

**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 léxico que consiste em um caracter não contido no alfabeto da gramática.

|  |  |
| --- | --- |
| **Programa: teste1**  program teste1  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 | <),34>  <;,30>  <b,41>  <=,35>  <10,40>  <;,30>  <result,41>  <=,35>  <(,33>  <a,41>  <\*,28>  <c,41>  <),34>  </,29>  <(,33>  <b,41>  <+,26>  <5,40>  <-,27>  <345,40>  <),34>  <;,30>  <write,41>  <(,33>  <result,41>  <),34>  <;,30>  <teste1,41>  <=,35>  <1,40>  <;,30>  <end,41>  <------ TABELA DE SÍMBOLOS ------->  result  a  b  c  teste1  program  CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo) |
| **Resultado do Teste 1:**  <program,41>  <teste1,41>  <declare,41>  <a,41>  <,,31>  <b,41>  <,,31>  <c,41>  <:,32>  <int,41>  <;,30>  <result,41>  <:,32>  <int,41>  <start,41>  <read,41>  <(,33>  <a,41>  <),34>  <;,30>  <read,41>  <(,33>  <c,41> |

**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. | **Programa sem erros léxicos: teste2\_Corrigido**  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:**  eating: declare  eating: a  eating: :  eating: int  eating: ;  eating: b  eating: :  eating: int  eating: ;  eating: nome  eating: :  eating: string  eating: ;  eating: start  eating: read  eating: (  eating: a  eating: )  eating: ;  eating: read  eating: (  eating: nome  eating: )  eating: ;  eating: b  eating: =  eating: a  eating: \*  eating: a  eating: ;  eating: b  eating: =  eating: b  eating: +  eating: a  eating: /  eating: 2  eating: \*  eating: (  eating: 3  eating: +  eating: 5  eating: )  eating: ;  eating: write  eating: (  eating: nome  eating: )  eating: ;  eating: write  eating: (  eating: b  eating: )  eating: ;  eating: end  ERRO - Caractere não especificado na linha 12 ch: ".".  NESP Léxico  <------ TABELA DE SÍMBOLOS ------->  a  b  nome  CONSTRUÍDO COM SUCESSO (tempo total: 2 segundos) | **Resultado do Teste 2 Corrigido:**  <declare,41>  <a,41>  <:,32>  <int,41>  <;,30>  <b,41>  <:,32>  <int,41>  <;,30>  <nome,41>  <:,32>  <string,41>  <;,30>  <start,41>  <read,41>  <(,33>  <a,41>  <),34>  <;,30>  <read,41>  <(,33>  <nome,41>  <),34>  <;,30>  <b,41>  <=,35>  <a,41>  <\*,28>  <a,41>  <;,30>  <b,41>  <=,35>  <b,41>  <+,26>  <a,41>  </,29>  <2,40>  <\*,28>  <(,33>  <3,40>  <+,26>  <5,40>  <),34>  <;,30>  <write,41>  <(,33>  <nome,41>  <),34>  <;,30>  <write,41>  <(,33>  <b,41>  <),34>  <;,30>  <end,41>  <------ TABELA DE SÍMBOLOS ------->  a  b  nome  CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo) |

**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 três erros léxicos que consiste em identificador mal formado (inicalizados de um númer) e caracteres nao especificados no alfabeto da gramática.

|  |  |
| --- | --- |
| **Programa: teste4**  declare  i, j, k, @total, 1soma: 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 | **Programa sem erros léxicos: teste4\_Corrigido**  declare  i, j, k, total, soma1: 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 |
| **Resultado do Teste 4:**  eating: declare  eating: i  eating: ,  eating: j  eating: ,  eating: k  eating: ,  ERRO - Caractere não especificado na linha 2 ch: "@".  <------ TABELA DE SÍMBOLOS ------->  a  total  soma  i  j  k  CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo) | **Resultado do Teste 4 Corrigido:**  <declare,41>  <i,41>  <,,31>  <j,41>  <,,31>  <k,41>  <,,31>  <total,41>  <,,31>  <soma1,41>  <:,32>  <int,41>  <;,30>  <start,41>  <i,41>  <=,35>  <4,40>  <\*,28>  <(,33>  <5,40>  <-,27>  <3,40>  <\*,28>  <50,40>  </,29>  <10,40>  <;,30>  <j,41>  <=,35>  <i,41>  <\*,28>  <10,40>  <;,30>  <k,41>  <=,35>  <i,41>  <\*,28>  <j,41>  </,29>  <k,41>  <;,30>  <k,41>  <=,35>  <4,40>  <+,26>  <a,41>  <;,30>  <write,41>  <(,33>  <i,41>  <),34>  <;,30>  <write,41>  <(,33>  <j,41>  <),34>  <;,30>  <write,41>  <(,33>  <k,41>  <),34>  <;,30>  <end,41>  <------ TABELA DE SÍMBOLOS ------->  a  total  i  soma1  j  k  CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo) |

