

SCC122 – Estruturas de Dados

Listas Generalizadas e Polinômios

Profa. Roseli Ap. Francelin Romero

2º. Semestre de 2011

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 (n_term): variável
 - $P1 = 3, P2 = 5, P3 = 1$
- (b) n°. de variáveis (n_var): 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 mais simples:
 - Lista Dinâmica Encadeada
 - Nós representados através de uma struct com ***n_var + 2*** campos, por exemplo:
 - Considere o polinômio:
 - $P2 = x^{10}y^3z^2 + 2x^8y^2z^2 + x^4y^4z + 6x^3y^4z + 2yz$

Lista generalizada e polinômios

- Solução mais simples:

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

- $n_var = 3$ (x , y e z)

- Struct

- typedef struct termo{

```
int exp_x;
```

```
int exp_y;
```

```
int exp_z;
```

```
int coeficiente;
```

```
termo* prox;
```

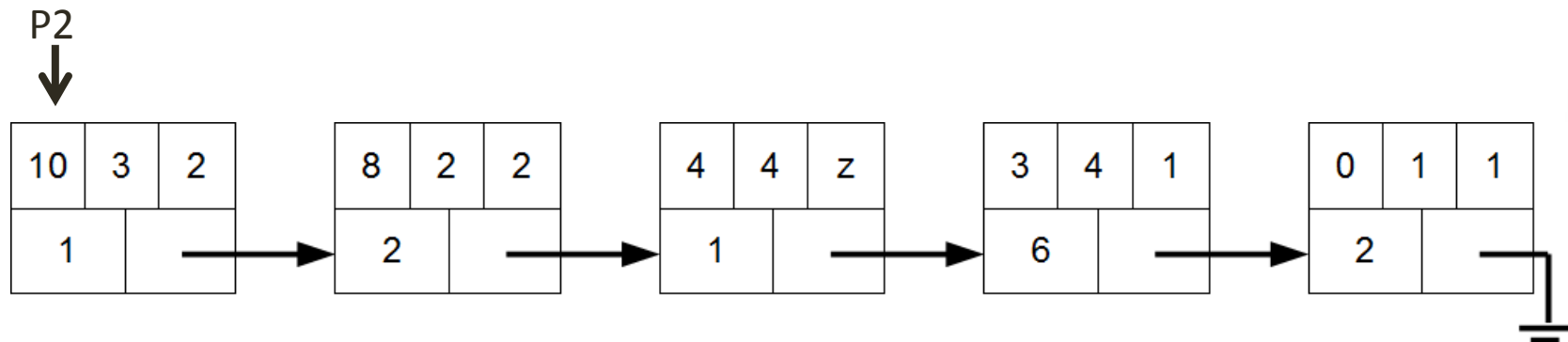
```
};
```

exp x	exp y	exp z
coeficiente		prox

Lista generalizada e polinômios

- Solução mais simples:

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

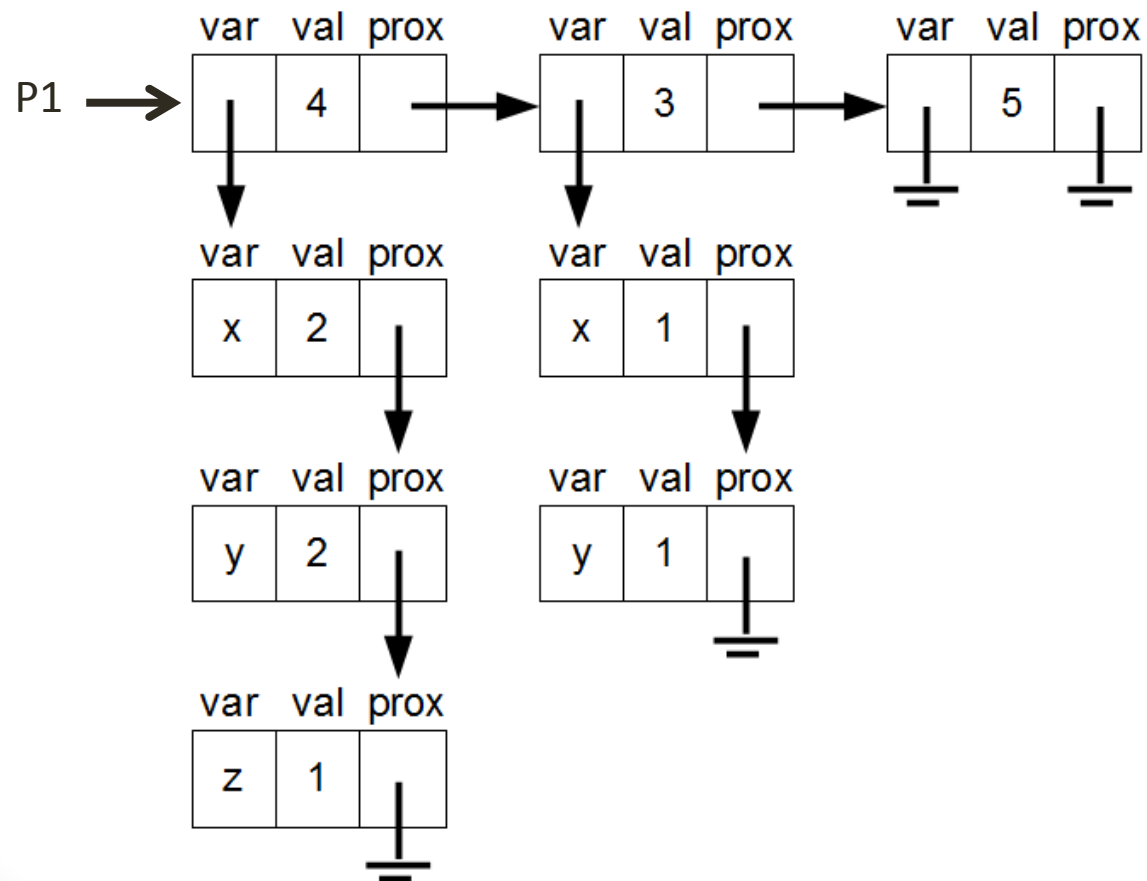


Lista generalizada e polinômios

- Outras soluções:
 - Lista generalizada
 - As variáveis são sempre ordenadas, por exemplo, no caso dos polinômios dados na ordem x, y, z :
- Consequências:
 - Registros de tamanho fixo (3 campos);
 - Listas de tamanho variável;
 - Possibilidade de representar qualquer polinômio com qualquer n° de variáveis e qualquer grau.

Lista generalizada e polinômios

- Outras soluções: Listas generalizadas
 - Exemplo: $P1 = 4x^2y^3z + 3xy + 5$



Lista generalizada e polinômios

- Outras soluções: Listas generalizadas
 - Declaração em C: Comando Union
 - Exemplo: Caso anterior

//Inicialmente é necessário a declaração dos 2 union

```
typedef union{
    char atomo;
    struct termo* sublista;
}var;

typedef union{
    int coeficiente; //ou float
    int expoente;
}val;
```


Lista generalizada e polinômios

- Outras soluções: Listas generalizadas
 - Declaração em C: Comando Union

//Declaração da struct

```
typedef struct{
    int tipo_var; //0 = char; 1 = sublista
    int tipo_val; //0 = expoente; 1 = coeficiente
    var Variavel;
    val Valor;
    struct termo *prox;
}termo;
```

```
typedef struct termo *polinomio;
polinomio lista;
```

Lista generalizada e polinômios

- Outras soluções: Listas generalizadas
 - Uma outra possibilidade utilizando o mesmo conceito de listas generalizadas, porém com uma contextualização diferente, seria o fatoramento do polinômio como uma etapa de pré-processamento para o armazenamento dos dados.
- Exemplo:
 - Considerando o polinômio P2. Uma possível representação com fatoramento seria:

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

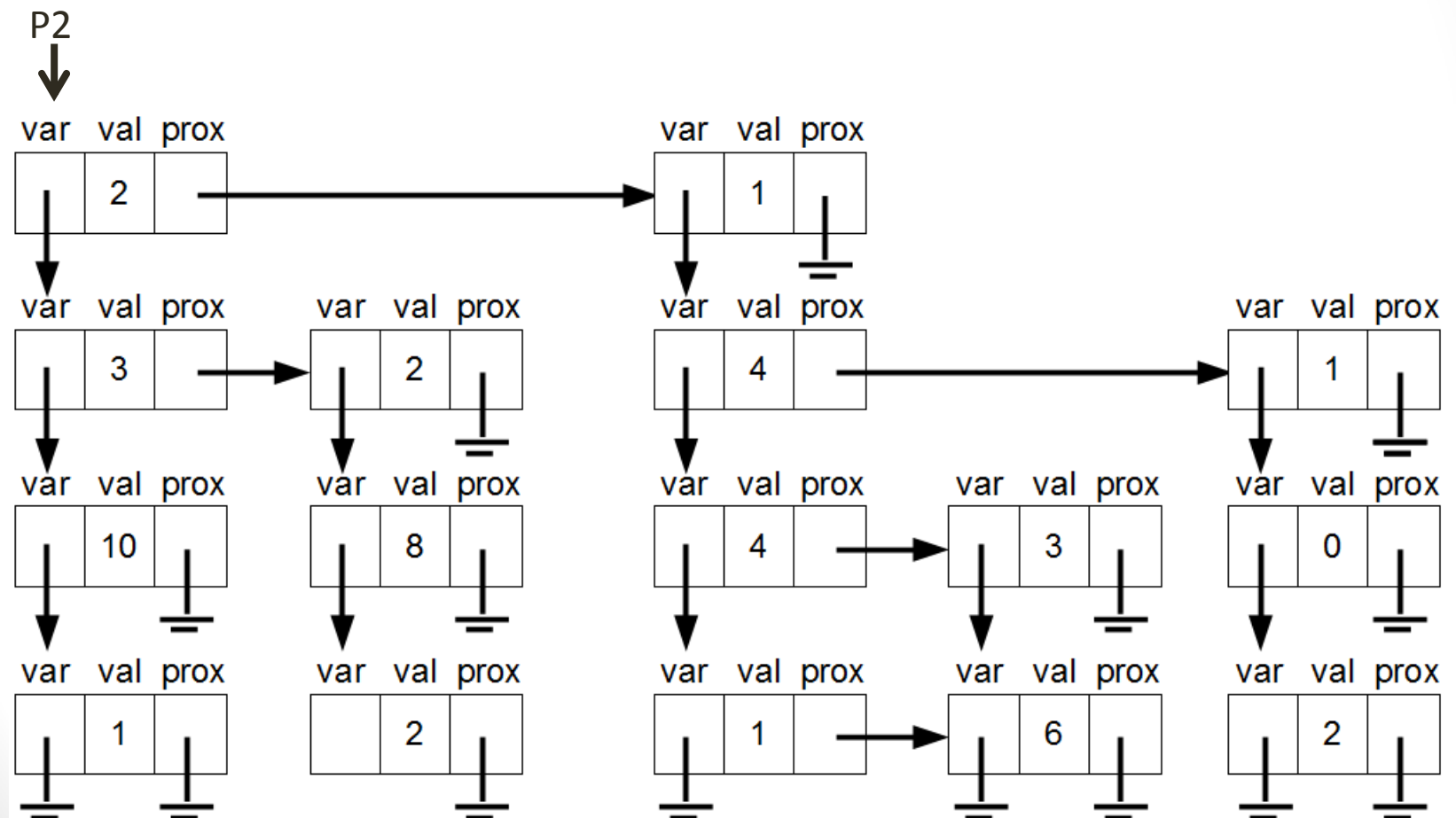
$$P2 = (x^{10}y^3 + 2x^8y^2) z^2 + (x^4y^4 + 6x^3y^4 + 2y)z$$

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

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

Lista generalizada e polinômios

- Exemplo: P2 com fatoramento



Lista generalizada e polinômios

- Exemplo: P2 com fatoramento
- Note que para o caso anterior o campo *var* da *struct* não seria um *union*, uma vez que ele se comporta apenas como um ponteiro para o próximo nível do polinômio.
- Além disso, cada nível da lista generalizada neste caso possui uma função diferente para o campo *val*.
 - Nível 1: expoentes da variável *z*
 - Nível 2: expoentes da variável *y*
 - Nível 3: expoentes da variável *x*
 - Nível 4: coeficientes do polinômio

Lista generalizada e polinômios

- Exercícios

1. Implementar uma função que determina a profundidade máxima de uma lista generalizada.

- Faça a versão recursiva (3 casos)

