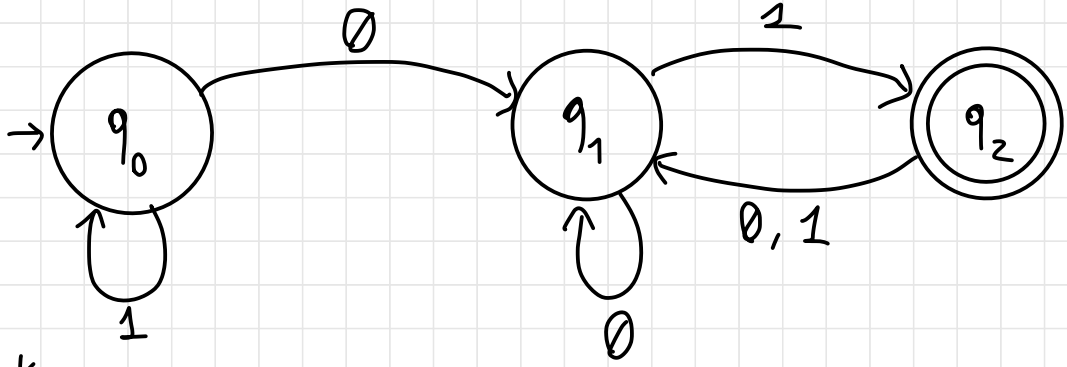
API - esercitazione 2

10/03/2021

$L = (0|1)^* \, 0 \, 1^{2K+1}$



$L_0 = 1^*$, $L_1 = (0|1)^* \, 0 \, 1^{2K}$, $L_2 = L = (0|1)^* \, 0 \, 1^{2K+1}$

Vogliamo dimostrare che:

$\forall \, \omega \in L_0 : \delta^*(q_0, \omega) = q_0$ ✓

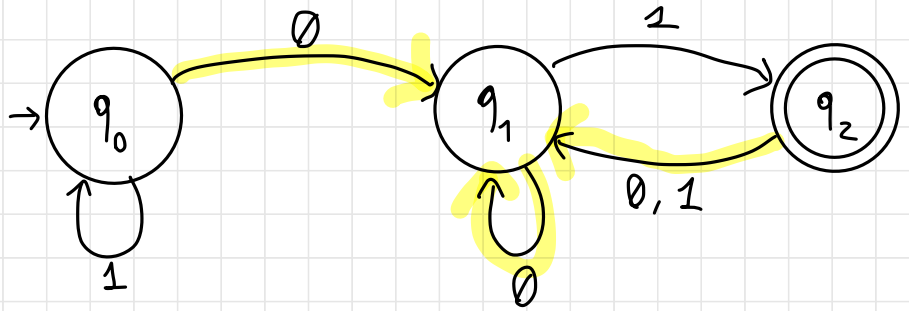
$\forall \, \omega \in L_1 : \delta^*(q_0, \omega) = q_1$

$\forall \, \omega \in \underbrace{L_2}_{\| \; L} : \delta^*(q_0, \omega) = q_2$

$w \in L_1$: $\delta^*(q_0, w) = \delta(\delta^*(q_0, w'), i)$ $L_1 = (0|1)^*01^{2k}$

$i = \emptyset$:

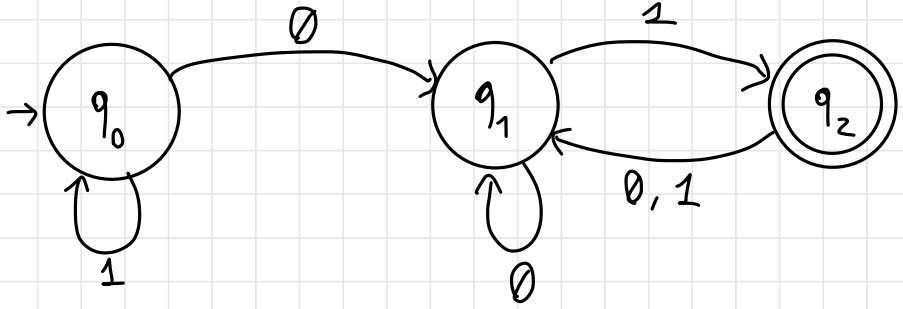
$$\delta(\delta^*(q_0, w'), \emptyset) = q_1 \quad \checkmark$$



