

**RAPORT**

Lucrare de laborator Nr.3

*La Limbaje formale și compilatoare*

Tema: *Forma Normală Chomsky (FNC)*

**A efectuat : st.gr. MI-202 Marga Dorina**

**A verificat: dr. conf. Univ. Marusic Galina**

**Chișinău 2021**

Sarcina lucrării:

1. Eliminaţi ε producţii
2. Eliminaţi redenumirile
3. Eliminaţi simbolurile neproductive
4. Eliminaţi simbolurile inaccesibile
5. Aduceţi la Forma Normală Chomsky

**Varianta 7**

G=(VN, VT, P, S) VN={S, A, B, C, E} VT={a, b}

P={1. S→bA

2. S→B

3. A→a

4. A→aS

5. A→bAaAb

6. B→AC

7. B→bS

8. B→aAa

9. C→ε

10. C→AB

11. E→BA

**1. Eliminăm ε producţii:**

a) 

b) pentru producţia C→ ε 

P'={1. S→bA

2. S→B

3. A→a

4. A→aS

5. A→bAaAb

6. B→AC 11. B→A

7. B→bS

8. B→aAa

9. C→AB

10. E→BA}

**2. Eliminăm redenumirile:**

***O producţie de forma X*→*Y, X şi Y neterminale, se numeşte redenumire.***

Redenumirile din P' sunt: S→B, B→A

pentru S→B

pentru B→A

P''={ 1. S→bA

2. A→a 10. B→a

3. A→aS 11. B→aS

4. A→bAaAb 12. B→ bAaAb

 5. B→AC 13. S→AC

6. B→bS 14. S→bS

7. B→aAa 15. S→aAa

8. C→AB

9.E→BA

**3. Eliminăm simbolurile neproductive:**





VN={S, A, B, C, E}



P'''= P''={ 1. S→bA

2. A→a 10. B→a

3. A→aS 11. B→aS

4. A→bAaAb 12. B→ bAaAb

 5. B→AC 13. S→AC

6. B→bS 14. S→bS

7. B→aAa 15. S→aAa

8. C→AB

9.E→BA

**4. Eliminăm simbolurile inaccesibile:**

Iniţia*l *

**

**

**

PIV= {

1. S→bA

2. A→a

3. A→aS

 4. A→bAaAb

5. S→AC

6. S→bS

7. S→aAa}

1. **Aducem la Forma Normală Chomsky**

***O gramatică în forma normală Chomsky este o gramatică cu reguli de forma A*→** ***BC, D*→*i, unde A,B,C,DVN  şi i VT***

