**鲁滨逊归结原理**

**工学 21217030 孟庆喜**

**鲁滨逊归结原理介绍**

归结原理也称消解原理，是鲁宾逊提出的一种证明子句不可满足性，从而实现定理证明的一种理论及方法。由谓词公式转化为子句集的过程可以看出，在子句集中子句之间的关系是合取关系，其中只要有一个子句不可满足，则子句集就不可满足。空子句是不可满足的，即一个子句集中若含空子句，则它是不可满足的。归结原理的基本思想就是检查子句集中是否含空子句，若含，则子句集S不可满足，或说证明一个谓词公式是永假的过程，就是归结由此公式转换成的子句集包含空子句的过程。由命题逻辑中的归结原理可以得出如下的推论：

设c1与c2是子句集S中的两个子句，c12是它们的归结式,若用c12代替c1和c2后得到新子句集S1，则由S1的不可满足性可以推出S的不可满足性。这个定理说明，证明子句集S的不可 满足性问题可以转化成证子句集S1的不可满足性问题，…直到从子句集Sn 中推出一个空子句来。归结的一般过程是水平浸透法，即对子句集中遍历每个子句，对每个子句遍历每个谓词公式，进行归结。在第n次归结时，生成n+1级子句集，利用1…n的子句集对n+1级子句集进行归结，直到新生成的子句集为NIL（归结成功）或不能继续归结（归结失败）。

**算法流程**

1. 开始，创建子句集，并从字符串中创建子句，将子句加入子句集S。
2. 判断子句集S是否含有空子句，是，终止，否，转3.
3. 在子句集中寻找含有互补谓词的子句C1和C2。
4. 计算互补谓词的合一替换集合M。并判断是否可以合一，否转3，是转5.
5. 将C1和C2中生成的新子句C加进子句集。
6. C是空子句，则终止，否，转3.

**数据结构设计**

(1)谓词对象的表示

谓词在计算机内部表示为一个类，该类有两个类型，一个是int型num表示谓词符号的标识，谓词符号的否定形式用负数标识，但绝对值相同。另一个是string类型，表示该谓词的参数表。

class Predication

{

public:

int num;

string ArgList;

};

