

SCC0202 – Algoritmos e Estruturas de Dados I

Listas Generalizadas

Prof.: Dr. Rudinei Goularte

(rudinei@icmc.usp.br)

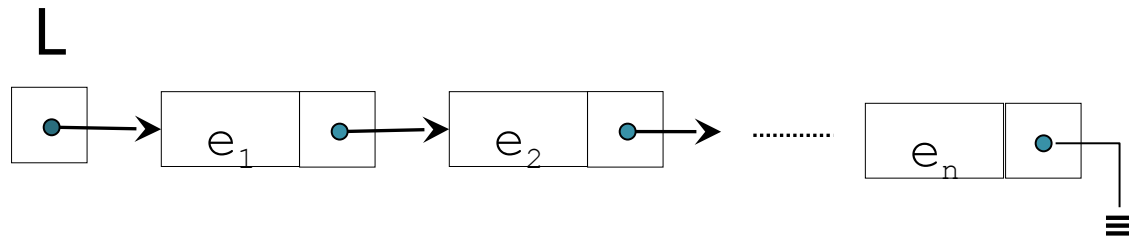
Instituto de Ciências Matemáticas e de Computação - ICMC

Sala 4-229

Generalizando o conceito de lista

2

- Uma lista $L = (e_1, e_2, e_3, e_4, \dots, e_n)$ pode ser definida como uma sequência formada por um elemento e_1 seguido por uma lista $(e_2, e_3, e_4, \dots, e_n)$ que é definida recursivamente de maneira análoga até que a lista (e_n) seja formada por e_n seguida de lista vazia $()$.
- Lista encadeada dinâmica ilustra bem essa situação



Generalizando o conceito de Lista

- Até agora... consideramos que todos os elementos e_i são do mesmo tipo e um **átomo**, ou seja, indivisível (não lista).
- Mas... cada elemento e_i da lista pode também ser uma lista, chamada **sub-lista**.

$$L = (a, (b, c), d, (e), ())$$

$\begin{array}{ccccc} | & \underbrace{\hspace{1.5cm}} & | & \underbrace{\hspace{1.5cm}} & \underbrace{\hspace{1.5cm}} \\ e_1 & e_2 & e_3 & e_4 & e_5 \end{array}$

- L em 5 elementos
 - e_1 e e_3 são **átomos**
 - e_2 , e_4 e e_5 são **sub-listas**

Lista generalizada

4

- Uma **lista generalizada** é aquela que pode ter como elemento ou um átomo ou uma outra lista (sub-lista)
 - ▣ Átomo: integer, real, char, string, etc.

- **Definição formal**
 - ▣ Uma lista generalizada A é uma sequência finita de n elementos $\alpha_0, \alpha_1, \dots, \alpha_n$, em que α_i são átomos ou listas. Os elementos α_i , com $0 \leq i \leq n$, que não são átomos são chamados sublistas de A.

Lista generalizada

- Alguns conceitos...
 - ▣ **Tamanho da Lista:** número de elementos da lista
 - ▣ **Profundidade da Lista:** quantidade de níveis de aninhamento de sub-listas
 - $L = () \Rightarrow \text{Profundidade}(L) = 0;$
 - ▣ Cabeça e cauda
 - ▣ Cabeça: primeiro elemento da lista (átomo ou lista)
 - ▣ Cauda: o restante (**sempre** uma outra lista, mesmo que vazia)

