

Lista de Exercícios #10

Geração de Código

1. Transforme a gramática abaixo em um esquema de tradução que gere TAC.

```
S → x = E;  
E → E1 + T  
E → E1 - T  
E → T  
T → T1 * F  
T → T1 / F  
T → F  
F → ( E )  
F → digit  
F → var
```

2. Suponha que os operadores do código TAC (+ - */) só suportem operandos do mesmo tipo e que existe somente dois tipos na linguagem: **int** e **float**. Altere o esquema de tradução do exercício 1. para utilizar apropriadamente os operadores TAC. Mostre exemplos que estressam a funcionalidade.
3. Suponha que uma linguagem tenha apenas escopo global. Declarações de variáveis dentro de funções não são autorizadas nesta linguagem. Crie um esquema de tradução para calcular os endereços dessas variáveis em relação ao endereço base do segmento de dados. Esta informação deve ser mantida na tabela de símbolos.
4. Altere o esquema de tradução do exercício 3. para gerar TAC considerando diferentes entradas válidas de expressões aritméticas. O acesso ao conteúdo das variáveis é feito levando-se em conta o endereço base do segmento de dados.
5. Suponha que os operandos + - */ do código TAC não aceitam endereços como parâmetros. Por exemplo, o seguinte código TAC é ilegal. Na linha 1, temos um endereço no primeiro parâmetro do operando *; na linha 2, temos um endereço no segundo parâmetro do operando +:

```
t0 = (fp+0) * 5.2  
t1 = t0 + (fp+8)
```

Sabendo desta limitação, altere o esquema de tradução resultante do exercício 4. para que os valores em memória (parâmetros de endereçamento) sejam primeiramente colocados em registradores para que os operandos possam trabalhar sobre eles. Utilize o operando **load** do nosso TAC hipotético para carregar os valores da memória para os registradores. Exemplo:

```
t0 = load (fp+0)  
t1 = load (fp+8)  
t2 = t0 * 5.2  
t3 = t2 + t1
```

6. Crie um esquema de tradução para a declaração de variáveis com escopo aninhado. Mostre o funcionamento com entradas válidas.
7. Incremente o esquema de tradução resultante do exercício 6. para a geração de código TAC para expressões aritméticas. Considere as limitações TAC apresentadas no exercício 5..
8. Como você implementaria a geração de código para o comando de atribuição em uma linguagem com escopo aninhado?

9. Considere uma linguagem orientada a objetos com o conceito de herança. Como você implementaria o endereçamento dos atributos de instância?