
Listas Genéricas (não homogêneas) e Listas Generalizadas

Exercícios e Aplicações

18/10/2010

Exercícios

1. Implementar uma função que determina a profundidade máxima de uma lista generalizada
 - ❑ Faça a versão recursiva (3 casos)

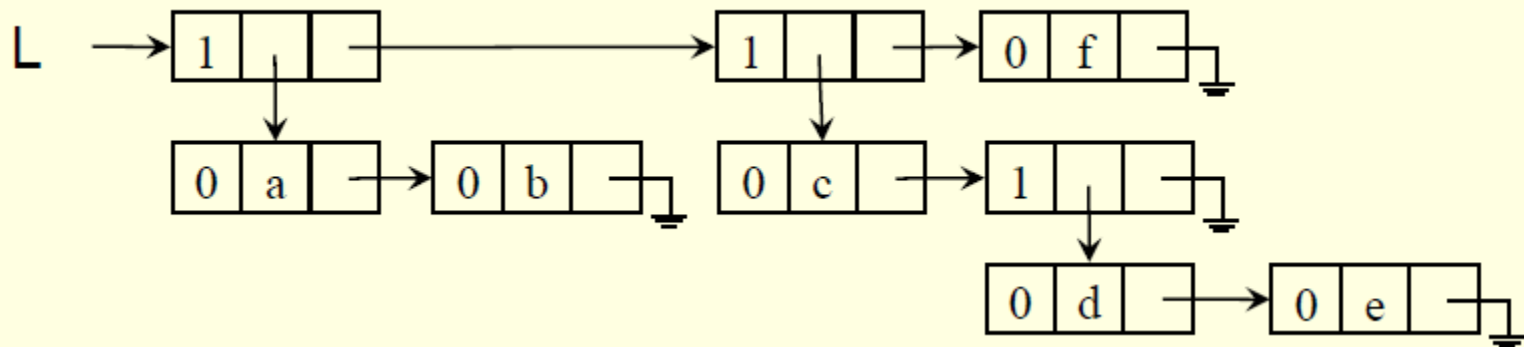
$L = (a, ()) \Rightarrow \text{Prof}(L) = 1$

$S = (a, (b)) \Rightarrow \text{Prof}(S) = 2$

$A = (a, b, c) \Rightarrow \text{Prof}(A) = 1$

$B = () \Rightarrow \text{Prof}(B) = 0;$

Por exemplo, para o caso abaixo, a função deveria retornar profundidade 3



```
int Prof(noptr S){
```

```
int prof, aux;
```

```
if (S==NULL)
```

```
    prof=0;
```

```
else if (S->tag==0) {
```

```
    prof=1;
```

```
    aux=Prof(S->cdr);
```

```
    if (aux > prof) prof = aux;
```

```
}
```

```
else if (S->tag==1) {
```

```
    prof = 1 + Prof(S->elemento.car_lista);
```

```
    aux = Prof(S->cdr);
```

```
    if (aux > prof) prof = aux;
```

```
}
```

```
return(prof);
```

```
}
```

2. Implementar uma sub-rotina para verificar se duas listas generalizadas são estruturalmente iguais

- O conteúdo em si não importa
 - 4 casos:
 - ❑ Caso 1: 2 listas vazias (true)
 - ❑ Caso 2: uma das listas é vazia e a outra não (false)
 - ❑ Caso 3: 2 listas não vazias
 - 2 átomos -- se Iguais_Est na cauda (true)
 - 2 listas – se Iguais_Est na car_lista e Iguais_Est na cauda (true)
 - ❑ Caso 4: se diferem na car_lista (false)
-

//Verifica se duas listas são iguais na estrutura; conteúdo não importa

```
int Iguais_Est(noptr L1, noptr L2){
if ((L1==NULL) && (L2==NULL))
    return 1;
else if ((L1==NULL) || (L2==NULL))
    return 0;
else // as 2 listas não são vazias
    if ((L1->tag== 0) && (L2->tag==0))
        return(Iguais_Est(L1->cdr,L2->cdr));
    else if (((L1->tag==1) && (L2->tag==1)) &&
        (Iguais_Est(L1->elemento.car_lista,L2->elemento.car_lista)))
        return(Iguais_Est(L1->cdr,L2->cdr));
    else return 0; // diferem na car_lista
}
```

Lista generalizada e polinômios

Considere os polinômios:

$$P1 = 4x^2y^3z + 3xy + 5$$

$$P2 = x^{10}y^3z^2 + 2x^8y^2z^2 + x^4y^4z + 6x^3y^4z + 2yz$$

$$P3 = 3x^2y$$

(a) n°de termos: variável

- $P1=3, P2=5, P3=1$

(b) n°de variáveis: variável

- $P1=P2=3, P3=2$

(c) nem todo termo é expresso com todas as variáveis

Lista generalizada e polinômios

- Objetivos

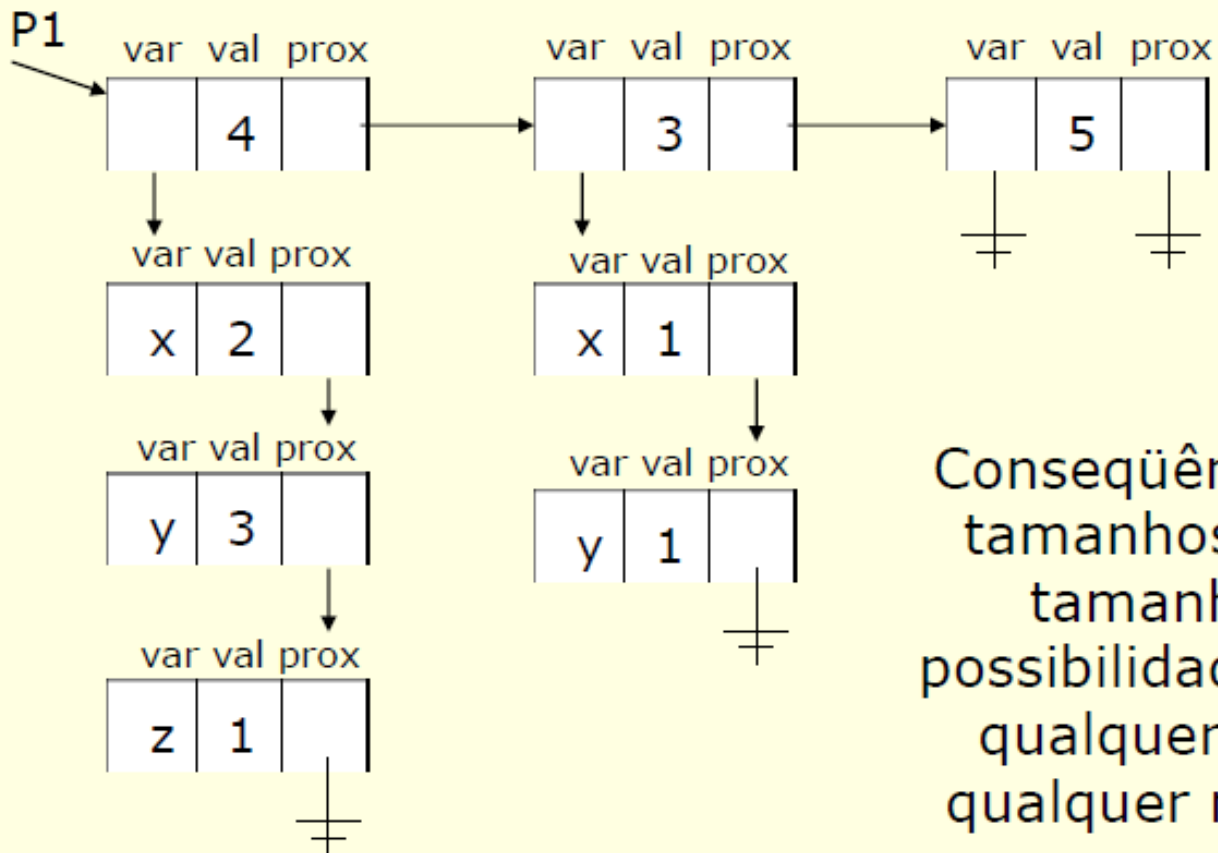
- representar de forma a otimizar o uso de memória
 - representação única para todo polinômio

- Solução: lista generalizada

- As variáveis são sempre ordenadas: x, y, z

Lista generalizada e polinômios

Ex: $P1 = 4x^2y^3z + 3xy + 5$



Conseqüência: registros de tamanhos fixos; listas de tamanhos variáveis; possibilidade de representar qualquer polinômio com qualquer n° de variáveis e qualquer grau

Implementação

Declaração em C: Union

```
struct bloco {
```

```
    int tipo; 0 = expoente; 1 = coef
```

```
    union {
```

```
        char atomo;
```

```
        struct bloco *sublista;
```

```
    } var;
```

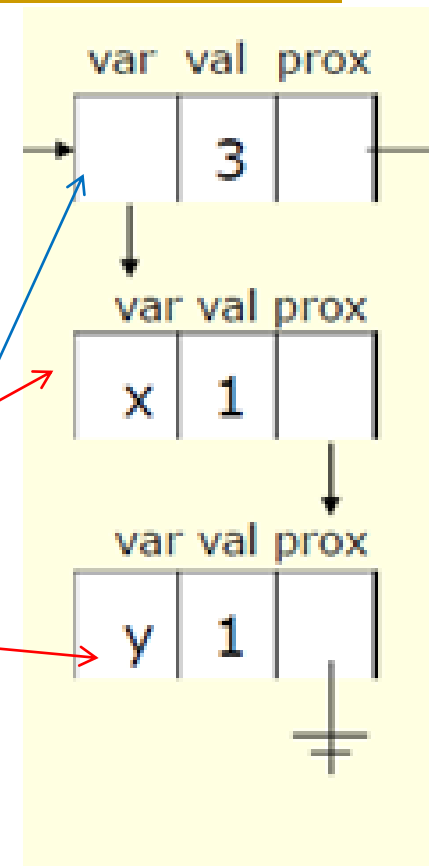
```
    float val;
```

```
    struct bloco *prox;
```

```
};
```

```
typedef struct bloco *pol;
```

```
pol lista;
```



Falta representar o campo tipo no desenho acima

Exercício

- Implementar uma função que:
 - (a) receba um polinômio representado via lista generalizada e os valores das variáveis
 - (b) percorra a lista generalizada e compute o resultado do polinômio
 - (c) retorne o resultado para quem chamou a função
- Quantas variáveis tem um dado polinômio?
- Como verificar se 2 polinômios são iguais?

Lista Linear

- Uma lista é chamada de linear se todos os seus elementos forem átomos.

