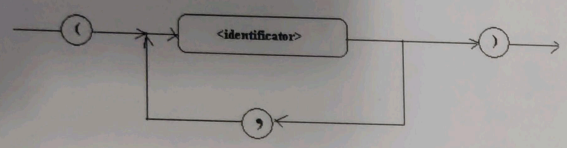
Seminar 1

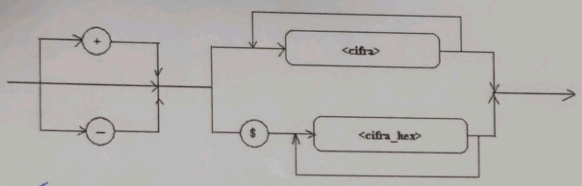
Minilimbaje de programare. Identificarea si specificarea elementelor lexicale si sintactice.

1. Diagrame de sintaxa

Diagramele de sintaxa sunt reprezentari grafice folosite pentru a descrie sintaxa limbajelor de programare.

1.1. Dati cate 2 exemple valide care respecta urmatoarele specificatii.





1.2. Descrieti, folosind diagrame de sintaxa, sintaxa sectiunii de declaratii de variabile in Pascal. "Neterminalele" (vor aparea in dreptunghiuri cu colturile rotunjite si) vor fi: <ID> si <type> .

2. BNF si EBNF

2.1. Dati o descriere echivalenta in BNF si EBNF corespunzatoare diagramelor din sectiunea precedenta.

3. Elemente lexicale si sintactice ale unui limbaj de programare

3.1. Fie o gramatica ce descrie sintaxa unui mini-limbaj de programare, data prin regulile de productie:

```
<program>      → begin <lista_instr> end .
<lista_instr>  → <instr> ; <lista_instr>
<lista_instr>  → <instr>
<instr>        → <atribuire>
<instr>        → <instr_if>
<atribuire>    → ID = <expr>
<expr>         → <expr> + <variabila>
<expr>         → <variabila>
<variabila>    → ID
<instr_if>     → if ( <expr> ) then <atribuire>
```

— neterminale
restul terminale

a) Identificati terminalele si neterminalele gramaticii
b) Dati doua "mini-programe" care sunt descrise de specificatiile date.
ID este un atom lexical. Regulile de formare a acestui atom lexical ID (identificator), specificate folosind expresii regulare, sunt:

a (a | b | c) *

3.2. Fie urmatorul exemplu de program Pascal:

```
var f, a1, a2, a3 : integer;
begin
        a1:= 7;
        a2:= 11;
        a3:= a1+a2+3;
        f := 5
end.
```

a) Identificati elementele lexicale si structurile sintactice.
b) Descrieti sintaxa structurilor sintactice folosind unul dintre mecanismele de specificare: BNF, EBNF sau gramatica independenta de context.
c) Scrieti un program diferit de cel de mai sus care respecta descrierile date.

d) Presupunand ca operatorii si cuvintele cheie din exemplul de mai sus au asociate coduri – numere naturale in ordine crescatoare,
in ordinea in care ele apar in program,
descrieti continutul tabelei FIP, precum si a tabelei de simboluri, atunci cand se folosesc 2 tabele de simboluri, una pentru constante, una pentru identificatori, pentru fiecare dintre urmatoarele 3 organizari:

i. Tabel sortat lexicographic
ii. Arbore binar de cautare

iii. **Tabela de dispersie**
(Alegeti o functie de dipersie simplu de calculat, alegeti dimensiunea tabelei ca fiind 11.)

Poate aveti nevoie:
ASC('a') = 97
ASC('0') = 48

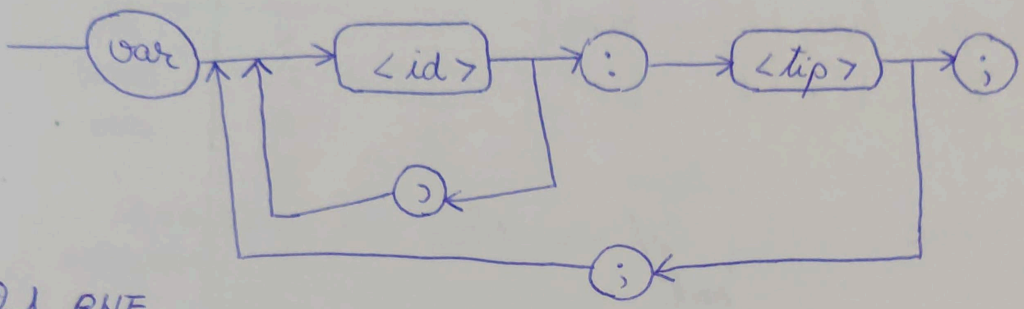
Seminar 1

1.1 (a, b) (abc)

+$A0 999

1.2 var a, b, c : integer;

X : real;



2.1 BNF

<program> ::= (<lista_id>)

<lista_id> ::= <id> | <id>, <lista_id>

<id> ::= <litera> | <litera><lista_caractere>

<litera> ::= a|b|...|z

<lista_caractere> ....

EBHF

program = "(" lista_id ")".

lista_id = id {"," id}.

id = litera {lista_caractere}.

...

BNF

<program> ::= + <numar> | - <numar> | <numar>

<numar> ::= <numar_diamal> | <numar_hex>

<numar_diamal> ::= <cifra> | <cifra><cifre>

<cifre> ::= <cifra> | <cifra><cifre>

<cifra> ::= 0 |1|... |9

<numar_hex> ::= $<cifra_hex> | $<cifra_hex><cifra_hex>

<cifre_hex> ::= <cifra_hex> | <cifra_hex><cifre_hex>

<cifra_hex> ::= 0|1|...|9|A|...|F

EBNF

program = [semn] numar .
semn = "+" | "-" .
numar = numar_decimal | numar_hex .
numar_decimal = cifra {cifra} .
cifra = "0" | "1" | ... | "9" .
numar_hex = "$" cifra_hexa {cifra_hexa} .
cifra_hexa = "0" | ... | "9" | "A" | ... | "F" .

3.1 b) begin
aa = ac
end.

begin
if ac then aa = ac + ab
end.

3.2 a) - identificatori: a1, a2, a3, f
- constante: 3, 5, 7, 11
- cuvinte cheie (rezervate): "var", "integer", "end", "begin"
- operatori: ":=", "+"
- separatori: " ", ":", "␣", ";", "/n/r", "."

b) <program> → <bloc_decl> <bloc_instr> .
<bloc_decl> → var <lista_decl>
<lista_decl> → <decl_var> ; <decl_var> <lista_decl>
<lista_decl> → <decl_var>
<decl_var> → <lista_var> : integer;

<lista_var> → <id>
<lista_var> → <id>, <lista_var>
<bloc_instr> → begin <lista_instr> end
<lista_instr> → <instr> ; <lista_instr>
<lista_instr> → <instr>
<instr> → <atribuire>
<atribuire> → id := <expr>
<expr> → id
<expr> → const
<expr> → <expr> + <expr>

c) var a1 : integer;
a2 : integer;
begin
a1 := a2 + 4 + 5
end.

d)

Atom lexical	Cod atom
id	0
const	1
var	2
,	3
:	4
integer	5
;	6
begin	7
:=	8
+	9
end	10
.	11

=>

FIP

Cod atom	Poz. TS
2	
1 0	4, 1, 3
3	
a1 0	1, 2, 9
3	
a2 0	2, 3, 10
3	
a3 0	3, 4, 0
4	
5	
6	
7	
a1 0	1, 2, 9
8	
4 1	3, 1, 7
6	
a2 0	2, 3, 10
8	
11 1	4, 2, 0
6	
a3 0	3, 4, 0
8	
a1 0	1, 2, 9
9	
a2 0	2, 3, 10
9	
3 1	1, 3, 3
6	
f 0	4, 1, 3
8	
5 1	2, 4, 5
10	
11	

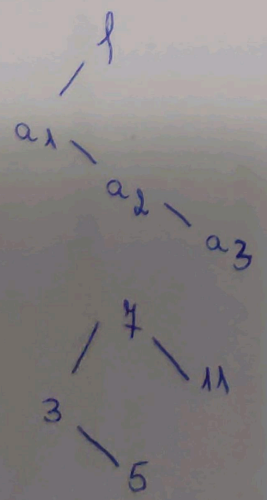
i) TS id

Simbol	
1	a1
2	a2
3	a3
4	f

TS Const

Simbol	
1	3
2	5
3	4
4	11

ii)



=>

TS id

	Simbol	St.	Dr.
1	f	2	-1
2	a1	-1	3
3	a2	-1	4
4	a3	-1	-1

=>

TS const

	Simbol	St.	Dr.
1	4	3	2
2	11	-1	-1
3	3	-1	4
4	5	-1	-1

iii) $hash(x) = asc(x) \% 11$

=> TS id

	symbol
0	a_3
1	
2	
3	f
4	
5	
6	
7	
8	
9	a_1
10	a_2

TS const

	symbol
0	11
1	
2	
3	3
4	
5	5
6	
7	4
8	
9	
10	

Seminar 2

Limbaje. Specificari. Gramatici independente de context simple

1. Multimi si limbaje

Se cere sa se defineasca (folosind multimi si descrierea proprietatilor specifice ale elementelor) urmatoarele limbaje. Se poate folosi concatenare, operatia * - inchiderea reflexiv tranzitiva.

A. limbajul numerelor naturale in reprezentare binara
B. limbajul numerelor intregi in reprezentare binara
C. limbajul numerelor reale pozitive in reprezentare binara
D. limbajul numerelor naturale in reprezentare zecimala
E. limbajul numerelor intregi in reprezentare zecimala
F. limbajul numerelor reale pozitive in reprezentare zecimala

Ex:
A: $L_A = \{1w \mid w \in \{0, 1\}^*\} \cup \{0\}$

2. Gramatici independente de context simple

1. Descrieti limbajul generat de urmatoarele gramatici:

a) G=(N, Σ, S, P)
N = {A,B}
Σ= { a, b}
S = A
P: A -> a B
A -> B
B -> b

b) G=(N, Σ, S, P)
N = { <propozitie>, <subiect>, <predicat-nominal>, <verb-copulativ>, <nume-predicativ>, <substantiv>, <adjectiv>, <verb>, <determinant>}
Σ = {o, orice, functie, derivabila, continua, este}
S = <propozitie>
P: <propozitie> -> <subiect> <predicat-nominal> 1
<subiect> -> <determinant> <substantiv> 2
<predicat-nominal> -> <verb-copulativ> <nume-predicativ> 3
<verb-copulativ> -> este 4
<nume-predicativ> -> <adjectiv> 5
<adjectiv> -> derivabila | continua 6
<substantiv> -> functie 7
<determinant> -> o | orice 8

2. Dati cate o gramatica care genereaza propozitiile:
a) ab, ac
b) abc

3. BNF si EBNF

1. Dati o descriere echivalenta in BNF si EBNF pentru doua dintre limbajele definite in sectiunile precedente.

4. Descrieri de limbaje folosind mecanisme generative

1. Fie L un limbaj peste alfabetul {a,b} definit dupa cum urmeaza:
(i) ab ∈ L
(ii) Daca x ∈ L atunci axb ∈ L
(iii) Niciun alt cuvant nu apartine lui L .
a) Descrieti limbajul definit mai sus folosind multimi si descrierea proprietatilor specifice ale elementelor.
b) Descrieti limbajul definit mai sus folosind o gramatica independenta de context

5. Gramatici independente de context si limbajul generat

1. Sa se construiasca o gramatica care genereaza limbajul:

$L = \{x^n y^n \mid n \in \mathbf{N}\}$

Pentru gramatica construita, demonstrati ca $L(G) = L$.

2. Analog pt. $L = \{a^{2n}bc \mid n \in \mathbf{N}\}$

3. Analog pt. $L = \{a^{2n+1} \mid n \in \mathbf{N}\}$

Seminar 2

$\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$
* cu ε (0...n)
\+ fără ε (1...n)

1. A: $L_A = \{1w \mid w \in \{0,1\}^*\} \cup \{0\}$

B: $L_B = \{s1w \mid s \in \{+,-,\varepsilon\}, w \in \{0,1\}^*\} \cup \{0\}$

~~C: $L_C = \{1w \mid w \in \{0,1\}^*\} \cup \{0\}$~~

~~D $L_D = \{mc \mid c \in \{0,1,\dots,9\}, m \in \{c$~~

C: $L_C = L_A \cup \{1w, v \mid w \in \{0,1\}^*, v \in \{0,1\}^+\} \cup \{0, w \mid w \in \{0,1\}^+\}$

D: $L_D = \{uv \mid u \in \{1,2,\dots,9\}, v \in \{,1,2,\dots,9\}^*\} \cup \{0\}$

E: $L_E = \{suv \mid s \in \{-,+,\varepsilon\}, u \in \{1,2,..,9\}, v \in \{0,1,\dots,9\}^*\} \cup \{0\}$ ε

F: $L_F = L_D \cup \{w\,u,v \mid w \in \{1,..,9\}, u \in \{0,1,\dots,9\}, v \in \{0,1,..,9\}^+\} \cup \{0,u \mid u \in \{0,1,\dots,9\}^+\}$

2.

1a) $G = (N, \Sigma, S, P)$

$N = \{A, B\}$ - neterminale

$\Sigma = \{a, b\}$ - terminale

$S = A$ - ce începi "producția"; simbol start

P: $A \xrightarrow{1} aB$ - mulțimea producțiilor

$A \xrightarrow{2} B$

$B \xrightarrow{3} b$

$L(G) = \{ab, b\} = \{wb \mid w \in \{a, \varepsilon\}\}$

$A \overset{1}{\Rightarrow} aB \overset{3}{\Rightarrow} ab$

$A \overset{2}{\Rightarrow} B \overset{3}{\Rightarrow} b$

nu se înlocuiesc mai multe.! 5 neterminale deodată!

b) ~~<propoziție> $\xrightarrow{1}$ <subiect><PN>~~

$\langle P \rangle \overset{1}{\Rightarrow} \langle S \rangle \langle PN \rangle \overset{2}{\underset{3}{\Rightarrow}} \langle D \rangle \langle Subst \rangle \langle VC \rangle \langle NP \rangle \overset{5}{\Rightarrow}$

$\Rightarrow \langle D \rangle \langle Subst \rangle \langle VC \rangle \langle A \rangle$

$\Rightarrow 0$ 4 6 4 8 funcție este derivabilă continuă

orice

$\Rightarrow L(G) = \{$ u functie este v $|$ u $\in \{$o, orice$\}$, v $\in \{$derivabilă, continuă$\}\}$

2. a) $G = (N, \Sigma, S, P)$

$N = \{A, B\}$

$\Sigma = \{a, b, c\}$

$S = A$

P: $A \to aB$

$B \to b$

$B \to c$

b) $G = (N, \Sigma, S, P)$

$N = \{A, B, C\}$

$\Sigma = \{a, b, c\}$

$S = A$

P: $A \to aB$

$B \to bC$

$C \to c$

3.

