

## Lista de Exercícios #08

### Análise Semântica, Esquemas S e L-Atribuído

1. Crie a árvore de derivação anotada para diversas entradas válidas da gramática seguinte:

$$\begin{aligned} L &\rightarrow E \mathbf{n} && \{ L.val = E.val \} \\ E &\rightarrow E_1 + T && \{ E.val = E_1.val + T.val \} \\ E &\rightarrow E_1 - T && \{ E.val = E_1.val - T.val \} \\ E &\rightarrow T && \{ E.val = T.val \} \\ T &\rightarrow T_1 * F && \{ T.val = T_1.val * F.val \} \\ T &\rightarrow T_1 / F && \{ T.val = T_1.val / F.val \} \\ T &\rightarrow F && \{ T.val = F.val \} \\ F &\rightarrow ( E ) && \{ F.val = E.val \} \\ F &\rightarrow \mathbf{digit} && \{ F.val = digit.lexval \} \end{aligned}$$

2. Crie a árvore de derivação anotada para diversas entradas válidas da gramática seguinte:

$$\begin{aligned} E &\rightarrow TE' && \{ E'.h = T.val; E.val = E'.s \} \\ E' &\rightarrow +TE'_1 && \{ E'_1.h = E'.h + T.val; E'.s = E'_1.s \} \\ E' &\rightarrow -TE'_1 && \{ E'_1.h = E'.h - T.val; E'.s = E'_1.s \} \\ E' &\rightarrow \epsilon && \{ E'.s = E'.h; \} \\ T &\rightarrow FT' && \{ T'.h = F.val; T.val = T'.s \} \\ T' &\rightarrow *FT'_1 && \{ T'_1.h = T'.h * F.val; T'.s = T'_1.s \} \\ T' &\rightarrow /FT'_1 && \{ T'_1.h = T'.h / F.val; T'.s = T'_1.s \} \\ T' &\rightarrow \epsilon && \{ T'.s = T'.h; \} \\ F &\rightarrow \mathbf{digit} && \{ F.val = digit.lexval; \} \end{aligned}$$

3. Crie o grafo de dependências para as diversas entradas válidas utilizadas no exercício 1..
4. Crie o grafo de dependências para as diversas entradas válidas utilizadas no exercício 2..
5. O que o seguinte esquema de tradução faz? Mostre exemplos.
- $$\begin{aligned} A &\rightarrow A_1 \mathbf{digit} && \{ A.val = A_1.val + digit.lexval; \} \\ A &\rightarrow \mathbf{digit} && \{ A.val = digit.lexval; \} \end{aligned}$$
6. Altere o esquema de tradução do exercício 5. para que seja colocado no atributo `val` do símbolo raiz o valor decimal correspondente à sequência de tokens. Por exemplo, para a sequência 345, o valor de *A.val* deve ser  $3 * 10^2 + 4 * 10^1 + 5 * 10^0$ .
7. Considerando o esquema de tradução resultante do exercício 6., mostre o comportamento da pilha para um certo número de entradas válidas e levando em conta uma análise descendente.
8. A gramática abaixo gera adições de constantes inteiras e ponto-flutuante. Quando dois inteiros são somados, o tipo resultante é inteiro; quando dois ponto-flutuantes são somados, o tipo resultante é ponto-flutuante; caso contrário, é um erro. Defina um esquema de tradução para determinar o tipo de cada subexpressão.
- $$\begin{aligned} E &\rightarrow E + T \\ E &\rightarrow T \\ T &\rightarrow r \\ T &\rightarrow i \end{aligned}$$
9. Considerando esta G, crie o grafo de dependência para *float id<sub>1</sub>, id<sub>2</sub>, id<sub>3</sub>*

```

D → T L      L.h = T.tipo
T → int      T.tipo = inteiro
T → float    T.tipo = flutuante
L → L1, id   L1.h = L.h
                adicionaTipo (id.key, L.h)
L → id        adicionaTipo (id.key, L.h)

```

10. O que devemos fazer em um esquema de tradução L-atribuído para realizar uma análise ascendente?

11. Considerando o esquema de tradução abaixo, o que acontece quando temos a entrada **x - 2 + y**?

```

E → T { R.h = T.ptr; } R { E.ptr = R.s; }
R → + T { R1.h = geraNo('+', R.h, T.ptr); } R1 { R.s = R1.s; }
R → - T { R1.h = geraNo('-', R.h, T.ptr); } R1 { R.s = R1.s; }
R → ε { R.s = R.h; }
T → ( E ) { T.ptr = E.ptr; }
T → id { T.ptr = geraFolha(id, id.nome); }
T → enum { T.ptr = geraFolha(num, num.val); }

```

12. Por que o esquema de tradução do exercício 11. não funciona em analisadores ascendentes? Altere o esquema de tradução do exercício 11. para que seja possível realizar uma análise ascendente. Mostre seu funcionamento através de entradas válidas.

13. O esquema de tradução do exercício 1. é perfeitamente adaptado para uma análise descendente. Por quê?

14. Altere o esquema de tradução do exercício 1. para que ele funcione de forma descendente considerando uma pilha. Mostre seu funcionamento com entradas válidas.

15. O esquema de tradução dirigida pela sintaxe que segue **traduz** uma linguagem com terminais a, b e c em uma linguagem cujos terminais são 2, 3, 4, 5, e 6. Usando um analisador **ascendente** que executa as ações semânticas imediatamente após reduzir a regra correspondente, diga o resultado da tradução da entrada **aaadbc**?

```

S → AB { print "1" }
S → AS { print "2" }
A → a { print "3" }
B → bC { print "4" }
B → dB { print "5" }
C → c { print "6" }

```

16. O esquema de tradução dirigida pela sintaxe que segue **traduz** uma linguagem com terminais a, b e c em uma linguagem cujos terminais são 2, 3, 4, 5, e 6. Usando um analisador **ascendente** que executa as ações semânticas imediatamente após reduzir a regra correspondente, diga o resultado da tradução da entrada **aaadbc**?

```

S → AB { S.node = new node (A.node, B.node); }
S → AS1 { S.node = new node (A.node, S1.node); }
A → a { A.node = new node ("a"); }
B → bC { B.node = new node ("b", C.node); }
B → dB { B.node = new node ("d", B.node); }
C → c { C.node = new node ("c"); }

```