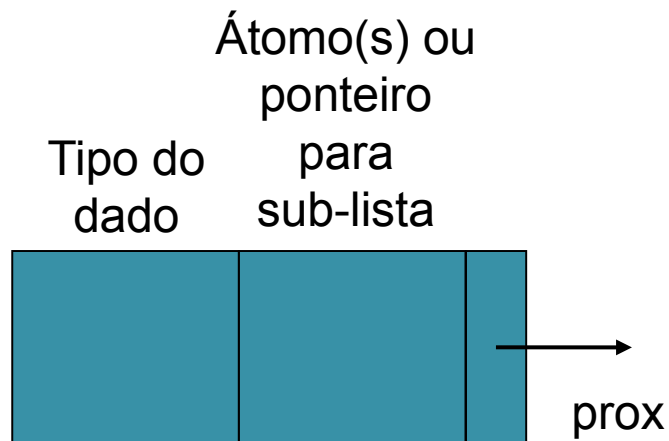
Lista generalizada

6

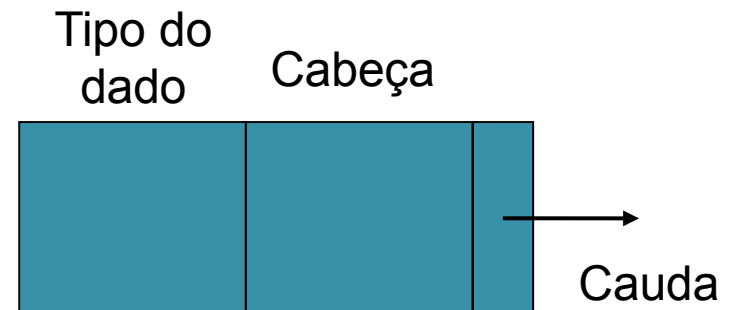
- Suponha que uma lista seja representada por elementos entre parênteses (no estilo da linguagem de programação LISP) ou entre colchetes (no estilo de PROLOG)
 - ▣ (a,b,c) ou [a,b,c]
 - ▣ (a,(b,c)) ou [a,[b,c]]
 - ▣ (a,(b),(c)) ou [a,[b],[c]]
 - ▣ (a,b,()) ou [a,b,[]]

Lista generalizada

- Tipo=0 indica átomo e tipo=1 indica sub-lista
- Estrutura básica do nó



ou

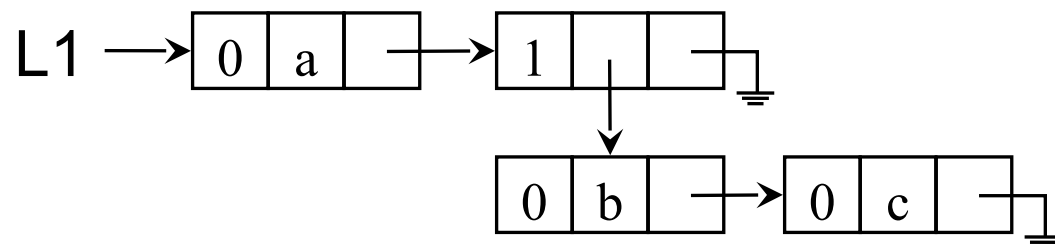


Lista generalizada

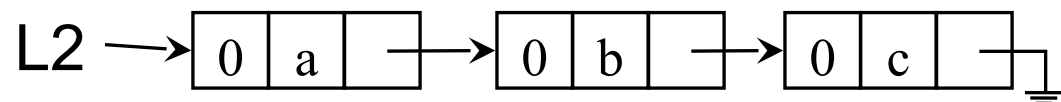
8

□ Exemplos de representação

L1 = (a,(b,c))



L2 = (a,b,c)

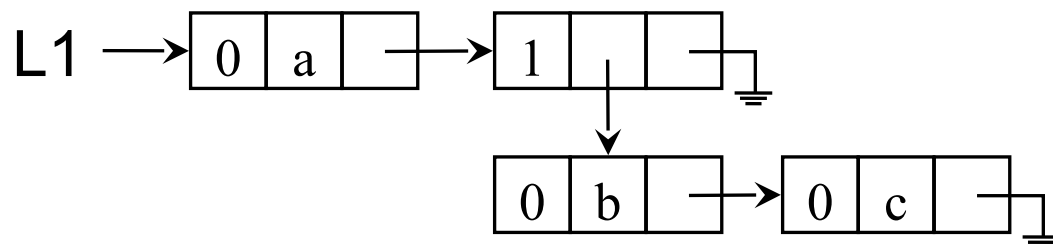


Lista generalizada

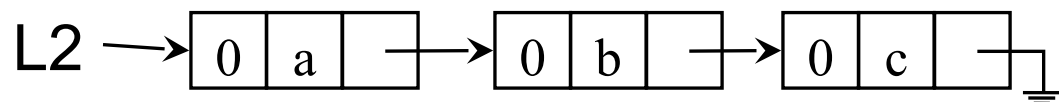
9

□ Exemplos de representação

$L1 = (a, (b, c))$



$L2 = (a, b, c)$



Profundidade (L2)? Qtd elementos (L2)?

Cabeça(L2)? Cauda(L2)? Cabeça(Cauda(L2))?

Cabeça(L1)? Cauda(L1)? Cabeça(Cauda(L1))?