**Arquivo de Teste 5**

Esse arquivo contém um erro onde o comentário é iniciado entretanto, não é encerrado.

|  |  |
| --- | --- |
| **Programa: teste5**  declare  /\*Este e um programa com if  j, k: int;  a, j: real;  start  read(j);  read(k);  if (k != 0)  result = j/k  else  result = 0  end;  write(res);  end | **Programa sem erros léxicos: teste5\_Corrigido**  declare  /\*Este e um programa com\*/ if  j, k: int;  a, j: real;  start  read(j);  read(k);  if (k == 0)  result = j/k  else  result = 0  end;  write(res);  end |
| **Resultado do Teste 5:**  eating: declare  WARNING! Comentário não encerrado!  Fim do arquivo  <------ TABELA DE SÍMBOLOS ------->  CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo) | **Resultado do Teste 5 Corrigido:**  <declare,41>  <if,41>  <j,41>  <,,31>  <k,41>  <:,32>  <int,41>  <;,30>  <a,41>  <,,31>  <j,41>  <:,32>  <real,41>  <;,30>  <start,41>  <read,41>  <(,33>  <j,41>  <),34>  <;,30>  <read,41>  <(,33>  <k,41>  <),34>  <;,30>  <if,41>  <(,33>  <k,41>  <==,20>  <0,40>  <),34>  <result,41>  <=,35>  <j,41>  </,29>  <k,41>  <else,41>  <result,41>  <=,35>  <0,40>  <end,41>  <;,30>  <write,41>  <(,33>  <res,41>  <),34>  <;,30>  <end,41>  <------ TABELA DE SÍMBOLOS ------->  result  a  res  j  k  real |

**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;  umavariavelcomonomemuitogrande: 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:**  <declare,41>  <a,41>  <,,31>  <b,41>  <,,31>  <c,41>  <,,31>  <maior,41>  <:,32>  <int,41>  <;,30>  ERRO! Identificador excedeu o número permitido de caracteres!  <umavariavelcomonomem,41>  <uitogrande,41>  <:,32>  <string,41>  <;,30>  <start,41>  <read,41>  <(,33>  <a,41>  <),34>  <;,30>  <read,41>  <(,33>  <b,41>  <),34>  <;,30>  <read,41>  <(,33>  <c,41>  <),34>  <;,30>  <maior,41>  <=,35>  Erro! Valor de inteiro maior que o maximo permitido  <12345678901234567890123456789,40>  <;,30>  <if,41>  <(,33>  <a,41>  <>,21>  <b,41>  <and,41>  <a,41>  <>,21>  <c,41>  <),34>  <then,41>  <maior,41>  <=,35>  <a,41>  <;,30>  <else,41>  <if,41>  <(,33>  <b,41>  <>,21>  <c,41>  <),34>  <then,41>  <maior,41>  <=,35>  <b,41>  <;,30>  <else,41>  <maior,41>  <=,35>  <c,41>  <;,30>  <end,41>  <end,41>  <write,41>  <(,33>  <“Maior idade: ”,36>  <),34>  <;,30>  <write,41>  <(,33>  <maior,41>  <),34>  <;,30>  <end,41>  <end,41>  <------ TABELA DE SÍMBOLOS ------->  a  b  c  maior  umavariavelcomonomem  uitogrande  CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo) | **Resultado do Teste 6 Corrigido:**  <declare,41>  <a,41>  <,,31>  <b,41>  <,,31>  <c,41>  <,,31>  <maior,41>  <:,32>  <int,41>  <;,30>  <variavel,41>  <:,32>  <string,41>  <;,30>  <start,41>  <read,41>  <(,33>  <a,41>  <),34>  <;,30>  <read,41>  <(,33>  <b,41>  <),34>  <;,30>  <read,41>  <(,33>  <c,41>  <),34>  <;,30>  <maior,41>  <=,35>  <123456,40>  <;,30>  <if,41>  <(,33>  <a,41>  <>,21>  <b,41>  <and,41>  <a,41>  <>,21>  <c,41>  <),34>  <then,41>  <maior,41>  <=,35>  <a,41>  <;,30>  <else,41>  <if,41>  <(,33>  <b,41>  <>,21>  <c,41>  <),34>  <then,41>  <maior,41>  <=,35>  <b,41>  <;,30>  <else,41>  <maior,41>  <=,35>  <c,41>  <;,30>  <end,41>  <end,41>  <write,41>  <(,33>  <“Maior idade: ”,36>  <),34>  <;,30>  <write,41>  <(,33>  <maior,41>  <),34>  <;,30>  <end,41>  <end,41>  <------ TABELA DE SÍMBOLOS ------->  a  b  c  maior  variavel  CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo) |