$i = 1$: $\delta(\delta^*(q_0, x'), 1)$ $\quad x' \in L_2$

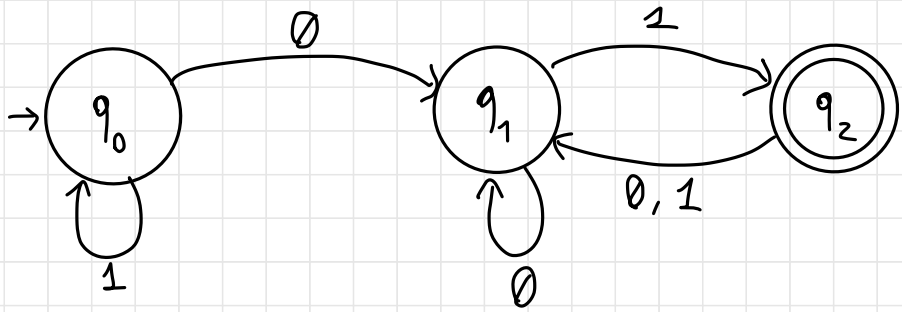
$= \uparrow$ $\delta(q_2, 1) = q_1$ ✓

Hp. induttiva



$w \in L_2$: $\delta(\delta^*(q_0, w'), i) = \delta(\delta^*(q_0, w'), 1)$

$w' \in L_1$ $\rightarrow$ $\delta(q_1, 1) = q_2$



PDA

FSA non riescono a contare un n° arbitrario

$$L = a^n \quad , \quad n \geq 1$$

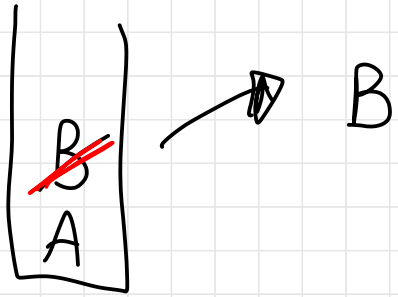
$$\| \quad a^+$$

$$a^n b^n \quad , \quad n \geq 1$$

Idea: aggiungo una memoria

Pila : LIFO

last-in-first-out



Inserimento in pila: push

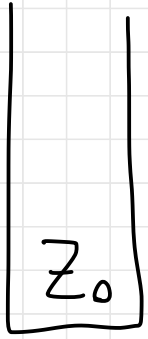
lettura dalla pila: pop

Pila è una mem. distruttiva

$$A = \langle Q, I, \Gamma, \delta, q_0, F, Z_0 \rangle$$

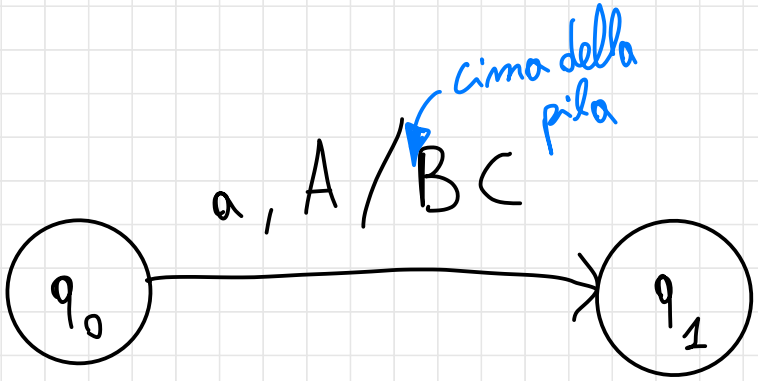
alfabeto di pila

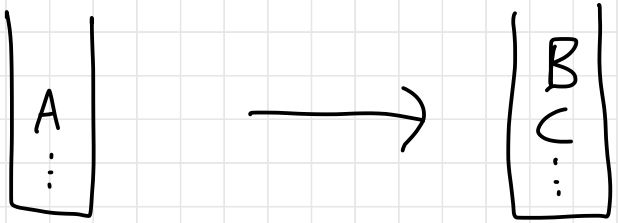
$Z_0 \in \Gamma$ simbolo di inizio pila



$$\delta : Q \times I \cup \{\varepsilon\} \times \Gamma \rightarrow Q \times \Gamma^*$$

$$\delta : Q \times I \cup \{\varepsilon\} \times \Gamma \rightarrow Q \times \Gamma^*$$



