

## Primeira Questão

A gramática de uma linguagem denominada SIMPLE é apresentada na Figura 1, abaixo. O símbolo ID representa um identificador. A Figura 2 apresenta um exemplo de programa escrito na linguagem SIMPLE.

```
P -> C ";" P | λ
C -> ID "=" E | "print" E
E -> T | E "+" T | E "-" T
T -> F | T "*" F | T "/" F
F -> ID | "(" E ")"
```

Fig.1: Gramática da linguagem SIMPLE.

```
a = x + y / z;
print (x + y) * z;
```

Fig.2: Exemplo de programa.

Considere um esquema de tradução para a gramática de SIMPLE em que uma variável “cont” é acessada globalmente, com valor inicial 0. Suponha que um atributo “i” inteiro esteja associado aos símbolos E, T e F. As ações do esquema são definidas abaixo (mostrando apenas as produções que têm ações semânticas).

```
E -> T { // ação 1
  E.i = T.i;
}
```

```
E1 -> E2 "+" T { // ação 2
  imprime E2.i, T.i, cont;
  E1.i = cont; ++cont;
}
```

```
E1 -> E2 "-" T { // ação 3
  imprime E2.i, T.i, cont;
  E1.i = cont; ++cont;
}
```

```
T -> F { // ação 4
  T.i = F.i;
}
```

```
T1 -> T2 "*" F { // ação 5
  imprime T2.i, F.i, cont;
  T1.i = cont; ++cont;
}
```

```
T1 -> T2 "/" F { // ação 6
  imprime T2.i, F.i, cont;
  T1.i = cont; ++cont;
}
```

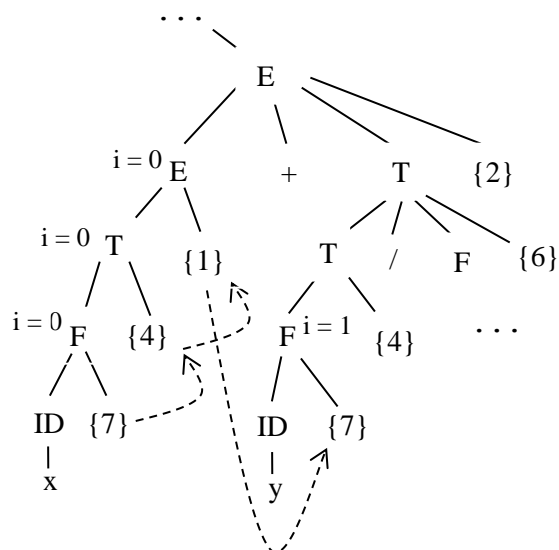
```
F -> ID { // ação 7
  imprime ID "->" cont;
  F.i = cont; ++cont;
}
```

```
F -> "(" E ")" { // ação 8
  F.i = E.i;
}
```

Apresente uma árvore de derivação anotada para o programa da Figura 2, mostrando o valor do atributo “i” nos nós da árvore e onde as ações semânticas são inseridas.

Mostre também o que será impresso, considerando a ordem da avaliação das ações, e o valor de “cont” (global) sendo alterado.

Observe, ao lado, um exemplo de trecho da árvore desenhada, com atributos e ações, e o que está sendo impresso.



cont  
0

Saída produzida:

```
x->0
y->1
...
```

# Segunda Questão

A gramática de uma linguagem denominada DECL é apresentada na Figura 1, abaixo. O símbolo ID representa um identificador, e NUM representa um número inteiro. A Figura 2 apresenta um exemplo de programa escrito na linguagem DECL.

```
P -> D P | λ
D -> T V ";"
T -> "int" | "double"
V -> W Z
W -> ID | ID "[" NUM "]"
Z -> "," W Z | λ
```

Fig.1: Gramática da linguagem DECL.

```
int a, b[5];
double x[4];
```

Fig.2: Exemplo de programa.

Considere o esquema de tradução abaixo, para a gramática de DECL, em que ID possui um atributo “text” contendo o nome do identificador associado e NUM possui um atributo “val” contendo o valor do número inteiro associado. Os atributos “text” e “val” são automaticamente definidos por um processo externo (ex: analisador léxico). Os símbolos W e Z possuem um atributo “i” que armazena um número inteiro, calculado pelos comandos do esquema de tradução. Além disso, uma variável “cur” é acessada globalmente.

<pre>P -&gt; D P   λ D -&gt; T V ";" T -&gt;   "int"   { // ação 1     cur = 4; }       "double"   { // ação 2     cur = 8; }</pre>	<pre>V -&gt;   W   { // ação 3     Z.i = W.i; }   Z W -&gt;   ID   { // ação 4     W.i = cur;     print(ID.text, W.i); }       ID "[" NUM "]"   { // ação 5     W.i = cur * NUM.val;     print(ID.text, W.i); }</pre>	<pre>Z<sub>1</sub> -&gt;   "," W   { // ação 6     Z<sub>2</sub>.i = Z<sub>1</sub>.i + W.i; }   Z<sub>2</sub>       λ   { // ação 7     print(Z<sub>1</sub>.i); }</pre>
---	---	---

Apresente uma árvore de derivação anotada para o programa da Figura 2, mostrando o valor dos atributos “i” nos nós da árvore e onde as ações semânticas são inseridas. Observe que as ações 1, 2, 4, 5 e 7 estão após o último símbolo do lado direito das produções em que aparecem; as ações 3 e 6 estão entre os símbolos do lado direito das produções em que aparecem.

Mostre também o que será impresso, considerando a ordem da avaliação das ações, e o valor de “cur” (global) sendo alterado.