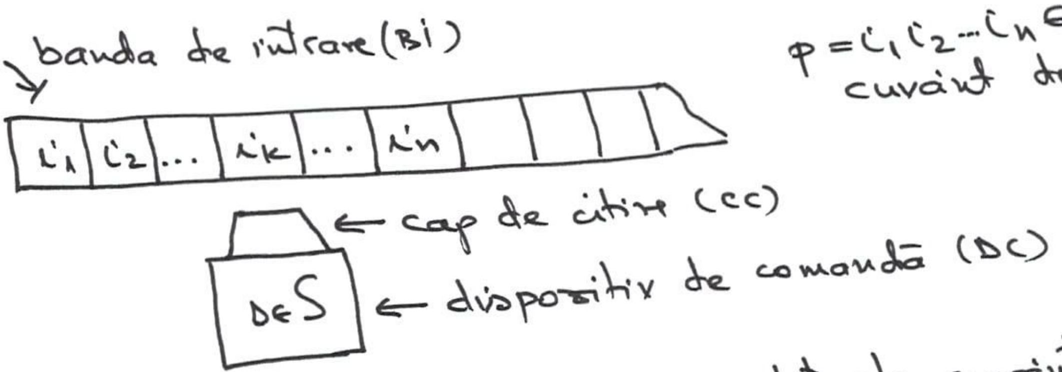
AUTOMATE FINITE

Mecanisme (matematice) pentru recunoașterea lb. regulate.

Schema mecanismului



$p = i_1 i_2 \ldots i_n \in I^*$
cuvânt de examinat

Bi – șir de celule în care se înscriu literele cuvântului examinat $p = i_1 \ldots i_n \in I^*$. I – alfabet de bandă

DC – are un CC aflat în dreptul unei celule (ce conține i_k)
– conține o "stare internă" $s \in S$. S – alfabetul stărilor

FUNCȚIONARE : șir de pași (funcționare "discretă")

Descriere pas DC aflat în starea s citește litera i_k din dreptul CC și în funcție de cele 2 elemente schimbă starea în $s' \in S$ ($f(s, i_k) \ni s'$) și deplasează Bi o poziție stânga.

Recunoașterea cuvintelor

Inițial DC se află într-o stare specială $s_0 \in S$ numita stare inițială, CC în dreptul primei celule a Bi, cuvântul de analizat $p = i_1 \ldots i_n \in I^*$ inscris pe Bi. începând cu prima celulă

Examinarea cuvântului revine la un șir de pași în care se schimbă starea la fiecare pas în fc. de starea curentă și litera citită.

După citirea ultimei litere se examinează starea DC (numită "stare reziduală"). Spunem că p este recunoscut dacă starea reziduală face parte dintr-o submulțime specială de stări S_f numita $S_f \subseteq S$ mulțime de stări finale

Exemplu.

AF pentru recunoașterea cuvintelor peste $I=\{a,b\}$ ce conțin subcuvântul bb

Descrierea unui pas

f	a	b
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_2	s_2

Stare inițială s_0

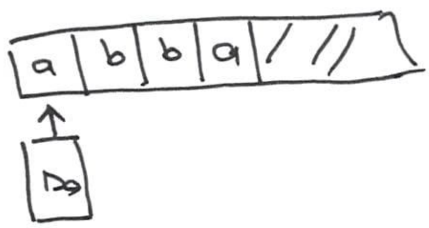
Stări finale $S_f=\{s_2\}$

$I=\{a,b\}$

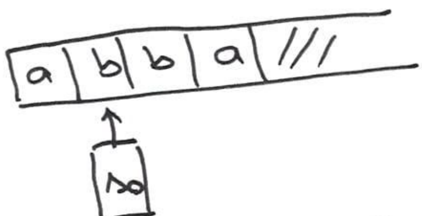
$S=\{s_0,s_1,s_2\}$

Examinarea cuvântului $p=abba$

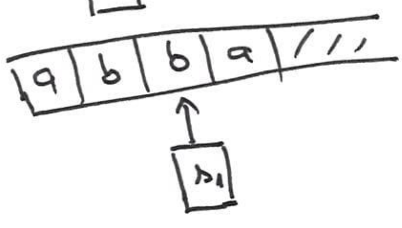
inițial:



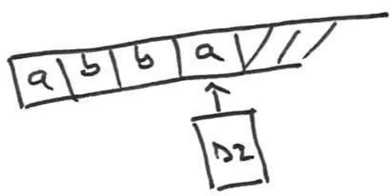
pas 1:



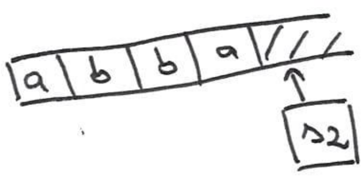
pas 2:



pas 3:



pas 4:



$s_2 \in S_f \Rightarrow p=abba$ „recunoscut"

DEFINIȚIA FORMALĂ

$AF = (S, I, f, s_0, S_f)$

S - alfabetul stărilor

I - alfabetul de bandă (alfabetul intrărilor)

f - funcție de evoluție, descrie un pas elementar

$f : S \times I \to \mathcal{P}(S)$

$s_0 \in S$ - starea inițială

$S_f \subseteq S$ - stări finale

Obs: 1) Dacă $f(s,i) = \phi$ ptr. $s \in S$ și $i \in I$ spunem că AF se "blochează", adică orice continuare a examinării nu conduce la recunoaștere

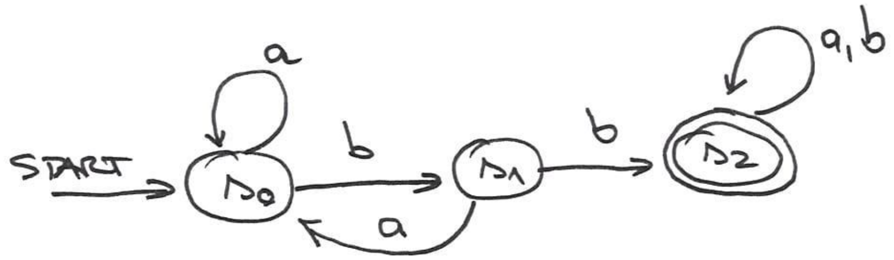
2) Dacă $|f(s,i)| \leq 1$ $\forall s \in S, i \in I$ spunem că AF are funcționare deterministă sau este determinist (notam AFD). In caz contrar AF este nedeterminist

3) Reprezentarea grafică a funcției de evoluție f se face printr-un graf orientat ale cărui noduri sunt etichetate cu elementele lui S, iar arcele construite astfel: nodurile s, s' se unesc cu un arc etichetat cu i dacă există $i \in I$ astfel încât $s' \in f(s,i)$.

Nodul s_0 are un arc etichetat cu START.

Nodurile $s \in S_f$ sunt notate cu cercuri concentrice

PTR. EXEMPLUL PRECEDENT



Convenție: Dacă $f(s,i) = \phi$ atunci arcul nu se reprezintă

LIMBAJUL RECUNOSCUT DE AUTOMATE FINITE

Se extinde funcționarea de la un pas la un șir de pași prin prelungirea funcției $f: S \times I \to \mathcal{P}(S)$ la $f': \mathcal{P}(S) \times I^* \to \mathcal{P}(S)$ astfel:

a) $f'(s, \lambda) = \{s\}$, $\forall s \in S$

b) $f'(\phi, i) = \phi$, $\forall i \in I$

c) $f'(Z, i) = \bigcup_{s \in Z} f(s, i)$, $\forall Z \in \mathcal{P}(S)$, $Z \neq \phi$, $i \in I$

d) $f'(Z, pi) = f'(f'(Z, p), i)$, $\forall Z \in \mathcal{P}(S)$, $p \in I^*$, $i \in I$

Obs: 1) Prin abuz de limbaj notăm extinderea f' tot cu f.

2) Relațiile a), b), c), d) constituie o definiție prin recurență corectă matematic. Rel. d) precizează că evaluarea unui cuvânt se face de la stânga la dreapta.

