

Exercise : Properties CFG.

$w$  contains always  
TRES '1'.

$$1) L_1 = \{ w \in \{0,1\}^* : \#_1(w) \geq 3 \}$$

$$2) L_2 = \{ w \in \{0,1\}^* : w = w^R, |w| = 2n, n \in \mathbb{N} \}$$

$$3) L_3 = \{ a^i b^j c^k : i + j = k, n, i, k \geq 0 \}$$

Mostriamo una proprietà importante delle CFG.

Si possono scrivere in forma normale o CNF o di CHOMSKY.

DEF Una CFG è in forma normale se ogni regola è della forma:

$$A \rightarrow BC$$

$$A \rightarrow a \quad \text{oppure}$$

ovvero  $A, B, C \in V$ ,  $a \in \Sigma$ ,  $B, C \neq \epsilon$ .

Si ammette inoltre la regola  $S \rightarrow \epsilon$ .

THM. Ogni CFG ammette CFG equivalente  
in forma normale.

DIM. Dobbiamo trasformare le regole della CFG.

Per prime cose aggiungo  $S_0$  nuova variabile  
iniziale e la regola  $S_0 \rightarrow S$ .

Possiamo poi eliminare le  $\epsilon$ -regole:  $A \rightarrow \epsilon$ .

Per ogni occorrenza di  $A$  sul lato destro di una  
regola, aggiungo nuove regole con occorrenze cancellate.  
Te.

Esempio:  $A \rightarrow \epsilon$

$B \rightarrow u A v$ ;  $u, v$  stringhe di

## vorwahr / Terminals

$\Rightarrow B \rightarrow \underline{uv} \mid \underline{uAv}$   
 $\underline{A} \rightarrow \underline{\epsilon} \mid \underline{5}$

Esempio:

$B \rightarrow \underline{uAvAw}$

$B \rightarrow \underline{uvw} \mid \underline{uAvw} \mid \underline{uAvAw}$

$A \rightarrow \underline{5}$

$\mid \underline{uAvAw}$

Regole UNITARIE :  $A \rightarrow B$ . Per ogni occorrenza

di regole  $B \rightarrow u$ , esempio  $A \rightarrow u$ .

( $u$  sempre sequenza di vorwahr / Terminals)

$$\underline{\quad} B \rightarrow \underline{\quad} C \underline{\quad}; \quad \underline{\quad} C \rightarrow \underline{\quad} A \mid \underline{\quad} C \underline{\quad}$$

$$B \rightarrow A \mid C$$

$$C \rightarrow A \mid C$$

Define:  $A \rightarrow \mu_1 \mu_2 \dots \mu_k \quad k \geq 3$

①  $A \rightarrow \mu_1 A_1, A_1 \rightarrow \mu_2 A_2, A_2 \rightarrow \mu_3 A_3$

$$A_{k-2} \rightarrow \mu_{k-1} \mu_k$$

$$\mu_i = 1 \text{ VARIABLE} / 1 \text{ TERMINAL}$$

② Per ogni  $w_i$  terminale lo rimpiazzo con  $V_i$  e aggiungo  $V_i \rightarrow w_i$

Esempio:  $A \rightarrow a B C$

①  $A \rightarrow a A_1$   
 $A_1 \rightarrow B C$   $\curvearrowright$   $A \rightarrow V A_1$  ②  
 $A_1 \rightarrow B C$   
 $V \rightarrow a$

Esempio:  $A \rightarrow a B C$

$A \rightarrow a A_1$   
 $A_1 \rightarrow B C$   $\curvearrowright$   $A \rightarrow V_1 A_1$   
 $A_1 \rightarrow B V_2$



$$V_1 \rightarrow a$$

$$V_2 \rightarrow c$$

Esercizio:

$$S \rightarrow ASA \mid aB$$

$$A \rightarrow B \mid S$$

$$B \rightarrow b \mid \varepsilon$$

1)  $S_0 \rightarrow S :$

$$S_0 \rightarrow S$$

$$S \rightarrow ASA \mid aB$$

$$A \rightarrow B \mid S$$

$$B \rightarrow b \mid \varepsilon$$

2)  $\varepsilon$ -REGULAR:  $B \rightarrow \varepsilon$   $\mathbb{N}$

$$S_0 \rightarrow S$$

$$S \rightarrow ASA \mid aB \mid a$$

$$A \rightarrow B \mid S \mid \varepsilon$$

$$B \rightarrow b$$

$$S_0 \rightarrow S$$

$$A \rightarrow \varepsilon.$$

$$S \rightarrow ASA | aB | a | SA | AS | S$$

$$A \rightarrow B | S ; B \rightarrow b$$

3) Règles unitaires:  $S_0 \rightarrow S, S \rightarrow S$

$$S_0 \rightarrow ASA | aB | a | SA | AS$$

$$S \rightarrow ASA | aB | a | SA | AS$$

$$A \rightarrow B | S$$

$$B \rightarrow b$$

$A \rightarrow B :$      $S_0 \rightarrow A S A \mid e B \mid e \mid S A \mid A S$   
 $S \rightarrow A S A \mid e B \mid e \mid S A \mid A S$   
 $A \rightarrow b \mid S$   
 $B \rightarrow b$

$A \rightarrow S :$      $S_0 \rightarrow A S A \mid e B \mid e \mid S A \mid A S$   
 $S \rightarrow A S A \mid e B \mid e \mid S A \mid A S$   
 $A \rightarrow b \mid A S A \mid e B \mid e \mid S A \mid A S$

$$B \rightarrow b$$

4) Tutti il resto

$$S_0 \rightarrow AA_1 | UB | e | SA | AS$$

$$S \rightarrow AA_1 | UB | e | SA | AS$$

$$A \rightarrow b | AA_1 | UB | e | SA | AS$$

$$A_1 \rightarrow SA$$

$$U \rightarrow e$$

$$B \rightarrow b$$

# AUTOMI A PILA

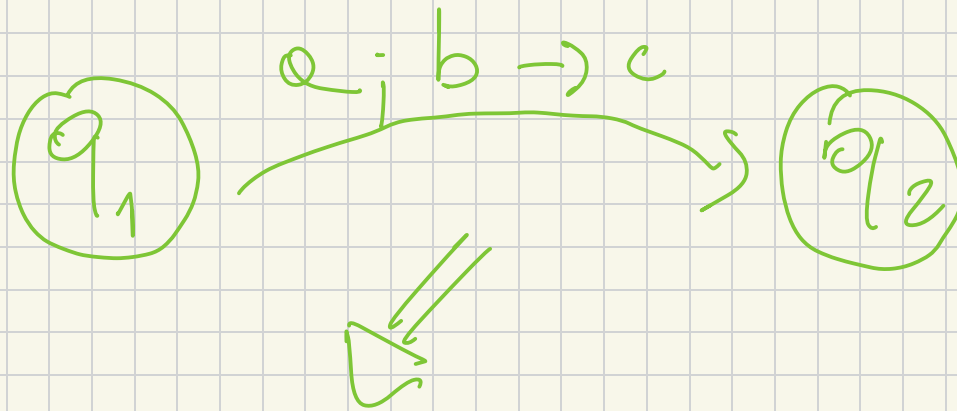
Potremmo il NFA in modo da avere un automa equivalente alla CFG.

Sarebbe non-deterministico, ma ovvio che aggiungere una PILA - in cui scrivere / rimuovere simboli in modo che - LIFO.

- Potrebbe sovrascrivere il simbolo in cima con altro simbolo.

- POP : rimuovere simbolo in cima

— PUSH : aggiungere simbolo in cima  
e spremere per altro verso il basso.



PDA legge  $a$   
 $b$  in cima alla pila

$b \rightarrow c$  (  $\begin{matrix} \epsilon \rightarrow c & \text{PUSH} \\ b \rightarrow \epsilon & \text{POP} \end{matrix}$  )

Così sarà alfabeto di PICA

⌈

DEF

Un PDA  $\bar{M} = (Q, \Sigma, \Gamma, \delta, q_0, F)$

ove  $Q, \Sigma, q_0, F$  come nei DFA/NFA e

-  $\Gamma$  è ALFABETO FINITO di push

-  $\delta: Q \times \Sigma \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma)$

Definiamo come compute,  $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$  accetta  $w = w_1 \dots w_m \in \Sigma^*$  se

