



SCC-223 Estruturas de Dados I

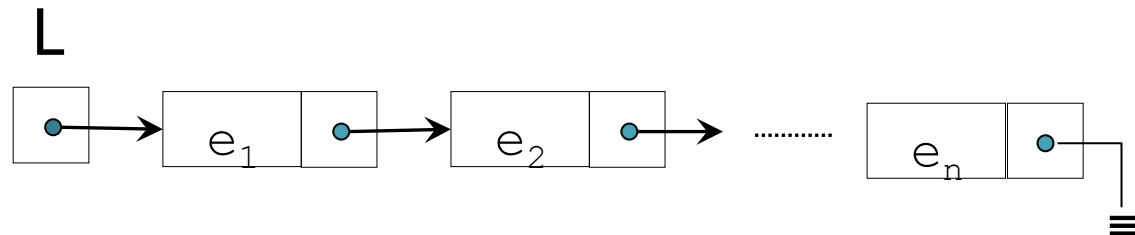
Lista Generalizada

Profa. Elaine Parros Machado de Sousa

Generalizando o conceito de Lista

- Uma lista $L = (e_1, e_2, e_3, \dots, e_n)$ pode ser definida como:
uma sequência formada por um elemento e_1 seguido de uma lista (e_2, e_3, \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 definição...



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**.

$$L = (a, \underbrace{(b, c)}_{e_2}, d, \underbrace{(e)}_{e_4}, \underbrace{(\quad)}_{e_5})$$

e_1 e_2 e_3 e_4 e_5

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

Lista generalizada

- **Ideia:**

- Uma **Lista generalizada** é uma lista que pode ter como elemento ou um **átomo** ou uma **outra lista** (sub-lista)

- **Definição**

- Uma **lista generalizada** L é uma sequência finita de $n \geq 0$ 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 **sub-listas** de L .

Lista generalizada

- Uma lista generalizada pode ser denotada por **elementos entre parênteses** (no estilo da linguagem de programação LISP) ou **entre colchetes** (no estilo de PROLOG)
 - (a,b,c,d) ou [a,b,c,d]
 - (a,(b,c),d) ou [a,[b,c],d]
 - (a,(b),(c,(d),()),()) ou [a,[b],[c,[d],[]]]

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:** primeiro elemento da lista
 - átomo ou lista
 - **Cauda:** a lista restante
 - **sempre** uma outra lista, mesmo que vazia

Lista generalizada

- Exemplo:

$L1 = (a, b, c)$

$L2 = (a, (b, c))$

✓ Tamanho de L1?

✓ Profundidade de L1?

✓ Cabeça(L1)?

✓ Cauda(L1)?

✓ Cabeça(Cauda(L1))?

✓ Cauda(Cauda(L1))?

✓ Tamanho de L2?

✓ Profundidade de L2?

✓ Cabeça(L2)?

✓ Cauda(L2)?

✓ Cabeça(Cauda(L2))?

✓ Cauda(Cauda(L2))?

Lista generalizada

- Exemplo:

$L1 = (a, b, c)$

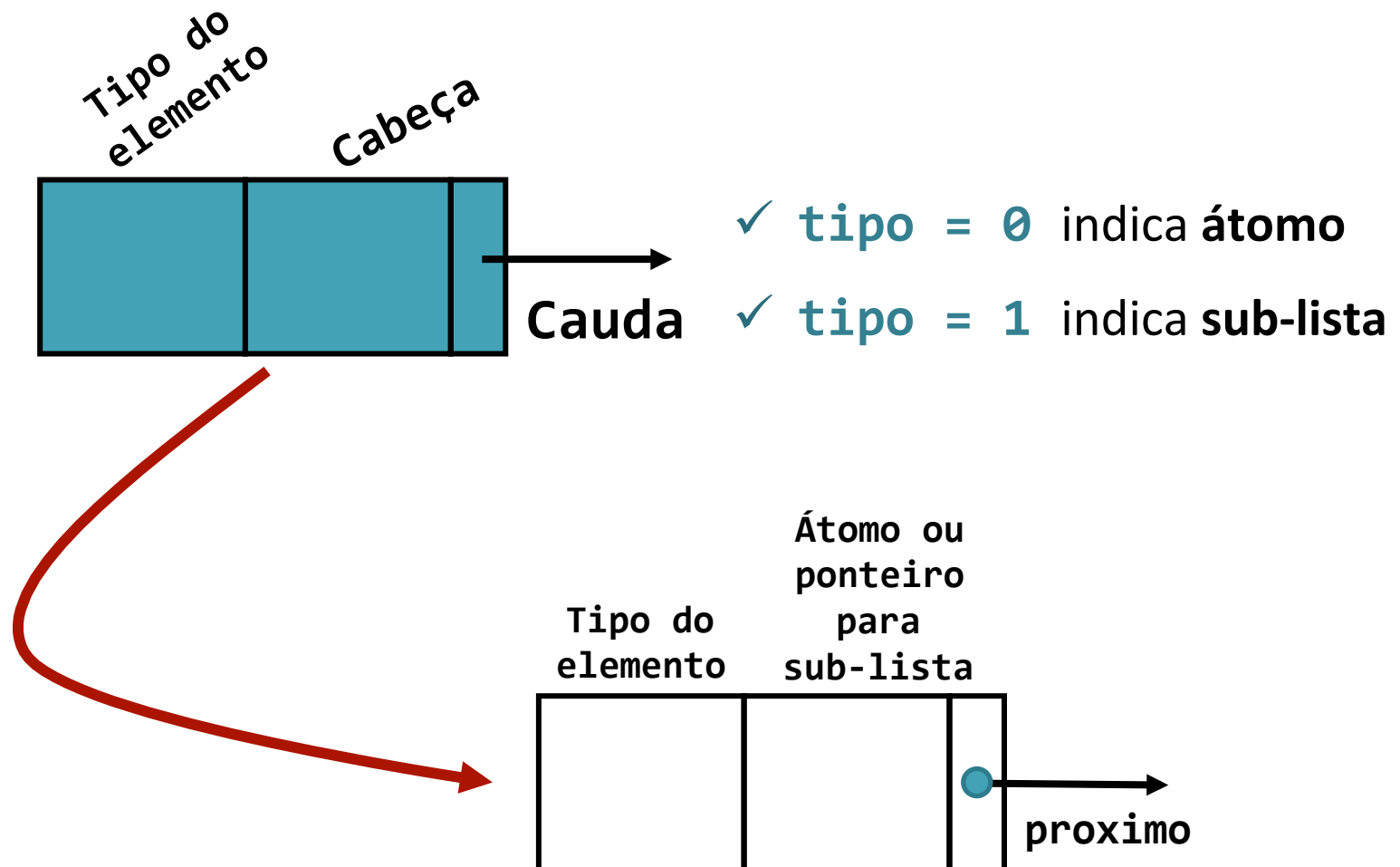
- ✓ Tamanho de L1 = 3
- ✓ Profundidade de L1 = 1
- ✓ Cabeça(L1) = a
- ✓ Cauda(L1) = (b, c)
- ✓ Cabeça(Cauda(L1)) = b
- ✓ Cauda(Cauda(L1)) = (c)

$L2 = (a, (b, c))$

- ✓ Tamanho de L2 = 2
- ✓ Profundidade de L2 = 2
- ✓ Cabeça(L2) = a
- ✓ Cauda(L2) = ((b, c))
- ✓ Cabeça(Cauda(L2)) = (b, c)
- ✓ Cauda(Cauda(L2)) = ()

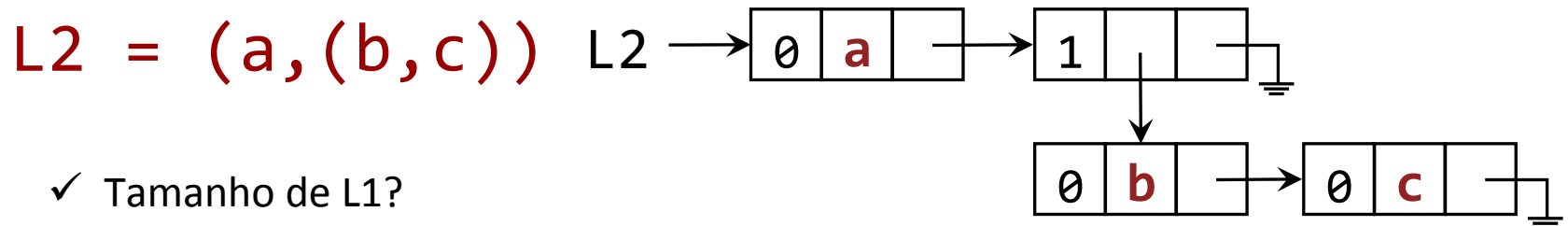
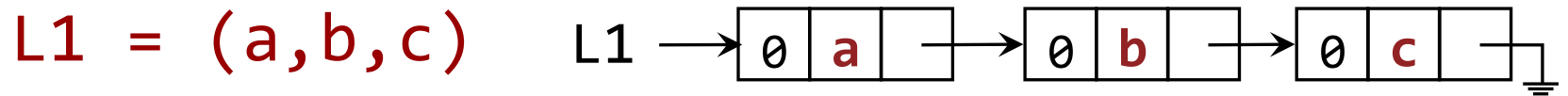
Lista generalizada

