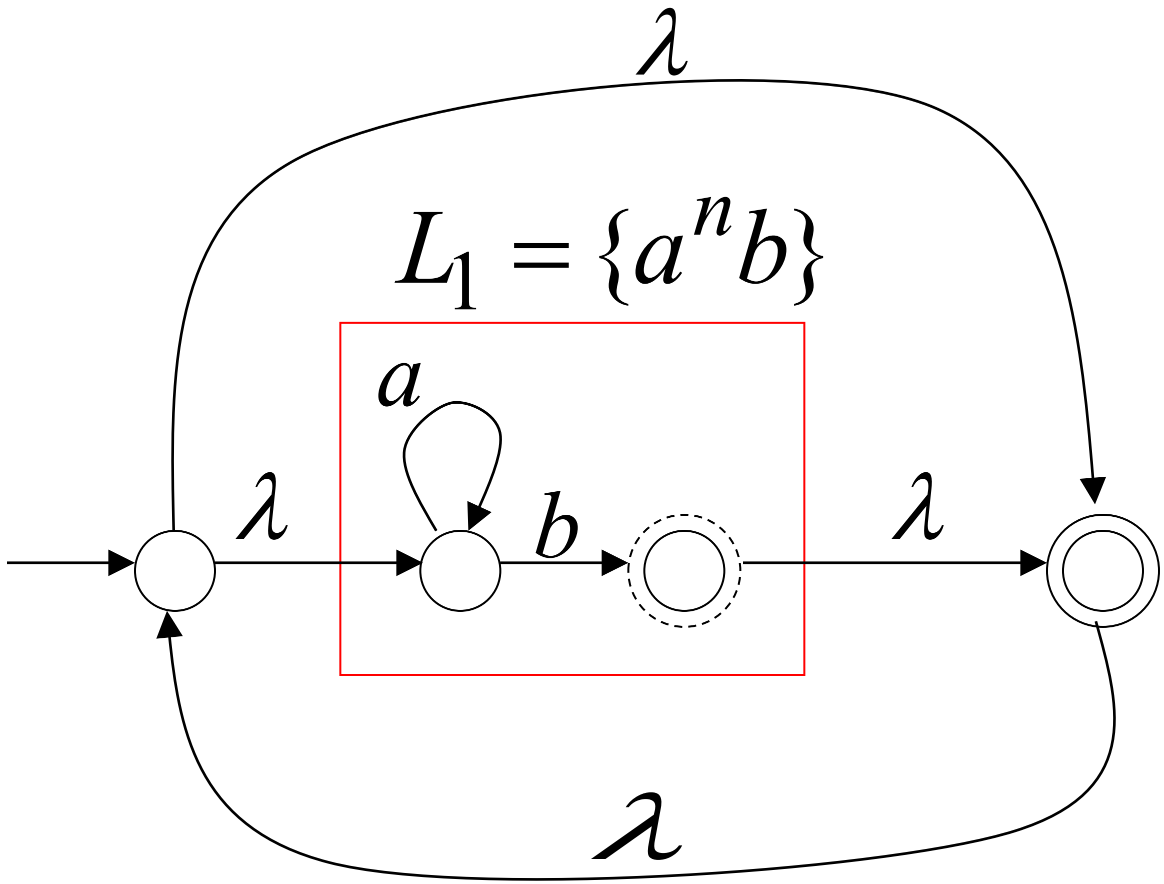
$w_i \in L_1$



$\lambda \in L_1{}^*$

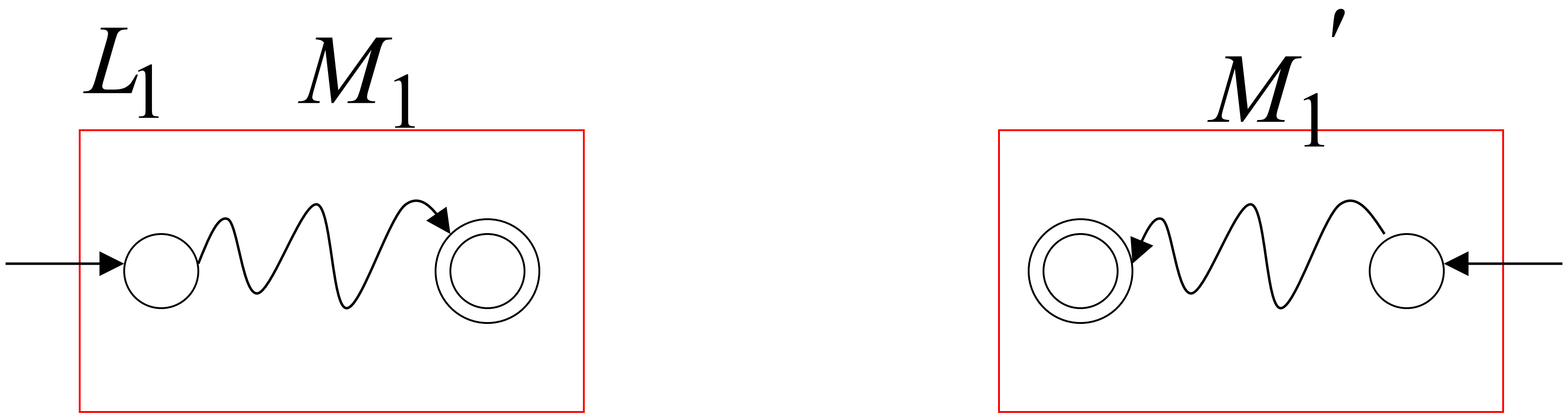
esempio

NFA per $L_1{}^* = \{a^n b\}^*$



Reverse

NFA per L_1^R

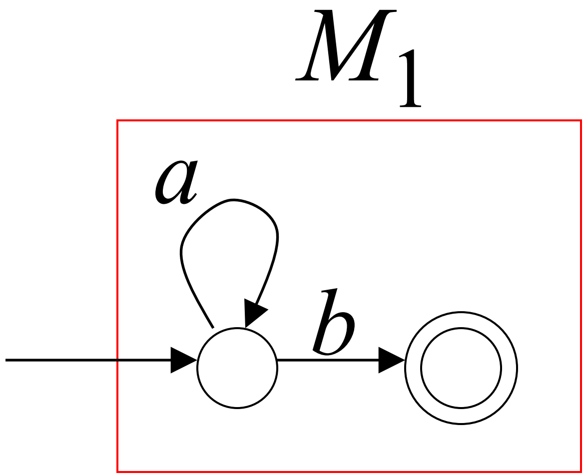


1. Reverse tutte le transizioni

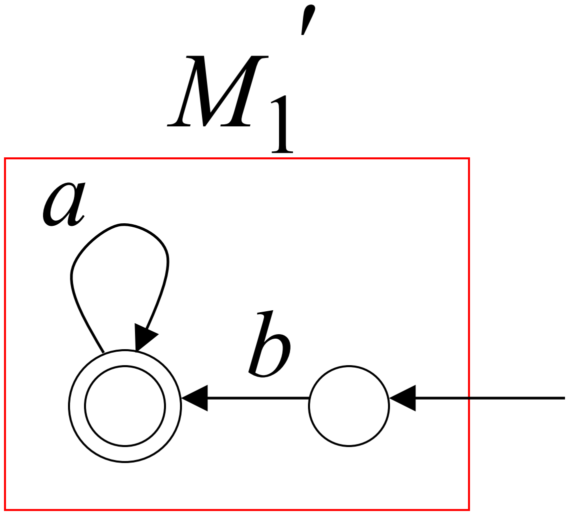
2. Stato iniziale quello finale,
quello finale stato iniziale