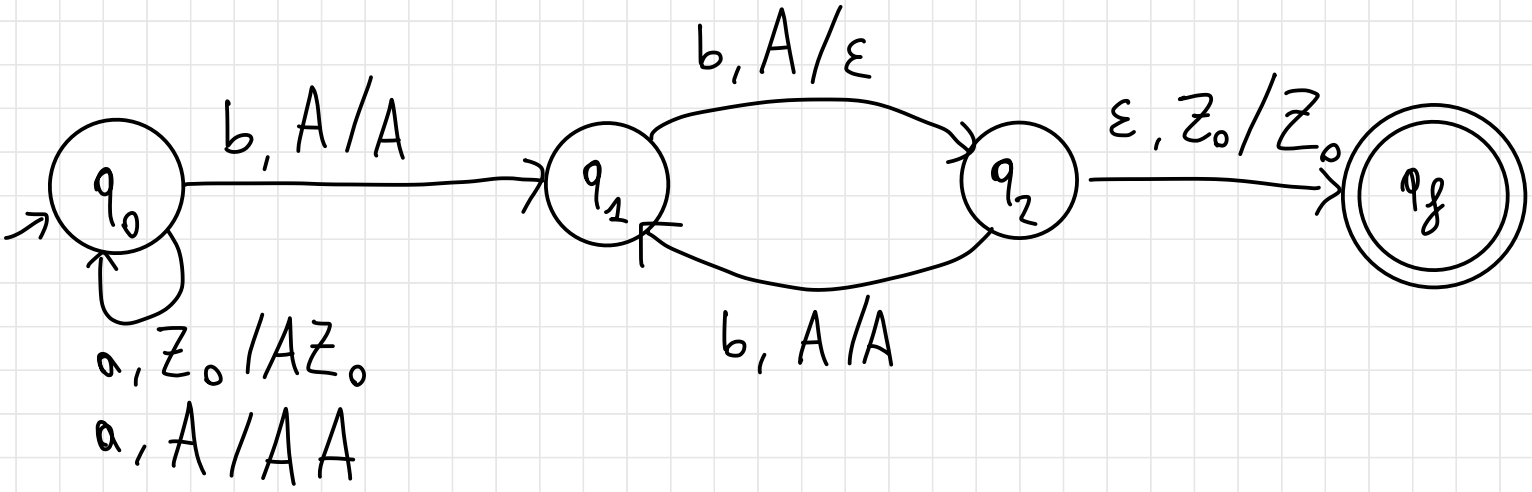


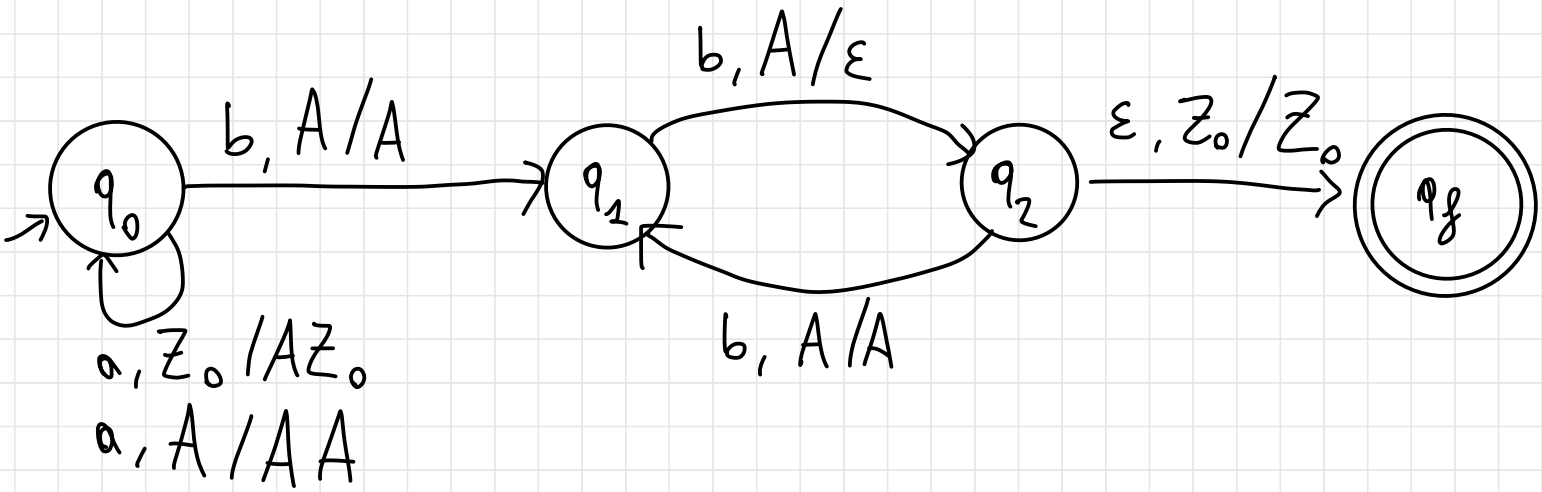
es.: $L = a^n b^{2n}$, $n \geq 1$

Idea: → ad ogni 'a' letta, impilo 'A' → $\begin{vmatrix} A^n \\ Z_0 \end{vmatrix}$

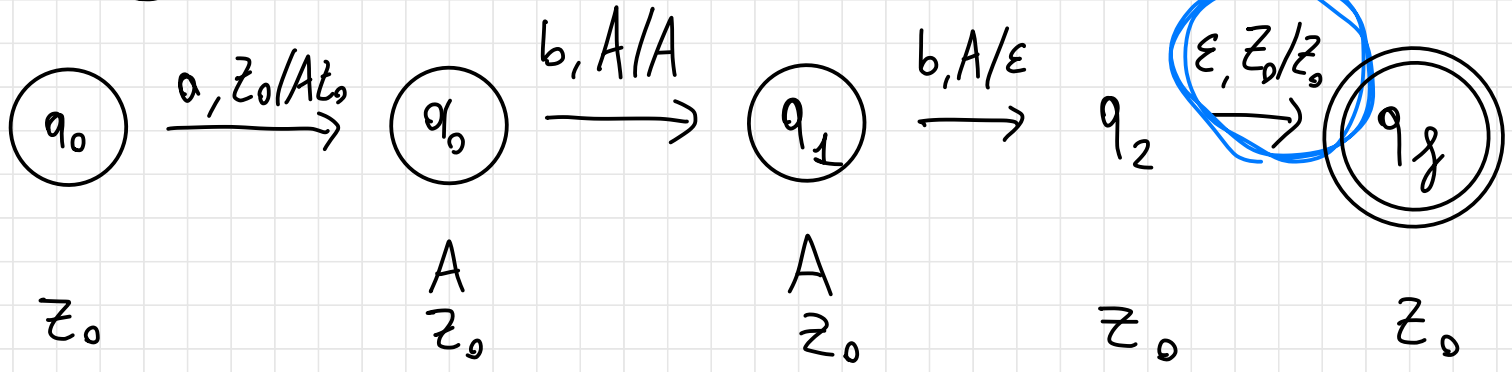
ogni due 'b' lette, spilo 'A'

controllo alla fine delle 'b' che ci sia solo 'Z_0'

