program ::= app identifier body

body ::= [ decl-list] start stmt-list stop

decl-list ::= decl {";" decl}

decl ::= type ident-list

ident-list ::= identifier {"," identifier}

type ::= integer | real

stmt-list ::= stmt {";" stmt}

stmt ::= assign-stmt | if-stmt | while-stmt | repeat-stmt | read-stmt | write-stmt

assign-stmt ::= identifier ":=" simple\_expr

if-stmt ::= if condition then stmt-list else-stmt

else-stmt ::= end | else stmt-list end

condition ::= expression

repeat-stmt ::= repeat stmt-list stmt-suffix

stmt-suffix ::= until condition

while-stmt ::= stmt-prefix stmt-list end

stmt-prefix ::= while condition do

read-stmt ::= read "(" identifier ")"

write-stmt ::= write "(" writable ")"

writable ::= simple-expr | literal

expression ::= simple-expr | simple-expr relop simple-

expr

simple-expr ::= term simple-expr’

simple-expr’ ::= λ | addop term simple-expr’

term ::= factor-a term’

term’ ::= λ | mulop factor-a term’

fator-a ::= factor | "!" factor | "-" factor

factor ::= identifier | constant | "(" expression ")"

relop ::= "=" | ">" | ">=" | "<" | "<=" | "!="

addop ::= "+" | "-" | "||" mulop ::= "\*" | "/" | "&&"

constant ::= integer\_const | float\_const

integer\_const ::= digit {digit}

float\_const ::= digit {digit} “.” digit {digit}

literal ::= " {" {caractere} "}"

identifier ::= letter |“\_” {letter | digit | “\_”}

letter ::= [A-Za-z]

digit ::= [0-9]

caractere ::= um dos 256 caracteres