# AUTOMI A PILA

**Introdotti da A. G. Oettinger in "Automatic Syntactic Analysis and the pushdown store" Proc. Symp. Applied Math., 1961 e da**

**M.P. Schutzenberger in "Context free languages and pushdown automata" Information and Control, 1963**

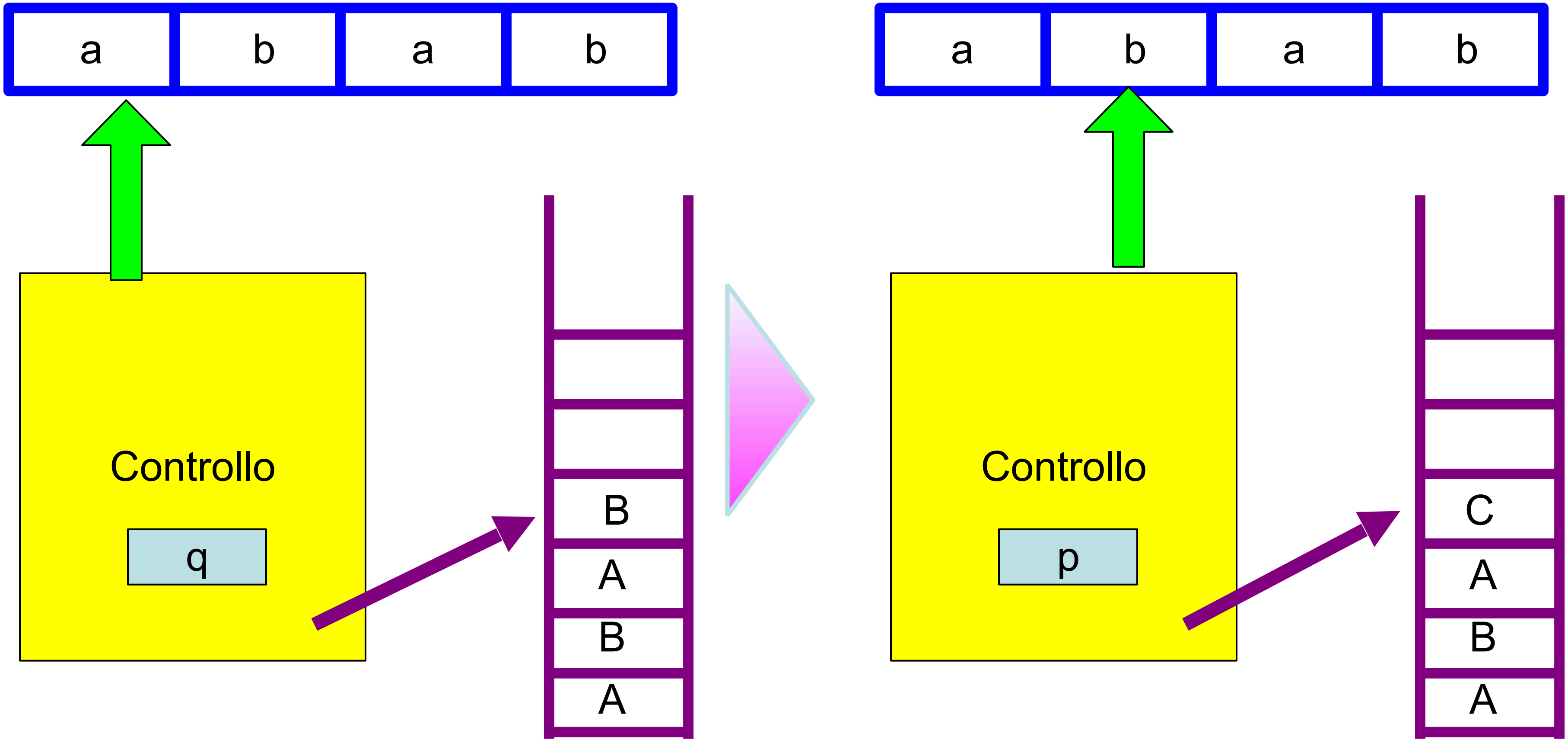
**M.P. Schutzenberger**

## Schutzenberger (1920-1996)

**è stato un ricercatore e un punto di riferimento per varie generazioni di ricercatori nel campo dei linguaggi formali e l'informatica teorica.**

