AUTOMI A PILA

Anche detti PDA (Push-Down Automata) e sono una estensione dei DFA che consente di riconoscere linguaggi non regolari. Infatti come vedremo i PDA sono equivalenti alle CFG.

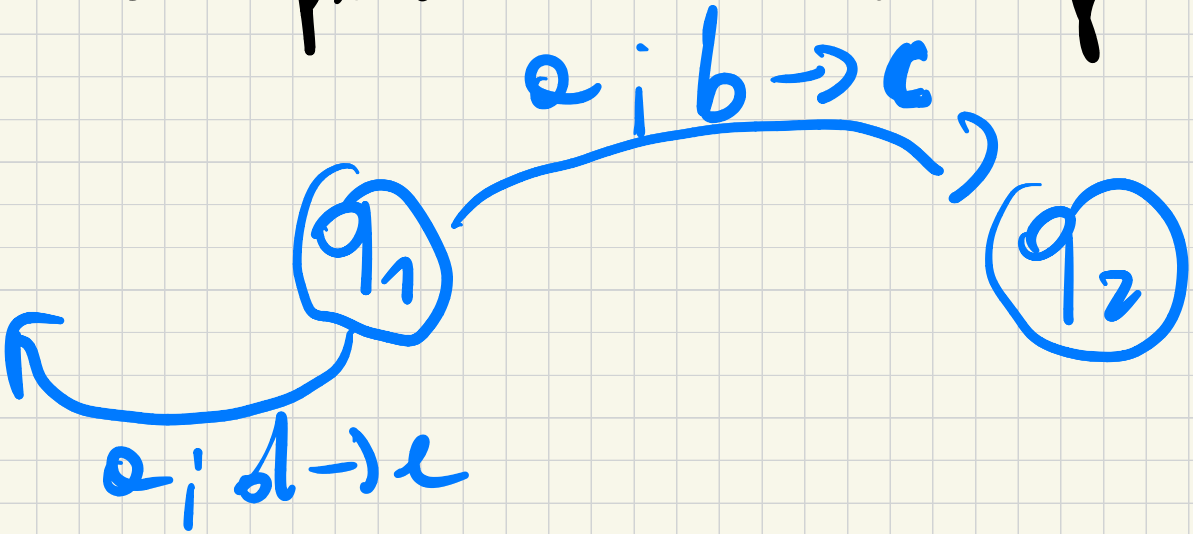
PDA: NFA + una pila "LIFO"

Ad ogni passo di computazione il PDA può operare sulla cima della pila:

- Sostituzione simbolo in cima alla pila.

- PUSH, inserimento simbolo in cima alla pila
- POP, eliminiamo il simbolo in cima.

Dal punto di vista grafico:



Il PDA si trova nello stato q_1 legge a e in cima alla pila c'è b. Rimpiazza b con c e va nello stato q_2

Alfabeto della pila $\Gamma_\varepsilon = \Gamma \cup \{\varepsilon\}$ può essere diverso da Σ_ε.

La funzione di transizione: dominio $Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon$, l'immagine sarà $Q \times \Gamma_\varepsilon$ (insieme delle parti).

DEF (PDA) Una PDA è una tupla $(Q, \Sigma, \Gamma, \delta, q_0, F)$ dove Q, Σ, q_0, F come nei DFA/NFA e inoltre:

- Γ è alfabeto finito di pila.

- $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \longrightarrow \mathcal{P}(Q \times \Gamma_\varepsilon)$.

Che succede in una transizione?

$$(q, c) \in \delta(p, a, b)$$

$$p, q \in Q \, ; \; a \in \Sigma_\varepsilon, \; b, c \in \Gamma_\varepsilon .$$

-) $a, b, c \neq \varepsilon$ come prima.

-) $c \neq \varepsilon$, $b = \varepsilon$, a in lettura, il PDA fa PUSH di c.

-) $c = \varepsilon$, $b \neq \varepsilon$, e in lettura, il PDA fa pop di b.

