

CR4S1 A

Drăghici Andreu - Monia

Introducere amănunțită pe sintaxă folosind  
generatoarea de parșere.

### 1. Gramatica:

program ::= varDecl / funDecl / empty

varDecl ::= type ID ';' / type ID '[' INTLITERAL  
'] ';' ;

type ::= INT / BOOL / void

funDecl ::= type ID parameters block

parameters ::= '(' ')' / '(' parameterList ')'

parameterList ::= formalDecl / parameterList ','

formalDecl

formalDecl ::= type ID

block ::= '{' declList stmtList '}'

declList ::= declList varDecl / empty



Draghici Andreea-Monia

CEUSIA

stmtList ::= stmtList stmt / empty

stmt ::= cin READ id ';' / cout WRITE  
exp ';' / subscriptExprn exp ';' / id exp ';' /  
if exp block / if exp block ELSE block  
/ WHILE exp block / RETURN exp ';' /  
RETURN ';' / fnCallStmt ';' /

exp ::= exp '+' exp / exp '-' exp / exp '\*'  
exp / exp '/' exp / '(' exp / exp AND AND exp  
/ exp OR OR exp / exp EQ EQ exp / exp NOT EQ exp  
/ exp '<' exp / exp '>' exp / exp LESS EQ exp /  
exp GREATER EQ exp / '-' atom / atom ;

atom ::= INT LITERAL / STRING LITERAL /  
TRUE / FALSE / '(' exp ')' / fnCallExprn /  
subscriptExprn / id

fnCallExprn ::= id / id actualList

fnCallStmt ::= id / id actualList

actualList ::= exp / actualList ';' exp



Ingheci Andreea-Monia

CE 4SIA

subscript Expr ::= id exp

id ::= ID

2. Simboluri terminale (simboluri sau cuvinte cheie)

INT, BOOL, VOID, EQ, FGEQ, OROR,

ANDAND, NOTEQ, GREATER, LESSEQ, TRUE,

FALSE, WHILE, IF, ELSE, RETURN, TYPE, +, -,

\*, /, !, <, >, :, (, ), [, ], ,, ', ID,

INTLITERAL, STRINGLITERAL, {, }, ., <<, >>,

CIN, COUT.

3. Simboluri non-terminale (simboluri ce reprezintă o parte din structura sintactică a limbajului).

program, varDecl, funDecl, parameters, funclist,  
funclDecl, block, decllist, stmtList, stmt, exp,  
atom, funCallExpr, funCallStmt, atndList,  
subscriptExpr.

4. Atributele legate de simbolurile non-terminale  
se referă la informațiile suplimentare care sunt  
stocate în nodurile arboreului sintactic abstract



Droghia Andreu - Maria

CEUSIA

pentru numite simboluri non-terminali din gramatica limbajului sursă.

Pentru simbolul non-terminal voided, atributul poate fi tipul variabilei (INT, BOOL sau VOID) și numele variabilei.

În cazul simbolului non-terminal fnded, atributul poate fi tipul returnat de funcție, numele funcției și parametrii opționali.

Pentru simbolul non-terminal exp, atributul poate fi valoarea sau tipul expresiei.

4. Gestionarea atributelor sintetizate și menținute în analizatoarele LR se referă la modul în care analizatorul utilizează atributelor pentru a construi obiectele sintactice al codului sursă.

Atributelor sintetizate se referă la atributelor care sunt calculate pe baza atributelor cuprins în simbol non-terminal.

Exemplu:

INT, BOOL, VOID poate reprezenta un atribut sintetic, fiind calculat pe baza atributelor cuprins în simbolului non-terminal voided.

Atributelor menținute se referă la atributelor

Onoahici Andreea-Maria

CRUSIA

care sunt propagate de la un simbol non-terminal la unul dintre copii săi.

Exemplu:

~~Un atribut menținut de la simbolul non-terminal~~  
~~Indice la simbolul non-terminal~~

Un atribut menținut care se poate fi utilizat  
se poate fi contextul semantic al unui block. Poate  
fi menținut de la simbolul non-terminal  
Indice la simbolul non-terminal block.