DEPARTAMENTO DE MÁTEMATICA E COMPUTAÇÃO

COMPILADORES

TRABALHO

TRANSFORMAÇÃO DA LALG EM LL1

BRUNO SANTOS DE LIMA

LEANDRO UNGARI CAYRES

=============================================================

<programa> ::= program <identificador>; <bloco>.

<bloco> ::= [<parte de declarações de variáveis>][<parte de declaração de sub-rotinas>]<comando composto>

<parte de declarações de variáveis> ::= <declarações de variáveis> {; <declaração de variáveis>}

<declarações de variáveis> ::= <tipo> <lista de identificadores>

<lista de identificadores> ::= <identificador>{, <identificador>}

<parte de declarações de sub-rotinas> ::= {<declaração de procedimento>;}

<declaração de procedimentos> ::= procedure <identificador>[<parâmetros formais>];<bloco> V

<parâmetros formais> ::= (<seção de parâmetros formais> {; <seção de parâmetros formais>})

<seção de parâmetros formais> ::= [var] <lista de identificadores> : <identificador> V

<tipo> := int | boolean

<comando composto> ::= begin <comando> {; <comando>} end

<comando> ::= <atribuições> | <chamada de procedimentos> | begin <comando>{; <comando>} end | <comando condicional1> | <comando repetitivo1>

<atribuição> ::= <variável> := <expressão>

<chamada de procedimento> ::= <identificador> [(<lista de expressões>)]

<comando condicional1> ::= if <expressão> then

<atribuições> | <chamada de procedimentos> | begin <comando>{; <comando>} end | <comando condicional1> | <comando repetitivo1>

[<comando condicional2> ]

/\*

<comando condicional1’> ::= if <expressão> then

<atribuições> | <chamada de procedimentos> | begin <comando>{; <comando>} end | <comando condicional1’> | <comando repetitivo1>

[<comando condicional2> ]

\*/

<comando condicional2> ::= else <atribuições> | <chamada de procedimentos> | begin <comando>{; <comando>} end | <comando condicional1> | <comando repetitivo1>

<comando repetitivo1> ::= while <expressão> do

<atribuições> | <chamada de procedimentos> | begin <comando>{; <comando>} end | <comando condicional1> | <comando repetitivo1>

<expressão> ::= <expressão simples> [<relação><expressão simples>]

<relação> := = | <> | < | <= | >= | >

<expressão simples> := [+ | -] <termo> {(+ | - | or) <termo>}

<termo> ::= <fator> {(\* | div | and) <fator>}

<fator> ::= <variável> | <numero> | ([+ | -] <fator> {(\* | div | and) <fator>} {(+ | - | or) <fator> {(\* | div | and) <fator>}} [<relação>[+ | -] <fator> {(\* | div | and) <fator>} {(+ | - | or) <fator> {(\* | div | and) <fator>}}]) | not <fator>

<variável> ::= <identificador> [<expressão>]

<lista de expressões> ::= <expressão> {, <expressão>}

<numero> ::= <digito> {<digito>}

<digito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<identificador> := <letra> {<letra>|<digito>}

<letra> := \_ | a-z | A-Z