Le configurazioni: elementi di
$$Q \times \Sigma^* \times \Gamma^*$$

Accettazione: PDA M accetta $w = w_1 \dots w_n$ t.c. $w_j \in \Sigma$ se $\exists$ $r_0, \dots, r_m \in Q$ e stringhe $s_0, \dots, s_m \in \Gamma^*$ t.c.:

- All'inizio, $r_0 = q_0$ e $s_0 = \varepsilon$.
- $\forall\, i = 0, \dots, m$

$$(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a)$$

dove $s_i = at$ e $s_{i+1} = bt$

$a, b \in \Gamma_\varepsilon$; $t \in \Gamma^*$.

- $r_m \in F$.

Posso mettere le configurazioni in relazione.

$$(p, ax, by) \vdash_M (q, x, cy)$$

$p, q \in Q$; $c, b \in \Gamma_\varepsilon$, $y \in \Gamma^*$, $a \in \Sigma_\varepsilon$, $x \in \Sigma^*$

sse $(q, c) \in \delta(p, a, b)$

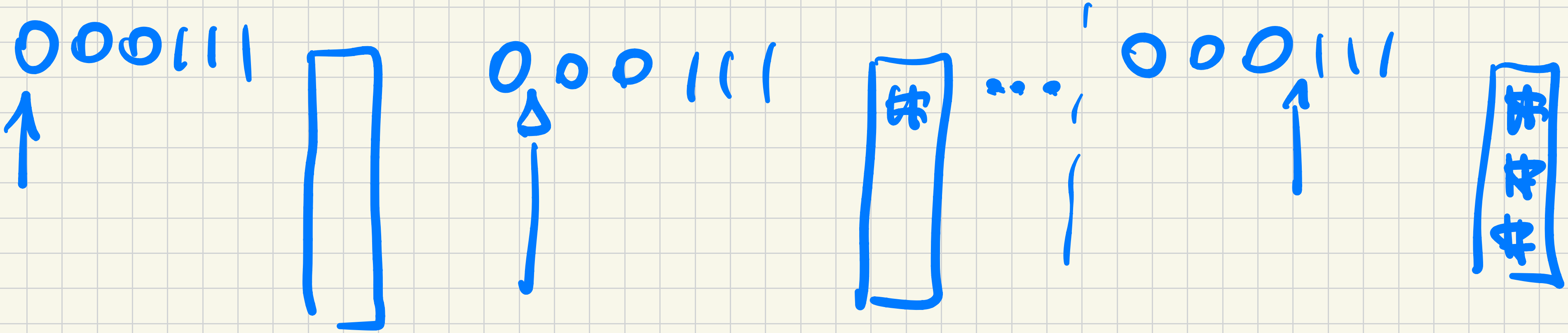
Come fatto per i DFA/NFA posso considerare la chiusura simmetrica e transitiva "$\vdash_M^*$". Pertanto:

$$L(M) = \{ w \in \Sigma^* : (q_0, w, \varepsilon) \vdash_M^* (q, \varepsilon, y), \; q \in F, \; y \in \Gamma^* \}$$