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

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

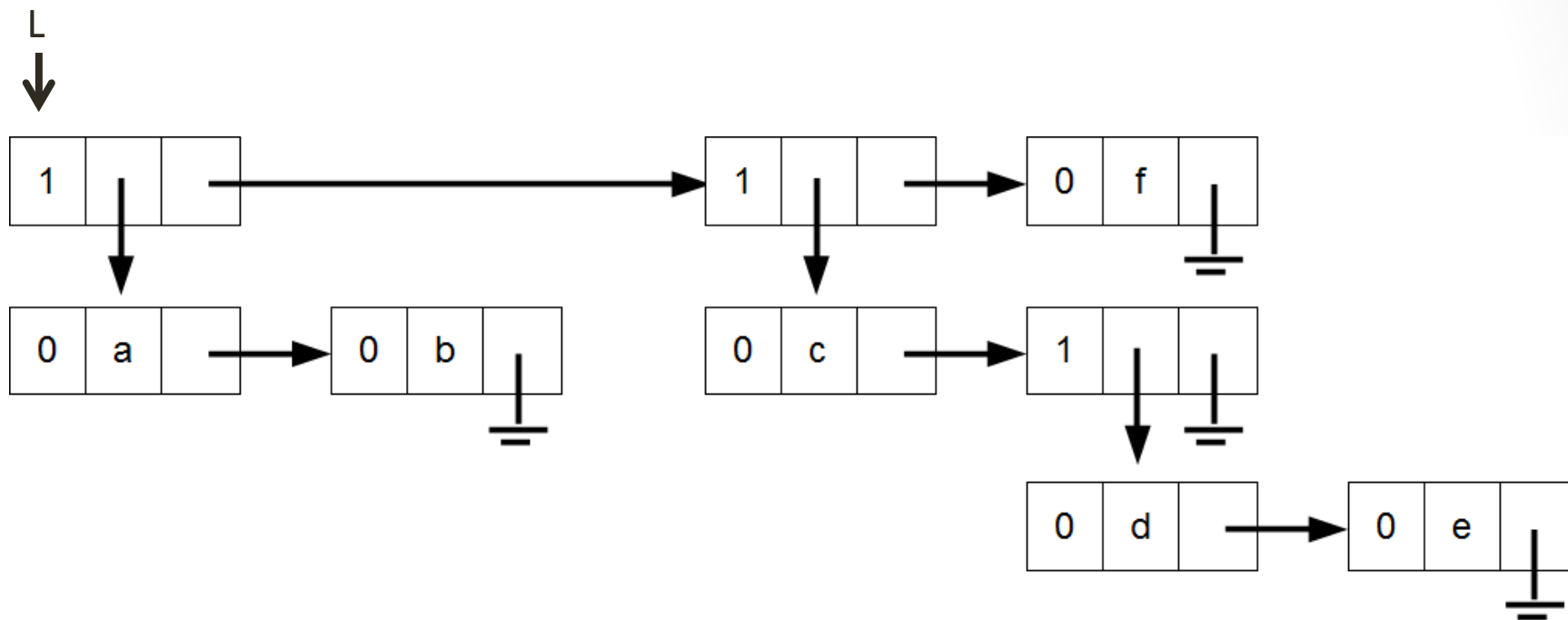
$$A = (a,b,c) \Rightarrow \text{Prof}(A) = 1$$

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

- Por exemplo, o caso a seguir, a função deveria retornar profundidade 3

Lista generalizada e polinômios

- Exercício (1)



Lista generalizada e polinômios

- Exercício (1)

```
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);
```

```
}
```

Lista generalizada e polinômios

- Exercícios

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: duas listas não vazias
 - 2 átomos – se Iguais_Est na cauda (true)
 - 2 listas – se iguais na car_lista e Iguais_Est na cauda (true)
 - Caso 4: diferem na car_lista (false)

Lista generalizada e polinômios

- Exercício (2)

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

```
int Iguais_Est(noptr L1, noptrL2){
if ((L1 == NULL) && (L2 == NULL)) return 1;
else if ( ((L1 != NULL) && (L2 == NULL)) || ((L1 == NULL) && (L2 != NULL)) )
    return 0;
else //as duas listas são não 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

- Exercícios

3. 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?