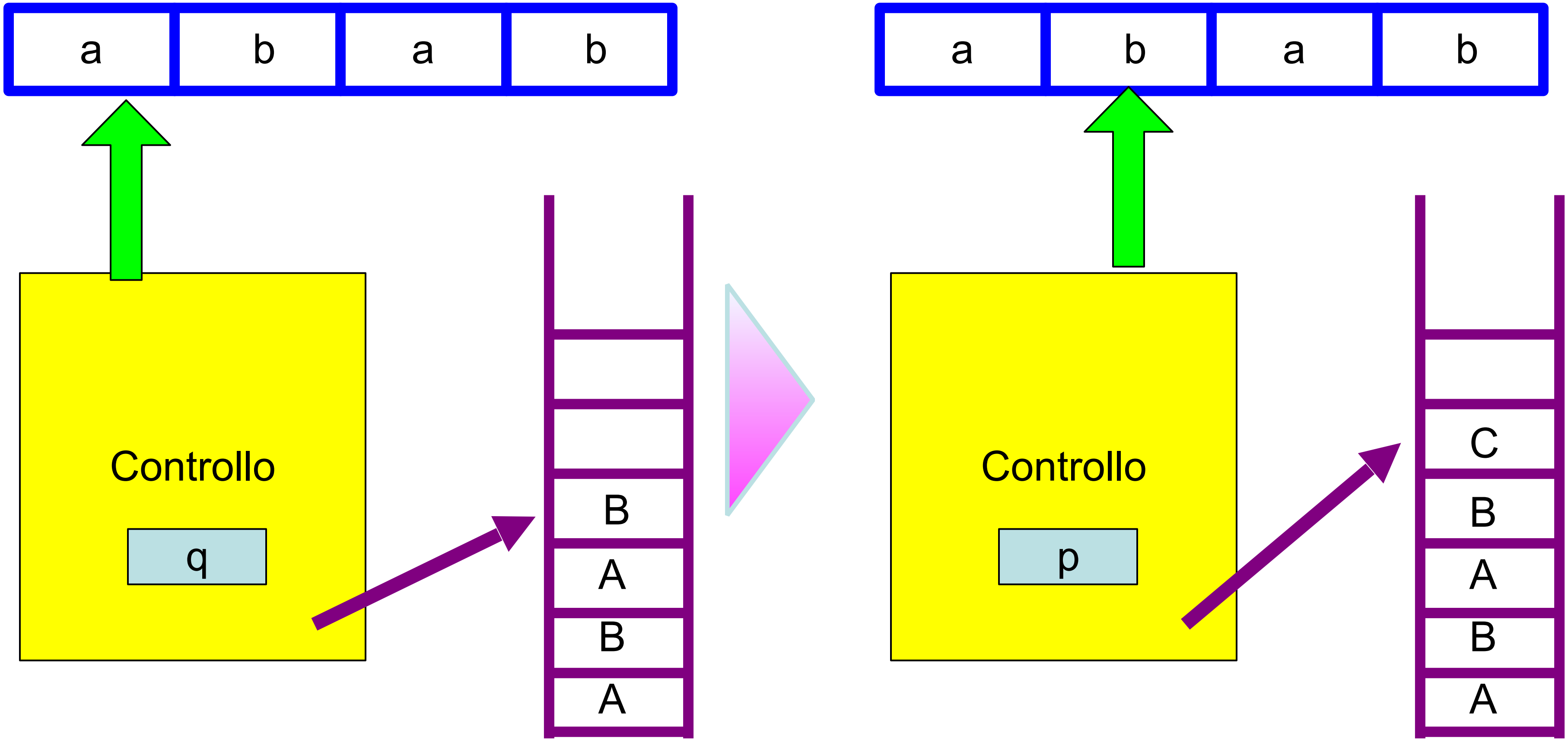
**Ha insegnato alla Facoltà di Scienze dell'Università di Parigi (VI e VII).**