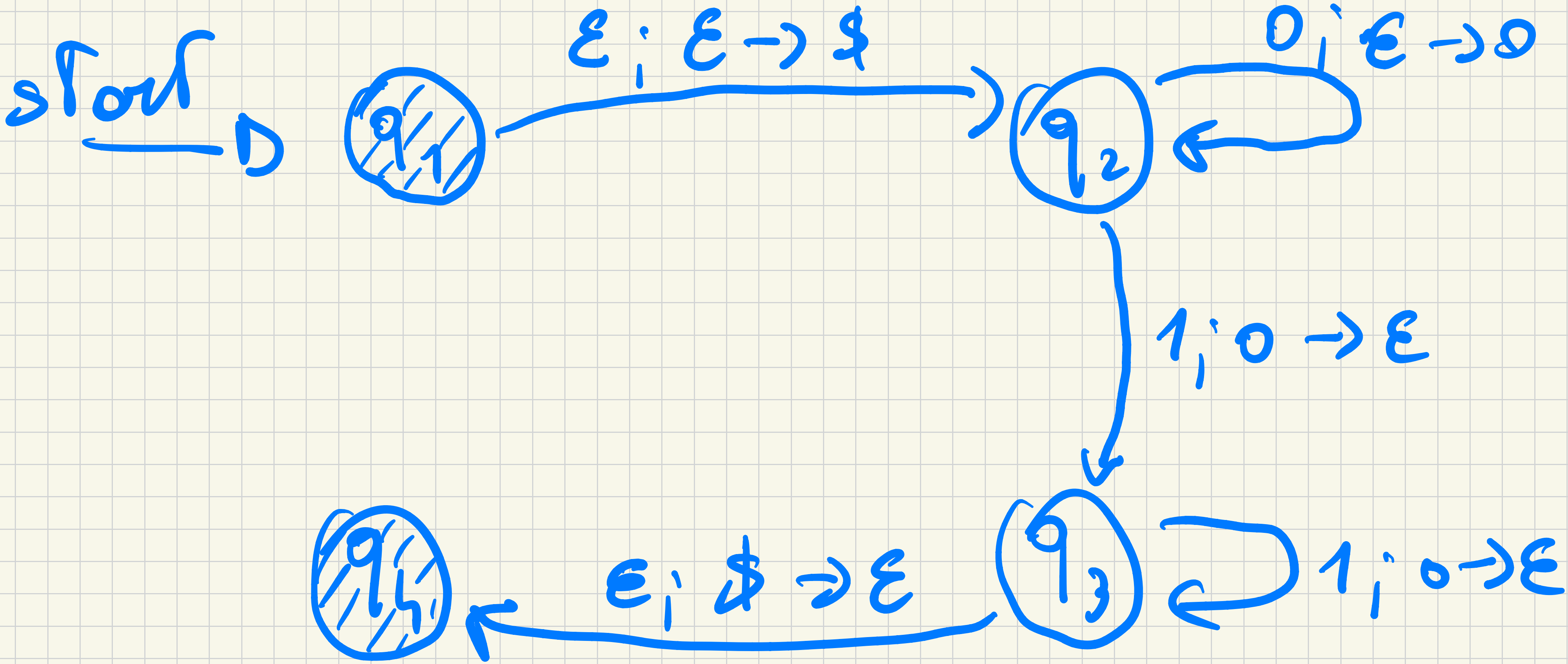
Nota: PDA accetta indipendentemente dal contenuto della pila. Wlog. si può assumere che la pila deve essere vuota.

Esempio: PDA per $L = \{0^n 1^n : n \geq 0\}$.

L'idea: $w = 000111$. Fino a quando leggo '0' faccio push di '#'; quando leggo '1' faccio pop di '#'. Se alla fine la pila è vuota, accetto.



