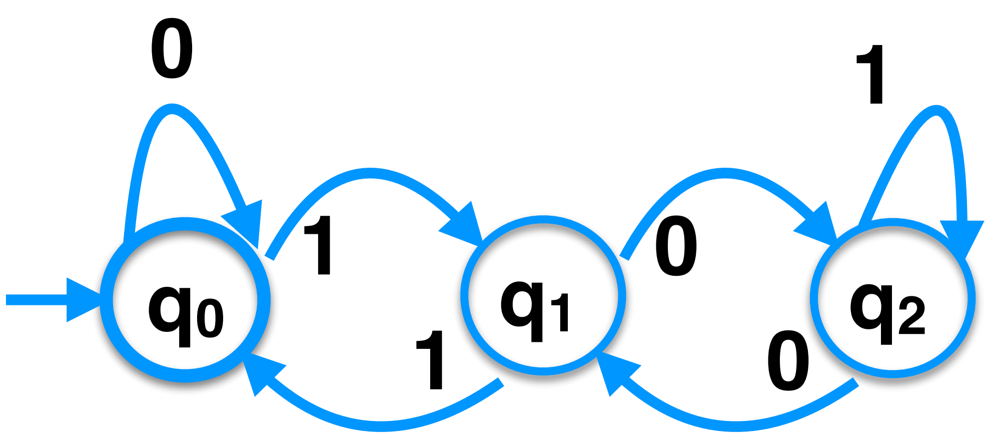
In questa lezione

Proprietà di chiusura rispetto a unione, intersezione e complemento

ESEMPIO 1

L= {x | x in {0,1}* e x rappresenta in binario un intero multiplo di 3}



ESEMPIO 2

L= { x | x in {0,1}* e |x| = 3} è l'insieme (finito) delle parole binarie di lunghezza 3

Operazioni sui linguaggi

Ricordiamo le operazioni insiemistiche:

$L, L' \subseteq \Sigma^*$

- **unione: $L \cup L'$**
- **intersezione: $L \cap L'$**
- **complemento: $\neg L = \Sigma^* - L$**

- **Ricordiamo che queste operazioni non sono indipendenti:**
- **$\neg(X \cup X') = \neg X \cap \neg X'$**
- **$\neg(X \cap X') = \neg X \cup \neg X'$**

Il concetto di CHIUSURA

Una classe di linguaggi C è chiusa rispetto a un' operazione binaria ^ se, dati X,Y in C, allora anche X^Y è in C.

La definizione è analoga in casi di diversa arità (numero di argomenti) dell'operazione

L(DFA) è chiusa rispetto a varie importanti operazioni: unione, intersezione, complemento, concatenazione e stella di Kleene.

PROPRIETÀ DI CHIUSURA

Utilizzando le proprietà di chiusura si possono costruire automi molto complessi la cui correttezza è relativamente facile da provare:

1. si costruiscono gli automi di partenza, di una semplicità tale che la loro correttezza possa essere verificata facilmente in modo diretto

2. si utilizzano gli algoritmi di chiusura per le opportune operazioni ottenendo gli automi voluti.

La correttezza degli automi risultanti è garantita dalla correttezza degli algoritmi di chiusura.

PRIME CHIUSURE: OPERAZIONI INSIEMISTICHE

L(DFA) è chiusa rispetto alle operazioni insiemistiche: unione, intersezione, complemento.

