

1 Compilador para a linguagem L2012-1

Considere a L2012-1 definida pela seguinte gramática:

```
program ::= program identifier ";" decl_list compound_stmt
decl_list ::= decl_list " ;" decl
           | decl
decl ::= dcl_var
       | dcl_proc
dcl_var ::= ident_list ":" type
ident_list ::= ident_list " ," identifier
           | identifier
type ::= integer
       | real
       | boolean
       | char
dcl_proc ::= tipo_retornado PROCEDURE identifier espec_parametros corpo
tipo_retornado ::= integer
                | real
                | boolean
                | char
                | vazio
corpo ::= ":" decl_list ";" compound_stmt id_return
       | vazio
id_return ::= identifier
          | vazio
espec_parametros ::= "(" lista_parametros ")"
lista_de_parametros ::= parametro
                   | lista_de_parametros , parametro
parametro ::= modo type : identifier
modo ::= value
       | reference
compound_stmt ::= begin stmt_list end
stmt_list ::= stmt_list ";" stmt
          | stmt
stmt ::= assign_stmt
      | if_stmt
      | repeat_stmt
      | read_stmt
      | write_stmt
      | compound_stmt
      | function_ref_par
```

```

assign_stmt ::= identifier " :="  expr
if_stmt    ::= if cond then stmt
            |  if cond then stmt
              else stmt
cond       ::= expr
repeat_stmt ::= repeat stmt_list until expr
read_stmt  ::= read "("  ident_list  ")"
write_stmt ::= write "("  expr_list  ")"
expr_list  ::= expr
            |  expr_list " ,"  expr
expr       ::= simple_expr
            |  simple_expr RELOP simple_expr
simple_expr ::= term
            |  simple_expr ADDOP term
term       ::= factor_a
            |  term MULOP factor_a
factor_a   ::= "-" factor
            |  factor
factor     ::= identifier
            |  constant
            |  "("  expr  ")"
            |  NOT factor
            |  function_ref_par
function_ref_par ::= variable "("  expr_list  ")"
variable        ::= simple_variable_or_proc
simple_variable_or_proc ::= identifier
constant        ::= integer_constant
                  |  real_constant
                  |  char_constant
                  |  boolean_constant
boolean_constant ::= false | true

```

Considere as seguintes convenções léxicais:

1. Identificadores são definidos pelas seguintes expressões regulares:

```

letter ::= A | B | ... Z
        |  a | b | ... z
digit  ::= 0 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
identifier ::= letter ( letter | digit ) *

```

Na implementação pode-se limitar o tamanho do identificador.

2. Constantes são definidas da seguinte forma:

```
unsigned_integer ::= digit digit ★  
    sign ::= + | - |  $\mathcal{E}$   
    scale_factor ::= "E" sign unsigned_integer  
unsigned_real ::= unsigned_integer (  $\mathcal{E}$  | "." digit★)(  $\mathcal{E}$  | scale_factor)  
integer_constant ::= unsigned_integer  
real_constant ::= unsigned_real  
char_constant ::= "'" caractereASCII "'"
```

3. Os operadores de relação (RELOP's) são:

RELOP ::= = | < | ≤ | > | ≥ | !=

4. Os operadores de adição (ADDOP's) são:

ADDOP ::= + | - | **or**

5. Os operadores (MULOP's) são:

MULOP ::= ★ | / | **div** | **mod** | **and**

6. As funções padrões são: sin, log, cos, ord, chr, abs, sqrt, exp, eof, eoln.

7. A linguagem L2012-1 é um subconjunto bastante reduzido das linguagens imperativas mais comuns. A sua semântica supomos ser óbvia, mas os casos de dúvida prevalece a semântica do Pascal por ser mais simples.

RESULTADOS DESEJADOS

A sua tarefa é construir um **compilador** para a L2012-1 e um interpretador de quádruplas (ou usar o interpretador disponível na página do curso), de forma a ser possível executar programas em L2012-1. O compilador deve gerar código para a linguagem intermediária de quádruplas, cujas instruções disponíveis devem ser definidas conforme a necessidade.

Além da listagem do programa, exibindo compilação e execução de exemplos, a documentação do projeto também deverá ser produzida. Esta documentação deverá, **no mínimo**, incluir o seguinte:

Análise Léxica:

definição dos tokens e estruturas de dados usadas.