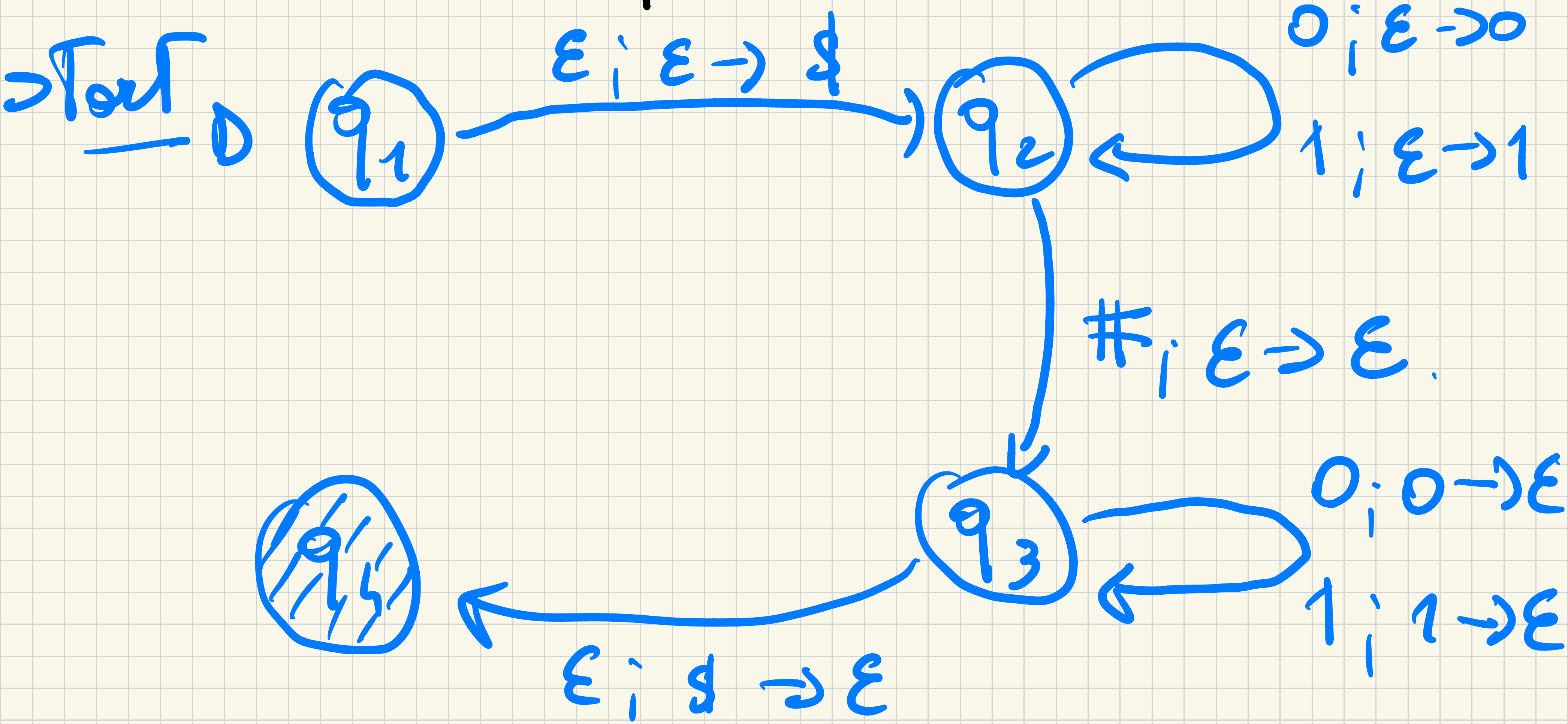
- $Q = \{q_1, q_2, q_3, q_4\}$
- $\Sigma = \{0, 1\}$; $\Gamma = \{0, \$\}$ #
- $F = \{q_1, q_4\}$; $q_0 = q_1$



Esempio: $L = \{ w \# w^R : w \in \{0,1\}^* \}$

- $Q = \{ q_1, q_2, q_3, q_4 \}$
- $\Sigma = \{ 0, 1, \# \}$
- $\Gamma = \{ 0, 1, \$ \}$
- $F = \{ q_4 \}$; $q_0 = q_1$.

Idea: scrivo w nella pila finché non trovo #. Poi provo a fare matching del resto dell'input con il contenuto della pila.



Se non ci fosse # ?? Lo studiamo

La transizione $q_2 \to q_3$ con

$$\varepsilon, \varepsilon \to \varepsilon.$$

Pure q_1 è accettante in questo caso.

Come prossimo passo, mostriamo equivalenza tra CFG e PDA.

TEO Un linguaggio è ACONTESTUALE se e solo se esiste un PDA che lo riconosce.

DIM. Saranno due direzioni.

LEMMA Se L è acontestuale, allora $\exists\ M \in PDA$ t.c. $L = L(M)$.

