CURS 5 - LFA

- Nu toate limbajele sunt modelate de catre DFA-uri
- Majoritatea limbajelor NU sunt regulate/DFA.
- Automatul finit are memorie finita
- Cum putem arata ca un limbaj este regulat?
- Cum putem arata ca un limbaj nu este regulat?

Limbaje finite

📌 **Teorema:** Orice limbaj finit L este modelat de un DFA

- Demonstratie:
    - pt fiecare cuvantd din L se poate construi cate un automat (linie) cu stari disjuncte
    - se mai defineste o stare, cu λ-mişcări către stările initiale ale automatelor
    - se transforma λ-NFA in NFA si apoi in DFA
- Limbajele infinite pot fi regulate/DFA sau neregulate:
    - ex limbaj infinit regulat : L={0^n | n }

$$L = \{0^n | n \geq 0\}$$

    - ex limbaj infinit neregulat

$$L = \{0^n 1^n | n \geq 0\}$$

📌 **Lema de pompare:**

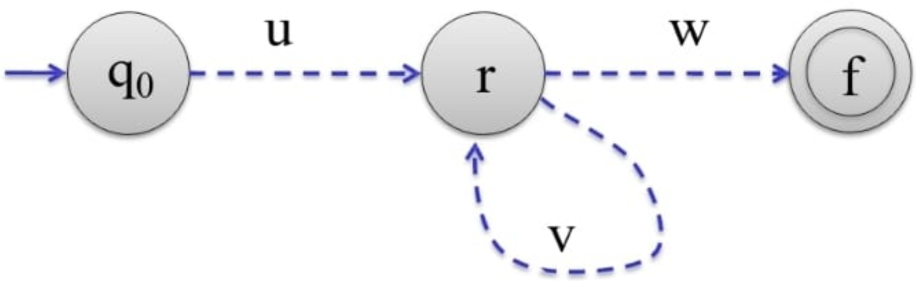
Fie

L=limbaj regulat, exista atunci un numar p, astfel incat orice cuvant $z \in L$ cu proprietatea ca $|w| \geq p$ poate fi descompus in trei parti $z = uvw$ astfel incat:

1.
$|uv| \leq p$

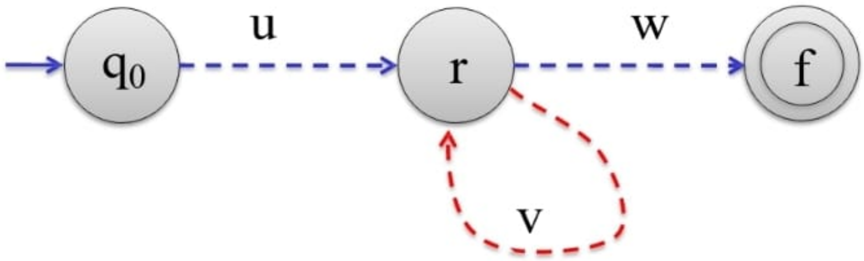
2.
$|v| > 0$

3. pt orice i ≥ 0 avem ca
$uv^iw \in L$



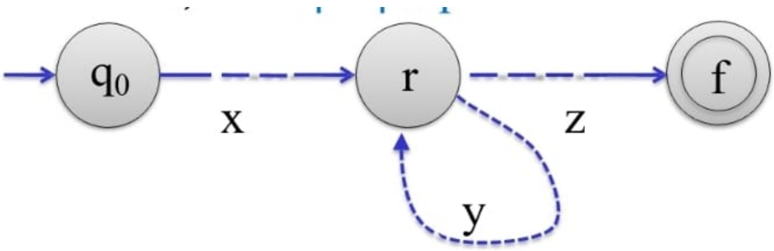
- teorema importanta in zona de automate finite
- orice limbaj regulat/DFA satsiface aceasta lema/proprietate
- daca un limbaj nu are aceasta proprietate ⇒ acel limbaaj NU este regulat
- Ideea de baza:
    - presupunem ca, nr de stari din automat =p
    - orice cuvant w, $|w| \geq p$ trebuie sa aiba un ciclu
    - notam cuvantul procesat prin acest ciclu cu v, cuvantul pana in ciclu cu u si cuvantul dupa ciclu cu w
    - ciclul poate fi repetat de oricate ori si apoi sa mearga cu w catre o stare finala
    - deci partea v poate fi "pompata"
    - asadar mergem prin u pana in starea care incepe ciclul resp.