**Arquivo de Teste 7**

Esse arquivo contém um erro léxico que consiste em um termo literal que é inicializado entretanto, não é finalizado.

|  |  |
| --- | --- |
| **Programa: teste7**  declare  i, j, k, declare: int;  start  i = 4 ;  j = \*;(((\*  tt=2;  write(j);  write(j)\*\*\*\*/  write("Altura é /\* & demais);  k = i\* j / k;  k = 4 + a $;  write(j);  write(k\*);  end;  end | **Programa sem erros léxicos: teste7\_Corrigido**  declare  i, j, k, declare: int;  start  i = 4 ;  j = \*;(((\*  tt=2;  write(j);  write(j)\*\*\*\*/  write("Altura é &demais");  k = i\* j / k;  k = 4 + a;  write(j);  write(k\*);  end;  end |
| **Resultado do Teste 7:**  <declare,41>  <i,41>  <,,31>  <j,41>  <,,31>  <k,41>  <,,31>  <declare,41>  <:,32>  <int,41>  <;,30>  <start,41>  <i,41>  <=,35>  <4,40>  <;,30>  <j,41>  <=,35>  <\*,28>  <;,30>  <(,33>  <(,33>  <(,33>  <\*,28>  <tt,41>  <=,35>  <2,40>  <;,30>  <write,41>  <(,33>  <j,41>  <),34>  <;,30>  <write,41>  <(,33>  <j,41>  <),34>  <\*,28>  <\*,28>  <\*,28>  <\*,28>  </,29>  <write,41>  <(,33>  ERRO! String literal não finalizada!  <------ TABELA DE SÍMBOLOS ------->  tt  i  j  k | **Resultado do Teste7 Corrigido:**  <declare,41>  <i,41>  <,,31>  <j,41>  <,,31>  <k,41>  <,,31>  <declare,41>  <:,32>  <int,41>  <;,30>  <start,41>  <i,41>  <=,35>  <4,40>  <;,30>  <j,41>  <=,35>  <\*,28>  <;,30>  <(,33>  <(,33>  <(,33>  <\*,28>  <tt,41>  <=,35>  <2,40>  <;,30>  <write,41>  <(,33>  <j,41>  <),34>  <;,30>  <write,41>  <(,33>  <j,41>  <),34>  <\*,28>  <\*,28>  <\*,28>  <\*,28>  </,29>  <write,41>  <(,33>  <"Altura é &demais",36>  <),34>  <;,30>  <k,41>  <=,35>  <i,41>  <\*,28>  <j,41>  </,29>  <k,41>  <;,30>  <k,41>  <=,35>  <4,40>  <+,26>  <a,41>  <;,30>  <write,41>  <(,33>  <j,41>  <),34>  <;,30>  <write,41>  <(,33>  <k,41>  <\*,28>  <),34>  <;,30>  <end,41>  <;,30>  <end,41>  <------ TABELA DE SÍMBOLOS ------->  tt  a  i  j  k  CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo) |

**Arquivo de Teste 8**

Esse arquivo contém alguns erros léxicos e foram inseridos caracteres de formatação (\t) para verificar o comportamento do analisador léxico.

|  |  |
| --- | --- |
| **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 | **Programa sem erros léxicos: teste8\_Corrigido**  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:**  eating: teste8  eating: b  eating: c  eating: ;  eating: result  eating: ;  eating: read  ERRO - Caractere não especificado na linha 6 ch: "[".  NESP Léxico  <------ TABELA DE SÍMBOLOS ------->  result  a  teste8  b  c  programa  wr  ite  CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo) | **Resultado do Teste 8 Corrigido:**  <programa,41>  <teste8,41>  <declare,41>  <a,41>  <,,31>  <b,41>  <,,31>  <c,41>  <:,32>  <int,41>  <;,30>  <result,41>  <:,32>  <int,41>  <;,30>  <start,41>  <read,41>  <a,41>  <;,30>  <read,41>  <(,33>  <b,41>  <),34>  <;,30>  <c,41>  <=,35>  <-,27>  <30,40>  <;,30>  <a,41>  <=,35>  <15,40>  <,,31>  <7,40>  <;,30>  <wr,41>  <ite,41>  <(,33>  <a,41>  <+,26>  <c,41>  <),34>  <;,30>  <write,41>  <(,33>  <" sucesso!",36>  <),34>  <;,30>  <end,41>  <------ TABELA DE SÍMBOLOS ------->  result  a  teste8  b  c  programa  wr  ite  CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos) |