TRANSFORMĂRI $G \to AF$, $AF \to G$
EXEMPLE APLICARE LEMA DE POMPARE.

SE CONSIDERĂ O GRAMATICĂ DE TIPUL 3 și SE CERE UN AF ECHIVALENT.

ETAPA I NORMALIZAREA GRAMATICII

$$G : \begin{cases} A \to pB \\ C \to q \end{cases} , \; p,q \in V_T^* , \; C,A,B \in V_N$$

$$\downarrow$$

$$G' : \begin{cases} A \to iB \\ C \to j \end{cases} , \; i,j \in V_T , \; C,A,B \in V_N$$

Se admite regula de completare (pt generarea λ) $S \to \lambda$ și $S \notin dr$

Algoritm:
1) Elimin reguli de ștergere
2) Elimin redenumiri "$A \to B$"
3) Scurtez reguli neconvenabile

$$A \to i_1 i_2 \dots i_k B \longrightarrow \begin{cases} A \to i_1 Z_1 \\ Z_1 \to i_2 Z_2 \\ \vdots \\ Z_{k-1} \to i_k B \end{cases}$$

$$C \to j_1 \dots j_k \longrightarrow \begin{cases} C \to j_1 Z_1 \\ \vdots \\ Z_{k-1} \to j_k \end{cases}$$

ETAPA II CONSTRUCȚIA AF ("citim" diagrama din regulile G)

$$AF = (V_N \cup \{X\}, V_T, f, x_0, S_f)$$

$$S_f = \begin{cases} \{X\}, \text{ dacă } \lambda \notin L(G) \\ \{X, x_0\}, \text{ dacă } \lambda \in L(G) \end{cases}$$

Dacă $A \to iB \in P$ atunci $f(A,i) \ni B$

Dacă $C \to j$ atunci $f(C,j) \ni X$

Obs: Se folosește transformarea la compunerea AF folosind teoremele de închidere!

Exemplul 1: Se consideră limbajul

$$L = \{a^{2n} \mid n \geq 0\}$$

1) Găsiți o gramatică regulată G ce generează L
2) Construiți un AFD echivalent

Rezolvare

1) Rescrie $L = \{a^{2n} \mid n \geq 0\} = \{(aa)^n \mid n \geq 0\}$

Evident o gram. ce generează L este (rețeta 2)

$G: \quad S \to aaS \mid \lambda$

G nu este în formă normală (S apare în dreapta și am reguli prea lungi)

a) Elimin regulile de ștergere în G.

$G_1: S \to aaS \mid aa \qquad L(G_1) = L \setminus \{\lambda\}$

adaug regula de completare

$G_2: \begin{cases} A \to \lambda \mid S \\ S \to aaS \mid aa \end{cases}$

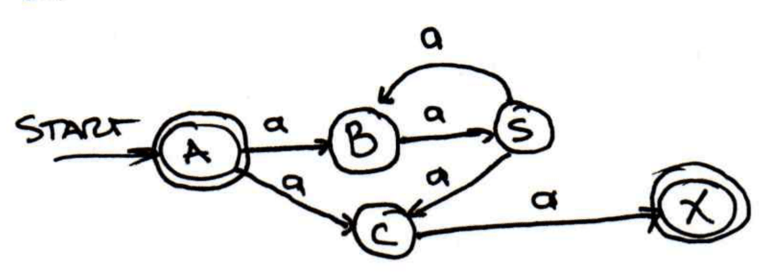
b) Elimin redenumire $A \to S$

$G_3: \begin{cases} A \to \lambda \mid aaS \mid aa \\ S \to aaS \mid aa \end{cases}$

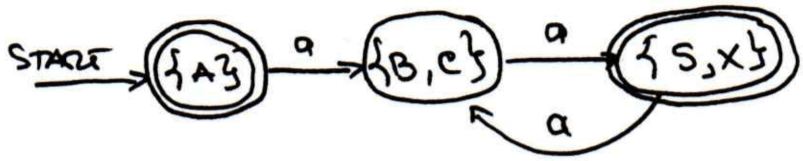
c) Scurtez reguli

$G_4: \begin{cases} A \to \lambda \mid aB \mid aC \\ B \to aS \\ C \to a \\ S \to aB \mid aC \end{cases}$ (refolosesc simbolurile!)

