

**Centro Federal de Educação Tecnológica de Minas Gerais**

**Compiladores**

**Trabalho Prático – Parte III**

**Analisador Semântico**

Membros: Juliano Martins

Mariana Athayde

Thais Diniz Braz

Belo Horizonte / MG  
 2015

**Implementação**

O analisador semântico tem a função de verificar a consistência semântica do programa com a definição da linguagem. Ele deve:

1. Realizar verificação de tipo: Incluir o tipo dos identificadores na Tabela de Símbolos e garantir que os tipos de operandos casam com o tipo esperado por um operador.
2. Verificar classe de identificadores: especifica a classe dos identificadores na Tabela de Símbolos, como variável, procedimento, função e etc.
3. Verificar unicidade de identificadores: Garantir que um objeto seja definido apenas uma vez.

Como a linguagem definida não possui mais de um ambiente, não é necessário verificar classes, item II. Além disso, em nosso trabalho, o item III foi garantido pelo analisador léxico, ao criar os tokens e inserir os identificadores na Tabela de Símbolos.

Utilizando regras semânticas na gramática é possível verificar onde as verificações devem ser inseridas. Por exemplo ao se declarar uma nova variável seu tipo deve ser especificado e adicionado na TS; para se fazer uma soma ambos as partes devem ser do tipo inteiro, etc.

A gramática possui três tipos básicos: inteiro, boolean e String.

**Forma de uso do compilador**

Para utilizar o compilador deve-se chamar o arquivo compilador.jar, passando como parâmetro os arquivos que serão compilados. Lembrando que nessa primeira parte do trabalho, compilar consiste apenas em imprimir os Tokens reconhecidos (ou os erros para os não reconhecidos) e as entradas da Tabela de Símbolos.

Exemplo:

O comando ‘java -jar compilador.jar teste1’ compila o arquivo teste1.

**Testes**

Nesta etapa foram realizados os testes no analisador semântico, para tal foram considerados 8 arquivos, dentre eles 7 contendo erros léxicos. Abaixo estarão as evidências de execução de cada um dos testes antes e depois da correção dos erros, é importante ressaltar que aqui não são corrigidos os outros erros que não dizem respeito ao analisador léxico (erro sintático/semântico).

**Arquivo de Teste 1**

Esse arquivo contém um erro semântico que consiste em uma variável não declarada.

|  |  |
| --- | --- |
| **Programa: teste1** | **Programa: teste1\_Corrigido** |
| declare  a, b, c: int;  result: int  start  read (a);  read (c);  b = 10;  result = (a \* c)/(b + 5 - 345);  write(result);  teste1 = 1;  end | declare  a, b, c: int;  result: int  start  read (a);  read (c);  b = 10;  result = (a \* c)/(b + 5 - 345);  write(result);  end |
| **Resultado do Teste 1:** | **Resultado do Teste 1 Corrigido:** |
| ERROR!!! Variável "teste1" não declarada!  <------ TABELA DE SÍMBOLOS ------->  result.....tipo = inteiro  b.....tipo = inteiro  c.....tipo = inteiro  a.....tipo = inteiro  teste1.....tipo = ??? | <------ TABELA DE SÍMBOLOS ------->  result.....tipo = inteiro  b.....tipo = inteiro  c.....tipo = inteiro  a.....tipo = inteiro |

**Arquivo de Teste 2**

Esse arquivo contém um erro léxico que consiste em um caracter não contido no alfabeto da gramática.

|  |
| --- |
| **Programa: teste2** |
| declare  a: int;  b: int;  nome: string;  start  read (a);  read (nome);  b = a \* a;  b = b + a/2 \* (3 + 5);  write(nome);  write(b);  end |
| **Resultado do Teste 2:** |
| <------ TABELA DE SÍMBOLOS ------->  b.....tipo = inteiro  a.....tipo = inteiro  nome.....tipo = String  BUILD SUCCESSFUL (total time: 1 second) |

**Arquivo de Teste 3**

Esse arquivo contém um erro léxico que consiste em um caracter não contido no alfabeto da gramática.

|  |  |
| --- | --- |
| **Programa: teste3**  declare  cont: int;  media, altura, soma: int;  start  cont = 5;  soma = 0;  do  write(“Altura: ”);  read (altura);  soma = soma # altura;  cont = cont - 1;  while(cont);  write(“Media: ”);  write (soma / qtd);  end | **Programa sem erros léxicos: teste3\_Corrigido**  declare  cont: int;  media, altura, soma: int;  start  cont = 5;  soma = 0;  do  write(“Altura: ”);  read (altura);  soma = soma \* altura;  cont = cont - 1;  while(cont);  write(“Media: ”);  write (soma / qtd);  end |
| **Resultado do Teste 3:**  eating: declare  eating: cont  eating: :  eating: int  eating: ;  eating: media  eating: ,  eating: altura  eating: ,  eating: soma  eating: :  eating: int  eating: ;  eating: start  eating: cont  eating: =  eating: 5  eating: ;  eating: soma  eating: =  eating: 0  eating: ;  eating: do  eating: write  eating: (  eating: “Altura: ”  eating: )  eating: ;  eating: read  eating: (  eating: altura  eating: )  eating: ;  eating: soma  eating: =  eating: soma  ERRO - Caractere não especificado na linha 10 ch: "#".  <------ TABELA DE SÍMBOLOS ------->  qtd  altura  soma  media  cont | **Resultado do Teste 3 Corrigido:**  <declare,41>  <cont,41>  <:,32>  <int,41>  <;,30>  <media,41>  <,,31>  <altura,41>  <,,31>  <soma,41>  <:,32>  <int,41>  <;,30>  <start,41>  <cont,41>  <=,35>  <5,40>  <;,30>  <soma,41>  <=,35>  <0,40>  <;,30>  <do,41>  <write,41>  <(,33>  <“Altura: ”,36>  <),34>  <;,30>  <read,41>  <(,33>  <altura,41>  <),34>  <;,30>  <soma,41>  <=,35>  <soma,41>  <\*,28>  <altura,41>  <;,30>  <cont,41>  <=,35>  <cont,41>  <-,27>  <1,40>  <;,30>  <while,41>  <(,33>  <cont,41>  <),34>  <;,30>  <write,41>  <(,33>  <“Media: ”,36>  <),34>  <;,30>  <write,41>  <(,33>  <soma,41>  </,29>  <qtd,41>  <),34>  <;,30>  <------ TABELA DE SÍMBOLOS ------->  qtd  altura  soma  media  cont  CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos) |

