

Análise Semântica

Etapa Final

PROF. DR. ANDRÉ LUIS MENESES SILVA

CONTATO: ANDRELUMESI@ACADEMICO.UFS.BR

Agenda

- ▶ Tutorial
- ▶ Nova Gramática
- ▶ Análise Semântica
- ▶ AbstractVisitor
- ▶ Visitor
- ▶ SemanticVisitor

Tutorial

- ▶ Nesse novo tutorial, o exemplo anterior foi alterado com as seguintes mudanças:
 - ▶ Adição do tipo de dado boolean
 - ▶ Suporte a definição de funções
 - ▶ Introdução dos comandos while, call e assign

Gramática Antiga

- ▶ Só dava suporte a expressões.

Gramática Antiga

- ▶ Só dava suporte a expressões.
 - ▶ $\text{exp} \rightarrow \text{exp} + \text{exp} \mid \text{exp} * \text{exp} \mid \text{exp} \wedge \text{exp} \mid \text{call} \mid \text{assign} \mid \text{num} \mid \text{id}$
 - ▶ $\text{call} \rightarrow \text{id} (\text{params}) \mid \text{id} ()$
 - ▶ $\text{params} \rightarrow \text{exp}, \text{params} \mid \text{exp}$
 - ▶ $\text{assign} \rightarrow \text{id} = \text{exp}$

Gramática Antiga

- ▶ Só dava suporte a expressões.
 - ▶ $\text{exp} \rightarrow \text{exp} + \text{exp} \mid \text{exp} * \text{exp} \mid \text{exp} \wedge \text{exp} \mid \text{call} \mid \text{assign} \mid \text{num} \mid \text{id}$
 - ▶ $\text{call} \rightarrow \text{id}(\text{params}) \mid \text{id}()$
 - ▶ $\text{params} \rightarrow \text{exp}, \text{params} \mid \text{exp}$
 - ▶ $\text{assign} \rightarrow \text{id} = \text{exp}$

Exemplos:

- $3 + 4 * 5$
- $\text{soma}(1+2, b, c)$
- $a = 3+5$

Nova Gramática

- ▶ $\text{exp} \rightarrow \text{exp} + \text{exp} \mid \text{exp} * \text{exp}$
 $\mid \text{exp} \wedge \text{exp} \mid \text{call}$
 $\mid \text{assign} \mid \text{num} \mid \text{id}$
- ▶ $\text{call} \rightarrow \text{id} (\text{params}) \mid \text{id} ()$
- ▶ $\text{params} \rightarrow \text{exp}, \text{params} \mid \text{exp}$
- ▶ $\text{assign} \rightarrow \text{id} = \text{exp}$

Novas Regras

- $\text{program} \rightarrow \text{funcdecl} \mid \text{funcdecl program}$
- $\text{funcdecl} \rightarrow \text{signature body}$
- $\text{signature} \rightarrow \text{id id} (\text{sigParams})$
- $\text{sigparams} \rightarrow \text{ID ID}$
 $\mid \text{ID ID COMMA sigparams}$
- $\text{body} \rightarrow \{ \text{stms} \}$
- $\text{stms} \rightarrow \text{stm} \mid \text{stm stms}$
- $\text{stm} \rightarrow \text{exp} ; \mid \text{while} (\text{exp}) \text{body}$
- $\text{return exp} ;$
- $\text{call} \rightarrow \text{id} (\text{params})$

Exemplo de código

```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```


Exemplo de código

```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```

Importante notar:

- Tipagem dinâmica de variáveis (feita em tempo de execução)
- Tipagem estática na definição de funções (parâmetros)

Exemplo de código

```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```

Importante notar:

- Tipagem dinâmica de variáveis (feita em tempo de execução)
- Tipagem estática na definição de funções (parâmetros)

Análise Semântica deve:

- Deve fazer a checagem de tipos
- O que será avaliado?
 - Chamadas de função
 - Expressão do comando while
 - Retorno da função

Exemplo de código

```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```

Importante notar:

- Tipagem dinâmica de variáveis (feita em tempo de execução)
- Tipagem dinâmica de variáveis (feita em tempo de execução)



Como fazer?

- Expressão do comando while
- Expressão do comando while
- Retorno da função

Exemplo de código

```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```

Como fazer?

- Tabela de Símbolos
 - Uma lista dinâmica
- Visitor Semântico

Exemplo de código

```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```

Como fazer?

