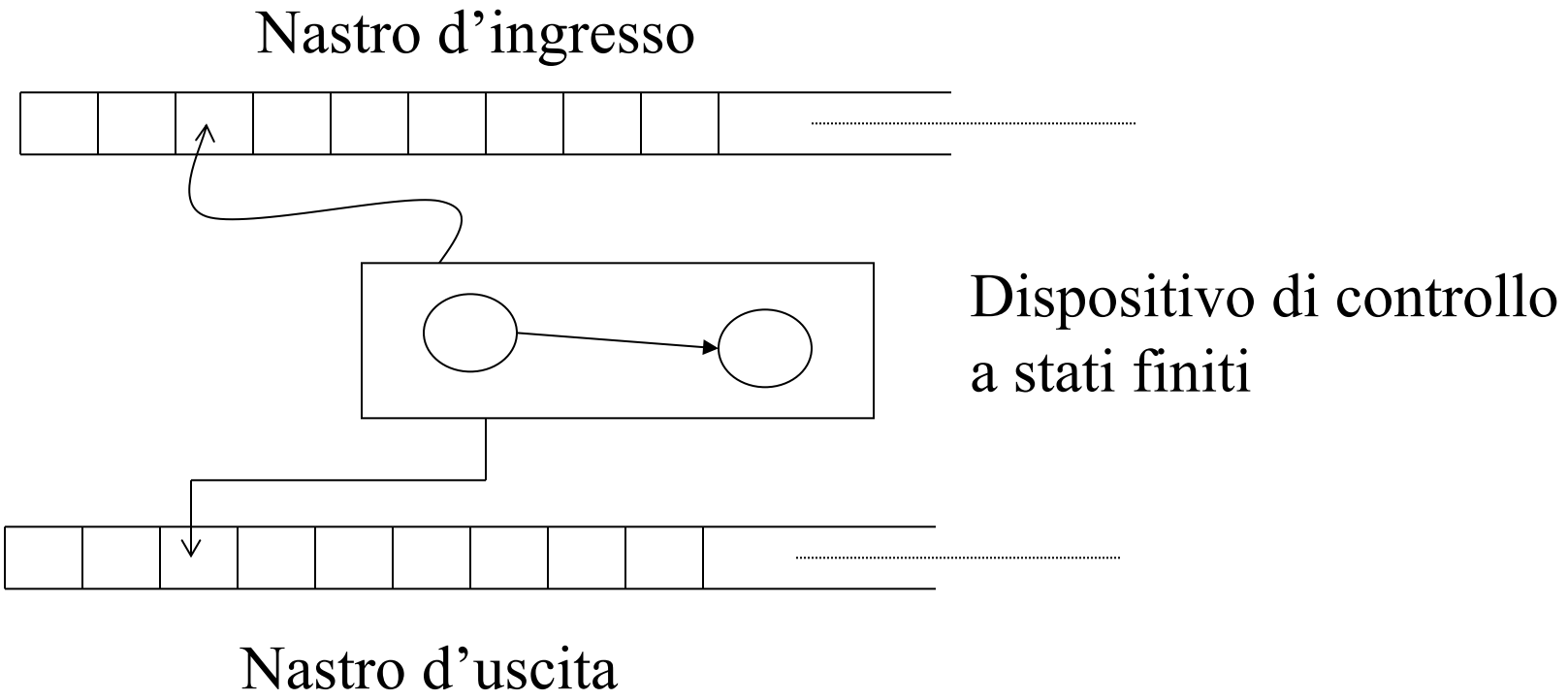
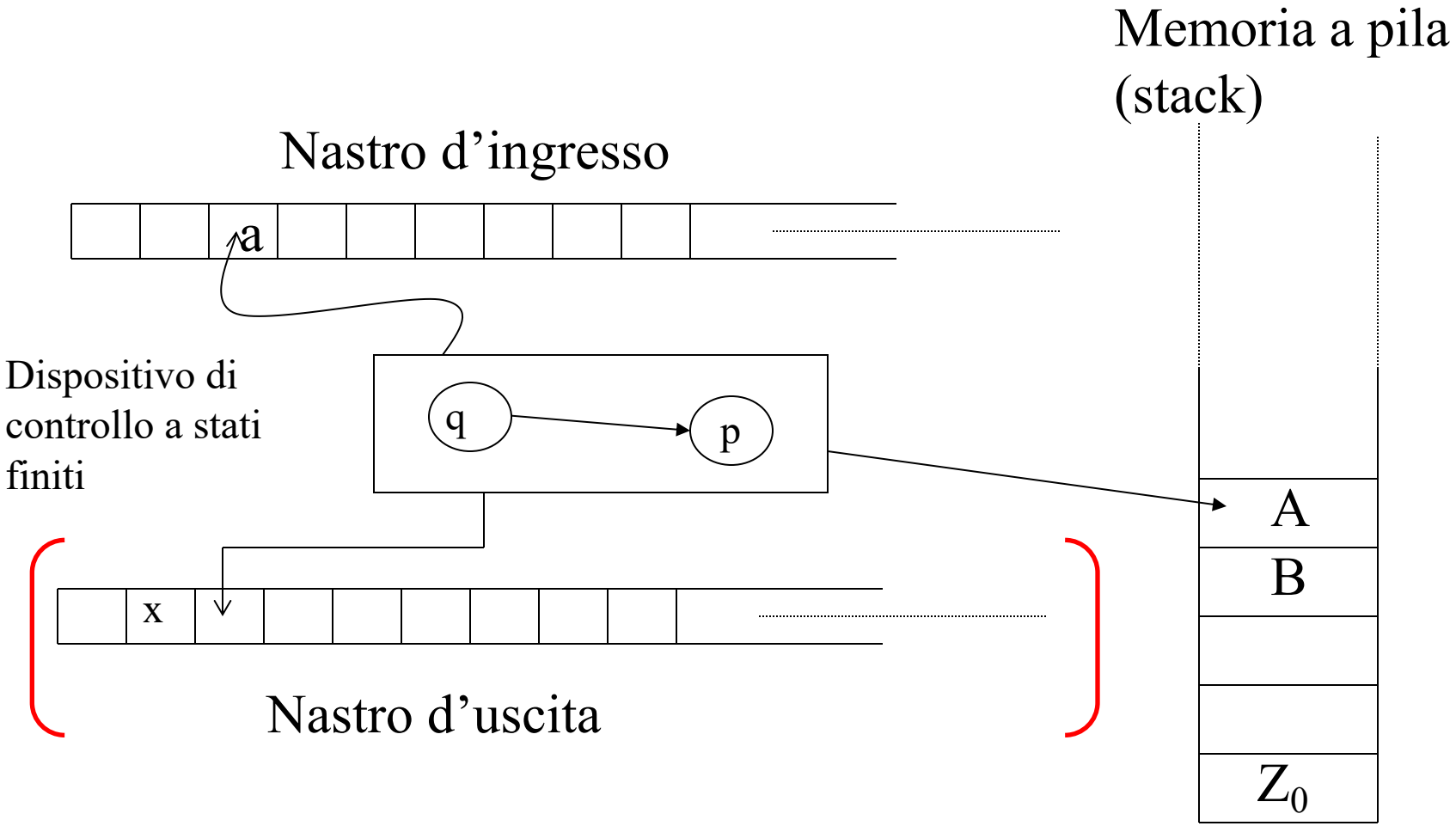
Aumentare il potere degli FSA

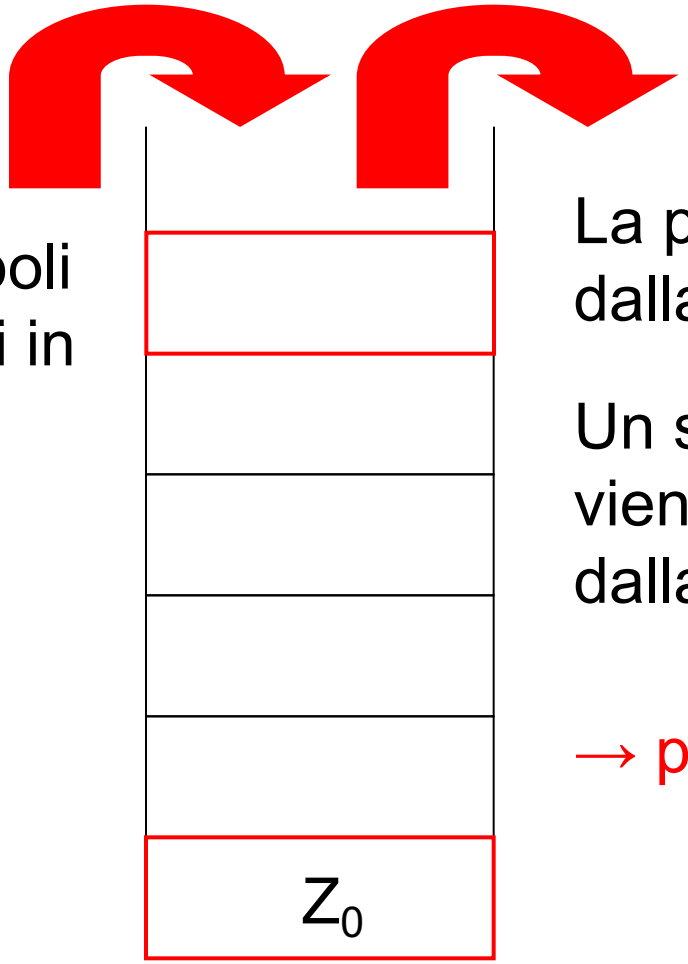
- Punto di vista meccanico



Ora “arricchiamolo”



Cos'è una pila?