# AUTOMI A PILA: modello  mentale



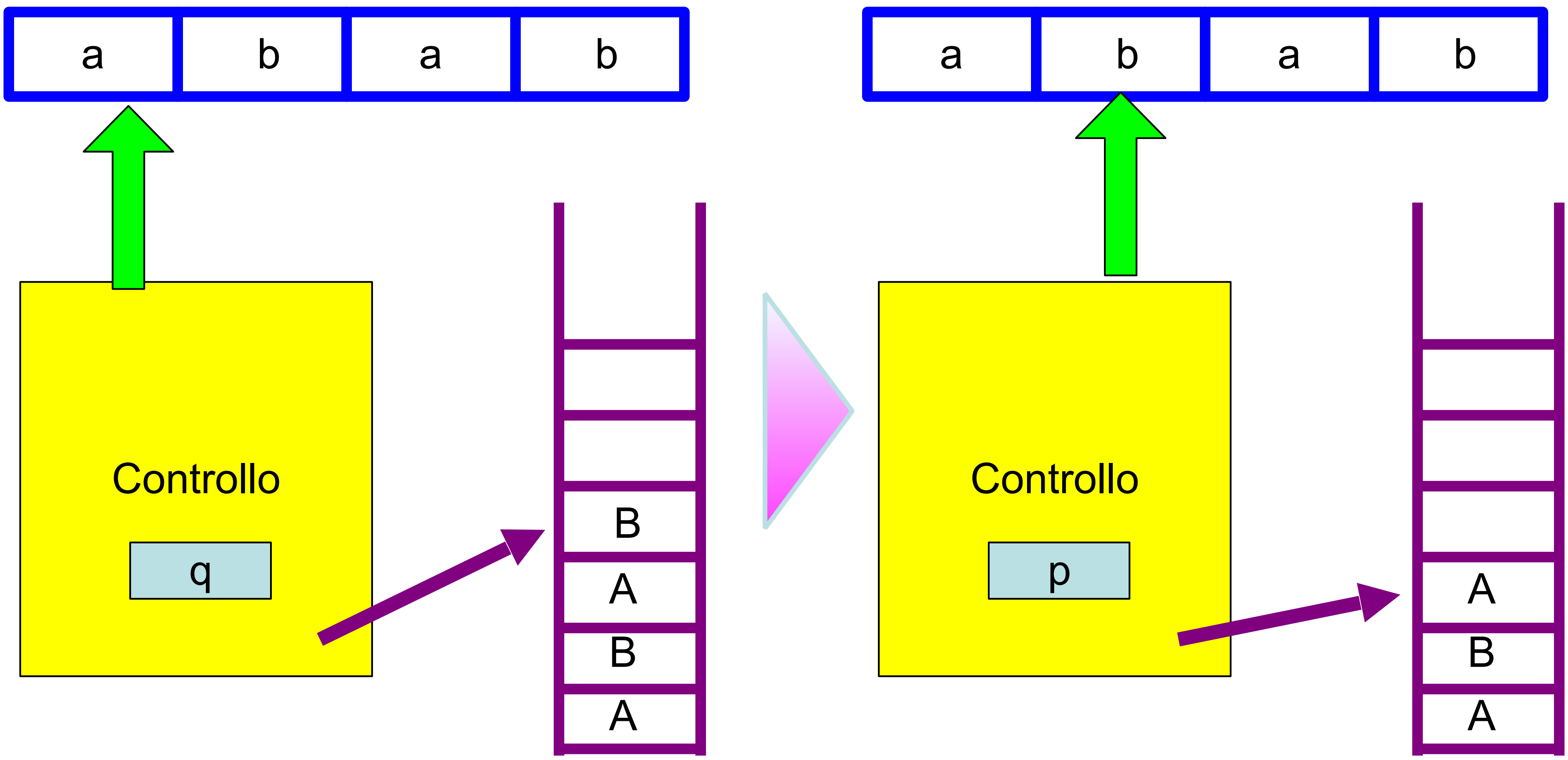

**In generale ora una mossa dipende dallo stato, dal simbolo in lettura sul nastro di input e dal simbolo in lettura in cima alla pila e la sua esecuzione comporta cambio stato, spostamento testina di lettura e sostituzione simbolo in cima alla pila.**

# AUTOMI A PILA: Push




**Se si ignora il simbolo in cima alla pila si esegue una Push: si inserisce in cima alla pila. Questa può anche essere una ε-mossa (sull'input)**

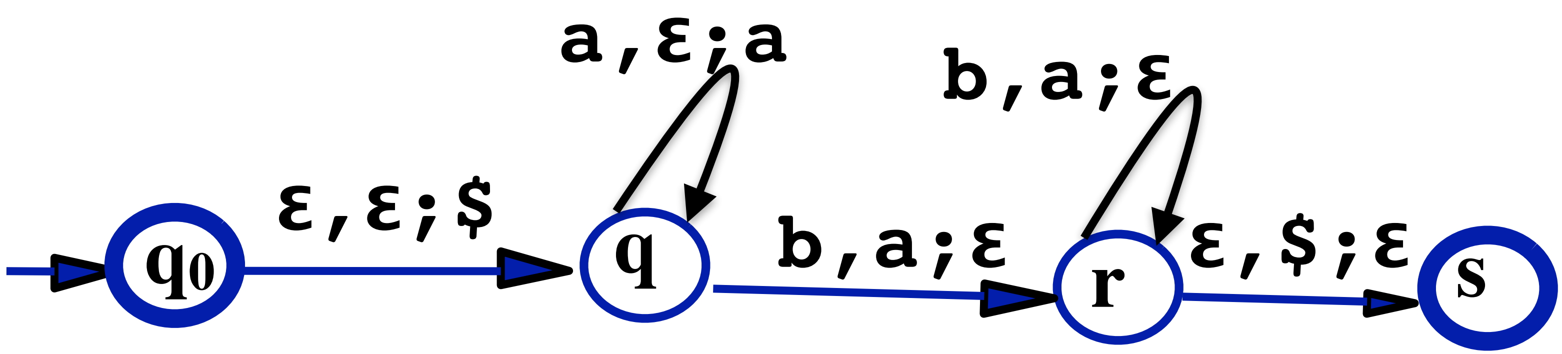
# AUTOMI A PILA: Pop




**Se si cancella il simbolo in cima alla pila si ottiene la Pop, che può anche essere eseguita con una ε-mossa (sull'input)**

# Esempio 1

Un PDA che accetta $L = \{a^n b^n \mid n >= 0\}$

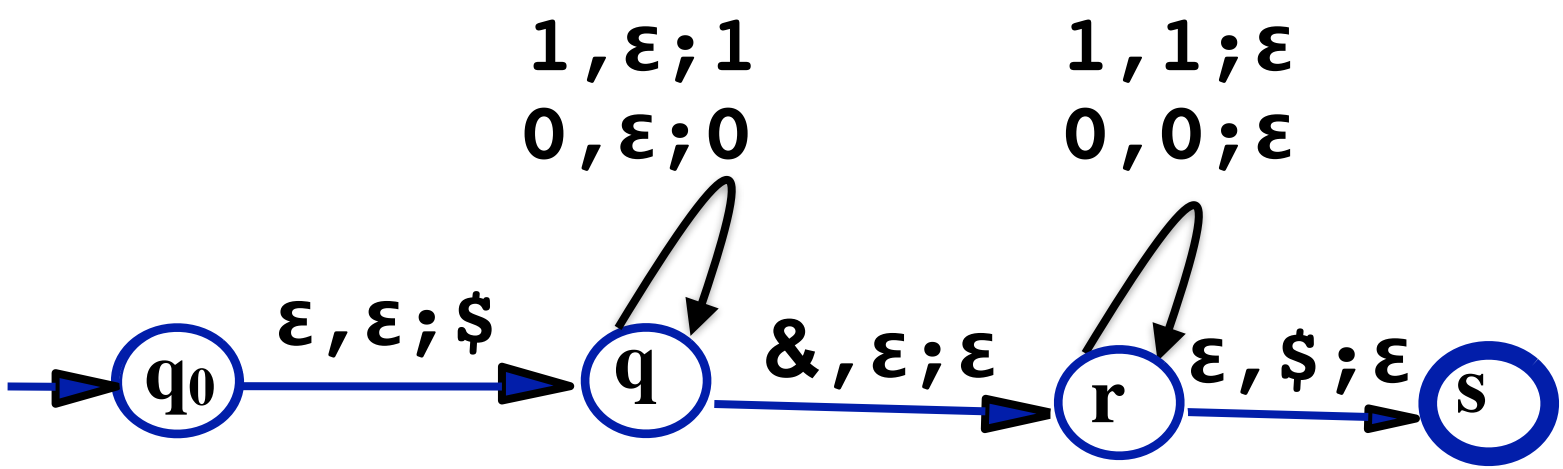



Come nel caso dei DFA una parola è accettata solo se si entra in uno stato di accettazione avendo terminato la lettura dell'input.