Lista generalizada

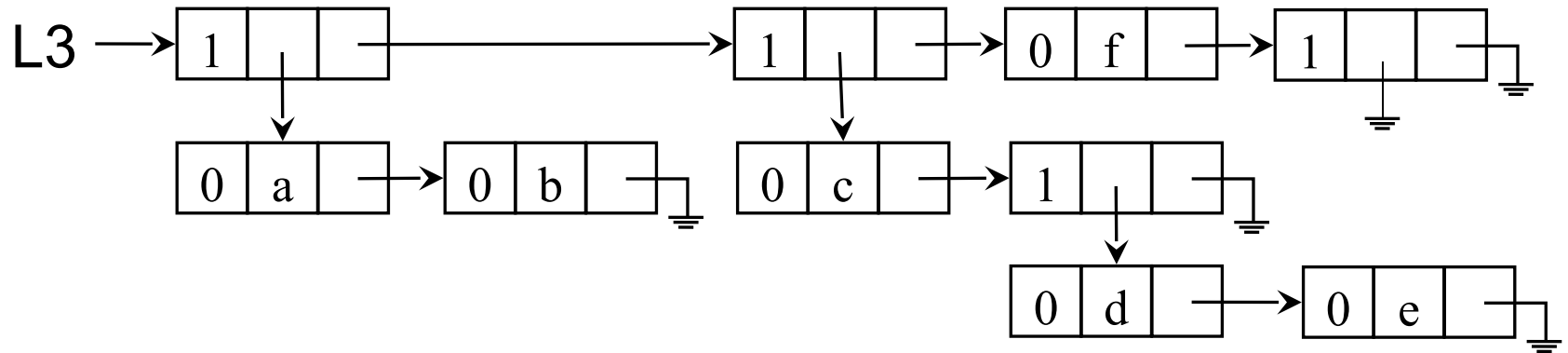
10

- Exercício: faça a representação da lista L3
- $((a,b),(c,(d,e)),f,())$

Lista generalizada

11

- Exercício: faça a representação da lista L3
((a,b),(c,(d,e)),f,())

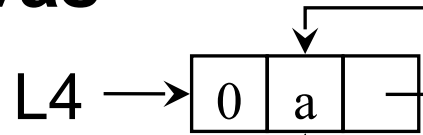


Lista generalizada

12

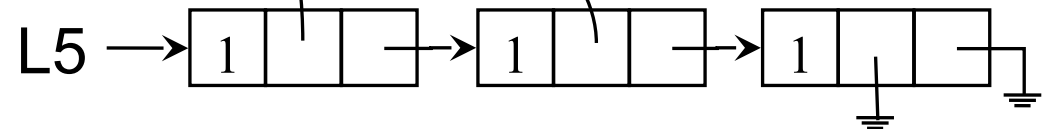
Listas Recursivas

$L4 = (a, L4)$



Listas Compartilhadas

$L5 = (L4, L4, ())$



Lista generalizada

- Declaração em C
 - ▣ Union

Lista generalizada

□ Declaração em C

▣ Union

```
typedef struct no_ NO;  
typedef struct lista_ LISTA;  
  
struct lista_{  
    NO *inicio;  
    int profundidade;  
};  
  
struct no_{  
    int tipo;  
    union {  
        ITEM *atomo;  
        struct no_ *sublista;  
    } info;  
  
    NO *prox;  
  
};
```

Lista generalizada

□ Exercício

- Implementar uma sub-rotina para imprimir uma lista generalizada

Lista generalizada

□ Exercício

- Implementar uma sub-rotina para verificar se um átomo x está em uma lista generalizada
 - Entrada: nó de início da lista e a chave de busca
 - Saída: o nó contendo a chave ou NULL
- Faça a árvore de ativação dessa sub-rotina para a busca pelo elemento 'd' na lista generalizada $(f,(a,b),(c,(d,e)),())$

Lista generalizada

□ Exercício

- ▣ Implementar uma sub-rotina para verificar se duas listas generalizadas são completamente iguais

Lista generalizada

□ Exercício

- ▣ Implementar uma sub-rotina para verificar se duas listas generalizadas são estruturalmente iguais
 - O conteúdo em si não importa

□ Exercício

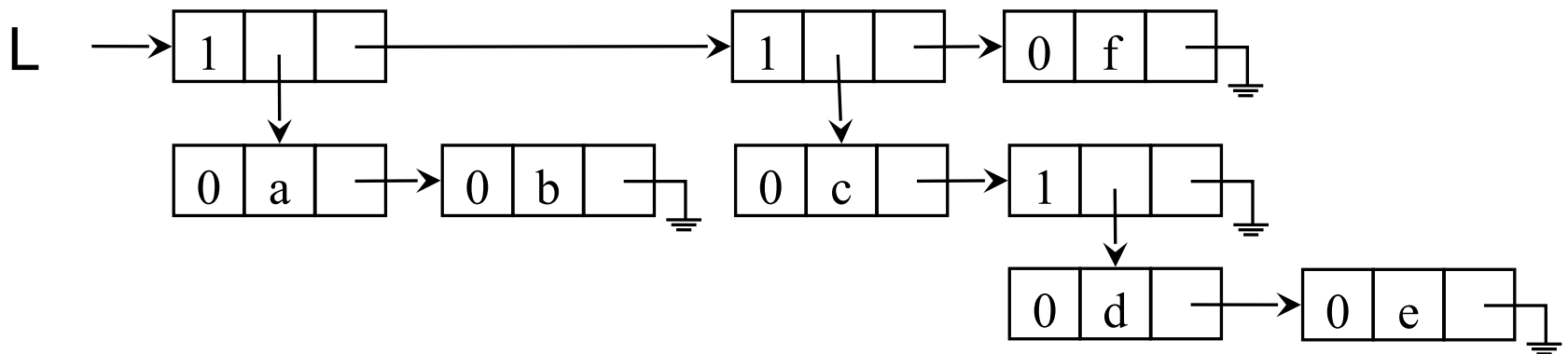
▣ Implementar uma sub-rotina que determina a profundidade máxima de uma lista generalizada