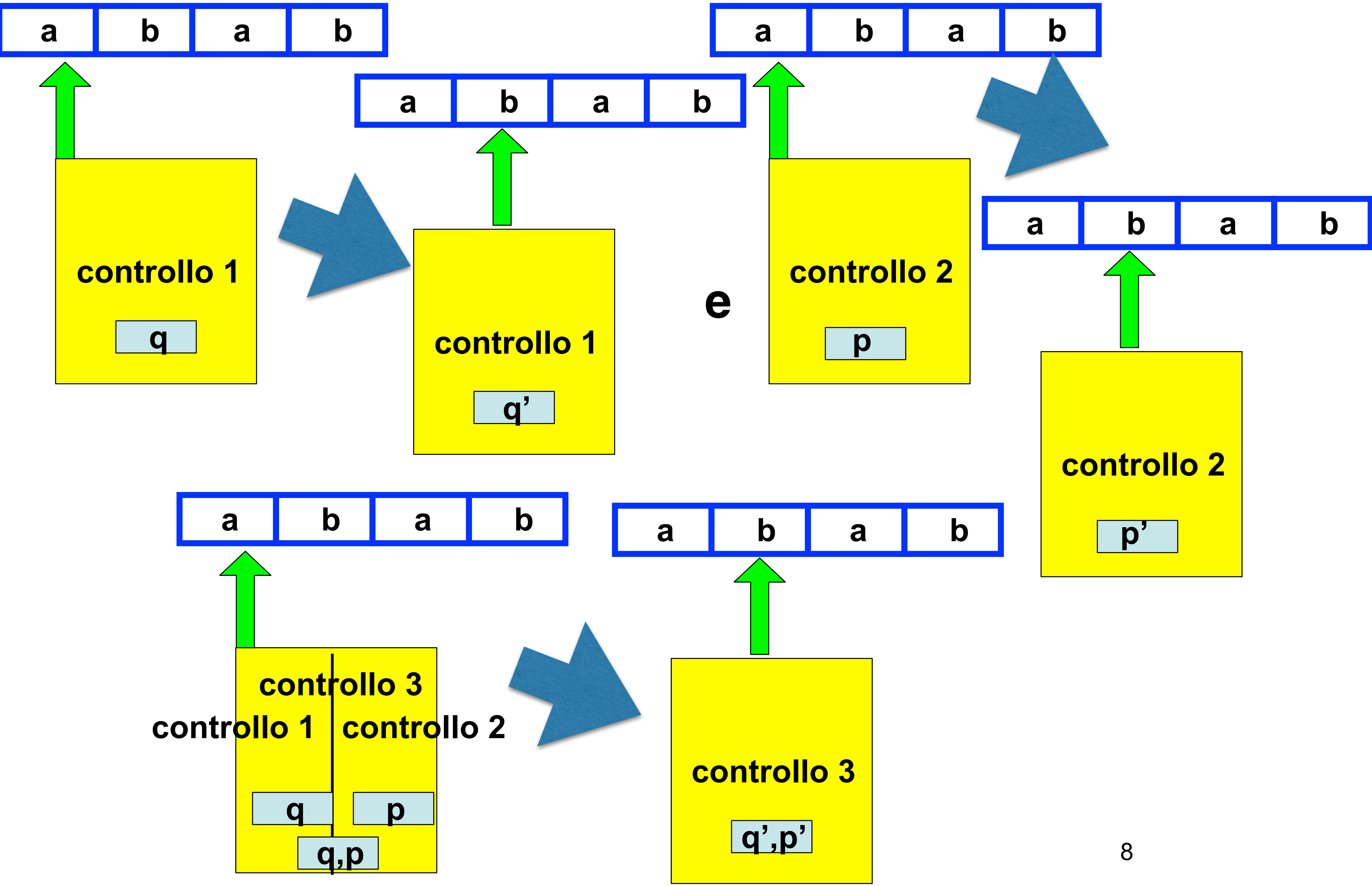
Proveremo quindi che

$L, L' \in \mathcal{L}(DFA) \Rightarrow L \cup L' \in \mathcal{L}(DFA)$

$L, L' \in \mathcal{L}(DFA) \Rightarrow L \cap L' \in \mathcal{L}(DFA)$

$L \in \mathcal{L}(DFA) \Rightarrow \neg L \in \mathcal{L}(DFA)$

idea della prova chiusura unione



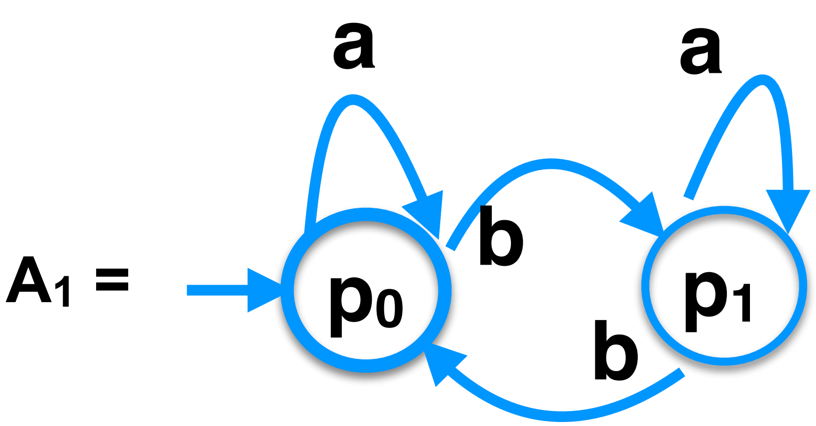