Per accettare la parola vuota, si può rendere finale lo stato iniziale.

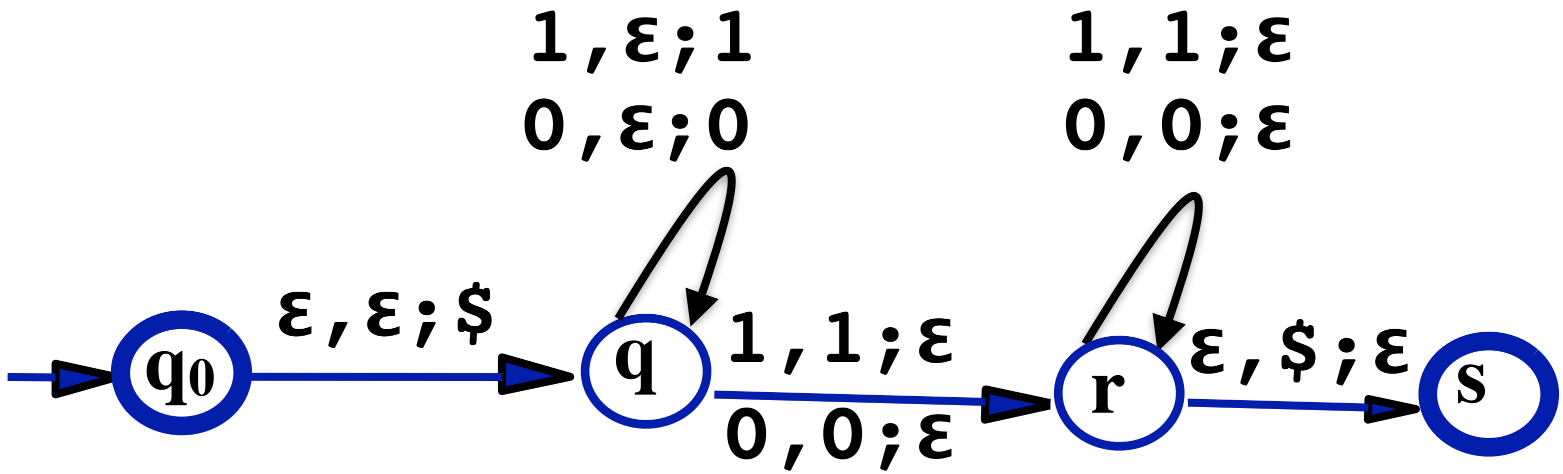
# Esempio 2

**Un PDA che accetta L = {x&*rev*(x) | x è in {0,1}*}**



# Esempio 3

Un PDA che accetta L = {x*rev*(x) | x è in {0,1}*}



# MODELLO FORMALE

**Un automa pushdown non deterministico, in breve PDA da Push Down Automaton, è una 6-pla**

**$M = (Q,\Sigma,\Gamma,\Delta,q_0,F)$ dove**

1. **$Q,\Sigma,\Gamma$ sono insiemi finiti rispettivamente di stati, simboli di input e simboli di pila**

2. **$\Delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \cup \{\varepsilon\} \to P(Q \times (\Gamma \cup \{\varepsilon\}))$ è la funzione di transizione ,**

3. **$q_0 \in Q$ è lo stato iniziale**

4. **$F \subseteq Q$ è l'insieme degli stati finali o di accettazione.**

**NOTA: P(X) è l'insieme dei sottoinsiemi di X.**

# Passi di calcolo

Se $(q,B) \in \Delta(p,a,A)$, con $a \in \Sigma \cup \{\varepsilon\}$, $A$, $B \in \Gamma \cup \{\varepsilon\}$ allora l'automa nello stato $p$ può eseguire uno dei seguenti passi di calcolo,

1. se $a \in \Sigma$, $A$, $B \in \Gamma$ con $a$ in lettura sul nastro di input e $A$ in cima alla pila passa nello stato $q$, rimpiazza $A$ con $B$ in cima alla pila e muove la testina di lettura del nastro di input di una cella verso destra.

2. (PUSH) se $a \in \Sigma$, $A = \varepsilon$ , $B \in \Gamma$ con $a$ in lettura sul nastro di input e indipendentemente dal simbolo in cima alla pila, passa nello stato $q$, impila $B$ in cima alla pila e muove la testina di lettura del nastro di input di una cella verso destra.

3. (POP) se $a \in \Sigma$, $A \in \Gamma$, $B = \varepsilon$ con $a$ in lettura sul nastro di input e $A$ in cima alla pila, passa nello stato $q$, elimina $A$ dalla cima della pila e muove la testina di lettura del nastro di input di una cella verso destra.

Se $a = \varepsilon$ , i tre tipi di mosse avvengono senza che la testina di lettura si sposti.

# Configurazione

Come nel caso dei DFA, il concetto di **configurazione** è utile per descrivere la situazione in cui si trova la macchina complessivamente:

- lo stato,
- l'input ancora da esaminare e
- il contenuto della pila.

Quindi una configurazione è un elemento di

$$Q \times \Sigma^* \times \Gamma^*.$$

La configurazione iniziale è $(q_0,x,\varepsilon)$, se $q_0$ è lo stato iniziale e $x$ è la parola input, infatti inizialmente la pila è vuota.