PROPRIETĂȚI ALE FUNCȚIEI DE EVOLUȚIE

1. Dacă Z_k este o familie de părți ale lui S atunci $f(\bigcup_k Z_k, i) = \bigcup_k f(Z_k, i)$

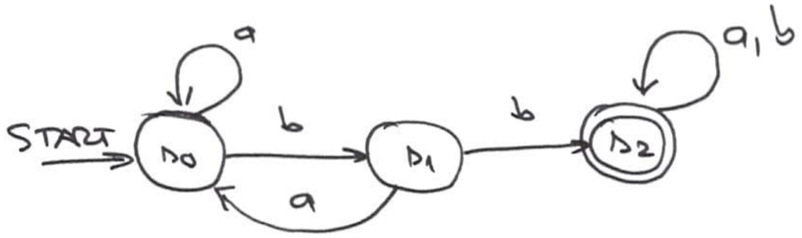
2. $f(Z, p) = \bigcup_{s \in Z} f(s, p)$ $\forall Z \in \mathcal{P}(S)$, $p \in I^*$

3. $f(s, pq) = f(f(s, p), q)$, $\forall s \in S$, $p, q \in I^*$

LIMBAJ RECUNOSCUT

$$L(AF) = \{p \in I^* \mid f(s_0, p) \cap S_f \neq \phi\}$$

EXEMPLUL 1 AFD



analiza unui cuvânt $p = abba$

Teoretic cu definiția formală

Calculăm $f(s_0, abba) \overset{d)}{\underset{3.}{=}} f(f(s_0, abb), a) =$

$\overset{d)}{\underset{3.}{=}} f(f(f(s_0, ab), b), a) \overset{d)}{\underset{3.}{=}}$

$\overset{d)}{\underset{3.}{=}} f(f(f(f(s_0, a), b), b), a) =$

$= f(f(f(\{s_0\}, b), b), a) =$

$\overset{c)}{=} f(f(f(s_0, b), b), a) =$

$= f(f(\{s_1\}, b), a) =$

$\overset{c)}{=} f(f(s_1, b), a) =$

$= f(\{s_2\}, a) \overset{c)}{=} f(s_2, a) = \{s_2\}$

Deci $f(s_0, p) = \{s_2\}$

Dar $f(s_0, p) \cap S_f \neq \emptyset \Rightarrow p \in L(AF)$

Obs: Dacă $f(s, i) = \{s'\}$ atunci prin abuz de limbaj scriem $f(s, i) = s'$ (cazul AFD)

URMĂRIREA EVOLUȚIEI AF LA EXAMINARE

$\xrightarrow{START} s_0 \xrightarrow{a} s_0 \xrightarrow{b} s_1 \xrightarrow{b} s_2 \xrightarrow{a} s_2$

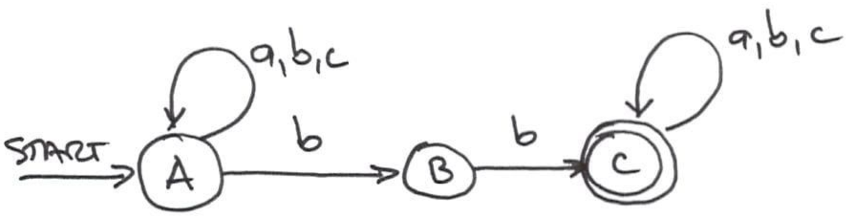
Sau

$s_0 \xrightarrow{a} s_0 \xrightarrow{b} s_1 \xrightarrow{b} s_2 \xrightarrow{a} s_2 \in S_f$ TRAIECTORIE

Obs: Cuvânt recunoscut <=> avem o traiectorie de la s_0 la o stare finală $s' \in S_f$ etichetată cu literele cuvântului examinat

$p' = bab$ $\quad s_0 \xrightarrow{b} s_1 \xrightarrow{a} s_0 \xrightarrow{b} s_1 \notin S_f \Rightarrow p' \notin L(AFD)$

EXEMPLUL 2 AFN



Obs: 1) $f(A,b) = \{A,B\}$ deci AFN nedeterminist
2) $f(B,a) = \phi$ – arcul cu originea B și eticheta a lipsă