1. EBNF pt. F

Numar = NumarNatural ["," ParteZecimală].

NumarNatural = CifraNenula {Cifra} | "0".

CifraNenula = "1" | "2" | ... | "9".

Cifra = CifraNenula | "0".

ParteZecimală = {Cifra} CifraNenula.

2. BNF pt. F

<Numar> ::= <NumarN> | <NumarN>,<NumarZ>

<NumarN> ::= <CifraNenula> | <CifraNenula><ListaCifre>

<CifraNenula> ::= 1|2|...|9

<ListaCifre> ::= <Cifra> | <Cifra><ListaCifre>

<Cifra> ::= 0 | <CifraNenula>

<NumarZ> ::= <ListaCifre><CifraNenula> | <CifraNenula>

4.

1. $ab \in L$

dacă $x \in L$ atunci $axb \in L$

a) $1 \to ab \in L$, $2 \to aabb \in L$, $3 \to aaabbb \in L$ ...

~~$L = \{vw \mid v \in \{a\}$~~ $L = \{a^m b^m \mid m \in \mathbb{N}^*\}$

b) $G=(N,\Sigma,S,P)$

$N=\{A\}$

$\Sigma=\{a,b\}$

$S=A$

$P=$ $A \to ab$

$A \to aAb$

5.

a. $L=\{x^n y^n \mid n\in\mathbb{N}\}$ dem. $L=L(G)$

$G=(N,\Sigma,S,P)$

$N=\{A\}$

$\Sigma=\{x,y\}$

$S=A$

$P:$ $A \xrightarrow{1} \varepsilon$

$A \xrightarrow{2} xAy$

I $L\subseteq L(G)$

$w=x^n y^n$, n fixat

$A \overset{2}{\Rightarrow}$ de n ori $\underbrace{xx\ldots x}_{n\text{ ori}}\underbrace{Ayy\ldots y}_{n\text{ ori}} = x^n A y^n \overset{1}{\Rightarrow} x^n y^n \in L(G)$

II $L(G)\subseteq L$

$P(n)$: $x^n A y^n$

$x^{n-1} A y^{n-1}$

$P(1)$: $A \overset{1}{\Rightarrow} \varepsilon$

$A \overset{2}{\Rightarrow} xAy$

pp. RA $P(k)$: $x^k A y^k \in L(A)$

$P(k+1)$: $x^{k+1} A y^{k+1} \in L$

$x^k A y^k \overset{1}{\Rightarrow} x^k y^k$ - $P(k+1)$

$\overset{2}{\Rightarrow} xx^k A y^k y = x^{k+1} A y^{k+1}$ - $P(k+1)$ (A)

2. $L = \{a^{2n}bc \mid n \in \mathbb{N}\}$

$G = (N, \Sigma, S, P)$

$N = \{A, B\}$

$\Sigma = \{a, b, c\}$

$S = A$

P: $A \to BBbc$, $B \to \varepsilon$, $B \to aaB$ | sau | $A \xrightarrow{1} bc$, $A \xrightarrow{2} a^2A$

3. $L = \{a^{2n+1}bc \mid n \in \mathbb{N}\}$

$G = (N, \Sigma, S, P)$

$N = \{A, B\}$

$\Sigma = \{a, b, c\}$

$S = A$

P: $A \to BabBc$, $B \to \varepsilon$, $B \to aaB$ | sau | $A \to abc$, $A \to a^2A$

2. dem.

I $L \subseteq L(G)$

fie $w = a^{2m}bc \in L$, m fixat

$A \xrightarrow{2} a^2A \xrightarrow{2} a^4A \xrightarrow{2} \dots \xrightarrow{2} a^{2m}A \xrightarrow{1} a^{2m}bc \in L(G) \Rightarrow$

(de $m-2$ ori)

$\Rightarrow w \in L(G) \Rightarrow L \subseteq L(G)$

II $L(G) \subseteq L$

$P(n)$: $a^{2n}A$

$a^{2n}bc$

$P(1)$: $A \xrightarrow{1} bc$

$A \xrightarrow{2} a^2A$

$P(k)$: $a^{2k}A \xrightarrow{1} a^{2k}bc$

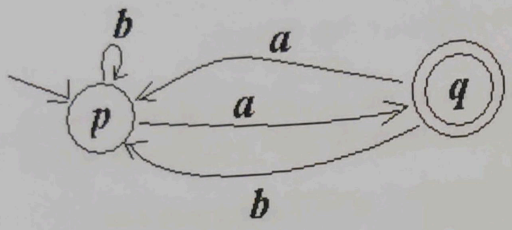
$\xrightarrow{2} a^2a^{2k}A = a^{2(k+1)}A$

Seminar 3

Automate finite

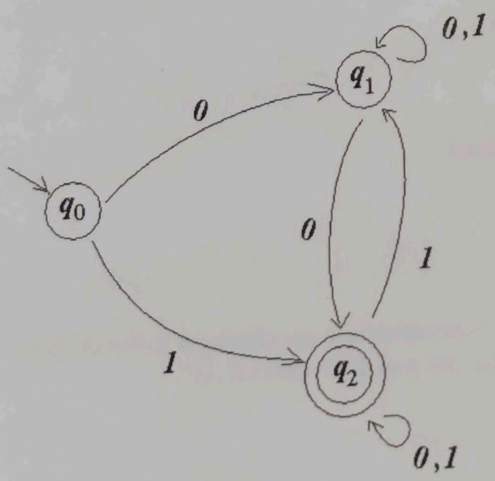
1.1 Probleme cu AF

1. Sa se reprezinte tabelar urmatoarele automate finite. Sint ele AFD sau AFN?



dit. complet

δ	a	b	
p	q	p	0
q	p	p	1



δ	0	1	
q0	q1	q2	0
q1	{q1,q2}	q1	0
q2	q2	{q1,q2}	1

nedit.

2. Sa se reprezinte sub forma de graf automatul finit: M=(Q,Σ, δ,q_0,F)

	0	1	
q0	q2	q1	1
q1	q3	q0	0
q2	q0	q3	0
q3	q1	q2	0

Verificati apoi, bazandu-va pe graful obtinut, ca:
a) secventele 1010, 1100 sunt acceptate de automat
b) secventa 1011 nu este acceptata de automat

3. Sa se construiasca un AF care accepta

a) $L = \{ a\ a\ a \}$, $\Sigma = \{ a \}$

b) $L = \{ w_1\ a\ a\ a\ w_2 \mid w_1, w_2 \in \{a, b\}^* \}$

c) cuvinte peste alfabetul {0,1} cu proprietatea ca:
orice cuvant al limbajului contine cel putin 2 zerouri consecutive

d) cuvinte peste alfabetul {a, b, c} cu proprietatea ca:
primul simbol al cuvantului este acelasi cu cel cu care se termina cuvantul

e) cuvinte peste alfabetul {a, b, c} cu proprietatea ca:
exista un simbol in cuvant care mai apare cel putin o data in cuvant

f) $L = \{c^{3n}, n \in N^*\}$

g) limbajul ce contine secvente peste Σ={a,b} cu nr. par de simb. **a** si nr.par de simb **b**

h) limbajul ce contine secv. peste Σ ={a,b} cu nr. impar de simb. **a** si impar de simb **b**

i) $L=\{ 1^n 0^m 1\ u \mid n>=0, m>=1, u \in \{0,1\}^* \}$

j) $L = \{ 0\ (10)^n 01^m \mid n>=0, m>=0\} \cup \{(10)^n 01^m \mid n>=1, m>=0\}$ cu cel mult 4 stari

k) $L=\{ 0^m 1^n \mid m,n \in N\} \cup \{1^p 0^q \mid p,q \in N\}$

l) $L = \{ w_1\ a\ a\ w_2 \mid w_1 \in \{b, ab\}^*, w_2 \in \{a, b\}^* \}$

1.2 Structuri de date pentru automate finite

Descrieti o modalitate de reprezentare pentru AF
(ganditi cateva modalitati)

ex:

- AF care are ca alfabet multimea caracterelor reprezentabile in calculator (alfabet fixat)
- structura de date care are in vedere ca operatia cea mai frecventa cu AF
  este verificare acceptare secventa

```
StateMachine
  description: String          /** A description of the state machine */
  startState :State
  states: Set<State>           /** Can be List, Map, … */
                               /** What about redundancy? */

State
  description: String          /** The name of the state */
  isAcceptState: Boolean
  transitions: MultiMap<Symbol, State> /** Can be List, Map … ? */
```

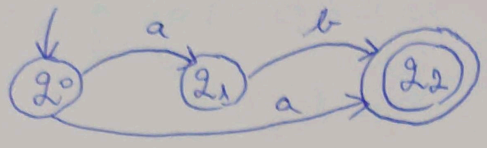
Seminar 3

$M = (Q, \Sigma, \delta, q_0, F)$

$\Sigma = \{a, b\}$

$Q = \{q_0, q_1, q_2\}$

$F = \{q_2\}$



δ	a	b	
q_0	$\{q_1, q_2\}$		0
q_1		q_2	0
q_2			1

~~cel puțin~~ o stare în fiecare poz. => det. complet ($|\delta(q,a)| = 1$)

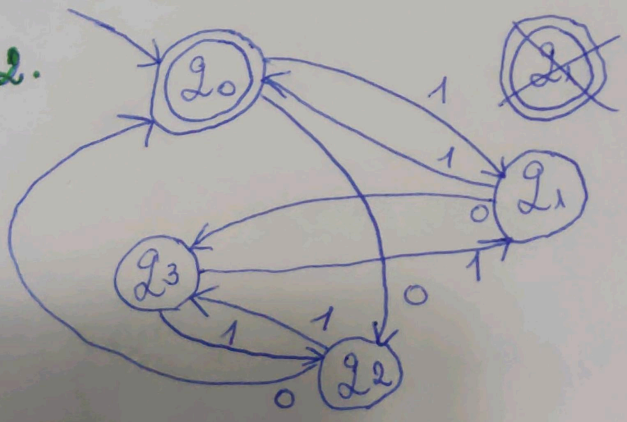
dacă în cel puțin o poz. sunt mai multe => nedet.

$(q_0, ab) \vdash (q_1, b) \vdash (q_2, \varepsilon)$

↑ dacă e finală => acceptat ($\in F$) cu ε

$(q_0, abb) \vdash (q_1, ab) \vdash (q_0, b)$ => refuzat

2.



a) $(q_0, 1010) \vdash (q_1, 010) \vdash (q_3, 10) \vdash (q_2, 0) \vdash (q_0, \varepsilon)$

$q_0 \in F$ => acceptat

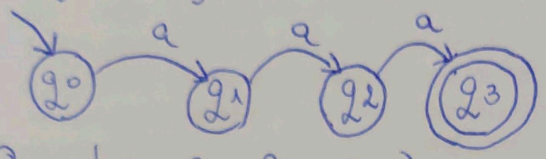
$(q_0, 1100) \vdash (q_1, 100) \vdash (q_0, 00) \vdash (q_2, 0) \vdash (q_0, \varepsilon)$

$q_0 \in F \Rightarrow$ acceptat

b) $(q_0, 1011) \vdash (q_1, 011) \vdash (q_3, 11) \vdash (q_2, 1) \vdash (q_3, \varepsilon)$

$q_3 \notin F \Rightarrow$ respins

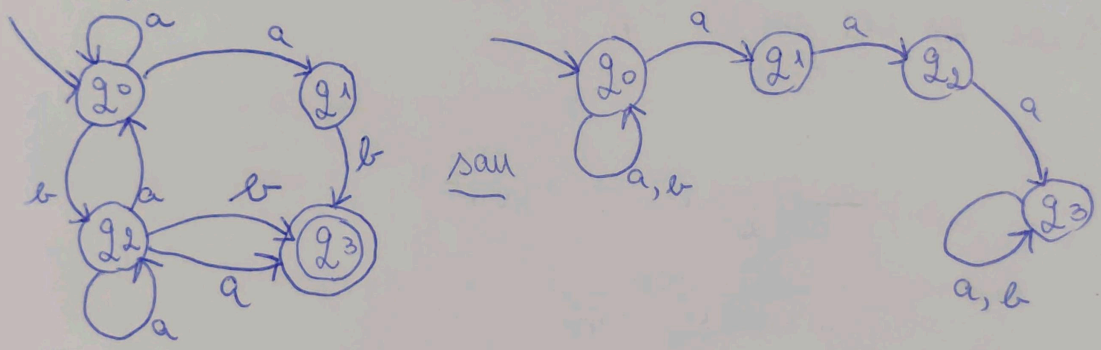
3. a) $L = \{aaa\}$, $\Sigma = \{a\}$ $\Rightarrow M = (Q, \Sigma, \delta, q_0, F)$