- apoi mergem de oricate ori prin ciclul v
- in final, mergem cu w catre o stare finala si acceptam cuvantul



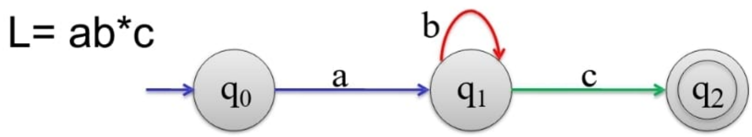
- Descriere informala (ca pt prosti):
    - Pentru fiecare limbaj regulat există un număr/o constantă p (care nu se modifică) pentru care lema de pompare spune că orice cuvânt de lungime mai mare decât această constantă se poate descompune în 3 componente uvw iar partea din mijloc v poate fi pompată
    - În plus partea din mijloc (care e pompată) are cel puțin o literă
    - Prefixul şi/sau sufixul (u sau w) pot fi λ
    - Partea pompată face parte din primele p litere ale cuvântului.
- Demonstratie formala:
    - Fie un automat DFA M automatul care modelează limbajul L. Fie p numărul de stări ale automatului M
    - Fie z un cuvânt din L de lungime mai mare sau egală cu p
    - Considerăm prefixul de lungime exact p litere al lui z şi procesarea prin automatul M a acestui prefix.
    - Plecăm din starea inițială q_0 şi apoi fiecare stare va procesa câte o literă din prefixul de p litere considerat
    - Pentru că considerăm cuvântul de lungime p, vom ajunge în starea q_p , deci trecem prin $p+1$ stări
    - Din principiul cutiei şi pentru că avem p stări în automatul M, dar prefixul de lungime p al cuvântului va fi procesat de $n+1$ stări rezultă că cel puțin o stare apare de două ori

- Notăm o asemenea stare care apare de două ori cu r
- Fie cuvântul u prefixul lui z cu care se ajunge în prima apariție a lui r
- Fie v subșirul din z (care apare imediat după u în z) prin care se ajunge din r în aceeași stare r
- Fie w sufixul lui z pe care îl citim după v. Adică uv concatenat cu w ne dă z
- w ne va duce din starea r intr o stare finală
- u este un cuvânt care ciclează din r în aceeași stare r
- de asemenea v are cel puțin o literă pentru că ne mutăm din stare în stare cu cel puțin o literă; uv fac parte din primele p litere ale cuvântului z, deci $|uv| \leq p$

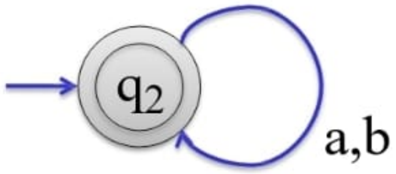


- De vreme ce cuvântul z e acceptat rezultă că există un drum de la starea r la o stare finală f, $q_0 \to r \to f$ asta înseamnă că cuvântul uw nu va merge prin ciclu şi va ajunge la starea finală f ($uv^0w = uw$)
- Pe de altă parte, se poate merge un număr arbitrar de ori prin ciclu şi tot se ajunge în starea finală f: uv^iws
- deci cuvantul z se poate descompune in cele 3 parti u, v, w cu proprietatile lemei de pompare: prop 1 ⇒ am ales prefixul de exact p litere si din el am definit v ca fiind bucata pe ciclul repsectiv; prop 2 si 3⇒ pt ca ciclul are cel putin o litera pt orice i ≥ 0 avem ca prop 3

- Exemple de limbaje:
  - Pt orice cuvant de lungime > 3 din L, putem sa il scriem uvw: $u = a; v = b$ (sau b^k cu $k > 1$) si $w = c$ (sau $w = b^1c$)

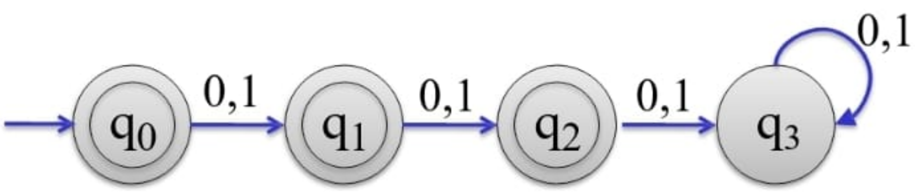


- accepta toate cuvintele peste alfabetul $\sum = \{a, b\}$, deci $L = \sum^*$;
  - in acest caz putem lua u si w cuvantul vid λ



> 📌 **Teorema:**
>
> Daca un DFA A are limbajul acceptat finit si numărul de stări din A este n, atunci lungimea celui mai lung cuvânt din L este mai mică decât n ($|z| < n$)