Intuizione: Il PDA deve riconoscere l'insieme delle stringhe generabili usando le regole della grammatica. Sia G la grammatica, con S v.n. e R le regole.

$$S \to V_1 a V_2$$

$$V_1 \to b V_2 V_3 \mid c V_2 V_4$$

$$\begin{array}{|c|}\hline S \\ \hline \$ \\ \hline \end{array} \longrightarrow \begin{array}{|c|}\hline V_1 \\ \hline e \\ \hline V_2 \\ \hline \$ \\ \hline \end{array} \longrightarrow$$

stesso discorso

per V_1

$w = e \ldots$

$$\begin{array}{|c|} b \\ V_2 \\ V_3 \\ e \\ V_2 \\ \$ \end{array} \qquad \qquad \begin{array}{|c|} c \\ V_2 \\ V_4 \\ e \\ V_2 \\ \$ \end{array}$$

Ad alto livello:

*) Inserisci S nella pila.

*) Ripeti:

- Se in cima alla pila c'è variabile A, seleziona non-det. una delle regole di G di tipo $A \to \dots$ e sostituisci A in maniera coerente nella pila.

- Se in cima c'è un terminale

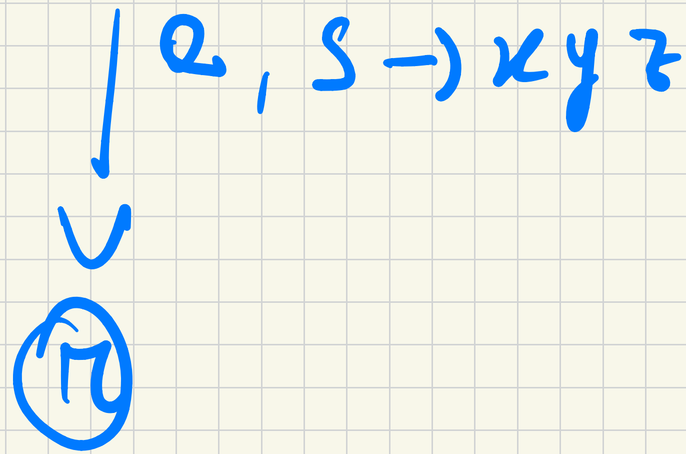
Made with Goodnotes

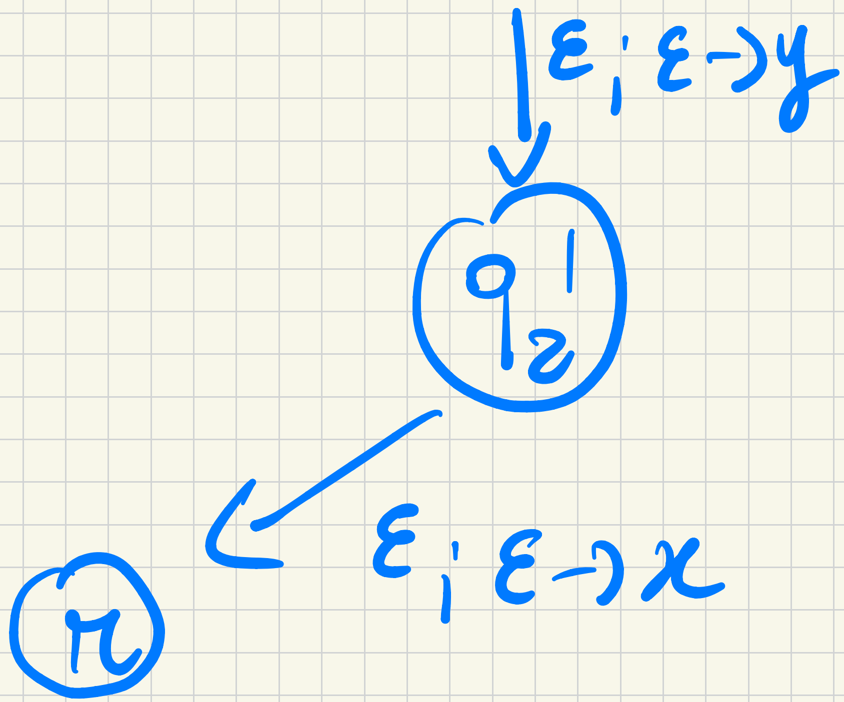
a, lo tira fuori e prova a fare matching con il prossimo carattere di input.
Se non matcha, rifiuta quel ramo di computazione.