$Q = \{q_0, q_1, q_2, q_3\}$

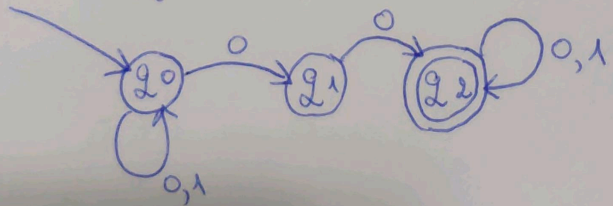
$F = \{q_3\}$

b) $L = \{w_1 aaa w_2 \mid w_1, w_2 \in \{a, b\}\}$



c) $\Sigma = \{0, 1\}$

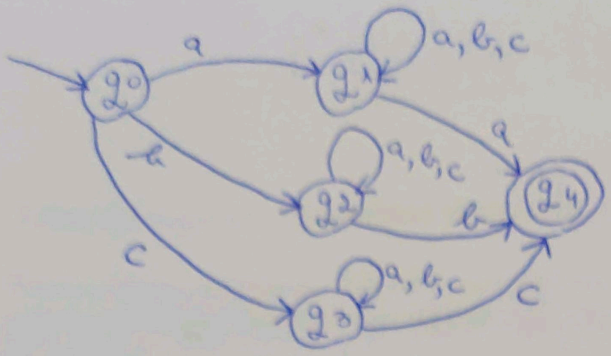
$L = \{w_1 00 w_2 \mid w_1, w_2 \in \{0,1\}^*\}$



d) $\Sigma = \{a, b, c\}$

$L = \{awa \mid w \in \{a,b,c\}^*\} \cup \{bwb \mid w \in \{a,b,c\}^*\} \cup$

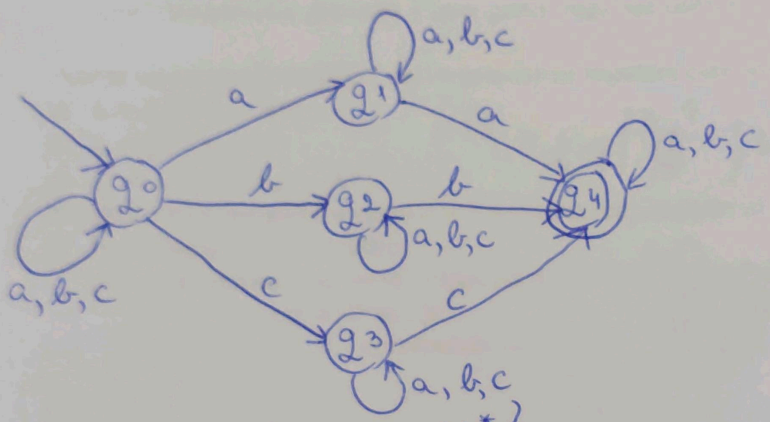
$\{cwc \mid w \in \{a,b,c\}^*\}$



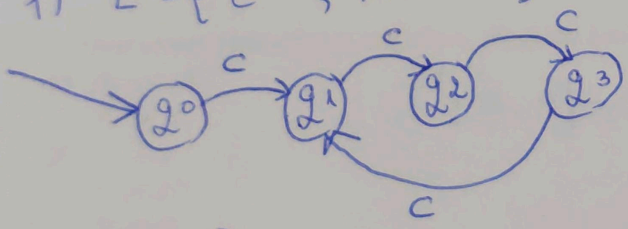
e) $\Sigma = \{a, b, c\}$

$w_1 a w_2 a w_3$, $w_1 b w_2 b w_3$, $w_1 c w_2 c w_3$

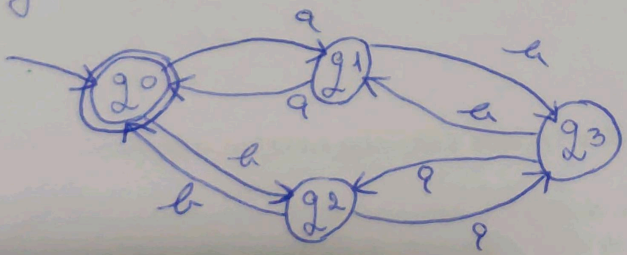
$w_1, w_2, w_3 \in \{a, b, c\}^*$



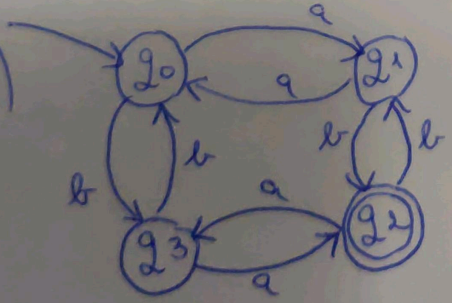
f) $L = \{c^{3n},\ n \in \mathbb{N}^*\}$



g) $\Sigma = \{a, b\}$

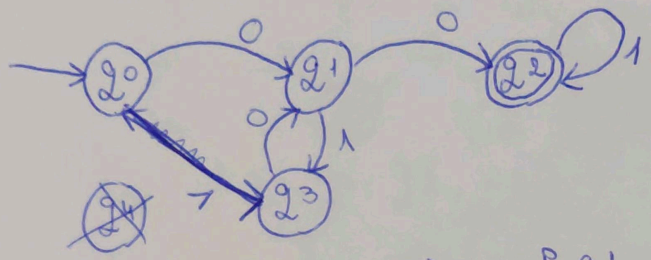


h)

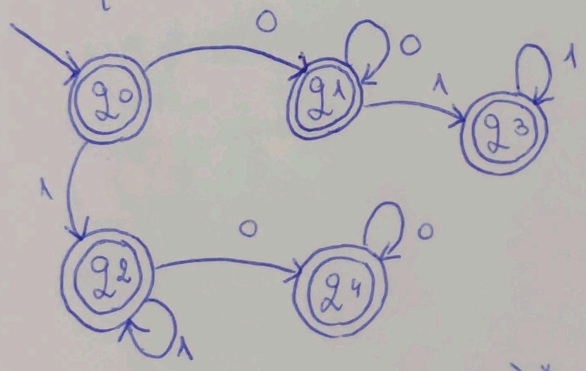


i) $L = \{ 1^n 0^m 1 u \mid n \geq 0, m \geq 1, u \in \{0,1\}^* \}$

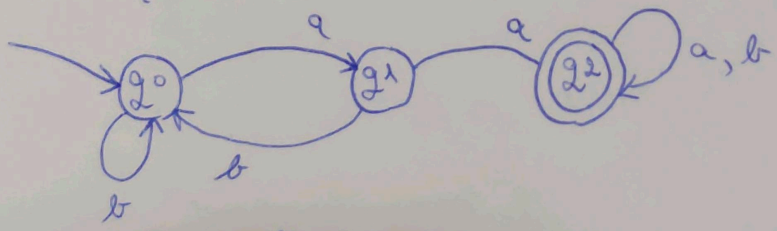
j) $L = \{ 0_{\#_1} (10)^n 0 1^m \mid n \geq 0, m \geq 0 \} \cup \{ (10)^n 0 1^m \mid n \geq 1, m \geq 0 \}$



k) $L = \{ 0^m 1^n \mid m, n \in \mathbb{N} \} \cup \{ 1^p 0^q \mid p, q \in \mathbb{N} \}$



l) $L = \{ w_1 a a w_2 \mid w_1 \in \{b, ab\}^*, w_2 \in \{a, b\}^* \}$



1.2 State Machine:

description : String
start State : State
states : Set<State>

State

description : String
isAccepted : Boolean
transition : Multimap<Symbol, State>

Seminar 4

Limbaje regulare : gramatici si automate

1. Pentru urmatoarele limbaje, dati AF care le accepta. Apoi dati o gramatica echivalenta. Este regulara? Daca nu, dati gramatica regulara echivalenta.
   a) $L = \{a\}$
   b) $L = \{a^n | n \in \mathbf{N}\}$
   c) $L = \{a^n b | n \in \mathbf{N}\}$
   d) $L = \{\varepsilon\} \cup \{a^n b | n \in \mathbf{N}\}$
   e) $L = \{a^m b^n | m,n \in \mathbf{N}, m+n > 0\}$
   f) $L = \{ab^n | n \in \mathbf{N}\}$
2. Descrieti constructia generala a unei gram. regulare echivalente cu un AF dat.

3. Pentru urmatoarele limbaje, dati o gramatica regulara care le genereza.
   a) $L = \{a^{3n} | n \in \mathbf{N}^*\}$
   b) $L = \{a^{3n} | n \in \mathbf{N}\}$
   c) $L = \{a^m b^n | m,n \in \mathbf{N}^*\}$
   d) limbajul constantelor numerice fara semn reprezentate in baza 2
   e) limbajul identificatorilor
   obs.: este permisa scrierea compacta a regulilor de productie folosind | si …
   ex: S -> a|…|c
   corespunde la: S -> a
   S -> b
   S -> c
   unde: a,…, z – terminale

4. Pentru urmatoarele gram. regulare, descrieti limbajul generat. Dati AF echivalent.
   a) $A \to aA$
   $A \to b$
   b) $S \to \varepsilon$
   $S \to aA$
   $A \to b$
   c) $S \to \varepsilon$
   $S \to aA$
   $A \to bA$
   $A \to c$

5. Descrieti constructia generala a unui AF echivalent cu o gram. regulara data.

6. Pentru urmatoarele limbaje, dati AF care le accepta. Apoi dati gr. regulara echivalenta, aplicand alg. (general) de construire. Apoi dati AF echiv. cu gr. regulara, aplicand alg. general de construire.
   a) $L = \{a^{2n} | n \in N\}$
   b) $L = \{a^m b^n | m,n \in N\}$

Seminar 4

$G = (N, \Sigma, P, S)$

L → gramatică

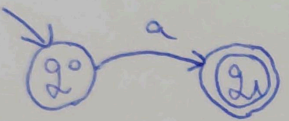
Gramatică regulară:

- $S \to a$
- $A \to aB$, $A \to a$
- $S \to \varepsilon$, atunci S nu mai apare în membrul drept al niciunei reguli de producție

Exemple care NU sunt regulare:

$A \to abC$, $A \to AB$, $AB \to cB$, $A \to Aa$

1. a)



$G = (N, \Sigma, P, S)$

$S = A$

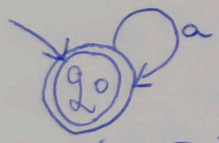
$N = \{A\}$

$\Sigma = \{a\}$

$P: A \to a$

regulară

b)



$G = (N, \Sigma, P, S)$

$S = A$

$N = \{A\}$

$\Sigma = \{a\}$

$P: A \to aA$

$A \to \varepsilon$

neregulară

$S = B$

$\Rightarrow$

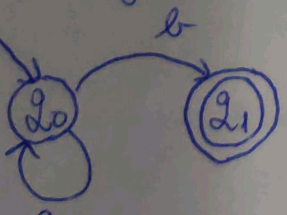
$P: A \to aA$

$A \to \varepsilon$

$B \to aA$

$B \to \varepsilon$

c)



$G = (N, \Sigma, P, S)$

$S = A$

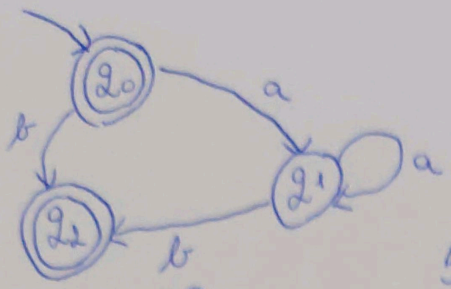
$N = \{A\}$

$\Sigma = \{a, b\}$

$P: A \to b$

$A \to aA$ regulară

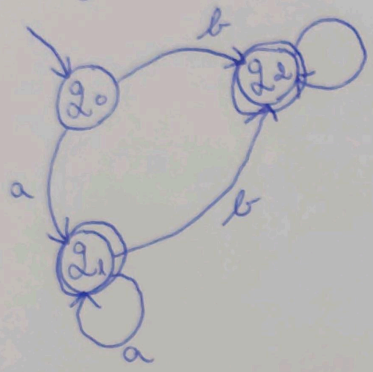
d)



$S = A$

P: $A \to aB$
$A \to b$
$A \to \varepsilon$
$B \to aB$
$B \to b$

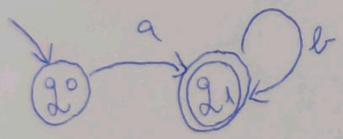
e)



$G = (N, \Sigma, P, S)$

P: $S \to a$
$S \to b$
$S \to aA$
$S \to bB$
$A \to a$
$A \to b$
$A \to aA$
$B \to b$
$B \to bB$

f)



P: $A \to a$
$A \to aB$
$B \to b$
$B \to bB$

Alg. I Algoritmul I de transf. în gramatică regulară

1. copiez toate regulile de producție, mai puțin $S \to \varepsilon$
2. dacă am $S \to \varepsilon$, copiez ruz. de prod. care îl conțin pe S, cu și fără S în membrul drept ↳ fără $S \to \varepsilon$
3. adaug S' ca simbol de start și copiez regulile cu S în stânga a.î. să fie S' în stânga + $S' \to \varepsilon$

2. $M = (Q, \Sigma, \delta, q_0, F) \Rightarrow G = (N, \Sigma, P, S)$

Alg. II transformare

$S = q_0$, $\Sigma = \Sigma$, $N = Q$

P: $\bigcup$
- $A \to aB$, dacă $B \in \delta(A, a)$
- $A \to a$, dacă $B \in \delta(A, a)$ și $B \in F$
- dacă $B \in F$ și $\delta(B, a), \forall a \in \Sigma = \emptyset$ atunci nu mai scriu regula $A \to aB$ scriu doar $A \to a$

dacă $S(q_0) \in F$ adaug $S \to \varepsilon$ (și eventual aplic alg. I).

