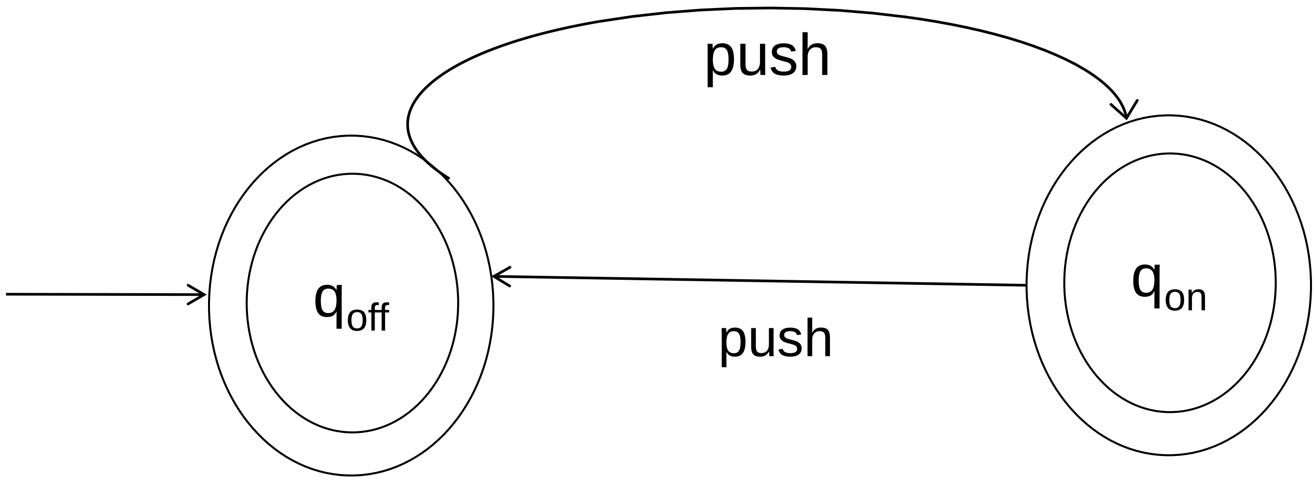
LIMBAJE REGULATE (RECAPITULARE)

1. Exemple practice de automate finite deterministe (AFD);
2. Definitia formala a unui AFD,
3. O metoda de definire a unui AFD care sa recunoasca un limbaj dat
4. Automate finite nedeterministe
   - definitie
   - echivalenta
5. Operatii de inchidere (regulate)
6. Expresii regulate
   - definitie
   - echivalenta
7. Lema de pompare pentru limbaje regulate

LIMBAJE REGULATE (RECAPITULARE)

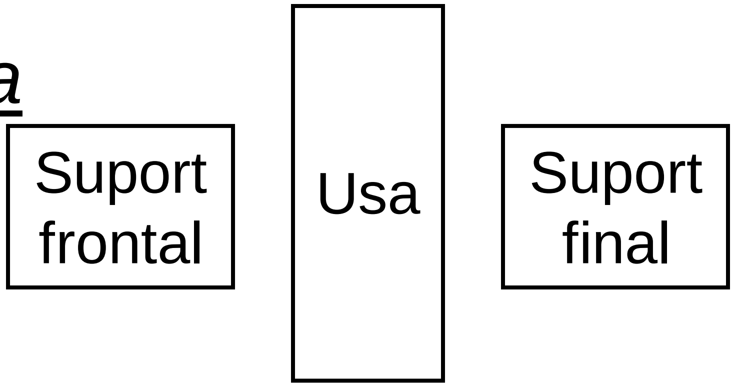
Ex. 1: AFD pt un comutator electric



 2

LIMBAJE REGULATE (RECAPITULARE)

