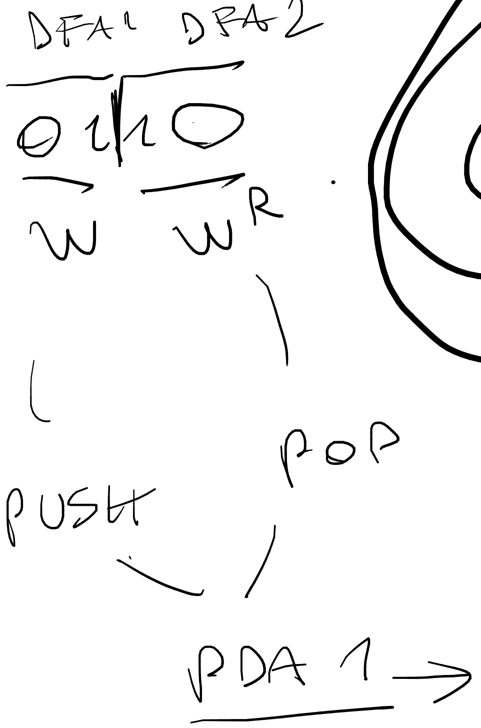
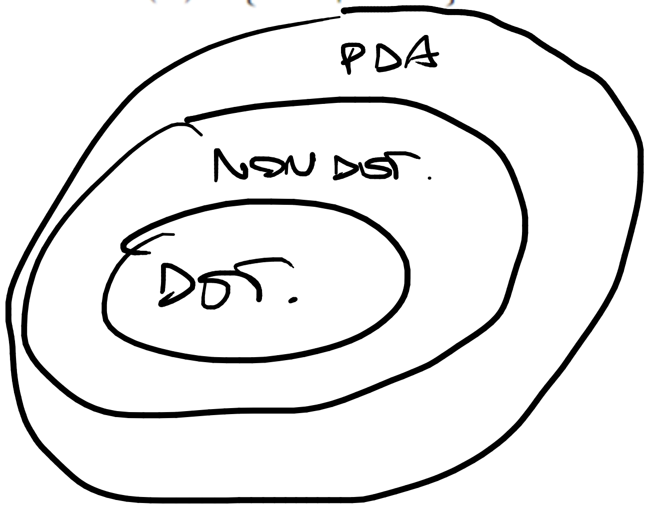
3. (12 punti) Dimostra che se B è un linguaggio regolare, allora il linguaggio

$$\text{PALINDROMIZE}(B) = \{ww^R \mid w \in B\}$$

è un linguaggio context-free.





SE L è REGOLARE → DFA / NFA

∃ NFA N EQUIVALENTE

TALE CHE $N = (Q, \Sigma, \delta, q_0, F)$

∃ UNA CFG EQUIVALENTE C

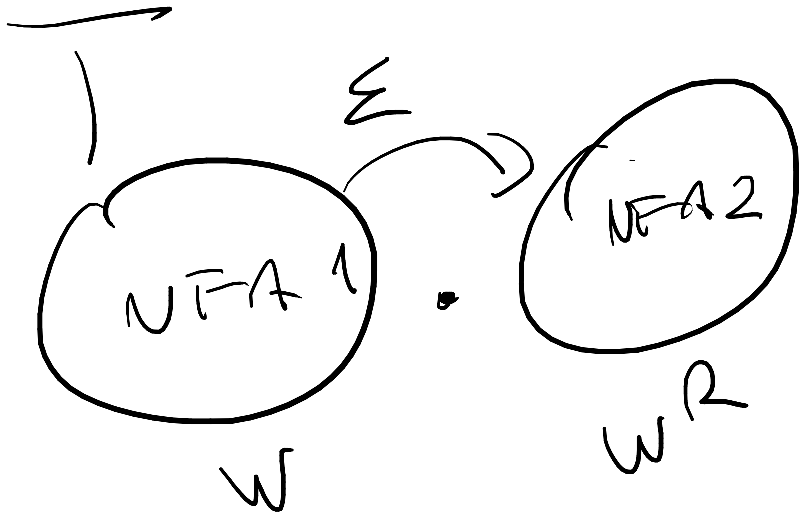
TALE CHE $C = (V, \Sigma, R, S)$

IN FORMA CHOMSKY TALE CHE:

$\Sigma_N = \Sigma_C$ (ST. ALFABETO)

$V_N = V_C$ (ST. VARIABILI)

R_N (ε-NFA)



$\left[N = NFA_1 \cdot NFA_2 \right]$ → CHIUSO PER LING. REGOLARI

$(S_N = F_C)$ → INIZIO DI NFA = FINE DI PDA

PRODUCE REGOLE DEL TIPO:

(PDA = CFG) $S \to aA$

$A \to aBa$

⋮ (STRUTTURA SPECULARE)

↓

MANTIENI "PASSO" IN MEZZO

GENERALE

- $S \to A_S$
- $A_a \to \varepsilon \to$ CASO 1
- $A_a \to aA(\delta(q_i, a))_i a \to$ CASO2

$A \to aBa$

$B \to cDc$

$A \to BC$

$A \to a$

CFG

2. (12 punti) Date due stringhe u e v, diciamo che u è una *permutazione* di v se u ha gli stessi simboli di v con ugual numero di occorrenze, ma eventualmente in un ordine diverso. Per esempio, le stringhe 01011,e 00111 sono entrambe permutazioni di 11001.

Dimostra che il seguente linguaggio non è regolare:

$$L_2 = \{uv \mid u, v \in \{0,1\}^* \text{ e } u \text{ è una permutazione di } v\}.$$

01011 = V

00111 = U = PERMUTAZIONE

$w = xy^iz$, $i \geq 0$, $x \neq \varepsilon$

$w = 0^p 1^p 0^p 1^p$ $p > 0$

$x = 0^p 0^p$

$y = 0^q 1^q$

$z = 0^{k-p-q} 1^{k-p-q}$

xy^0z

$y = 0^k 1^k$

NON REGOLARE

1. (12 punti) Se L è un linguaggio sull'alfabeto $\{0,1\}$, la *rotazione a sinistra* di L è l'insieme delle stringhe

$$\mathrm{ROL}(L) = \{wa \mid aw \in L, w \in \{0,1\}^*, a \in \{0,1\}\}.$$

Per esempio, se $L = \{0, 01, 010, 10100\}$, allora $\mathrm{ROL}(L) = \{0, 10, 100, 01001\}$. Dimostra che se L è regolare allora anche $\mathrm{ROL}(L)$ è regolare.

$$\left\{ wa \mid aw \in L, \; w \in \{0,1\}^*, \; a \in \{0,1\} \right\}$$

ROL

$$\left[L \text{ è REGOLARE} \to \text{DFA} \right]$$

$$A = (Q, \Sigma, \delta, q_0, F)$$

$\exists$ B EQUIVALENTE) DFA

S.T. $B = (Q', \Sigma', \delta', q_0', F')$

$\to \Sigma = \Sigma'$

$\to Q = Q'$

$\rightarrow \underline{q_0}$ INIZIALE $\rightarrow F'$

$A \in q_0' = q_1 \in B \rightarrow \ldots$

$A \in q_1' = q_2 \in B \rightarrow \ldots$

DFA per $a^{-1}L$:

- $A = (Q, \Sigma, \delta, \delta(q_0,a), F)$ dove $\delta(q_0,a)$ è il nuovo stato iniziale

DFA per {a}:

- $B = (\{q_1, q_2\}, \Sigma, \delta', q_1, \{q_2\})$ con $\delta'(q_1,a) = q_2$

Concatenazione A ∘ B:

- **Stati:** $Q \cup \{q_1, q_2\}$
- **Iniziale:** $\delta(q_0,a)$
- **Finali:** $\{q_2\}$
- **Transizioni:**
  - Mantieni transizioni di A
  - **Per ogni $f \in F$:** $\delta''(f, \varepsilon) = q_1$ (ε-transizione)
  - $\delta''(q_1, a) = q_2$

"Lo stato finale dell'uno è l'inizio dell'altro":

```
A: q₀' →* f ∈ F  --ε--> q₁ --a--> q₂ ∈ F'
     (riconosce w)      (riconosce a)
```

Risultato finale: ROL(L) = U_{a∈{0,1}} L(A_a ∘ B_a)

δ

0100 10100

100

Transizioni:

1. **"Indovina e sposta":** $\delta'(q_0, b) = \{(\delta(q_0,b), b)\}$ per $b \in \{0,1\}$
   - Non-deterministicamente indovina che b è il simbolo da spostare
2. **Simula normalmente:** $\delta'((q,a), c) = \{(\delta(q,c), a)\}$ per $c \in \{0,1\}$
   - Continua simulazione "ricordando" che a deve essere verificato alla fine
3. **Verifica finale:** Accetta in (f,a) se dopo aver letto wa, siamo in stato f e il simbolo memorizzato è a

Meccanismo:

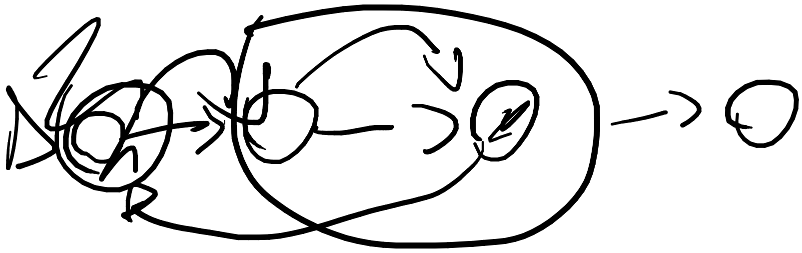
```
Input: wa
↓
Indovina a, simula su w, verifica che δ*(q₀,aw) ∈ F
```

Non esiste costruzione "naturale" che preservi la struttura originale.

Ecco perché:

- I linguaggi regolari NON sono chiusi sotto alcune rotazioni
- ROL funziona solo per costruzioni "artificiali" (quotient)
- Il non-determinismo serve proprio per gestire questi shift

LOOP INTERNO



***1.67** Let the ***rotational closure*** of language A be $RC(A) = \{yx \mid xy \in A\}$.

a. Show that for any language A, we have $RC(A) = RC(RC(A))$.

b. Show that the class of regular languages is closed under rotational closure.

↑ SIMULO A...

(b) Show that the class of regular languages is closed under rotational closure.

Solution. Let A be an arbitrary regular language and $M_A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ be a DFA that recognizes A. To prove that $RC(A)$ is also regular, we construct from M_A (as a building block) an NFA N that recognizes $RC(A)$. We first elaborate on the basic ideas and then give a formal definition for N.

Suppose N is given an input $w = yx$ for some $x, y \in \Sigma^*$ such that $xy \in A$. Let q_x be the state in which M_A ends up after reading x. Starting from q_x, M_A should end at some final state after reading y. For N to accept w, we let N simulate M_A from q_x and, after reading y and reaching a final state, make an epsilon transition (which needs to be added to M_A) to the initial state q_A of M_A and continue simulating M_A with the rest of the input. If N eventually ends up at q_x, then the input w is of the correct form of yx such that $xy \in A$. Any state of M_A may act as q_x. For N to start

and finish the simulation at the same state, we need $|Q_A|$ copies of M_A, one for each state in Q_A, with an epsilon transition added from every final state to the initial state. To start the simulation of M_A from any state, N has an epsilon transition from its initial state to every state of M_A.

So, $N = (Q_A \times Q_A \cup \{q_0\}, \Sigma_\varepsilon, \delta, q_0, \bigcup_{q \in Q_A} \{(q,q)\})$, where

$$
\begin{cases}
\delta(q_0, \varepsilon) = \bigcup_{q \in Q_A} \{(q,q)\} & \\
\delta((q_1,q_2),a) = \{(q,q_2) \mid \delta_A(q_1,a) = q\} & q_1, q_2 \in Q_A \text{ and } a \in \Sigma \\
\delta((q_1,q_2),\varepsilon) = \{(q_A,q_2)\} & q_1 \in F_A \text{ and } q_2 \in Q_A \\
\delta(q,a) = \emptyset & \text{otherwise}
\end{cases}
$$

□

3. (12 punti) Mostra che per ogni PDA P esiste un PDA P_2 con due soli stati tale che $L(P_2) = L(P)$.
Suggerimento: usate la pila per tenere traccia dello stato di P.

Dato $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, costruisco $P_2 = (Q_2, \Sigma, \Gamma_2, \delta_2, s_0, Z_{20}, F_2)$:

$\forall P$, PDA $P \Rightarrow \exists P_2$ 1 2 stati $L(P_2) = L(P)$

Componenti di P_2:

Stati: $Q_2 = \{s_1, s_2\}$ (solo 2 stati!)
Alfabeto pila: $\Gamma_2 = Q \times \Gamma$ (coppie [stato, simbolo])
Stato iniziale: $s_0 = s_1$
Simbolo pila iniziale: $Z_{20} = [q_0, Z_0]$
Stati finali: $F_2 = \{s_2\}$