3. a)
$A \to aB$
$B \to aC$
$C \to aA$
$C \to a$

b)
$A \to \varepsilon$
$A \to aB$
$B \to aC$
$C \to a$
$C \to aA$
neregulară X

$S = A$ $\Rightarrow$

$S \to \varepsilon$
$S \to aB$
$A \to \varepsilon$
$A \to aB$
$B \to aC$
$C \to a$
$C \to aA$
regulară ✓

$S = S$

c)
~~$S \to aA$~~
$A \to aA$
$A \to aB$
$B \to bB$
$B \to b$

d) $L = \{0\} \cup \{1w \mid w \in \{0,1\}^*\}$

$A \to 1$
$A \to 0$
$A \to 1B$
$B \to 0B$
$B \to 1B$
$B \to 0$
$B \to 1$

e) id (a...z){a...z0...9}

S -> a|b|...|z

S -> aA|bA|...|zA

A -> a|b|...|z

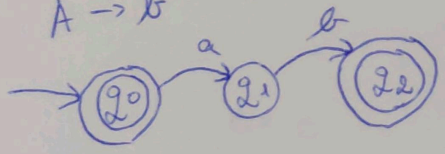
A -> 0|1|...|9

A -> 0A|1A|...|9A

A -> aA|bA|...|zA

4. a) A -> aA | => $L = \{a^m b \mid m \in \mathbb{N}\}$
A -> b

b) S -> ε | => $L = \{\varepsilon, ab\}$
S -> aA
A -> b



c) S -> ε | => $L = \{\varepsilon\} \cup \{ab^m c \mid m \in \mathbb{N}\}$
S -> aA
A -> bA
A -> c

5. $G = (N, \Sigma, P, S)$ => $M = (Q, \Sigma, \delta, q_0, F)$

Alg. III $Q = N \cup \{K\}$, $K \notin F$

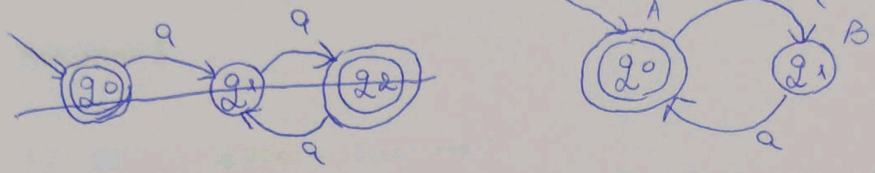
$\Sigma = \Sigma$

$q_0 = S$

$F = \{K\}$ sau $\{K, S\}$ dacă am $S \to \varepsilon \in P$

$\delta: \begin{cases} \delta(A,a) = \{B \mid A \to aB \in P\} \cup \{K \mid A \to a \in P\} \\ \delta(K,a) = \emptyset, \forall a \in \Sigma \end{cases}$

6. a)

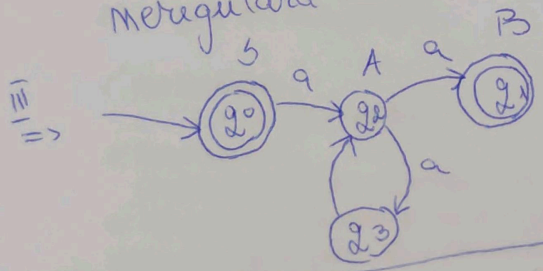


II $\Rightarrow$ P: $A \to aB$, $B \to aA$, $B \to a$, $A \to \varepsilon$

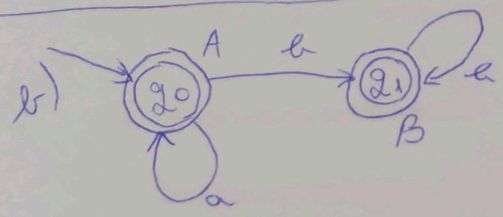
I $\Rightarrow$ $S \to \varepsilon$, $S \to aB$, $A \to aB$, $B \to a$, $B \to aA$ $\Rightarrow$

neregulară

III $\Rightarrow$



b)



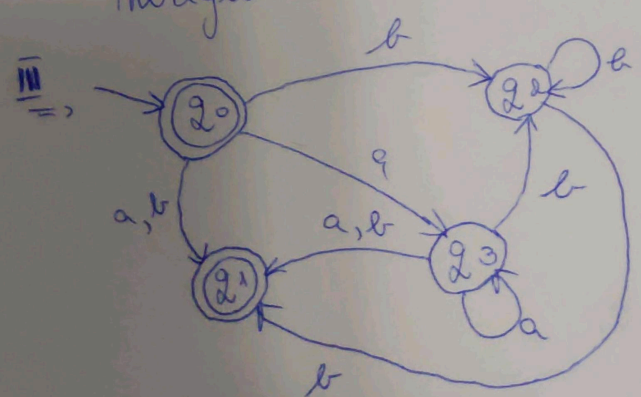
II $\Rightarrow$ P: $A \to \varepsilon$, $A \to a$, $A \to b$, $A \to aA$, $A \to bB$, $B \to b$, $B \to bB$

neregulară X

I $\Rightarrow$ $S \to \varepsilon$, $S \to a$, $S \to b$, $S \to aA$, $S \to bB$, $A \to a$, $A \to b$, $A \to aA$, $A \to bB$, $B \to b$, $B \to bB$

regulară ✓

III $\Rightarrow$



Seminar 5

1.1 limbaje regulare – a fi sau anu fi

1. Sa se verifice daca urmatoarele limbaje sint regulare. Daca nu sint – demostrati! Daca sint, doar argumentati. (Puteti indica un AF care le accepta. Puteti argumenta si altfel? ☺)

a. $L = \{a^n b^{2n} \mid n \in \mathbf{N}^*\}$

b. $L = \{a^k \mid k$ – nr.prim$\}$

c. L = {a^n^2 | n ∈ **N***}

d. L = {a^2^n | n ∈ **N***}

e. Fie k – un nr. natural fixat

$L = \{a^{kn} \mid n \in N\}$

f. $L = \{a^n b^m c^p \mid m, n, p \in \mathbf{N}\}$

2. Fie limbajul:

$L = \{a^k \mid k$ - neprim$\}$

a) Este regular?

(*Indicatie: folositi proprietatile de inchidere ale limbajelor regulare*)

b) Alegeti un nr. natural **p** astfel incat, alegand un cuvant din limbaj, de lungime mai mare decat **p**, sa puteti da o descompunere $w=xyz$ astfel incat xy^iz sa fie tot un cuvant din limbaj, pentru orice **i** – numar natural nenul.

1.2 Limbaje de toate felurile

1. a) Dati un limbaj regular. Dati AF care il accepta. Dati o gram. regulara ce il genereaza. Dati o gramatica i.c. care nu e regulara ce il genereaza.

b) Dati un limbaj i.c. care nu este regular. Dati g.i.c. ce il genereaza.

1.3 Gramatici regulare echivalente cu o gramatica data (dar care genereaza un limbaj regular)

1. Dati o gramatica regulara echivalenta cu gram. data prin urmatoarele r.p.:

a) S->abS
S->ab

b) S->Sa
S->b

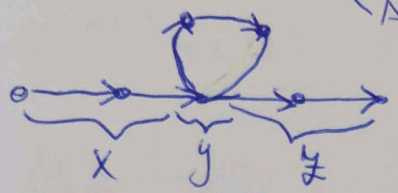
Seminar 5

Lema de pompare pentru limbaje regulare

L regular atunci $\exists p \geq 1$ pentru care $\forall w \in L, |w| \geq p$

$\exists w = xyz, 0 < |y| \leq p$, $w_i = xy^i z \in L, \forall i \in \mathbb{N}$

↖ sau $0 < |xy| \leq p$ (condiție mai tare)



⇓ ⚡

Negația: $\forall p \geq 1$, $\exists w \in L$, $\forall w = xyz$, $0 < |y| \leq p$, $|w| \geq p$

$w_i = xy^i z$ $\exists i \in \mathbb{N}$ pt. care $w_i \notin L$

1.1

1.a) $L = \{a^n b^{2n}\}$ (dreptunghiuri, când avem două cicluri care depind unul de altul, nu e regular)

presupunem RA că L - regulat

$\exists p \geq 1$ a.î. $\forall w \in L$, $|w| \geq p$

$\exists w = xyz$; $0 < |y| \leq p$

$w_i = xy^i z$, $w_i \in L$, $\forall i \in \mathbb{N}$

$w = a^p b^{2p}$ $\Rightarrow |w| = 3p \geq p$ ✓

$x = a^{p-k}$

$y = a^k$

$z = b^{2p}$

$\Rightarrow w_i = a^{p-k}(a^k)^i b^{2p}$

$w_i = a^{p-k} \cdot a^{ik} b^{2p}$

$w_i = a^{p-k+ik} b^{2p}$

de ex: $w_2 = a^{p+k} b^{2p}$ $\Rightarrow w_2 \notin L \Rightarrow L$ nu este regular

dar $k \neq p$

b) $L = \{a^k \mid k$ - nr prim$\}$

pp. RA, L-regular, conf. lemei de pompare

$\Rightarrow \exists p \geq 1$ a.î. $\forall w \in L$ cu $|w| \geq p$

$\exists w = xyz$, $0 < |y| \leq p$

$w_i = xy^i z$, $w_i \in L$, $\forall i \in \mathbb{N}$

$w = a^p \Rightarrow |w| = p \leq p$

$y = a^j \Rightarrow 0 < |y| \leq p$

$\Rightarrow w = a^{p-j} a^j$

$w_i = a^{p-j}(a^j)^i = a^{p-j+ij}$

ex: $w_{p+1} = a^{p-j+(p+1)j} = a^{p+pj} = a^{p(j+1)}$

dar $p(j+1)$ nu este prim $\Rightarrow w_{p+i} \notin L$ $\Rightarrow$ L nu est regular

c) $L = \{a^{n^2} \mid n \in \mathbb{N}^*\}$

$w = a^{p^2}$ $0 < |y| \leq p$ $(0 < j \leq p)$

$y = a^j \Rightarrow w_i = a^{p^2 - j}(a^j)^i = a^{p^2 - j} \cdot a^{ij} = a^{p^2 + j(i-1)}$

$x = a^{p^2-j}$

$y = \varepsilon$

ex: $w_2 = a^{p^2+j}$,

$p^2 < p^2 + j < (p+1)^2 = p^2 + 2p + 1 \quad / -p^2$

$0 < j < 2p + 1$ (A) (deoarece $0 < j \leq p$) $\Rightarrow w_2 \in L \Rightarrow$ L nu este regular

d) $L = \{a^{2^n} \mid n \in \mathbb{N}^*\}$

$w = a^{2^p}$

$y = a^j$, $0 < j \leq p$

$\Rightarrow w = a^{2^p - j} \cdot a^j$

$w_i = a^{2^p - j} \cdot (a^j)^i = a^{2^p - j + ij} = a^{2^p + j(i-1)}$

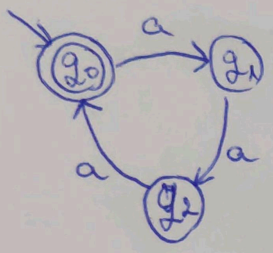
$w_2 = a^{2^p + j}$

$0 < j < p \iff 2^p < j + 2^p < \underbrace{p + 2^p}_{< 2^{p+1}}$, $p \geq 2$

$\Rightarrow 2^p + j$ nu este de forma $2^m \Rightarrow w_2 \notin L \Rightarrow L$ nu este regular

e) k-nr. nat. fixat

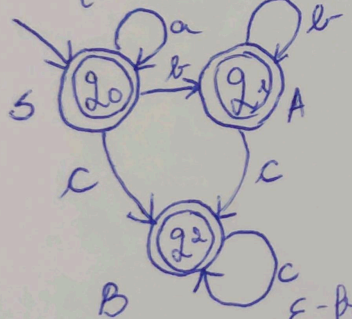
$L = \{ a^{km} \mid m \in \mathbb{N} \}$ ex: a^{2m}, a^{3m}



Limbaj regular $\iff$ poți desena un AF $\iff$

$\iff$ gramatică regulară $\iff$ expr. reg.

f) $L = \{ a^m b^m c^p \mid m, m, p \in \mathbb{N} \}$



expr. reg. $a^* b^* c^*$

$\rightarrow \varepsilon$-free

$S \to \varepsilon$
$S \to a$
$S \to aS$
$S \to b$
$S \to bA$
$S \to c$
$S \to cB$
$A \to b$
$A \to cB$
$A \to c$
$B \to c$
$B \to cB$
$A \to bA$

$S' \to \varepsilon$
$S' \to a$
$S' \to b$
$S \to a$
$S \to b$
$S \to c$
$S \to aS$
$S \to bA$
$S \to cB$
$A \to b$
$A \to c$
$A \to bA$
$A \to cB$
$B \to c$
$B \to cB$

$S' \to c$
$S' \to aS$
$S' \to bA$
$S' \to cB$

2. $L = \{a^k \mid k \text{ - nprim}\}$

Proprietăți de închidere

L_1, L_2 reg. $\Rightarrow L_1 \cup L_2, L_1 \cap L_2, L_1 L_2, L_1^*$, compl$(L_1) \Leftrightarrow \bar{L_1}$ ~~regulare~~ sunt regulare

a) pp L reg. $\Rightarrow$ (prop. închidere) compl(L) regular

$\bar{L} = \{a^k \mid k \text{ prim}\}$ $\Rightarrow L$ nu e regular

la 1b) am arătat că nu e reg.

$w = a^p$, $|w| \geq p$

$y = a^j$

$w_i = a^{p-j} \cdot a^{ji} = a^{p+j(i-1)}$

$w_{p+1} = a^{p+pj} = a^{p(j+1)}$

$p(j+1)$ nu e prim ---

b) p - natural ≥ 1

$p = ?$ a.î. $\forall w \in L, \exists w = xyz$ a.î

$w_i = xy^i z \in L, \forall i \in \mathbb{N}^*$

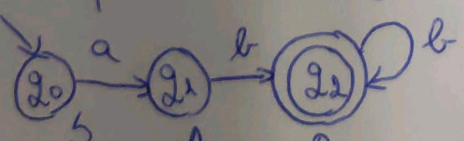
$p = 3$ $|w| > 3$ $\Rightarrow w = a^2 \cdot a^2$

$w = a^4$

$w_i = a^2 \cdot a^{2i} = a^{2(i+1)}$, $\forall i \in \mathbb{N}^*$

1.2

1.a) $L = \{ab^n \mid n \in \mathbb{N}^*\}$



$S \to aA$ $B \to bB$

$A \to b$ $B \to b$

$A \to bB$

$S \to abB$

$B \to bB$

$B \to \varepsilon$

b) $L = \{ a^n b^{n+1} \mid n \in \mathbb{N} \}$

$S \to b$

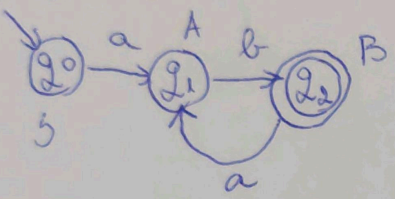
$S \to aAbb$

$A \to aAb$

$A \to \varepsilon$

1.3

a) $S \to abS$ | $S \to ab$ $\Rightarrow (ab)^n, n \geq 1$



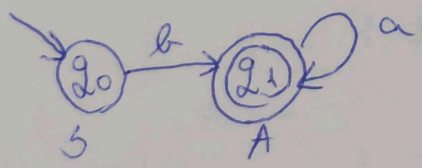
$S \to aA$

$A \to bB$

$A \to b$

$B \to aA$

b) $S \to Sa$ | $S \to b$ $\Rightarrow ba^n, n \geq 0$



$S \to bA$

$A \to \varepsilon$

$A \to aA$

Seminar 6

1. a) 01110111 $(1^*01)^*(11+0)^*$ + = sau

$(\underline{1^*01})\ (\underline{1^*01})\ (\underline{11}+0)^*$

ε 11

=> ~~acceptată~~ da

b) 11100111 $(1^*0)+(0^*11)$

$(\underline{1^*0})(\underline{1^*0})$ ↳nu => nu

111 ε

c) 11100111 $(1^*0)+(0^*11)$

$(\underline{1^*0})$ ↳nu => nu

111

d) 11100111 $(1^*0)^*(0^*11)$

$(\underline{1^*0})\ (\underline{0^*11})$

111 0

=> da

e) 011100101 $01^*01^*(11^*0)^*$

$\underline{01^*}\ \underline{01^*}$?