PV= {

1. S→XA 8. X→b

2. A→a

3. A→YS 9. Y→a

 4. A→XZ1 10. Z1→AZ2 11. Z2→YZ3 12. Z3→AX

5. S→AC

6. S→YS

7. S→YZ4 13. Z4→AY

*Listingul programului:*

*#include <stdio.h>*

*#include <stdlib.h>*

*#include <string.h>*

*#include <windows.h>*

*#include <conio.h>*

*int i,j;*

*void afisare\_gramatica (char p[7][15])*

*{*

*printf("G=(VN, VT, P, S) \n");*

*printf("VN={S, A, B, C, E} VT={a, b}\n");*

*for(j=0;j<11;j++)*

*{*

*for(i=0;i<7;i++)*

*{*

*if(p[i][j]!='\_')*

*{*

*printf("%c",p[i][j]);*

*}*

*}*

*printf("\n");*

*}*

*}*

*void eliminare\_epsilon (char p[7][15])*

*{*

*int k=0,n=0;*

*char Nepsilon[n];*

*for(j=0;j<11;j++)*

*{*

*if(p[3][j]=='e')*

*{*

*Nepsilon[k++]=p[0][j];*

*n++;*

*}*

*}*

*Nepsilon[n]='\0';*

*printf("Nepsilon= {%s}",Nepsilon);*

*printf("\n");*

*}*

*void eliminare\_redenumiri(char p[7][15])*

*{*

*int k=0,n=0;*

*printf("Redenumirile:\n");*

*for(j=0;j<15;j++)*

*{*

*if((p[3][j]=='B'&& p[4][j]=='\_')||(p[3][j]=='A'&& p[4][j]=='\_'));*

*{*

*for(i=0;i<7;i++)*

*{*

*if(p[i][j]!='\_')*

*printf("%c",p[i][j]);*

*}*

*printf("\n");*

*}*

*}*

*}*

*void eliminare\_inaccesibile(char p[9][15])*

*{*

*printf("Simbolurile inaccesibile= B, E\n");*

*printf("Productiile''' ={\n");*

*for(j=0;j<15;j++)*

*{*

*for(i=0;i<9;i++)*

*{*

*if(j!=5 && j!=6 && j!=7 && j!=10 && j!=11 && j!=12 && j!=13 && p[i][j]!='\_')*

*{*

*printf("%c",p[i][j]);*

*}*

*}*

*printf("\n");*

*}*

*printf("B->aB\nS->bAa\nB->bAa\nS->aS\nB->aS\nS->a\nB->a\n");*

*}*

*void FNC()*

*{*

*printf("Productiile de forma FNC = {\n");*

*printf("S->XB\nA->a\nA->YS\nA->XZ1\nS->AC\nS->YS\nS->YZ4\nX->b\n");*

*printf("Y->a\nZ1->AZ2\nZ2->YZ3\nZ3->YZ3\nZ3->AX\nZ4->AY\n");*

*}*

*int main()*

*{*

*system("color 5");*

*int temp, k=0;*

*char p[7][15];*

*p[0][0]='S'; p[0][1]='S'; p[0][2]='A'; p[0][3]='A'; p[0][4]='A'; p[0][5]='B'; p[0][6]='B'; p[0][7]='B'; p[0][8]='C'; p[0][9]='C'; p[0][10]='E'; p[0][11]='B'; p[0][12]='B'; p[0][13]='B';p[0][14]='S';p[0][15]='S';p[0][16]='S';*

*p[1][0]='-'; p[1][1]='-'; p[1][2]='-'; p[1][3]='-'; p[1][4]='-'; p[1][5]='-'; p[1][6]='-'; p[1][7]='-'; p[1][8]='-'; p[1][9]='-'; p[1][10]='-'; p[1][11]= '-';p[1][12]='-';p[1][13]='-';p[1][14]='-';p[1][15]='-';p[1][16]='-';*

*p[2][0]='>'; p[2][1]='>'; p[2][2]='>'; p[2][3]='>'; p[2][4]='>'; p[2][5]='>'; p[2][6]='>'; p[2][7]='>'; p[2][8]='>'; p[2][9]='>'; p[2][10]='>'; p[2][11]='>';p[2][12]='>';p[2][13]='>';p[2][14]='>';p[2][15]='>';p[2][16]='>';*

*p[3][0]='b'; p[3][1]='B'; p[3][2]='a'; p[3][3]='a'; p[3][4]='b'; p[3][5]='A'; p[3][6]='b'; p[3][7]='a'; p[3][8]='e'; p[3][9]='A'; p[3][10]='B'; p[3][11]='A';p[3][12]='a';p[3][13]='b';p[3][14]='A';p[3][15]='b';p[3][16]='a';*

*p[4][0]='A'; p[4][1]='\_'; p[4][2]='\_'; p[4][3]='S'; p[4][4]='A'; p[4][5]='C'; p[4][6]='S'; p[4][7]='A'; p[4][8]='\_'; p[4][9]='B'; p[4][10]='A'; p[4][11]='\_';p[4][12]='S';p[4][13]='A';p[4][14]='C';p[4][15]='S';p[4][16]='A';*

*p[5][0]='\_'; p[5][1]='\_'; p[5][2]='\_'; p[5][3]='\_'; p[5][4]='a'; p[5][5]='\_'; p[5][6]='\_'; p[5][7]='a'; p[5][8]='\_'; p[5][9]='\_'; p[5][10]='\_'; p[5][11]='\_';p[5][12]='\_';p[5][13]='a';p[5][14]='\_';p[5][15]='\_';p[5][16]='a';*