- Representação da Lista: **estrutura básica do nó**



Lista generalizada

- Exemplos

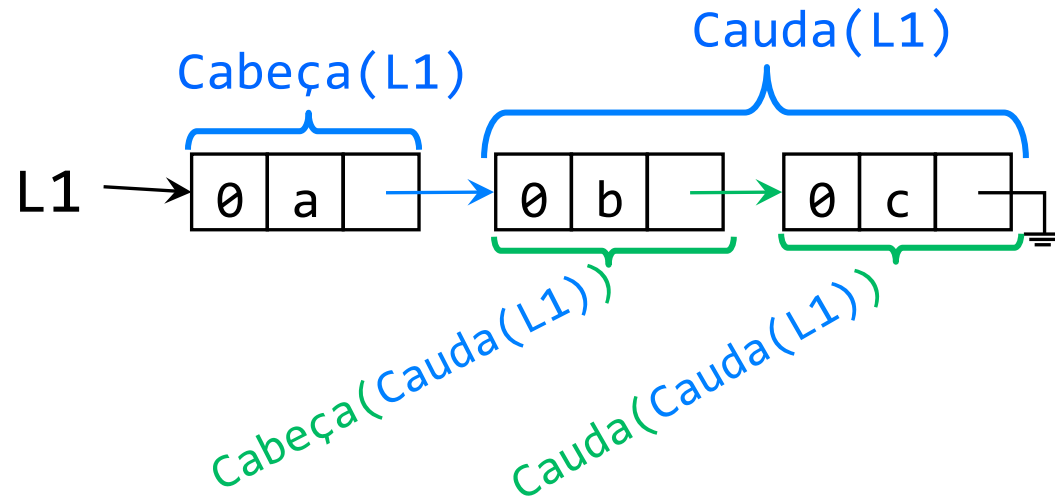


- ✓ Tamanho de L1?
- ✓ Profundidade de L1?
- ✓ Cabeça(L1)?
- ✓ Cauda(L1)?
- ✓ Cabeça(Cauda(L1))?
- ✓ Cauda(Cauda(L1))?

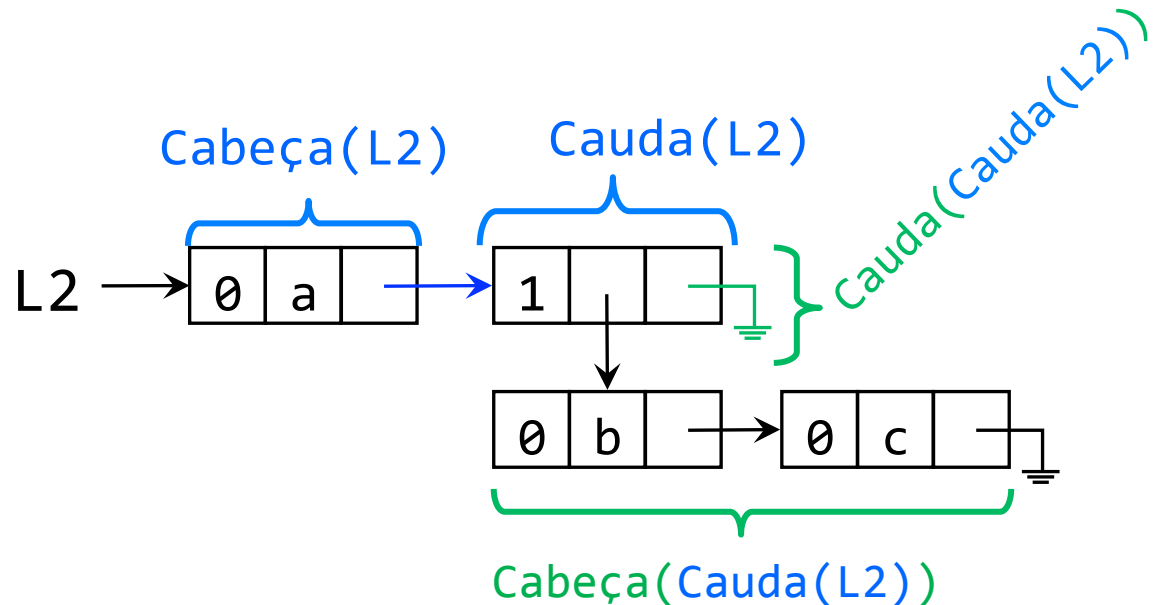
- ✓ Tamanho de L2?
- ✓ Profundidade de L2?
- ✓ Cabeça(L2)?
- ✓ Cauda(L2)?
- ✓ Cabeça(Cauda(L2))?
- ✓ Cauda(Cauda(L2))?