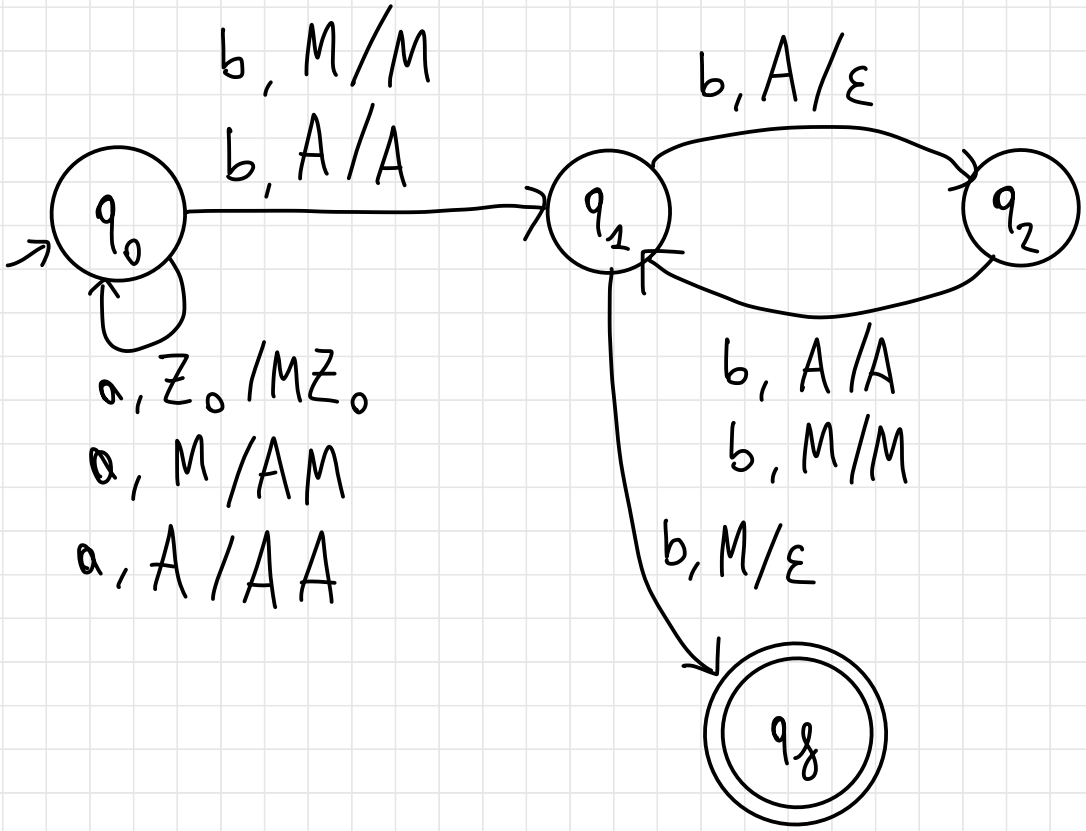



b, A/ε
b, A/A
ε, Z₀/Z₀
q₀
q₁
q₂
qf
a, Z₀/AZ₀
a, A/AA
b, A/A

abbb ∉ L


q₀
a, Z₀/AZ₀
q₀
b, A/A
q₁
b, A/ε
q₂
ε, Z₀/Z₀
qf
Z₀
A
Z₀
A
Z₀
Z₀
Z₀

es.: $L = a^n b^{2n}$, $n \geq 1$ senza ε-mossa



es.: $L = a^n b^p c a^n \cup a^n b^p d b^p$, $n, p \geq 1$

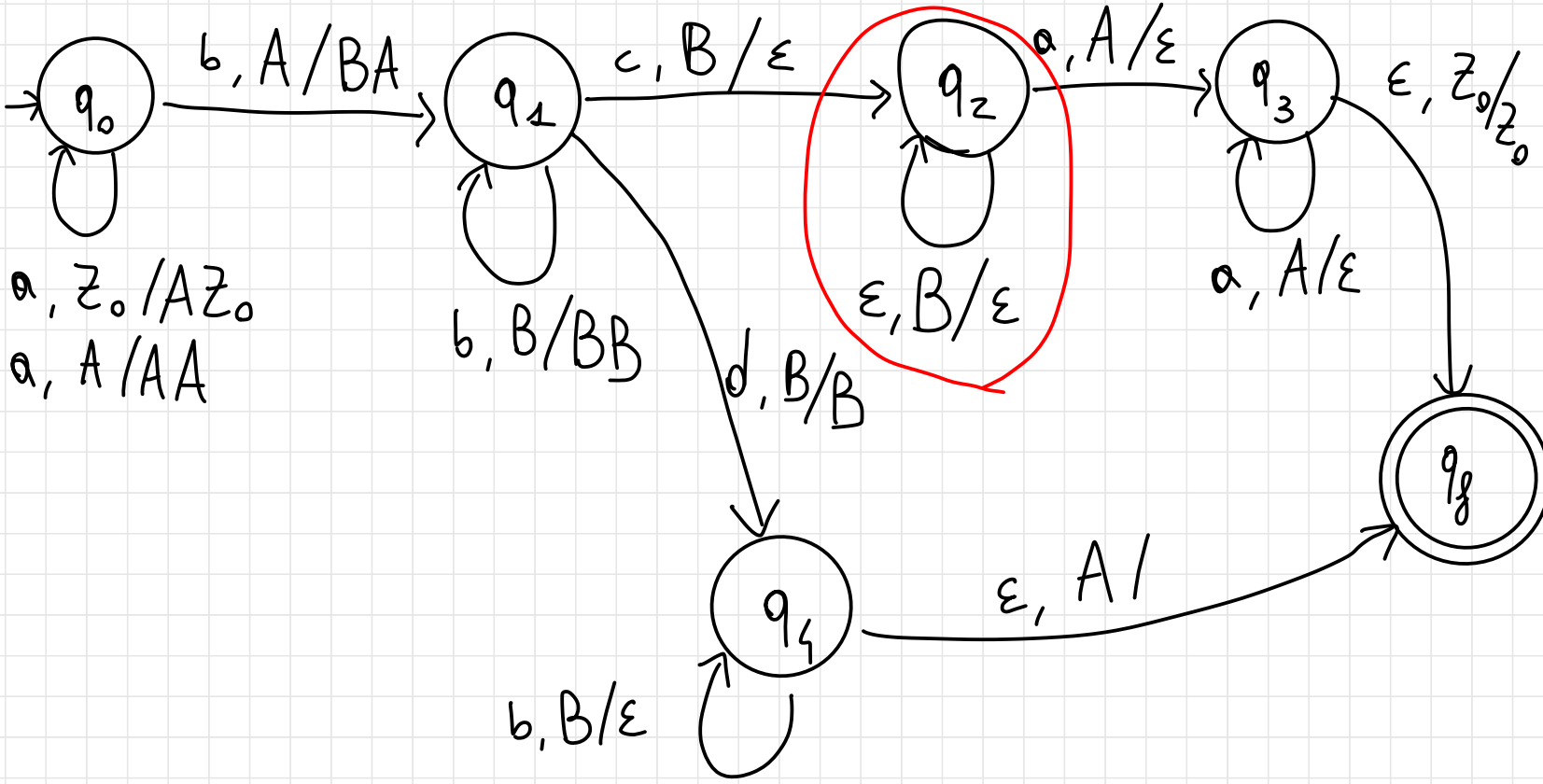
Idea: leggo 'a', impilo 'A'

leggo 'b', impilo 'B'