111 ε

=> nu

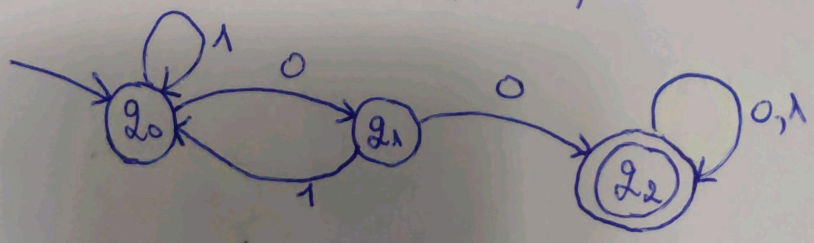
f) 10000011 $(10^*+11)^*(0^*1)^*$

$(\underline{10^*+11})\ (\underline{0^*1})^*$

0000 11

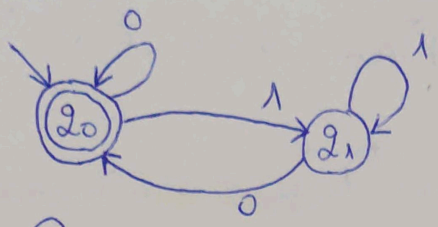
=> da

2. a) $(01+1)^*00(0+1)^*$

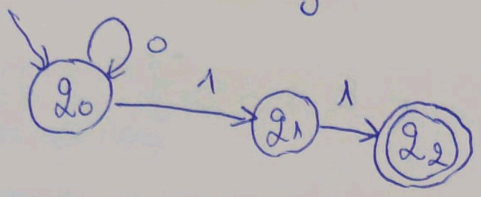


b) $(1^*0)^* + 0^*11$

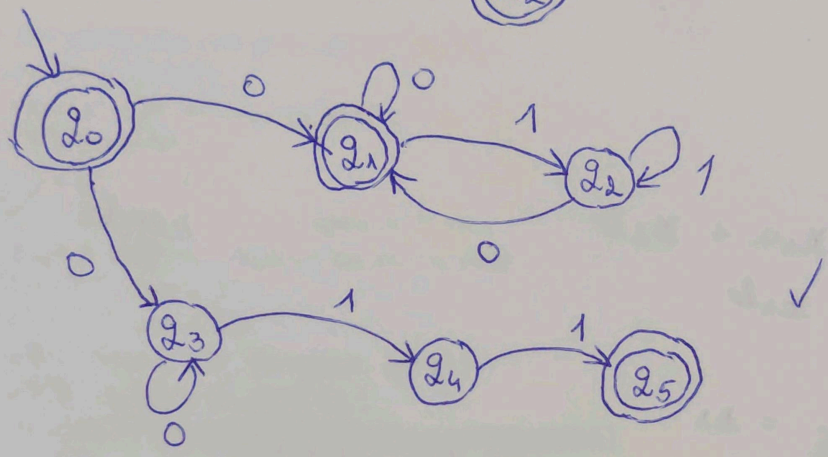
$(1^*0)^*$:



0^*11:

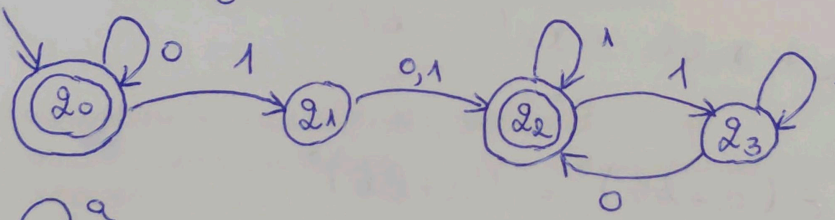


total:



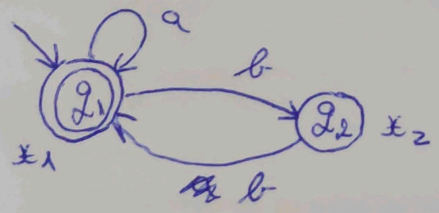
✓

=>



dupe net!

3.



	$k=0$	$k=1$	$k=2$
r_{11}^k	$\varepsilon + a$	a^*	
r_{12}^k	b	a^*b	
r_{21}^k	b	ba^*	
r_{22}^k	ε	$\varepsilon + ba^*b$	

$$r_{12}^1 = r_{12}^0 + r_{11}^0 (r_{11}^0)^* r_{12}^0 =$$

$$= b + (\varepsilon + a)(\varepsilon + a)^* b =$$

$$= b + (\varepsilon + a)^+ b = (\varepsilon + (\varepsilon + a)^+) b =$$

$$= (\varepsilon + a)^* b = a^* b$$

$$R_{21}^{(1)} = R_{21}^{(0)} + R_{21}^{(0)} (R_{11}^{(0)})^* R_{11}^{(0)} = b + b(\varepsilon + a)^* (\varepsilon + a) = ba^*$$

$$R_{22}^{(1)} = R_{22}^{(0)} + R_{21}^{(0)} (R_{11}^{(0)})^* R_{12}^{(0)} =$$

$$= \varepsilon + b \underbrace{(\varepsilon + a)^*}_{a^*} b = \varepsilon + ba^*b$$

⋮

Metoda 2

$$\begin{cases} x_1 = x_1 a + x_2 b \\ x_2 = x_1 b \end{cases}$$

regulă $X = \beta + \alpha X$
$\Rightarrow X = \beta \alpha^*$

$\cup\ x_{final} = x_1$

$$x_1 = x_1 a + x_1 bb = x_1 \underbrace{(a + bb)}_{\alpha}$$
$$\beta = 0$$

$$\Rightarrow x_1 = \varepsilon (a + bb)^* = (a + bb)^*$$

Seminar 7

1 Limbaje si gramatici de toate felurile

1. Definiti (folosind multimi) limbajul secventelor de simb. *a* care apar in numar par. Dati o gramatica regulara, una independenta de context si una ce nu e independenta de context care genereaza acest limbaj.

2. Sa se dea cate o gramatica care genereaza limbajele:
   1. $L = \{ww \mid w \in \{a, b\}^*\}$
   2. $L = \{wxw \mid w \in \{a, b\}^+, x \in \{a, b\}^*\}$
   3. $L = \{a^n b^n c^n \mid n \in \mathbf{N}^*\}$
   4. $L = \{a^n b^n c^n d^n \mid n \in \mathbf{N}^*\}$
   5. $L = \{a\text{^}2^n \mid n \in \mathbf{N}\}$ a apare de 2^n ori
   6. $L = \{w \mid w \in \{a, b, c\}^*, nr_a(w) = nr_b(w) = nr_c(w)\}$
   7. $L = \{a^n b^n c^m d^m \mid n,m \in \mathbf{N}\}$
   8. $L = \{a^n b^m c^m d^n \mid n,m \in \mathbf{N}\}$
   9. $L = \{a^n b^m c^k \mid n,m,k \in \mathbf{N}$, (n=m) sau (m=k) $\}$
   10. $L = \{a^n b^m c^k \mid n,m,k \in \mathbf{N}$, m+n = k $\}$
   11. L = { w ∈ {a, b}*| w incepe si se termina cu acelasi simbol }

2 Ambiguitate in gramatici independente de context

1. Sa se arate ca gramaticile urmatoare sint ambigue si sa se gaseasca o gramatica echivalenta neambigua.
   a) S-> aS | Sb | c
   b) S -> if b then S else S | if b then S | stmt
   c) S -> (S | S) | (S) | 1

3. Reprezentari pentru gramatici independente de context

Fie gramatica:
G = ({E,T,F},{a,+,*,(,)} , P, E)
P:
$E \rightarrow E + T$
$E \rightarrow T$
$T \rightarrow T * F$
$T \rightarrow F$
$F \rightarrow (E)$
$F \rightarrow a$

Pentru gramatica data, ilustrati modurile de reprezentare:
a) Folosind liste liniare (de exemplu: vectoriala)
b) reprezentarea cu ajutorul unor liste inlantuite ramificate
- Liste dublu inlantuite
- Liste triplu inlantuite

Seminar 4

Gramatici regulare

$S \to \varepsilon$
$S \to aA$
$S \to a$

Gramatici independente de context (GIC)

$S \to \alpha$
$\alpha \in N \cup \Sigma$

ex. $S \to ABC$
$S \to aaBbb$

Gramatici dependente de context (GDC)

$\alpha A \beta \to \alpha \gamma \beta$

1. $L = \{a^{2m} \mid m \in \mathbb{N}\}$

G. reg.

$A \to \varepsilon$
$A \to aB$
$B \to a$
$B \to aC$
$C \to aB$

GIC – $A \to \varepsilon$
$A \to aaA$

GDC $A \to \varepsilon$
$A \to aB$
$aB \to aa$
$aB \to aaaB$

2. $L = \{ww \mid w \in \{a,b\}^*\}$ ex. $\underbrace{aba}_{w}\underbrace{aba}_{w}$

$S \to \varepsilon$
$S \to aSA$
$S \to bSB$
$aS \to aD$
$bS \to bD$
$DB \to Db$

$DA \to Da$
$aA \to Aa$
$bA \to Ab$
$aB \to Ba$
$bB \to Bb$
$D \to \varepsilon$

2. $L = \{wxw \mid w \in \{a,b\}^+, x \in \{a,b\}^*\}$

$S \to aSA$

$S \to bSB$

~~$AB \to aXA$~~

$\{A \to X\}$ – sau $\begin{cases} A \to \varepsilon \\ aS \to aXA \\ bS \to bXA \end{cases}$

$X \to \varepsilon$

$X \to aX$

$X \to bX$

$AB \to Ab$

$AA \to Aa$

$aA \to Aa$

$bB \to Bb$

$aB \to Ba$

$bA \to Ab$

– "delimitator"? $\to A$

3. $L = \{a^n b^n c^n \mid n \in \mathbb{N}^*\}$

$S \to abc$

$S \to aAbc$

$Ab \to bAc$

$Ac \to BbCc$

$B \to Bb$

$aB \to aa$

(*) $aB \to aaA$

a bc
↑ ↑ ~~subscai~~
puncte cheie

"cursor"? $\to A, B$

$aAbc$

$abAc$

$abBbcc$

$aBbbcc$

aa (*)

$aaAbbcc$

4. $L = \{a^m b^m c^m d^m \mid m \in \mathbb{N}^*\}$

$S \to abcd$

$S \to aAbcd$

$Ab \to bA$

$Ac \to cA$

$Ad \to cBdd$

abcd
↑ ↑
~~cursoai~~ puncte cheie

$aAbcd$

$abAcd$

$abcAd$

$abcBdd$

$abcBcdd$

$abBccdd$

$cB \to Bc$
$bB \to Bb$
$aB \to aab$
$aB \to aaAb$

aBbccdd ↓
aab
aaAbbccdd ↓

5. $L = \{a^{2^n} \mid n \in \mathbb{N}\}$

a
aa
aaaa

$S \to LaR$
$L \to \varepsilon$
$R \to \varepsilon$
$L \to L\Delta$ cursor Δ
$\Delta a \to aa\Delta$
$\Delta R \to R$

6. $L = \{w \mid w \in \{a,b,c\}^*, nr_a(\ ) = nr_b(w) = nr_c(w)\}$

prea mult pentru capacitatea mea...

$S \to \varepsilon$
$S \to ABCS$
$AB \to BA$
$AC \to CA$
$BA \to AB$
$BC \to CB$
$CA \to AC$
$CB \to BC$
$A \to a$
$B \to b$
$C \to c$

4. $L = \{a^m b^n c^m d^n \mid n, m \in \mathbb{N}\}$

cicluri $3 \geq$ GDC, altfel GIC

$S \to \varepsilon$
$S \to aAb$
$A \to aAb$
$A \to \varepsilon$

$S \to c\Delta d$
$\Delta \to c\Delta d$
$\Delta \to \varepsilon$

$S \to aAbc\Delta d \iff S \to A\Delta$

8. $L = \{ a^n b^m c^m d^n \mid m, n \in \mathbb{N} \}$

$S \to \varepsilon$
$S \to aAd$
$A \to aAd$
$A \to \varepsilon$
$A \to bBc$
$B \to bBc$
$B \to \varepsilon$

$S \to bBc$

9. $L = \{ a^n b^m c^k \mid n, m, k \in \mathbb{N}, (n = m) \text{ sau } (n = k) \}$

$S \to \varepsilon$
$S \to aAbC$
$A \to aAb$
$A \to \varepsilon$
$C \to cC$
$C \to \varepsilon$

$S \to \Delta bEc$
$E \to bEc$
$E \to \varepsilon$
$\Delta \to a\Delta$
$\Delta \to \varepsilon$

10. $L = \{ a^n b^m c^k \mid n + m = k, n, m, k \in \mathbb{N} \}$ peste puteri

$k = 0$ $S \to \varepsilon$

$m = 0$ $S \to b\Delta c$
$\Delta \to \varepsilon$
$\Delta \to b\Delta c$

$n = 0$ $S \to aEc$
$E \to \varepsilon$
$E \to aEc$

altfel $S \to aAbcc$
$Ab \to bA$
$Ac \to bBcc$
$Bc \to c$
$Bc \to bBcc$
$bB \to Bb$
$aB \to aaC$
$Cb \to bC$

$Cc \to cc$
$Cc \to Bcc$
$aA \to aaC$

11. $L = \{ w \in \{a,b\}^* \mid w$ începe și sf. cu același $\}$

$S \to \varepsilon$
$S \to aAa$
$S \to bAb$
$A \to \varepsilon$
~~$A \to b$~~
$A \to aA$
$A \to bA$

2.