- Exemplu: A are 4 stări. L(A) conține toate cuvintele de lungime 2 sau mai puțin L(A) este regulat doar ca lema de pompare nu pare sa funcționeze aici (de ce? nu stiu)



- Lema de pompare e utilizata pt a arata ca un limbaj NU este regulat
- Daca vrem sa demonstram ca un limbaj este regulat ⇒ e suficient sa dam un automat pt acel limbaj
- <u>Tehnica de demonstrare a ex cu lema de pompare:</u>
  1. presupunem prin absurd ca limbajul este regulat

2. atunci exista un p fix pentru L si cautam/gasim un cuvant w de lungime mai mare decat p
3. demonstram ca nu exista nicio descompunere a lui w in $w = uvw$ astfel incat avem toate cele 3 prorpietati din L.P. De fapt, prop 1 si 2 ne dau restrictii pt u dar mai ales pt v iar din prop 3 vom avea contradictia

> 📌 Lema: Daca L este regulat atunci $\exists n$, a.i. $\forall\ w \in L$ si $|w| \geq n$ $\exists w = xyz$ a.i. 1+2+3

- regulat(L) ⇒ lema(L)   (reciproca e falsa)
- Greseli comune:
  - $\exists w \in L$ a.i. lema(w) ⇒ regulat(L)
  - $\forall w \in L$, lema(w) ⇒ regulat(L)
  - $\exists w \in L$ si $\exists w = uvw$ a.i. $uv^iw \notin L$ ⇒ ~regulat(L)
  - $\exists w \in L$ si $\forall w = uvw$, $uv^iw \notin L$ ⇒ ~regulat(L)
- Exercitiu:

$$L = \{0^m 1^m | m > 0\}$$

  - Aratam cu lema de pompare ca limbajul nu e regulat
  - presupunem prin absurd ca L este regulat
  - fie nr dat de lema de pompare pt L (este nr de stari din eventualul automat care il accepta)
  - consider $w = 0^p 1^p$
  - w este evident din L, $|w| > p$ asadar din LP w poate fi descompus in cel putin o descompunere $w = uvw$
  - u si v sunt ambele in prefixul de lungime p, deci ammandoua au doar 0-uri in componenta
  - notam $s =$ lungimea lui u si $t =$ lungimea lui v
  - deci:
    - $u = 0^s, s >= 0$

- $v = 0^t, t >= 1, s + t <= n$
- $w = 0^{n-s-t}1^n$

- din LP avem ca uv^iw este in L pt orice i

$$\underbrace{000\ldots000}_{s}\underbrace{00}_{t}0111\ldots111$$

- luam i=2
- cuvantul obtinut $uv^2w = 0^s0^{2t}0^{n-s-t}1^n = 0^{n+1}1^n$
- acest cuvant nu este in L pt t≥1: nr de 0-uri > nr de 1-uri
- am gasit o contradictie cu LP ⇒ nu exista descompuneri ale lui $w = 0^p1^p$ cu proprietatile 1,2,3
- deci L nu e limbaj regulat

Problema din examenul de anul trecut

9. (10 puncte) Spuneţi dacă limbajul următor este sau nu regulat. Dacă limbajul este regulat construiţi un automat finit determinist care să îl accepte, dacă nu, demonstraţi folosind lema de pompare pentru REG că limbajul nu este regulat $L = \{a^kwcw \mid w \in \{a, b, c\}^*, k \geq 4\}$.
ALTERNATIV pentru max 5 puncte: $L = \{a^{k-1}b^{2l+3} \mid k, l \geq 5\}$.

- Deci $L_1=\{a^kwcw| k>3\}$
- $L_2=\{a^{k-1}b^{2l+3}|k,l>4\}$

$L_2=\{a^{k-1}b^{2l+3}|k,l>4\}$

- Se poate construi un DFA pentru
  - $L_3=\{a^k|\ k>3\}$: un ciclu de lungime 3 si de asemenea pentru
  - $L_4=\{b^{2l+3}|\ l>4\}$: un ciclu de lungime 2 precedat de un drum ("o linie") de lungime 3

- Se poate face concatenarea lui L3 cu L4 si acesta este DFA-ul pentru limbajul respectiv

$L_1=\{a^k wcw|\ k>3\}$

- Limbajul nu e REG
- Demonstratie prin lema de pompare
- Alegem cuvantul $a^4b^pa^pcb^pa^p$