Ex. 2: AFD pt o usa automata



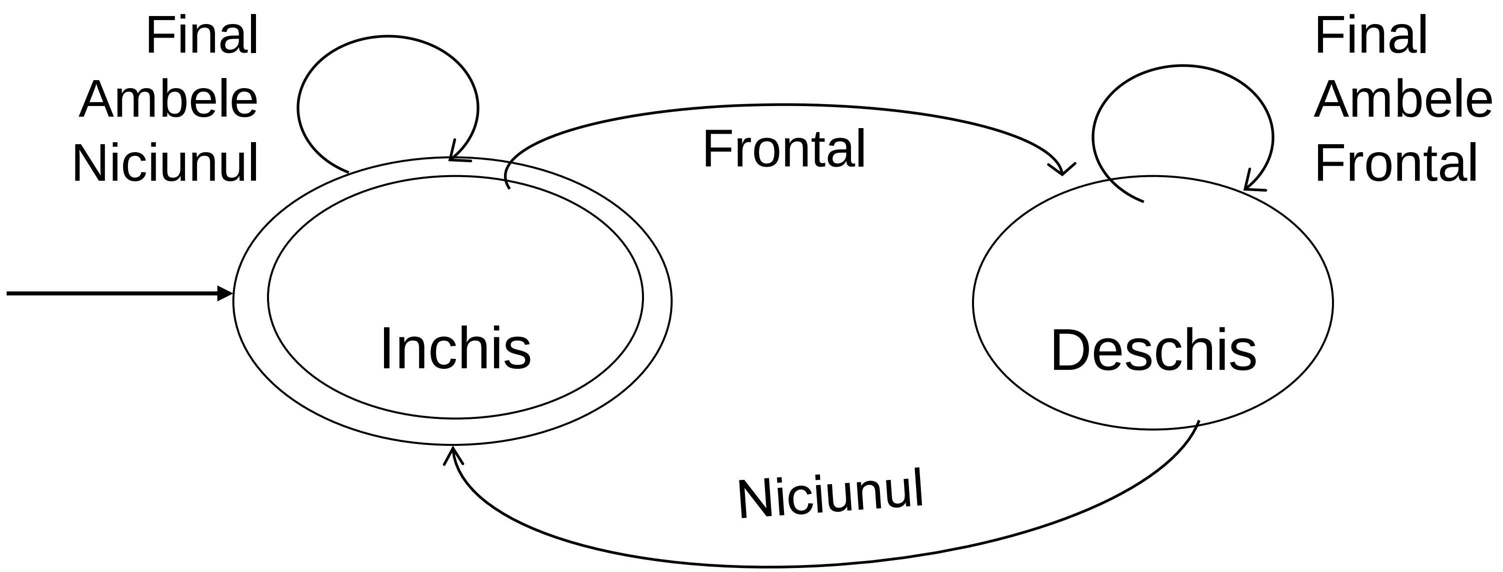
Dam pt acest autoamat cele 3 descrieri posibile

1) Definitia in limbajul natural;
2) Definitia cu ajutorul diagramei de stare;
3) Definitia formala

(1) Definitia in limbajul natural:

LIMBAJE REGULATE (RECAPITULARE)

(2) Definitia cu ajutorul diagramei de stare



LIMBAJE REGULATE (RECAPITULARE)

(3) Definitia formala

	Pe niciun suport	*Pe suportul frontal*	*Pe suportul final*	*Pe ambele suporturi*
Inchis	Inchis	Deschis	Inchis	Inchis
Deschis	Inchis	Deschis	Deschis	Deschis