INPUT

a , b → c

STACK PUSH

P_2 PUSH POP

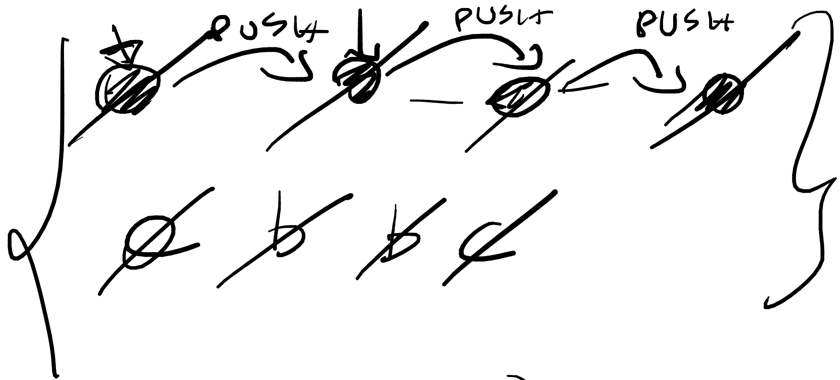
$w = abbabbcdd$

$$
\begin{pmatrix}
\varepsilon, \varepsilon \to a & \varepsilon, \varepsilon \to c \\
\varepsilon, \varepsilon \to b & \varepsilon, \varepsilon \to d
\end{pmatrix}
$$

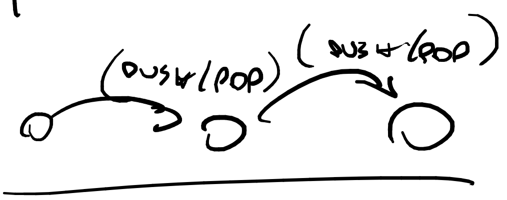
```
1. Simulazione delle transizioni di P:
Per ogni δ(q, a, A) = (q', α) in P con α = B₁B₂ … Bₖ:
δ₂(s₁, a, [q, A]) = {(s₁, [q', B₁][q', B₂] … [q', Bₖ])}
Meccanismo:

Poppa [q, A] (stato corrente + top della pila)
Pusha [q', Bᵢ] per ogni Bᵢ ∈ α (nuovo stato + nuovi simboli)

2. Transizione verso accettazione:
Per ogni qf ∈ F e A ∈ Γ:
δ₂(s₁, ε, [qf, A]) = {(s₂, [qf, A])}
```

S_1 = PUSH

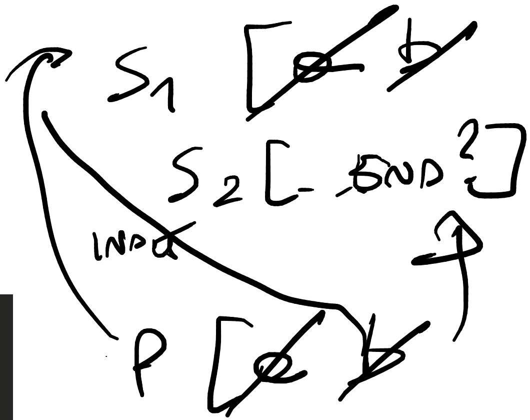
S_2 = POP



→ COME LA PILA...



S_1 = PUSH/POP



Meccanismo:

- Poppa [q, A] (stato corrente + top della pila)
- Pusha [q', Bᵢ] per ogni Bᵢ ∈ α (nuovo stato + nuovi simboli)

2. Transizione verso accettazione: Per ogni qf ∈ F e A ∈ Γ:

δ2(s1, ε, [qf, A]) = {(s2, [qf, A])}

SE P ACCETTA → ACCETTA

(NO AUTOMATISMI)

2. (12 punti) Considera l'alfabeto $\Sigma = \{0, 1\}$, e sia L_2 l'insieme di tutte le stringhe che contengono almeno un 1 nella loro prima metà:

$$L_2 = \{uv \mid u \in \Sigma^*1\Sigma^*, v \in \Sigma^* \text{ e } |u| \leq |v|\}.$$

Dimostra che L_2 non è regolare.

$u\, v$ $\quad \Sigma^* 1 \Sigma^*$ $\quad \Sigma = \{0,1\}^*$

$v \in \Sigma^*$

$w = \underbrace{\underbrace{0^p 1}_{u}\ 0^{2p}}_{u \text{ (ancora)}}$ $\quad w\ v$

$|u| \leq |v|$

$p \leq 2p$

$x = 1$

$y = 0^p$

$z = 0^{k-p}\, 0^{2p}$

$w = xy^iz,\ y \neq \varepsilon,\ i \geq 0$

$$w = 0^k 1 0^{k+1}$$

$$w = xy^i z$$

$x = \varepsilon$

$y = 0^p$

$z = 10^{p+1}$

$$xy^i z = xy^2 z =$$

$$= 0^{2p} 1 0^{p+1}$$

$$= 0^{3p+1} 1$$

$$w = xy^2 z = 10^2 0^{k-p+2p}$$

$$= 10^{k+p+2}$$