esempio

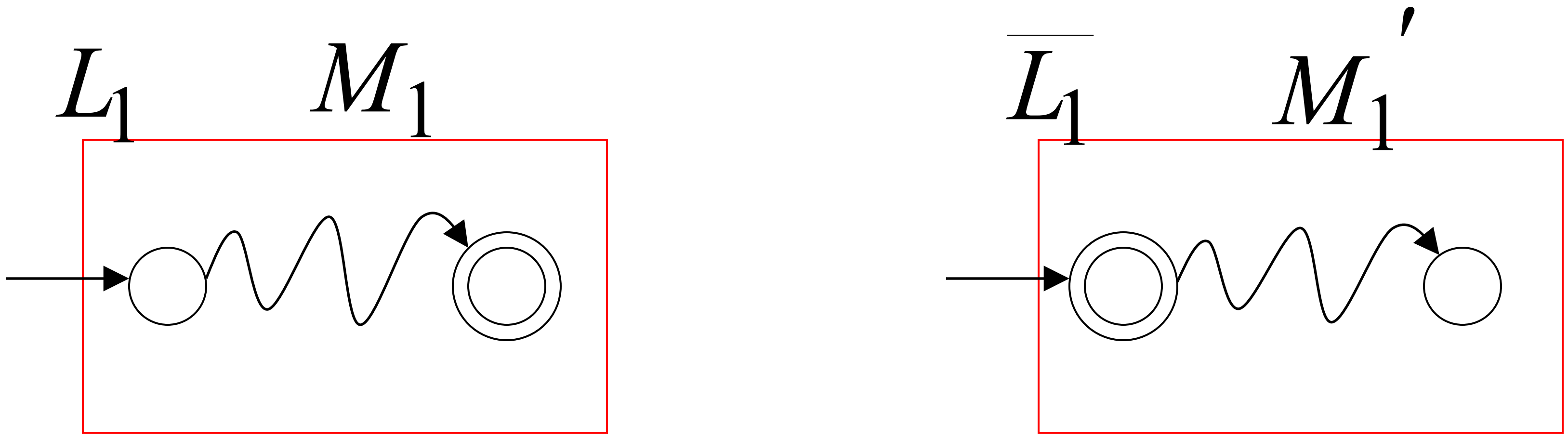
$$L_1 = \{a^n b\}$$



$$L_1^R = \{ba^n\}$$



Complemento

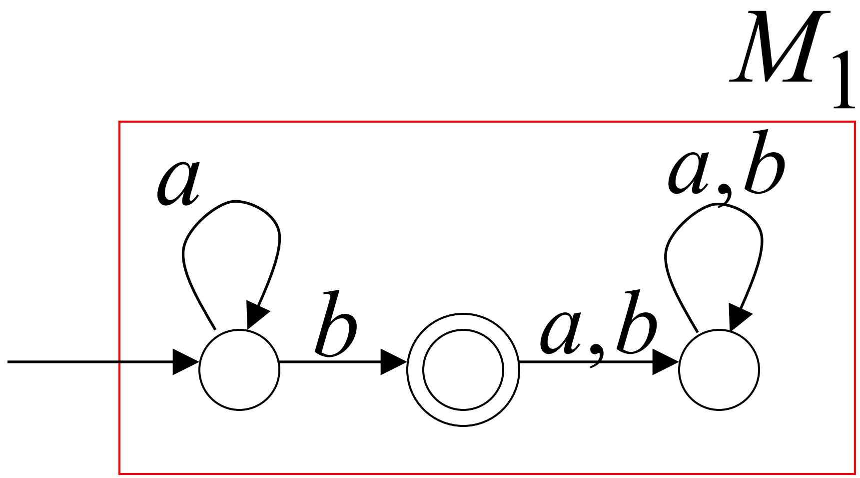


prendiamo il DFA che accetta L_1

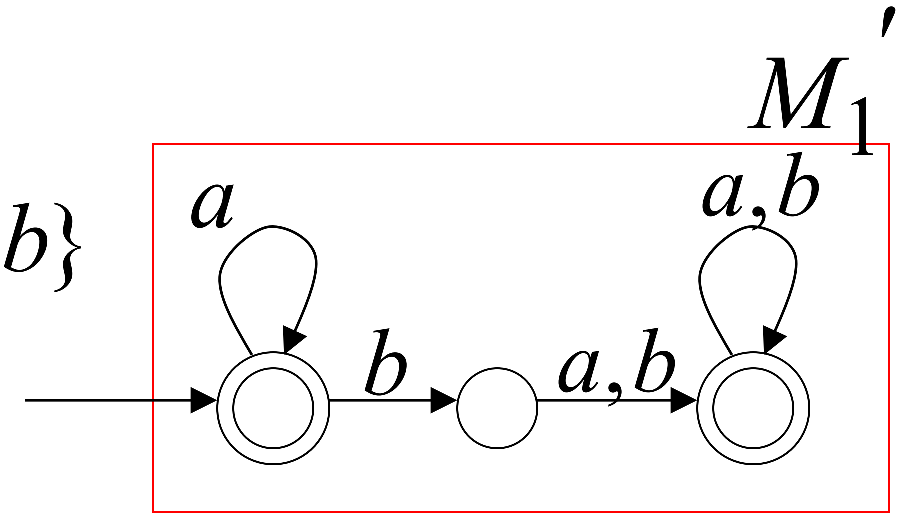
1. Stati non finale diventano finale, e vice-versa, resta lo stato iniziale.

esempio

$$L_1 = \{a^n b\}$$



$$\overline{L_1} = \{a,b\}^* - \{a^n b\}$$



Intersezione

$$\left.\begin{array}{l} L_1 \ \text{regolari} \\ \\ L_2 \ \text{regolari} \end{array}\right\} \xrightarrow{\text{mostriamo}} \begin{array}{c} L_1 \cap L_2 \\ \text{regolari} \end{array}$$

leggi DeMorgan : $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$

$L_1\,,\ L_2$ regolari

$\Rightarrow$ $\overline{L_1}\,,\ \overline{L_2}$ regolari

$\Rightarrow$ $\overline{L_1} \cup \overline{L_2}$ regolari

$\Rightarrow$ $\overline{\overline{L_1} \cup \overline{L_2}}$ regolari

$\Rightarrow$ $L_1 \cap L_2$ regolari

esempio

$$L_1 = \{a^n b\}$$ regolari

$$L_2 = \{ab, ba\}$$ regolari

$$\Rightarrow L_1 \cap L_2 = \{ab\}$$

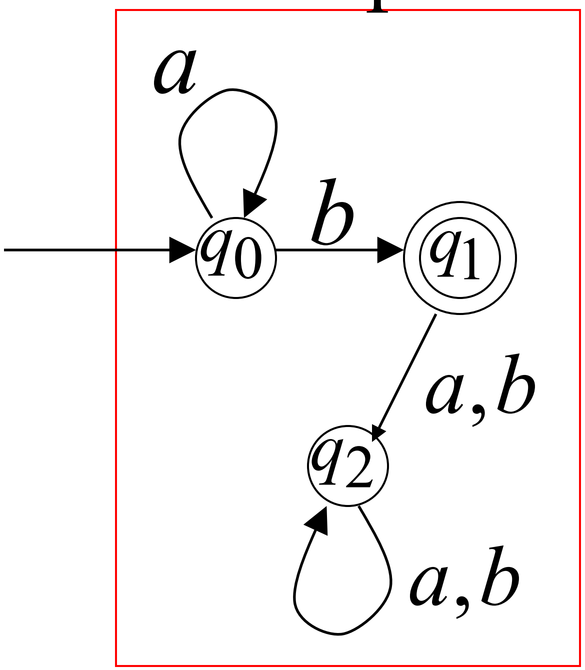
regolari

esempio:

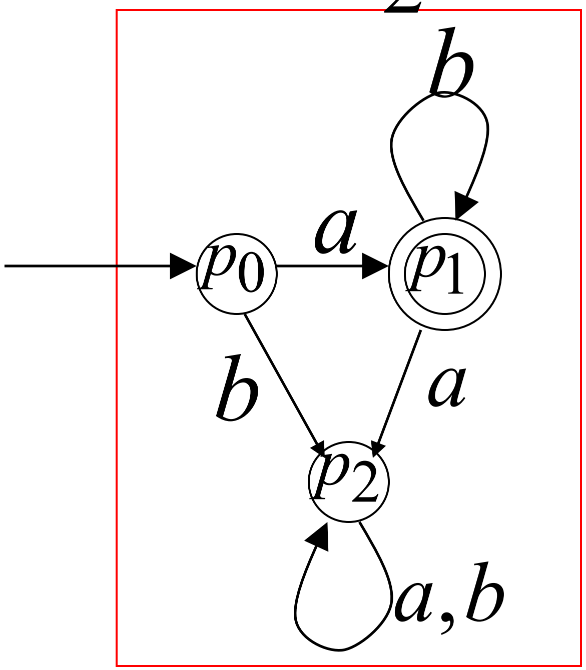
$$L_1 = \{a^n b\} \quad n \geq 0$$

$$L_2 = \{ab^m\} \quad m \geq 0$$

M_1

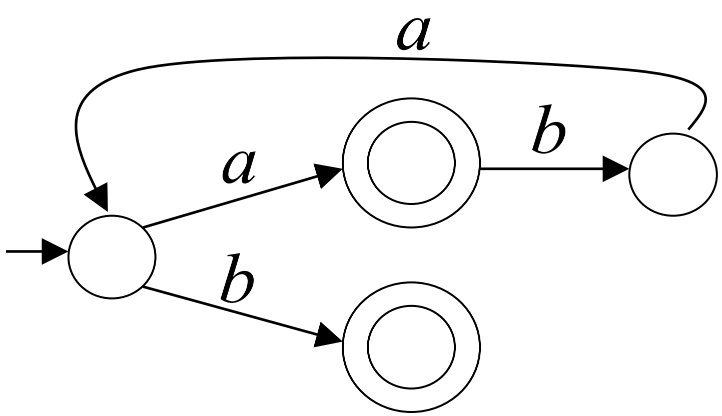


M_2



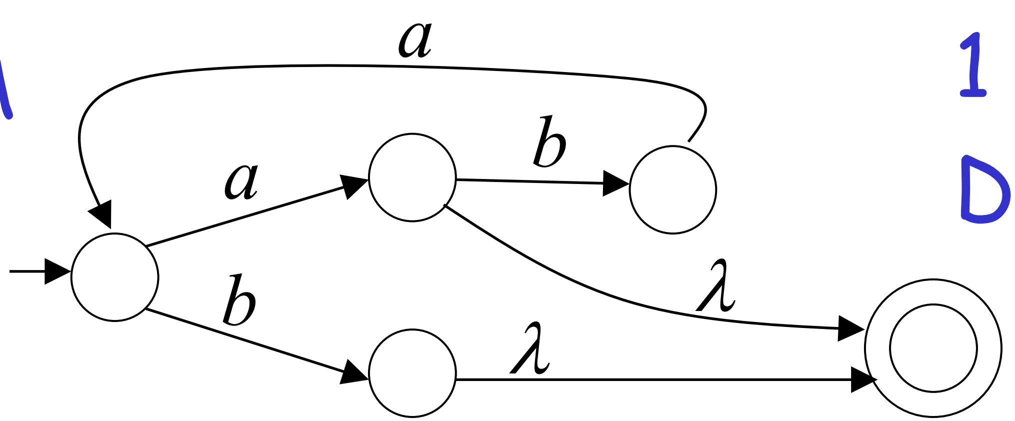
Useremo nfa con un solo stato finale

NFA



2 stati di
accettazione

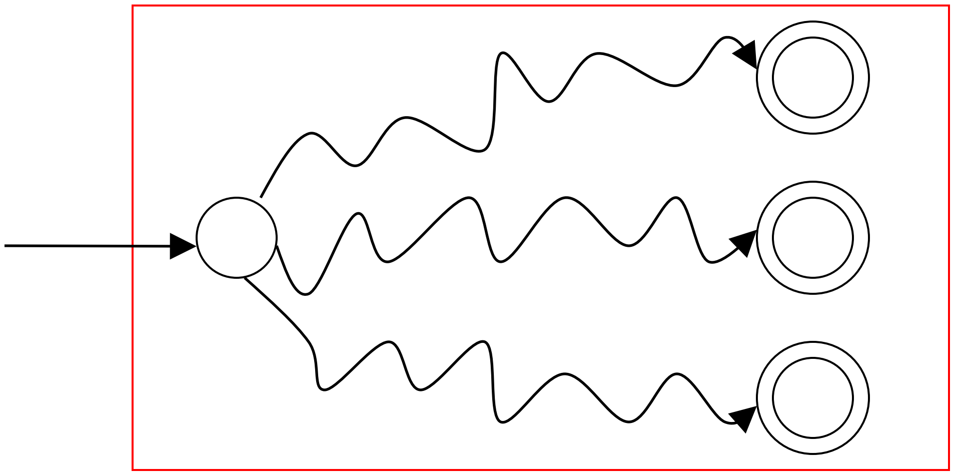
Equivalente
NFA



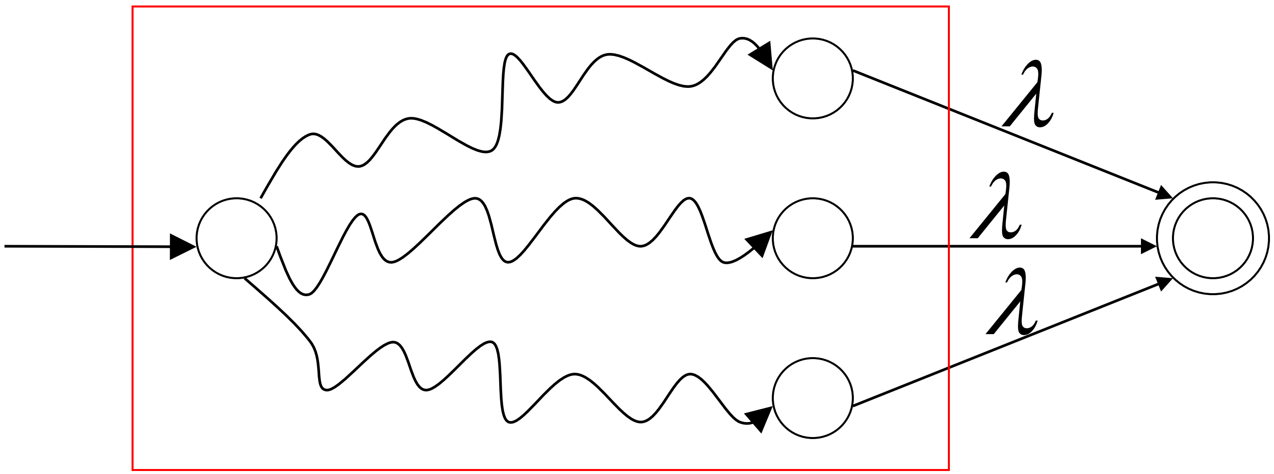
1 solo stato
Di accettazione

In Generale

NFA

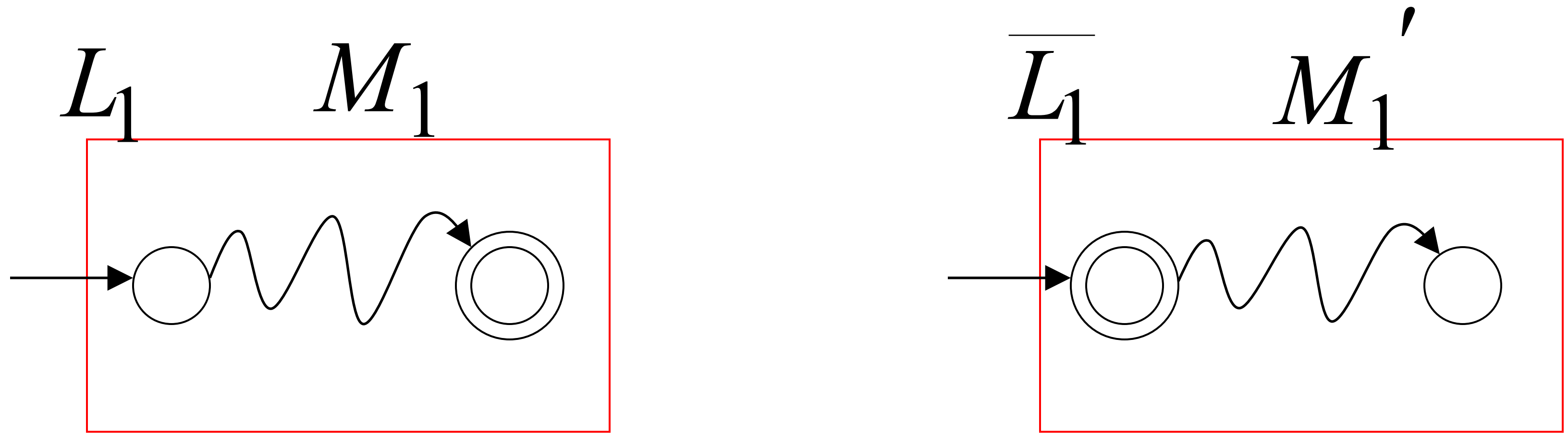


Equivalent NFA



Un solo Stato di accettazione

Complemento



1 prendiamo il DFA che accetta L_1

2. Stati non finale diventano finale, e vice-versa

Chiusura rispetto intersezione

macchina M_1

DFA per L_1

macchina M_2

DFA per L_2

Costruiamo un DFA M che accetta $L_1 \cap L_2$

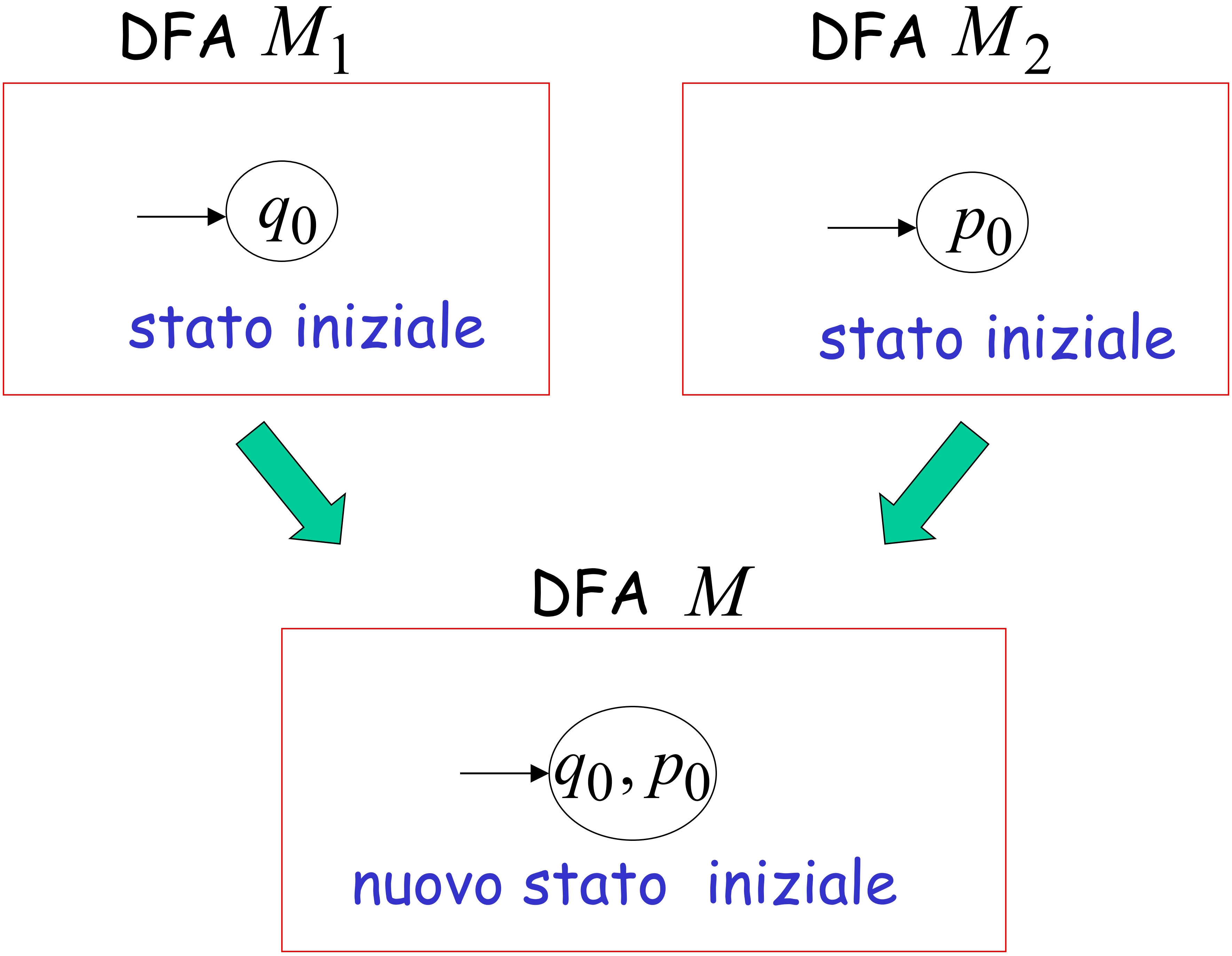
M Simula in parallelo M_1 e M_2

Stati in M

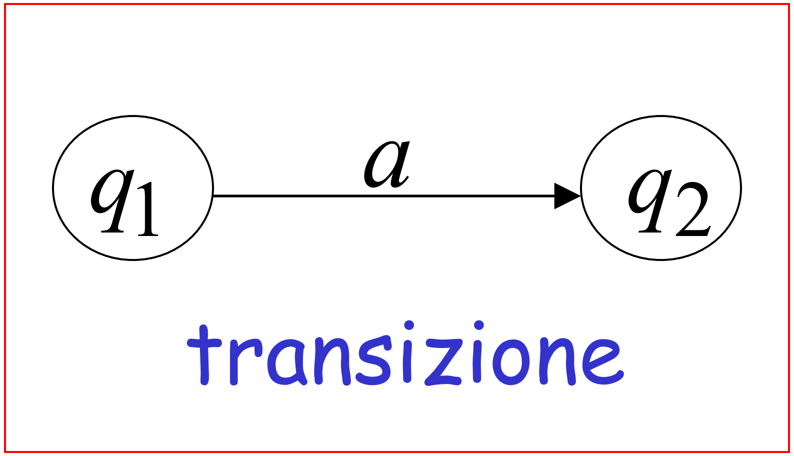
q_i, p_j

Stato in M_1

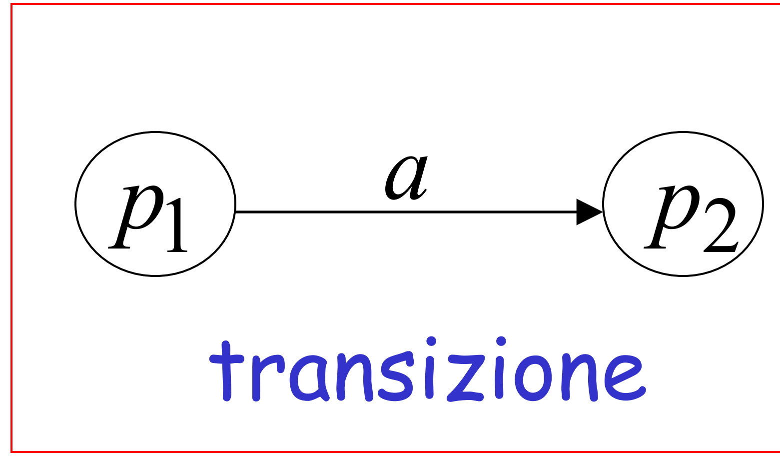
Stato in M_2


DFA M_1
q_0
stato iniziale
DFA M_2
p_0
stato iniziale
DFA M
q_0, p_0
nuovo stato iniziale