# Passi di calcolo

**La relazione di transizione sulle configurazioni descrive una mossa dell'automa M, è denotata $\vdash_M$ ed è così definita:**

**per** $p, q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, $x \in \Sigma^*$, $A, B \in \Gamma \cup \{\varepsilon\}$ **e** $\gamma \in \Gamma^*$

$$(p, ax, A\gamma) \vdash_M (q, x, B\gamma) \text{ se } (q, B) \text{ in } \Delta(p, a, A).$$

# Accettazione

**Dato un PDA $M = (Q,\Sigma,\Gamma,\Delta,q_0,F)$**

**il linguaggio accettato da M, L(M), è**

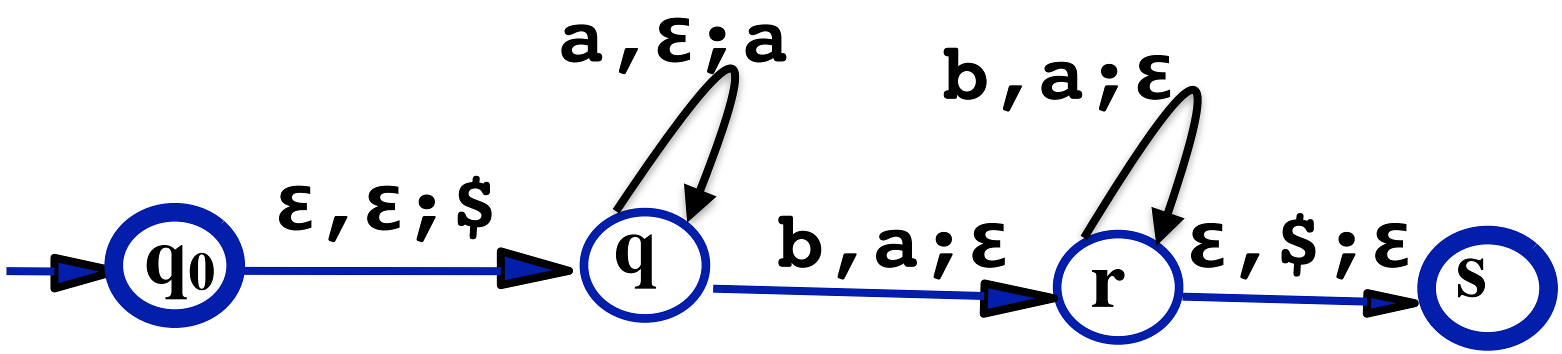
**$L(M) = \{x \mid x \in \Sigma^* \text{ e } (q_0,x,\varepsilon) \vdash_M^* (q,\varepsilon,\gamma), \text{ per } q \in F \text{ e } \gamma \in \Gamma^*\}$,**

**cioè l'insieme delle parole in input per l'automa che determinano una sequenza di configurazioni in cui la prima è quella iniziale e l'ultima, completata la lettura dell'input, è in uno stato finale.**

***In un modello formale alternativo ma che si dimostra equivalente si considerano accettate le parole allo svuotamento della pila, indipendetemente dallo stato di arrivo.***