I nuovi simboli sono inseriti in cima

La pila viene letta dalla cima

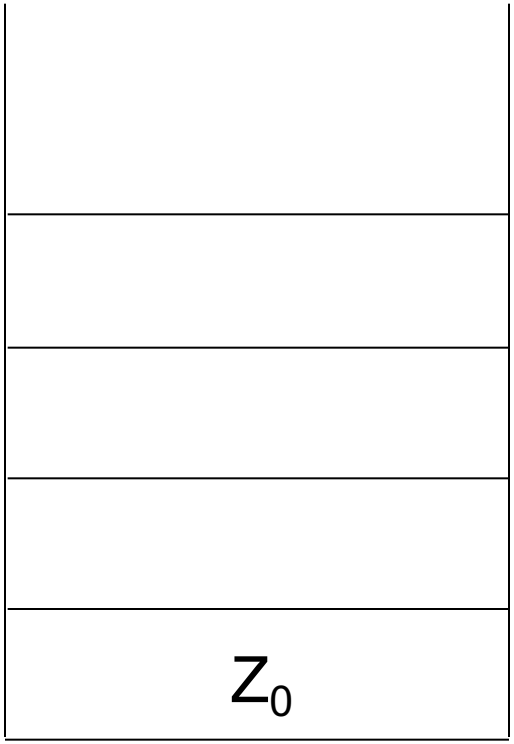
Un simbolo letto viene estratto dalla pila

→ politica LIFO

Esempio

Inserire nel seguente ordine i simboli

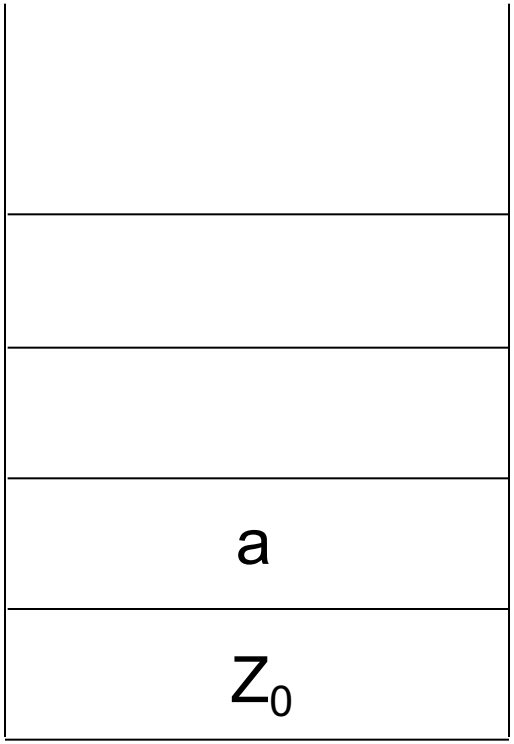
- a
- b
- c



Esempio

Inserire nel seguente ordine i simboli

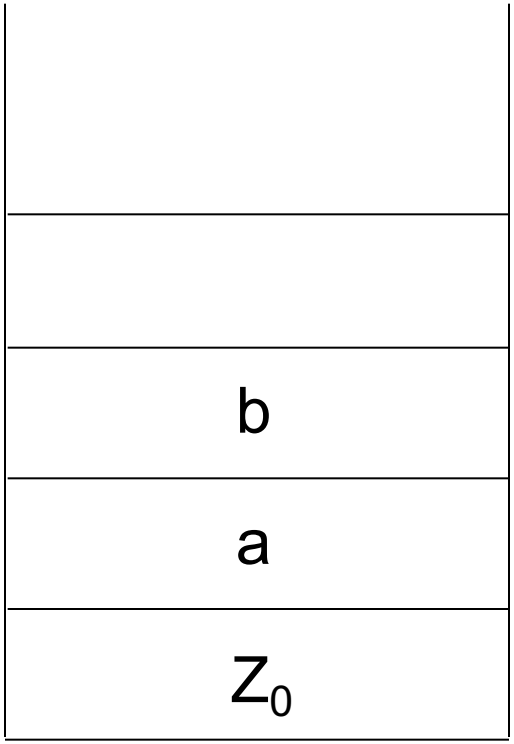
- a
- b
- c



Esempio

Inserire nel seguente ordine i simboli

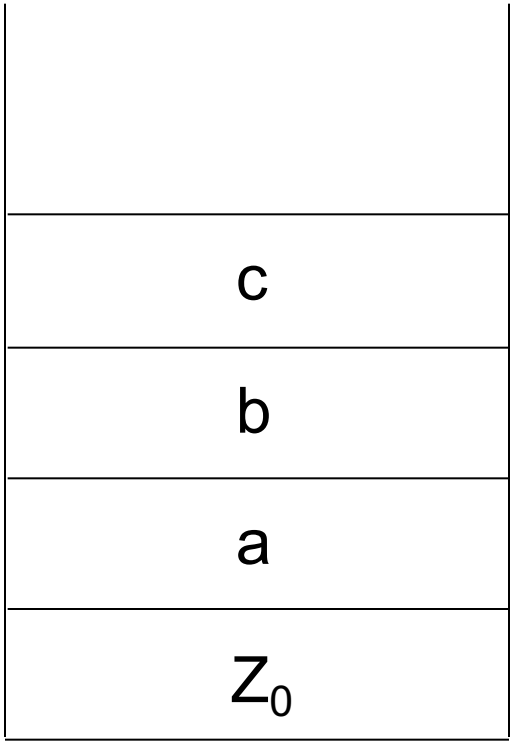
- a
- b
- c



Esempio

Inserire nel seguente ordine i simboli

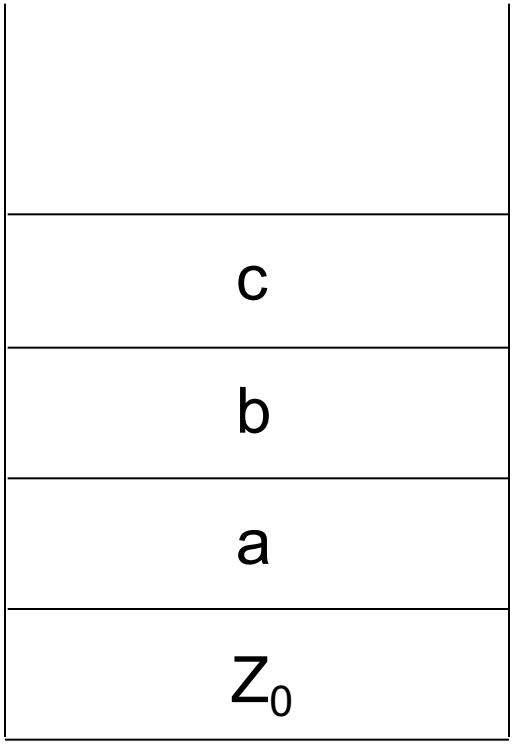
- a
- b
- c



Esempio

Inserire nel seguente ordine i simboli

- a
- b
- c

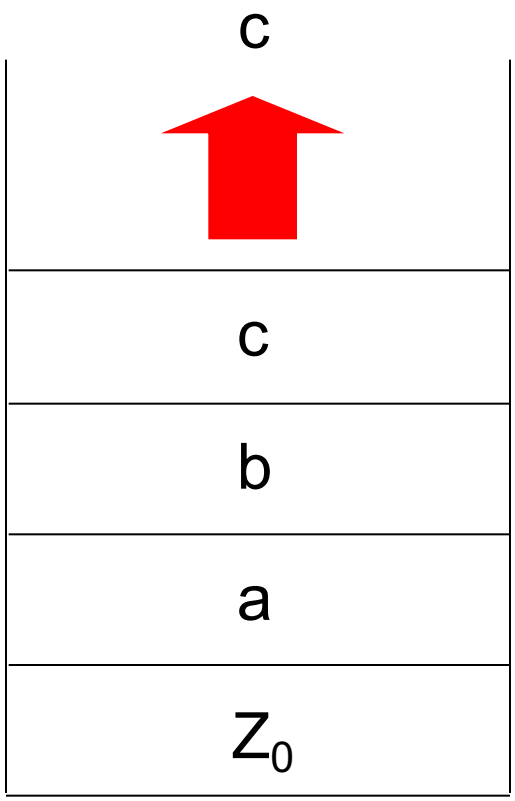


Leggere dalla pila

Esempio

Inserire nel seguente ordine i simboli

- a
- b
- c



Leggere dalla pila

Automi a pila

- Gli automi a stati finiti possono essere arricchiti con una pila

  → Pushdown Automata (PDA)