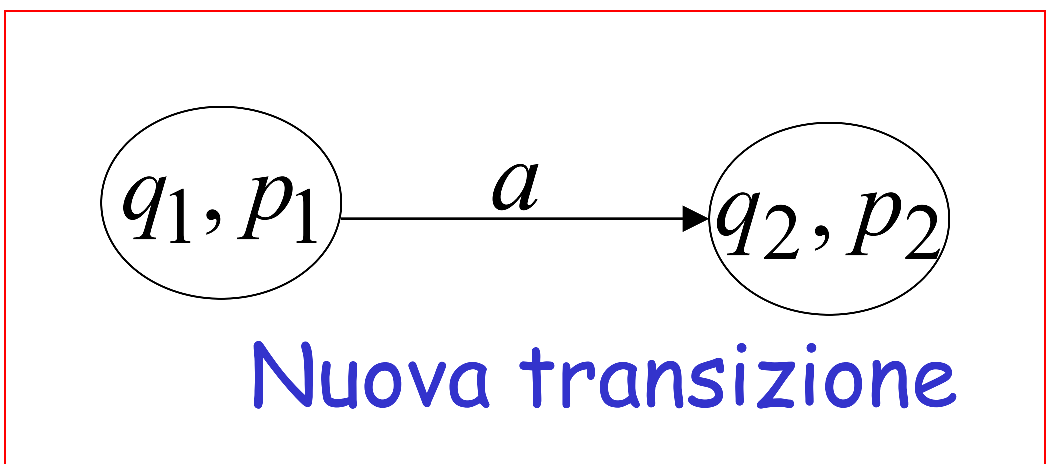
DFA M_1

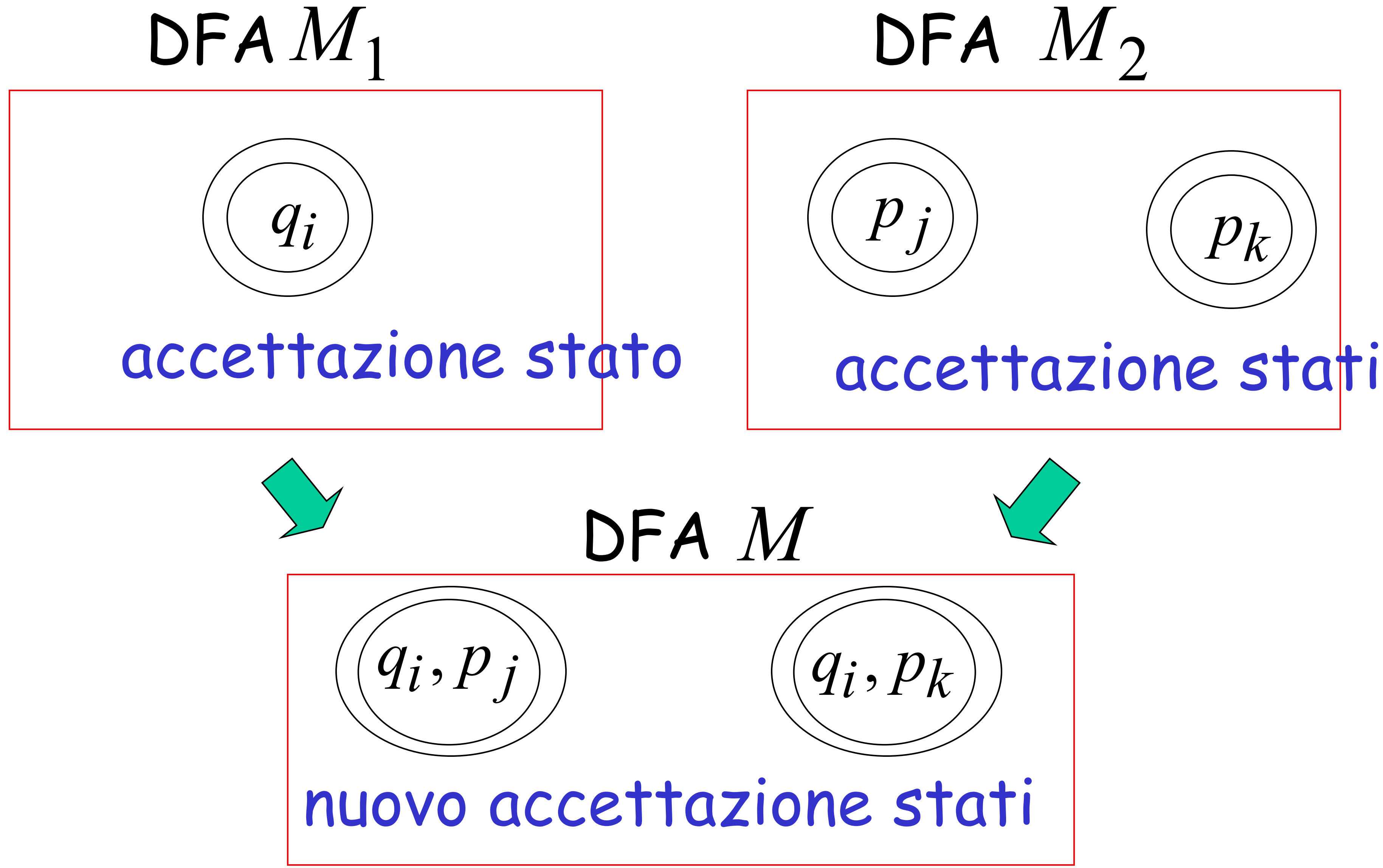


DFA M_2



DFA M



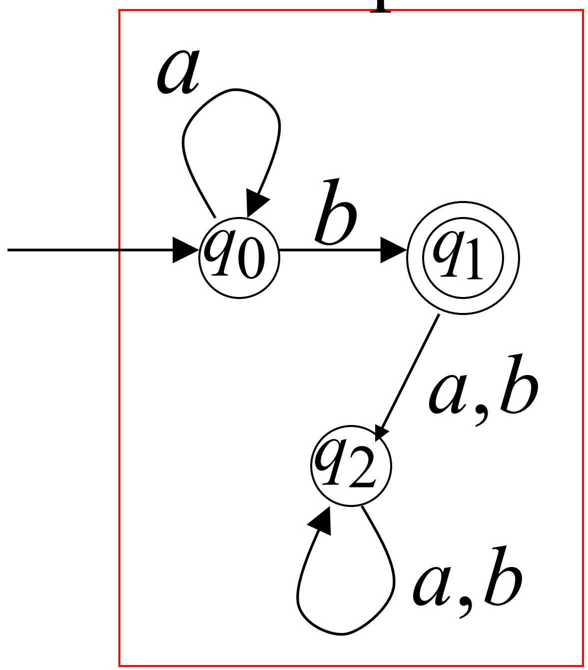

DFA M_1
q_i
accettazione stato
DFA M_2
p_j
p_k
accettazione stati
DFA M
q_i, p_j
q_i, p_k
nuovo accettazione stati

esempio:

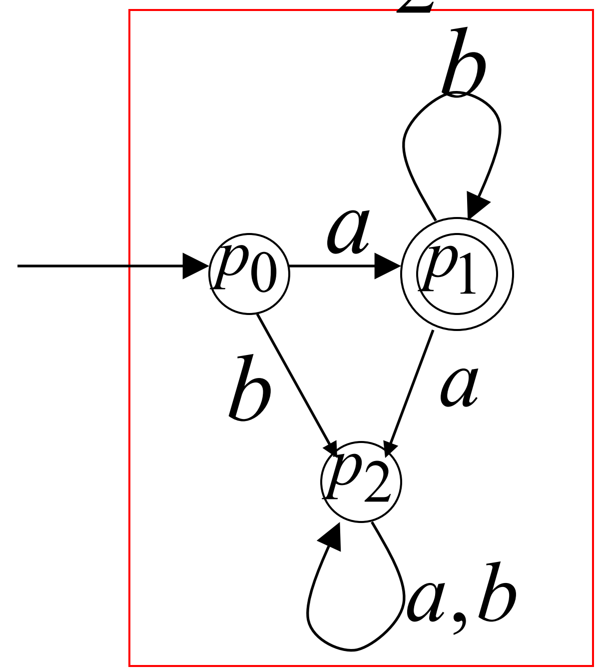
$$L_1 = \{a^n b\} \quad n \geq 0$$

$$L_2 = \{ab^m\} \quad m \geq 0$$

M_1

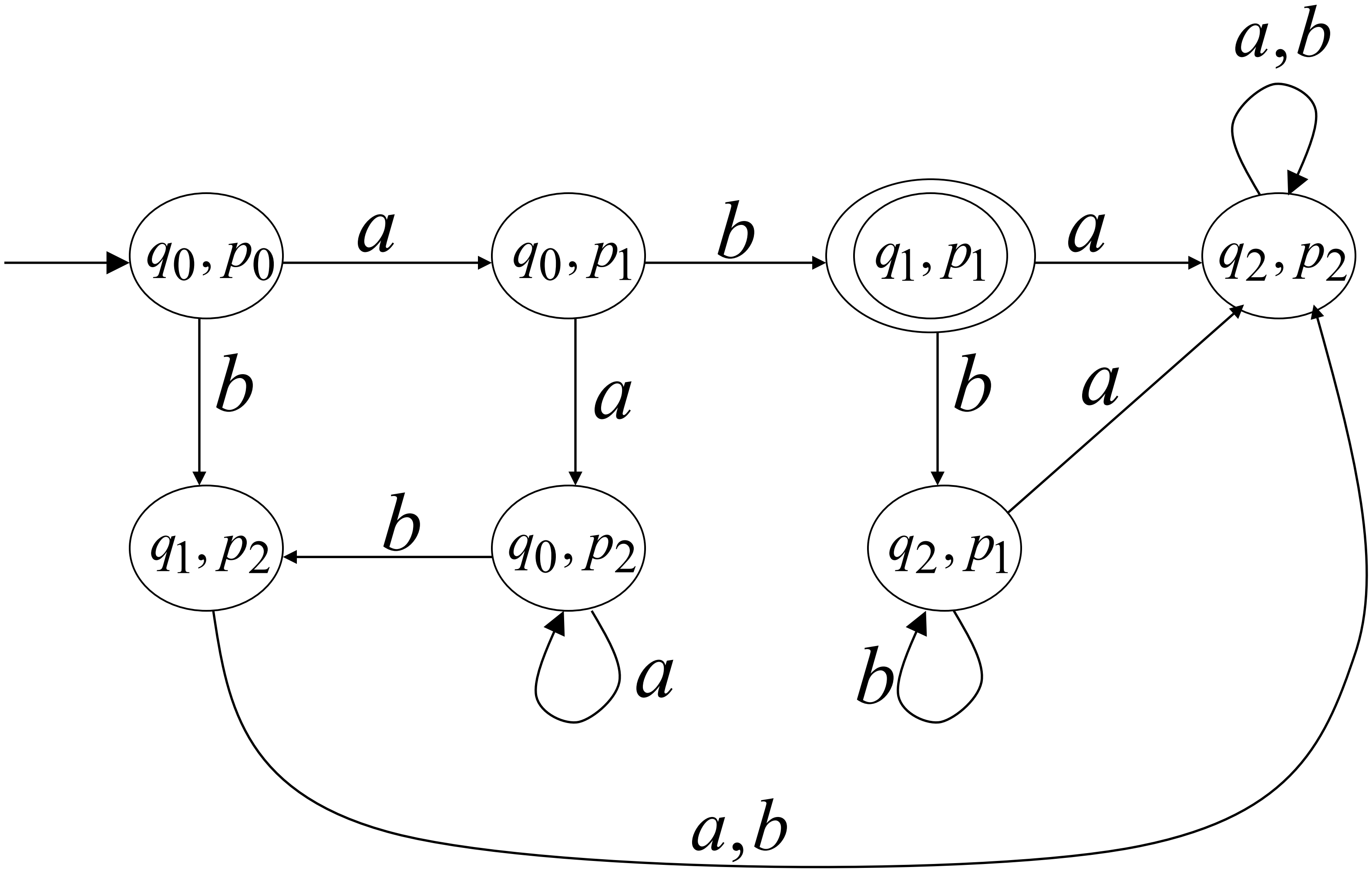


M_2



Intersezione automata

$$L = \{a^n b\} \cap \{ab^n\} = \{ab\}$$



Se appariene ad entrambi

Sia la stringa di lunghezza n
Esistono due cammini di lunghezza n, uno per ogni automa. Dallo stato iniziale a quello finale.
Se lng 1 vero, dimostrare vero per n+1.
Ultimo tratto da n a n+1 arco nei due automa e arco automa costruito. Considera stringa n e considera come stato finale quello prima dello stato finale, vedi arco che riconosce il carattere n, simula i due automi.
Continua ad andare indietro fino a raggiungere lo stato iniziale.

Nell'automa costruito esiste un cammino di lng n che li simula

M Simula in parallelo M_1 e M_2

M accettazione stringa w Se e solo se:

M_1 accetta w string

e M_2 accetta w string

$$L(M) = L(M_1) \cap L(M_2)$$

Espressioni regolari e linguaggi regolari

Teorema

$$\left\{\begin{array}{l}\text{Linguaggi} \\ \text{Generati da} \\ \text{Espressioni regolari}\end{array}\right\} = \left\{\begin{array}{l}\text{Linguaggi} \\ \text{regolari}\end{array}\right\}$$

Dimostrazione - Parte 1

$$\left\{ \begin{array}{l} \text{Linguaggi} \\ \text{Generati da} \\ \text{Espressioni regolari} \end{array} \right\} \subseteq \left\{ \begin{array}{c} \text{Linguaggi} \\ \text{regolari} \end{array} \right\}$$

per ogni espressione regolare r
il linguaggio $L(r)$ è regolare

Dimostrazione per induzione sulla lunghezza

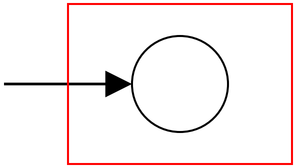
r

Base induzione

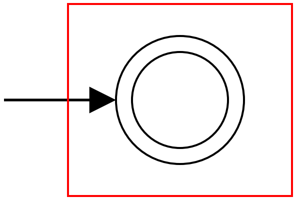
Espressioni regolari di base: $\emptyset, \quad \lambda, \quad \alpha$

corrispondente

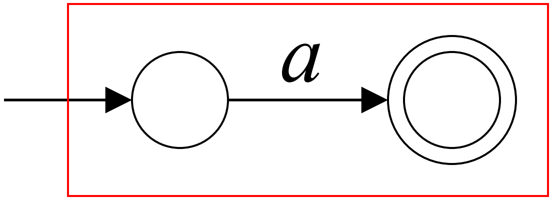
NFAs



$$L(M_1) = \emptyset = L(\emptyset)$$



$$L(M_2) = \{\lambda\} = L(\lambda)$$



$$L(M_3) = \{a\} = L(a)$$

Linguaggi regolari

Ipotesi induttiva

supponi

Per le espressioni regolari r_1 e r_2,

$L(r_1)$ e $L(r_2)$ sono linguaggi regolari.

Esistono due automi uno per ogni linguaggio

Passo induttivo

Proviamo che:

$$L(r_1 + r_2)$$

$$L(r_1 \cdot r_2)$$

$$L(r_1 *)$$

$$L((r_1))$$

Sono linguaggi regolari

Ricorda che, per def. di espressione regolare

$$L(r_1 + r_2) = L(r_1) \cup L(r_2)$$

$$L(r_1 \cdot r_2) = L(r_1)\, L(r_2)$$

$$L(r_1 *) = (L(r_1))*$$

$$L((r_1)) = L(r_1)$$

Per ipotesi induttiva :

$L(r_1)$ e $L(r_2)$ sono linguaggi regolari

Inoltre sappiamo, slides precedenti:

I linguaggi regolari sono chiusi rispetto:

Unione $L(r_1) \cup L(r_2)$

Concatenatzione $L(r_1)\, L(r_2)$

Star $(L(r_1))^*$

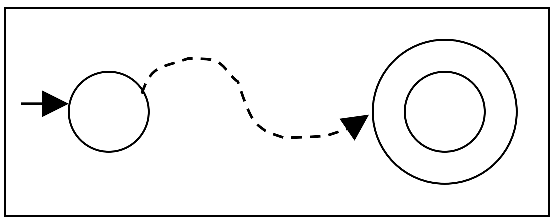
Usando la chiusura dele operazioni

Possiamo costruire un NFA M tale che:

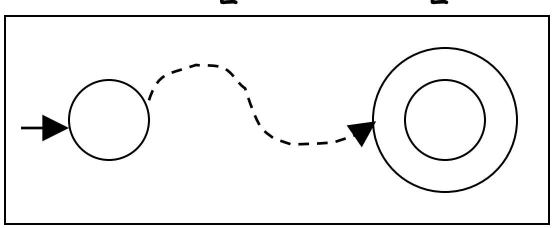
$$L(M) = L(r)$$

esempio: $r = r_1 + r_2$

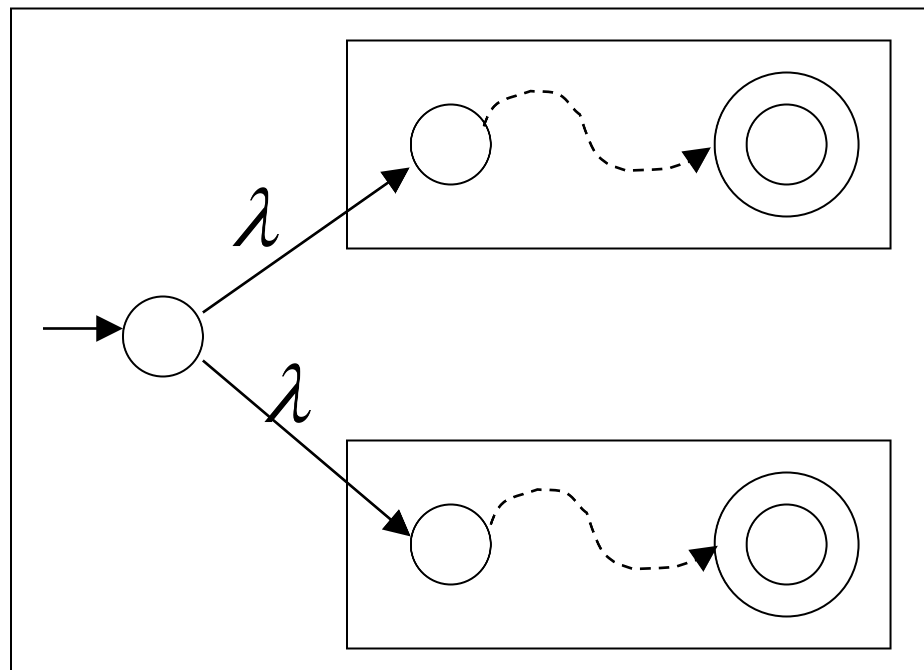
$L(M_1) = L(r_1)$



$L(M_2) = L(r_2)$



$L(M) = L(r)$



Stella e puntino. esercizio

Stella: torna indietro con lambda.
Puntino: collega i finali del primo con l'iniziale con un lambda.

dim - Part 2

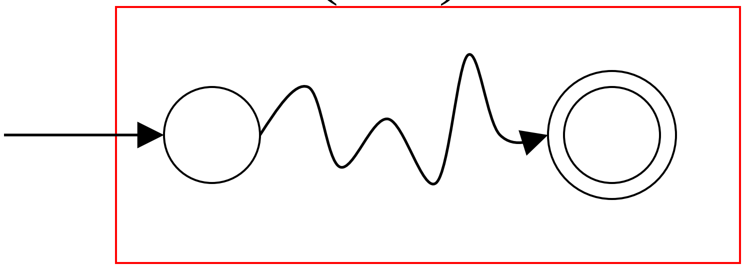
$$\left\{\begin{array}{l}\text{Linguaggi}\\ \text{Generati da}\\ \text{Espressioni regolari}\end{array}\right\} \supseteq \left\{\begin{array}{l}\text{Linguaggi}\\ \text{regolari}\end{array}\right\}$$