# Esempio 1

**Un PDA che accetta $L = \{a^n b^n \mid n >= 0\}$**

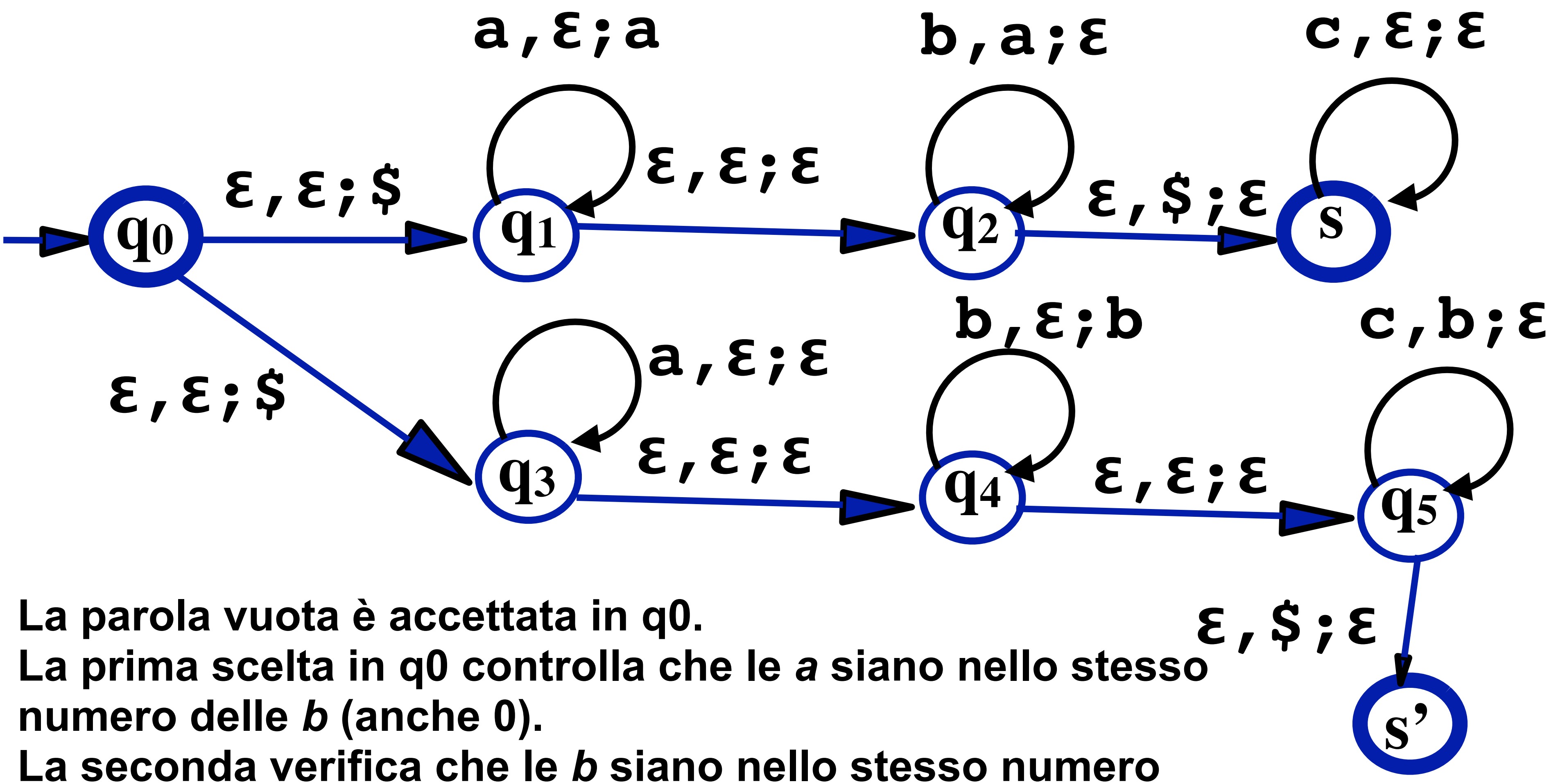



**Come nel caso dei DFA una parola è accettata solo se si entra in uno stato di accettazione avendo terminato la lettura dell'input.**

**Per accettare la parola vuota, si può rendere finale lo stato iniziale.**

# Esercizi

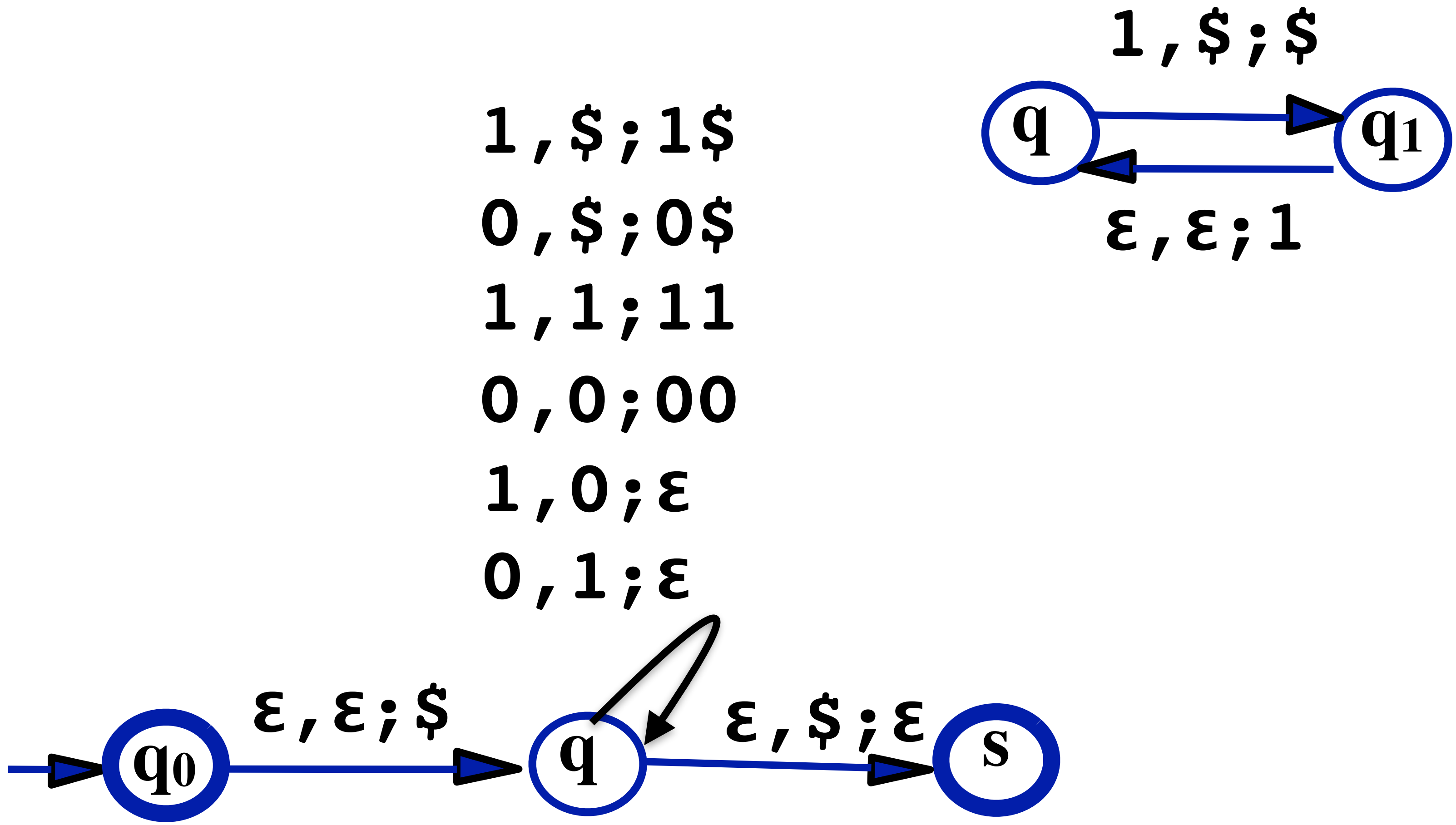
**Esiste un PDA per L={$a^i b^j c^k$ | i=j o j=k, con i,j,k≥0}?**




**La parola vuota è accettata in q0.**
**La prima scelta in q0 controlla che le *a* siano nello stesso numero delle *b* (anche 0).**
**La seconda verifica che le *b* siano nello stesso numero delle *c*,**

