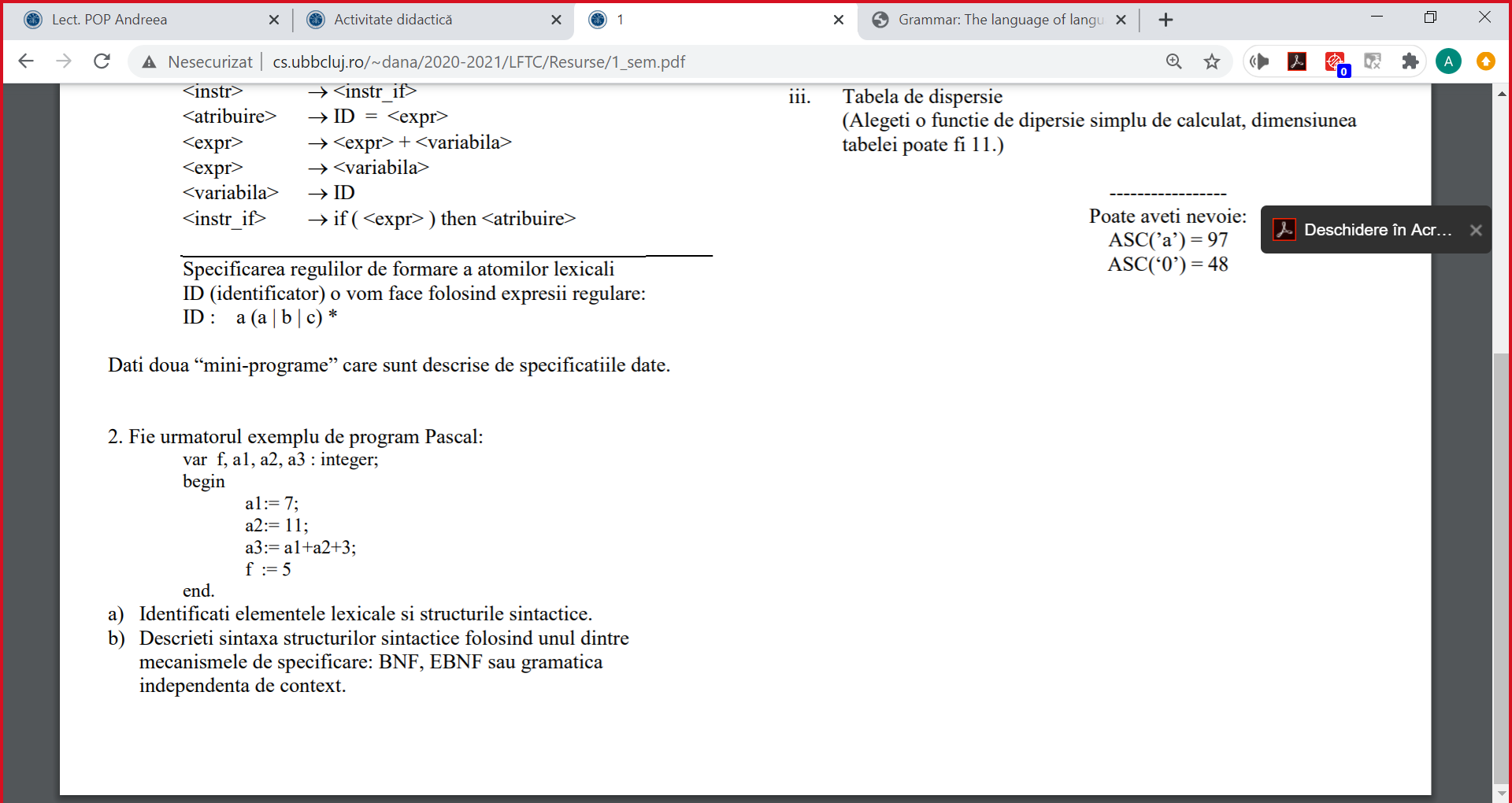


begin  
aa = ab;  
ac = ab + ac  
end.

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



**Elemente lexicale:**

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

Operatori: „:=”, ”:”, ”+”

ID: f, a1, a2, a3

CONST: 7, 11, 3, 5

Separatori: ”,”, ”;”, ”.”