■ $A=(a,(b)) \rightarrow \text{prof}(A)=2$

■ $B=(a,b,c) \rightarrow \text{prof}(B)=1$

■ $C=() \rightarrow \text{prof}(C)=0$

■ Por exemplo, para o caso abaixo, a sub-rotina deveria retornar profundidade 3



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

■ $P1=P2=3, P3=2$

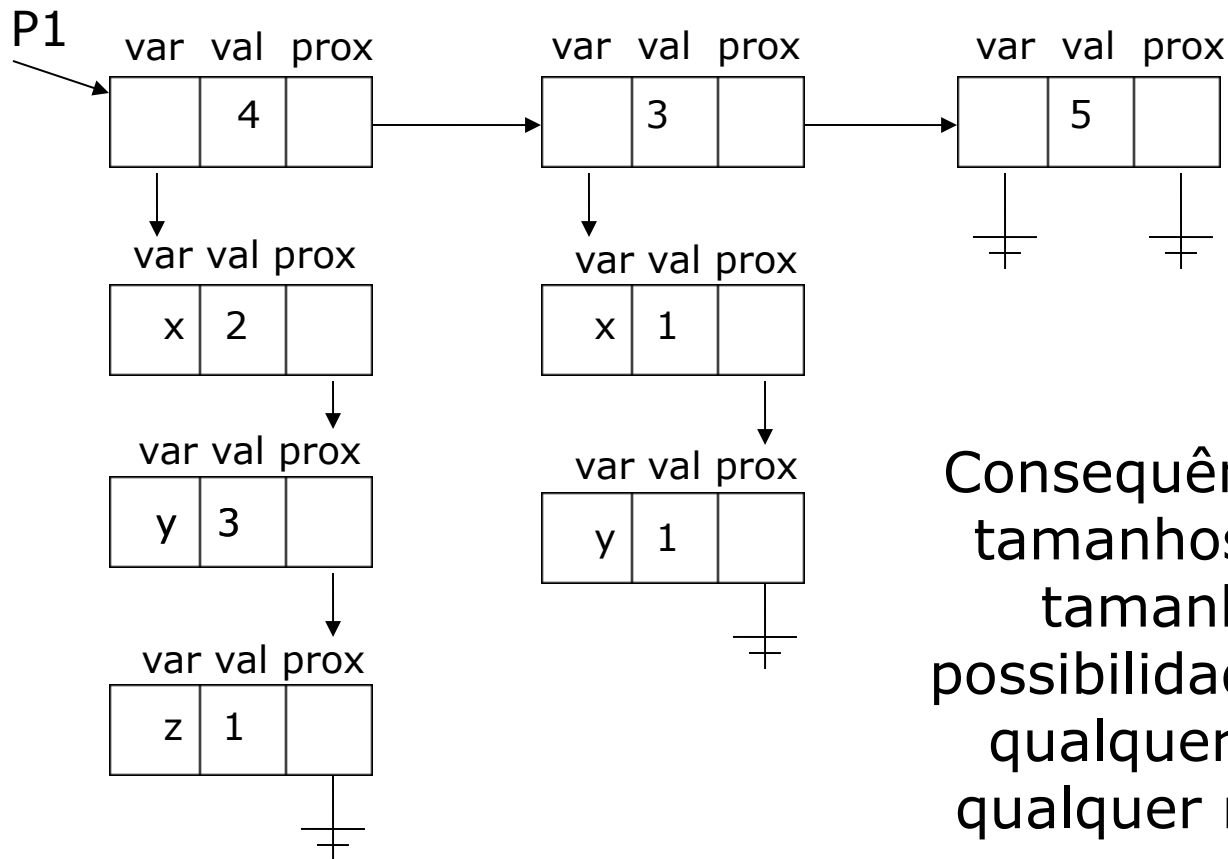
c) nem todo termo é expresso com todas as variáveis

Lista generalizada e polinômios

- Objetivos
 - representar de forma organizada e robusta
 - representação única para todo polinômio
- Solução: lista generalizada

Lista generalizada e polinômios

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




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

Exercício para casa

23

- Implementar uma sub-rotina 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 sub-rotina

- 
- Material baseado nos originais produzidos pelos professores:
 - ▣ Thiago A. S. Pardo
 - ▣ Maria das Graças V. Nunes
 - ▣ Maria Cristina F. de Oliveira