- Tabela de Símbolos
- Table = []

Exemplo de código

```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```


Como fazer?

- beginScope('main')
- Table = [{'scope': 'main'}]

Exemplo de

Como fazer?

- addFunction
- Table = [{ 'scope': 'main', 'some': { 'bindable': 'fun',
'params': ['a', 'int', 'b', 'int'], 'type': 'int' } }




```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```


Exemplo de

Como fazer?

- `beginScope('some')`
- `Table = [{ 'scope': 'main', 'some': { 'bindable': 'fun', 'params':
['a', 'int', 'b', 'int', 'type': 'int'] }, { 'scope': 'some' }]`




```
int some(int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```

Exemplo de


Como fazer?

- addVar(a)
- Table = [{ 'scope': 'main', 'some': { 'bindable': 'fun', 'params': ['a', 'int', 'b', 'int'], 'type': 'int' } }, { 'scope': 'some', 'a': { 'bindable': 'var', 'type': 'int' } }]



```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```

Exemplo de




```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```

Como fazer?

- addVar(b)
- Table = [{ 'scope': 'main', 'some': { 'bindable': 'fun', 'params': ['a', 'int', 'b', 'int'], 'type': 'int' } }, { 'scope': 'some', 'a': { 'bindable': 'var', 'type': 'int' }, 'b': { 'bindable': 'var', 'type': 'int' } }]

Exemplo de




```
int some (int a, int b){  
  a = 88 + 44;  
  b = 70;  
  sumparabola(1, 2, 3);  
  while (true){  
    c = 38;  
    sumparabola(5, true, false);  
    while (c){  
      sumparabola(5, true, true);  
    }  
  }  
  some();  
  sumparabolac(2);  
  return true;  
}
```

Como fazer?

- addVar(a)
- Table = [{ 'scope': 'main', 'some': { 'bindable': 'fun', 'params': ['a', 'int', 'b', 'int'], 'type': 'int' } }, { 'scope': 'some', 'a': { 'bindable': 'var', 'type': 'int' }, 'b': { 'bindable': 'var', 'type': 'int' } }]

Exemplo de




```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```

Como fazer?

- addVar(b)
- Table = [{ 'scope': 'main', 'some': { 'bindable': 'fun', 'params': ['a', 'int', 'b', 'int'], 'type': 'int' } }, { 'scope': 'some', 'a': { 'bindable': 'var', 'type': 'int' }, 'b': { 'bindable': 'var', 'type': 'int' } }]

Exemplo de




```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```

Como fazer?

- `getbindable('sumparabola')`
- `Table = [{ 'scope': 'main', 'some': { 'bindable': 'fun', 'params': ['a', 'int', 'b', 'int'], 'type': 'int' } },
 { 'scope': 'some', 'a': { 'bindable': 'var', 'type': 'int' }, 'b': { 'bindable': 'var', 'type': 'int' } }]`

Exemplo de



```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```

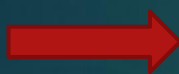
Como fazer?

- `getbindable('sumparabola')`
- `Table = [{ 'scope': 'main', 'some': { 'bindable': 'fun', 'params': ['a', 'int', 'b', 'int'], 'type': 'int' } },
 { 'scope': 'some', 'a': { 'bindable': 'var', 'type': 'int' }, 'b': { 'bindable': 'var', 'type': 'int' } }]`

Como não existe
bindable
sumparabola

ERRO SEMÂNTICO

Exemplo de



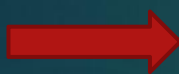
```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```

Como fazer?

- `Visitor.visitExp(self) == 'boolean'`

```
Table = [{'scope': 'main', 'some': {'bindable': 'fun', 'params': ['a', 'int', 'b',  
    'int'], 'type': 'int'}},  
        {'scope': 'some', 'a': {'bindable': 'var', 'type': 'int'}, 'b': {'bindable':  
    'var', 'type': 'int'}}]
```

Exemplo de



```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```

Como fazer?


- `Visitor.visitExp(self) == 'boolean'`

```
Table = [{'scope': 'main', 'some': {'bindable': 'fun', 'params': ['a', 'int', 'b',  
    'int'], 'type': 'int'}},  
        {'scope': 'some', 'a': {'bindable': 'var', 'type': 'int'}, 'b': {'bindable':  
    'var', 'type': 'int'}}]
```

Como true é
boolean

Ok !!!

