

# Compiladores

## Relatório do trabalho

Integrantes: Gedeão Pereira Lima  
Rodrigo Luiz Panta Junior  
Suellen Regina dos Santos

Data: 28/05/2023

### Forma de uso do compilador

- Executando a partir do arquivo .jar:

Navegue até a pasta contendo o arquivo SynteticAnalyzer.jar e LexicalAnalyzer.jar no terminal do seu sistema operacional (durante a realização do trabalho, foi utilizado o linux, com o Java Runtime Environment na versão 11.0.18), execute o seguinte comando:

```
$ java -jar SynteticAnalyzer.jar [caminho para o arquivo de casos de teste]
```

```
$ java -jar LexicalAnalyzer.jar [caminho para o arquivo de casos de teste]
```

- Compilando e executando a partir dos arquivos .java:

Do diretório raiz do projeto, execute:

```
$ javac Compiler/Test/SyntaticTest.java
```

```
$ java Compiler/Test/SyntaticTest [caminho para o arquivo de casos de teste]
```

```
$ javac Compiler/Test/LexicalTest.java
```

```
$ java Compiler/Test/LexicalTest [caminho para o arquivo de casos de teste]
```

Os comandos acima compilam apenas a classe que contém o método main, LexicalTest ou SyntaticTest. Caso as demais classes não sejam compiladas juntamente, inclua no comando javac todas os arquivos .java nas sub-pastas de Compiler:

```
$ javac Compiler/Test/SyntaticTest.java Compiler/Lexical/Lexer.java Compiler/Lexical/Tag.java [e todos os demais]
```

```
$ javac Compiler/Test/LexicalTest.java Compiler/Lexical/Lexer.java Compiler/Lexical/Tag.java [e todos os demais]
```

Foi realizado as seguintes mudanças na Mudanças na gramática:

- `program` ::= **program** identifier [decl-list] **begin** stmt-list **end** “.”
- `if-stmt` ::= **if** condition **then** stmt-list **end-else**
- `end-else` ::= **end**  
| **else** stmt-list **end**
- `expression` ::= simple-expr { relop simple-expr }

Com o código construído foi realizado os testes especificados sem os erros léxicos;

## Execução Test1.txt

```
1  program teste1
2
3      a, b is int;
4      result is int;
5      a,x is float;
6
7  begin
8
9      a = 12;
10     x = 12;
11     read (a);
12     read (b);
13     read (c)
14     result = (a*b + 1) / (c+2);
15     write {Resultado: };
16     write (result);
17
18 end.
```

Erro: read (c)

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test1.txt
Unexpected token: result of type: ID on line: 14
Expected: type: END
```

```
12     read (b);
13     read (c);
14     result = (a*b + 1) / (c+2);
```

Erro: write {Resultado: };

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test1.txt
Unexpected token: Resultado: of type: STRING CONSTANT on line: 15
Expected: type: OPEN PAR
```

```
13     read (c);
14     result = (a*b + 1) / (c+2);
15     write ({Resultado: });
16     write (result);
17
```

Erro: write ({Resultado: };

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test1.txt
Unexpected token: ; of type: SEMI COLON on line: 15
Expected: type: CLOSE PAR
```

```
12     read (b);
13     read (c);
14     result = (a*b + 1) / (c+2);
15     write ({Resultado: });
16     write (result);
17
18 end.
```

Erro: write (result);

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test1.txt
Unexpected token: end of type: END on line: 18
```

```
15     write ({Resultado: });
16     write (result)
17
18 end.
```

### Sintático Corrigido

```
1  program teste1
2
3      a, b is int;
4      result is int;
5      a,x is float;
6
7  begin
8
9      a = 12;
10     x = 12;
11     read (a);
12     read (b);
13     read (c);
14     result = (a*b + 1) / (c+2);
15     write ({Resultado: });
16     write (result)
17
18 end.
```

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test1.txt
Syntatic Analysis completed successfully!
```

### Execução Test2.txt