ANALIZA CUVINTELOR $p = abb$

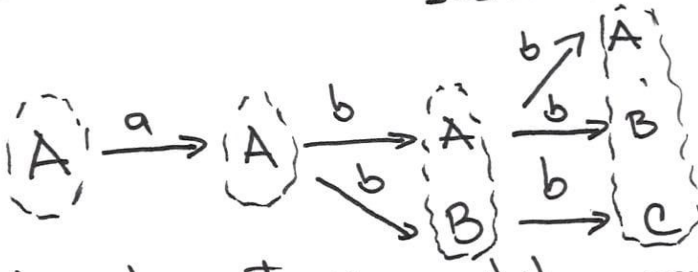
$$f(A,abb) \overset{3.}{\underset{d)}{=}} f(\underline{f(A,ab)},b) \overset{3.}{\underset{d)}{=}} f(f(\underline{f(A,a)},b),b) =$$

$$= f(\underline{f(A,b)},b) = f(\{A,B\},b) \overset{c)}{=}$$

$$\overset{c)}{=} f(A,b) \cup f(B,b) = \{A,B\} \cup \{C\} = \{A,B,C\}$$

$$f(A,abb) = \{A,B,C\} \cap \mathcal{F} \neq \phi \Rightarrow p = abb \in L(AF)$$

ANALIZA GRAFICĂ La fiecare pas sunt marcate submulțimile de stări posibile



Sunt 3 traiectorii posibile pentru analiza cuvântului $p = abb$. Printre ele există una ce conduce la stare finală deci p este recunoscut

p cuvânt recunoscut $\Leftrightarrow$ există o traiectorie etichetată cu literele lui p, de la starea inițială la o stare finală

Evident $L(AF) = \{ w \in \{a,b,c\}^* \mid w$ conține subcuvântul bb $\}$

Def: AF echivalente $\Leftrightarrow$ recunosc același limbaj

Obs:

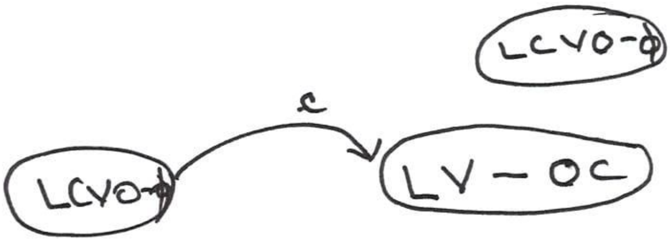
Exemple interesante de aplicații ale AF

1. Rezolvarea problemei de la grădiniță:

"Pe malul unui râu se află un om, un lup, o capră și o varză. Omul are o barcă în care poate intra doar el singur, sau însoțit de un singur obiect (lup, capră, varză) și trebuie să treacă pe celălalt mal fără să dispară un obiect atunci când face o singură traversare. Se presupune că în lipsa omului lupul mănâncă capra, capra mănâncă varza."

Vom construi un AF pentru rezolvarea problemei și identificăm toate soluțiile posibile (o infinitate).

FORMAL definim stările posibile ale sistemului prin precizarea poziției obiectelor

