

COMPILADORES

PROVA 2 – 28 DE JANEIRO DE 2020

PROF. MARCUS RAMOS

Questão 1 (1,5 ponto): Considere o comando de atribuição $x := y + 1$ da linguagem Pascal (ou mini-Pascal). Quantas e quais são as verificações de contexto que devem recair sobre o mesmo?

Deve-se, pela ordem:

- Verificar que os identificadores x e y foram previamente declarados e são visíveis no local onde o comando é executado;
- Verificar que x e y foram declarados como variáveis;
- Verificar que o tipo de y é compatível com a operação de soma;
- Inferir o tipo de $y + 1$;
- Verificar que o tipo de $y + 1$ é compatível com o tipo de x .

Questão 2 (1,5 ponto): Quais são os objetivos da subfase de verificação de tipos em um analisador de contexto? Por que a subfase de identificação deve acontecer antes da subfase de verificação de tipos?

Os objetivos são:

- Inferir o tipo de todas as expressões e subexpressões usadas no programa-fonte;
- Determinar se os operandos possuem tipos compatíveis com as operações que estão sendo invocadas.

Deve acontecer antes pois é através da identificação que são obtidas informações sobre o tipo dos nomes declarados (como por exemplo as variáveis).

Questão 3 (1,5 ponto): Mostre o código gerado (TAM) para a expressão:

$$((a+2) * c = (b > d)) \&\& (a < b - 1)$$

Considere as precedências usuais, o uso dos parênteses e a arquitetura de máquina de pilha.

LOAD	a
LOADL	2
CALL	add
LOAD	c
CALL	mult
LOAD	b
LOAD	d
CALL	gt
CALL	eq
LOAD	a
LOAD	b
LOADL	1
CALL	sub

CALL lt
CALL and

Questão 4 (1,5 ponto): Considere a seguinte declaração em Pascal:

```
var m: array [2..5] of array [3..8] of integer;
```

Suponha que address [m]=1000 e size [integer] = 2. Pergunta-se:

- Qual o endereço de m[4]?

address [m] + (4 - 2) * (8 - 3 + 1) * 2 = 1000 + 2 * 12 = 1024

- Qual o endereço de m[4][4]?

address [m] + (4 - 2) * (8 - 3 + 1) * 2 + (4 - 3) * 2 = 1000 + 2 * 12 = 1036 + 2 * 1 = 1038

- Qual o endereço de m[i][j]?

address [m] + (i - 2) * (8 - 3 + 1) * 2 + (j - 3) * 2 = 1000 + 12 * i - 24 + 2 * j - 6 =
1000 - 24 - 6 + 12 * i + 2 * j = 970 + 12 * i + 2 * j

- Qual a origem virtual de m?

1000 - 24 - 6 = 970

Questão 5 (2,0 pontos): Descreva, de forma sucinta, as três formas de alocação de memória estudadas em sala de aula. Indique onde elas são usadas e como funcionam.

- Alocação estática: usada para variáveis globais e locais estáticas, que são variáveis cujo tempo de vida coincide com o tempo total de execução do programa. As variáveis são criadas no início da execução do programa e são destruídas no final da execução do programa.
- Alocação automática: usada para variáveis locais não-estáticas. O tempo de vida delas vai do instante em que o bloco onde elas foram declaradas começa a execução até o término da execução deste mesmo bloco. Elas são alocadas numa pilha, pois a desalocação acontece na ordem inversa da alocação.
- Alocação dinâmica: a criação de uma variável acontece sob o comando explícito do programador numa área chamada de heap. Dependendo da linguagem, a desalocação pode ser feita sob comando do programador (Pascal, C) ou então de forma automática por um coletor de lixo (Java). Pode produzir uma fragmentação excessiva e indesejada do heap. Indicada para estruturas dinâmicas.

Questão 6 (2,0 pontos): Usando os templates de código estudados em sala de aula, mostre o código gerado (TAM) para o seguinte fragmento de programa Pascal:

```
while (a<b) do if (a=c) then a := a + 1 else a := a + 2
```

L_0000

```
LOAD    a  
LOAD    b  
CALL    lt
```

```

JUMPIF (0) L_0001
LOAD a
LOAD c
CALL eq
JUMPIF (0) L_0002
LOAD a
LOADL 1
CALL add
STORE a
JUMP L_0003
L_0002
LOAD a
LOADL 2
CALL add
STORE a
L_0003
JUMP L_0000
L_0001
```