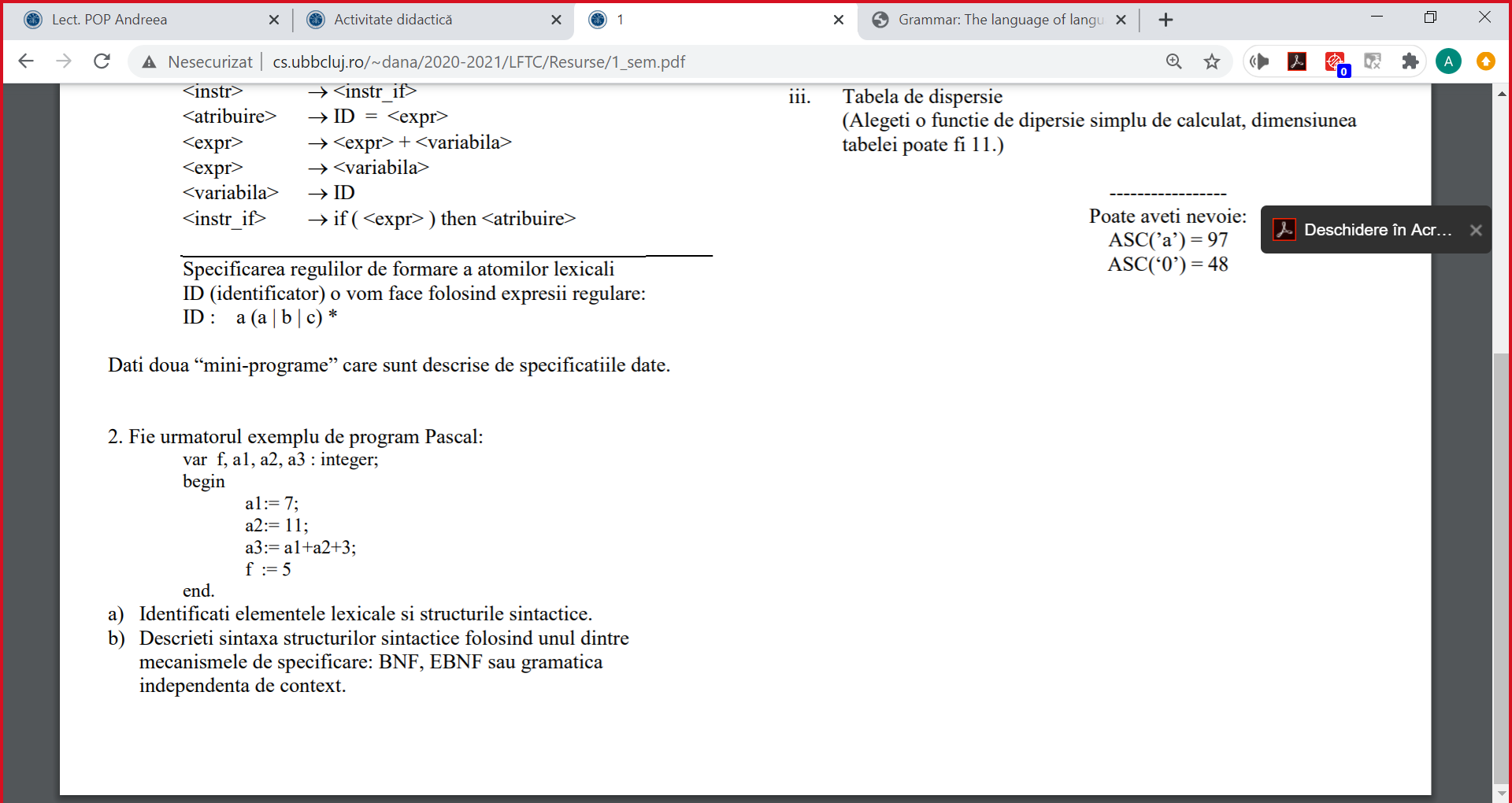
**Structuri sintactice:**

Declarație var

Atribuire

Expresie (aritmetică)

Corpul programului



**BNF:**

<program> ::= <decl\_var> begin <lista\_instr> end .

