5.4 广义表的定义

顾名思义，广义表是线性表的推广，也称其为列表。

ADT Glist {

数据对象：$D = \{e_i \mid i=1,2,..,n；\ n \geq 0；$

$e_i \in AtomSet$ 或 $e_i \in Glist$，

AtomSet为某个数据对象 }

数据关系： $LR = \{<e_{i-1}, e_i> \mid e_{i-1}, e_i \in D, 2 \leq i \leq n\}$

基本操作:

} ADT Glist

广义表是**递归**定义的**线性结构**，广义表一般记作：

$LS = (\alpha_1, \alpha_2, \cdots, \alpha_n)$

其中：n是它的长度，α_i 或为原子 或为广义表。习惯上，用大写字母表示广义表的名称，用小写字母表示原子。 称α_1为LS的表头，其余元素组成的表$(\alpha_2, \cdots, \alpha_n)$是LS的表尾。

例如: A = ( )

F = (d, (e))

D = ((a,(b,c)), F)

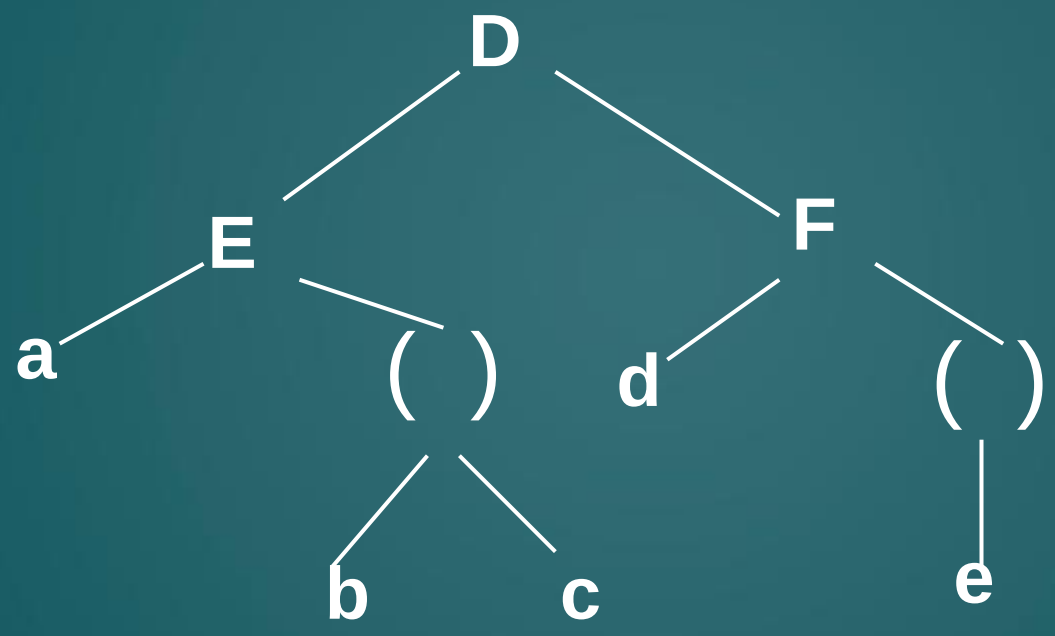
C = (A, D, F)

B = (a, B) = (a, (a, (a, … ,)))

广义表是一个**多层次**的**线性结构**

例如：

D=(E, F)



其中:

E=(a, (b, c))

F=(d, (e))

广义表 LS = ($\alpha_1, \alpha_2, \ldots, \alpha_n$)的结构特点:

1) 广义表中的数据元素有相对**次序**；

2) 广义表的**长度**定义为最外层括号包含元素个数；

3) 广义表的**深度**定义为所含括弧的重数；

注意：“原子”的深度为 **0**

“空表”的深度为 **1**

4) 广义表可以**共享**；

5) 广义表可以是一个**递归**的表。

递归表的深度是无穷值，长度是有限值。

6) 任何一个**非空广义表** LS = (α_1, α_2, ..., α_n)

均可分解为

表头 Head(LS) = α_1 和

表尾 Tail(LS) = (α_2, ..., α_n) 两部分。

例如: D = (E, F) = ((a, (b, c)) , F)

Head(**D**) = E Tail(**D**) = (F)

Head(**E**) = a Tail(**E**) = ((b, c))

Head(**((b, c))**) = (b, c) Tail(**((b, c))**) = ( )

Head(**(b, c)**) = b Tail(**(b, c)**) = (c)

Head(**(c)**) = c Tail(**(c)**) = ( )

基本操作：

- **结构的创建和销毁**

  InitGList(&L);　　DestroyGList(&L);

  CreateGList(&L, S);　CopyGList(&T, L);

- 状态函数

  GListLength(L);　GListDepth(L);

  GListEmpty(L);　GetHead(L);　　GetTail(L);

- **插入和删除操作**

  InsertFirst_GL(&L, e);

  DeleteFirst_GL(&L, &e);

- **遍历**

  Traverse_GL(L, Visit());