Costruzione formale

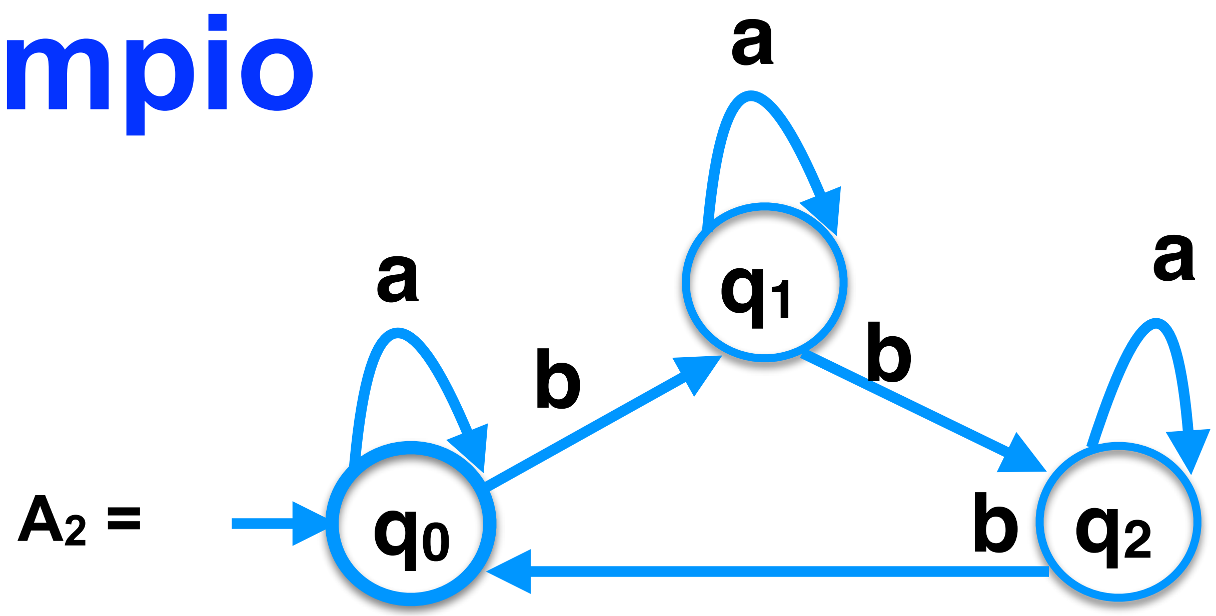
Dati due DFA $M_1 = (Q_1,\Sigma,\delta_1,q_0,F1)$ e $M_2 = (Q_2,\Sigma,\delta_2,p_0,F_2)$, è possibile costruire due DFA M e M' che accettano $L(M_1) \cup L(M_2)$ e $L(M_1) \cap L(M_2)$, rispettivamente, e cioè con $L(M) = L(M_1) \cup L(M_2)$ e $L(M') = L(M_1) \cap L(M_2)$.

