Droophici Andree- Moria CRUSIA

C-min us-min us Project

1. C -- grammon

1.1 Rules

program: program vorbeel/program Inded 1/4 empty */;

VonDecl: + pre iol 's' | + pre iol '['inTLITERAL']' 's' ;

type: inT /BOOL IVOID;

Ended: type id porometers block;

i'(' termals List')' (' ')' : crutemarag

furmals List: formal Decl I formals List', formal Decl;

formal Ded : type id;

black; '{ declist start List '}'

declist: de dist vor decl 1/2 empt y */;

: 1* y tyme * 1 1 trata trist trata : tois trata

Start: CIN READ id !;"

1 CIM READ id '[' exp'] !!

1 COUT WIRITE EXP 1;1

| subscript Expr '=' exp';'

1 101 = 1 exp 1; 1

1 i = 1(1 e+p1)1 black

lif 1(1exp1) block ELSE block

| XI HILE 1(1 exp 1) block

IRETURN exp 1; 1

IRETURN 1;1

```
Draphici Amolneeo-Morio
CRUSIA
   Ifmcolls+mt ';'
   ,
 exp: exp 1+1 exp.
     1 exp 1-1 exp
     1 exp 1 *1 exp
      gre 1/1 gre 1
      1 1 11 exp
       1 EXP ANDAND EXP
       1 exp DROR exp
       1 exp EZÉZ exp
       1 exp NOTES exp
        1 exp 1 < 1 . exp
        1 exp 1>1exp
        1 exp LESSEZ exp
        1 RXP GREATEREZ EXP
         11_1 stam
         1 stam
 Harritelal
       1 STRINGLITERAL
       1 TRUE
       1 FALSE
       11(1 expi)
       I forcoll Expo
       1 subscript Expr
```

```
Onoghici Andreeo-Morio
 CRUSIA
forcoll Expn: jol '(1 1)1
           ! io! '( 'oduallist ')'
forcoll stant: id ((1))
            1 iol "( 'octual List ')'
octualList: exp 1 octualList 1, exp i
Subscript Expn: id '['exp 13';
id; id;
1.2 Terminal or mbols
 1. 147
 2. BOOL
 3. 401 D
 4. Eg =) " ="
 6. ESEQ=)"=="
 6. OROR =) " 11"
 7. ANDAND > " 38"
 8. MOTES => " ! = "
 g. GREATER Eg =>">="
  10. LESSEZ =) " = = "
```

In a th

12. "-"

Dnoghici Andrea - Maria.

CRUSIA

13. 1 *"

14. 4/1

15. "!"

16. nc"

170 4>4

18. TRUE

19. FALSE

20. WHILE

21. i7

22. ELSE

23. RETURN

24. TYPE

(1 · 1) 25.

26. 4(1

24. ")"

11 [" 28.

29. "J"

30. 11 , 4

31. " 1"

32. 10

33. INTLÍTERAL

34. STRINGLITERAL

35. " { "

36. 431

37: " ."

1 Droghici Andrees - Monie 1 CE 451A

38, il ec "

39. ">>"

40. cin

41. COUT

1.3. Nou-terminal symbols

c brookows

< yor Decl >

< fm bed>

< porometers>

< tein downer >

< formal Decl >

< bleck>

< ded List >

< teil timte >

< +tm+>

< exp>

consto >

< fmcollExpn >

< fm Coll Stmt >

< toll lauto >

< subscript Expro

Broghici Andrua-Monia CE 451A 1.4 Input file example * Ordinary recursion */ export Apri Apr (bits 32 m) { ig, a sectid if m == 1 { return (1,1); I check D, P = SP1 (m-1); noturn (s+m, p*m);

2. Symtox-directed thouslotor

2.1. Eliminoting left recursion

Enomotica este successivo, si mu poote fi sonolizato in

Nonterminable < program>, < stantlist>, < exp>, < octual list> sunt losate necursive.

Aõugam potru via simboluri nuterminde c program 1) c startistes, capi, cochintes ia c triblantos, capi, ciquo, chailtantas

Onoghici Amobula - Monia CEUSIA Productii: < bushows > > < bushow 1> < branchow 1> -> bundrow T rouged 1 prugram 1 fabecl 1 * ptque *1 1 3 (< 1 mongrand > < stantList> > < stantList1> tunta 1 tail tunta <- < I teil tunta > 1 * ptyme * 1 3 <- < I toil timbe > CEXBUS -> , + , C EXBUS C EXD 1> COXP1> > 1-1 C 2xp1> C 2xp1> < 0xp 1> -> 1 x 1 = 8xp1> = 6xp1> < exp1> -> 1/1 = exp1> = exp1> < 6+b1> -> I HAIDHAID, C 6+b1> < 6+b1> < exp1> -> (Egfg' < exp1> < exp1> COXPIN -> 1 NOTEQ 1 E EXPINE EXPIN <expl>>> 'C'Cetpl>Cetpl> C exp1>-> 1> 1 < exp1> < exp1> < exp1> > LESSEQ' e exp1>c exp1>