**Arquivo de Teste 4**

Esse arquivo contém apenas um erro semântico que consiste na variável não declarada “a”.

|  |  |
| --- | --- |
| **Programa: teste4** | **Programa sem erros semânticos: teste4\_Corrigido** |
| declare  i, j, k, total, soma: int;  start  i = 4 \* (5-3 \* 50 / 10);  j = i \* 10;  k = i\* j / k;  k = 4 + a ;  write(i);  write(j);  write(k);  end | declare  i, j, k, total, soma1: int;  start  i = 4 \* (5-3 \* 50 / 10);  j = i \* 10;  k = i\* j / k;  k = 4 + 1;  write(i);  write(j);  write(k);  end |
| **Resultado do Teste 4:** | **Resultado do Teste 4 Corrigido:** |
| Variável "a" não declarada! Linha: 7  <------ TABELA DE SÍMBOLOS ------->  total.....tipo = inteiro  a.....tipo = ???  soma.....tipo = inteiro  j.....tipo = inteiro  k.....tipo = inteiro  i.....tipo = inteiro | <------ TABELA DE SÍMBOLOS ------->  total.....tipo = inteiro  soma.....tipo = inteiro  j.....tipo = inteiro  k.....tipo = inteiro  i.....tipo = inteiro |

**Arquivo de Teste 5**

Esse arquivo contém dois erros semânticos, ambos de variáveis não declaradas, sendo que a primeira variável não declarada aparece duas vezes.

|  |  |
| --- | --- |
| **Programa: teste5** | **Programa sem erros semânticos: teste5\_Corrigido** |
| declare  // Este é um comentario if  j, k: int;  a, j: int;  start  read(j);  read(k);  if (k <> 0)  then  result = j/k;  else  result = 0;  end;  write(res);  end | declare  // Este é um comentario if  j, k: int;  a, j: int;  result : int;  start  read(j);  read(k);  if (k <> 0)  then  result = j/k;  else  result = 0;  end;  write(result);  end |
| **Resultado do Teste 5:** | **Resultado do Teste 5 Corrigido:** |
| ERROR!!! Variável "result" não declarada!  ERROR!!! Variável "result" não declarada!  ERROR!!! Veriável "res" não declarada!  <------ TABELA DE SÍMBOLOS ------->  res.....tipo = ???  result.....tipo = ???  a.....tipo = inteiro  j.....tipo = inteiro  k.....tipo = inteiro | <------ TABELA DE SÍMBOLOS ------->  result.....tipo = inteiro  a.....tipo = inteiro  j.....tipo = inteiro  k.....tipo = inteiro |

**Arquivo de Teste 6**

Esse arquivo contém dois erros léxicos: possui identificador maior do que o tamanho máximo permitido e extrapola o limite de tamanho do tipo inteiro.

|  |  |
| --- | --- |
| **Programa: teste6**  declare  // Um programa para calcular a maior idade  a, b, c, maior: int;  nomemenor: string;  start  read(a);  read(b);  read(c);  maior = 12345678901234567890123456789;  if ( a>b and a>c ) then  maior = a;  else  if (b>c) then  maior = b;  else  maior = c;  end  end  write(“Maior idade: ”);  write(maior);  end  end | **Programa sem erros léxicos: teste6\_Corrigido**  declare  // Um programa para calcular a maior idade  a, b, c, maior: int;  variavel: string;  start  read(a);  read(b);  read(c);  maior = 123456;  if ( a>b and a>c ) then  maior = a;  else  if (b>c) then  maior = b;  else  maior = c;  end  end  write(“Maior idade: ”);  write(maior);  end  end |
| **Resultado do Teste 6:** | **Resultado do Teste 6 Corrigido:** |

**Arquivo de Teste 7**

O arquivo 7 possuía apenas erros sintáticos.

|  |
| --- |
| **Programa: teste7** |
| declare  i, j, k : int;  start  i = 4 ;  j = 1 ;  k=2;  write(j);  write(j);/\*  write("Altura é demais");  k = i\* j / k;\*/  k = 4 + i;  write(j);  write(k);  end |
| **Resultado do Teste 7:** |
| <------ TABELA DE SÍMBOLOS ------->  j.....tipo = inteiro  k.....tipo = inteiro  i.....tipo = inteiro |

**Arquivo de Teste 8**

O teste 8 possuía apenas erros sintáticos.

|  |
| --- |
| **Programa: teste8** |
| programa teste8  declare  a, b, c: int;  result: int;  start  read [a];  read (b);  c = -30;  a = 15,7;  wr ite (a+c);  write (" sucesso!);  end |
| **Resultado do Teste 8:** |
| <------ TABELA DE SÍMBOLOS ------->  result.....tipo = inteiro  b.....tipo = inteiro  c.....tipo = inteiro  a.....tipo = inteiro |