Prova. Definiamo il DFA $M = (Q_1XQ_2,\Sigma,\delta,(q_0,p_0),F)$,
con $\delta((q,p),a) = (\delta_1(q,a),\delta_2(p,a))$.
Con questa definizione di funzione di transizione, si può dimostrare che $\delta^*((q,p),x) = (\delta_1^*(q,x),\delta_2^*(p,x))$ per ogni $x \in \Sigma^*$ per induzione sulla lunghezza dell'input. Chiamiamo questo automa l'automa prodotto di M_1 e M_2.
Definendo $F = F_1XQ_2 \cup Q_1XF_2$, si accettano tutte le parole accettate da M_1 o da M_2 e quindi $L(M) = L(M_1) \cup L(M_2)$.
Il DFA $M' = (Q_1XQ_2,\Sigma,\delta,(q_0,p_0),F')$ è l'automa prodotto , con stati finali $F' = F_1XF_2$. M' accetta tutte e sole le parole accettate da entrambi e quindi $L(M') = L(M_1) \cap L(M_2)$

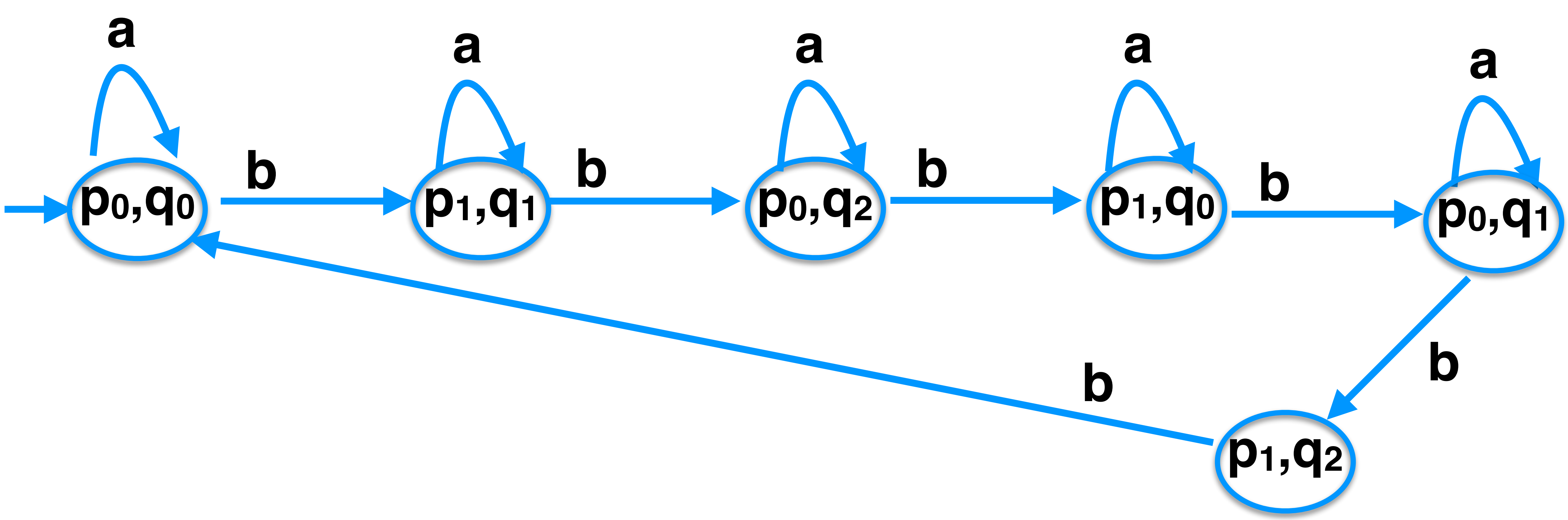
Qui l'ipotesi della totalità della funzione di transizione semplifica la prova!