Per ogni linguaggio regolare L esiste una espressione regolare r con $L(r) = L$

Convertiremo un NFA che accetta L In una espressione regolare

Poichè L è regolare , allora esiste un NFA M che lo accettà

$$L(M) = L$$



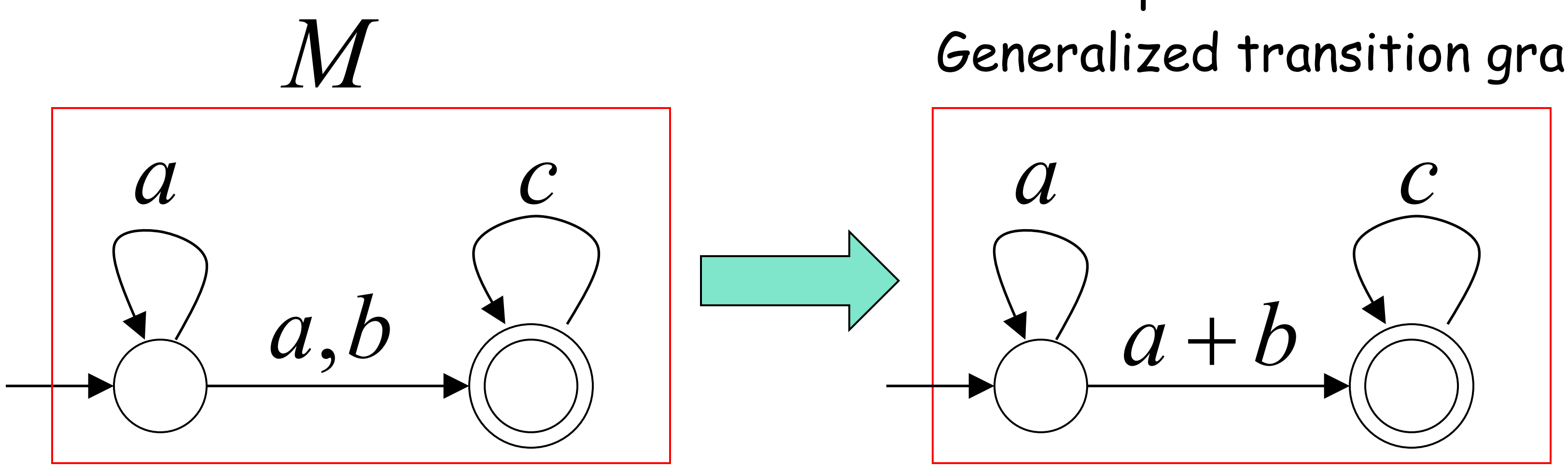
Prendiamo l'automa con un solo stato finale da M costruiamo l'equivalente

Generalized Transition Graph

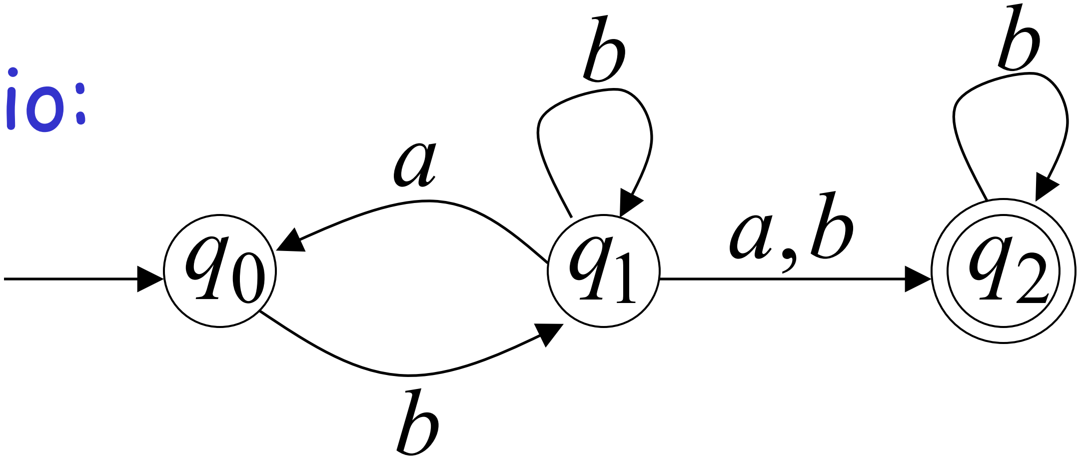
Nel quale i caratteri di transizione, transition labels, sono espressioni regolari

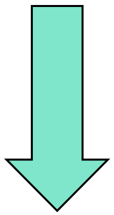
Esempio:

Il corrispondente Generalized transition graph



Un altro esempio:

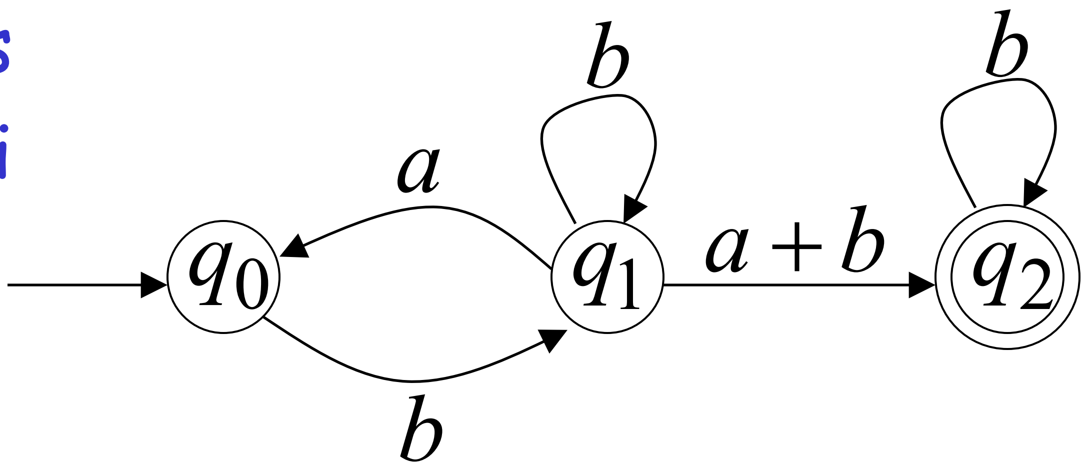




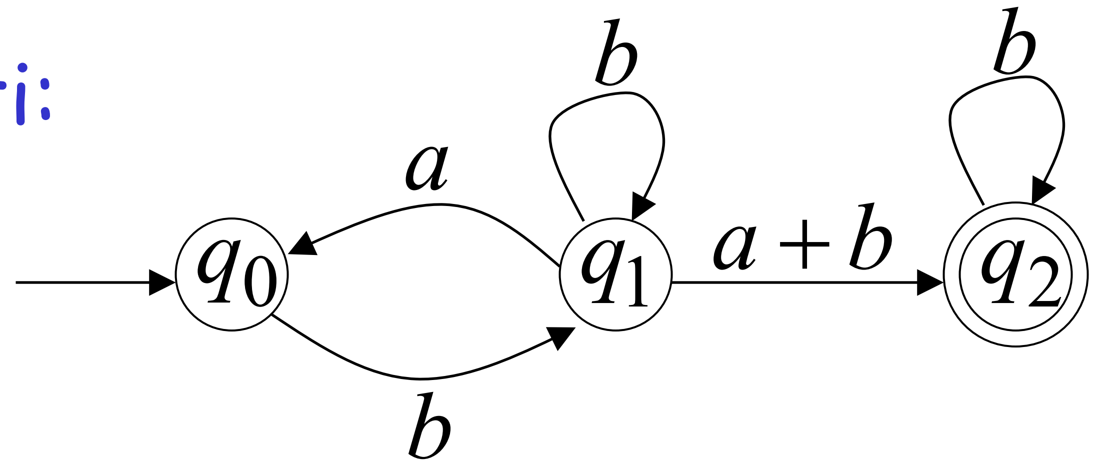
Transition labels

Sono espressioni

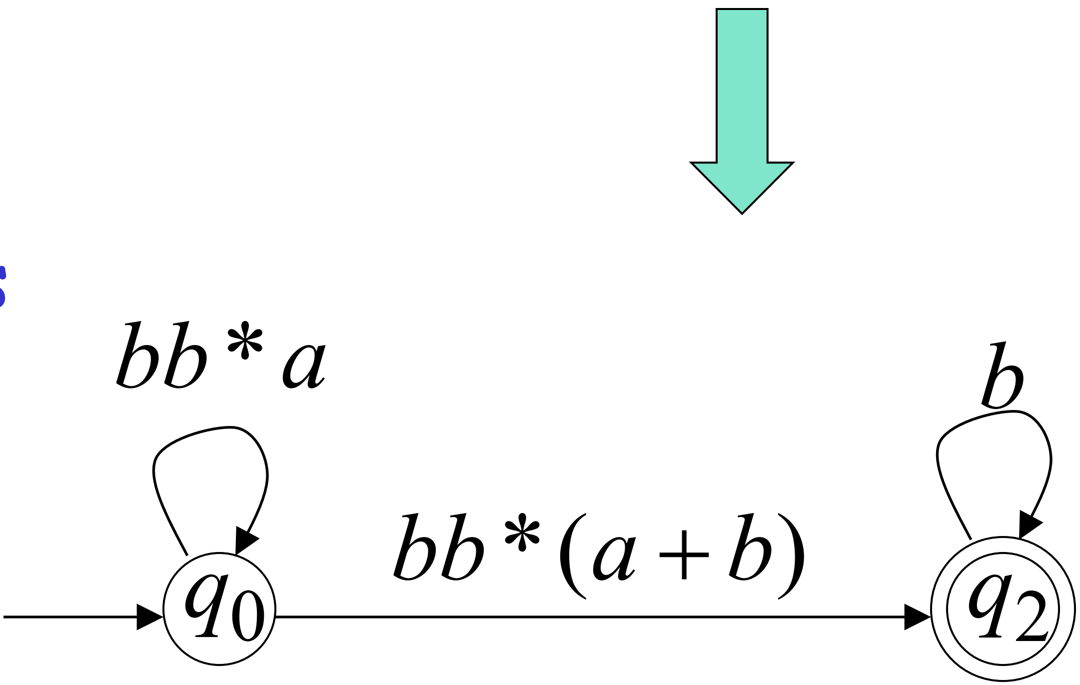
regolari



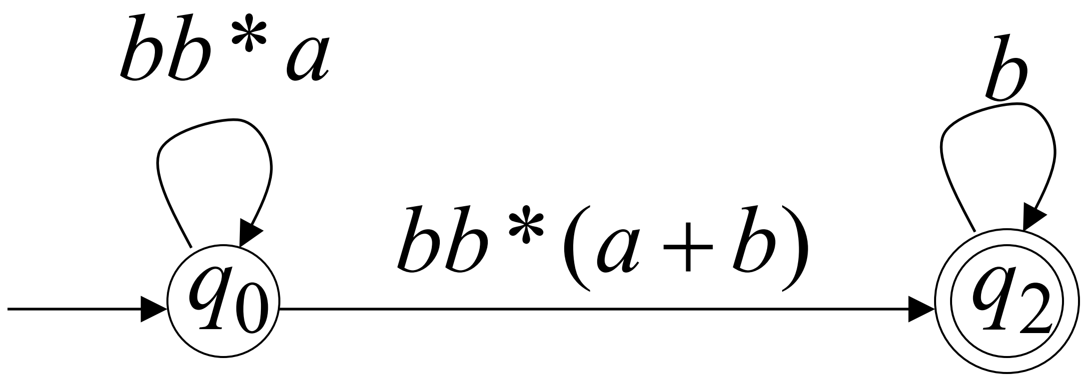
Ridurre gli stati:



Transition labels sono espressioni regolari



Espressione regolare che si ottiene:



$$r=(bb^*a)^*bb^*(a+b)b^*$$

$$L(r)=L(M)=L$$

Stato iniziale solo archi uscenti, nessuno rientrante

Solo uno finale tutti entranti e nessun uscente.

