

TP Compiladores - Parte III
Analisador Sintático
11/12/2023

Guilherme de Assis Lima

Código Fonte e arquivo .jar

O código fonte e arquivo .jar estão disponíveis em:

<https://github.com/guicompeng/compilador/tree/semantico>

Forma de uso do compilador

Para executar o compilador, basta rodar o comando:

```
java -jar AnalisadorSemantico.jar [file-path]
```

Exemplo:

```
java -jar AnalisadorSemantico.jar testes/1-corrigido-semantico.txt
```

Abordagem

A abordagem utilizada foi baseada no livro texto da disciplina, com algumas alterações no código para adaptar a especificação do compilador.

Tabela de símbolos

Na etapa 1, foi implementado a tabela de símbolos juntamente com a classe do analisador léxico. Após a alteração realizada na etapa 2, foi criada a classe “SymbolTable.java” que é do tipo *tabela única*, sendo instanciada pela classe do analisador sintático “Parser.java”.

Na etapa 3 (atual) a tabela de símbolos foi efetivamente utilizada. Toda vez que uma variável é declarada, ela é inserida na tabela de símbolos, que contém as informações: lexema, nível e tipo. A inserção das variáveis na tabela de símbolos são feitas na produção:

```
ident-list ::= identifier {“,” identifier}
```

Para facilitar a implementação, foi criado um atributo chamado *curType* (tipo atual), que “salva” a informação do tipo das variáveis que estão sendo declaradas. Isso foi feito para não precisar trafegar essa informação entre os métodos que implementam as produções.

Além disso, existem dois métodos na tabela de símbolo para ajudar a controlar o nível da variáveis:

- *blockInput*: método que pode ser chamado na entrada de um bloco
- *blockOutput*: método chamado na saída do bloco. Esse método, além de atualizar o contador de nível, remove todas as linhas do nível antigo.

Também existe o método para buscar um identificador na tabela de símbolos: *findRow* que retorna a variável de maior nível se encontrá-la ou *null* se não encontrar.

Abordagem do analisador semântico

O código que realiza a análise semântica foi feito dentro do próprio parser. Para saber se a variável foi declarada (antes de utilizá-la), em todas as produções possíveis de utilizar identificadores, foi verificado se existe a variável na tabela de símbolos. As seguintes produções podem utilizar identificadores:

- *assign-stmt ::= identifier "=" simple-expr*
- *read-stmt ::= read "(" identifier ")"*
- *factor ::= identifier | constant | "(" expression ")"*

Todas as outras regras semânticas também foram implementadas, bastando alterar alguns métodos existentes do parser para realizar as verificações. Essas regras podem ser testadas nos 7 testes que foram realizados.

Algumas das regras implementadas são:

- Toda variável deve ser declarada antes do seu uso.
- O operador "+", quando aplicado a dado do tipo string, representa concatenação.
- Os demais operadores aritméticos são aplicáveis somente aos tipos numéricos.
- O resultado da divisão entre dois números inteiros é um número real.
- Somente tipos iguais são compatíveis nesta linguagem.
- As operações de comparação resultam em valor lógico (verdadeiro ou falso)

Testes

Para realizar os testes, foram utilizados os programas corrigidos do analisador sintático, e depois gerado um novo programa que contém as correções para o analisador semântico.

Fonte 1

Comando: `java -jar AnalisadorSemantico.jar testes/1-corrigido-sintatico.txt`

Corrigido sintático

```
1. class Teste1
2.   int a,b,c;
3.   float result;
4.
5.   {
6.     write("Digite o valor de a:");
7.     read (a);
8.     write("Digite o valor de c:");
9.     read (c);
10.    b = 10;
11.    result = (a * c)/(b - 345);
12.    write("O resultado e: ");
13.    write(result);
14.  }
15.
```

Saída do compilador

Semântico: ok

Não há nenhum erro semântico neste código.

Fonte 2

Comando: `java -jar AnalisadorSemantico.jar testes/2-corrigido-sintatico.txt`

Corrigido sintático

```
1. class Teste2
2. /* Teste de comentário
3. com mais de uma linha */
4. int a, valor, b_1, b_2;
5.
6. {
7.     write("Entre com o valor de a: ");
8.     read (a);
9.     b_1 = a * a;
10.    write("O valor de b1 e: ");
11.    write (b_1);
12.    b_2 = b + a/2 * (a + 5);
13.    write("O valor de b2 e: ");
14.    write (b2);
15. }
16.
```

