SUBSTITUȚII

Def: Fie Σ_1, Σ_2 două alfabete. $\sigma: \Sigma_1^* \to 2^{\Sigma_2^*}$ s.n. substituție dacă:
1) $\sigma(\lambda) = \{\lambda\}$
2) $\sigma(x \cdot y) = \sigma(x) \cdot \sigma(y)$

ex.: $\Sigma_1 = \{a,b\}$ $\Sigma_2 = \{a,b,c\}$, $\sigma: \Sigma_1^* \to 2^{\Sigma_2^*}$

$\sigma(a) = \{ab, ac, b\}$, $\sigma(b) = \{b, ba\}$

$\sigma(ba) = \{b, ba\} \cdot \{ab, ac, b\} = \{bab, bac, bb, baab, baac\}$

Def: $K: \Sigma_1^* \to \Sigma_2^*$ s.n. morfism dacă:
1) $K(\lambda) = \lambda$
2) $K(x \cdot y) = K(x) \cdot K(y)$, $(\forall)\ x, y \in \Sigma_1^*$

ÎNCHIDEREA LA SUBSTITUȚII ȘI MORFISME INVERSE

Fie $K: \Sigma_1^* \to \Sigma_2^*$ un morfism.

Pentru $w \in \Sigma_2^*$, $K^{-1}(w) = \{x \mid x \in \Sigma_1^*, \$\ K(x) = w\}$

și pentru limbaje:

$$K^{-1}(L) = \{x \mid K(x) \in L, x \in \Sigma_1^*\}, L_2 \subseteq \Sigma_2^*$$

(2) Să se demonstreze că $L = \{a^n b a^n \mid n \geq 1\}$ nu e regulat. (fără lema de pompare)

pp. că L e regulat $\Rightarrow$ pentru orice morfism R avem $R^{-1}(L)$ e regulat. Fie $R_1: \{a,b,c\}^* \to \{a,b\}^*$ cu $R_1(a) = a$
$R_1(b) = ba$.
$R_1(c) = a$

$\Rightarrow R_1^{-1}(L)$ e regulat

$R_1^{-1}(L) = \{x^n b y^{n-1} \mid x, y \in \{a, c\}, n \geq 1\}$ e regulat

Fie încă un morfism $R_2: \{a;b,c\}^* \to \{0,1\}^*$, $R_2(a) = 0$
$R_2(b) = 1$, $R_2(c) = 1$ $\Rightarrow$ $R_2(R_1^{-1}(L)) \cap 0^* 1^* = \{a^n b^n \mid n \geq 1\}$

sau $R_1^{-1}(L) \cap a^* b c^* = \{a^n b c^{n-1} \mid n \geq 1\} = L'$. $R_2(L') =$
$= \{a^n b^n \mid n \geq 1\}$ contradicție

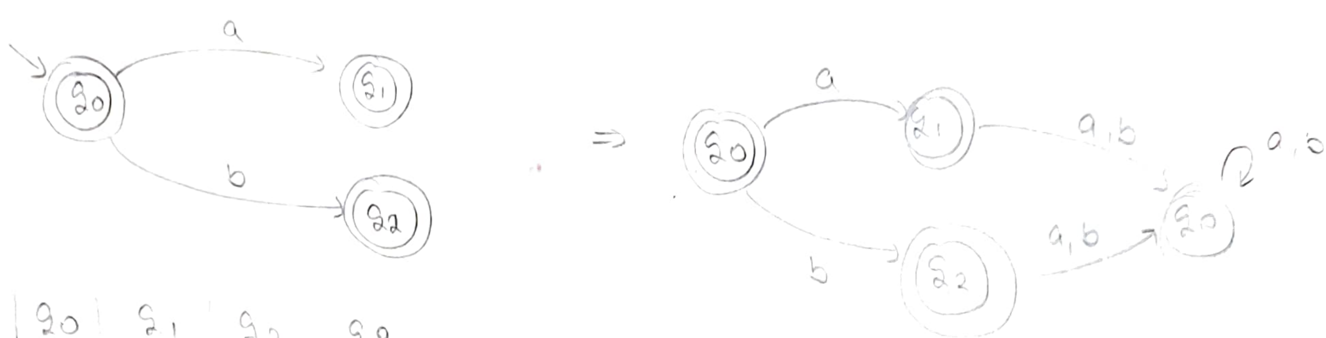
~~Myhire Herodu~~

DFA – DFA min

Pas 1:
Împart mulțimea de stări în 2 grupuri: finale și nefinale.