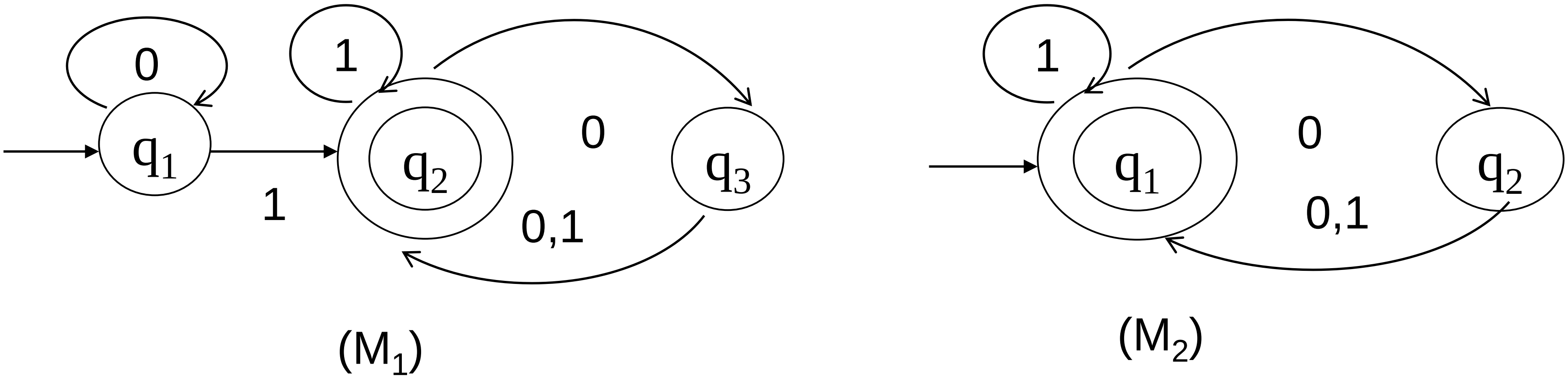
5

LIMBAJE REGULATE (RECAPITULARE)

1. Exemple practice de automate finite deterministe (AFD);
2. Definitia formala a unui AFD,
3. O metoda de definire a unui AFD care sa recunoasca un limbaj dat
4. Automate finite nedeterministe
   - definitie
   - echivalenta
5. Operatii de inchidere (regulate)
6. Expresii regulate
   - definitie
   - Echivalenta

   Lema de pompare pentru limbaje regulate

LIMBAJE REGULATE (RECAPITULARE)



√ 1, 01, 11, 0101, 010100, 01001, ……

⊗ 0, 10, 01010, …..

$L(M_1) = L_1 = \{w \in \{0,1\}^* \mid w=\alpha 1 \text{ sau } w=\alpha 00,\ \alpha \in \{0,1\}^*\}$

$L(M_2) = L_1 \cup \{ \lambda, 00 \}$

LIMBAJE REGULATE (RECAPITULARE)

Definitie

AFD = $(Q, \Sigma, \delta, q_0, F)$, unde:

Q = multime finita, nevida, ale carei elemente se numesc stari;

Σ = multime finita, nevida, numita alfabet de intrare, ale carei elemente se numesc simboluri;

$\delta : Q \times \Sigma \rightarrow Q$, numita functia de tranzitie;

$q_0 \in Q$, numita starea initiala;

$F \subseteq Q$ numita multimea starilor finale.

LIMBAJE REGULATE (RECAPITULARE)

Exemplu

M_1: $Q = \{q_1, q_2, q_3\}$;

$\Sigma = \{0,1\}$;

$q_0 = q_1$;

$F = \{q_2\}$

δ :

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

LIMBAJE REGULATE (RECAPITULARE)

Observatie

L(M) = limbajul **recunoscut** de AFD M.

$F = \emptyset \subseteq Q$

LIMBAJE REGULATE (RECAPITULARE)

1. Exemple practice de automate finite deterministe (AFD);
2. Definitia formala a unui AFD,
3. O metoda de definire a unui AFD care sa recunoasca un limbaj dat
4. Automate finite nedeterministe
   - definitie
   - echivalenta
5. Operatii de inchidere (regulate)
6. Expresii regulate
   - definitie
   - echivalenta
7. Lema de pompare pentru limbaje regulate

LIMBAJE REGULATE (RECAPITULARE)

Ideea metodica a proiectarii unui AFD:

“proiectantul devine un AFD”

Elementul esential in aceasta strategie:

CE INFORMATIE DESPRE FRAZA CITITA TREBUIE MEMORATA DE AFD?

Limbajul: **infinit;** automatul:numar **finit** de stari, deci memorie **finita.**

Care este informatia cruciala?

12

LIMBAJE REGULATE (RECAPITULARE)

Exemplu:

Fie $\Sigma = \{0,1\}$ si $L = \{w \in \{0,1\}^* \mid \#_1(w) = 2k+1,\ k \in N\}$

Pas 1: stabilim informatia de memorat:

- nr de smb 1 citite pana la momentul crt este sau nu impar?
- la citirea unui nou smb:
  - daca acesta este 0 -> raspunsul trebuie lasat neschimbat;
  - daca acesta este 1 -> raspunsul trebuie comutat.