Esempio

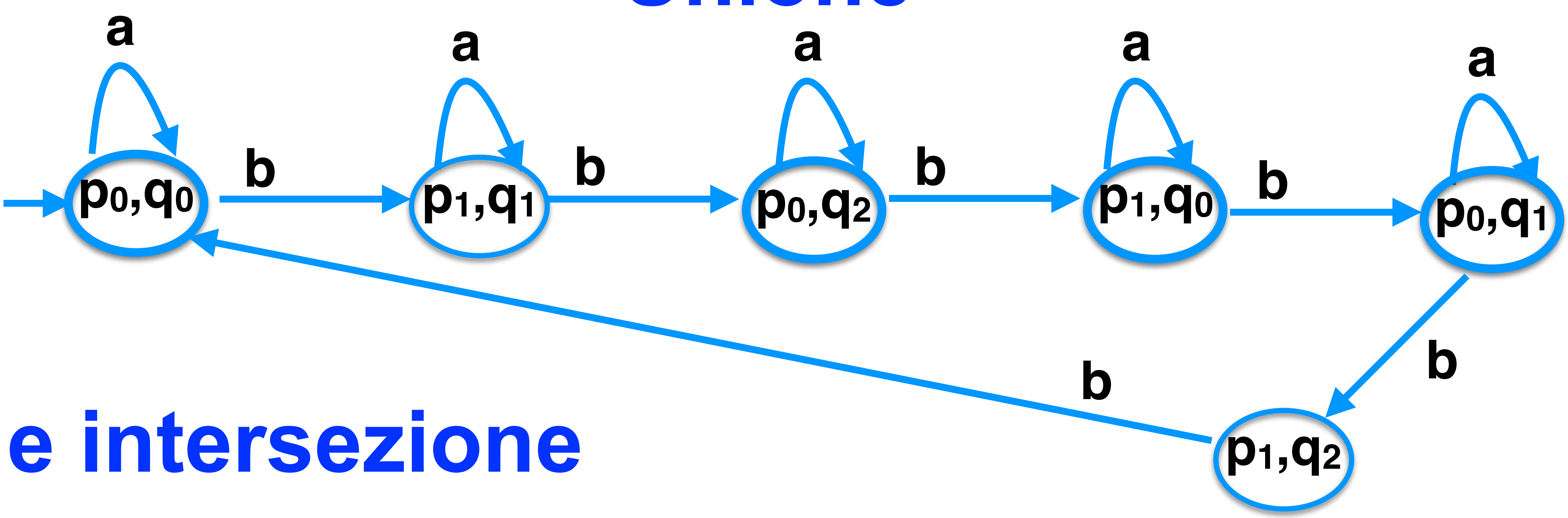




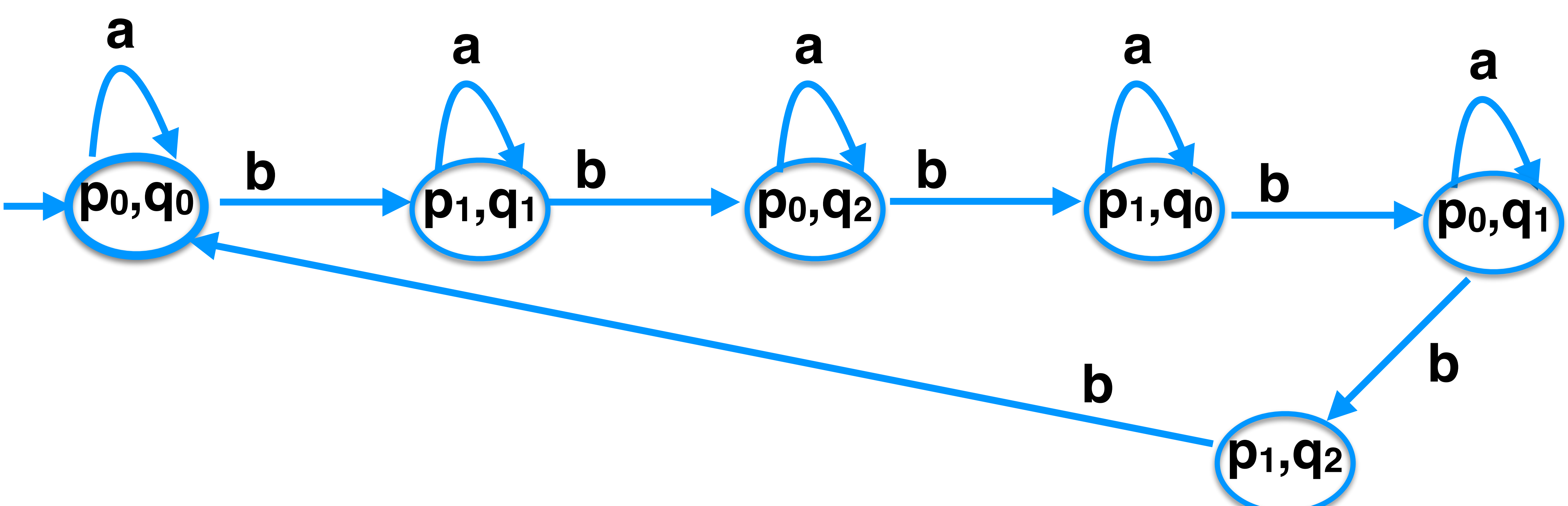
Automa prodotto



Unione



e intersezione



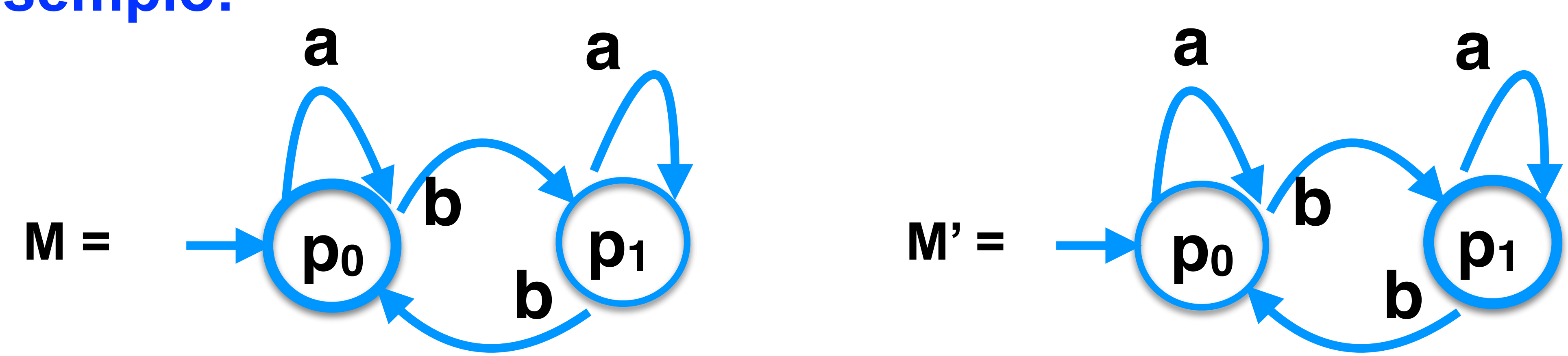
a a

Complemento

Dato un DFA $M = (Q,\Sigma,\delta,q_0,F)$ è possibile costruire un DFA M' che accetta $\Sigma^* - L(M) = \neg L(M)$.

Prova. Si consideri il DFA $M' = (Q,\Sigma,\delta,q_0,Q\text{-}F)$.
Si può osservare che
$x \notin L(M) \Leftrightarrow \delta^*(q_0,x) \notin F \Leftrightarrow \delta^*(q_0,x) \in Q - F \Leftrightarrow x \in L(M')$

Esempio:



Anche qui l'ipotesi della totalità della funzione di transizione semplifica la prova!