Exemplo de



```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```


Como fazer?

- addVar(c)

```
Table = [{ 'scope': 'main', 'some': { 'bindable': 'fun', 'params': ['a', 'int', 'b',  
    'int'], 'type': 'int' } }, { 'scope': 'some', 'a': { 'bindable': 'var', 'type': 'int' }, 'b':  
    { 'bindable': 'var', 'type': 'int' }, 'c': { 'bindable': 'var', 'type': 'int' } } ]
```

Exemplo de

```
int some (int a, int b){  
  a = 88 + 44;  
  b = 70;  
  sumparabola(1, 2, 3);  
  while (true){  
    c = 38;  
    sumparabola(5, true, false);  
    while (c){  
      sumparabola(5, true, true);  
    }  
  }  
  some();  
  sumparabolac(2);  
  return true;  
}
```



Como fazer?

- getbindable('sumparabola')

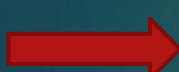
```
Table = [{ 'scope': 'main', 'some': { 'bindable': 'fun', 'params': ['a', 'int', 'b',  
  'int'], 'type': 'int' } }, { 'scope': 'some', 'a': { 'bindable': 'var', 'type': 'int' }, 'b':  
  { 'bindable': 'var', 'type': 'int' }, 'c': { 'bindable': 'var', 'type': 'int' } } ]
```

Como não existe
bindable
sumparabola

ERRO SEMÂNTICO

Exemplo de

```
int some (int a, int b){  
  a = 88 + 44;  
  b = 70;  
  sumparabola(1, 2, 3);  
  while (true){  
    c = 38;  
    sumparabola(5, true, false);  
    while (c){  
      sumparabola(5, true, true);  
    }  
  }  
  some();  
  sumparabolac(2);  
  return true;  
}
```



Como fazer?

- getbindable('sumparabola')

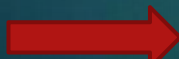
Table = [{ 'scope': 'main', 'some': { 'bindable': 'fun', 'params': ['a', 'int', 'b', 'int'], 'type': 'int' } }, { 'scope': 'some', 'a': { 'bindable': 'var', 'type': 'int' }, 'b': { 'bindable': 'var', 'type': 'int' }, 'c': { 'bindable': 'var', 'type': 'int' } }]

Como c é int

ERRO SEMÂNTICO

Exemplo de

```
int some (int a, int b){  
  a = 88 + 44;  
  b = 70;  
  sumparabola(1, 2, 3);  
  while (true){  
    c = 38;  
    sumparabola(5, true, false);  
    while (c){  
      sumparabola(5, true, true);  
    }  
  }  
  some();  
  sumparabolac(2);  
  return true;  
}
```



Como fazer?

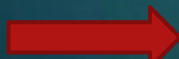
- getbindable('some')

```
Table = [{ 'scope': 'main', 'some': { 'bindable': 'fun', 'params': ['a', 'int', 'b',  
  'int'], 'type': 'int' } }, { 'scope': 'some', 'a': { 'bindable': 'var', 'type': 'int' }, 'b':  
  { 'bindable': 'var', 'type': 'int' }, 'c': { 'bindable': 'var', 'type': 'int' } } ]
```

Apesar de existir
bindable some, o
parâmetro da chamada é
incompatível.

Exemplo de

```
int some (int a, int b){  
    a = 88 + 44;  
    b = 70;  
    sumparabola(1, 2, 3);  
    while (true){  
        c = 38;  
        sumparabola(5, true, false);  
        while (c){  
            sumparabola(5, true, true);  
        }  
    }  
    some();  
    sumparabolac(2);  
    return true;  
}
```



Como fazer?

- getbindable('some')

```
Table = [{ 'scope': 'main', 'some': { 'bindable': 'fun', 'params': ['a', 'int', 'b',  
    'int'], 'type': 'int' } }, { 'scope': 'some', 'a': { 'bindable': 'var', 'type': 'int' }, 'b':  
    { 'bindable': 'var', 'type': 'int' }, 'c': { 'bindable': 'var', 'type': 'int' } } ]
```

Apesar de existir
bindable some, o
parâmetro da chamada é
incompatível.

AbstractVisitor

► Código

Código

- ▶ SemanticVisitor

TableSymbol

► Código