  – in italiano anche AP
- I PDA differiscono dagli automi a stati finiti in due modi:
  – Possono usare la cima della pila per decidere quale transizione effettuare
  – Possono manipolare la pila durante una transizione

Mosse di un PDA

In base a

- simbolo letto dall'ingresso (ma si può anche non leggere nulla)
- simbolo letto dalla cima della pila
- stato del dispositivo di controllo

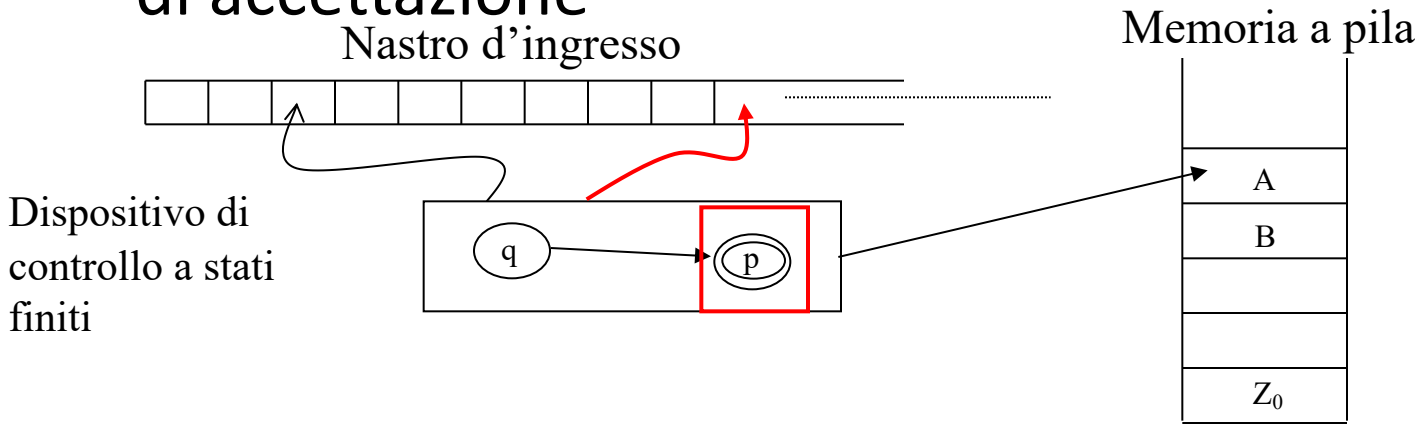
il PDA

- *cambia il proprio stato*
- *sposta in avanti la testina di lettura*
- *sostituisce il simbolo letto dalla pila con una stringa α (eventualmente vuota)*

Accettazione

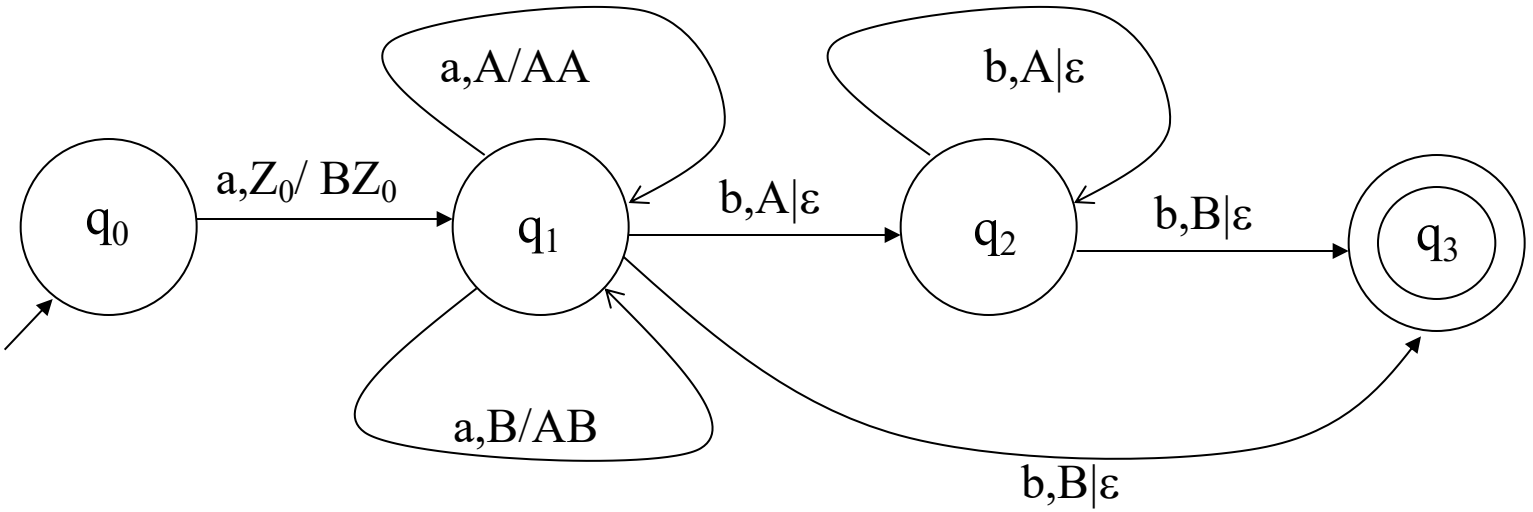
La stringa d'ingresso x è riconosciuta (accettata) se

- il PDA la legge completamente
- quando raggiunge la fine di x si trova in uno stato di accettazione

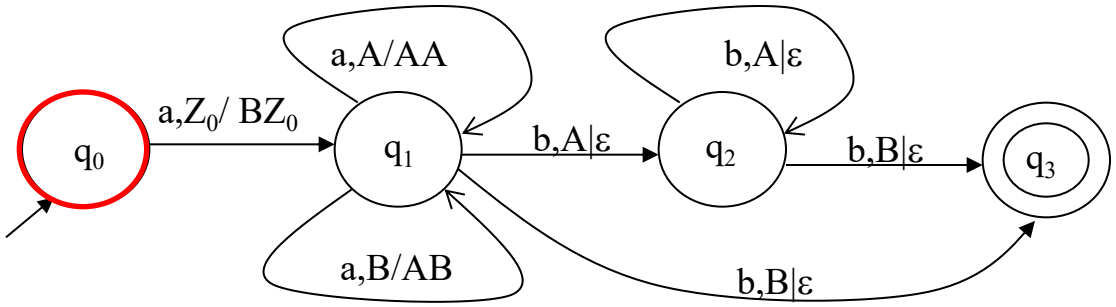


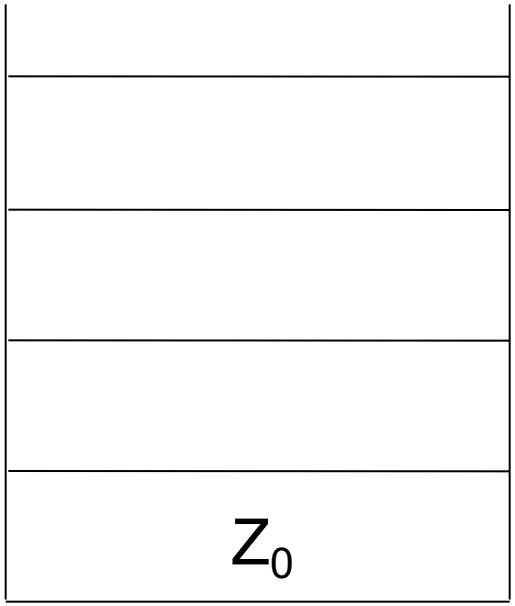
PDA: un primo esempio

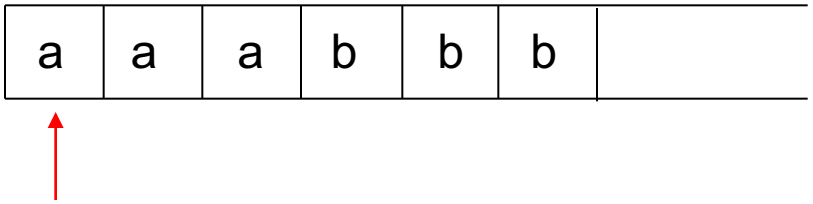
$L=\{a^n b^n \mid n>0\}$



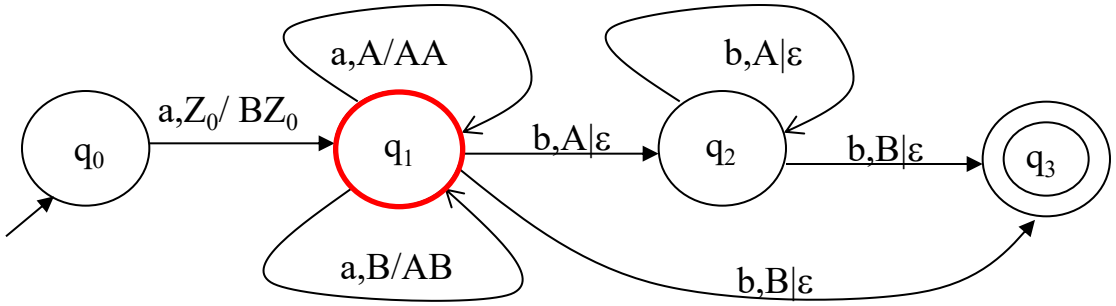
Esempio (2)