*p[6][0]='\_'; p[6][1]='\_'; p[6][2]='\_'; p[6][3]='\_'; p[6][4]='A'; p[6][5]='\_'; p[6][6]='\_'; p[6][7]='\_'; p[6][8]='\_'; p[6][9]='\_'; p[6][10]='\_'; p[6][11]='\_';p[6][12]='\_';p[6][13]='A';p[6][14]='\_';p[6][15]='\_';p[6][16]='\_';*

*p[7][0]='\_'; p[7][1]='\_'; p[7][2]='\_'; p[7][3]='\_'; p[7][4]='b'; p[7][5]='\_'; p[7][6]='\_'; p[7][7]='\_'; p[7][8]='\_'; p[7][9]='\_'; p[7][10]='\_'; p[7][11]='\_';p[7][12]='\_';p[7][13]='b';p[7][14]='\_';p[7][15]='\_';p[7][16]='\_';*

*p[8][0]='\0'; p[8][1]='\0';p[8][2]='\0';p[8][3]='\0';p[8][4]='\0';p[8][5]='\0'; p[8][6]='\0';p[8][7]='\0';p[8][8]='\0';p[8][9]='\0';p[8][10]='\0'; p[8][11]='\0';p[8][12]='\0';p[8][13]='\0';p[8][14]='\0';p[8][15]='\0';p[8][16]='\0';*

*afisare\_gramatica(p);*

*printf("Tastati ENTER pentru a afisa urmatoarea etapa!!\n ");*

*getch();*

*eliminare\_epsilon(p);*

*printf("Productiile dupa eliminarea epsilon={\n");*

*for(j=0;j<11;j++)*

*{*

*if(p[3][j]=='e')*

*{*

*temp=j;*

*}*

*}*

*for(j=0;j<11;j++)*

*{*

*for(i=0;i<8;i++)*

*{*

*if(j!=temp&&p[i][j]!='\_')*

*{*

*printf("%c",p[i][j]);*

*}*

*}*

*printf("\n");*

*}*

*printf("Tastati ENTER pentru a afisa urmatoarea etapa!!\n ");*

*getch();*

*printf("\n\n\\t\t Epsilon productii eliminte\nProductiile' = \n{");*

*for(j=0;j<11;j++)*

*{*

*for(i=0;i<8;i++)*

*{*

*if(j!= 8 && p[i][j]!='\_')*

*{*

*printf("%c",p[i][j]);*

*}*

*}*

*printf("\n");*

*}*

*getch();*

*printf("Tastati ENTER pentru a afisa urmatoarea etapa!!\n ");*

*eliminare\_redenumiri(p);*

*printf("\n\t\t\tEliminare redenumiri\t\n Productiile'' = \n{");*

*for(j=0;j<11;j++)*

*{*

*for(i=0;i<8;i++)*

*{*

*if(j!= 2 && j!=12 && p[i][j]!='\_')*

*{*

*printf("%c",p[i][j]);*

*}*

*}*

*printf("\n");*

*}*

*printf("\n\tS->aB\nA->a\nA->aS\nA->bAaAb\nB->AC\nB->bS\nB->aAa\nC->AB\nE->BA\nB->a\nB->aS\nB->bAaAb\nS->AC\nA->bS\nS->aAa\n");*

*getch();*

*printf("Nu exista simnoluri neproductive!\n");*

*printf("Tastati ENTER pentru a afisa urmatoarea etapa!!\n ");*

*getch();*

*eliminare\_inaccesibile(p);*

*system("pause");*

*system("cls");*

*FNC(); getch();*

*return 0;*

*}*

**Concluzii:**

Realizând această lucrare de laborator, am reușit să acumulez cunoștințe despre generarea unui șir, despre componetele unui automat finit, ele fiind: alfabetul terminal, alfabetul neterminal, starea inițială, mulțimea stărilor finale. Cu ajutorul acestora s-au realizat producții, care, prin urmare, am generat cuvinte aceptate de limbaj. Acestor șiruri am construit arborii de derivare și am reprezentat sub forma unui graf automatul. În cele din urmă, am constatat că această gramatică aparține tipului 3, după Chomsky, întrcât toate producțiile sunt de forma, o literă mare derivă în una mică și mare sau doar în una mică. La sfarșitul realizării acestor sarcini am scris un program în C, care a afișat la ecran șirurile generate, arborii de derivare și automatul finit.