数据结构作业(第五次)

PB20111686 黄瑞轩

基于扩展线性链表存储结构重写求广义表的深度算法

广义表的拓展线性链表储存表示如下：

```
typedef enum { ATOM, LIST }ElemTag; // ATOM == 0, 原子；LIST == 1，子表
typedef struct GLNode {
    // 公共部分，用于区分原子节点和表节点
    ElemTag tag;
    // 原子节点和表节点的联合部分
    union {
        AtomType atom;
        struct GLNode* hp;
    };
    // 相当于线性链表的next，指向下一个元素节点
    struct GLNode* tp;
}*GList;
```

此表示下的求深度函数：

```
int GListDepth(GList L) {
    if (!L)return 1;
    if (L->tag == ATOM)return 0;
    int max = 0;
    GList pp = L;
    for (max = 0; pp; pp = pp->tp) {
        int dep = GListDepth(pp->tp);
        if (dep > max)max = dep;
    }
    return max + 1;
}
```

5.32

沿用上题的广义表定义。

```
bool IsEqual(GList a, GList b) {
    if (a & b == NULL) {
        if (a | b == NULL) {
            return true;
        }
        else {
            return false;
        }
    }
    else {
        if (a->tag ^ b->tag == 1) {
            return false;
        }
        else {
            if (a->tag == ATOM) {
                if (a->atom == b->atom) {
                    return IsEqual(a->tp, b->tp);
                }
                else {
                    return false;
                }
            }
            else {
                if (IsEqual(a->hp, b->hp)) {
                    return IsEqual(a->tp, b->tp);
                }
                else {
                    return false;
                }
            }
        }
    }
}
```