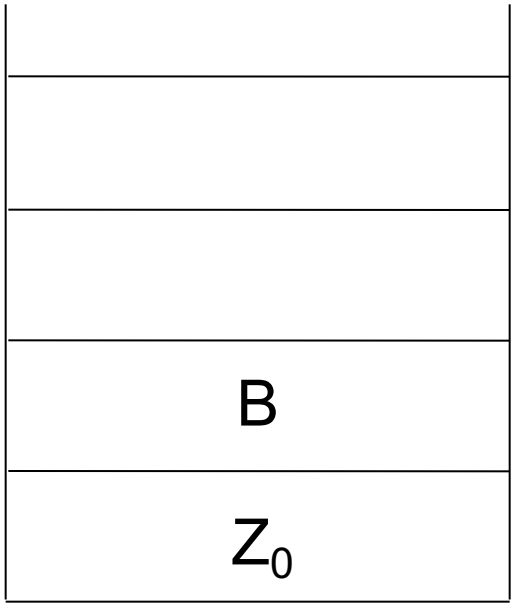


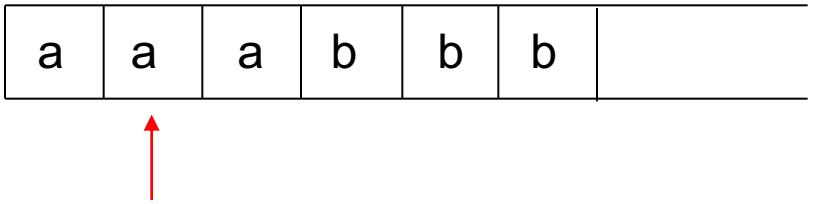




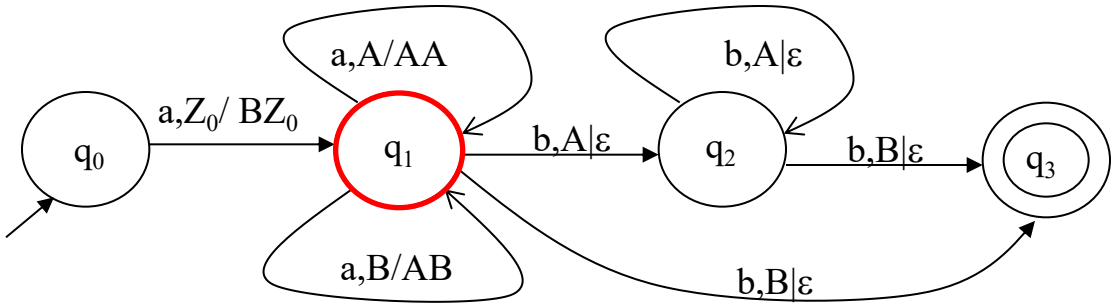
Esempio (2)

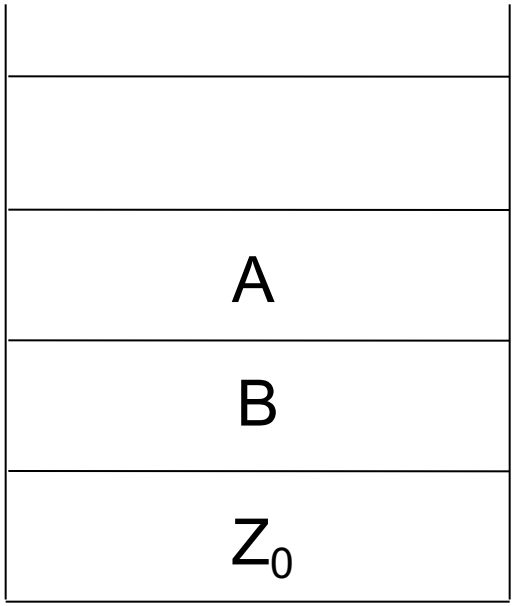


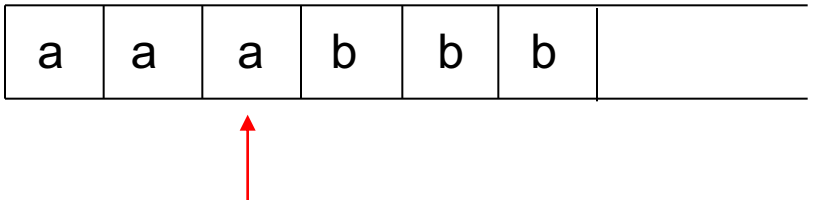




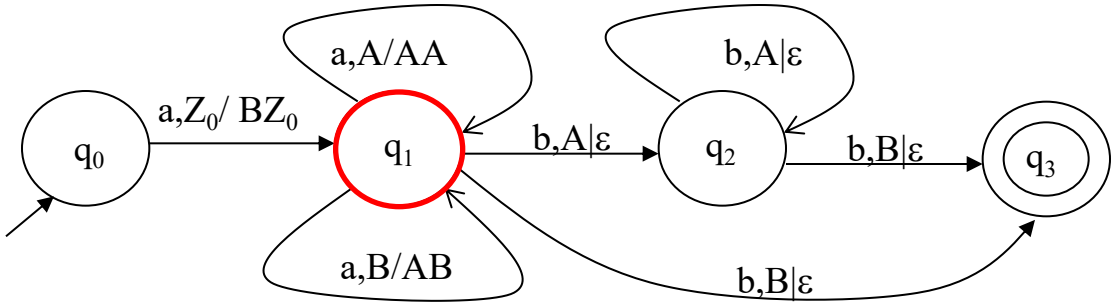
Esempio (2)

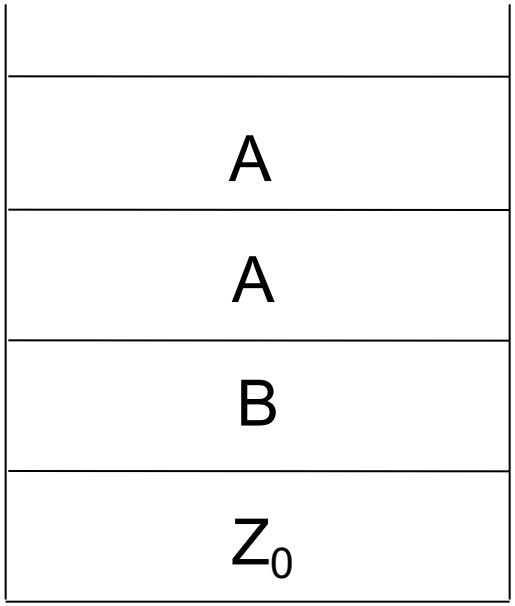


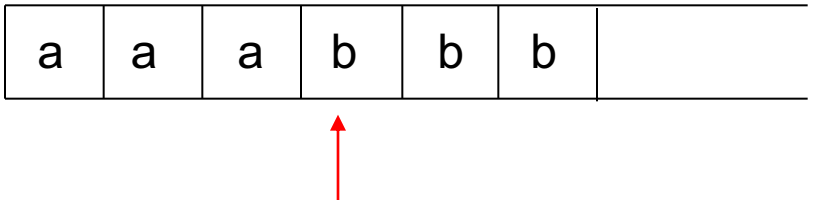




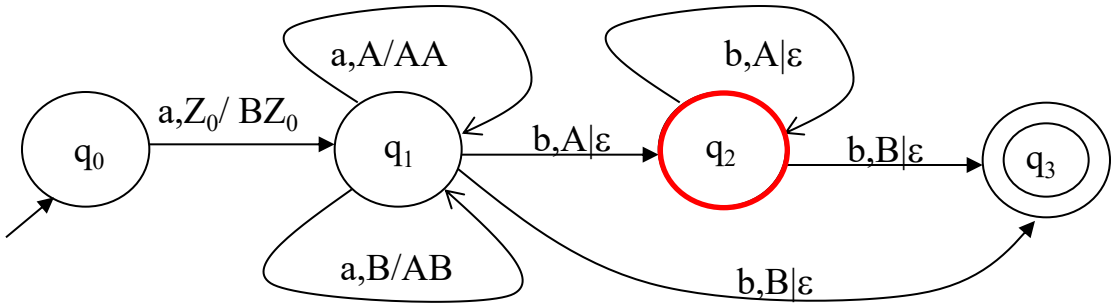
Esempio (2)