```
1  program teste2
2      a, b, c is int;
3      d, var is float;
4      teste2 = 1;
5      read (a);
6      b = a * a;
7      c = b + a/2 * (35/b);
8      write c;
9      val := 34.2
10     c = val + 2.2 + a;
11     write (val)
12 end.
```

Erro: begin

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test2.txt
Unexpected token: = of type: ASSIGN on line: 4
Expected: type: IS
```

```
2      a, b, c is int;
3      d, var is float;
4  begin
5      teste2 = 1;
6      read (a);
7      b = a * a;
```

Erro: write c;

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test2.txt
Unexpected token: c of type: ID on line: 9
Expected: type: OPEN PAR
```

```
C:\CEFETMG\CP\ImplementacaoCompilador>
```

```
7      b = a * a;
8      c = b + a/2 * (35/b);
9      write (c);
10     val := 34.2
11     c = val + 2.2 + a;
```

Erro: val = 34.2

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test2.txt
Unexpected token: c of type: ID on line: 11
Expected: type: END
```

```
6      read (a);
7      b = a * a;
8      c = b + a/2 * (35/b);
9      write (c);
10     val = 34.2;
11     c = val + 2.2 + a;
12     write (val)
```

**Sintático Corrigido**

```
1  program teste2
2      a, b, c is int;
3      d, var is float;
4  begin
5      teste2 = 1;
6      read (a);
7      b = a * a;
8      c = b + a/2 * (35/b);
9      write (c);
10     val = 34.2;
11     c = val + 2.2 + a;
12     write (val)
13 end.
```

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test2.txt
Syntatic Analysis completed successfully!
```

## Execução Test3.txt

```
1  program
2    a, aux is int;
3    b is float
4    begin
5    b = 0;
6    read(a);
7    read(b);
8    if (a>b) then //troca variaveis
9    aux = b;
10   b = a;
11   a = aux
12   end;
13   write(a;
14   write(b)
```

Erro: Falta nome do programa

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test3.txt
Unexpected token: , of type: COLON on line: 2
Expected: type: ID
```

```
1  program teste3
2    a, aux is int;
3    b is float
4    begin
```

Erro: ;

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test3.txt
Unexpected token: begin of type: BEGIN on line: 4
Expected: type: SEMI COLON
```

```
1  program teste3
2    a, aux is int;
3    b is float;
4    begin
```

Erro: //

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test3.txt
Unexpected token: / of type: DIV on line: 8
```

```
5    b = 0;
6    read(a);
7    read(b);
8    if (a>b) then /*troca variaveis*/
9    aux = b;
10   b = a;
11   a = aux
```

Erro: write(a;

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test3.txt
Unexpected token: ; of type: SEMI COLON on line: 13
Expected: type: CLOSE PAR
```

```
8      if (a>b) then /*troca variaveis*/
9          aux = b;
10         b = a;
11         a = aux
12     end;
13     write(a);
14     write(b)
```

Erro: end.

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test3.txt
Unexpected END of file
```

```
12     end;
13     write(a);
14     write(b)
15 end.
```

#### Sintático Corrigido

```
1  program teste3
2      a, aux is int;
3      b is float;
4  begin
5      b = 0;
6      read(a);
7      read(b);
8      if (a>b) then /*troca variaveis*/
9          aux = b;
10         b = a;
11         a = aux
12     end;
13     write(a);
14     write(b)
15 end.
```

#### Execução Test4.txt

```

1  program teste4
2      /* Teste4 do meu compilador */
3      pontuacao, pontuacaoMaxina, disponibilidade is inteiro;
4      pontuacaoMinima is char;
5  begin
6      pontuacaoMinima = 50;
7      pontuacaoMaxima = 100;
8      write({Pontuacao do candidato: });
9      read(pontuacao);
10     write({Disponibilidade do candidato: });
11     read(disponibilidade);
12
13     while (pontuacao>0 && (pontuacao<=pontuacaoMaxima) do
14         if ((pontuacao > pontuacaoMinima) && (disponibilidade==1)) then
15             write({Candidato aprovado.})
16         else
17             write({Candidato reprovado.})
18         end
19         write({Pontuacao do candidato: });
20         read(pontuacao);
21         write({Disponibilidade do candidato: });
22         read(disponibilidade);
23     end
24 end