Parcialmente Corrigido

```
1. class Teste2
2. /* Teste de comentário
3. com mais de uma linha */
4. int a, valor, b_1, b_2, b;
5.
6. {
7.     write("Entre com o valor de a: ");
8.     read (a);
9.     b_1 = a * a;
10.    write("O valor de b1 e: ");
11.    write (b_1);
12.    b_2 = b + a/2 * (a + 5);
13.    write("O valor de b2 e: ");
14.    write (b2);
15. }
```

Parcialmente Corrigido

```
1. class Teste2
2. /* Teste de comentário
3. com mais de uma linha */
4. int a, valor, b_1, b_2, b;
5.
6. {
7.     write("Entre com o valor de a: ");
8.     read (a);
9.     b_1 = a * a;
10.    write("O valor de b1 e: ");
11.    write (b_1);
12.    b_2 = (b + a)/(2 * (a + 5));
13.    write("O valor de b2 e: ");
14.    write (b2);
15. }
```

Resultado:

Erro semântico na linha 12
Causado por: "b" não foi declarado

Nessa correção, "b" foi declarado, porém um novo erro apareceu na mesma linha 12. "b" é somado com "a/2", porém "a/2" resulta em FLOAT. Como b é do tipo INT e a divisão de "a/2" é FLOAT, o seguinte erro aparece:

Erro semântico na linha 12
Causado por: Há tipos incompatíveis nessa expressão (INT com FLOAT)

Foi aplicado parênteses nessa correção. Agora (b + a) é somado primeiro, sendo que "b" e "a" são INT. Depois, é feito (2 * (a + 5)), todos também são INT. Por causa do parênteses, é feito a divisão de (INT / INT), ou seja, são tipos compatíveis, porém a divisão disso resulta em um FLOAT.

Por causa disso, b_2 precisava ser do tipo FLOAT, mas como b_2 é INT, o seguinte erro aparece:

Erro semântico na linha 12
Causado por: Tipos incompatíveis. A variável b_2 foi declarada com o tipo INT mas o tipo a direita do '=' é FLOAT

```
1. class Teste2
2. /* Teste de comentário
3. com mais de uma linha */
4. int a, valor, b_1, b;
5. float b_2;
6. {
7.     write("Entre com o valor de a: ");
8.     read (a);
9.     b_1 = a * a;
10.    write("O valor de b1 e: ");
11.    write (b_1);
12.    b_2 = (b + a)/(2 * (a + 5));
13.    write("O valor de b2 e: ");
14.    write (b2);
15. }
```

Corrigido

```
1. class Teste2
2. /* Teste de comentário
3. com mais de uma linha */
4. int a, valor, b_1, b;
5. float b_2;
6. {
7.     write("Entre com o valor de a: ");
8.     read (a);
9.     b_1 = a * a;
10.    write("O valor de b1 e: ");
11.    write (b_1);
12.    b_2 = (b + a)/(2 * (a + 5));
13.    write("O valor de b2 e: ");
14.    write (b_2);
15. }
```

b_2 foi declarado como float, o que corrigiu o erro da linha 12. Porém, agora um novo erro apareceu:

Erro semântico na linha 14
Causado por: "b2" não foi declarado

Resultado:

Semântico: ok

Fonte 3

Comando: `java -jar AnalisadorSemantico.jar testes/3-corrigido-sintatico.txt`

Corrigido sintático

```
1. class Teste3
2. /** Verificando fluxo de controle
3. Programa com if e while aninhados **/
4. int i;
5. int media, soma;
6. {
7.     soma = 0;
8.
9.     write("Quantos dados deseja informar?" );
10.    read (qtd);
11.    if (qtd>=2){
12.        i=0;
13.        do{
14.            write("Altura: ");
15.            read (altura);
16.            soma = soma+altura;
17.            i = i + 1;
18.        }while( i < qtd);
19.        media = soma / qtd;
20.        write("Media: ");
21.        write (media);
22.    }
23.    else{
24.        write("Quantidade inválida.");
25.    };
26. }
27.
```

Parcialmente Corrigido

```
1. class Teste3
2. /** Verificando fluxo de controle
3. Programa com if e while aninhados **/
4. int i;
5. int media, soma, qtd;
6. {
7.     soma = 0;
8.
9.     write("Quantos dados deseja informar?" );
10.    read (qtd);
11.    if (qtd>=2){
12.        i=0;
13.        do{
14.            write("Altura: ");
15.            read (altura);
16.            soma = soma+altura;
17.            i = i + 1;
18.        }while( i < qtd);
19.        media = soma / qtd;
20.        write("Media: ");
21.        write (media);
22.    }
23.    else{
24.        write("Quantidade inválida.");
25.    };
26. }
```

Resultado:

Erro semântico na linha 10
Causado por: "qtd" não foi declarado

Resultado:

Erro semântico na linha 16
Causado por: "altura" não foi declarado

Parcialmente Corrigido

```
1. class Teste3
2. /** Verificando fluxo de controle
3. Programa com if e while aninhados **/
4. int i;
5. int media, soma, qtd, altura;
6. {
7.     soma = 0;
8.
9.     write("Quantos dados deseja informar?" );
10.    read (qtd);
11.    if (qtd>=2){
12.        i=0;
13.        do{
14.            write("Altura: ");
15.            read (altura);
16.            soma = soma+altura;
17.            i = i + 1;
18.        }while( i < qtd);
19.        media = soma / qtd;
20.        write("Media: ");
21.        write (media);
22.    }
23.    else{
24.        write("Quantidade inválida.");
25.    };
26. }
```

Parcialmente Corrigido

```
1. class Teste3
2. /** Verificando fluxo de controle
3. Programa com if e while aninhados **/
4. int i;
5. int soma, qtd, altura;
6. float media;
7. {
8.     soma = 0;
9.
10.    write("Quantos dados deseja informar?" );
11.    read (qtd);
12.    if (qtd>=2){
13.        i=0;
14.        do{
15.            write("Altura: ");
16.            read (altura);
17.            soma = soma+altura;
18.            i = i + 1;
19.        }while( i < qtd);
20.        media = soma / qtd;
21.        write("Media: ");
22.        write (media);
23.    }
24.    else{
25.        write("Quantidade inválida.");
26.    };
27. }
```

Resultado:

A divisão de int com int resulta em float. Como a variável media foi declarada como int, há um erro semântico:

Erro semântico na linha 19

Causado por: Tipos incompatíveis. A variável media foi declarada com o tipo INT mas o tipo a direita do '=' é FLOAT

Resultado:

Semântico: ok

Fonte 4

Comando: `java -jar AnalisadorSemantico.jar testes/4-corrigido-sintatico.txt`

Corrigido sintático

```
1. class Teste4
2. // Outro programa de teste
3. int idade, j, k, total;
4. string nome, texto;
5. {
6.     write("Digite o seu nome: ");
7.     read(nome);
8.     write("Digite o seu sobrenome");
9.     read(sobrenome);
10.    write("Digite a sua idade: ");
11.    read (idade);
12.    k = i * (5-i * 50 / 10);
13.    j = i * 10;
14.    k = i * j / k;
15.    texto = nome + " " + sobrenome + ", os
    números gerados sao: ";
16.    write (text);
17.    write(j);
18.    write(k);
19. }
20.
```

Parcialmente corrigido

```
1. class Teste4
2. // Outro programa de teste
3. int idade, j, k, total;
4. string nome, texto, sobrenome;
5. {
6.     write("Digite o seu nome: ");
7.     read(nome);
8.     write("Digite o seu sobrenome");
9.     read(sobrenome);
10.    write("Digite a sua idade: ");
11.    read (idade);
12.    k = i * (5-i * 50 / 10);
13.    j = i * 10;
14.    k = i * j / k;
15.    texto = nome + " " + sobrenome + ", os
    números gerados sao: ";
16.    write (text);
17.    write(j);
18.    write(k);
19. }
```

Resultado:

```
Erro semântico na linha 9
Causado por: "sobrenome" não foi declarado
```

Resultado:

```
Erro semântico na linha 12
Causado por: "i" não foi declarado
```


Parcialmente corrigido

```
1. class Teste4
2. // Outro programa de teste
3. int idade, j, k, total, i;
4. string nome, texto, sobrenome;
5. {
6.     write("Digite o seu nome: ");
7.     read(nome);
8.     write("Digite o seu sobrenome");
9.     read(sobrenome);
10.    write("Digite a sua idade: ");
11.    read (idade);
12.    k = i * (5-i * 50 / 10);
13.    j = i * 10;
14.    k = i * j / k;
15.    texto = nome + " " + sobrenome + ", os
    números gerados sao: ";
16.    write (text);
17.    write(j);
18.    write(k);
19. }
```

Parcialmente corrigido

```
1. class Teste4
2. // Outro programa de teste
3. int idade, j, k, total, i;
4. string nome, texto, sobrenome;
5. {
6.     write("Digite o seu nome: ");
7.     read(nome);
8.     write("Digite o seu sobrenome");
9.     read(sobrenome);
10.    write("Digite a sua idade: ");
11.    read (idade);
12.    k = i * (5.0-i * 50.0 / 10.0);
13.    j = i * 10;
14.    k = i * j / k;
15.    texto = nome + " " + sobrenome + ", os
    números gerados sao: ";
16.    write (text);
17.    write(j);
18.    write(k);
19. }
```

Resultado:

Na linha 12, quando é feito “i*50 / 10” isso resulta em um FLOAT.