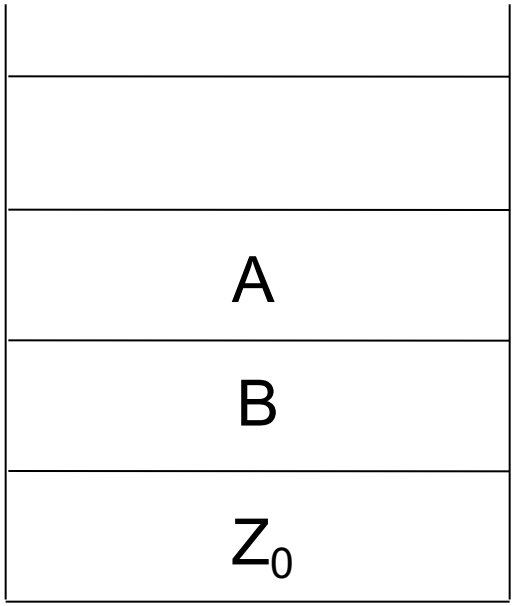


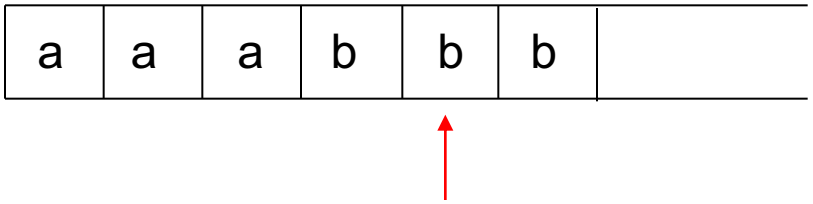




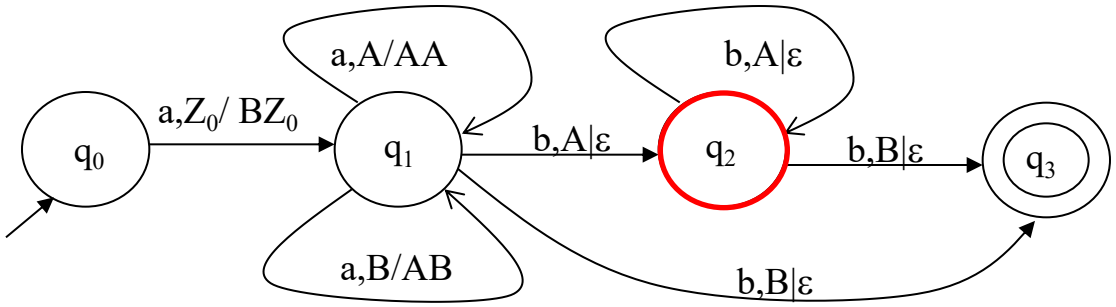
Esempio (2)

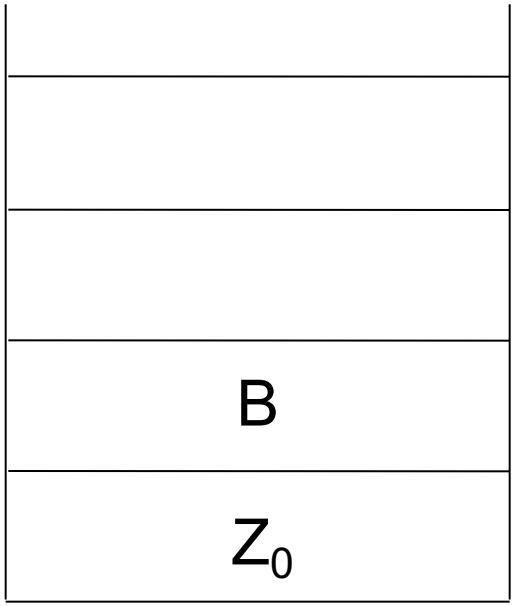


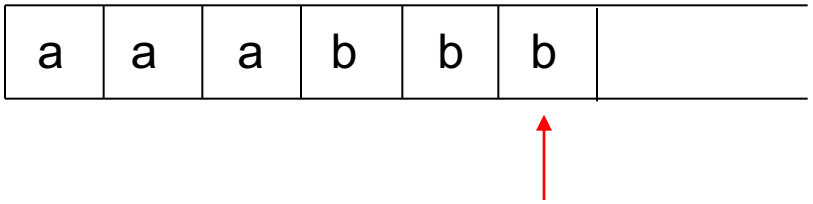




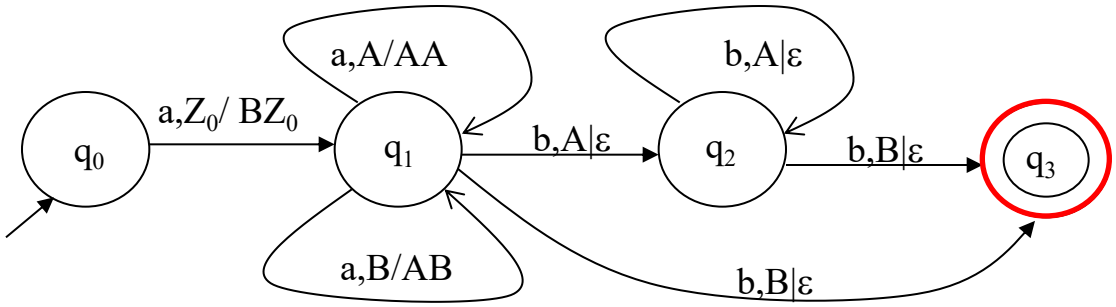
Esempio (2)

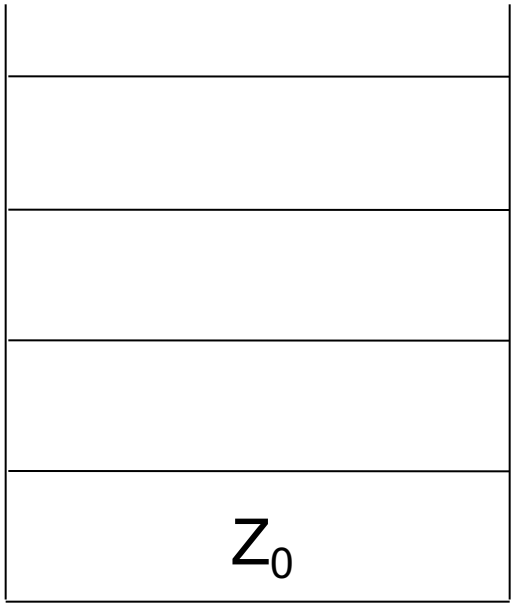


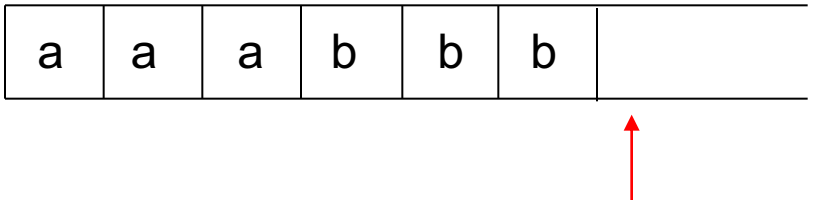




Esempio (2)







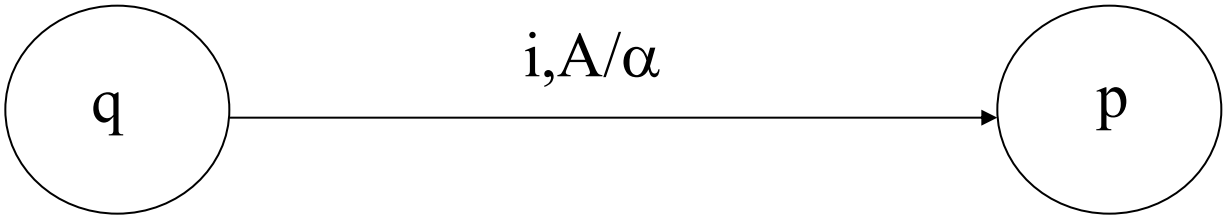
Formalmente

Un PDA è una tupla $<Q, I, \Gamma, \delta, q_0, Z_0, F>$

- Q è un insieme finito di stati
- I è l'alfabeto di ingresso
- Γ è l'alfabeto di pila
- δ è la funzione di transizione
- $q_0 \in Q$ è lo stato iniziale
- $Z_0 \in \Gamma$ è il simbolo iniziale di pila
- $F \subseteq Q$ è l'insieme di stati finali