Lista generalizada

$L1 = (a, b, c)$



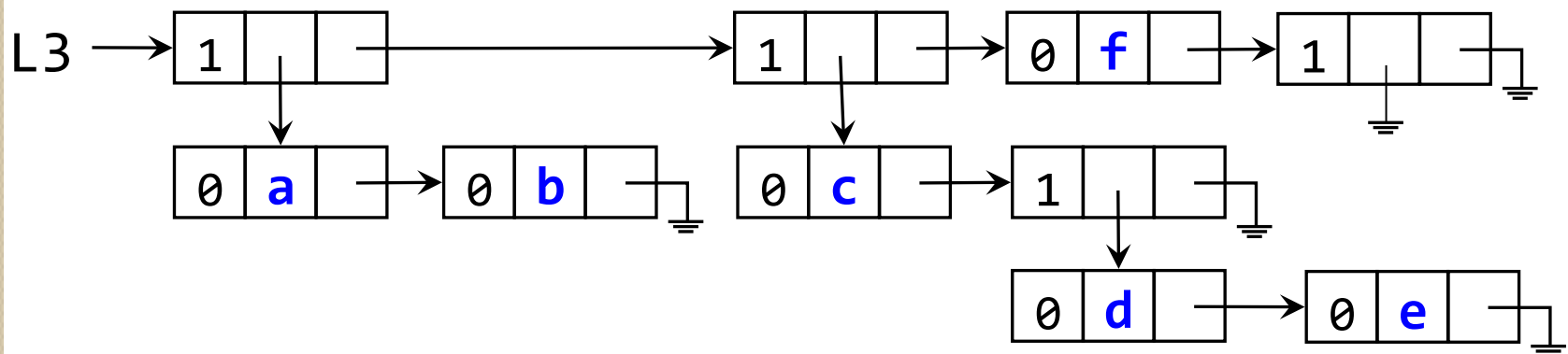
$L2 = (a, (b, c))$



- Ilustre a representação (usando nós) da lista **L3** e responda às questões seguintes.

L3 ((a,b),(c,(d,e)),f,())

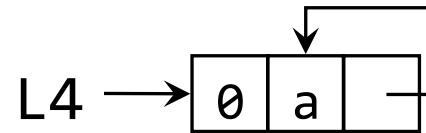
- ✓ Tamanho de L3?
- ✓ Profundidade de L3?
- ✓ Cabeça(L3)?
- ✓ Cauda(L3)?
- ✓ Cabeça(Cabeça(L3))?
- ✓ Cauda(Cabeça(L3))?
- ✓ Cabeça(Cauda(L3))?
- ✓ Cauda(Cauda(L3))?
- ✓ Cabeça(Cauda(Cauda(L3)))?
- ✓ Cauda(Cauda(Cauda(L3)))?
- ✓ Cabeça(Cabeça(Cauda(L3)))?
- ✓ Cauda(Cabeça(Cauda(L3)))?
- ✓



Lista generalizada

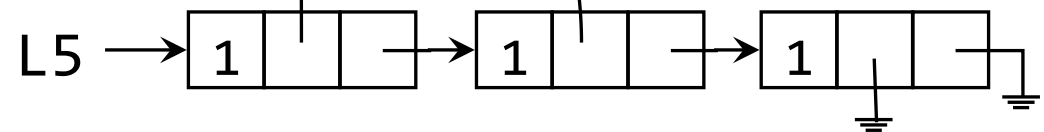
➤ Listas Recursivas

$L4 = (a, L4)$



➤ Listas Compartilhadas

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



Implementação

```
typedef struct no_ NO;
typedef struct lista_ LISTA;

struct lista_{
    NO *inicio;
    int profundidade; /*prof. da lista general.*/
};

struct no_{
    union { /*define campo genérico que armazena
            atomo OU sublista*/
        ITEM atomo;
        NO *sublista;
    } info;
    int tipo;
    NO *prox;
};
```

Exercício – não precisa entregar

1) Elaborar o algoritmo de uma função para verificar se um átomo **x** está em uma lista generalizada

- Versão 1: apenas na lista principal (primeiro nível da lista)
- Versão 2: em qualquer parte dela



Listas Generalizadas

- Exemplo de Aplicações?

Exemplo de Aplicação

- Considere os polinômios:

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

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

$$P3 = 3x^2y$$

- ✓ nro de termos: **variável**

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

- ✓ nro de variáveis: **variável**

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

- ✓ nem todo termo é expresso com todas as variáveis



Exemplo de Aplicação

- Objetivos
 - representar de forma organizada e robusta
 - registros de tamanho fixo
 - representação única para qualquer polinômio
- Solução: **lista generalizada**

Exercício para Entrega – 26/09

- **Exercício:** ilustre a representação dos 3 polinômios (exemplos anteriores) usando listas generalizadas.
- Resolver no papel, com caneta e letra legível 😊
- Colocar nome completo e NUSP
- Fotografar/escanear e submeter em **Atividade Exercício Polinômio até 26/09 23:55h.**