
Análise semântica (continuação)

Função, interação com o compilador
Tabela de símbolos
Análise semântica

Prof. Thiago A. S. Pardo

1

Tratamento semântico

- **Verificação do uso adequado** dos elementos do programa
 - **Declaração de identificadores**
 - Erro: identificador não declarado ou declarado duas vezes
 - **Compatibilidade de tipos** em comandos
 - Checagem de tipos
 - **Concordância entre parâmetros** formais e atuais, em termos de número, ordem e tipo

2

Tratamento semântico

- Verificador de tipos
 - Especificado na **gramática de atributos** e implementado como tal
 - Compilação de mais de uma passagem, possivelmente
 - Comandado pela **análise sintática**
 - Possibilidade de compilação de uma única passagem

3

Tratamento semântico

- Exemplo: verificação de tipos na gramática de atributos

```
<exp>1 ::= <exp>2 div id
  se busca(id)=falso
    então ERRO("variável não declarada")
  senão se exp2.tipo<>inteiro ou id.tipo<>inteiro
    então ERRO("tipos inválidos para a operação")
  se não ocorreu erro então
    exp1.tipo=inteiro
    exp1.val=exp2.val / id.val
```

4

Tratamento semântico

- Exemplo: verificação simples de tipos em uma regra sintática de atribuição de tipos iguais

procedimento atribuição(Seg)

Início

```
    se (simbolo=id)
        então obtem_simbolo(cadeia,simbolo)
            se busca(cadeia,simbolo,cat="var")=FALSE
                então ERRO("variável não declarada")
                senão tipo1:=recupera_tipo(cadeia,simbolo,cat="var");
            senão ERRO(Seg+{simb_atrib});
    se (simbolo=simb_atrib)
        então obtem_simbolo(cadeia,simbolo)
        senão ERRO(Seg+{id});
    expressao(tipo2);
    se tipo1<>tipo2 então ERRO("tipos incompatíveis na atribuição");
    se (simbolo=simb_ponto-virgula)
        então obtem_simbolo(cadeia,simbolo)
        senão ERRO(Seg+P(comandos));
```

fim

Na LALG

- Principais erros semânticos
 - Variável ou procedimento não declarado
 - Variável ou procedimento declarado mais de uma vez
 - Incompatibilidade de parâmetros formais e reais: número, ordem e tipo
 - Uso de variáveis de escopo inadequado
 - Atribuição de um real a um inteiro
 - Divisão que não é entre números inteiros
 - Readln e writeln com variáveis de tipo diferentes

Exercício

- Em duplas, especifique a gramática de atributos completa

```
<cmd-IO> ::= readln ( <lista_var> ) |  
           writeln ( <lista_var> )  
<lista_var> ::= ident <mais_var>  
<mais_var> ::= , <lista_var> | λ
```

7

Possível resposta

```
<cmd-IO> ::= readln ( <lista_var> )  
<cmd-IO> ::= writeln ( <lista_var> )  
<lista_var> ::= ident <mais_var>  
              se busca(ident)=falso  
                então ERRO("identificador não declarado")  
                lista_var.tipo="indefinido"  
              senão  
                ident.tipo=recupera_tipo(ident)  
                se (mais_var.tipo<>"") e (ident.tipo<>mais_var.tipo)  
                  então ERRO("tipos incompatíveis")  
                  lista_var.tipo="indefinido"  
                senão lista_var.tipo=id.tipo  
<mais_var> ::= , <lista_var>  
              mais_var.tipo=lista_var.tipo  
<mais_var> ::= λ  
              mais_var.tipo=""
```

8

Exercício

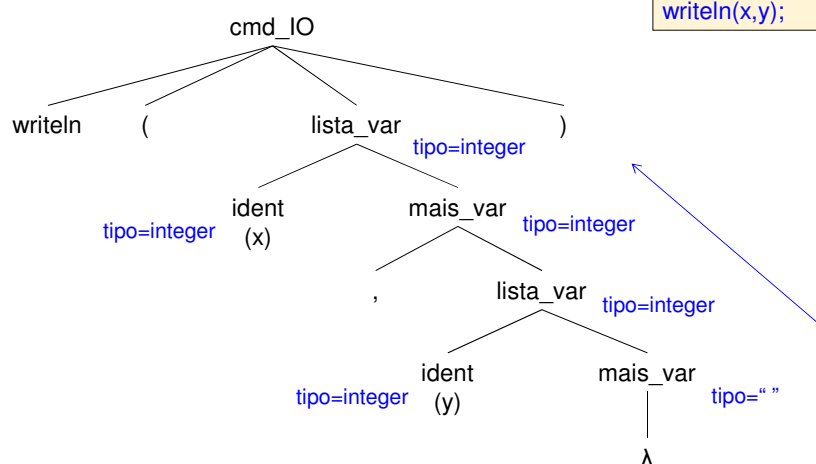
- Usando sua gramática de atributos, faça sobre a árvore sintática o cômputo dos atributos para o código destacado abaixo

```
program p;  
var x, y: integer;  
...  
writeln(x,y);
```

9

Possível resposta

```
program p;  
var x, y: integer;  
...  
writeln(x,y);
```



10

Exercício

- Usando sua gramática de atributos, implemente o procedimento sintático recursivo descendente correspondente

11

Possível resposta

```
procedimento cmd_IO(S)
início
se (símbolo=readln) ou (símbolo=writeln)
então obten_símbolo(cadeia,símbolo)
senão ERRO(S+{símb_abre_par})
se (símb=símb_abre_par)
então obten_símbolo(cadeia,símbolo)
senão ERRO(S+{id})
se (símb=id)
então
se busca(cadeia, token="id", cat="var")=false
então ERRO("identificador não declarado")
senão tipo1=recupera_tipo(cadeia, token="id", cat="var")
obtem_símbolo(cadeia,símbolo)
enquanto (símbolo=símb_virgula) faça
obtem_símbolo(cadeia,símbolo)
se (símb=id)
então
se busca(cadeia, token="id", cat="var")=false
então ERRO("identificador não declarado")
senão tipo2=recupera_tipo(cadeia, token="id", cat="var");
se (tipo1 <> tipo2)
então ERRO("tipos incompatíveis");
obtem_símbolo(cadeia,símbolo)
fim-então
senão ERRO(S+{símb_virgula,símb_fecha_par});
fim-enquanto
fim-então
se (símb=símb_fecha_par)
então obten_símbolo(cadeia,símbolo)
senão ERRO(S)
fim
```

```
<cmd-IO> ::= readln ( <lista_var> ) |
writeln ( <lista_var> )
<lista_var> ::= ident <mais_var>
<mais_var> ::= , <lista_var> | λ
```