**5.4广义表（Lists）**

**5.4.1广义表的定义**

**1．广义表的概念**

**广义表是线性表的推广,它是由n个元素,,……,组成的有限序列。**

**其中: 可以是一个原子或是另一个广义表。**

**记作：LS=(,,…,)**

**其中：LS为表的名称（大写），可以是元素（称为原子），也可以是广义表（称为子表），n是表长； 当表非空时，称为表头（Head）其余元素（,…,)为表尾（Tail）。**

**为区分原子和广义表，通常用大写字母表示广义表，小写字母表示原子。**

**举例说明广义表：**

**（1）A=（）— 表A为空表，长度为0；**

**（2）B=（e）— 表B只有一个原子e，长度为1；**

**（3）C=（a,（b，c，d））— 表C的长度为2，两个元素分别是原子a和子表（b，c，d）；**

**（4）**D=（A，B，C）— **表D的长度为3，三个元素都是表。将三个子表的值带入后表D=**（（），（e），**（a,（b，c，d）））**

**（5）**E=（a，E）— E是一个递归表，它的长度为2。E相当于一个无穷的列表E=（a，（a，（a，……）））

**对广义表的说明：**

**（1）广义表是一个多层次的结构，表的元素可以是子表，子表的元素还可以是子表，……。表的深度就是指含括号的层数。广义表可以用图形（树）表示；**

**其中：圆圈表示广义表，方块表示原子。**

**广义表A,B,C,D的树形表示：**

e

d

c

b

a

d

e

c

b

a

**（2）广义表可以为其它广义表所共享。如表D共享子表A，B和C。**

**（3）广义表可以是一个递归表。如表E。**

**2. 广义表的取表头和表尾运算**

**根据表头和表尾的定义可知，对于一个非空广义表，表头可能是一个原子，也可能是一个子表，而表尾一定是子表。**

**如：Head(B)=e,Tail(B)=()，**

**Head(D)=A,Tail(D)=(B,C)**

**Head((B,C))=B,Tail((B,C))=(C)**

**注意：若有广义表A=()和N=(())，二者是不同的。**

**表A是空表（没有元素），长度为0，而表N是非空表（有一个元素“（）”），长度为1。**

5.4.2 广义表的存储

**由于广义表的数据元素既可以是原子也可以是子表，因此很难用顺序结构表示，常用链式结构表示。**

**结点的链式结构：**

tag atom/hp tp

**广义表的结点分为两种：原子结点和表结点，使用标志域tag加以区别。**

**●原子结点的表示：tag=0，atom为结点的数据;**

tag=0 atom

**●表结点的表示：tag=1，hp为指向子表的指针（向下），tp是指向同一层下一个结点的指针（向右）；**

tag=1 hp tp

**对于域atom/hp,tp互相排斥，所以用C语言描述时使用“共用体union”。**

**定义：**

**typedef enum{ATOM,LIST}ElemTag;**

**typedef struct GLNode**

**{**

**ElemTag tag;**

**union//共用体的使用**

**{**

**AtomType atom;**

**struct**

**{**

**struct GLNode \*hp,\*tp;**

**}ptr;**

**};**

**}\*GList;**

**例：广义表的链式存储**

**A NULL**

**B C C**

1

0 a

1

1

1

1

1 ^

0 e

0 c

0 b

0 d

**D**

1 ^

1

1

**E**

1

1

0 a