Funzione di transizione

- δ è la funzione di transizione
- $\delta: Q \times (I \cup \{\varepsilon\}) \times \Gamma \rightarrow Q \times \Gamma^*$
  - $\delta(q, i, A)=<p, \alpha>$
- Notazione grafica:

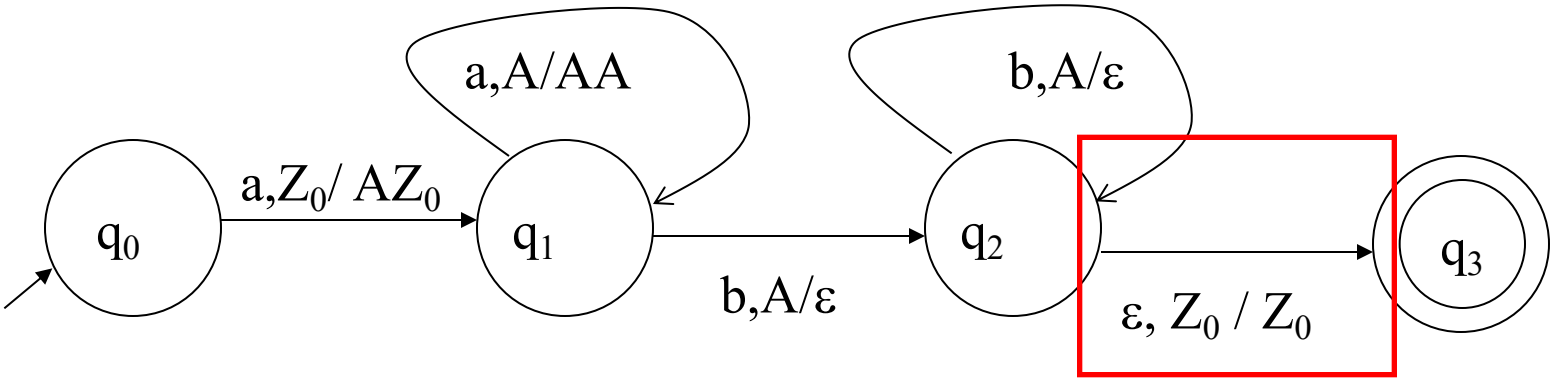


Note

- Q, I, q_0 e F sono definite come negli FSA
- δ è una funzione parziale
- Z_0 è il simbolo iniziale di pila, ma non è essenziale
  - E' utile per semplificare le definizioni
- δ(q, ε, A)=<p, α>?
  - Una "ε-mossa" è una mossa spontanea
  - ε non significa che l'ingresso sia vuoto!

Esempio

Ancora $L=\{a^n b^n \mid n>0\}$



Informalmente

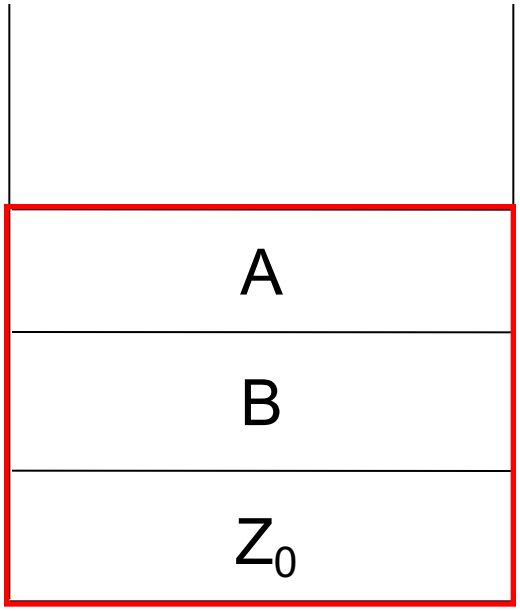
- Una configurazione è una generalizzazione della nozione di stato
- Una configurazione mostra
  - lo stato corrente del dispositivo di controllo
  - la porzione della stringa d'ingresso che parte dalla testina
  - la pila
- E' una fotografia del PDA

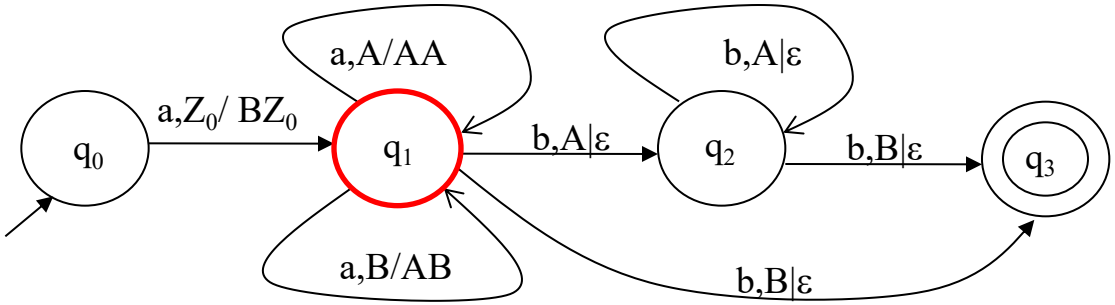
Formalmente

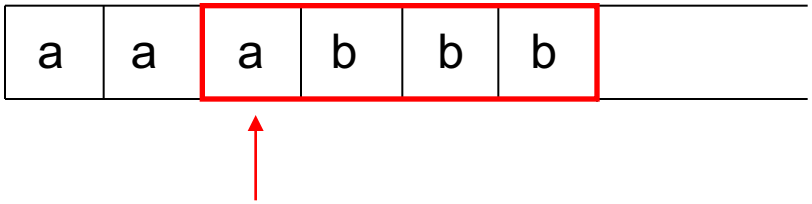
- Una configurazione c è una tripla <q, x, γ>
  - $q \in Q$ è lo stato corrente del dispositivo di controllo
  - $x \in I^*$ è la porzione non letta della stringa d'ingresso
  - $\gamma \in \Gamma^*$ è la stringa di simboli nella pila
- Convenzione per la pila
  - Alto-sinistra, basso-destra
  - Si può anche fare nell'altro modo, basta essere coerenti!

Esempio

- c=<q_1, abbb, ABZ_0>







Transizioni

- Le transizioni tra configurazioni (⊢) dipendono dalla funzione di transizione
  - La funzione di transizione mostra come commutare da una fotografia di un PDA a un'altra
- Ci sono due casi
  - La funzione di transizione è definita per un simbolo d'ingresso
  - La funzione di transizione è definita per una ε-mossa

Transizioni: caso 1

- $\delta(q, i, A) = \langle q', \alpha \rangle$ è definita
- $c=\langle q,x,\gamma \rangle \vdash c'=\langle q', x', \gamma' \rangle$
  - $\gamma=A\beta$
  - $x=iy$

  allora
  - $\gamma'= \alpha\beta$
  - $x'=y$

Transizioni: caso 2

- $\delta(q, \varepsilon, A) = \langle q', \alpha \rangle$ è definita
- $c=\langle q,x,\gamma \rangle \vdash c'=\langle q', x', \gamma' \rangle$
  - $\gamma=A\beta$

  allora
  - $\gamma'= \alpha\beta$
  - $x'=x$

Nota importante

- Una ε-mossa è una mossa spontanea
  - Se $\delta(q,\varepsilon,A)\neq\perp$ e A è il simbolo in cima alla pila, la transizione può sempre essere eseguita
- Se $\delta(q,\varepsilon,A)\neq\perp$, allora $\forall i\in I \quad \delta(q,i,A)=\perp$
  - Se questa proprietà non fosse soddisfatta, entrambe le transizioni sarebbero consentite
  - Non determinismo