1.a) $S \to aS$
$S \to Sb$
$S \to c$

ambiguă

$S \to aS$
$S \to cB$
$B \to \varepsilon$
$B \to bB$

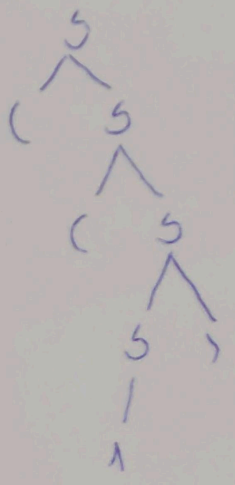
b) $S \to$ ~~if b then S~~
$S \to$ ~~if b then S else S~~
$S \to$ stmt

$S \overset{st}{\Rightarrow}$ ~~if b then S~~ $\overset{st}{\Rightarrow}$ ~~if b then~~ if b then S else S $\Rightarrow$

$S \overset{st}{\Rightarrow}$ if b then S $\overset{st}{\Rightarrow}$ if b then if b then S else S $\Rightarrow$

$\overset{st}{\Rightarrow}$

c) $S \to (S$
$S \to S)$
$S \to (S)$
$S \to \lambda$



((λ)

((λ) => ambiguă

-> $S \rightarrow (S$

$S \rightarrow 1A$

$A \rightarrow A)$

$A \rightarrow \varepsilon$

3. $G = (N, \Sigma, P, S)$

CFGrammar

startSymbol : Symbol

nonTerminals : Set<Symbol>

terminals : Set<Symbol>

rules : List<Rule>

sau

Multimap<Symbol, List<Symbol>>

Symbol

value : String

isTerminal : boolean

Rule :

LHS : Symbol

RHS : ~~Symbol~~ List<Symbol>

Seminar 8

Automate push-down (APD)

1. Construiti APD care accepta urmatoarele limbaje dupa criteriul stivei vide:

a) $L = \{a^n b^{2n} \mid n>=0\}$
b) $L = \{a^n b^m \mid m,n >=0\}$
c) $L = \{a^n b^m \mid n >= m >= 0\}$
d) $L = \{a^m b^n \mid n >= m >= 0\}$
e) $L = \{ww^{tilda} \mid w \in \{a,b\}^*, w^{tilda}$ este inversul lui $w\}$
f) $L = \{w \mid w \in \{a, b\}^*, nr_a(w) = nr_b(w)\}$
g) $L = \{a^{2n} b^{2n} \mid n>=0\}$

h) $L = \{a^n b^n \mid n>=0\} \cup \{b^n a^n \mid n>=0\}$
i) $L = \{a^n b^n \mid n>=0\} \cup \{a^n b2^n \mid n>=1\}$
j) $\{w\ x \mid w^{tilda}$ is a substring of x, where $x \in \{a, b\}^*$, $w \in \{a, b\}^*, |w|>=1\}$

2. Pentru limbajul de la punctul **f**, dati o gramatica independenta de context (GIC) ce il genereaza. Construit APD echivalent cu GIC data (aplicand algoritmul de constructie).

3. Pentru APD de la punctele **e** si **f**, dati APD care accepta acelasi limbaj dupa criteriul starii finale. ?

Seminarul 8

Sem 4

3.b)

$E \to E+T$

$E \to T$

$T \to T*F$

$T \to F$

$F \to (E)$

$F \to a$

~~refuz!~~

$M_F = (Q, \Sigma, \delta, q_0, F)$

PD - push down

$M_{PD} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

(Γ: alfabetul stivei; q_0: stare inițială; Z_0: stivă)

$\delta(q_0, w, Z_0) \vdash (q_1, \underbrace{A}_{\text{din } \Gamma^*})$

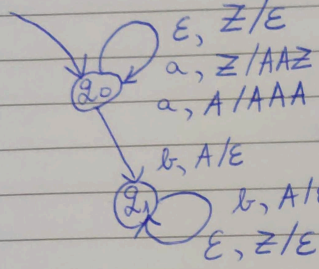
$L_\varepsilon = (q, \varepsilon, \varepsilon)$ - stiva vidă ($q \in Q$)

$L_f = (q_f, \varepsilon, \text{orice})$

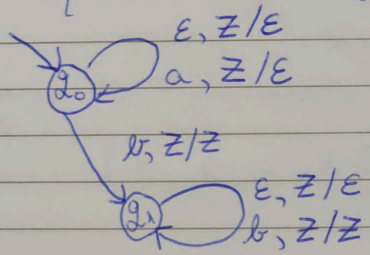
$q_f \in F$

1. a)

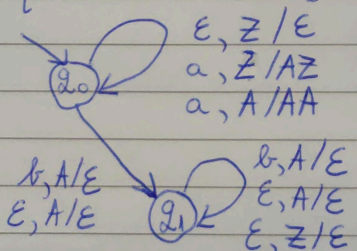
$L = \{a^n b^{2n} \mid n \geq 0\}$



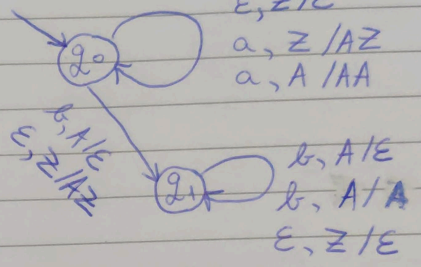
b) $L = \{a^n b^m \mid m, n \geq 0\}$



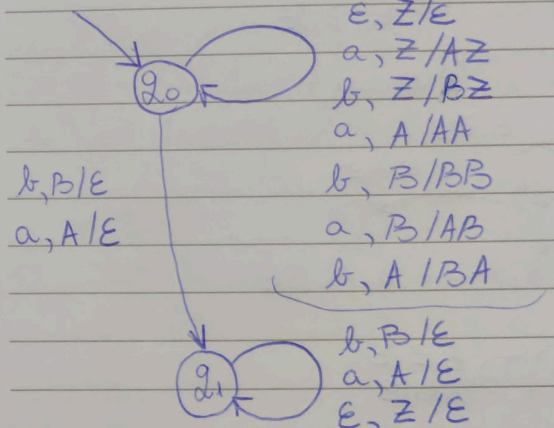
c) $L = \{a^n b^m \mid n \geq m \geq 0\}$



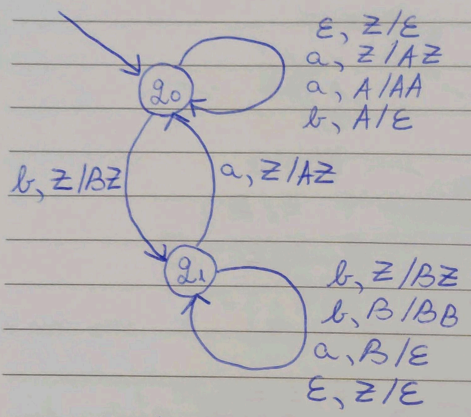
d) $L = \{a^n b^m \mid n \geq m \geq 0\}$



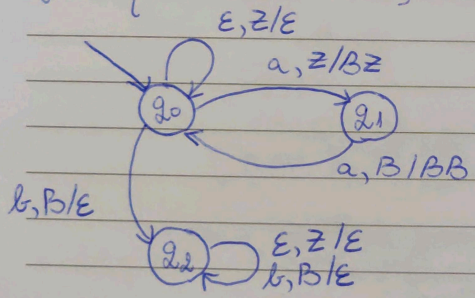
e) $L = \{ww^{\sim} \mid w \in \{a,b\}^*, w^{\sim}$ - inversul lui $w\}$



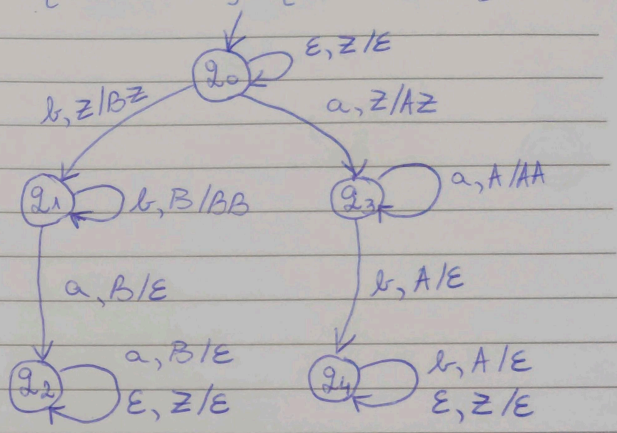
f) $L = \{w / w \in \{a,b\}^*, nr_a(w) = nr_b(w)\}$



g) $L = \{a^{2m} b^{2m} / m \geq 0\}$



h) $L = \{a^m b^m / m \geq 0\} \cup \{b^m a^m / m \geq 0\}$



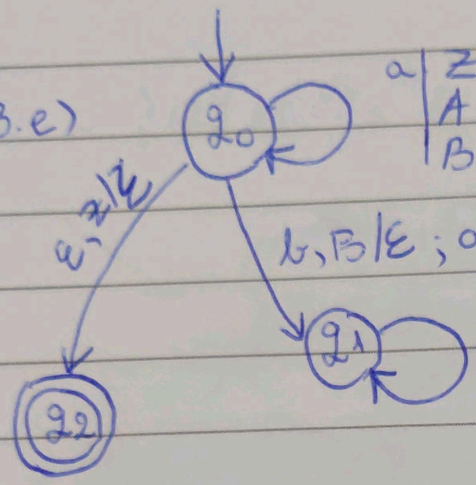
2. w, $nr_a(w) = nr_b(w)$

$S \to \varepsilon$
$S \to aSb$
$S \to bSa$
$S \to SS$

$A \to \alpha$
$\delta(q, \varepsilon, A) = \cup \{(q, \alpha)\}$
$\delta(q, a, a) = \{(q, \varepsilon)\}$

		a	b	ε
q	S			(q, aSb), (q, ε), (q, bSa), (q, SS)
	a	(q, ε)		
	b		(q, ε)	

3.e)



a	Z/AZ
	A/AA
	B/AB

b	Z/BZ
	A/BA
	B/BB

Analiza sintactica
Seminar saptamanile 9 , 10, 11, 12

Continut:

1 Analizorul descendent cu reveniri

1. Fie gramatica:

$S \rightarrow aSbS$
$S \rightarrow aS$
$S \rightarrow c$

Folosind analizorul descendent cu reveniri verificati daca :

- $acbc \in L(G)$ (?)
- $cb \in L(G)$ (?).

2. Analog pentru gramatica:

$S \rightarrow + S S$
$S \rightarrow - S S$
$S \rightarrow a$

si secventa: +a-aa .

2 Functiile $FIRST_1$, $FOLLOW_1$

1. Determinati $FIRST_1$ si $FOLLOW_1$ pentru neterminalele urmatorei gramatici:

$S \rightarrow abA$
$S \rightarrow \varepsilon$
$A \rightarrow Saa$
$A \rightarrow b$

3 Analiza sintactica descendenta LL(1)

1. Fie gramatica:

S -> if c then S endif
S -> if c then S else S endif
S -> stmt

Daca inlocuim: ***if c then*** cu ***a***, ***else*** cu ***b*** , ***endif*** cu ***c*** , si ***stmt*** cu ***i*** avem:

S -> a S c
S -> a S b S c
S -> i

Pentru una dintre cele 2 gramatici de mai sus:
a) Verificati daca gramatica este LL(1).
b) Incercati sa transformati gramatica in una echivalenta LL(1) aplicand factorizarea la stanga. Verificati daca noua gram. este LL(1).
c) Folosind un analizor descendent verificati daca secventa:
if c then if c then stmt else stmt endif endif
(sau echivalenta ei scrisa cu a,b,c,i)
apartine limbajului generat de gramatica.

2. Fie gramatica ambigua:

$S \rightarrow$ if c then S else S | if c then S | stmt

Daca inlocuim: ***if c then*** cu ***a***, ***else*** cu ***b*** si ***stmt*** cu ***i*** avem:

S -> a S
S -> a S b S
S -> i

Pentru una dintre cele 2 gramatici de mai sus:
a) Verificati daca gramatica este LL(1).
b) Incercati sa transformati gramatica in una echivalenta LL(1) aplicand factorizarea la stanga. Verificati daca gramatica obtinuta este LL(1).
c) Discutati, impreuna cu cadrul didactic, cum se poate modifica tabelul de analiza astfel incat sa se elimine conflictele.
d) Folosind analizorul LL(1), verificati daca secventa:
if c then if c then stmt else stmt
(sau echivalenta ei scrisa cu a,b,c,i)
apartine limbajului generat de gramatica.

3. Fie gramatica:

E -> T + E | T
T -> T * F | F
F -> (E) | a

a) Verificati daca gramatica este LL(1).
b) Incercati sa transformati gramatica in una echivalenta LL(1) aplicand factorizarea la stanga. Verificati daca gramatica obtinuta este LL(1).
c) Folosind un analizor descendent verificati daca secventa:

a + a

apartine limbajului generat de gramatica.

4. Fie gramatica:

List -> id
List -> id sep List

a) Verificati daca gramatica este LL(1).
b) Incercati sa transformati gramatica in una echivalenta LL(1) aplicand factorizarea la stanga. Verificati daca gramatica obtinuta este LL(1).

5. Fie gramatica:

S -> begin Slist end
S -> stmt
SList -> S
SList -> S ; SList

a) Verificati daca gramatica este LL(1).
b) Incercati sa transformati gramatica in una echivalenta LL(1) aplicand factorizarea la stanga. Verificati daca gramatica obtinuta este LL(1).