LIMBAJE REGULATE (RECAPITULARE)

Pas 2: reprezentam informatia de memorat ca o lista finita de posibilitati:

- numar par de simboluri 1, pana acum;
- numar impar de simboluri 1, pana acum.

Pas 3: asignam fiecarei posibilitati cate o stare:

- q_{par}
- q_{impar}

LIMBAJE REGULATE (RECAPITULARE)

Pas 4: definim tranzitiile, examinand modul in care se trece de la o posibilitate la alta la citirea fiecarui tip de simbol din Σ:

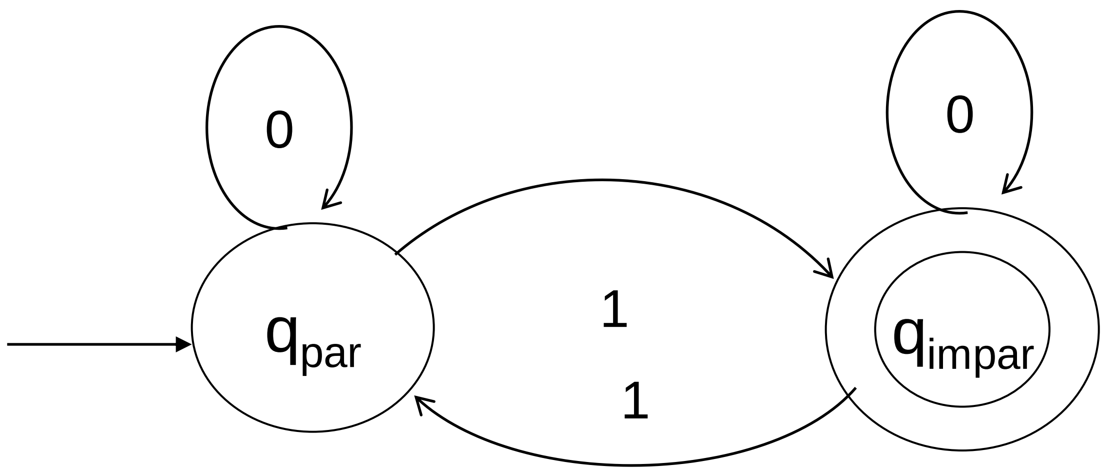
- se trece din orice stare in cealalta la citirea unui simbol 1
- se ramane in aceeasi stare la citirea unui simbol 0.

Pas 5: stabilirea starii initiale si a multimii starilor finale, examinand modul in care se intra/se paraseste fiecare posibilitate:

- initial se citesc 0 simboluri -> AFD porneste din starea q_{par}.
- starea finala trebuie sa fie cea in care acceptam secventa de intrare -> starea finala este q_{impar}.

LIMBAJE REGULATE (RECAPITULARE)

.



LIMBAJE REGULATE (RECAPITULARE)

1. Exemple practice de automate finite deterministe (AFD);
2. Definitia formala a unui AFD,
3. O metoda de definire a unui AFD care sa recunoasca un limbaj dat
4. Automate finite nedeterministe
   - definitie
   - echivalenta
5. Operatii de inchidere (regulate)
6. Expresii regulate
   - definitie
   - echivalenta
7. Lema de pompare pentru limbaje regulate

LIMBAJE REGULATE (RECAPITULARE)

Diferentele dintre un AFD si un AFN sunt:

1) □ $\forall q \in Q$: $\forall a \in \Sigma$:

in AFD pleaca o singura sageata pentru fiecare simbol de intrare,

in AFN pleaca

0,

1, sau

mai multe sageti pentru fiecare smb. de intrare;

2) Sagetile sunt etichetate:

in AFD: cu simboluri din Σ,

in AFN:

cu simboluri din Σ,

cu simboluri din Σ sau cu simbolul vid, λ.