# Esempi di linguaggi accettati da un PDA

$\{ x \in \{0,1\}^* \mid n_0(x) = n_1(x)\}$



# Automi Push Down Deterministici

Un PDA $M = (Q,\Sigma,\Gamma,\Delta,q_0,F)$ è deterministico se

- per ogni $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$ e $B \in \Gamma \cup \{\varepsilon\}$

  $|(\Delta(q,a,B))| \leq 1$

- per ogni $q \in Q$ e, $a \in \Sigma$

  $\Delta(q,\varepsilon,B) \neq \varnothing \Rightarrow \Delta(q,a,B) = \varnothing$ per $B \in \Gamma \cup \{\varepsilon\}$

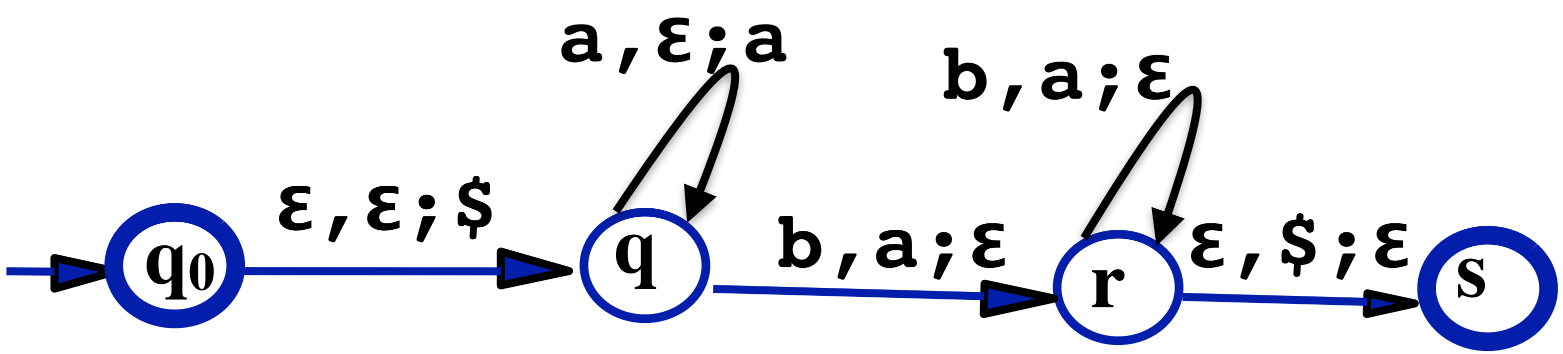
  $\Delta(q,a,\varepsilon) \neq \varnothing \Rightarrow \Delta(q,a,B) = \varnothing$ per $B \in \Gamma$

Così se c'è una ε-mossa eseguibile in un certo stato e con un certo simbolo in cima alla pila allora quella è l'unica mossa eseguibile in quello stato e con quel simbolo in cima alla pila.

Così se c'è una mossa che non dipende dal simbolo in lettura in cima alla pila, quella deve essere l'unica mossa possibile.