Condizione di accettazione

- Sia $\vdash^*$ la chiusura riflessiva e transitiva della relazione $\vdash$

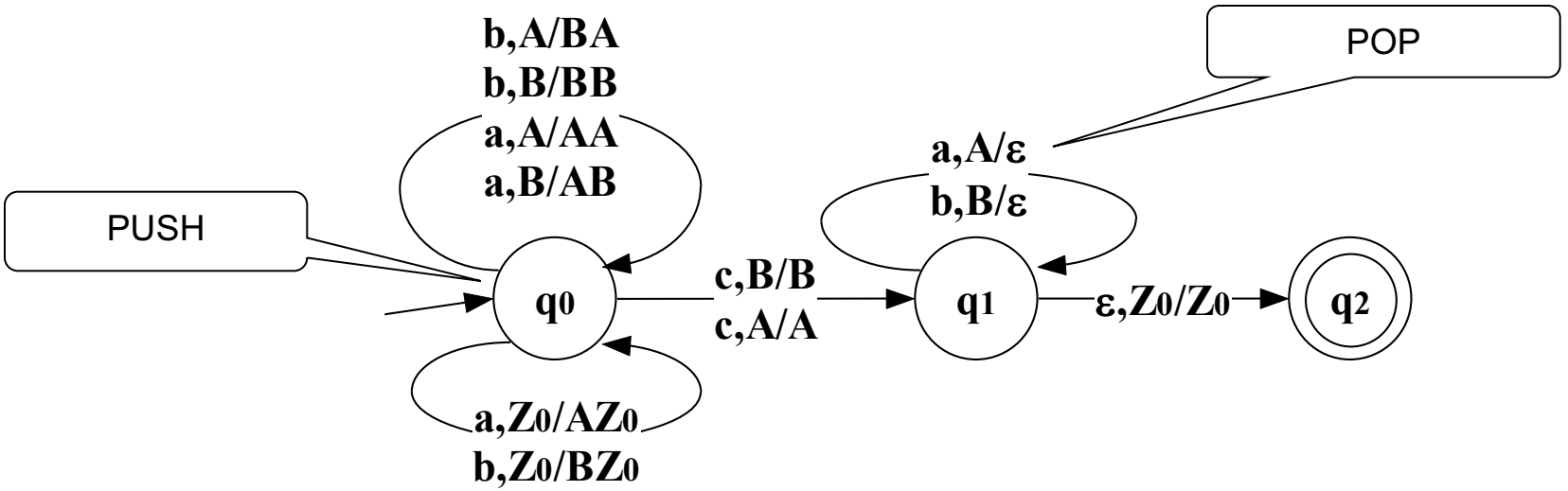
  Condizione di accettazione:

  $\forall x \in I^* \; (x \in L \Leftrightarrow \exists q \exists \gamma \; c_0 = \langle q_0, x, Z_0 \rangle \vdash^* c_F = \langle q, \varepsilon, \gamma \rangle \text{ e } q \in F)$

- Informalmente, una stringa è accettata da un PDA se c'è un cammino coerente con x sul PDA che va dallo stato iniziale a uno stato finale
  - La stringa d'ingresso dev'essere letta interamente

Esempio

- $L = \{wcw^R \mid w \in \{a,b\}^+\}$
  - Occorre usare una politica LIFO per memorizzare w



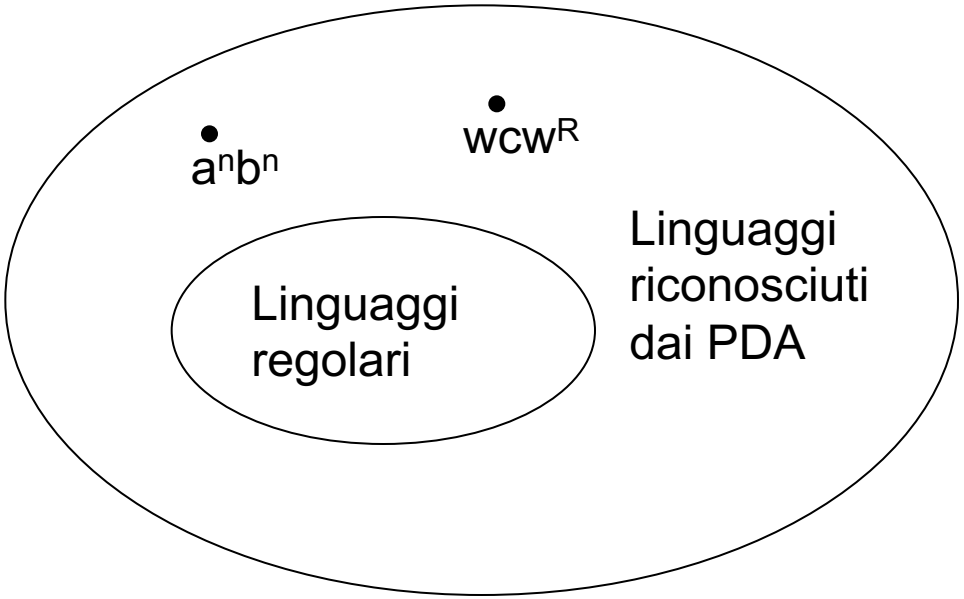
PDA vs FSA (1)

- Sappiamo che $a^n b^n$ non può essere riconosciuto da alcun FSA
  - Pumping Lemma

  ma può essere riconosciuto da un PDA
- Ogni linguaggio regolare può essere riconosciuto da un PDA
  - Dato un FSA $A=<Q,I,\delta,q_0,F>$ è immediato costruire un PDA $A'=<Q',I',\Gamma',\delta',q_0',Z_0',F'>$ tale che $L(A)=L(A')$

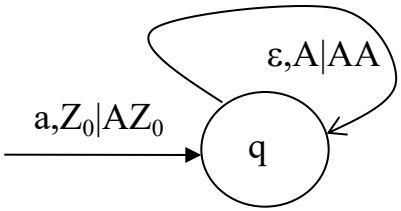
PDA vs FSA (2)

- I PDA sono più espressivi degli FSA



Cicli nei PDA

- A differenza degli FSA, i PDA potrebbero non fermarsi dopo un numero finito di mosse
  - Sono possibili "cicli" di ε-mosse
  - Esempio:



- Però i PDA ciclici non aggiungono potere espressivo alla classe dei PDA
  - Possiamo sbarazzarci di tali PDA
  - Vediamo come…

PDA aciclici

- **$\langle q,x,\alpha\rangle \vdash^*_d \langle q',y,\beta\rangle$** significa che $\langle q,x,\alpha\rangle \vdash^* \langle q',y,\beta\rangle$ e, per $\beta=Z\beta'$, $\delta(q',\varepsilon,Z)=\perp$
  - $\vdash^*_d$ è una sequenza di mosse che ha bisogno di un simbolo per procedere
- Un PDA è <u>aciclico</u> se e solo se

  $\forall x \in I^* \exists q \exists \gamma$ tali che $\langle q_0,x,Z_0\rangle \vdash^*_d \langle q,\varepsilon,\gamma\rangle$
  - legge sempre l'intera stringa d'ingresso e poi si ferma
  - cioè non può ciclare indefinitamente con ε-mosse
- Ogni PDA può essere trasformato in un PDA aciclico equivalente

Complemento

- La classe dei linguaggi riconosciuti dai PDA è chiusa rispetto alla complementazione
- Il complemento può essere costruito
  - eliminando i cicli
  - aggiungendo stati di errore
  - occupandosi di ε-mosse che collegano stati finali e non finali
  - scambiando gli stati finali con i non finali

PDA aciclici

- L'eliminazione dei cicli è essenziale, altrimenti si potrebbe non raggiungere mai la fine della stringa
- Che succede quando ci sono sequenze di ε-mosse che attraversano stati finali e non finali (e l'ingresso è letto interamente)?
  - Potremmo voler imporre l'accettazione solo alla fine di una sequenza (necessariamente finita) di ε-mosse
  - Altrimenti scambiare gli stati finali e non finali per ottenere il complemento non funziona
- Con queste precauzioni siamo sicuri che o il PDA o il suo complemento accetterà la stringa

Unione

- La classe di linguaggi riconosciuti dai PDA non è chiusa rispetto all'unione
- Non esiste alcun PDA che riconosca $\{a^n b^n | n \geq 1\} \cup \{a^n b^{2n} | n \geq 1\}$, ma
  - $\{a^n b^n | n \geq 1\}$ è riconoscibile da PDA
  - $\{a^n b^{2n} | n \geq 1\}$ è riconoscibile da PDA
- Schema intuitivo della dimostrazione
  - Se procedo come per riconoscere $\{a^n b^n | n \geq 1\}$ ma trovo una stringa di $\{a^n b^{2n} | n \geq 1\}$, mi rimangono n b da leggere, ma ho ormai perso l'informazione sul valore di n
  - Se procedo come per riconoscere $\{a^n b^{2n} | n \geq 1\}$ ma trovo una stringa di $\{a^n b^n | n \geq 1\}$ avrò n simboli rimasti nella pila, ma non ho più modo di verificare se sono proprio n