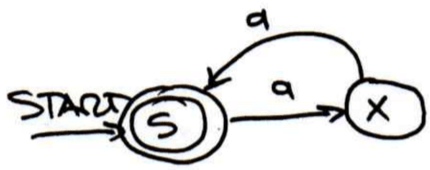
2) Desenăm AF (cu A stare finală fiindcă $\lambda \in L$)



Construim AFD echivalent



<u>Obs</u>: Un AFD minimal (cu nr. minim de stări)



Exemple de transformare G → ~~AF~~ pot fi considerate pentru gramaticile regulate "ghicite" la seminarul în care s-au asociat gramatici regulate pornind cu un "șablon" asociat cuvintelor unui limbaj.

Temă: Din exercițiile de la sfârșitul CAP. 1. pag 22-23

ex. 3 a, b, c, e, f, g, i, j

ex 1 f, g, d

"$AF \to G$"

Se consideră un AF (nu neapărat determinist) ptr recunoașterea unui limbaj L.

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

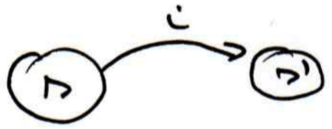
$$\Downarrow$$

$$G = (S, I, s_0, P)$$

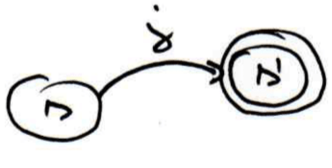
Dacă $f(s, i) \ni s'$ atunci "citesc" o regulă $s \to is' \in P$

Dacă $f(s, i) \ni s'$ și $s' \in S_f$ atunci "citesc" două reguli $\begin{cases} s \to is' \\ s \to i \end{cases}$

<u>Grafic</u> – ptr. fiecare arc din diagramă se generează o regulă (dacă ținta nu e stare finală) sau două reguli (dacă ținta este stare finală)



$s \to is'$



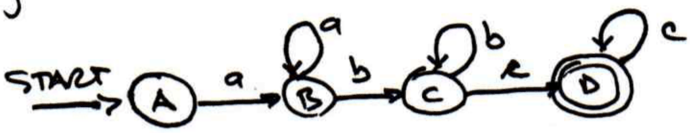
$\begin{cases} s \to js' \\ s \to j \end{cases}$

<u>Obs</u>: 1.) Cazul $\lambda \in L$ se tratează separat.

Dacă $s_0 \in S_f$ atunci adaug regula $s_0 \to \lambda \in P$

2). De obicei se obține o gramatică în formă normală. Excepție cazul $\lambda \in L$.

EXEMPLUL 2: Găsiți o gramatică ce generează limbajul recunoscut de AFD



Rezolvare

$f(A,a)=B \Rightarrow A \to aB \in P$

$f(B,a)=B \Rightarrow B \to aB \in P$

$f(B,b)=C \Rightarrow B \to bC \in P$

$f(C,b)=C \Rightarrow C \to bC \in P$

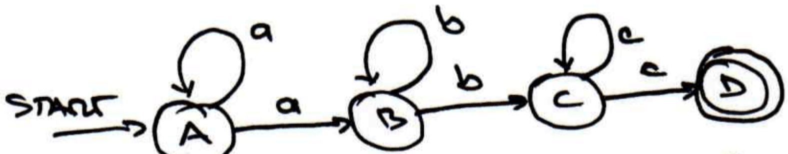
$f(C,c)=D \Rightarrow \left\{ \begin{array}{l} C \to cD \\ C \to c \end{array} \right. \in P$

$f(D,c)=D \Rightarrow D \to cD \mid c \in P$

Sau direct

$$G: \left| \begin{array}{l} A \to aB \\ B \to aB \mid bC \\ C \to bC \mid cD \mid c \\ D \to cD \mid c \end{array} \right.$$

Obs: Limbajul recunoscut este $L(AFD)=\{a^n b^m c^k \mid m,n,k \geq 1\}$ poate fi recunoscut de urm. AFN



Citirea gramaticii se ~~poate~~ face pornind cu AFN

$$G': \left\{ \begin{array}{l} A \to aA \mid aB \\ B \to bB \mid bC \\ C \to cC \mid c\underline{D} \mid c \end{array} \right.$$

Se observă imediat că pt această gramatică regula $C \to cD$ este 'neutilizabilă' pt. producerea unui cuvânt deoarece D nu dispare!

Exemplul 3: Se consideră limbajul

$$L = \{ w \in \{0,1\}^* \mid w \equiv 1 \pmod 3 \}$$

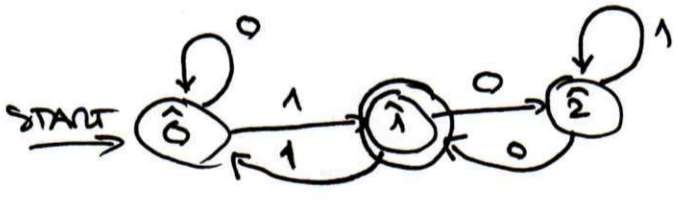
Construiți o gramatică regulată ce generează L.