Pas 2: Recursiv, verific dacă oricare 2 stări dintr-un set sunt echivalente. Dacă nu sunt echivalente se sparge mulțimea.

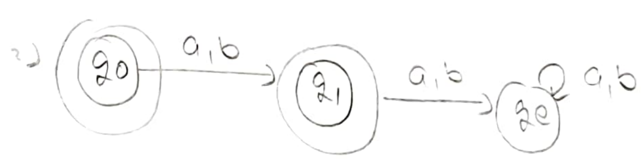
Pas 3: Când nu se mai fac modificări înseamnă că am găsit stările.



	q_0	q_1	q_2	q_e
q_0	1			
q_1	1	1		
q_2	1	1	1	
q_e	0	0	0	1

q_0 q_1 q_2 q_e

(q_0) $(q_1\ q_2)$ (q_e)

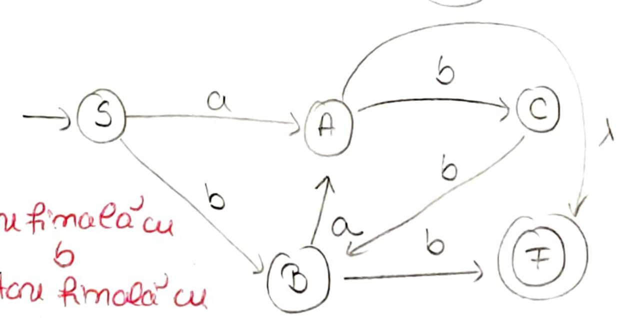


$S \to aA \mid bB$

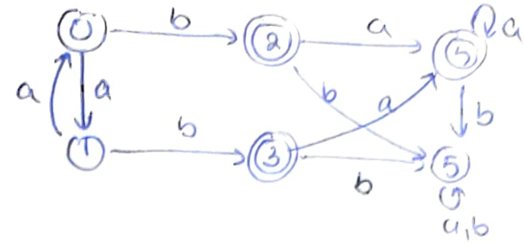
$B \to aA \mid b$ ← duce direct în stare finală cu b

$A \to bbB \mid \lambda$

↑ duce în stare finală cu λ



DFA - DFA min (Myhill Nerode)