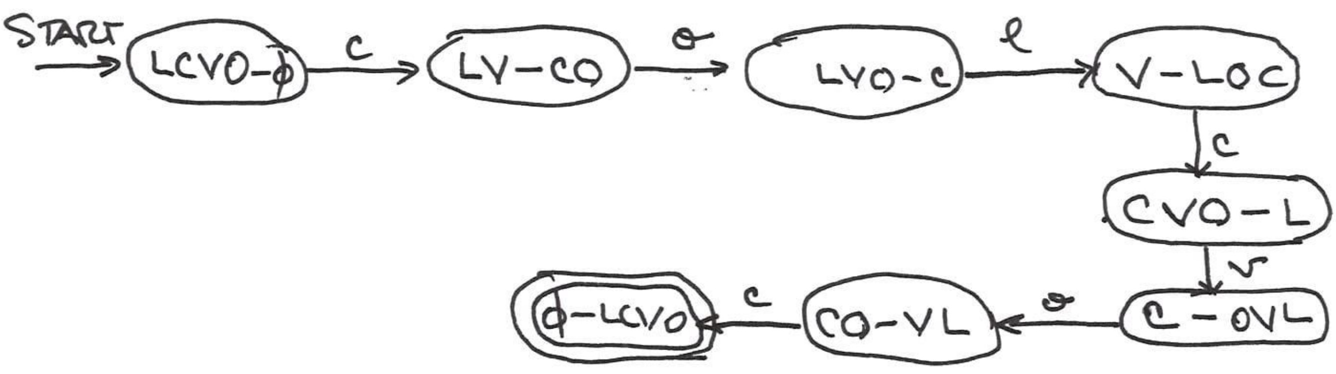


înseamnă că Lupul, Capra, Varza și Omul sunt pe malul stâng și nimic pe malul drept.

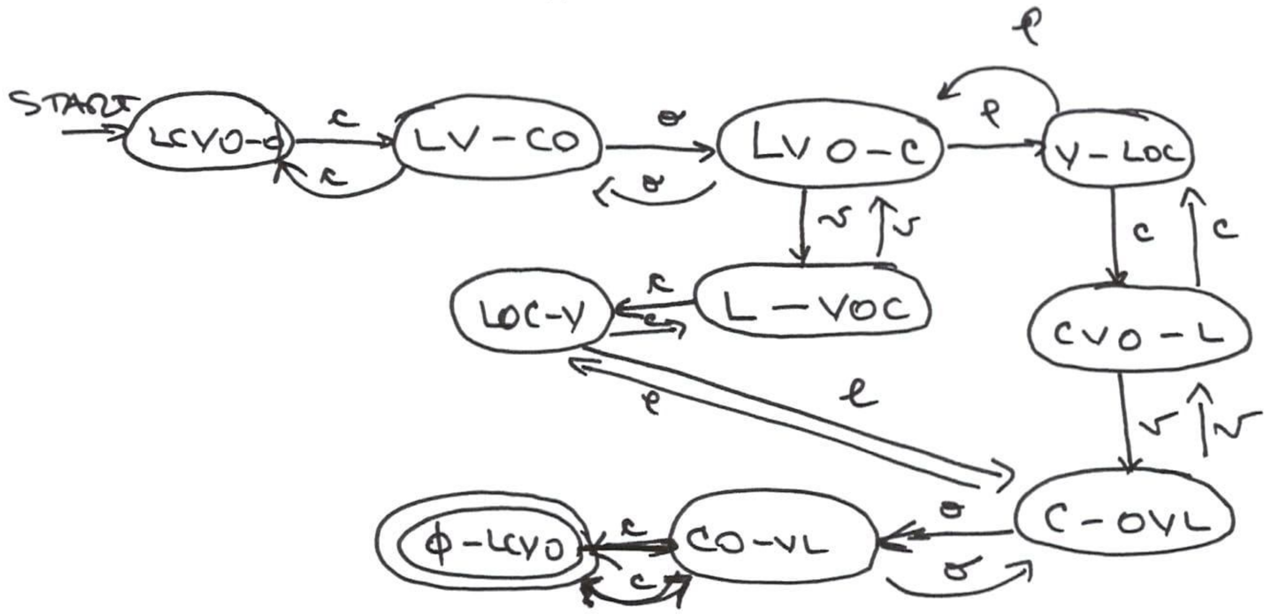
Acțiunile ce provoacă schimbarea stării sunt marcate cu arce: Omul trece obiectul Capra pe malul drept. Analog, eticheta v - trece varza, l - trece lupul, o - trece singur de pe un mal pe altul.

Construcția soluției cunoscute la grădiniță

$p = colcvoc \in L(AF)$ <u>soluție</u>



<u>Completarea automatului</u>



$p = colcvoc \in L(AF)$

$p_1 = covcloc \in L(AF)$

$p_2 = colcvlcvlcvoc \in L(AF)$

$p \in L(AF)$ – p soluție a problemei

Obs: $|L(AF)| = \infty$ => o infinitate de soluții