Rezolvare

Se poate refolosi ideea de la recunoașterea multiplilor lui 3 scriși în (binar) formatul zecimal exemplul 5 din seminarul precedent.

Adaptarea automatului revine la folosirea cifrelor 0 și 1, respectiv observația că $\lambda \notin L$. Aceasta permite folosirea a 3 stări $\hat{0}, \hat{1}, \hat{2}$

Diagrama AF este



Se identifică o gramatică echivalentă (folosind redenumiri ale stărilor)

$$G = \begin{cases} A \to 0A \mid 1B \mid 1 \\ B \to 1A \mid 0C \\ C \to 1C \mid 0B \mid 0 \end{cases}$$

Exemplul 4: Se consideră limbajul

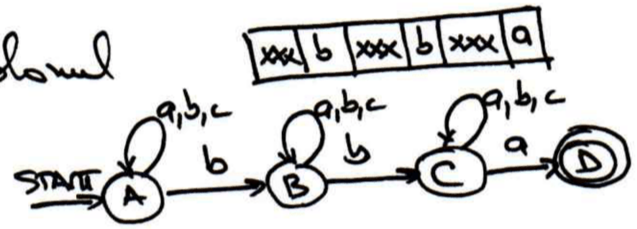
$L = \{ w \in \{a,b,c\}^* \mid w$ conține cel puțin două simboluri b și se termină cu $a \}$

a) Construiți un AFD ce recunoaște L

b) Folosind teorema de echivalență $\mathcal{L}_3 = \mathcal{R}$ găsiți o gramatică regulată ce generează L

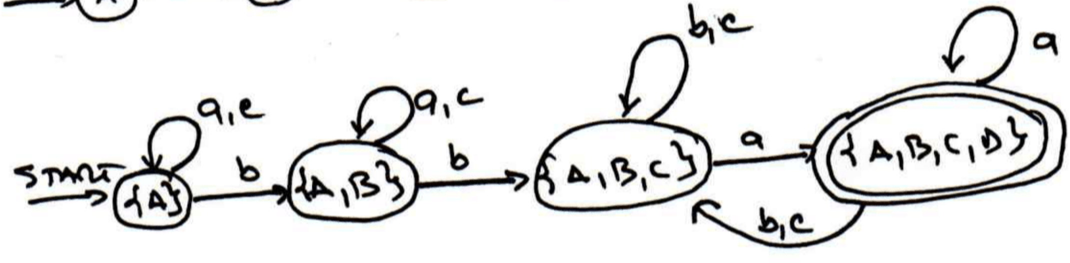
Rezolvare

a) Sablonul

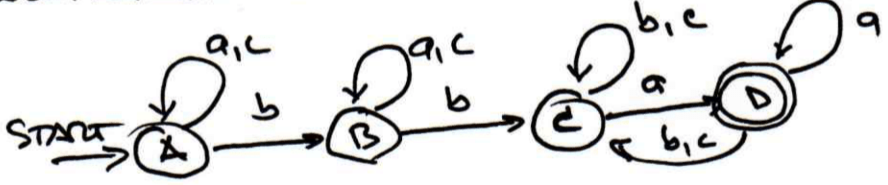


AF =

AFD



redenumim starile



b) Citirea se poate face de pe AF sau AFD

$$G_{AF}: \begin{cases} A \to aA \mid bA \mid cA \mid bB \\ B \to aB \mid bB \mid cB \mid bC \\ C \to aC \mid bC \mid cC \mid aD \mid a \end{cases}$$

$$G_{AFD}: \begin{cases} A \to aA \mid cA \mid bB \\ B \to aB \mid cB \mid bC \\ C \to bC \mid cC \mid aD \mid a \\ D \to cC \mid bC \mid aD \mid a \end{cases}$$

Evident

$L(G_{AF}) = L(G_{AFD}) = L$

Exerciții: Construiți gramatici ptr. generarea limbajelor din seminarul precedent

LEMA DE POMPARE

$\forall L \in \mathcal{L}_3 \ \exists n \in \mathbb{N}$ a.î. $\forall p \in L$ cu $|p| \geq n$ $\exists$ partiționarea

$p = uvw$ cu proprietățile

1) $v \neq \lambda$

2) Cuvântul nou format $p' = uv^k w \in L, \forall k \in \mathbb{N}$

Rezolvare tip: Demonstrați că $L \notin \mathcal{L}_3$.