<decl\_var> ::= var <lista\_ID> : <tip> ;

<tip> ::= integer | string | float

bloc\_decl := “var” lista\_decl | “let” lista\_decl | “const” lista\_decl | “int” lista\_decl

lista\_decl := decl “;” | decl “,” lista\_decl

decl := lista\_id | lista\_id “=“ valoare

lista\_id := id| id “,” lista\_id | id “=“ lista\_id

id := id\_start | id\_start id\_cont

id\_start := cifra | litera | “$” | “\_”

id\_cont := id\_val | id\_val id\_cont

id\_val := id\_start | cifra

cifra := “0” | cifra\_nenula

cifra\_nenula := “1”|…|”9”

valoare := numar | obiect

obiect:= “{“ lista\_elem “}” “;”

lista\_elem := elem | lista\_elem

elem := id “:” valoare “,”

numar := cifra\_nenula | “0” | cifra\_nenula lista\_cifre

lista\_cifre := cifra | cifra lista\_cifre

lista\_instr := intrare|iesire| atribuire | selectie | ciclare

intrare := “readline” “(“ lista\_id “)” “;”

iesire := “console.log” “(“ lista\_id “)” “;”

atribuire := id “=“ expresie “;”

expresie := id |numar | expresie operator expresie

operator:= op\_arit | op\_rel

op\_arit :=“+” |”-“ | “mod” |”div”

op\_rel := “<“ | “>” | “<=“ | “>=“ | “==“ | “===“ | “!==“

selectie := “if” “(“ cond\_select “)” “{“ corp\_instr “}” | “if” “(“ cond\_select “)” “{“ corp\_instr“}” “else” “{“ corp\_instr “}”

cond\_select:= expresie | expresie operator expresie

corp\_instr:= lista\_instr | lista\_instr corp\_instr

ciclare:=“for” “(“ bloc\_decl “;” expresie “;” op\_inc\_dec”)” “{“ corp\_instr “}”

op\_inc\_dec := op\_inc | op\_dec

op\_inc := “++” id | id “++”