(2)子句对象的表示

字句对象是由多个位子对象连在一起构成的一个链表。为了查找方便，将此链表设计成为一个双向链表。

Class Clause

{

Public：

int nItems //标识子句中谓词的数目。

list<Predication\*> PredList //该链表存储子句中的谓词信息，其元素是Predication指针类型。

prey, next: Clause; //指针类型成员，这两个元素是为了构造子句集的双向链表设置的，分别指向前驱和后继的子句。

}

（3）子句集对象的表示：

字句集对象是由多个字句对象构成的，这里用一个存储字句对象的数组来表示字句集对象。

Class ClauseSet

{

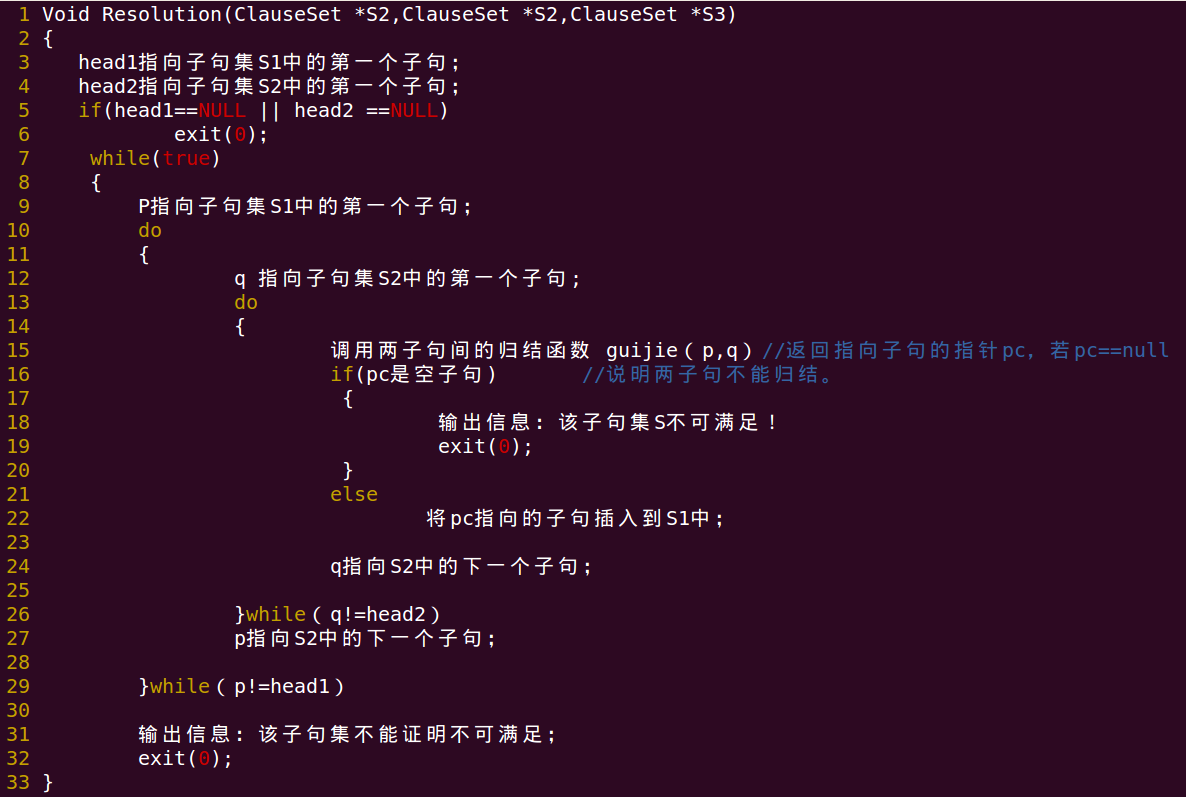
Public：

parrayClasuse[子句数]; //Clause 类型的指针数组

}

**函数实现伪代码**

整体子句集归结函数的伪代码如下图所示，该函数有三个参数，S1是前提条件子句集指针，S2是结论子句集指针，S3是容纳新子句的子句集指针。



具体的归结函数，该函数，从该函数从p和q指向的子句中寻找互补谓词对，并调用具体的归结实现函数生成新子句，并返回新子句。如果子句p和q不能归结则返回NULL。

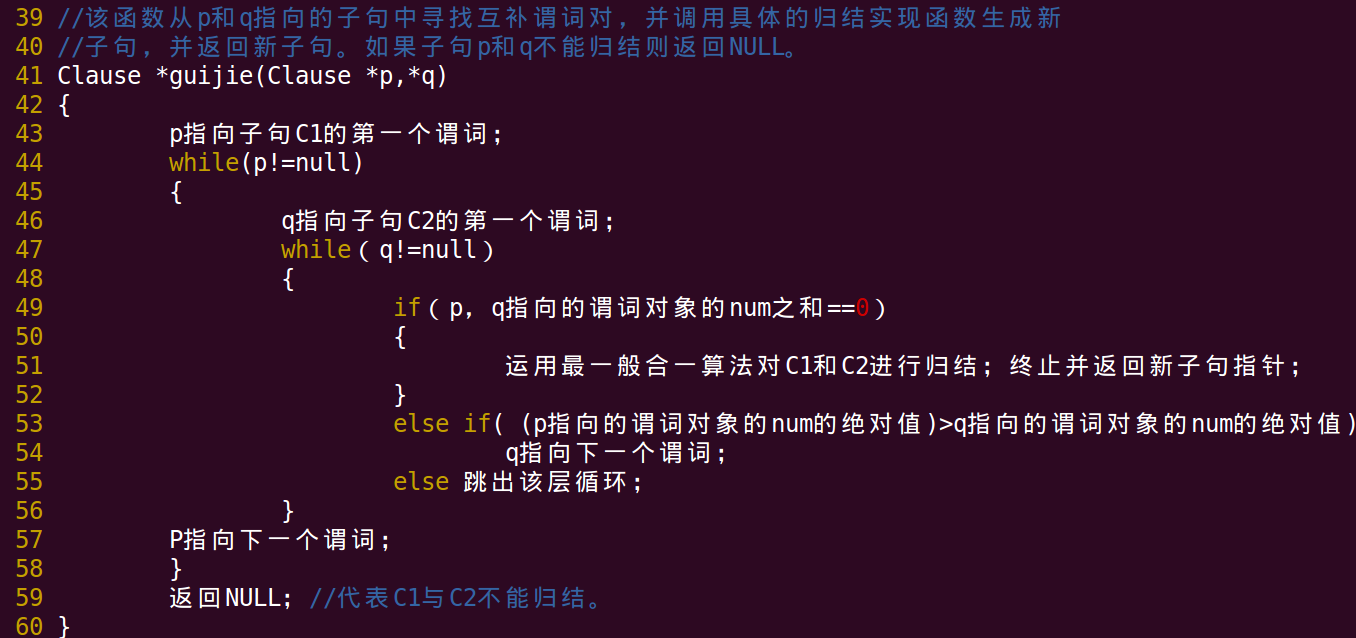
****

图2 程序伪代码

**算法结果**

针对归结失败和归结成功的两种情况，分别如下图所示：

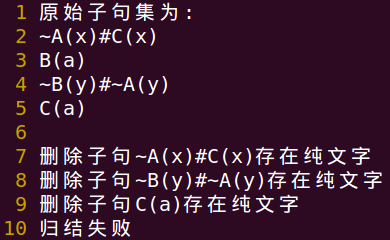


图3 归结失败

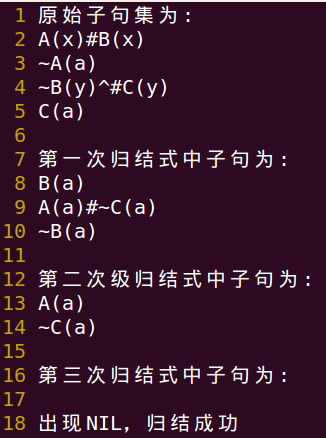


图4 归结成功

**结论**

本文档是鲁滨逊归结原理的实现的文档。文档对鲁滨逊归结原理进行了简单的介绍，并对其算法实现，给出了算法的执行流程、重要数据结构设计以及重要函数的伪代码。最后，给出了两个测试用例，在不同的测试用例上，使用鲁滨逊归结算法进行归结，得到的结果。