4 Analiza sintactica LR* (LR(0), SLR, LR(1),LALR)

1. Fie gramatica:

S → AA
A → aA
A → b

a) Verificati daca este LR(0)
b) Verificati daca este LR(1)
c) Verificati daca este LALR
d) Folosind un analizor de tip LR(K) , verificati daca secventa: abab apartine limbajului generat de gramatica.

Analizorul va fi ales in functie de raspunsul la intrebarile de mai sus.

2. Fie gramatica:

S -> if c then S endif
S -> if c then S else S endif
S -> stmt

Daca inlocuim: ***if c then*** cu ***a***, ***else*** cu ***b*** , ***endif*** cu ***c*** , si ***stmt*** cu ***i*** avem:

S -> a S c
S -> a S b S c
S -> i

Pentru una dintre cele 2 gramatici de mai sus:
a) Verificati daca gramatica este LR(0).
b) Verificati daca este SLR.
c) Este LR(1)?
d) Folosind un analizor de tip LR(K) , verificati daca secventa:

if c then if c then stmt else stmt endif endif

(sau echivalenta ei) apartine limbajului generat de gramatica.

Analizorul va fi ales in functie de raspunsul la intrebarile de mai sus.

3. Fie gramatica ambigua:

S → if c then S else S | if c then S | stmt

Daca inlocuim: ***if c then*** cu ***a***, ***else*** cu ***b*** si ***stmt*** cu ***i*** avem:

S -> a S
S -> a S b S
S -> i

Pentru una dintre cele 2 gramatici de mai sus, verificati daca este LR(1).

4. Dati gramatica echivalenta neambigua a gramaticii pentru structura if-then-else (gram. ambigua data in problema anterioara)

4. Dati gramatica echivalenta neambigua a gramaticii pentru structura if-then-else (gram. ambigua data in problema anterioara)
Dati un cuvant care, in gramatica originala ambigua, poate sa corespunda la doua structuri if cu proprietatea:
a) cel mai interior if **contine** ramura "else", iar cel exterior **nu contine** ramura "else"
b) cel mai interior if **nu contine** ramura "else", iar cel exterior **contine** ramura "else"
Verificati, folosind gramatica neambigua echivalenta si analiza LR(1) ca acel cuvant dat anterior este generat de gramatica echivalenta neambigua.

5. Fie gramatica:

E -> E + T
E -> T
T -> T * F
T -> F
F -> id
F -> (E)

Verificati daca gramatica este LR(1)

6. Fie gramatica:

S -> begin SL end
S -> stmt
SL -> S
SL -> S semicolon SL

a) Verificati daca gramatica este LR(0).
b) Este SLR?
c) Folosind un analizor de tip LR(K) , verificati daca secventa:
begin stmt semicolon stmt end
Analizorul va fi ales in functie de raspnsul la intrebarile de mai sus.

5 Gramatica de precedenta simpla. Exemplu

Mai exista (si vor fi studiate doar la seminar) si alte tipuri de analiza sintactica ascendenta. Dintre acestea, vom vedea doar cum se lucreaza cu (/un exemplu de) gramatici de precedenta simpla.

- Analiza ascendenta
- Depisteaza limita dreapta si a celei stanga pentru a face o reducere
  Se folosesc relatiile $<\bullet$, $\bullet>$, $=\bullet$ (relatii de precedenta)

Relatii de precedenta Wirth-Weber
$R_{<\bullet} \subset (N \cup \Sigma \cup \{\$\}) \times (N \cup \Sigma \cup \{\$\})$
$R_{=\bullet} \subset (N \cup \Sigma \cup \{\$\}) \times (N \cup \Sigma \cup \{\$\})$
$R_{\bullet>} \subset (N \cup \Sigma \cup \{\$\}) \times (\Sigma \cup \{\$\})$

$X =\bullet Y$: $A \rightarrow \alpha XYg \in P$
$X <\bullet Y$: $A \rightarrow \alpha XB\gamma \in P$, $B =>^{+} Y\gamma$
$X \bullet> a$: $A \rightarrow \alpha BY\gamma \in P$, $B =>^{+} \gamma X$, $Y =>^{*} a\delta$
$\$ <\bullet X$: $S =>^{+} X\alpha$
$X \bullet> \$$: $S =>^{+} \alpha X$

Definitie:
gramatica de precedenta simpla
este o gramatica indep. de context proprie
- unic invertibila:
  nu exista 2 reguli de productie cu acelasi membru drept
- intre oricare 2 simboluri exista cel mult o relatie de precedenta

Analizorul de precedenta simpla
- construieste tabelul de precedenta a operatorilor
- analizeaza o secventa de terminale

modelul stivei ~ LR
$<\bullet$ si $=\bullet$ - deplasare
$\bullet>$ - reducere $Y <\bullet X1 =\bullet \ldots =\bullet Xi .> Z$
$A \rightarrow X1 \ldots Xi$

Exemplu:

$S \rightarrow aSSb$

$S \rightarrow c$

	S	**a**	**b**	**c**	**$**
S	$\doteq$	$<$	$\doteq$	$<$	
a	$\doteq$	$<$		$<$	
b		$>$	$>$	$>$	$>$
c		$>$	$>$	$>$	$>$
$		$<$		$<$	

Cuvantul: accb ?$\in$L(G)

..

Seminar 10

3. LL(1)

3. $E \to T+E$
$E \to T$
$T \to T*F$
$T \to F$
$F \to (E)$
$F \to a$

$T \to F*T$ — recursivă

a)

	First$_1$	Follow$_1$
E	(, a	$,)
T	(, a	+, $,), *
F	(, a	+, $,), *

$First_1(T+E) \cap First_1(T) = \{(, a\} \neq \emptyset \Rightarrow$ nu e LL(1)

b)

$E \to TE'$ (1)
$E' \to +E$ (2)
$E' \to \varepsilon$ (3)
$T \to FT'$ (4)
$T' \to *T$ (5)
$T' \to \varepsilon$ (6)
$F \to (E)$ (7)
$F \to a$ (8)

	First$_1$	Follow$_1$
E	(, a	$,)
E'	+, ε	$,)
T	(, a	+, $,)
T'	*, ε	+, $,)
F	(, a	*, +, $,)

$First_1(+E) \cap$ ~~First~~ $Follow_1(E') = \emptyset$
$First_1(*T) \cap Follow_1(T') = \emptyset$
$First_1((E)) \cap First_1(a) = \emptyset$

$\Rightarrow$ este LL(1)

sau

cu tabel de analiză

$ - marcator de sf.

	+	*	(	a	)	$
E			TE',1	TE',1		
E'	+E,2				ε,3	ε,3
T			FT',4	FT',4		
T'	ε,6		*T,5		ε,6	ε,6
F			(E),7	a,8		
+	pop					
*		pop				
(			pop			
a				pop		
)					pop	
$						acc

c) $(w\$, S\$, \pi)$

↑ secvență ↑ stiva de intrare (S, simbol start)

$(a+a\$, E\$, \varepsilon) \vdash (a+a\$, TE'\$, 1) \vdash$

$(a+a\$, FT'E'\$, 14) \vdash (a+a\$, aT'E'\$, 148) \vdash$

$(+a\$, T'E'\$, 148) \vdash (+a\$, E'\$, 1486) \vdash$

$(+a\$, +E\$, 14862) \vdash (a\$, E\$, 14862) \vdash$

$(a\$, TE'\$, 148621) \vdash (a\$, FT'E'\$, 1486214) \vdash$

$(a\$, aT'E'\$, 14862148) \vdash (\$, T'E'\$, 1486214 8) \vdash$

$(\$, E'\$, 148621486) \vdash (\$, \$, 1486214863) \vdash acc$

4. ~~S → begin Slist end~~

List → id (1)

List → id sep List (2)

	First$_1$	Follow$_1$
List	id	$

F

F

	id	sep	$
List	id $_1$ / id sep list$_2$		
id	pop		
sep		pop	
$			acc

conflict => nu e LL(1)!

b) List → id L' (1)
L' → sep List (2)
L' → ε (3)

	First$_1$	Follow$_1$
List	id	$
L'	sep, ε	$

	id	sep	$
List	id L' (1)		
L'		sep List (2)	ε, 3
id	pop		
sep		pop	
$			acc

=> nu exista conflicte
=> este LL(1)

5. S → b SL e
S → ∧
SL → S
SL → S ; SL

	First$_1$	Follow$_1$
S	b, ∧	$, ;, e
SL	b, ∧	e

a) First$_1$(S) ∩ First$_1$(S ; SL) = {b, ∧} ≠ ∅ => nu e LL(1)

b) S → b SL e (1)
S → ∧ (2)
SL → S S' (3)
S' → ; SL (4)
S' → ε (5)

	First$_1$	Follow$_1$
S	b, ∧	$, ;, e
SL	b, ∧	e
S'	;, ε	e

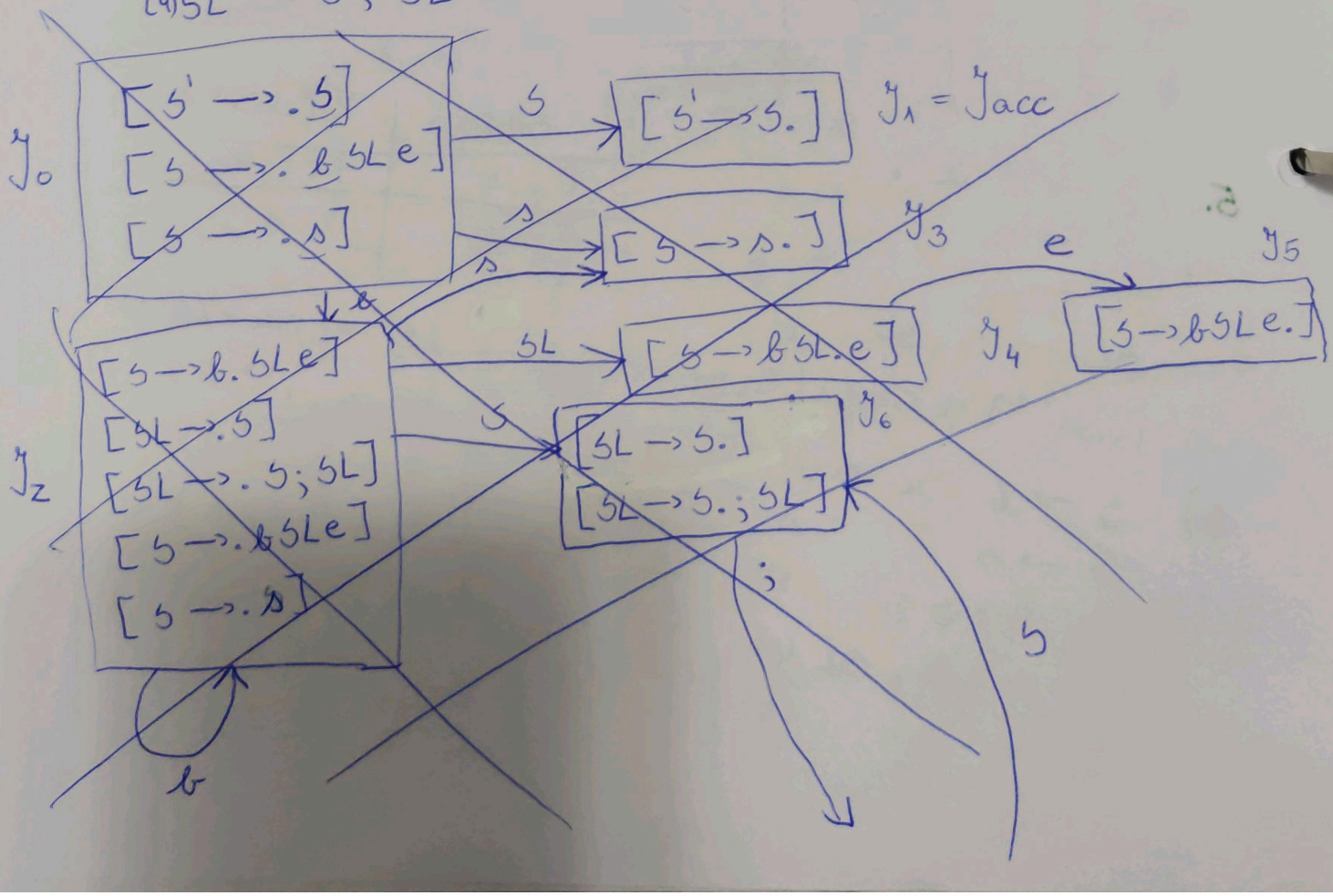
	b	e	∧	;	$
S	bSLe 1		∧ 2		
SL	SS' 3		SS' 3		
S'		;SL 4		;SL 4	
b	pop				
e		pop			
∧			pop		
;				pop	
$					acc

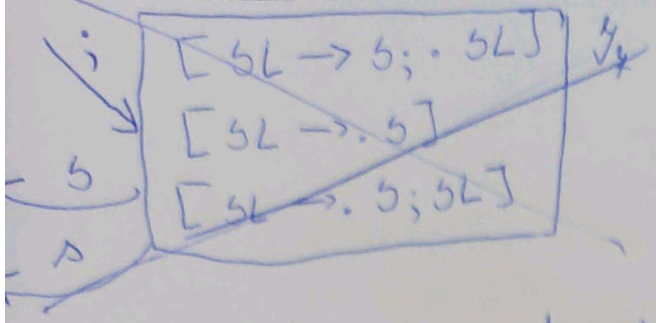
=> nu are conflicte
=> e LL(1)

4. LR

(0) S' → S ← adăugăm asta ca să obținem o gramatică îmbogățită

6. (1) S → b SL e
(2) SL → ∧
(3) SL → S
(4) SL → S ; SL





Λ - shift

cifră - cifra regulii (reduce)

	acțiune	S	SL	b	e	Λ	;
y_0	Λ	y_1		y_2		y_3	
y_1	acc						
y_2	Λ	y_6	y_4	y_2		y_3	
y_3	2						
y_4	Λ				y_5		
y_5	1						
y_6	3 / Λ						y_7
y_7	Λ	y_6	y_8	y_2		y_3	
y_8	4						

