

**实 验 报 告**

**（2016 / 2017 学年 第 一 学期）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | 离散数学 | | | | | |
| 实验名称 | 利用真值表求主析取范式和主合取范式 | | | | | |
| 实验时间 |  | 年 |  | 月 |  | 日 |
| 指导单位 | 计算机科学与技术系 | | | | | |
| 指导教师 |  | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 学生姓名 |  | 班级学号 |  |
| 学院(系) |  | 专 业 |  |

**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验名称** | 利用真值表求主析取范式和主合取范式 | | | **指导教师** |  |
| **实验类型** |  | **实验学时** | **4** | **实验时间** |  |
| 1. **实验目的和要求**   **目的：**  **编程实现用真值表法求任意变量的合式公式的主析取范式和主合取范式**  **要求：**  **1. 从屏幕输入变元个数和每种情况的真值 2. 规范列出合式公式的真值表 3. 给出相应主析取和主合取范式** | | | | | |
| 二、**实验环境(实验设备)**  **硬件：**  **CPU：Intel X86-64 CPU     内存：16.00GB**  **软件：**  **运行环境：x86\_64-apple-darwin16.1.0  编译器：Clang 8.0.0** | | | | | |
| **三、实验原理及内容**  从键盘读入变元的个数以及期望的运算结果；并根据变元的个数生成动态数组进行遍历，由遍历结果导出主合取范式与主析取范式。  程序使用了面向对象，为便于阅读，类声明、实现以及主函数流程分为三个文件，编译时由makefile进行连接。  实验代码如下：  **文件1：Truthtable.h**  #include <iostream>  #include <string>  #include <cmath>  #include <vector>  class Truthtable  {  private:  int variables,length;  std::string putin,expression,value;  public:  std::vector<std::vector<bool>> var;  Truthtable();  ~Truthtable();  void get\_variable\_number();  void get\_truth();  void exp\_format();  void ergodic();  void print\_table();  void primary\_disjunctive();  void principal\_conjunctive();  };  **文件2：Truthtable.cpp**  #include "Truthtable.h"  using namespace std;  Truthtable::Truthtable(){  }  Truthtable::~Truthtable(){  var.erase(var.begin(),var.end()); // 清空var，防止vector造成的内存空洞  }  void Truthtable::get\_variable\_number(){  cout<<"Please put in the number of the variable in your expression: ";  (cin>>variables).get();  length = int(pow(2,variables));  }  void Truthtable::get\_truth(){  cout<<"Then type your conclusion of the tabulation(Just \'T\' and \'F\'' ! Press Enter key to finish):";  getline(cin,putin);  }  void Truthtable::exp\_format(){ //进行输入筛选，去除非法字符，若给定答案少于应有个数则用F填充  char str[putin.size()+1];  char \* p = (char \*)putin.c\_str();  strcpy(str,p);  char q[length+1];  for (int i = 0; i < length; ++i){  q[i]='F';  }  q[length]='\0';  p = str;  for (int i=0;\*(p+i)!='\0';++i){  if(!((\*(p+i)=='T'||\*(p+i)=='F'))){  for(int j=i;\*(p+j)!='\0';++j){  \*(p+j)=\*(p+j+1);  }  --i;  }  }  for (int i=0;\*(p+i)!='\0';++i){  q[i]=p[i];  }  value = q;  }  void Truthtable::ergodic(){  int binary = length - 1; //用于遍历赋值  int e = binary; //遍历判断变量  var.resize(length);  for (int i = 0; i < length; ++i){var[i].resize(variables);} //展开动态数组  for (int i = 0; i < length; ++i){  for (int j = variables - 1; j >= 0; --j){ //位操作所以从最低位开始！  if (0x01&e){var[i][j] = true;}else{var[i][j] = false;} //使用掩码取内存最低位  e >>= 1;  }  e = --binary; //继续下一个条件，重置e  }  }  void Truthtable::print\_table(){  cout<<"Truthtable export:"<<endl<<"┌";  for (int i = 0; i < variables\*3; ++i){  cout<<"─";  }  cout<<"┬───┐"<<endl<<"│";  for (int i = 0; i < variables; ++i){  cout<<" "<<char('P'+i)<<" ";  }  cout<<"│Ans│"<<endl<<"├";  for (int i = 0; i < variables\*3; ++i){  cout<<"─";  }  cout<<"┼───┤"<<endl;  for (int i = 0; i < length; ++i){  cout<<"│";  for (int j = 0; j < variables; ++j){  cout<<" ";  if (var[i][j]){cout<<"T ";}else{cout<<"F ";}  }  cout<<"│ "<<value[i]<<" │"<<endl;  }  cout<<"└";  for (int i = 0; i < variables\*3; ++i){  cout<<"─";  }  cout<<"┴───┘"<<endl;  }  void Truthtable::primary\_disjunctive(){  cout<<"primary disjunctive normal form export:"<<endl<<"T=";  for (int i = 0; i < length; ++i){  char alpha = 'P';  if (value[i]=='T'){  cout << "(";  for(int j = 0; j < variables; j++){  if(var[i][j]){cout<<char(alpha++);}else{cout <<"┐"<<(char)(alpha++);}  cout << "∧";  }  cout<<'\b'<<")∨";  }  }  cout<<"\b "<<endl;  }  void Truthtable::principal\_conjunctive(){  cout<<"principal conjunctive normal form export:"<<endl<<"T=";  for (int i = 0; i < length; ++i){  char alpha = 'P';  if (value[i]=='F'){  cout << "(";  for(int j = 0; j < variables; j++){  if(!(var[i][j])) {cout<<char(alpha++);}else{cout <<"┐"<<(char)(alpha++);}  cout << "∨";  }  cout<<'\b'<<")∧";  }  }  cout<<"\b "<<endl;  }  **文件3：main.cpp**  #include "Truthtable.h"  using namespace std;  int main(int argc, char const \*argv[])  {    Truthtable truth;  truth.get\_variable\_number();  truth.get\_truth();  truth.exp\_format();  truth.ergodic();  truth.print\_table();  truth.primary\_disjunctive();  truth.principal\_conjunctive();  return 0;  }  **文件4：main.cpp**  install: Truthtable  Truthtable: Truthtable.o main.o  g++ -o Truthtable Truthtable.o main.o -std=c++11  main.o:  g++ -c -o main.o main.cpp -std=c++11  Truthtable.o: Truthtable.h Truthtable.cpp  g++ -c -o Truthtable.o Truthtable.cpp -std=c++11  clean:  rm Truthtable.o main.o Truthtable | | | | | |

**实 验 报 告**

|  |
| --- |
| 运行结果：  当输入的变量个数为2时：  QQ20161031-3@2x.png  当输入的变量个数为3时：  QQ20161031-0@2x.png  当输入用例给定的运算结果不足时：  QQ20161031-1@2x.png  当输入包含非法字符时：  QQ20161031-2@2x.png |

**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **四、实验小结**（包括问题和解决方法、心得体会、意见与建议等） | | | | | |
| **五、指导教师评语** | | | | | |
| **成 绩** |  | **批阅人** |  | **日 期** |  |