3) Modul de calcul

LIMBAJE REGULATE (RECAPITULARE)

Definitie

AFN = (Q, Σ, δ, q_0, F), unde:

Q = multime finita, nevida, ale carei elemente se numesc stari;

Σ = multime finita, nevida, numita alfabet de intrare, ale carei elemente se numesc simboluri;

$\delta : Q \times (\Sigma \cup \{\lambda\}) \rightarrow \mathcal{P}(Q)$, numita functia de tranzitie;

$q_0 \in Q$, numita starea initiala;

$F \subseteq Q$ numita multimea starilor finale.

Notatie

$\Sigma_\lambda = \Sigma \cup \{\lambda\}$)

LIMBAJE REGULATE (RECAPITULARE)

Exemplu

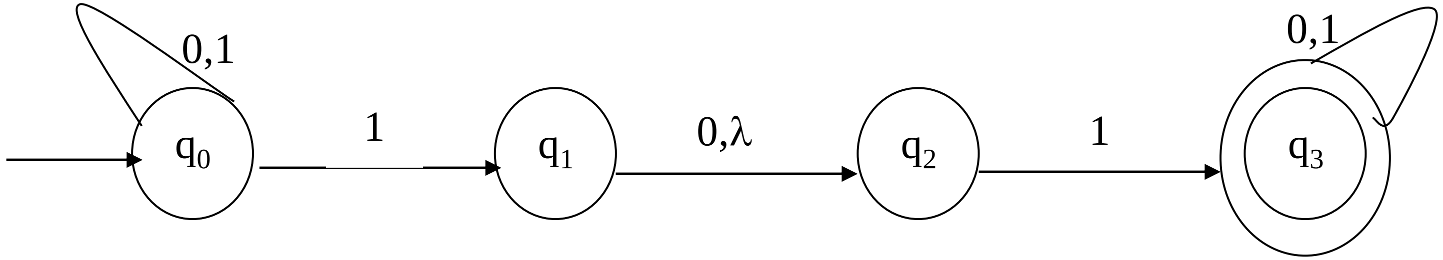
AFN care recunoaste limbajul:

$L_2 = \{ w \in \Sigma_\lambda^* \mid \exists\ u,v \in \Sigma_\lambda^*: w=u101v \text{ sau } w=u11v\}$

$\Rightarrow AFN_1 = (\{q_0,q_1,q_2,q_3\}, \{\lambda,0,1\}, \delta, q_0, \{q_3\})$, unde:

δ	λ	0	1
q_0	Φ	$\{q_0\}$	$\{q_0,q_1\}$
q_1	$\{q_2\}$	$\{q_2\}$	Φ
q_2	Φ	Φ	$\{q_3\}$
q_3	Φ	$\{q_3\}$	$\{q_3\}$

LIMBAJE REGULATE (RECAPITULARE)



21

LIMBAJE REGULATE (RECAPITULARE)

Teorema

AFN $\Leftrightarrow$ AFD

demonstratie

"$\Leftarrow$"

"$\Rightarrow$"

Fie AFN=$(Q, \Sigma, \delta, q_0, F)$; el se poate converti intr-un AFD=$(Q', \Sigma', \delta', q'_0, F')$ astfel:

$Q' = \mathcal{P}(Q)$, $\Sigma' = \Sigma$, $q_0' = \{q_0\}$,

$F' = \{ R \in Q' = \mathcal{P}(Q)$ | R contine cel putin o stare finala a lui AFN $\}$,

$\forall R \in Q'$ si $a \in \Sigma'$: $\delta'(R,a) = \{ q \in Q \mid \exists r \in R: q \in \delta(r,a) \} = \cup_{r \in R} \delta(r,a)$.

LIMBAJE REGULATE (RECAPITULARE)

Daca $\exists$ tranzitii etichetate cu λ, mai definim

Vid(R) = R $\cup$ {q$\in$Q | q poate fi atinsa din R cu ajutorul a 1 sau mai multe tranzitii etichetate cu λ} $\Rightarrow$

$\delta'(R,a) = \{q \in Q \mid \exists r \in R: q \in Vid(\delta(r,a))\} = \bigcup_{r \in R} Vid(\delta(r,a))$

$q_0' = Vid(\{q_0\})$.

<u>Corolar</u>

$\forall\ L \subseteq \Sigma^*,\ L \in \mathcal{L}_3 : \Leftrightarrow \exists$ AFN care recunoaste L.