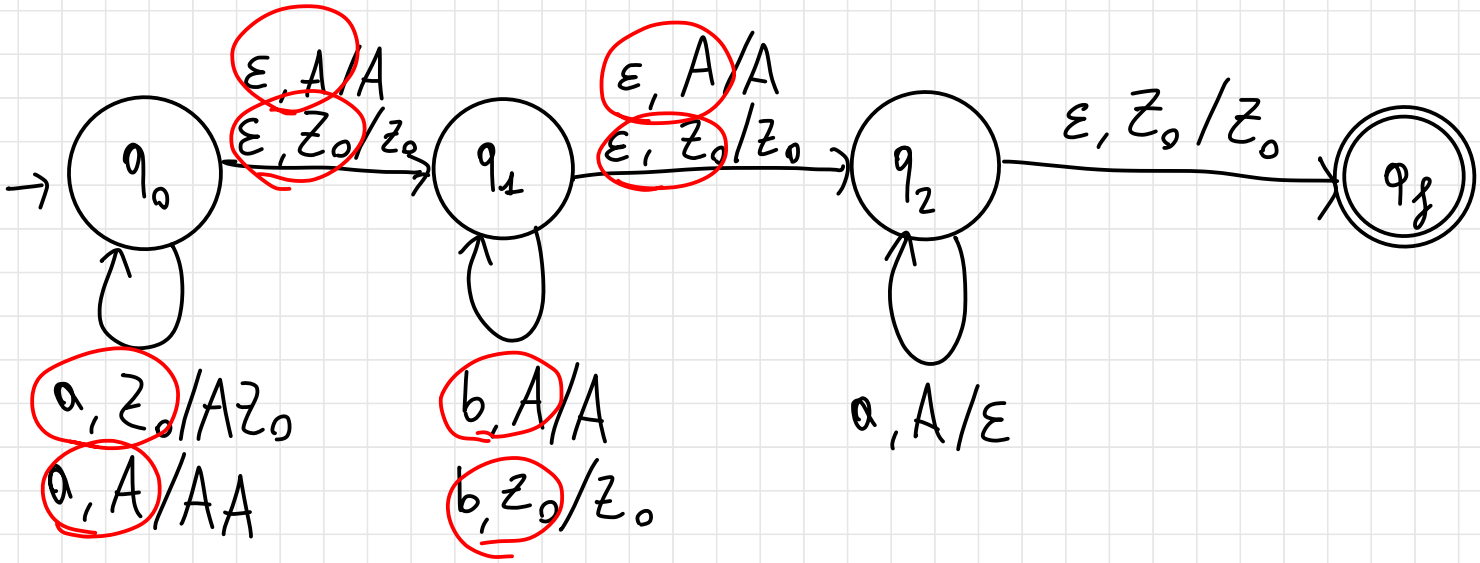
se leggo 'c': spilo le 'B', controllo 'a' con 'A'

se leggo 'd': controllo 'b' con 'B'