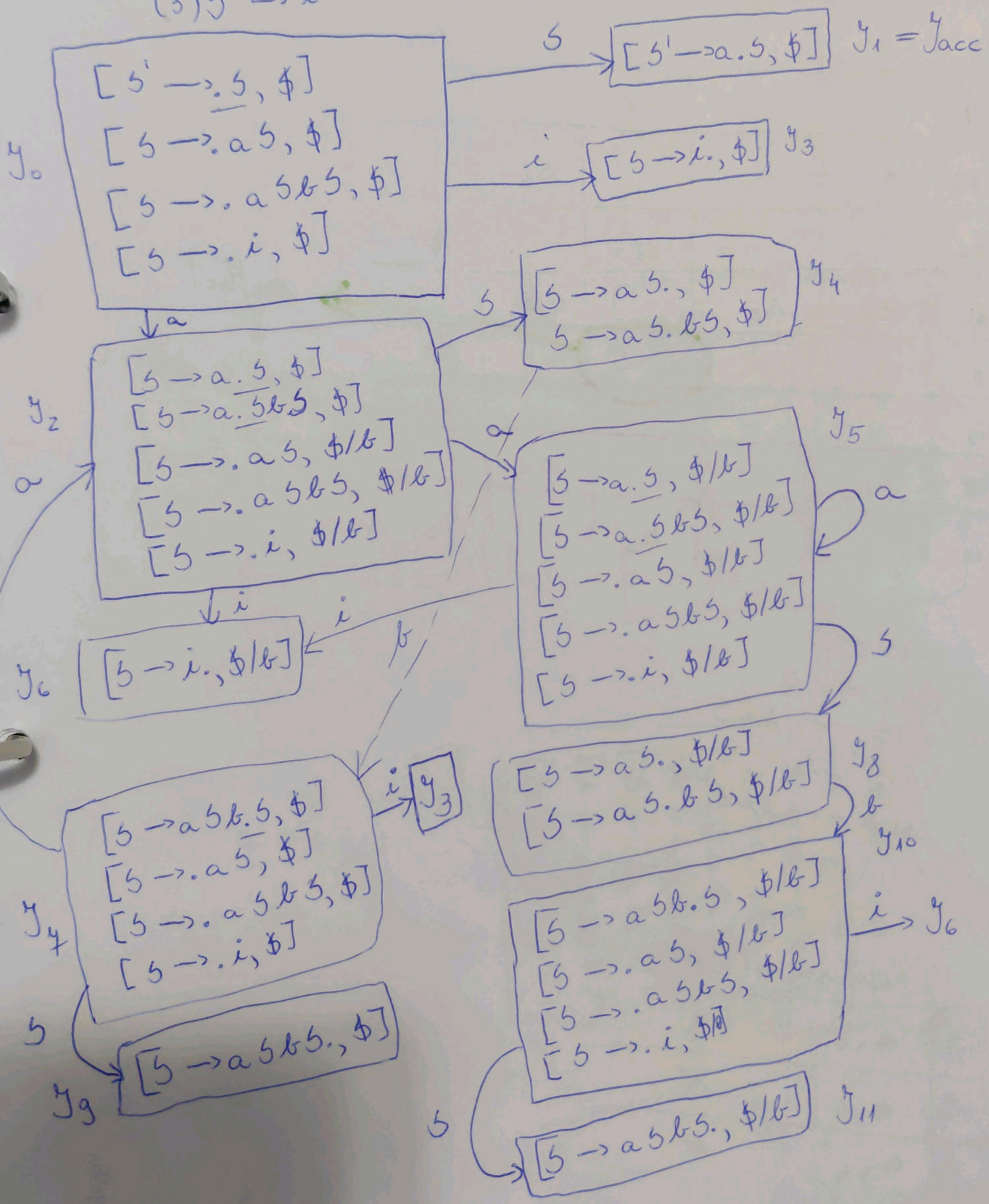
conflict => nu e LR(0)!

Seminar 12

4.3.

(0) $S' \to S$
(1) $S \to aS$
(2) $S \to aSbS$
(3) $S \to i$

	First 1	Follow 1
S'	a, i	\$
S	a, i	\$, b

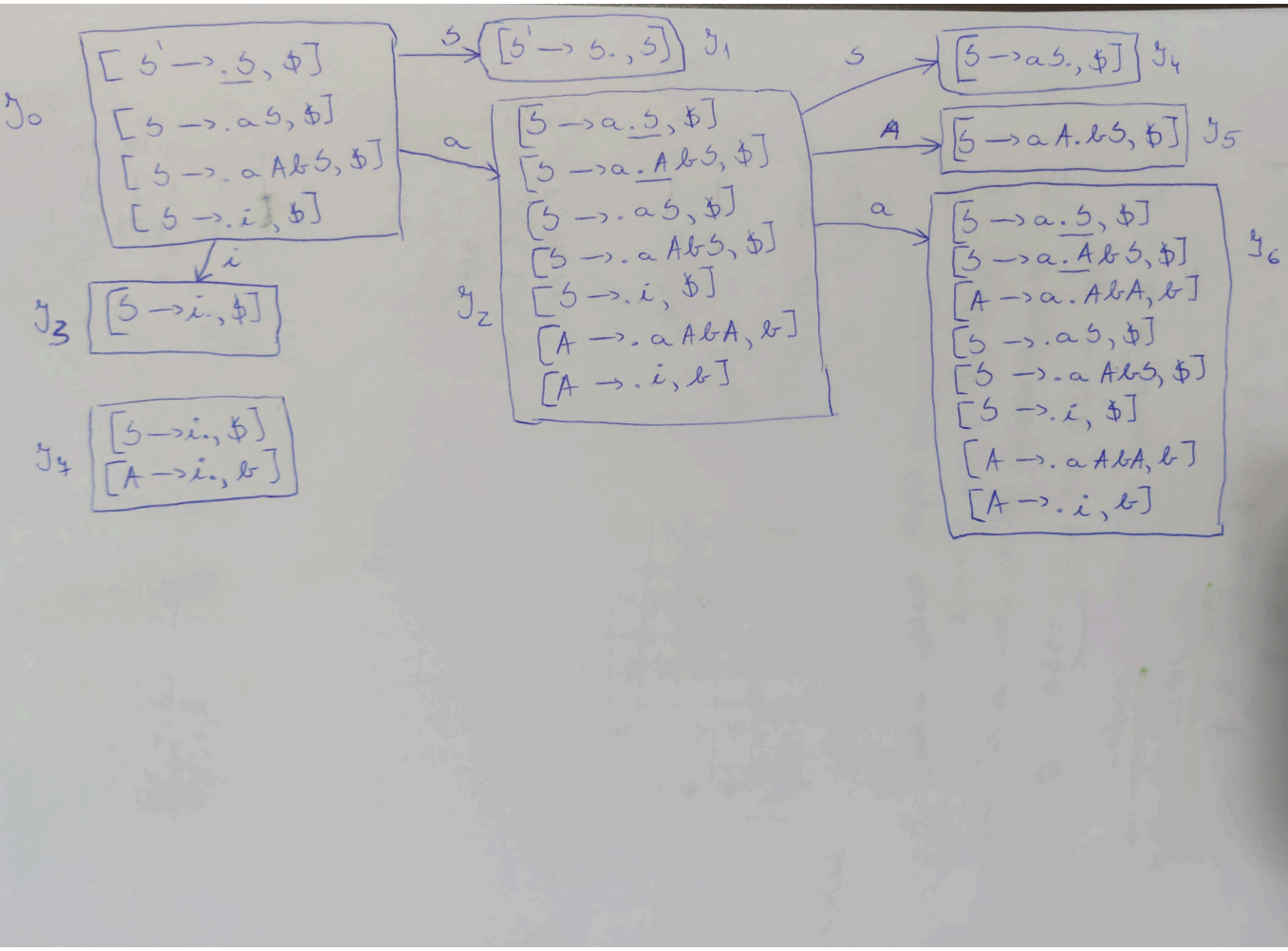


	S	a	b	i	$
I_0	s_1	s_2		s_3	
I_1					acc
I_2	s_4	s_5		s_6	
I_3					r_3
I_4			s_7		
I_5	s_8	s_5		s_6	
I_6			r_3		r_3
I_7	s_9	s_2		s_3	
I_8			r_1 / s_{10}		r_1
I_9					r_2
I_{10}	s_{11}	s_5		s_6	
I_{11}			r_2		r_2

conflict
⇓
non è LR(1)

(0) $S' \rightarrow S$
(1) $S \rightarrow aS$
(2) $S \rightarrow aAbS$
(3) $S \rightarrow i$
(4) $A \rightarrow aAbA$
(5) $A \rightarrow i$

	First$_1$	Follow$_1$
S'	a, i	$
S	a, i	$
A	a, i	b


I_0
[S' → .S, $]
[S → .aS, $]
[S → .aAbS, $]
[S → .i, $]
S
[S' → S., $]
I_1
a
[S → a.S, $]
[S → a.AbS, $]
[S → .aS, $]
[S → .aAbS, $]
[S → .i, $]
[A → .aAbA, b]
[A → .i, b]
I_2
i
[S → i., $]
I_3
S
[S → aS., $]
I_4
A
[S → aA.bS, $]
I_5
a
[S → a.S, $]
[S → a.AbS, $]
[A → a.AbA, b]
[S → .aS, $]
[S → .aAbS, $]
[S → .i, $]
[A → .aAbA, b]
[A → .i, b]
I_6
[S → i., $]
[A → i., b]
I_7

5. $S' \to E$ (0)
$E \to E+T$ (1)
$E \to T$ (2)
$T \to T*F$ (3)
$T \to F$ (4)
$F \to id$ (5)
$F \to (E)$ (6)

	First	Follow
S'	id, (	$
E	id, (	$, +,)
T	id, (	$, +,), *
F	id, (	$, +,), *

$[S' \to .E, \$]$
$[E \to .E+T, \$/+]$
$[E \to .T, \$/+]$
$[T \to .T*F, \$/+/*]$
$[T \to .F, \$/+/*]$

(5) 1. $S \to aSSb$
$S \to c$

	S	a	b	c	$
S	$\doteq$	$\lessdot$	$\doteq$	$\lessdot$	
a	$\doteq$	$\lessdot$		$\lessdot$	
b		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$
c		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$
$		$\lessdot$		$\lessdot$	

} doar $\lessdot$
doar $\gtrdot$

$\doteq$ $\lessdot$ $\gtrdot$
Rune?

$\lessdot \ldots \gtrdot$ când ai așa ceva, trebuie reduce

SS - există într-o reg. $\Rightarrow \doteq$

Sa - se poate deriva la dr. ceva ca să obținem?
- da $aSSb \Rightarrow aSaSSbb$ $\lessdot$

Sa - la fel ca mai sus, dar în partea dreaptă obligatoriu terminal
$\gtrdot$

$\$ a c c b \$ \Leftrightarrow \$ a c \doteq c b \$ \Leftrightarrow \$ a S c b \$ \Leftrightarrow$
$\lessdot \lessdot \gtrdot \gtrdot \gtrdot$ $\lessdot \doteq \lessdot \gtrdot \gtrdot$

$\$ a S S b \$ \Leftrightarrow \$ S \$ \Leftrightarrow$ acc
$\lessdot \doteq \doteq \doteq \gtrdot$

Seminar 13

Gramatici de atribute

1. Fie limbajul: L = {$a^n b^n c^n$ | n - natural}
   Dati o gramatica de atribute care il genereaza.

2. Descrieti o gramatica de atribute care determina valorile expresiilor aritmetice in forma postfixata. EX: 5 6 + 5 *

3. Descrieti o gramatica de atribute care, pentru o expresie aritmetica data in forma infixata, determina expresia aritmetica in forma postfixata. EX: a + b * c

4. Dati o gramatica care genereaza secvente de 0 si 1 in care nr de 0 este egal cu nr. de 1.
   Atributati gramatica.
   Folositi atributul nr (asociat radacinii arborelui de derivare) cu semnificatia: numarul de 0 dintr-o secventa data.
   Aratati cum se evalueaza atributul pentru secventa 0101.
   (*Puteti adauga si alte atribute daca este necesar.*)

5. a) Dati o gramatica care genereaza secvente de 0 si 1 in care nr de 0 este par si nr. de 1 este par.

b) Fie atributele cu semnificatia: nr0 este numarul de 0 si nr1 este numarul de 1 dintr-o secventa data. Asociati atributele simbolurilor gramaticii si definiti regulile de evaluare.

c) Aratati cum se evalueaza atributul pentru secventa 0101

Cod intermediar

1. a) Fie secventa de instructiuni

```
A := B + C*D
B := B + C*D
D := B + C*D
```

Traduceti in cod intermediar cu 3 adrese, reprezentare cvadruple si apoi in cod intermediar cu 3 adrese reprezentare triplete.

2. a) Traduceti in cod intermediar :

```
a := 0
for i := 1 to 5 do begin
        a := a+1;
        i := i+1
end
```

b) Care este valoarea lui a la iesirea din secventa de instructiuni? Dar valoarea lui i?

Seminar 14

TRANSLATARE SI TRANSLATOARE

1. Definiti un translator finit M a.i.: T(M) = { (a^n, b^n) | n >=1) }

2. Definiti un translator finit M a.i.: T(M) = { (a^n, $(ab)^n$) | n >=1) }

3. Definiti un translator finit M a.i.: T(M) = { (a^m, b^n) | n >= m >= 1) }

4. Să se construiască un translator push-down care transformă o expresie aritmetică de la forma polonezã prefixatã în forma polonezã postfixatã. Presupunem cã expresia aritmeticã contine operatorii binari +, *, si operanzii simbolizati prin a.

5. Să se construiască un translator push-down care translateaza limbajul {a^n | n>=1} in {$a^n b^n$ | n >=1}, dupa criteriul stivei vide.

Translator finit

$M = (Q, \Sigma, D, \delta, q_0, F)$

- Q alfabetul stărilor;
- Σ alfabetul de intrare;
- D alfabetul de iesire;
- $q_0 \in Q$ stare inițială;
- $F \subseteq Q$ mulțimea stărilor finale;
- $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}_0(Q \times D^*)$

multimea partilor finite

Translatare definita de M:

$T(M) = \{(x, y) \mid x \in \Sigma^*, y \in D^*, (q_0, x, \varepsilon) \vdash^* (q, \varepsilon, y), q \in F\}$