

Relatório de Alteração da Gramática

Gramática Original

Gramática Alterada

Inicia em <module>

<module> ::= "MODULE" ID ";" <declarations> "BEGIN" <statements> "END" ID "."

<declarations> ::= [<constdecl>] [<vardecl>] { <procdecl> ";" }

<declarations> ::= decl_list

<decl_list> ::= decl_list <procdecl> ";" | <const_opt> <var_opt>

<const_opt> ::= constdecl | ϵ

<var_opt> ::= vardecldecl | ϵ

<constdecl> -> "CONST" { ID "=" <expression> ";" }

<constdecl> ::= <constdecl> ID "=" <expression> ";" | "CONST"

<vardecl> -> "VAR" { <idlist> ":" <vartype> ";" }

<vardecl> ::= <vardecl> <idlist> ":" <vartype> ";" | "VAR"

<idlist> -> ID { "," ID }

<idlist> ::= <idlist> "," ID | ID

<vartype> ::= "BOOLEAN" | "INTEGER" | "REAL" | ID | <arraytype>

<arraytype> ::= "ARRAY" [<expression>] "OF" <vartype>

<arraytype> ::= "ARRAY" <arraytype_op> "OF" <vartype>

<arraytype_op> ::= <expression> | ϵ

<procdecl> ::= <procheader> ";" [<procbody>]

<procdecl> ::= <procheader> ";" <procdecl_op>

<procdecl_op> ::= <procbody> | ϵ

<procheader> ::= "PROCEDURE" ID <formalpars> [":" <vartype>]

<procheader> ::= "PROCEDURE" ID <formalpars> <procheader_op>

<procheader_op> ::= ":" <vartype> | ϵ

<procbody> ::= <declarations> "BEGIN" <statements> "END" ID

<formalpars> ::= "(" [<fpsection> { ";" <fpsection> }] ")"

<formalpars> ::= "(" <formalpars_op> ")"

<formalpars_op> ::= <formalpars_list> | ϵ

<formalpars_list> ::= <formalpars_list> ";" <fpsection> | <fpsection>

<fpsection> ::= ["VAR"] <idlist> ":" <vartype>

<fpsection> ::= <fpsection_op> <idlist> ":" <vartype>

<fpsection_op> ::= "VAR" | ϵ

```

<statements>      ::= <statement> { ";" <statement> }
<statements_list> ::= <statements_list> "," <statement> | <statement>

<expression>      ::= <expression> "OR" <andexp> | <andexp>
<andexp>           ::= <andexp> "AND" <relexp> | <relexp>
<relexp>           ::= <aritexp> <relop> <aritexp> | <aritexp>
<aritexp>          ::= <aritexp> <addlop> <term> | <term>
<term>             ::= <term> <multop> <factor> | <factor>
<factor>           ::= ["+" | "-" | "NOT" | "(" <vartype> ")"] <primary> | <proccall>
<factor>           ::= ADDLOP <primary> | "NOT" <primary> | "(" <vartype> ")" <primary> | <primary>

<primary>          ::= "(" <expression> ")" | <literal> | <variable>
<literal>          ::= BOOLEAN | INTEGER
<proccall>         ::= ID <actualpar>
<actualpar>        ::= "(" [ <expression> { "," <expression> } ] ")"
<actualpar>        ::= "(" <actualpar_op> ")"
<actualpar_op>     ::= <actualpar_list> | €
<actualpar_list>   ::= <actualpar_list> "," <expression> | <expression>

<relop>            ::= ">" | "<" | ">=" | "<=" | "=" | "#"
<addlop>           ::= "+" | "-"
<multop>           ::= "*" | "MOD" | "/"
<variable>         ::= ID | ID "[" <expression> "]" | ID "." "size"
<statement>        ::= <assignment> | <conditional> | <repetition> | <proccall>
<assignment>       ::= <variable> ":=" <expression>
<conditional>      ::= "IF" <expression> "THEN" <statements> { "ELSIF"
<expression> "THEN" <statements> } [ "ELSE" <statements> ] "END"
<conditional>      ::= <conditional_list> <conditional_op> "END" | €
<conditional_list> ::= <conditional_list> "ELSIF" <expression> "THEN" <statements>
| "IF" <expression> "THEN" <statements>

<repetition>       ::= "WHILE" <expression> "DO" <statements> "END" | "REPEAT"
<statement> "UNTIL" <expression>
<io_statement>     ::= "WRITE" "(" <expression> ")" | "WRITELN" | "WRITELN" "("
<expression> ")" | "READ" "(" <expression> ")"

```