$L = a^n b^p c a^n \cup a^n b^p d b^p, \quad n, p \geq 1$



ls.: $L = a^n b^m a^n$, $n, m \geq 0$

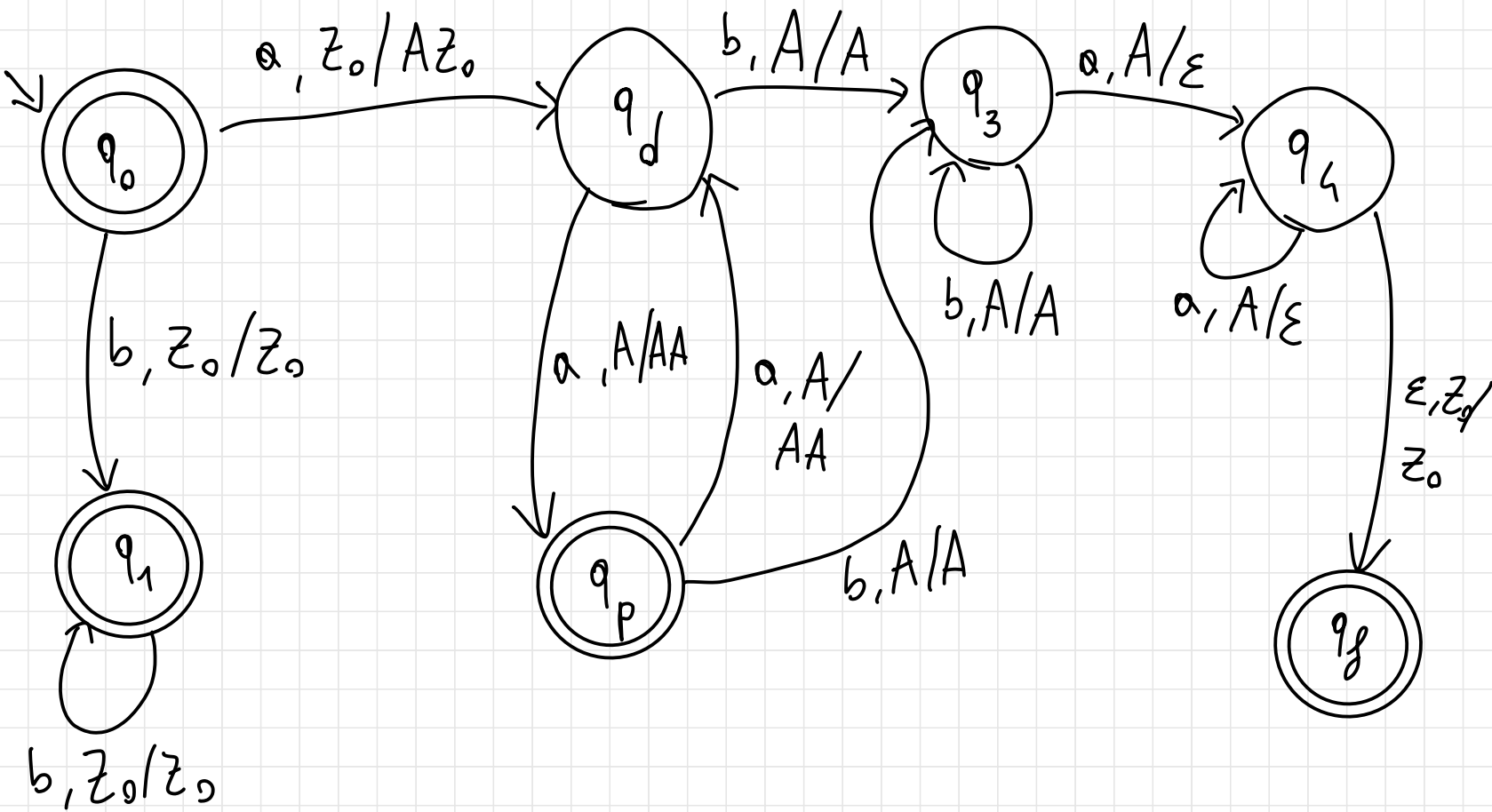


$L_1 = a^n b^m a^n \quad , \quad n, m \geq 1$

I problemi sorgono quando $n = 0$ oppure $m = 0$

L comprende:

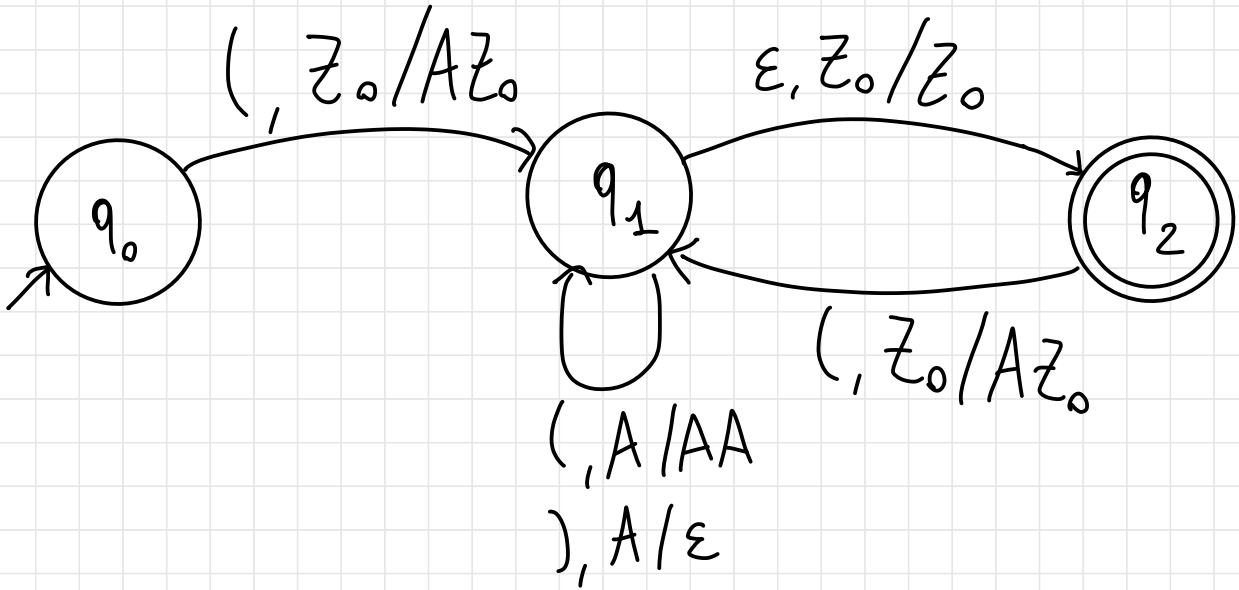
- ε $\quad (n = m = 0)$
- $b^m = b^+$ $\quad (n = 0, \ m \neq 0)$
- a^{2n} $\quad (n \neq 0, \ m = 0)$
- L_1 $\quad (n \neq 0, \ m \neq 0)$

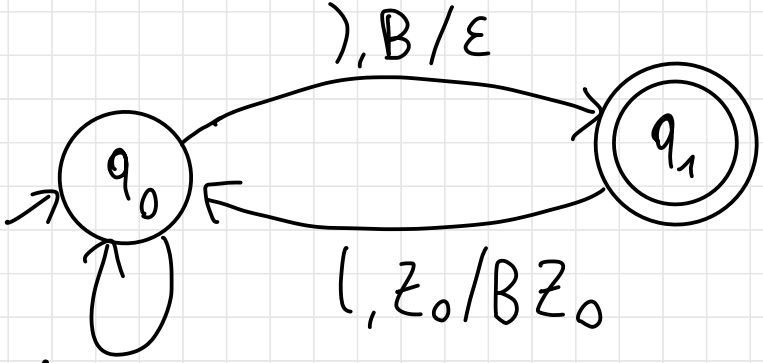

q_0
$a, Z_0/AZ_0$
q_d
$b, A/A$
q_3
$a, A/\varepsilon$
q_4
$b, Z_0/Z_0$
$a, A/AA$
$a, A/AA$
$b, A/A$
$a, A/\varepsilon$
$\varepsilon, Z_0/Z_0$
q_1
q_p
q_f
$b, Z_0/Z_0$
$b, A/A$

es.: Espressioni vuote ben parentesizzate (Dyck)

