《程序设计 II》课程实践环节指导

实验一

实验题目：文件读写及命题公式合法性判断

在正式开始你的实验之前，请务必花一些时间仔细阅读完本文档关于实验的描述与规定！

实验提交时间：**2012-05-27 22:00，提交到 FTP 服务器**。

一、实验目的与要求

（1） 要求完全采用 C 语言完成程序设计，不得使用 C++的 STL；
（2） 熟悉结构体及其指针的使用，并学习构建与遍历二叉树；
（3） 熟悉文件读写的基本方法，加深对字符串处理方法的理解；
（4） 加深学生对于命题逻辑公式定义的理解，会写出判断其合法性的程序；

二、实验描述

（1） 从文件“in*.txt”中读取命题公式，并从语法上判断公式是否正确。
（2） 将读取的公式转换为二叉树结构存储，所需的结构体等自定义。
（3） 如果命题公式正确，则遍历二叉树，根据先序遍历输出到文件：“学号_out*_1.txt”。
（4） 如果命题公式正确，则遍历二叉树，将联结词加上括号后全部输出到文件“学号_out*_2.txt”。
（5） 将命题公式转换为 CNF 范式并输出到文件中。
（6） 判断这个 CNF 范式是否为真，可以暴力枚举来求解，也可使用现成的算法。

基本要求：

每位同学必须完成以上(1)-(4)。

进阶要求：

学有余力同学继续完成(5)-(6)。

文件“in*.txt”（*表示测例的序号，我正在写随机生成命题公式的程序，来测试各位同学所完成程序的运行时间消耗，此时间消耗作为评分依据）中包含一个命题公式，这个命题公式单独占一行，所有原子命题均用**大写字母**表示。

为方便程序设计，我们对数理逻辑中的符号做一定改变。

Negation:¬符号用 ~ 符号表示；
Conjunction：∧符号用 & 符号表示；
Disjunction:∨符号用 | 符号表示；
Imply: → 符号用 > 符号表示。

以上四个联结词按以下优先级递增：

>，&，|，~，()

输出信息均输出到文件“学号_out*_1.txt”、“学号_out*_2.txt”中。其中输出的“学号_out*_2.txt”可以再作为输入测例。

读取一个命题公式后从语法上判断公式是否正确，如果正确则输出 FORMULA TRUE，否则输出 FORMULA FALSE，单独占一行。

命题公式正确的话则将其转换为 CNF 范式，并且判断 CNF 是否为真，为真则输出 CNF TRUE，否则输出 CNF FALSE，单独占一行。

为了便于测试各位同学编写的程序，统一规定各位同学提交的可执行程序命名为：“学号.exe”

读写的文件都与“学号.exe”在同一目录。

注意程序风格和代码注释。

三、实验评分

实验各部分评分细则如下：

表一

项目	权重(%)
1、文件读写	15
2、二叉树先序遍历	25
3、命题公式加上括号后输出	35
4、程序运行效率（统一在一台机器上测试，并公布）	15
5、代码风格	10
6、转换为 CNF	加分
7、判断 CNF 是否为真	加分

四、参考资料

大家可以上网查找相关资料，以下资料来自于网络。

1、命题公式（合式公式）：

命题公式有限次按以下规则生成：

（1）单个命题常项或变项 P、Q、R、0、1 是命题公式；

（2）若 A 是命题公式，则¬A 也是命题公式；

（3）若 A, B 是命题公式，则 A∧B、A∨B、A→B 也是命题公式。

2. 合取范式（CNF）

合取范式形如：

$A_1 \wedge A_2 \wedge \cdots\cdots \wedge A_n$

其中 $A_i(i=1,\cdots\cdots, n)$为析取式。

范式定理：所有命题公式都可以转换成 CNF 的等价公式。

这种变换基于了关于逻辑等价的规则: 双重否定律、德·摩根定律和分配律。

求 CNF 范式步骤：

1.利用 $A \rightarrow B = \neg A \vee B$ 消去联结词→；

2.重复使用双重否定律、德·摩根定律，将否定词内移到直接作用于命题变项：

$\neg(A \wedge B) = \neg A \vee \neg B$

$\neg(A \vee B) = \neg A \wedge \neg B$

$\neg\neg A = A$

3.重复使用分配律，可利用以下生成合取范式：

$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$

3. 二叉树

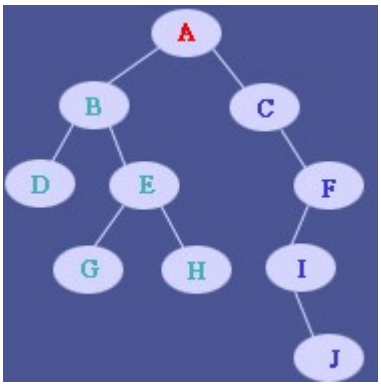
二叉树是每个节点最多有两个子树的树结构。通常子树被称作“左子树”（left subtree）和“右子树”（right subtree）。二叉树的每个结点至多只有二棵子树(不存在度大于 2 的结点)，二叉树的子树有左右之分，次序不能颠倒。

二叉树存储表示

二叉树通常用树结点结构来存储，结构体中保存当前节点信息及两个子节点地址，如果某个子节点为空，则指针一般赋值为 NULL。

4. 二叉树遍历

分为前序、中序、后序遍历。以中序遍历为例，利用递归的思想。如果要遍历一颗二叉树，如果只有一个节点，则只需要遍历这个节点；否则，先中序遍历根节点的左子树，再遍历根节点，最后中序遍历右子树。



以图中二叉树为例，中序遍历这棵二叉树的结果是 DBGEHACIJF

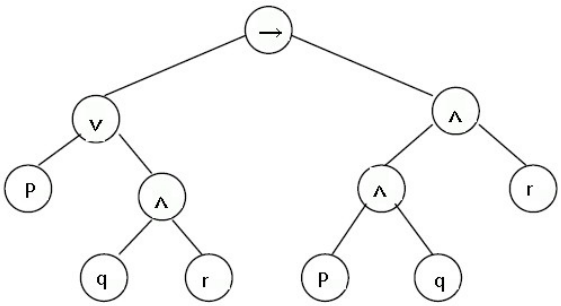
5. 综合举例：

假如给定的命题公式是：

$(p \vee (q \wedge r)) \rightarrow (p \wedge q \wedge r)$

通过判断，这个命题公式在语法上是正确的，因此输出 FORMULA TRUE。

接着，我们需要把命题公式用二叉树存储表示，存储结果如下



二叉树叶子节点保存元命题变量，其它节点保存关系符号。

注意，根据我们对符号的约束，文件里给定的命题公式表示成(p|(q&r))>(p&q&r)。

同理，CNF 合取范式也可以用二叉树存储表示。

命题公式(p∨(q∧r))→(p∧q∧r)可以转化为 CNF 合取范式(p∨¬q∨¬r)∧(¬p∨q∨r)∧(¬p∨q∨¬r)∧(¬p∨¬q∨r)

转化完成后，可以判断合取范式是否为真（即找 model），则是一个 SAT 问题（这是计算机领域的经典问题，曾在 1982 年由斯蒂芬.库克获得图灵奖），可使用暴力枚举的方法来解决，也可以使用现有的算法（具体算法自行搜索），或用 SAT 现成的求解器（miniSAT http://minisat.se/ 或 Zchaff http://www.princeton.edu/~chaff/zchaff.html ）求解。

Last modified: Mar.21,2012
万海：whwanhai@163.com