op\_dec:=“—“ id | id “—“

<lista\_ID> ::= ID

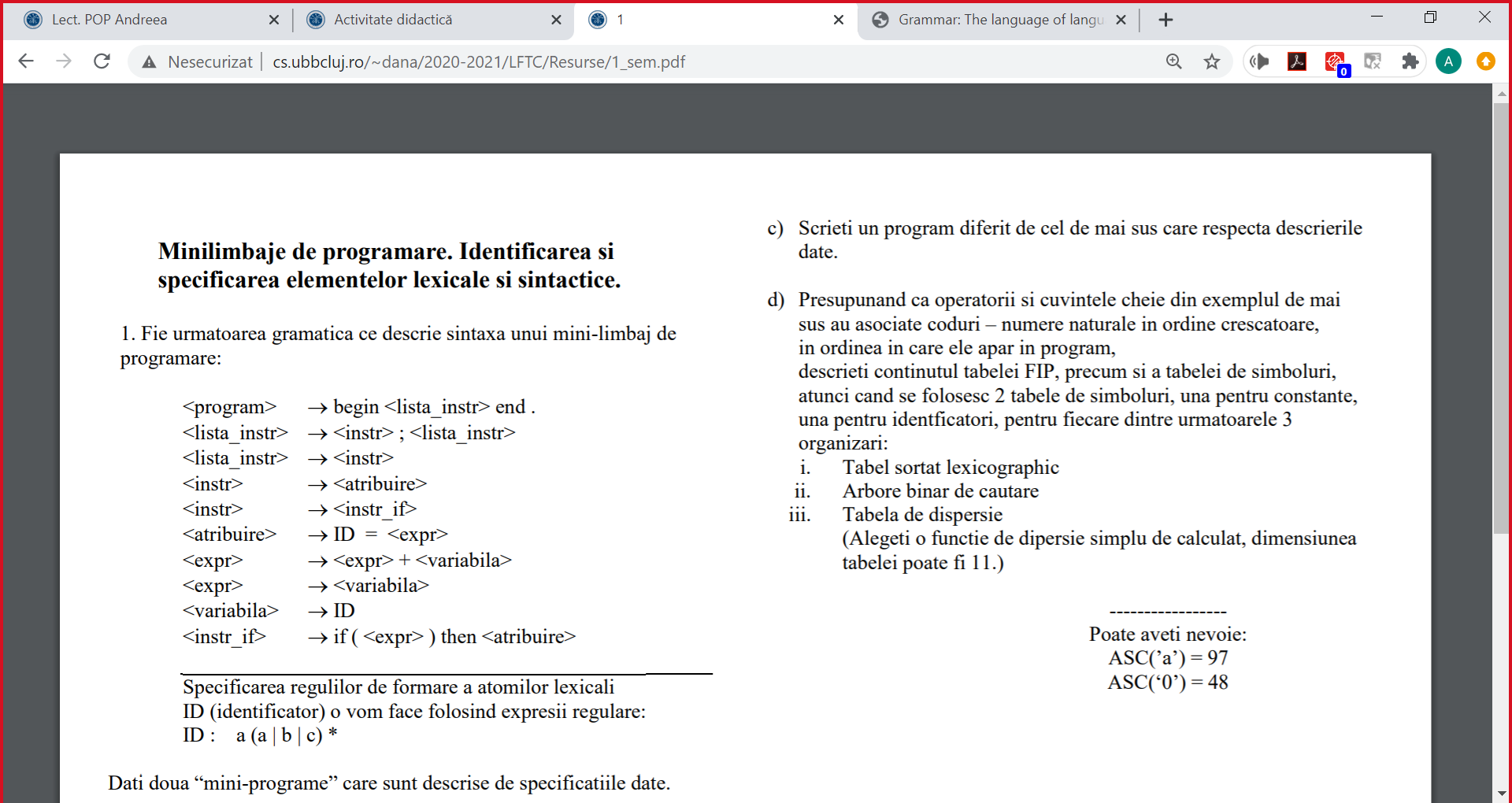
<lista\_ID> ::= ID , <lista\_ID>

<lista\_instr> ::= ID “:=” <expr>

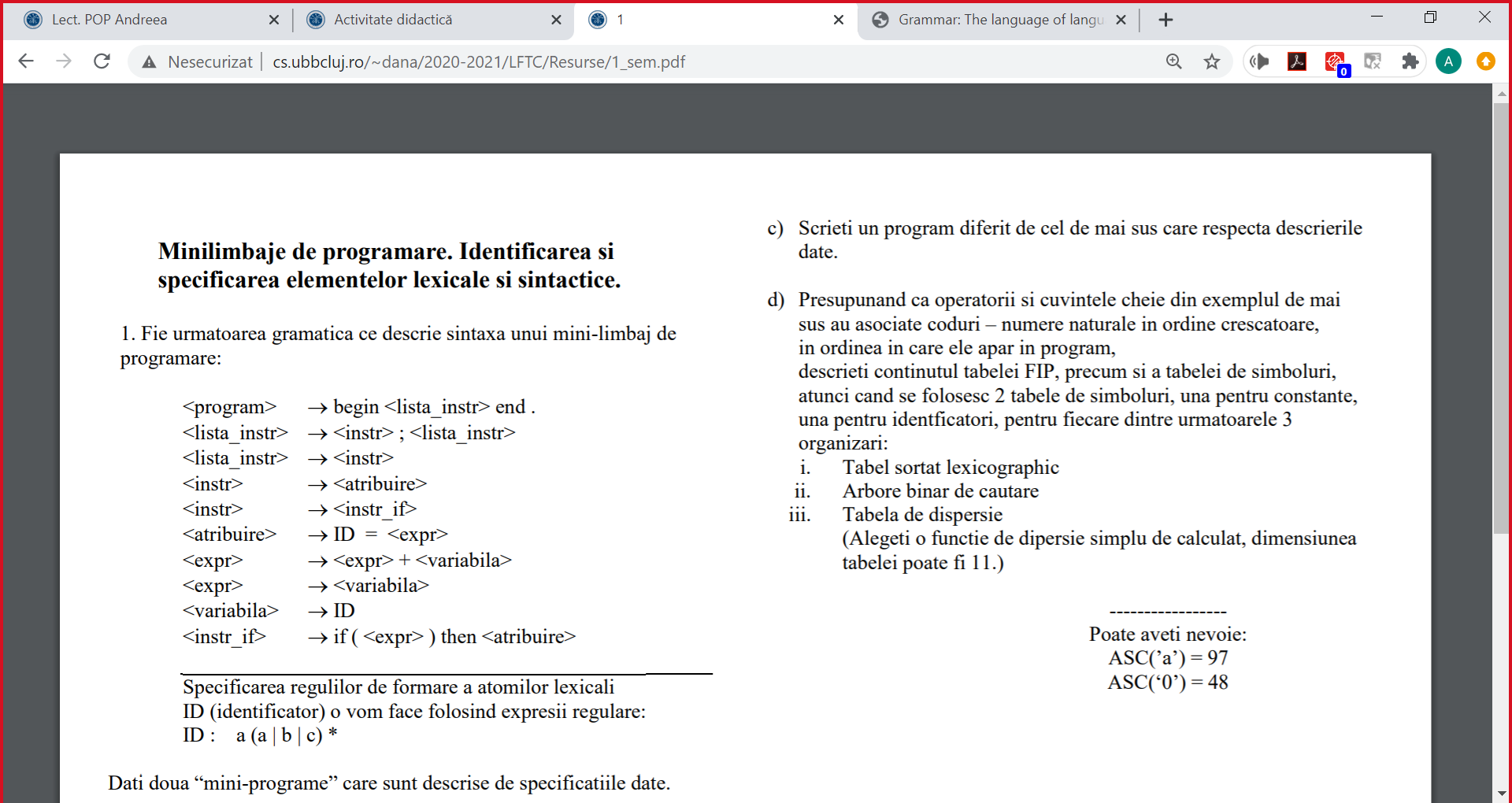
<lista\_instr> ::= ID “:=” <expr> ; <lista\_instr>