DIM. Sia $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$.
Per semplificare la descrizione abbreviamo le transizioni in questo modo:

q

$q \xrightarrow{a, s \to z} q'$





$$\Longrightarrow \quad (r, xyz) \in \delta(q, a, s)$$

In generale permetto l'azione

$$(r, u) \in \delta(q, a, s)$$

$$u = u_1 \ldots u_\ell$$

$$(q'_1, u_\ell) \in \delta(q, e, s)$$

$$(q'_2, u_{\ell-1}) \in \delta(q'_1, \varepsilon, \varepsilon)$$

$$\vdots$$

$$(r, u_1) \in \delta(q'_{\ell-1}, \varepsilon, \varepsilon)$$

Definiamo il PDA:

- $Q = \{ q_{start}, q_{loop}, q_{acc} \} \cup Q'$

  dove Q' sono gli stati ausiliari necessari a fare ciò che abbiamo detto sopra.

- q_{start} stato iniziale.

- q_{acc} stato finale.

Devo definire la δ:

- Inizializzazione: inserisco $S\$$ nella pila, ovvero

$$\delta(q_{start}, \varepsilon, \varepsilon) = \{(q_{loop}, S\$)\}$$

- Nello stato q_{loop}:
  - Se la cima contiene $A \in V$

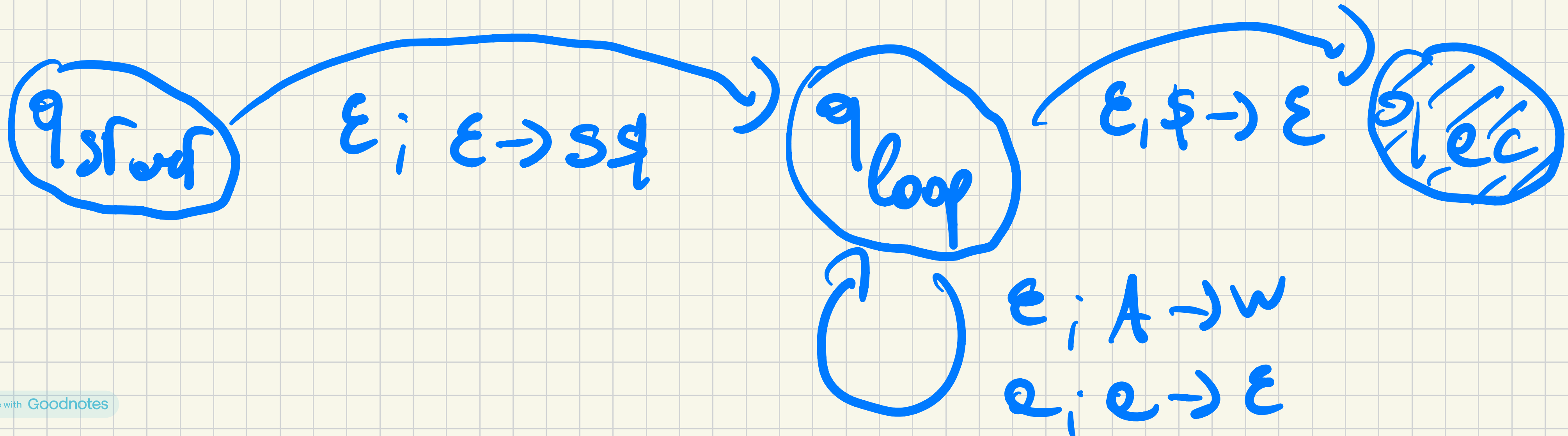
$$\delta(q_{loop}, \varepsilon, A) = \{(q_{loop}, w) : A \to w \text{ in } G\}$$