Depois é feito “5 - FLOAT”, que resulta em erro:

Erro semântico na linha 12
Causado por: Há tipos incompatíveis nessa expressão (INT com FLOAT)

Resultado:

O erro anterior foi corrigido substituindo “5” por “5.0”, e 50 por “50.0” e 10 por “10.0”. Porém agora um novo erro foi mostrado, pois a variável “i” é do tipo INT, mas é feito “i * (5.0-i * 50.0 / 10.0)”, sendo que “(5.0-i * 50.0 / 10.0)” é FLOAT ou seja, é feito INT * FLOAT.

Erro semântico na linha 12
Causado por: Há tipos incompatíveis nessa expressão (INT com FLOAT)

Parcialmente corrigido

```
1. class Teste4
2. // Outro programa de teste
3. int idade, j, k, total;
4. float i;
5. string nome, texto, sobrenome;
6. {
7.     write("Digite o seu nome: ");
8.     read(nome);
9.     write("Digite o seu sobrenome");
10.    read(sobrenome);
11.    write("Digite a sua idade: ");
12.    read (idade);
13.    k = i * (5.0-i * 50.0 / 10.0);
14.    j = i * 10;
15.    k = i * j / k;
16.    texto = nome + " " + sobrenome + ", os
    números gerados sao: ";
17.    write (text);
18.    write(j);
19.    write(k);
20. }
21.
```

Parcialmente corrigido

```
1. class Teste4
2. // Outro programa de teste
3. int idade, j, total;
4. float i, k;
5. string nome, texto, sobrenome;
6. {
7.     write("Digite o seu nome: ");
8.     read(nome);
9.     write("Digite o seu sobrenome");
10.    read(sobrenome);
11.    write("Digite a sua idade: ");
12.    read (idade);
13.    k = i * (5.0-i * 50.0 / 10.0);
14.    j = i * 10;
15.    k = i * j / k;
16.    texto = nome + " " + sobrenome + ", os
    números gerados sao: ";
17.    write (text);
18.    write(j);
19.    write(k);
20. }
21.
```

Resultado:

O “i” agora é do tipo FLOAT.

Apesar disso, um novo erro ocorreu na linha 13, pois a variável “k” também precisa ser FLOAT.

Erro semântico na linha 13

Causado por: Tipos incompatíveis. A variável k foi declarada com o tipo INT mas o tipo a direita do '=' é FLOAT

Resultado: a variável “k” agora é do tipo float, mas um novo erro é gerado, pois é feito “i * 10” .

Como “i” é do tipo FLOAT e “10” INT, é gerado o erro:

Erro semântico na linha 14

Causado por: Tipos incompatíveis. A variável j foi declarada com o tipo INT mas o tipo a direita do '=' é FLOAT

Parcialmente corrigido

```
1. class Teste4
2. // Outro programa de teste
3. int idade, j, total;
4. float i, k;
5. string nome, texto, sobrenome;
6. {
7.     write("Digite o seu nome: ");
8.     read(nome);
9.     write("Digite o seu sobrenome");
10.    read(sobrenome);
11.    write("Digite a sua idade: ");
12.    read (idade);
13.    k = i * (5.0-i * 50.0 / 10.0);
14.    j = i * 10.0;
15.    k = i * j / k;
16.    texto = nome + " " + sobrenome + ", os
    números gerados sao: ";
17.    write (text);
18.    write(j);
19.    write(k);
20. }
```

Parcialmente corrigido

```
1. class Teste4
2. // Outro programa de teste
3. int idade, total;
4. float i, k, j;
5. string nome, texto, sobrenome;
6. {
7.     write("Digite o seu nome: ");
8.     read(nome);
9.     write("Digite o seu sobrenome");
10.    read(sobrenome);
11.    write("Digite a sua idade: ");
12.    read (idade);
13.    k = i * (5.0-i * 50.0 / 10.0);
14.    j = i * 10.0;
15.    k = i * j / k;
16.    texto = nome + " " + sobrenome + ", os
    números gerados sao: ";
17.    write (text);
18.    write(j);
19.    write(k);
20. }
```

Resultado:

Aconteceu um erro, pois a variável "j" também precisa ser do tipo FLOAT:

Erro semântico na linha 14
Causado por: Tipos incompatíveis. A variável j foi declarada com o tipo INT mas o tipo a direita do '=' é FLOAT