LIMBAJE REGULATE (RECAPITULARE)

1. Exemple practice de automate finite deterministe (AFD);
2. Definitia formala a unui AFD,
3. O metoda de definire a unui AFD care sa recunoasca un limbaj dat
4. Automate finite nedeterministe
   - ⌘ definitie
   - ⌘ echivalenta
5. Operatii de inchidere (regulate)
6. Expresii regulate
   - ⌘ definitie
   - ⌘ echivalenta

   Lema de pompare pentru limbaje regulate

24

LIMBAJE REGULATE (RECAPITULARE)

<u>Definitie</u>

Operatii de inchidere = operatii regulate =

- reuniunea,
- concatenarea,
- operatia star (*).

$\forall$ A,B $\subseteq \Sigma^*$: $A \cup B = \{ x \in \Sigma^* \mid x \in A \text{ sau } x \in B \}$;

$A \circ B = \{ xy \in \Sigma^* \mid x \in A \text{ si } y \in B \}$;

$A^* = \{ x_1 x_2 \ldots x_k \in \Sigma^* \mid k \in N,\ k \geq 0,\ \forall 1 \leq i \leq k: x_i \in A \}$.

<u>Teorema</u>

$\mathcal{L}_3$ este inchisa la reuniune, concatenare, operatia star.

LIMBAJE REGULATE (RECAPITULARE)

1. Exemple practice de automate finite deterministe (AFD);
2. Definitia formala a unui AFD,
3. O metoda de definire a unui AFD care sa recunoasca un limbaj dat
4. Automate finite nedeterministe
   - ⌘ definitie
   - ⌘ echivalenta
5. Operatii de inchidere (regulate)
6. Expresii regulate
   - ⌘ definitie
   - ⌘ echivalenta
   - 7. Lema de pompare pentru limbaje regulate

26

LIMBAJE REGULATE (RECAPITULARE)

Definitie

Expresie regulata =

o expresie R care satisface una dintre urmatoarele conditii:

1. $\forall$ a $\in \Sigma$ este o expresie regulata (reprezentand limbajul $\{a\} \subseteq \Sigma^*$);
2. λ este o expresie regulata (reprezentand limbajul $\{\lambda\} \subseteq \Sigma^*$);
3. $\emptyset$ este o expresie regulata (reprezentand limbajul vid);
4. daca R_1 si R_2 sunt expresii regulate $\Rightarrow$
   - $(R_1 \cup R_2)$ este o expresie regulata,
   - $(R_1 o R_2)$ este o expresie regulata,
   - (R_1^*) este o expresie regulata.

Exemplu

$(0 \cup 1)0^*$

LIMBAJE REGULATE (RECAPITULARE)

<u>Teorema</u>

$\forall$ L$\subseteq\Sigma^*$, L$\in\mathcal{L}_3$: $\Leftrightarrow$ $\exists$ o expresie regulata R peste Σ care descrie L.

demonstratie

"$\Leftarrow$"

Fie L$\subseteq\Sigma^*$ un limbaj si fie R o expresie regulata peste Σ care descrie L.

Exista un algoritm de convertire a expresiei regulate R intr-un AFN;

Cf. corolarul de mai sus: limbajul recunoscut de un AFN este regulat.

"$\Rightarrow$"

Fie L$\subseteq\Sigma^*$ un limbaj regulat:

Cf. corolar si teorema anterioare: L este recunoscut de un AFD;

Exista un algoritm de convertire a unui AFD intr-o expresie regulata

1. se converteste AFD intr-un AFNG;
2. se converteste AFNG intr-o expresie regulata.

28

LIMBAJE REGULATE (RECAPITULARE)

Definitie

AFNG = automat finit nedeterminist generalizat =

$(Q, \Sigma, \delta, q_{start}, q_{accept})$, unde:

Q = multime finita, nevida, ale carei elemente se numesc stari;

Σ = multime finita, nevida, numita alfabet de intrare, ale carei elemente se numesc simboluri;

$q_{start} \in Q$, numita starea initiala;

$q_{accept} \in Q$, numita starea finala;

$\delta : (Q \setminus \{q_{accept}\}) \times (Q \setminus \{ q_{start}\}) \rightarrow \mathcal{R}$, numita functia de tranzitie.