<expr> ::= ID | ID <op> <expr> | CONST

<op> ::= + | - | \* | /



|  |  |
| --- | --- |
| <program> ::= <decl\_var> begin <lista\_instr> end .  <decl\_var> ::= var <lista\_ID> : <tip> ;  <tip> ::= integer | string | float  <lista\_ID> ::= ID  <lista\_ID> ::= ID , <lista\_ID>  <lista\_instr> ::= ID “:=” <expr>  <lista\_instr> ::= ID “:=” <expr> ; <lista\_instr>  <expr> ::= ID | ID <op> <expr> | CONST  <op> ::= + | - | \* | / | var a, b, c : integer;  begin  b:=3;  a:= b;  c:=a+7  end. |



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| tabelă ordonată lexicofgrafic (alphabetic) | | | | | | | |
| **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  a2  a3  f | 2  3  4  1 | |
| ID  CONST  var  ,  :  ;  :=  .  +  begin  end  integer  string  float  -  \*  / | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | var  f  ,  a1  ,  a2  ,  a3  :  integer  ;  begin  a1  :=  7  ;  a2  :=  11  ;  a3  :=  a1  +  a2  +  3  ;  f  :=  5  end  . | 3  1  4  1  4  1  4  1  5  12  6  10  1  7  2  6  1  7  2  6  1  7  1  9  1  9  2  6  1  7  2  11  8 | -  1  -  2  -  3  -  4  -  -  -  -  2  -  10  -  3  -  11  -  4  -  2  -  3  -  12  -  1  -  13  -  - | TS  Tabela de Simboluri (CONST) | | |
| Simbol (CONST) | | Cod TS |
| 11  3  5  7 | | 11  12  13  10 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Arbore binar de căutare | | | | | | | |
| **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 | |
|  | | |
| ID  CONST  var  ,  :  ;  :=  .  +  begin  end  integer  string  float  -  \*  / | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | var  f  ,  a1  ,  a2  ,  a3  :  integer  ;  begin  a1  :=  7  ;  a2  :=  11  ;  a3  :=  a1  +  a2  +  3  ;  f  :=  5  end  . | 3  1  4  1  4  1  4  1  5  12  6  10  1  7  2  6  1  7  2  6  1  7  1  9  1  9  2  6  1  7  2  11  8 | -  1  -  2  -  3  -  4  -  -  -  -  2  -  10  -  3  -  11  -  4  -  2  -  3  -  12  -  1  -  13  -  - | TS  Tabela de Simboluri (CONST) | | |
| Simbol (CONST) | | Cod TS |
|  | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| tabelă de dispersie cu dim. 11  funcția de dispersie = (suma codurilor ascii a tuturor caracterelor ) 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 | |
| f  a1  a2  a3 | 0  1  2  3  4  5  6  7  8  9  10 | |
| ID  CONST  var  ,  :  ;  :=  .  +  begin  end  integer  string  float  -  \*  / | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | var  f  ,  a1  ,  a2  ,  a3  :  integer  ;  begin  a1  :=  7  ;  a2  :=  11  ;  a3  :=  a1  +  a2  +  3  ;  f  :=  5  end  . | 3  1  4  1  4  1  4  1  5  12  6  10  1  7  2  6  1  7  2  6  1  7  1  9  1  9  2  6  1  7  2  11  8 | -  3  -  4  -  5  -  6  -  -  -  -  4  -  0  -  5  -  10  -  6  -  4  -  5  -  7  -  3  -  9  -  - | TS  Tabela de Simboluri (CONST) | | |
| Simbol (CONST) | | Cod TS= poz. Din tabelă |
| 7  3  5  11 | | 0  1  2  3  4  5  6  7  8  9  10 |