$((()))\in L$, $()()()\in L$, $(()())\in L$,

$\varepsilon \notin L$, $)(\notin L$ $(() \notin L$

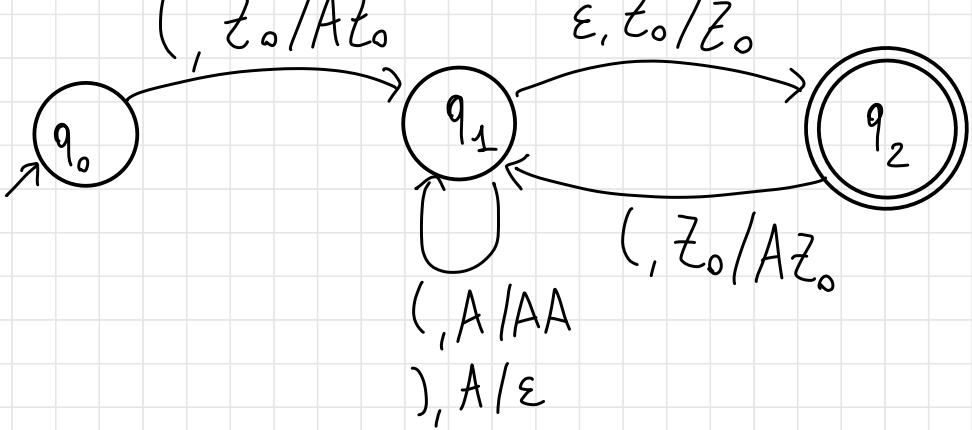




Stesso ling.

senza ε-mossa

(, Z0/ BZ0
(, B/AB
(, A/AA
), A/ε



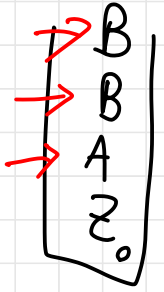
es.: $L = \{ w c w^R \mid w \in \{a, b\}^* \}$

$w = abb$, $w^R = bba$ (R = Reverse)

Idea: ad ogni 'a', impilo 'A'

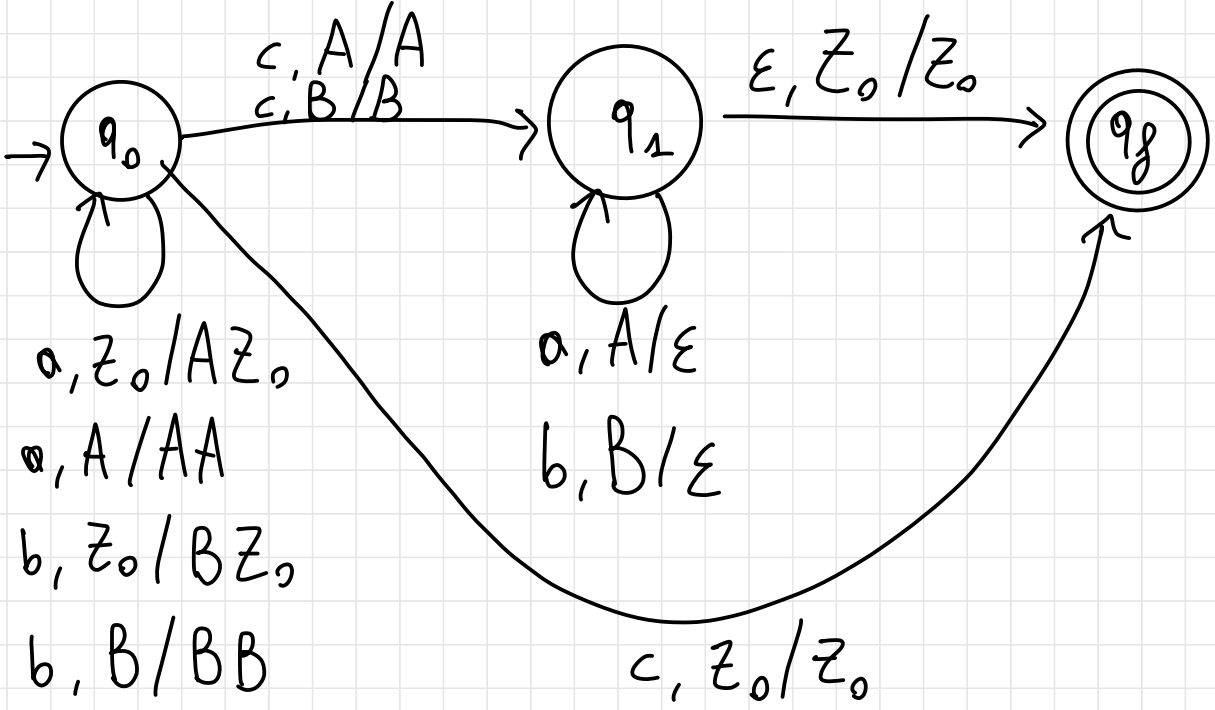
ad ogni 'b', impilo 'B'

leggo 'c': passo alla fase di "matching"



ad ogni 'a'/'b', spilo 'A'/'B'

controllo che alla fine ci sia 'Z_0'


c, A/A
c, B/B
ε, Z0/Z0
q0
q1
qf
a, Z0/AZ0
a, A/AA
b, Z0/BZ0
b, B/BB
a, B/AB
b, A/BA
a, A/ε
b, B/ε
c, Z0/Z0

sulle slide: $L = \{ w c w^R \mid w \in \{a, b\}^* \}$

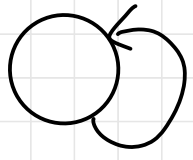
senza ε-mossa

servono <u>due</u> marker

sulle slide: Complemento di $L = a^n b^p c a^n \cup a^n b^p d b^p$

molto verboso.

Linee guida:

- No ε-mosse cicliche  $\varepsilon, A/AA$

  (si possono sempre eliminare)

- ε-mosse problematiche: fra dati fin. e non fin.

