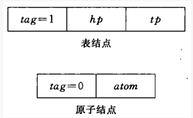
# 广义表

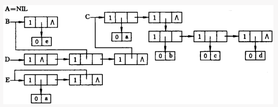
## 5.4 定义

- 即线性表的推广，也有人称之为列表
- 一般记作： $LS = (\alpha_1, \alpha_2, \ldots, \alpha_n)$
  - $\alpha_i$ 可以是单个元素，也可是广义表，分别称为LS的 原子 和 子表
  - 表头 —— $\alpha_1$
  - 表尾 —— $(\alpha_2, \alpha_3, \ldots, \alpha_n)$
  - 习惯
    - 大写字母表示 广义表
    - 小写字母表示原子
- 重要结论：
  - ① 列表的元素可以是子表，而子表的元素还可以是子表
  - ② 列表可以为其它列表所共享
  - ③ 列表可以是一个递归的表，即列表本身也可以是其本身的一个子表
- 注意
  - () 是空表
  - (()) 不是空表，可以分解为表头是空表，表尾也是空表

## 5.5 存储结构

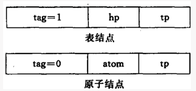
- 难以采用顺序存储，通常采用链式存储结构
  - 结构一
    - 表结点
      - 标志域 —— 区别是表结点还是原子结点
      - 指向表头的指针域
      - 指向表尾的指针域
    - 原子结点
      - 标准域
      - 值域
  - 结构二
    - 原子结点
      - 标准域
      - 值域
      - 指向表尾的指针域
    - 表结点
      - 标志域
      - 指向表头的指针域
      - 指向表尾的指针域

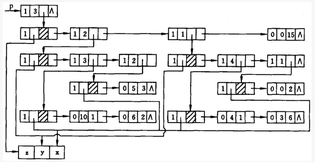



## 5.6 m元多项式的表示

- 先分解出一个主变元，随后再分解出第二个变元等等。由此，一个 m元的多项式首先是它的主变元的多项式,而其系数又是第二变元的多项式，由此可用广义表来表示 m 元多项式。
- 在每一层上增设一个表头结点并利用exp 指示该层的变元
- 头指针 p所指表结点中 exp 的值 3 为多项式中变元的个数




## 5.7 广义表的递归算法

- **算法**
  - ① 求广义表的深度
    int GListDepth(GList L)
  - ② 复制广义表
    Status CopyGList(GList &T, GList L)
  - ③ 建立广义表的存储结构
    - 一种是把广义表分解成表头和表尾两部分;
    - 另一种是把广义表看成是含有n个并列子表（设原子也视作子表）的表。
      - 用结构一
      Status CreateGList(GList &L, SString S)