Análise Sintática:

Comentário sobre o método escolhido e as modificações efetuadas na gramática dada, se houverem, visando a eliminação de conflitos e ambiguidades. Comente também sobre o que foi feito sobre a recuperação de erros.

Com o objetivo de simplificar o seu trabalho, não é exigido que se faça "recuperação de erros", isto é, erros sintáticos podem ser considerados fatais. Entretanto, a mensagem de erro correspondente deve ser expressa.

Erros decorrentes do mal uso de tipos, inconsistência do número de parâmetros em uma chamada de procedimento e uso de variáveis não declaradas **não** interferem com o funcionamento do analisador sintático. Erros deste tipo não devem descontinuar a compilação.

Tabela de Símbolos:

Organização, método de acesso e atributos dos símbolos.

Esquema de Tradução:

Descrição da estrutura de dados. Definição dos atributos usados e especificação das rotinas semânticas implementadas.

Saída do Interpretador:

Listagem de programas exemplos, da tabela de símbolos, do código gerado (quádruplas) e do resultado da execução do programa.

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
COMPILADORES

2012-1

Prof: Mariza Andrade da Silva Bigonha

Folha de Avaliação da Etapa 1 do Trabalho Prático: Análise Léxica - 3 pontos

Documentação: (total: 1 ponto)

- Descrição sucinta sobre o desenvolvimento do trabalho: (**1 ponto**)

Implementação e Resultados: (Total: 2 pontos)

- Apresentou os resultados dos programas fontes submetidos com os símbolos reconhecidos pelo Analisador Léxico? [S, N, Mais ou Menos] (**1 ponto**)
- Apresentou as listagens dos programas fontes submetidos ao seu Analisador Léxico? [S, N, Mais ou Menos] (**0.5 pontos**)
- Apresentou a listagem do código do analisador léxico? [S, N, Mais ou Menos] (**0.5 pontos**)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
COMPILADORES

2012-1

Prof: Mariza Andrade da Silva Bigonha

Folha de Avaliação da Etapa 2 do Trabalho Prático: Análise Sintática e Tabela de Símbolos - 3 pontos

Documentação: (total: 1 ponto)

Implementação e Resultados: (Total: 2 pontos)

- Fez Análise Sintática? [S, N]
- Apresentou a gramática modificada, resolveu os conflitos, caso necessário? [S, N, Mais ou Menos]
- Apresentou os resultados dos programas testes submetidos ao Analisador Sintático? [S, N, Mais ou Menos]
- Apresentou os resultados da T.S.? [S, N, Mais ou Menos]

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
COMPILADORES

2012-1

Prof: Mariza Andrade da Silva Bigonha

Folha de Avaliação da Etapa 3 do Trabalho Prático: Análise Semântica - 3 pontos

Documentação: (total: 1 pontos)

Implementação e Resultados: (Total: 2 pontos)

- Fez a verificação de tipos? [S, N]
- Apresentou os resultados dos programas testes submetidos ao Analisador Semântico? [S, N, Mais ou Menos]

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
COMPILADORES

2012-1

Prof: Mariza Andrade da Silva Bigonha

Folha de Avaliação da Etapa 4 do Trabalho Prático: Código Intermediário. Resultado da Execução dos exemplos - 14 pontos

Documentação: (total: 3 pontos)

- Descrição sucinta sobre o desenvolvimento do trabalho: **(1 ponto)**
- Descrição das estruturas de dados usadas: **(1 ponto)**
- Descrição de utilização: **(1 ponto)**

Implementação: (Total: 5 pontos)

- Gerou código para declarações e apresentou a listagem do programa fonte? [S, N, Mais ou Menos] **(2 pontos)**
- Gerou código para comandos e apresentou a listagem do programa fonte? [S, N, Mais ou Menos] **(2 pontos)**
- Gerou código para expressões e apresentou a listagem do programa fonte? [S, N, Mais ou Menos] **(1 ponto)**

Resultados: (Total: 6 pontos)

- Apresentou as listagens dos programas testes submetidos ao Compilador? [S, N, Mais ou Menos] **(1 ponto)**
- Apresentou a saída das quadruplas geradas pelo compilador para os programas testes? [S, N, Mais ou Menos] **(2 pontos)**
- Apresentou os resultados da execução dos programas testes? [S, N, Mais ou Menos] **(3 pontos)**