Nota: avendo dimostrato che L(DFA) è chiusa rispetto all'unione e al complemento si sarebbe potuto dedurre la chiusura rispetto all'intersezione, grazie alle leggi di De Morgan.

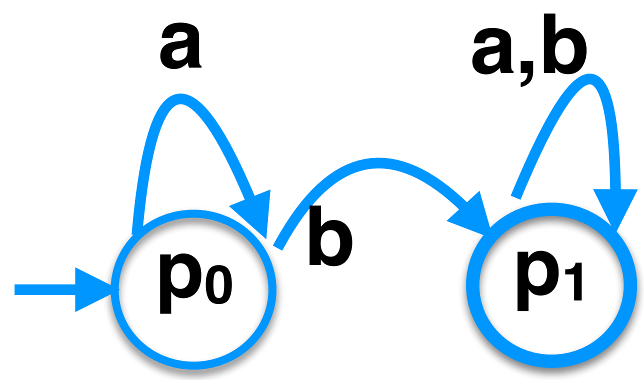
Automa prodotto

Algoritmo AutomaProdotto
input: due DFA $A_1=(Q_1,\Sigma,q_0,F_1,\delta_1)$ e $A_2=(Q_2,\Sigma,p_0,F_2,\delta_2)$
inserisci nell'insieme Q lo stato (q_0,p_0) non marcato
finché ci sono stati non marcati in Q
prendi (p,q) non marcato in Q
per ogni simbolo a in Σ
prendi (p,q) non marcato in Q
posto $r = \delta_1(p,a)$ e $t = \delta_2(q,a)$,
aggiungi (r,t) non marcato a Q, se non è già presente
marca (p,q)

