

## Listas Generalizadas

PEREIRA, Silvio do Lago, Estruturas de Dados Fundamentais, págs. 157 a 171

### Fundamentos

Uma **lista generalizada**  $L$ :  $[e_1, e_2, e_3, \dots, e_n]$ ,  $n \geq 0$ , é uma coleção onde cada elemento  $e_k$ ,  $1 \leq k \leq n$  é uma partícula indivisível (**átomo**) ou então é uma outra lista generalizada (**sublista**). O comprimento de uma lista generalizada  $L$  é definido como sendo o valor  $n$ . Se  $n=0$ , dizemos que a lista está vazia (**nula**), caso contrário,  $e_1$  é a **cabeça** da lista e  $[e_2, e_3, \dots, e_n]$  é a sua **cauda**.

### Exemplos de Listas Generalizadas

$[]$  é uma lista vazia cujo comprimento é 0;  
 $[a, b]$  tem comprimento 2 e contém os átomos **a** e **b**;  
 $[a, [b, c]]$  tem comprimento 2 e contém o átomo **a** e a lista **[b, c]**;  
 $[a, [b, [c]], d]$  tem comprimento 3 e contém os átomos **a**, **d**, e a lista **[b, [c]]**;  
 $[[[]]$  é uma lista de comprimento 1, contendo a lista nula  $[]$ .

As principais operações sobre listas generalizadas são: **Head()** e **Tail()**. Se **L** é uma lista não nula, então **Head(L)** é o valor do primeiro elemento de **L**, que pode tanto ser um átomo quanto uma lista; **Tail(L)** é a lista, que pode ser nula. Se **L** é uma lista vazia (ou um átomo), **Head(L)** e **Tail(L)** não têm valor definido.

Exemplo de funcionamento das operações **Head()** e **Tail()**

$L: [[Elis, [olhos, azuis]], [Yara, [idade, 23]]]$

$Head(L) \equiv [Elis, [olhos, azuis]]$   
 $Head(Head(L)) \equiv Elis$   
 $Tail(L) \equiv [[Yara, [idade, 23]]]$   
 $Head(Tail(L)) \equiv [Yara, [idade, 23]]$   
 $Tail(Head(Tail(L))) \equiv [[idade, 23]]$   
 $Head(Tail(Head(Tail(L)))) \equiv [idade, 23]$   
 $Tail(Head(Tail(Head(Tail(L)))) \equiv [23]$   
 $Head(Tail(Head(Tail(Head(Tail(L)))) \equiv 23$

Tanto as operações **Head()** e **Tail()** podem resultar numa lista generalizada, entretanto apenas **Head()** pode resultar num átomo.

Para construir uma lista generalizada:

Operação construtora básica **Cons()**.

A operação **Cons(H, T)** constrói uma lista cuja cabeça é **H** e cuja cauda é **T**.

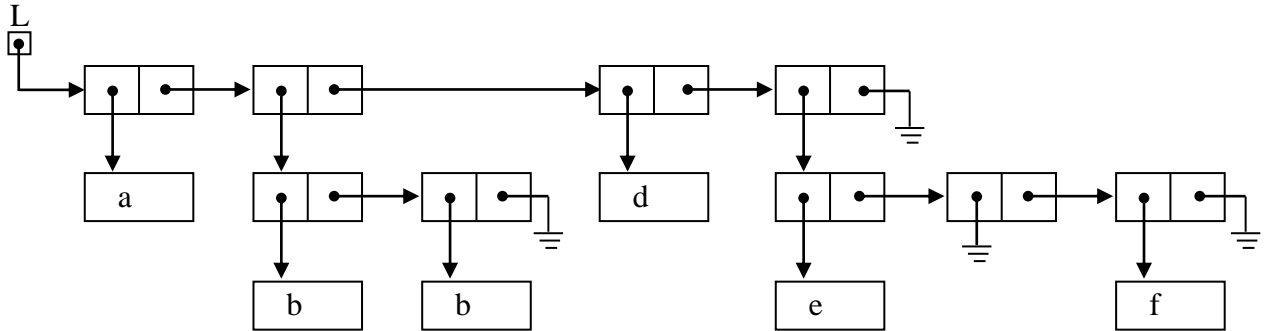
Exemplos:

$Cons(a, []) \equiv [a]$   
 $Cons([], []) \equiv [[]]$   
 $Cons(b, Cons(a, [])) \equiv [b, a]$   
 $Cons(Cons(a, []), Cons(b, [])) \equiv [[a], b]$   
 $Cons(Cons(a, Cons(b, [])), Cons(c, [])) \equiv [[a, b], c]$

## Representação de Listas Generalizadas

Dois tipos básicos de nodos:

- **Terminais:** nodos que armazenam os átomos contidos na lista generalizada;
- **Não-terminais:** nodos que descrevem a forma da lista, isto é, a estrutura de aninhamento e relacionamento dos átomos nela contidos.



Representação interna da lista generalizada L: [a, [b, c], d, [e, [], f]]

Nodos não-terminais são compostos por dois campos, um que armazena um ponteiro para baixo (**cabeça**) e outro que armazena um ponteiro para frente (**cauda**).

A estrutura da lista generalizada possui dois níveis básicos:

- **Superior:** uma lista encadeada, acessível através do ponteiro inicial L, que possui um nodo não-terminal para cada um dos elementos da lista;
- **Inferior:** acessível através dos “ponteiros para baixo” armazenados no nível superior, e pode conter uma mistura de nodos terminais e não-terminais. Se um nodo do nível superior representa um átomo, então seu ponteiro para baixo acessa um nodo terminal, caso contrário, seu ponteiro acessa uma lista generalizada (contendo os mesmos dois níveis básicos).