1. Determinarea stărilor echivalente.
   ! DFA-ul trebuie să fie complet
   - facem tabelul

	0	1	2	3	4	5
0	1					
1	0	1				
2	0	0	1			
3	0	0	1	1		
4	0	0	1	1	1	
5	0	0	0	0	0	0

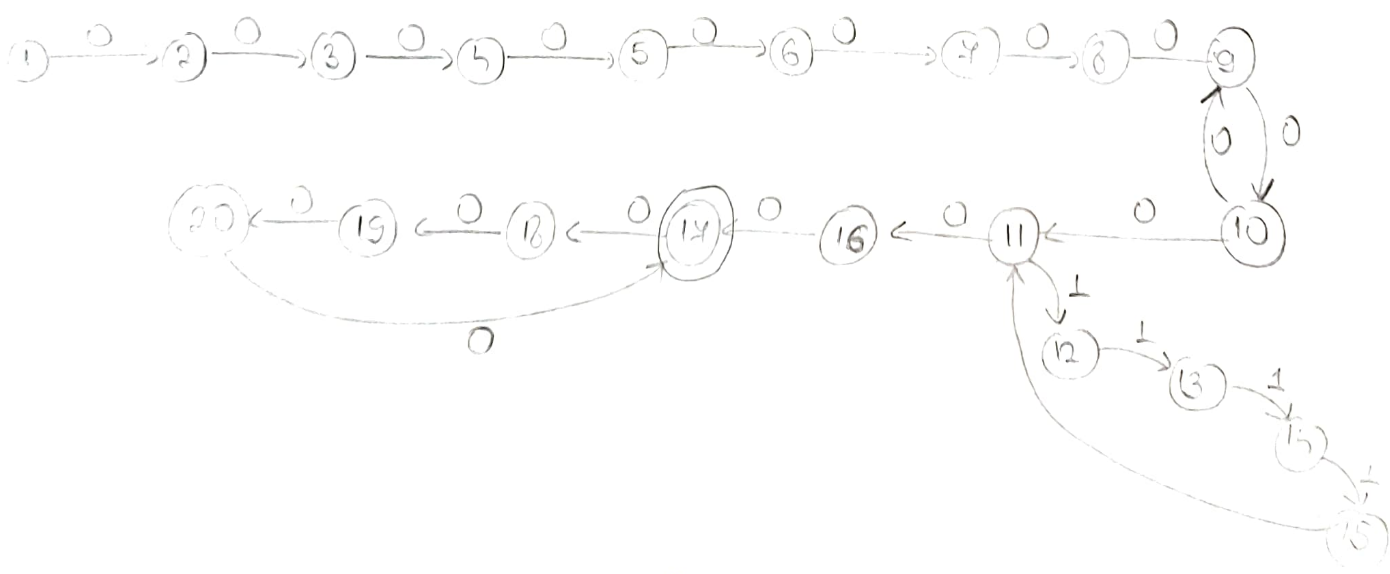
- marchez cu 0 perechile (stare finală, ne finală)
- marchez cu 0 toate perechile (q, r) ai ($\delta(q,\alpha), \delta(r,\alpha)$) sunt marcate ca false, $\alpha \in \Sigma$
- repetăm până nu mai există modificări

2. Gruparea stărilor echivalente
   - stările (a,b) sunt echivalente dacă au true între ele
   - tranzițiile vor fi aceleași din automatul inițial doar că ținând cont de grupări

3. Calcularea stărilor inițiale și finale
   - stare inițială = cea care conține starea inițială din automatul precedent
   - stările finale = compuse din stări finale

4. Eliminarea stărilor ~~dead~~ dead end
5. Eliminarea stărilor inaccesibile

① Construiți un ~~NFA~~ DFA pt: $L = \{0^{2k}1^{5l}0^{4j-2} \mid k \geq 5, l, j \geq 1\}$



TRANSFORMĂRI λNFA - NFA - DFA - DFA minim

λNFA - NFA

1. Calcularea λ - închiderii

λ^*	
0	
⋮	

2. Calcularea funcției de tranziție δ^*

	λ^*	a	λ^*

	λ^*	b	λ^*

3. =>

δ^*	a	b

3. Calcularea stărilor finale și inițiale
   - → starea inițială rămâne aceeași
   - → starea finală: stările care conțin starea finală din automatul inițial

4. Eliminarea stărilor redundante
   - → au aceeași caracter de stare finală și au aceleași tranziții
   - → înlocuiesc toate aparițiile lui x cu y

NFA → DFA

1. Pornim o coadă în care adăugăm starea inițială q_0. Pt fiecare stare din coadă și pentru fiecare caracter din alfabet facem calculul:

dacă $\delta(q, \alpha) = q_{x0}, \dots q_{xk}$, $k \geq 0$ atunci:

- creez starea $q_{x0} \dots q_{xk}$
- dacă această stare nu a mai fost vizitată, o adaug în coadă
- repetăm calculul până când coada devine vidă

2. → starea inițială rămâne aceeași
   → starea finală: toate care au în componența o stare din automatul inițial.

3. Redenumim stările