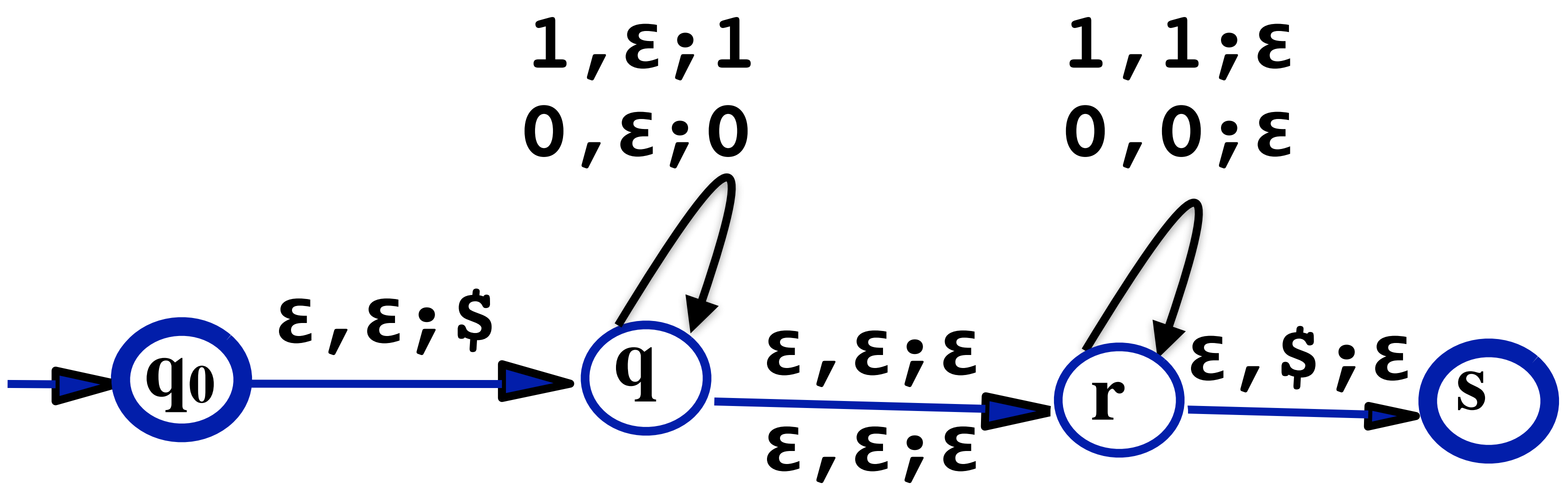
# Esempi

**IL PDA A mostrato tale che**
**$L(A) = \{a^n b^n \mid n \geq 0\}$ è deterministico**



# Esempi

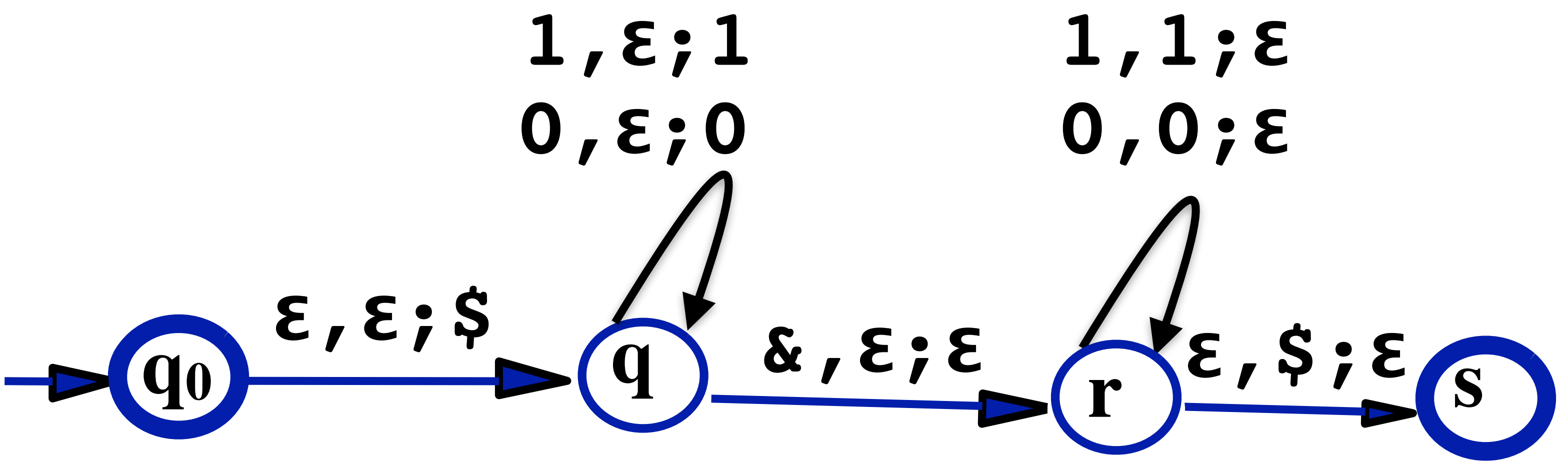
**il PDA A mostrato tale che L(A) =**
**Pal = { xrev(x) ∈{0,1}* | x = rev(x)} è non deterministico.**




**Si può dimostrare che non esiste un DPDA per Pal.**

# Esempi

**il PDA A tale che L(A)**
**= { x&rev(x) $\in$ {0,1}* | x = rev(x)} è deterministico.**



# Determinismo e non determinismo

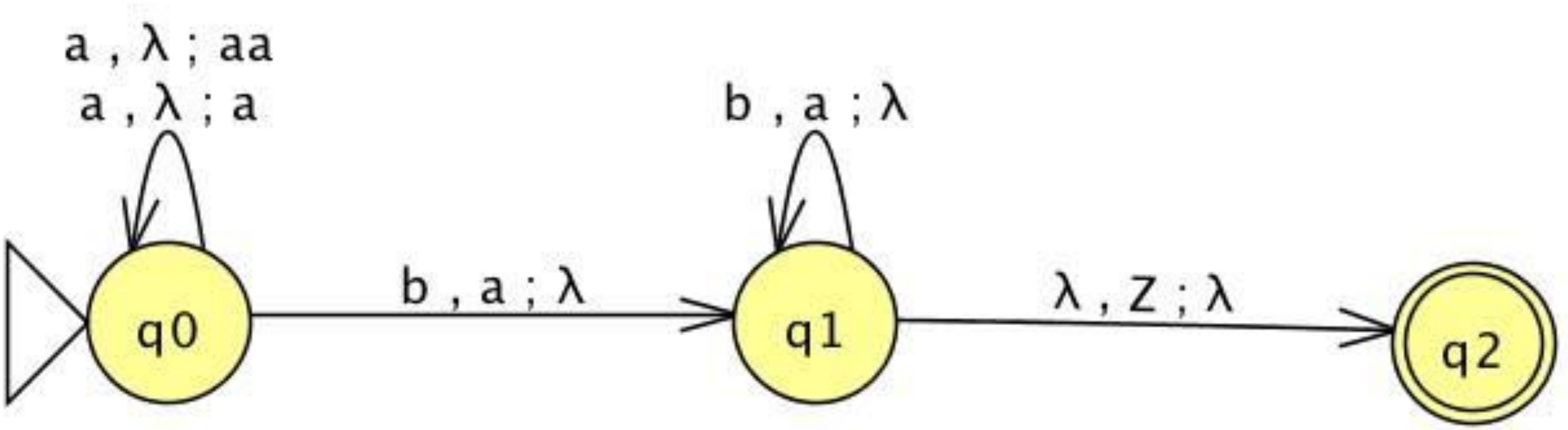
A differenza del caso dei DFA, esistono linguaggi accettati da un PDA che non possono essere accettati da un DPDA.

Quindi $\mathcal{L}$(PDA), la classe dei linguaggi accettati dai PDA, contiene strettamente $\mathcal{L}$(DPDA), la classe dei linguaggi accettati dai PDA deterministici, in breve

$$\mathcal{L}(\text{DPDA}) \subset \mathcal{L}(\text{PDA}).$$

# Esercizi

**Esiste un PDA per $L=\{a^n b^m \mid 0<n\leq m\leq 2n\}$?**




**La soluzione prevede che il simbolo Z di fine pila sia presente nella pila all'inizio del calcolo.**