Aloghici Andrew - Monio. CRUSIA Cexp1> -> +GREATEREZI CEXP1> < exp1> < (xp1> ->) ! | C (xp1> C exp1> -> 1-1 c otam> < < < > < < < < > < < < > < < < > < < < > < < < > < < < < > < < < < > < < < < > < < < < > < < < < > < < < < > < < < < > < < < < < > < < < > < < < > < < < > < < < > < < < > < < > < < > < < < > < < < > < < < > < < > < < < > < < < > < < < > < < < > < < > < < < > < < < > < < < > < < > < < < > < < < > < < < > < < > < < > < < < > < < < > < < < > < < < > < < < > < < < > < < < > < < > < < < > < < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < > < < > < < > < > < < > < > < > < > < < > < < > < < > < < > < < > < < > < < > < < > < < > < > < > < > < > < > < > < > < > < > < > < < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < > < < adual List > > < adual List 1> coducilist 1> > < exp 1> 100 dual List 1> 1, 1 < exp1> controlList1> > E Nous-gramatica genera La racedi limboj, care poste fi never - dop - gener. c program > -> < program 1> < program 1> -> program 1 varbed 1 program 1 Ambel 1 /* empty */ < Program 1> > E < vonDed>-> + pe fol '; ' 1 Jype id 'I' INT LITERAL 'J' 15'

Ctypes: inti BOOL

Droghici Andrewa-Moria CEUSIA AioV! <fm bed> > type id porameters black; c porometers > -> '(' ')' 1'(' formals List')' < formalized> -> type id; <block>=> 1 { declist stant List 13 }; Colection > -> declint von Decl 1 /* empty */ < 1 toistmte > < toistmte > c stant list 1> -> stant List 1 stant 1 /* empty */ 3 (- < 1 toil +mtc >

Droghici Amolhera-Monia CR451A

Otmt: Cix READ id'; / CIXI READ id '['exp']';' I COUT WHRITE EXP 1:1 |subsociptExpn= 1 = 1 = 1;1 lid = 1 exp1; lif (c) exp)) block lif (('exp')) block ELSE block | XIHILE ((1 exp1)) block I RETURN exp 1;1 IRETURN 131 I for Coll Stort 1:1

< exp> > < lxp1>

< exp1>-) 1+1 < exp1 > < exp1>

< exp1> -> 1-1 < exp1> < exp1>

< exp1> -> 1* '< exp1> < exp1>

< exp1> -> 1/1 = exp1> < exp1>

CEXPID -> , THPYMDIC EXPID < 6xb1>

COXPID => 1 HOTEG CEXPIDE EXPID

C exp1> -> '< ' < exp1> < exp1>

```
Dooghici Andrea Mania
CRUSIA
Cexp1> -> 1> 1 < cexp1> < cexp1>
Cexp1> -> LESSEZ 'Cexp1> < exp1>
Cexp1> -> | GREATE REQ! C.exp1> = exp1>
< exp1> -> 1 ! 1 cexp1>
< exp1> -> 1_1 cotous
< exp1 > -> &
Cotom> : INTLITERAL
         1 STRINGLITERAL
         ITRUE
        1 FALSE
        1 ((i exp 1))
        I Im Call Expn
        1 subscript Expn
        loi 1
+m Call Expn: id ((' 1)'
            I id ( octual List )
forcollStont: Ed (())
           1 id 10' octual List 1s'
```

Draghici Andrea - Monia CRUSIA < octual List > -> < octual List 1 > < 1949> <- <1 toil lanto > 1 control list1 > 1, cexp1> 3 C- < 1 trillauto > Foutswipt Exposi expos + octual List 1; exp1 C subscript Expn > : id '['exp']' 11: < bi> Simbolivile mon-terminale sut: < buddow > < bundlown> < yor Ded> < In Decl> < branches > < furnallist> < farmal Ded > < block > < decliat>

Droghici Anolnea-Monie CR USIA

< total

< start List 1>

< stant >

< exp>

c exp1>

Cotom>

< for Coll Expn>

< for call Stant>

< total List>

ToduckList 1>

< subscript Expr>

2.2 The First sits for man-terminal symbols

Decause in the recursive discent parsing the bookhood symbol is not used, it will be reploced with the trained symbols returned by the First() furtion.

First (< program>) = First (< program 1>)=

= First(dorbed>) U{E}

= First (z fmbed>) U{E}

First (< don Ded >) = First (< +ype >) = { id , i MILITERAL / 13/14

Droghici Amolnea - Moria CRUSIA

First (= type>) = { int, book, voiol}

first (< porometers>) = First (< furualsList>) = First (< furualsList>) = First (< furualset = lidy.

First (< block>) = first (< olellist>) UFirst (< n+m+List>)

First (cotmtlist>)= First (2 trut 4st1>) Uf & 3

Finat (z stant>) = { cin, cout, return, it, withite}

First (z exp>)={ ident, int-cust ?

First (< exp13) = {1 + 13

First (= exp(>) = 1 - 13

First (= exp>) = { 1 * 13

First (< exp>) = 2 / /13

Front (2 rul-operators) - 2'c')')', !==', 'c=', '>=', 1 = 1 %