

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

1. Fie urmatoarea gramatica ce descrie sintaxa unui mini-limbaj de programare:

```
<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>
```

Specificarea regulilor de formare a atomilor lexicali
ID (identificator) o vom face folosind expresii regulate:
ID : a (a | b | c) *

Dati doua “mini-programe” care sunt descrise de specificatiile date.

```
begin
ac = ab;
if (ac) then ab=aa
end.
begin abb = ac end.
```

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.

Elemente lexicale:

ID: f, a1, a2, a3

Operatori: ":=", "+"

CONST: 7, 11, 3, 5

Separatori: " ", ",", ":" , ";" , ". "

Cuvinte cheie/reservate: var, integer, begin, end

Structuri sintactice:

declarație var, atribuire, expresie, instr_compusă

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.

GIC:

```
<program> → <lista_declaratii> begin <lista_instructiuni> end.
<lista_declaratii> → <declaratie>
<lista_declaratii> → <declaratie> <lista_declaratii>
<declaratie> → var <lista_identificatori> : <tip_variabila>;
<lista_identificatori> → <identificator>
<lista_identificatori> → <identificator>, <lista_identificatori>
<tip_variabila> → integer
<lista_instructiuni> → <instructiune>
<lista_instructiuni> → <instructiune>; <lista_instructiuni>
<instructiune> → ID := <expresie>
<expresie> → CONST | ID
<expresie> → <expresie> <operator> <expresie>
<operator> → +
```

- c) Scrieti un program diferit de cel de mai sus care respecta descrierile date.

<pre> <program> → <lista_declaratii> begin <lista_instructiuni> end. <lista_declaratii> → <declaratie> <lista_declaratii> → <declaratie> <lista_declaratii> <declaratie> → var <lista_identificatori> : <tip_variabila>; <lista_identificatori> → <identificator> <lista_identificatori> → <identificator>, <lista_identificatori> <tip_variabila> → integer <lista_instructiuni> → <instructiune> <lista_instructiuni> → <instructiune>;<lista_instructiuni> <instructiune> → ID:= <expresie> <expresie> → CONST ID <expresie> → <expresie> <operator> <expresie> <operator> → + </pre>	<pre> var a, b: integer; begin a:=1; b:=a + 1 end. </pre>
--	---

- 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, descripti 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:
- Tabel sortat lexicographic
 - Arbore binar de cautare
 - Tabela de dispersie
(Alegeti o functie de dispersie simplu de calculat, dimensiunea tabelei poate fi 11.)

Poate aveti nevoie:
 $\text{ASC}('a') = 97$
 $\text{ASC}('0') = 48$

tabelă ordonată lexicografic (alphanumeric)

Atom lexical	Cod Atom	Programul (fis. de intrare)	FIP Forma Internă a Programului		TS Tabela de Simboluri (ID)		
			Cod Atom	COD TS	Simbol (ID)	Cod TS	
					a1	2	
ID	1		var	6	-	TS Tabela de Simboluri (CONST)	
CONST	2		f	1	1		
begin	3		,	10	-		
end	4		a1	1	2		
.	5		,	10	-		
var	6		a2	1	3		
;	7		,	10	-		
:	8		a3	1	4		
integer	9		:	8	-		
,	10		integer	9	-		
:=	11		;	7	-		
+	12		begin	3	-		
			a1	1	2		
			:=	11	-		
			7	2	1		
			;	7	-		
			a2	1	3		
			:=	11	-		
			11	2	2		
			;	7	-		
			a3	1	4		
			:=	11	-		
			a1	1	2		
			+	12	-		
			a2	1	3		
			+	12	-		
			3	2	3		
			;	7	-		
			f	1	1		
			:=	11	-		
			5	2	4		
			end	4	-		
			.	5	-		

Arbore binar de căutare

Atom lexical	Cod Atom	Programul (fis. De intrare)	FIP Forma Internă a Programului		TS Tabela de Simboluri (ID)	
			<u>Cod Atom</u>	<u>COD TS</u>	Simbol (ID)	Cod TS
					<pre> graph TD f1((f,1)) --- a12((a1,2)) a12 --- a23((a2,3)) a23 --- a34((a3,4)) </pre>	
ID	1		var	6	-	
CONST	2		f	1	1	
begin	3		,	10	-	
end	4		a1	1	2	
.	5		,	10	-	
var	6		a2	1	3	
;	7		,	10	-	
:	8		a3	1	4	
integer	9		:	8	-	
,	10		integer	9	-	
:=	11		;	7	-	
+	12		begin	3	-	
			a1	1	2	
			:=	11	-	
			7	2	1	
			;	7	-	
			a2	1	3	
			:=	11	-	
			11	2	2	
			;	7	-	
			a3	1	4	
			:=	11	-	
			a1	1	2	
			+	12	-	
			a2	1	3	
			+	12	-	
			3	2	3	
			;	7	-	
			f	1	1	

tabelă de dispersie cu dim. 11

funcția de dispersie = ascii char din mijloc (sau cel mai mare de lângă mijloc) mod 11

Atom lexical	Cod Atom	Programul (fis. De intrare)	FIP Forma Internă a Programului		TS Tabela de Simboluri (ID)	
			Cod Atom	COD TS	Simbol (ID)	Cod TS = poz. Din tabelă
					a3	0
					f	1
						2
						3
						4
						5
						6
						7
						8
					a1	9
					a2	10
ID	1		var	6	-	
CONST	2		f	1	3	
begin	3		,	10	-	
end	4		a1	1	9	
.	5		,	10	-	
var	6		a2	1	10	
;	7		,	10	-	
:	8		a3	1	0	
integer	9		:	8	-	
,	10		integer	9	-	
:=	11		;	7	-	
+	12		begin	3	-	
			a1	1	9	
			:=	11	-	
			7	2	0	
			;	7	-	
			a2	1	10	
			:=	11	-	
			11	2	5	
			;	7	-	
			a3	1	0	
			:=	11	-	
			a1	1	9	
			+	12	-	
			a2	1	10	
			+	12	-	
			3	2	7	
			;	7	-	
			f	1	3	