Pres. prin reducere la absurd că $L \in \mathcal{R}$.

" Din l.p rezultă că există un nr. natural n și p/r. orice cuvânt $p \in L$ suficient de lung ($|p| \geq n$) există o partiționare $p = uvw$ cu $v \neq \lambda$ astfel încât cuvântul obținut prin pomparea lui v de oricâte ori $p' = uv^k w \in L$."

Sau enunț l.p.

Contrazicem l.p. considerând o partiționare arbitrară a cuvântului $p \in L$ și identificând un nr n_0 de repetări astfel încât $p' = uv^{n_0} w \notin L$

Obs: – de obicei cazurile de partiționare se referă la ce tipuri de simboluri pot extrage din p, iar $n_0 = 0, 2$

Exemplu: $L = \{ a^n b^{n+1} \mid n \geq 0 \}$.

Pres....

Fie: $p = a^n b^{n+1} = \underbrace{a \dots a}_{n \text{ ori}} \underbrace{b \dots b}_{n+1 \text{ ori}}$ suficient de mare

Caz I: v conține doar simboluri b.

partiționarea este $p = \underbrace{a \dots a\, b \dots b}_{u} \underbrace{b \dots b}_{v} \underbrace{b \dots b}_{w}$

Alep $n_0 = 0$. $p' = uv^0 w = \underbrace{a \dots a\, b \dots b}_{u} \underbrace{b \dots b}_{w} = a^n b^{n+1-|v|} \notin L$ contrad.

Caz II: v conține a și b

partiționarea $p = \underbrace{a \dots a}_{u} \underbrace{a \dots a\, b \dots b}_{v} \underbrace{b \dots b}_{w}$

Alep $n_0 = 2$. $p' = uv^2 w = \underbrace{a \dots a}_{u} \underbrace{a \dots a\, b \dots b}_{v} \underbrace{a \dots a\, b \dots b}_{v} \underbrace{b \dots b}_{w} \notin L$ contrad.

<u>Exemplu</u> <u>dificil</u>: $L = \{a^n b^m a^n \mid m, n \geq 1\} \notin \mathcal{R}$

#

nu pot aplica direct rezolvarea tip. (Vezi curs)

Pres. prin reducere la absurd că $L \in \mathcal{R}$.

Fie $AFD = (S, \Sigma, f, s_0, S_f)$ astfel încât $L(AFD) = L$.

Considerăm un cuvânt $p \in L$ "suficient de lung"

$p = a^n b^m a^n$, cu $n > |S|$.

Detaliem traiectoria de recunoaștere pt. p.

$$\underbrace{s_0 \xrightarrow{a} s_1 \xrightarrow{a} \dots \xrightarrow{a} s_n} \xrightarrow{b} s_{n+1} \rightarrow \dots \xrightarrow{b} s_{n+m} \xrightarrow{a} s_{n+m+1} \rightarrow \dots \xrightarrow{a} s_{n+m+n} \in S_f$$

In prima parte a traiectoriei avem $n+1$ stări (până la primul b citit) deci $n > |S| \Rightarrow$ sunt cel puțin două stări cu aceiași etichetă X. Fie $0 \leq i < j \leq n$ pozițiile în care apare X.

$$s_0 \xrightarrow{a} s_1 \rightarrow \dots \rightarrow s_i = X \xrightarrow{a} \dots \xrightarrow{a} s_j = X \xrightarrow{a} \dots s_n \xrightarrow{b} \dots \xrightarrow{a} s_{n+m+n} \in S_f$$

Construim următoarea traiectorie de recunoaștere a unui cuvânt (prin omiterea secvenței între stările s_i și s_j)

$$s_0 \xrightarrow{a} s_1 \rightarrow \dots \rightarrow s_i = X = s_j \xrightarrow{a} \dots \xrightarrow{a} s_n \xrightarrow{b} \dots \xrightarrow{a} s_{n+m+n} \in S_f.$$

Cuvântul recunoscut este (citim pe traiectorie!)

$$p' = a^{n-(j-i)} b^m a^n$$

Dar $p' \notin L$. Contradicție cu ipoteza $L(AFD) = L$.

Conform principiului reducerii la absurd, ipoteza $L \in \mathcal{R}$ este falsă, adică $L \notin \mathcal{R}$.

<u>Obs</u>: Desigur $L \in \mathcal{L}_2$. O gramatică convenabilă este

$$\begin{cases} S \rightarrow aSa \mid aXa \\ X \rightarrow bX \mid b \end{cases} \in \mathcal{G}_2.$$