Per gli altri stati sono presenti archi uscenti per tutti gli altri stati ed entranti da tutti gli altri stati e su se stesso. Se non esiste un arco da q_i a q_j creiamo un arco con label insieme vuoto ϕ

Se k=2 slide precedente

Altimenti

prendiamo lo stato da eliminare q

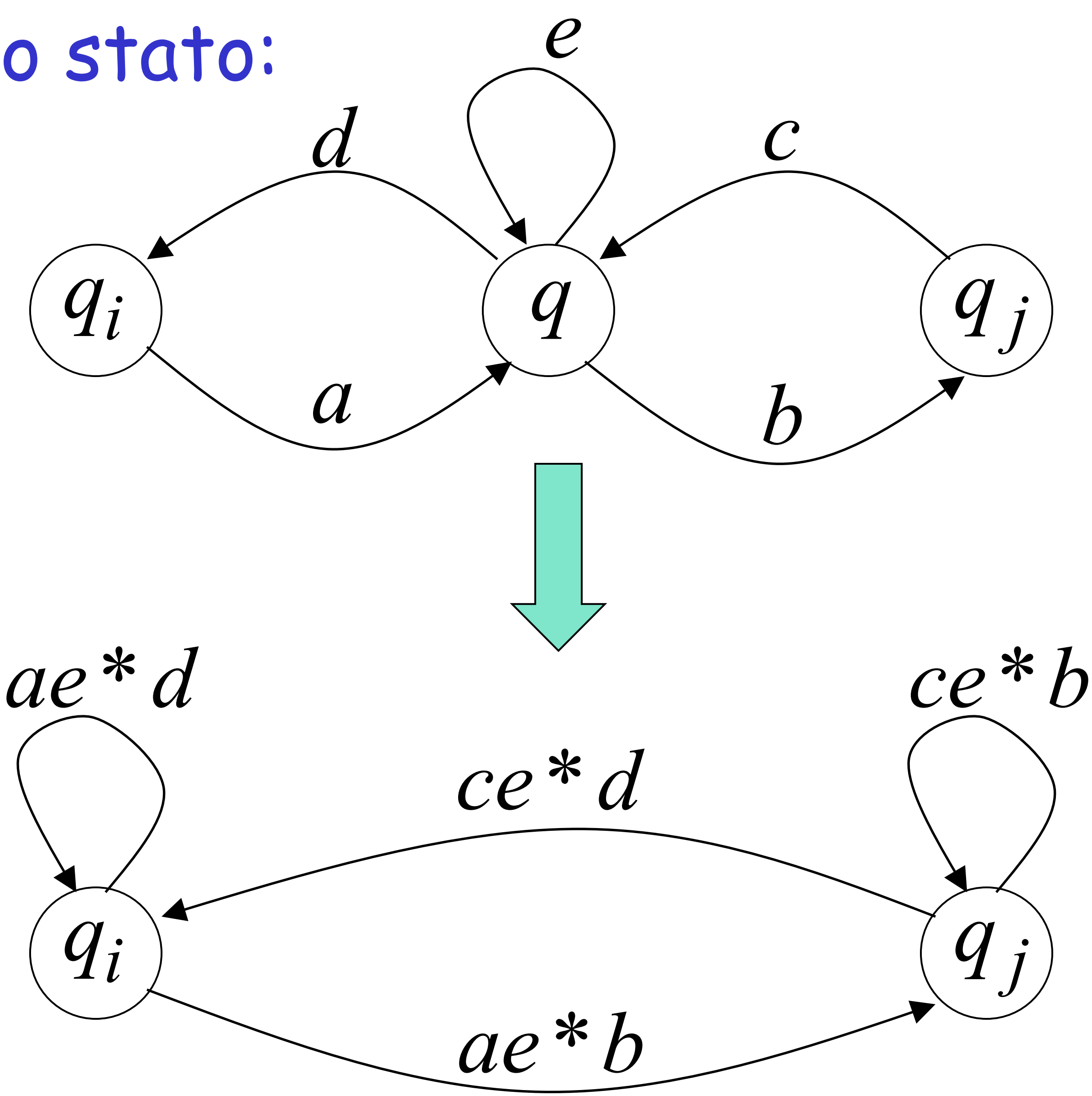
Per ogni q_i e q_j collegati via q

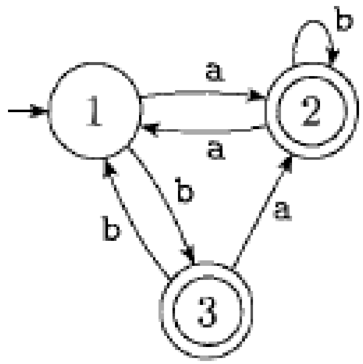
$$\delta^*\left(q\_1, q\_j\right) = (R\_1)(R\_2)^*(R\_3) \cup (R\_4)$$

vai da q_i a q, R_1
gira su q, (R_2)*
vai da q a q_j, R_3
direttamente da q_i a q_j, R_4

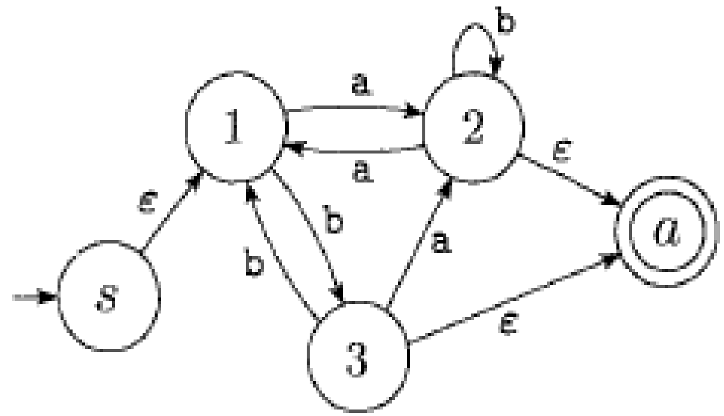
In generale

Rimuovere uno stato:



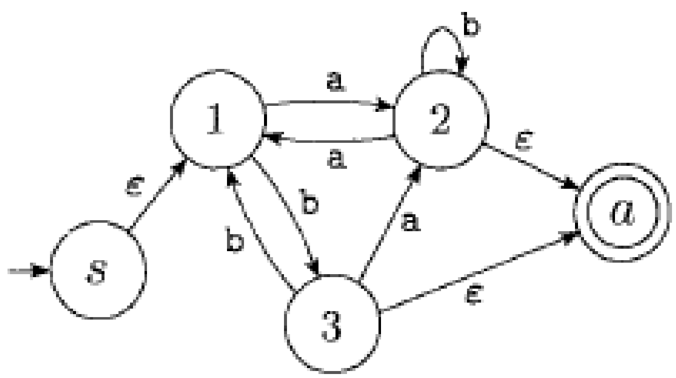


(a)

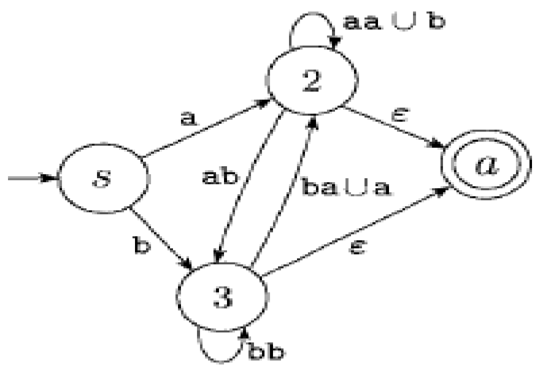


(b)

Per ogni q_i e q_j collegati via q

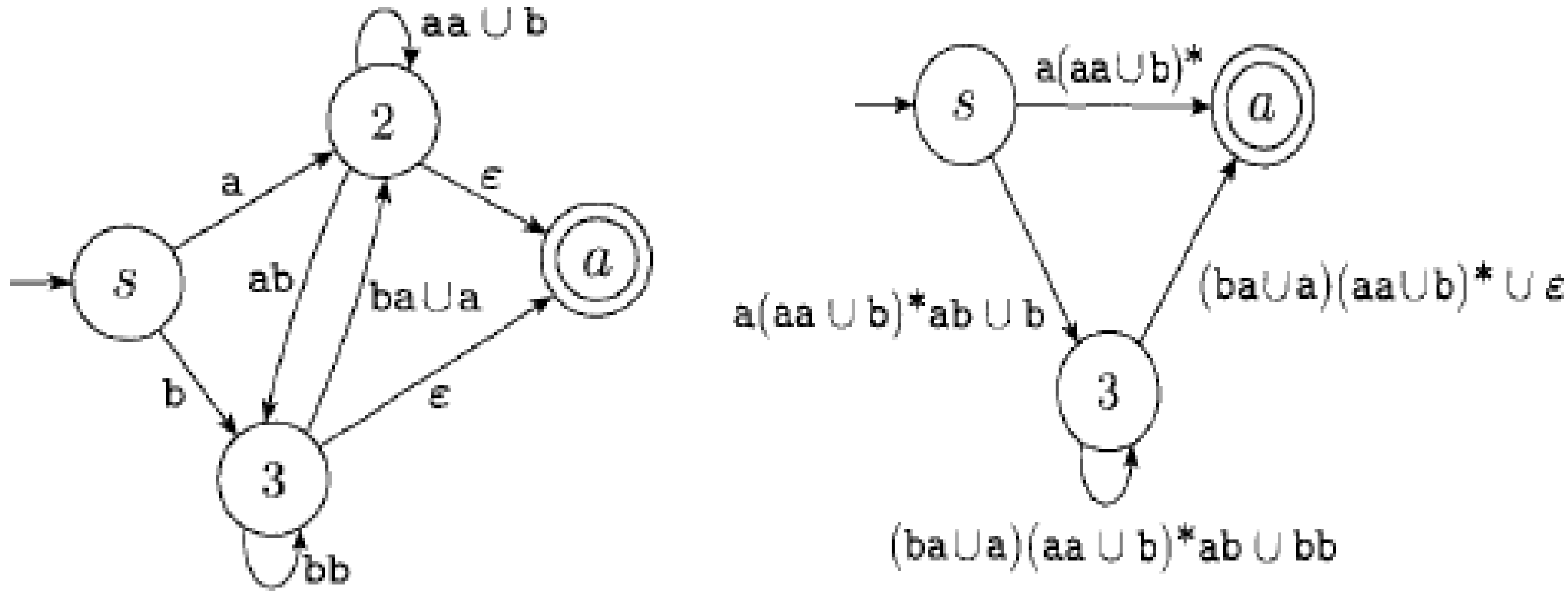


(b)