$\exists r_0, \dots, r_m \in Q$  e stringhe  $s_0, \dots, s_m$

-  $r_0 = q_0$  ;  $s_0 = \varepsilon$

$$\forall i = 0 \dots m$$

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

$$s_i = at$$

$$s_{i+1} = bt$$

$$a, b \in \Gamma_\varepsilon$$

$$r_m \in F$$

Altrimenti, possiamo mettere in relazione le CONFIGURAZIONI

$$\text{SSE} \quad (p, ax, by) \vdash_M (q, x, cy)$$

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

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

Poss estendere per riflessività / Transitività

$$\vdash_M^*$$

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

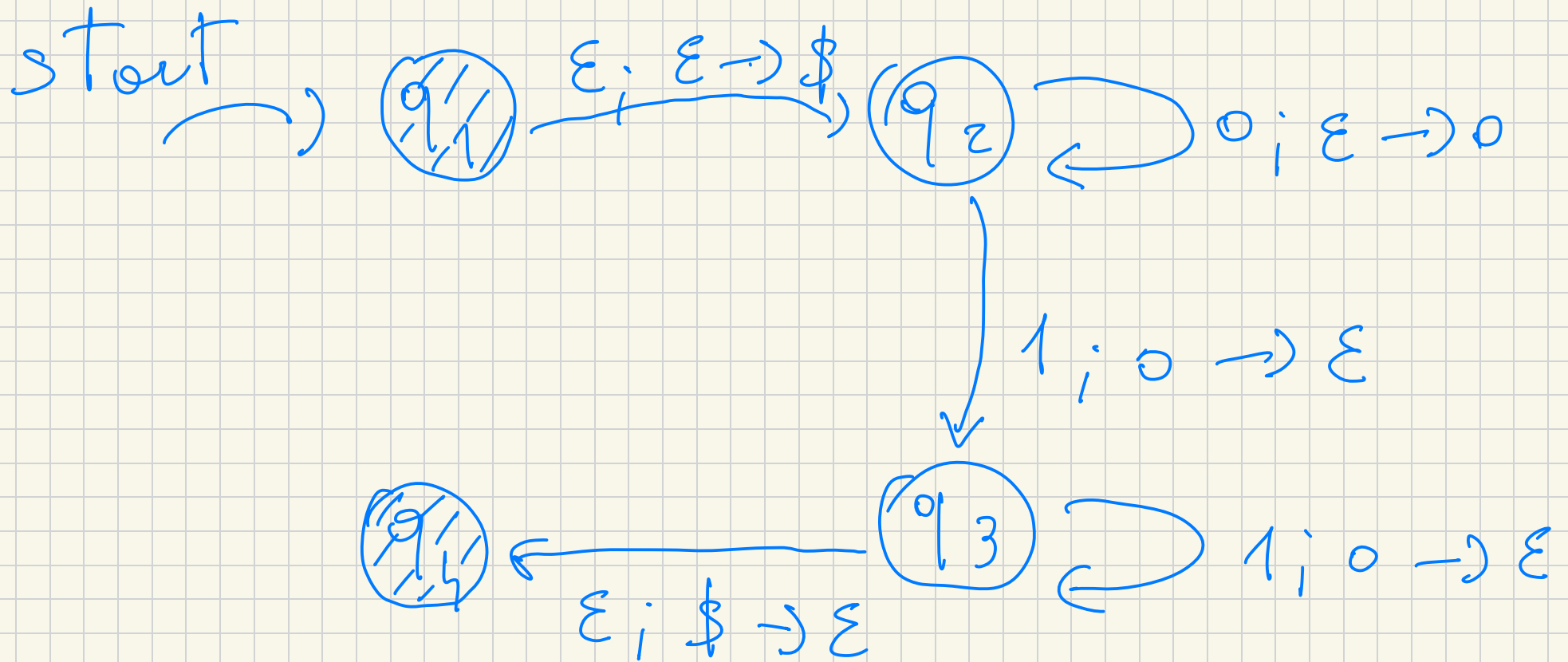
Sarà utile semplificare ed assumere wlog  
che un PDA sia sempre la parte prima  
di un accettore:  $(q_0, w, \varepsilon) \xrightarrow{M}^* (q \in F, \varepsilon, \varepsilon)$

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

Tobe: Finno e ch leggo '0', posta '0'.

Quando finito e leggo '1', pop '0'.

Accetto se e quando ho finito per la  
volute.



0 1 1