- Se la cima contiene $a \in \Sigma$

$$\delta(q_{loop}, a, a) = \{(q_{loop}, \varepsilon)\}$$

- Se la cima contiene $\$$

$$\delta(q_{loop}, \varepsilon, \$) = \{(q_{acc}, \varepsilon)\}$$

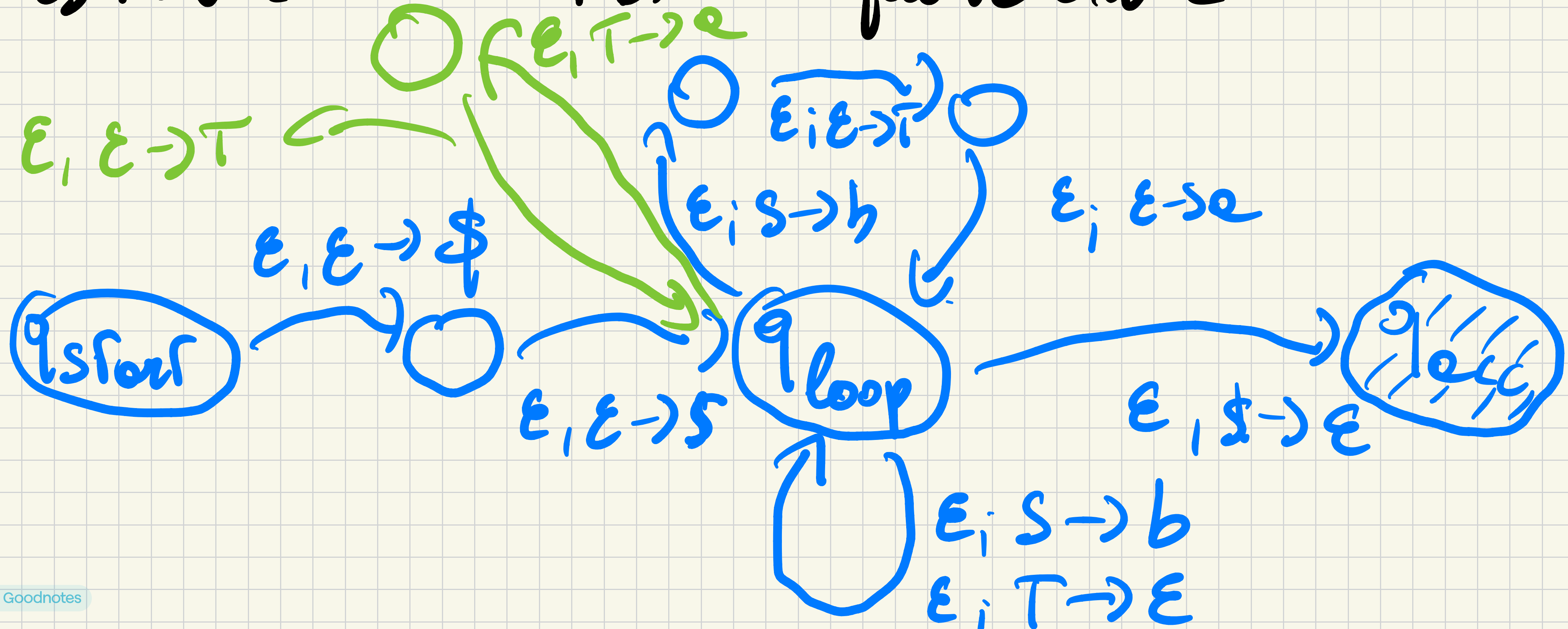


Esempio: Sia G la grammatica

$$S \to aTb \mid b$$

$$T \to Ta \mid \varepsilon$$

Costruire il PDA equivalente.



$e;e \to \varepsilon$

$b;b \to \varepsilon$

$S \to eTb \to eTeb \to eeb$