Resultado:

Erro semântico na linha 17
Causado por: "text" não foi declarado

Parcialmente corrigido

```
1. class Teste4
2. // Outro programa de teste
3. int idade, total;
4. float i, k, j;
5. string nome, texto, sobrenome;
6. {
7.     write("Digite o seu nome: ");
8.     read(nome);
9.     write("Digite o seu sobrenome");
10.    read(sobrenome);
11.    write("Digite a sua idade: ");
12.    read (idade);
13.    k = i * (5.0-i * 50.0 / 10.0);
14.    j = i * 10.0;
15.    k = i * j / k;
16.    texto = nome + " " + sobrenome + ", os
    números gerados sao: ";
17.    write (texto);
18.    write(j);
19.    write(k);
20. }
```

Resultado:

Semântico: ok

Fonte 5

Comando: `java -jar AnalisadorSemantico.jar testes/5-corrigido-sintatico.txt`

Corrigido sintático

```
1. class MinhaClasse
2.     float a, b, c;
3.     {
4.
5.         write("Digite um número");
6.         read(a);
7.         write("Digite outro número: ");
8.         read(b);
9.         write("Digite mais um número: ");
10.        read(c);
11.
12.        maior = 0;
13.        if ( (a>b) && (a>c) ) {
14.            maior = a;
15.        } else {
16.            if (b>c) {
17.                maior = b;
18.            }
19.            else {
20.                maior = c;
21.            };
22.        };
23.        write("O maior número é: ");
24.        write(maior);
25.    }
26.
```

Parcialmente Corrigido

```
1. class MinhaClasse
2.     float a, b, c, maior;
3.     {
4.
5.         write("Digite um número");
6.         read(a);
7.         write("Digite outro número: ");
8.         read(b);
9.         write("Digite mais um número: ");
10.        read(c);
11.
12.        maior = 0;
13.        if ( (a>b) && (a>c) ) {
14.            maior = a;
15.        } else {
16.            if (b>c) {
17.                maior = b;
18.            }
19.            else {
20.                maior = c;
21.            };
22.        };
23.        write("O maior número é: ");
24.        write(maior);
25.    }
26.
```

Resultado:

Erro semântico na linha 12
Causado por: "maior" não foi declarado

Resultado:

Erro semântico na linha 12
Causado por: Tipos incompatíveis. A variável maior foi declarada com o tipo FLOAT mas o tipo a direita do '=' é INT

Parcialmente Corrigido

```
1. class MinhaClasse
2.     float a, b, c, maior;
3. {
4.
5.     write("Digite um número");
6.     read(a);
7.     write("Digite outro número: ");
8.     read(b);
9.     write("Digite mais um número: ");
10.    read(c);
11.
12.    maior = 0.0;
13.    if ( (a>b) && (a>c) ) {
14.        maior = a;
15.    } else {
16.        if (b>c) {
17.            maior = b;
18.        }
19.        else {
20.            maior = c;
21.        };
22.    };
23.    write("O maior número é: ");
24.    write(maior);
25. }
26.
```

Resultado:

Semântico: ok

Fonte 6

Comando: `java -jar AnalisadorSemantico.jar testes/6-corrigido-sintatico.txt`

Corrigido sintático

```
1. classe Teste6
2. int a, b, resultado;
3. {
4.     write("Digite o valor de a:");
5.     read (a);
6.     write("Digite o valor de b:");
7.     read (b);
8.     resultado = a + b;
9.     write("A soma e: ");
10.    write(resultado);
11. }
```

Resultado:

Semântico: ok

Fonte 7

Comando: `java -jar AnalisadorSemantico.jar testes/7-corrigido-sintatico.txt`

Corrigido sintático

```
1. class Teste7
2. int peso, altura, resultado;
3. {
4.     write("Digite o peso:");
5.     read (peso);
6.     write("Digite a altura:");
7.     read (altura);
8.     resultado = peso/(altura*altura);
9.     write("O IMC e: ");
10.    write(resultado);
11. }
12.
```

Corrigido

```
1. class Teste6
2. int peso, altura;
3. float resultado;
4. {
5.     write("Digite o peso:");
6.     read (peso);
7.     write("Digite a altura:");
8.     read (altura);
9.     resultado = peso/(altura*altura);
10.    write("O IMC e: ");
11.    write(resultado);
12. }
```

Resultado:

```
Erro semântico na linha 8
Causado por: Tipos incompatíveis. A variável resultado
foi declarada com o tipo INT mas o tipo a direita do
'=' é FLOAT
```

Resultado:

```
Semântico: ok
```