```

Erro: pontuacao, pontuacaoMaxina, disponibilidade is inteiro;

```

C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test4.txt
Unexpected token: inteiro of type: ID on line: 3
Expected: type: SEMI COLON

```

```

1  program teste4
2      /* Teste4 do meu compilador */
3      pontuacao, pontuacaoMaxima, disponibilidade is int;
4      pontuacaoMinima is char;
5  begin

```

Erro: while (pontuacao>0 && (pontuacao<=pontuacaoMaxima) do

```

C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test4.txt
Unexpected token: do of type: DO on line: 13
Expected: type: CLOSE PAR

```

```

10     write({Disponibilidade do candidato: });
11     read(disponibilidade);
12
13     while (pontuacao>0 && (pontuacao<=pontuacaoMaxima)) do
14         if ((pontuação > pontuacaoMinima) && (disponibilidade==1)) then
15             write({Candidato aprovado.})
16         else

```

Erro: end

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test4.txt
Unexpected token: write of type: WRITE on line: 19
Expected: type: END
```

```
16      else
17          write({Candidato reprovado.})
18      end;
19      write({Pontuacao do candidato: });
20      read(pontuacao);
```

Erro: read(disponibilidade);

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test4.txt
Unexpected token: end of type: END on line: 23
```

```
17          write({Candidato reprovado.})
18      end;
19      write({Pontuacao do candidato: });
20      read(pontuacao);
21      write({Disponibilidade do candidato: });
22      read(disponibilidade)
23  end
```

Erro:end.

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test4.txt
Unexpected END of file
```

```
23      end
24  end.
```

### Sintático Corrigido

```
1  program teste4
2      /* Teste4 do meu compilador */
3      pontuacao, pontuacaoMaxima, disponibilidade is int;
4      pontuacaoMinima is char;
5  begin
6      pontuacaoMinima = 50;
7      pontuacaoMaxima = 100;
8      write({Pontuacao do candidato: });
9      read(pontuacao);
10     write({Disponibilidade do candidato: });
11     read(disponibilidade);
12
13     while (pontuacao>0 && (pontuacao<=pontuacaoMaxima)) do
14         if ((pontuacao > pontuacaoMinima) && (disponibilidade==1)) then
15             write({Candidato aprovado.})
16         else
17             write({Candidato reprovado.})
18         end;
19         write({Pontuacao do candidato: });
20         read(pontuacao);
21         write({Disponibilidade do candidato: });
22         read(disponibilidade)
23     end
24 end.
```



```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test4.txt
Syntatic Analysis completed successfully!
```

### Execução Test5.txt

```
1  /* Teste do meu compilador */
2
3  program teste5
4      a, b, c, maior is int;
5      outro is char;
6
7  begin
8      repeat
9          write({A});
10         read(a);
11         write({B});
12         read(b);
13         write({C});
14         read(c);
15
16
17
18         if ( (a>b) && (a>c) ) end
19             maior = a
20
21         else
22             if (b>c) then
23                 maior = b;
24
25             else
26                 maior = c
27             end
28         end;
29         write({Maior valor:});
30         write (maior);
31         write ({Outro? (S/N)});
32         read(outro);
33     until (outro == 'N' || outro == 'n')
34 end
```

Erro: if ( (a>b) && (a>c) ) end

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test5.txt
Unexpected token: end of type: END on line: 18
Expected: type: THEN
```

```

15
16
17
18 ▼      if ( (a>b) && (a>c) ) then
19         maior = a
20
21 ▼      else
22 ▼          if (b>c) then
23             maior = b;

```

Erro: maior = c;

```

C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test5.txt
Unexpected token: else of type: ELSE on line: 25

```

```

24
25         else
26             maior = c
27         end
28     end.

