

## Construção de Compiladores 1 - 2015.1 - Prof. Daniel Lucrédio

### Lista 07 - Análise semântica

1) Considere a DDS abaixo

Produção	Regras semânticas
$L \rightarrow E \ n$	$L.val = E.val$
$E \rightarrow E1 + T$	$E.val = E1.val + T.val$
$E \rightarrow T$	$E.val = T.val$
$T \rightarrow T1 * F$	$T.val = T1.val * F.val$
$T \rightarrow F$	$T.val = F.val$
$F \rightarrow ( E )$	$F.val = E.val$
$F \rightarrow \text{dígito}$	$F.val = \text{dígito.lexval}$

Construa as árvores sintáticas com o cálculo dos atributos para as seguintes expressões:

- a)  $(3 + 4) * (5 + 6) \ n$
- b)  $1 * 2 * 3 * (4 + 5) \ n$
- c)  $(9 + 8 * (7 + 6) + 5) * 4n$

2) Estenda a DDS abaixo de forma a tratar expressões completas como no exercício 1

Produção	Regras semânticas
$T \rightarrow F \ T'$	$T'.her = F.val$ $T.val = T'.sint$
$T' \rightarrow * \ F \ T'1$	$T'1.her = T'.her * F.val$ $T'.sint = T'1.sint$
$T' \rightarrow \epsilon$	$T'.sint = T'.her$
$F \rightarrow \text{dígito}$	$F.val = \text{dígito.lexval}$

3) Repita a letra a) do exercício 1 para a sua DDS do exercício 2

4) Considere a seguinte gramática simples de declarações de variáveis como na sintaxe de C:

```
decl → tipo varlista
tipo → 'int' | 'float'
varlista → id ',' varlista | id
```

- a) Construa a gramática de atributos para o atributo de tipo de dados, para o qual daremos o nome de tipoDados para diferenciar do não-terminal tipo
- b) Construa a árvore sintática com o cálculo dos atributos tipo\_dados para a cadeia `int a, b`
- c) Desenhe o grafo de dependência para a cadeia `int a, b` amarrado à árvore sintática construída na letra (b)
- d) Construa o esquema de TDS para a gramática do item (a)

5) O atributo tipoDados da questão 4 é sintetizado ou herdado? Por que?

6) Escreva um procedimento recursivo para calcular o atributo tipoDados da questão 4, em todos os nós necessários.

7) Enumere os nós da árvore construída na letra (c) da questão 4 indicando a ordem de computação de tipoDados de acordo com o algoritmo da questão 6. Que tipo de percurso é esse?

8) Considere a seguinte gramática simples de expressões com uma única operação, a divisão (/), e dois tipos de operandos: números inteiros (sequências de dígitos indicados pelo token num) e números de

ponto flutuante (indicados pela sequência num.num).

$S \rightarrow \text{exp}$

$\text{exp} \rightarrow \text{exp} \text{ '/' exp} \mid \text{num} \mid \text{num} \text{ '.' num}$

a) Construa uma gramática de atributos capaz de interpretar de maneira distinta a operação divisão dependendo do tipo de operandos envolvidos: se pelo menos um for ponto flutuante, a divisão será de ponto flutuante; se todos forem inteiros então a divisão será inteira. Três atributos deverão ser calculados: um que indica se a expressão é de ponto flutuante (**éFlut**), outro para o tipo da expressão (**tipoExp**) e um último para armazenar o valor da expressão (**val**).

Use **div** para divisão de inteira e **/** para divisão de ponto flutuante, assim  $5/2.0 = 1,25$  ( $5 / 2.0$ ) e  $5/2 = 2$  ( $5 \text{ div } 2$ ).

b) Construa a árvore sintática com o cálculo dos atributos para a cadeia **6/3.0**.

c) Desenhe o grafo de dependência para a cadeia **6/3.0** amarrado à árvore sintática construída na letra (b).

9) De que tipo (sintetizado ou herdado) é cada um dos 3 atributos calculados na questão 8? Justifique sua resposta.

10) Descreva com palavras (não precisa fazer o algoritmo) como seria o processo para calcular os 3 atributos da questão 8. Quantas passadas seriam necessárias para calculá-los e qual o percurso usado nesse cálculo?

11) Diga quais são os dois tipos de gramáticas de atributos apresentados em aula explicando quais são as características de cada uma delas.

12) Considerando-se a árvore sintática para uma dada cadeia, diga quais são os percursos mais indicados nessa árvore para o cálculo de um atributo sintetizado e de um atributo herdado. Por que? Qual desses dois tipos de atributos é mais fácil de ser calculado, por que?