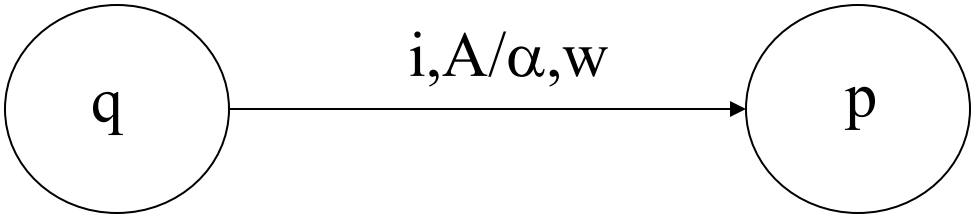
Altre operazioni

- La classe dei linguaggi riconosciuti da PDA non è chiusa rispetto all'intersezione
  - $A \cup B = (A^c \cap B^c)^c$
  - Poiché i linguaggi dei PDA sono chiusi rispetto al complemento, se fossero chiusi rispetto all'intersezione dovrebbero anche essere chiusi rispetto all'unione
- La classe dei linguaggi riconosciuti da PDA non è chiusa rispetto alla differenza
  - $A \cap B = A - B^c$
  - Poiché i linguaggi dei PDA sono chiusi rispetto al complemento, se fossero chiusi rispetto alla differenza dovrebbero anche essere chiusi rispetto all'intersezione

Definizione

- Un trasduttore a pila (pushdown transducer o PDT) è una tupla $<Q, I, \Gamma, \delta, q_0, Z_0, F, O, \eta>$
  - Q è un insieme finito di stati
  - …
  - $F \subseteq Q$ è l'insieme di stati finali
  - O è l'alfabeto di uscita
  - $\eta: Q \times (I \cup \{\varepsilon\}) \times \Gamma \rightarrow O^*$

$\delta(q,i,A)=<p,\alpha>$
$\eta(q,i,A)=w$



Note

- Q, I, Γ, δ, q_0, Z_0 e F sono definiti come nei PDA "accettori"
- η è definita solo dove è definita δ
- La pila può essere necessaria per due ragioni:
  - La richiede il linguaggio da riconoscere
  - La richiede la traduzione

Configurazione

Una configurazione c è una tupla <q, x, γ, z>

- $q \in Q$ è lo stato corrente del dispositivo di controllo
- $x \in I^*$ è la porzione non letta della stringa d'ingresso
- $\gamma \in \Gamma^*$ è la stringa di simboli nella pila
- z è la stringa già scritta sul nastro di uscita

Transizioni

- $c=\langle q, x, \gamma, z\rangle \vdash c'=\langle q', x', \gamma', z'\rangle$
  - Caso 1: $\delta(q, i, A) = \langle q', \alpha\rangle$ è definita e $\eta(q, i, A)=w$
    - $\gamma=A\beta$
    - $x=iy$

    allora
    - $\gamma'= \alpha\beta$
    - $x'=y$
    - $z'=zw$
  - Caso 2: $\delta(q, \varepsilon, A) = \langle q', \alpha\rangle$ è definita e $\eta(q, \varepsilon, A)=w$
    - $\gamma=A\beta$
    - $x=y$

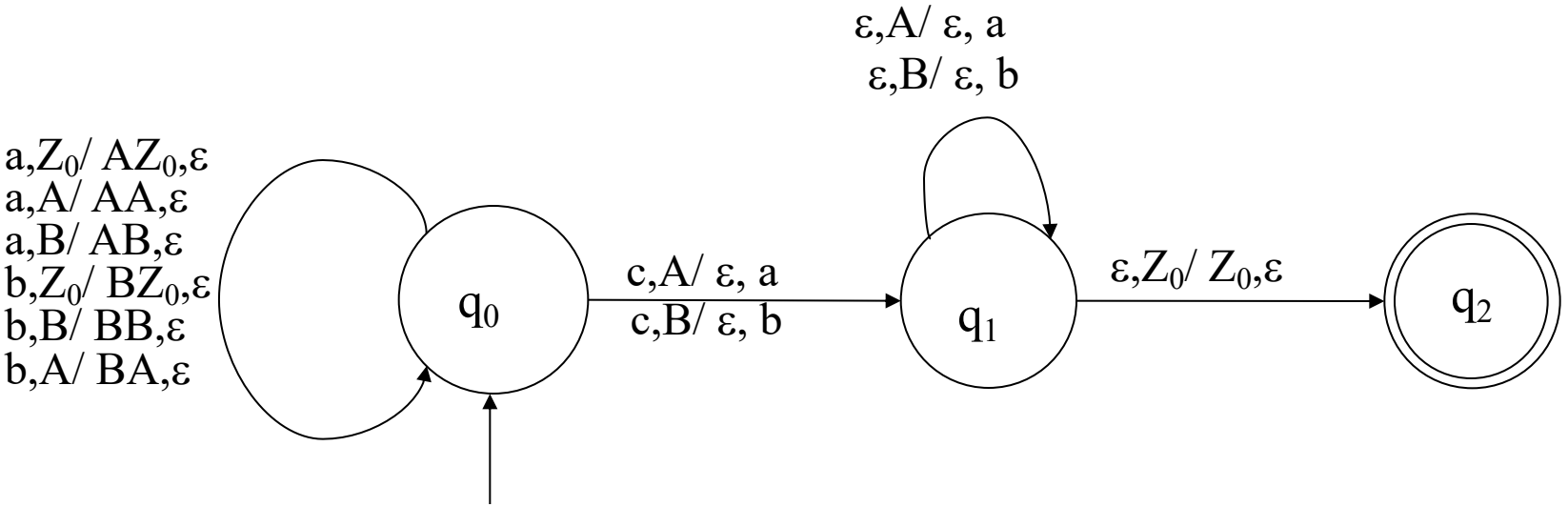
    allora
    - $\gamma'= \alpha\beta$
    - $x'=y$
    - $z'=zw$

Condizione di accettazione/traduzione

- $\forall x \in I^* \; \forall z \in O^* (x \in L \wedge z = \tau(x) \Leftrightarrow \exists q \exists \gamma$ $c_0 = \langle q_0, x, Z_0, \varepsilon \rangle \vdash^* c_F = \langle q, \varepsilon, \gamma, z \rangle \wedge q \in F$ $\wedge \neg \exists c \neq c_F$ tale che $c_F \vdash c$)
- Notare che la traduzione di x è definita solo se la stringa x è accettata
- L'ultima condizione serve a gestire casi di ε-mosse a fine stringa che compiono scritture
  - Quale problema potrebbe manifestarsi?

Esempio

$L=\{wc \mid w \in \{a,b\}^+\}$ e $\tau(wc)=w^R$

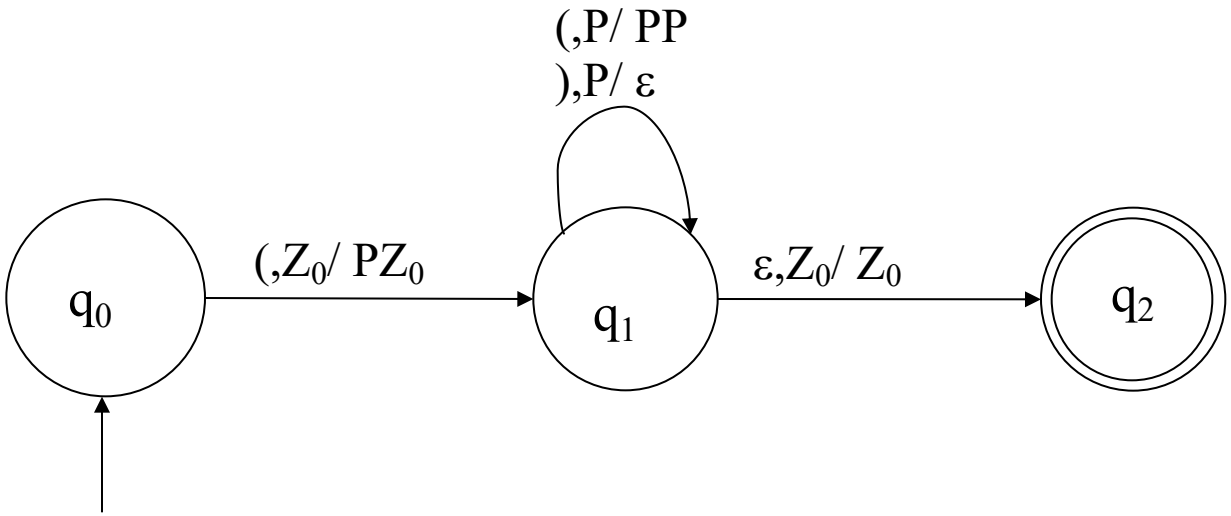


PDA e compilatori

- I PDA sono nel cuore dei compilatori
- La memoria a pila ha una politica LIFO
- La politica LIFO è adeguata per analizzare strutture sintattiche nidificate
  - Espressioni aritmetiche
  - Istruzioni composte
  - Record di attivazione
  - …

Esempio

- PDA per riconoscere stringhe ben parentetizzate
  - (()(()())) OK
  -)())()))) NO



  - Per casa: provare a costruire un PDA che riconosca il complemento di questo linguaggio