LIMBAJE REGULATE (RECAPITULARE)

1. Exemple practice de automate finite deterministe (AFD);
2. Definitia formala a unui AFD,
3. O metoda de definire a unui AFD care sa recunoasca un limbaj dat
4. Automate finite nedeterministe
   - ⌘ definitie
   - ⌘ echivalenta
5. Operatii de inchidere (regulate)
6. Expresii regulate
   - ⌘ definitie
   - ⌘ echivalenta
   7. Lema de pompare pentru limbaje regulate

LIMBAJE REGULATE (RECAPITULARE)

<u>Lema de pompare</u>

$\forall$ $L \subseteq \Sigma^*$, $L \in \mathcal{L}_3 \Rightarrow \exists$ $p \in N$ (numit lungimea sau constanta de pompare) a.i. $\forall s \in L$: $|s| \geq p$ atunci $\exists$ $x,y,z \in \Sigma^*$ cu proprietatea ca $s=xyz$ si:

(1) $\forall$ $i \geq 0$: $xy^iz \in L$;

(2) $|y| > 0$;

(3) $|xy| \leq p$.

<u>Observatie</u>

$x=\lambda \vee z=\lambda$ dar, cf. (2), $y \neq \lambda$.

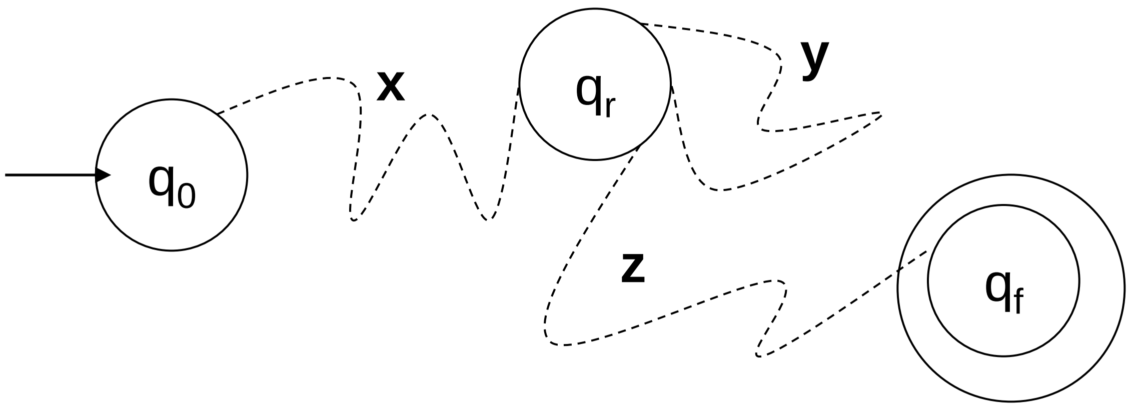
Ideea demonstratiei

$p=|Q|$

$n=|s| \Rightarrow n \geq p \Rightarrow n+1>p=|Q| \Rightarrow$

cel putin 1 repetitie: $q_0 q_i q_k \ldots q_r q_{r+1} \ldots q_r q_t \ldots q_f$

LIMBAJE REGULATE (RECAPITULARE)



Verificam conditiile:

- fie $s=xyyz$; xy^iz, $\forall i \geq 2>0$, $s=xz \Rightarrow$ (1);
- subsecv. y aduce M din q_r inapoi in $q_r \Rightarrow$ (2);
- q_r este prima stare care se repeta iar $n+1>p \Rightarrow$ repetitia apare in una dintre primele p+1 stari din secventa $\Rightarrow$ (3)

32

LIMBAJE REGULATE (RECAPITULARE)

1. Exemple practice de automate finite deterministe (AFD);
2. Definitia formala a unui AFD,
3. O metoda de definire a unui AFD care sa recunoasca un limbaj dat
4. Automate finite nedeterministe
   - definitie
   - echivalenta
5. Operatii de inchidere (regulate)
6. Expresii regulate
   - definitie
   - echivalenta
7. Lema de pompare pentru limbaje regulate

33