```

Erro: read(outro);

```

C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test5.txt
Unexpected token: until of type: UNTIL on line: 33

```

```

31         write ({Outro? (S/N)});
32         read(outro)
33     until (outro == 'N' || outro == 'n')
34 end
35

```

Erro: end

```

C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test5.txt
Unexpected END of file

```

```

33     until (outro == 'N' || outro == 'n')
34 end.
35

```

**Sintático Corrigido**

```

1  /* Teste do meu compilador */
2
3  ▼ program teste5
4      a, b, c, maior is int;
5      outro is char;
6
7  ▼ begin
8  ▼      repeat
9          write({A});
10         read(a);
11         write({B});
12         read(b);
13         write({C});
14         read(c);
15
16
17
18  ▼         if ( (a>b) && (a>c) ) then
19             maior = a
20
21  ▼         else
22  ▼             if (b>c) then
23                 maior = b
24
25  ▼             else
26                 maior = c
27             end
28         end;
29         write({Maior valor:});
30         write (maior);
31         write ({Outro? (S/N)});
32         read(outro)
33     until (outro == 'N' || outro == 'n')
34 end.

```

```

C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test5.txt
Syntatic Analysis completed successfully!

```

### Execução Test6.txt

```

1  ▼ program teste6
2      a, b is float;
3  ▼ begin
4
5      read (a);
6      read(b);
7  ▼      if (a>=b) then
8
9          write(c);
10         end

```

Erro: write(c);

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test6.txt
Unexpected token: end of type: END on line: 10
```

```
7      if (a>=b) then
8
9          write(c)
10     end.
```

Erro: end.

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test6.txt
Unexpected token: . of type: DOT on line: 10
Expected: type: END
```

```
7      if (a>=b) then
8
9          write(c)
10     end
```

Erro: end of file

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test6.txt
Unexpected END of file
```

```
8
9      write(c)
10     end
11 end.
```

**Sintático Corrigido**

```
1 program teste6
2     a, b is float;
3 begin
4
5     read (a);
6     read(b);
7     if (a>=b) then
8
9         write(c)
10    end
11 end.
```

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test6.txt
Syntatic Analysis completed successfully!
```

### Execução Test7.txt

```
1 program teste7 begin
2     a, b is float;
3     c is 'p';
4     read (a);
5     read(b);
6     if (a==b)
7         write(c);
8     end.
```

Erro: program teste7 begin

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test7.txt
Unexpected token: , of type: COLON on line: 2
Expected: type: ASSIGN
```

```
1 program teste7
2     a, b is float;
3     c is 'p';
4 begin
5
6     read (a);
```

Erro: c is 'p';

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test7.txt
Unexpected token: p of type: CHAR CONSTANT on line: 3
Expected: type: SEMI COLON
```

```
1 program teste7
2     a, b is float;
3     c is char;
4 begin
```

Erro: if (a==b)

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test7.txt
Unexpected token: write of type: WRITE on line: 9
Expected: type: THEN
```

```
6     read (a);
7     read(b);
8 ▼    if (a==b) then
9         write(c);
10     end.
```

Erro: write(c);

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test7.txt
Unexpected token: end of type: END on line: 10
```

```
8      if (a==b) then
9          write(c)
```

Erro: end.

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test7.txt
Unexpected token: . of type: DOT on line: 10
Expected: type: END
```

```
8      if (a==b) then
9          write(c)
10     end
```

Erro: End of file

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test7.txt
Unexpected END of file
```

```
9          write(c)
10     end
11 end.
```

**Sintático Corrigido**

```
1  program teste7
2      a, b is float;
3      c is char;
4  begin
5
6      read (a);
7      read(b);
8      if (a==b) then
9          write(c)
10     end
11 end.
```

```
C:\CEFETMG\CP\ImplementacaoCompilador>java -jar SynteticAnalyzer.jar test7.txt
Syntatic Analysis completed successfully!
```