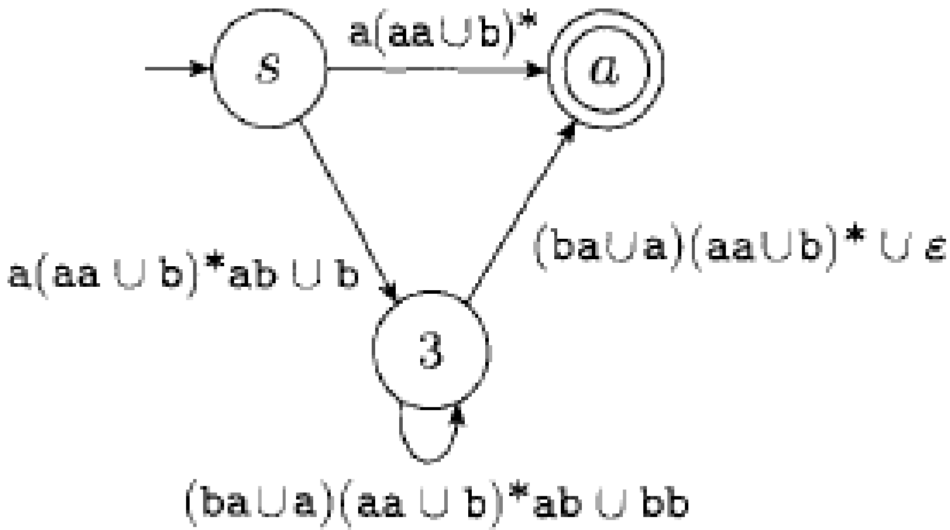


vai da q_i a q, R_1
gira su q, (R_2)*
via da q a q_j, R_3
direttamente da q_i a q_j, R_4

$$\delta^*(q\_1, q\_j) = (R\_1)(R\_2)^*(R\_3) \cup (R\_4)$$



vai da q_i a q, R_1
gira su q, (R_2)*
vai da q a q_j, R_3
direttamente da q_i a q_j, R_4



$(a(aa \cup b)^*ab \cup b)((ba \cup a)(aa \cup b)^*ab \cup bb)^*((ba \cup a)(aa \cup b)^* \cup \varepsilon) \cup a(aa \cup b)^*$

Algoritmo: Eliminare uno stato alla volta fino a che restano 2 stati.

Dimostrazione algoritmo funziona ovvero l'automa iniziale G e G', meno uno stato, accettano lo stesso linguaggio

G' con due stati allora otteniamo espressione regolare.

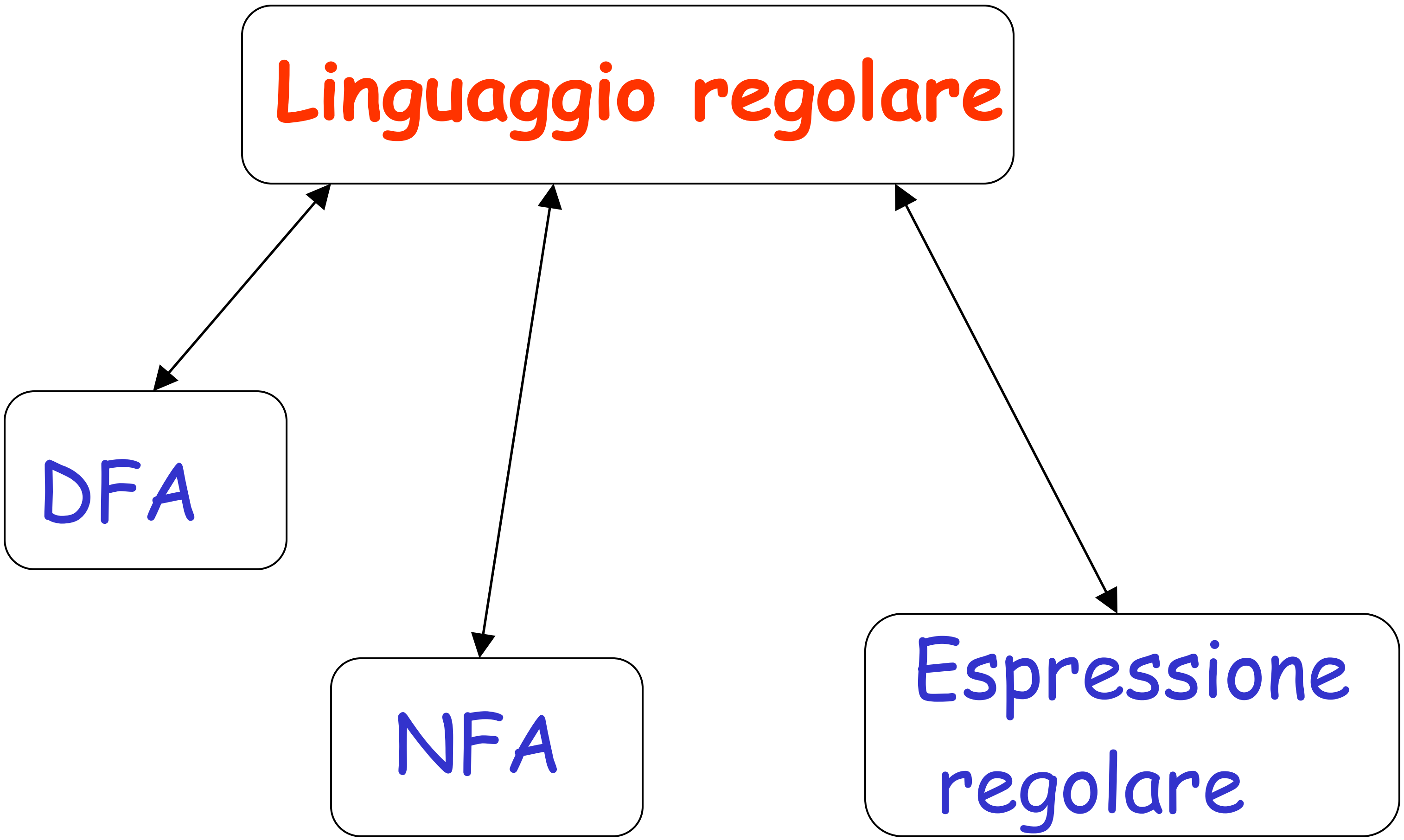
Vero per k-1 provare per k+1.

Prendiamo una stringa che viene accettata, esiste un cammino che accetta la stringa se non usa lo stato da eliminare bene G e G' accettano la stringa. In slang:

«Se G non usa lo stato da eliminare: bene G' «si fa lo stesso giro» »;

Se G usa lo stato da eliminare allora in G' lo stato in oggetto non esiste ma nei nuovi archi tutte le sottostringhe che venivano riconosciute tramite lo stato eliminato sono descritte dalle espressioni regolari sugli archi

Presentazione standard di un linguaggio regolare



esercizi
slide successive

esempio

Epressione regolare: $(a+b)\cdot a^*$

$$
\begin{aligned}
L((a+b)\cdot a^*) &= L((a+b))\,L(a^*) \\
&= L(a+b)\,L(a^*) \\
&= (L(a)\cup L(b))\,(L(a))^* \\
&= (\{a\}\cup\{b\})\,(\{a\})^* \\
&= \{a,b\}\,\{\lambda, a, aa, aaa, \ldots\} \\
&= \{a, aa, aaa, \ldots, b, ba, baa, \ldots\}
\end{aligned}
$$

Linguaggi associati alle espressioni regolari

$L(r)$: linguaggio associato all'espressione r

esempio

$$L((a+b\cdot c)^*)=\{\lambda, a, bc, aa, abc, bca, \ldots\}$$

Esempio

Espressione regolare $r = (a+b)^*(a+bb)$

$$L(r) = \{a, bb, aa, abb, ba, bbb, \ldots\}$$

esempio

Espressione regolare $r = (aa)^*(bb)^*b$

$$L(r) = \{a^{2n}b^{2m}b : \quad n,m \geq 0\}$$

esempio

Espressione regolare $r = (0+1)*00\,(0+1)*$

$L(r)$ = { tutte le stringhe che contengono 00 }

esempio

Espressione regolare $r = (1 + 01)^*(0 + \lambda)$

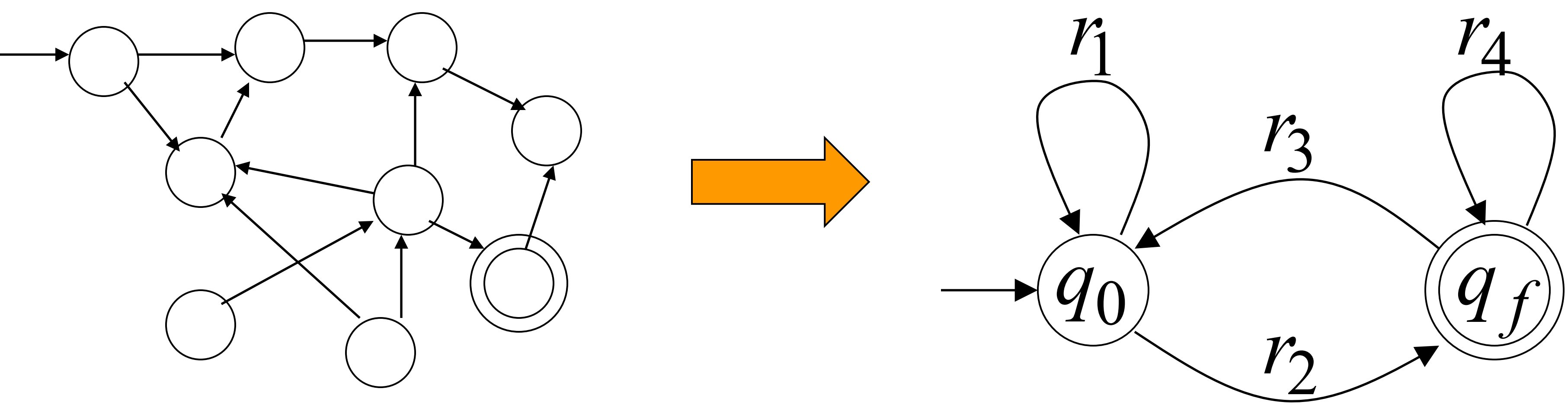
$L(r)$

= { tutte le stringhe senza sottostringhe 00 }

Ripetere il processo finchè
Due stati restano il grafo risultante
sarà il seguente

Grafo iniziale

Grafo risultante



L'espressione regolare risultante:

$$r = r_1 {}^* r_2 (r_4 + r_3 r_1 {}^* r_2)^*$$

$$L(r) = L(M) = L$$