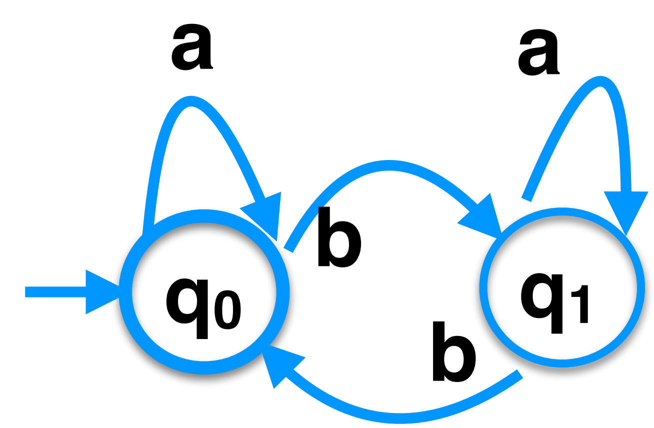
L'insieme degli stati marcati è l'insieme degli stati dell'automa prodotto (ridotto) e la funzione δ la sua funzione di transizione.
Poi scegliendo l'insieme degli stati finali si ottiene l'automa per l'unione o per l'intersezione.

Esempio

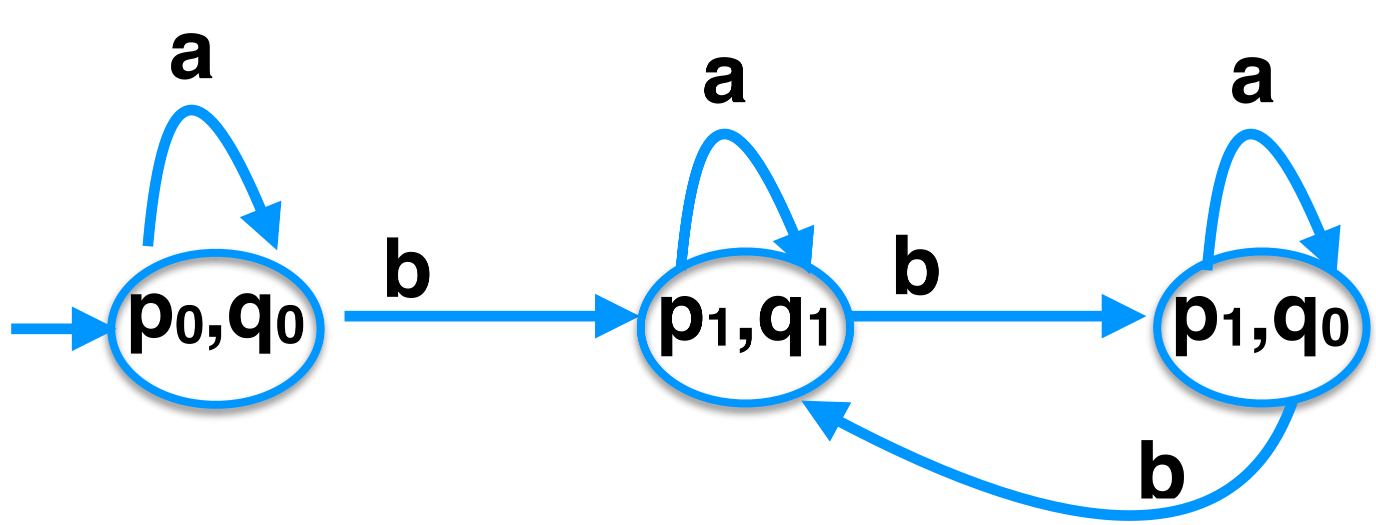
$A_1 =$



$A_2 =$



Automa prodotto costruito seguendo l’algoritmo



Automa prodotto costruito seguendo il teorema