Universidade Tecnológica Federal do Paraná
Curso de Engenharia da Computação
Compiladores
Prof. Marco Antonio Barbosa

Trabalho Prático 3

Objetivo: Desenvolver um interpretador para Expressões Aritméticas Simples para números inteiros.

Data de entrega/apresentação: 05/07/2022

Desenvolvimento: recomenda-se o uso da ferramenta Bison, ou similar a esta para outras linguagens.

Entrada: Expressão aritmética ou programa contendo uma expressão aritmética.

Saída: Se o tipo da expressão for inteiro, gerar como saída o resultado da expressão

Exemplo:
Exemplo de um programa na linguagem **Small L**
1) teste.sml

```
programa teste1 ;
var
 x , y : inteiro ;
 inicio
   x := 5 ;
   y := 10 ;
   x := x +  x * y ;
escreva ( y )
fim
```

ou, via prompt de comando:

:\ 5 + 5 * 10 <enter>
:\ 55

Universidade Tecnológica Federal do Paraná
Curso de Engenharia da Computação
Compiladores
Prof. Marco Antonio Barbosa

A gramática da linguagem **Small L**

<programa> ::= **programa** <identificador> **;** <bloco>

<bloco> ::= **var** <declaracao> **inicio** <comandos> **fim**

<declaracao> ::= <nome_var> **:** <tipo> **;** | <nome_var > **:** <tipo> **;** <declaracao>

<nome_var> ::= <identificador> | <identificador> **,** <nome_var>

<tipo> ::= **inteiro** | **real** | **booleano**

<comandos> ::= <comando> | <comando> **;** <comandos>

<comando> ::= <atribuicao> | <condicional> | <enquanto> | <leitura> | <escrita>

<atribuicao> ::= <identificador> **:=** <expressão>

<condicional> ::= **se** <expressão> **entao** <comandos> |
se <expressão> **entao** <comandos> **senao** <comandos>

<enquanto> ::= **enquanto** <expressao> **faca** <comandos>

<leitura> ::= **leia (** <identificador> **)**

<escrita> ::= **escreva (** <identificador> **)**

<expressao> ::= <simples> | <simples> <op_relacional> <simples>

<op_relacional> ::= **<>** | **=** | **< | > | <= | >=**

<simples> ::= <termo> <operador> <termo> | <termo>

<operador> ::= **+** | **-** | **ou**

<termo> ::= <fator> | <fator> <op> <fator>

<op> ::= ***** | **div** | **e**

<fator> ::= <identificador> | <numero> | **(**<expressao>**)** | **verdadeiro** | **falso | nao** <fator>

<identificador> ::= **id**

<numero> ::= **num**

Comentários: Uma vez que os comentários servem apenas como documentação do código fonte, ao realizar a compilação deste código faz-se necessário eliminar todo o conteúdo entre seus delimitadores: **{ }**

Tipos Numéricos: Inteiros ({naturais positivos e negativos}) e Reais (float)

Identificadores: Letras seguidas de zero ou mais letras ou dígitos

Referências

Flex&Bison–JohnLevine

Compiladores. Princípios, Técnicas e Ferramentas. Alfred V. Aho, Ravi Sethi and Jeffrey D. Ullman.

Compiladores Princípios e Práticas. Kenneth C. Louden.

Materiais complementares disponíveis no Moodle