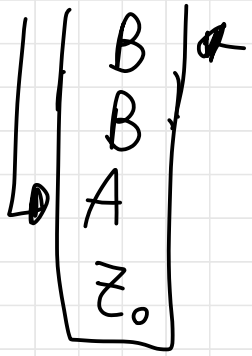
  V.: consiglio di eliminarle

- Completare PDA (Automa a Pila / Pushdown Automat
- Invertire stati fin. e non fin. on=)

wcw si può riconoscere con un PDA?

pila LIFO → NO

mem. distruttiva



wcw^R con PDA? Sì

ww^R con PDA? No con PDA <u>det.</u>

PDA traduttori

$$\langle Q, I, \Gamma, \delta, q_0, F, Z_0, \eta, O \rangle$$

O: alfabeto di output

$$\eta : Q \times I \cup \{\varepsilon\} \times \Gamma \longrightarrow O^*$$

es.: $L = a^k b^h c^h$, $h, k \geq 1$

trad. desiderata: $\tau(a^k b^h c^h) = d^{3h} e^k$

Idea:

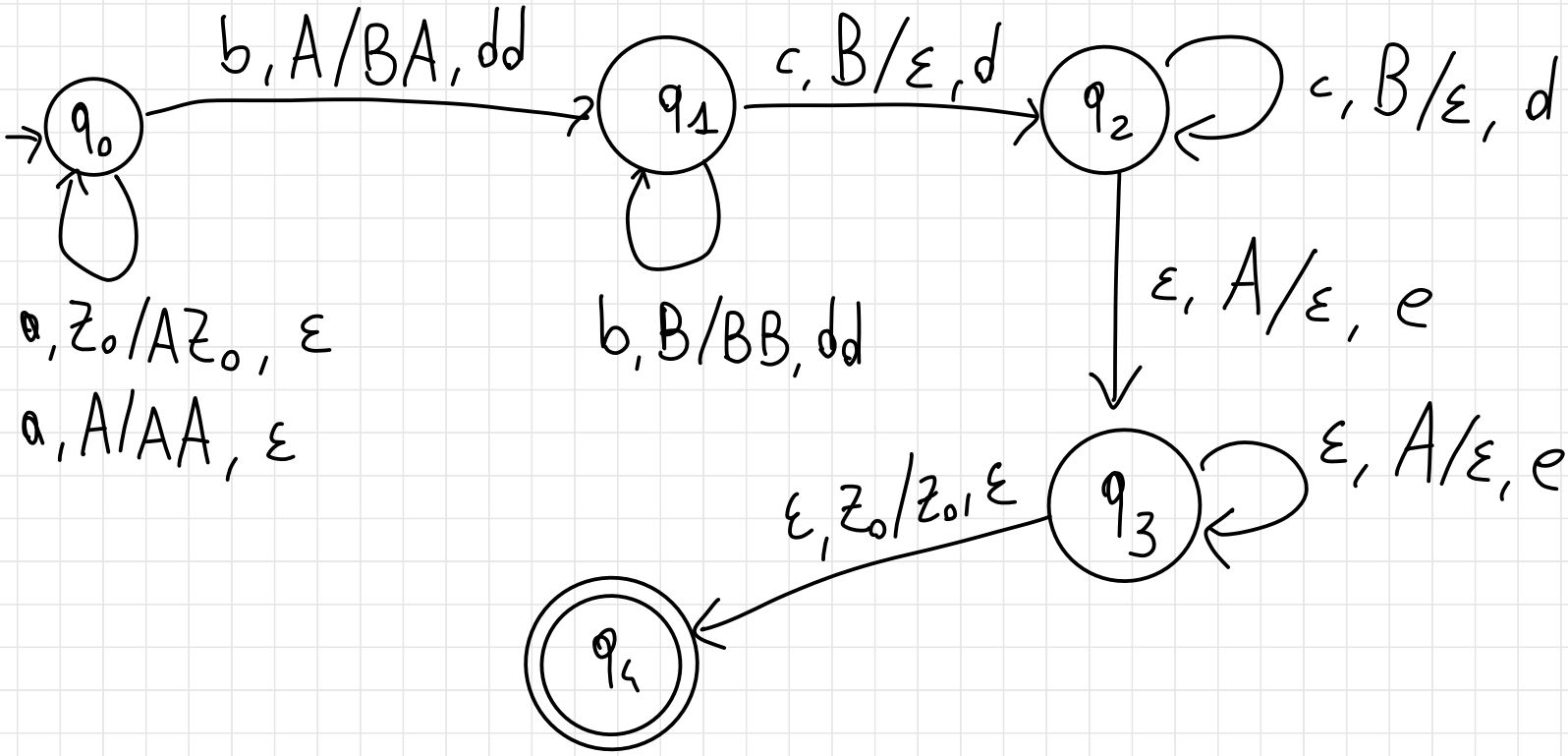
ad ogni 'a' letta, impilo 'A' → $\boxed{\begin{matrix} A^k \\ Z_0 \end{matrix}}$

ad ogni 'b' letta, scrivo in output 'dd', impilare 'B'

↕

ad ogni 'c' letta, scrivo in output 'd', spilo 'B'

alla fine, spilo le 'A' scrivendo 'e'


q0
q1
q2
q3
q4
b, A/BA, dd
c, B/ε, d
c, B/ε, d
a, Z0/AZ0, ε
a, A/AA, ε
b, B/BB, dd
ε, A/ε, e
ε, A/ε, e
ε, Z0/Z0, ε

sulle slide: esercizio su espressioni ben parentesizzate

$$L_1 = \{ a \mid ({}^{n} a ?)^{n} \mid n \geq 1 \}$$

$$L = \{ ({}^{n} y + z)^{n} \mid n \geq 0,\ y, z \in L \cup L_1 \}$$

$a + a$ $\qquad$ $(((\,(a) + a\,)))$

TM a k-nastri

$$\langle Q, I, \Gamma, \delta, q_0, F \rangle$$

Γ → alfabeto dei nastri

$\not b \in I$ "blank"

$$\delta : Q \times I \times \Gamma^k \longrightarrow Q \times \Gamma^k \times \{R, S, L\}^{k+1}$$

movimento testine

Condizione di accettazione:

se arrivo ad uno stato finale, accetto

es.: $L = a^n b^{n/2} c^{n/2}$, $n \geq 1$, 1 nostro dimen.