

**实 验 报 告**

**（2019 / 2020 学年 第 一 学期）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | 离散数学 | | | | | |
| 实验名称 | 利用真值表法求取主析取范式以及主合取范式的实现 | | | | | |
| 实验时间 | 2019 | 年 | 9 | 月 | 16 | 日 |
| 指导单位 | 计算机学院计算机科学与技术系 | | | | | |
| 指导教师 | 陈兴国 | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 学生姓名 | 张颖 | 班级学号 | B18030406 |
| 学院(系) | 计算机学院 | 专 业 | 计算机科学与技术系 |

**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验名称** | **利用真值表法求取主析取范式以及主合取范式的实现** | | | **指导教师** | 陈兴国 |
| **实验类型** | **验证** | **实验学时** | **4** | **实验时间** | 2019.9.16 |
| 1. **实验目的和要求**   中文五号宋体，英文五号Times new roman字体，1.25倍行距  描述本次实验的目的和要求。 | | | | | |
| 二、**实验环境(实验设备)**  中文五号宋体，英文五号Times new roman字体，1.25倍行距  硬件：微型计算机  软件：Windows 操作系统、Microsoft Visual C++6.0、Java等可视化编程语言 | | | | | |
| **三、实验原理及内容**  核心算法的源码：  $('#goBtn').click(function () {  let List = [],  argumentList = [],  conditions = [],  tfList = [],  resultList = [],  MainDisjunctiveNormalForm,  PrincipalConjunctiveNormalForm,  a = [],  finalArgument;  getValue()  argument()  isError()    /\*  错误以及空判断  \*/  function isError() {  if ((!!finalArgument.indexOf('∧') || !!finalArgument.indexOf('→') || !!finalArgument.indexOf('⇔') || !!finalArgument.indexOf('∨'))) {  truthTable()  mainDisjunctiveNormalForm()  principalConjunctiveNormalForm()  } else {  console.log("error")  $('body').append(`  <div class ="error">  <p>检查一下表达式是不是写错了噢~！</p>  <button id="close">好的好的，我马上乖乖写好（</button>  </div>  `)  $('.error').css('opacity', '0')  setTimeout(() => {  $('.error').css('opacity', '1')  }, 300)  }  $('#close').click(function () {  $('.error').css('opacity', '0')  setTimeout(() => {  $('.error').remove()  }, 300)  })  }  /\*  取得输入的表达式  \*/  function getValue() {  let list = [...$('.ui-draggable')];  list.sort(function (a, b) {  return $(a).offset().left - $(b).offset().left  }).map(item => {  List.push(item.innerText)  })  finalArgument = List.join('')  }  /\*  单条件运算符  \*/  function SingleCondition() {  typeof List == 'string' ?  List = List.split('→') :  List = List.join('').split('→')  List = `┐(${List[0]})∨(${List[1]})`.split('')  andOrNot()  }  /\*  合取和析取  \*/  function andOrNot() {  List = List.join('').replace(/┐/g, '!').replace(/∧/g, '&&').replace(/∨/g, '||').replace(/→/g, '∨')  }  /\*  双条件运算符  \*/  function DoubleConditions() {  List.join('').split('→').join(',').split(',').map(item => {  a.push(item)  })  a[0] = a[0].replace(/⇔/, '^').split('')  a[0].splice(0, 0, '!')  a[0] = a[0].join('')  List = [...a].join('→')  List.indexOf('→') ?  SingleCondition() :  andOrNot()  }  /\*  对表达式进行转化的总函数入口  \*/  function argument() {  List.join('').indexOf('⇔') >= 0 ?  DoubleConditions() :  List.join('').indexOf('→') >= 0 ?  SingleCondition() :  andOrNot()  List = List.split('')  }  /\*  创建真值表  \*/  function createTruthTable() {  let thTemplateList = [],  trList = [],  tdList = [];  argumentList.map(item => {  thTemplateList.push(`<th>${item}</th>`)  tdList.push(`<td></td>`)  })  for (let i = 0; i <= parseInt(tfList, 2); i++) {  trList.push(`<tr class="TruthFalse">${tdList.join('')}</tr>`)  }  $('.truthTable').append(`  <table border="1">  <tr>${thTemplateList.join('')}</tr>  ${trList.join('')}  </table>  `)  }  /\*  条件组合  \*/  function conditionalCombination() {  List.map(item => {  item.match('[A-Z]') ? argumentList.push(item) : ''  })  argumentList = [...new Set(argumentList)];  tfList = new Array(argumentList.length).fill(1).join('');  for (let i = 0; i <= parseInt(tfList, 2); i++) {  let item = i.toString(2).padStart(argumentList.length, '0')  conditions.push(item)  }  argumentList.push(List.join(''))  conditions = conditions.join('').replace(/0/g, 'F').replace(/1/g, 'T').split('')  }  /\*  填写真值表内容  \*/  function fillTruthTable() {  $(".TruthFalse").find("td:last-of-type").addClass("result");  $(".TruthFalse").find("td:not(.result)").each((index, cur) => {  cur.innerText = conditions[index]  })  for (let i = 0; i < conditions.length; i + argumentList.length - 1) {  resultList.push(conditions.splice(i, i + argumentList.length - 1))  }  let finalResult = [];  for (let k = 0; k <= parseInt(tfList, 2); k++) {  let arr = [...List];  for (let j = 0; j < List.length; j++) {  for (let i = 0; i < argumentList.length - 1; i++) {  if (List[j] == argumentList[i])  arr[j] = resultList[k][i]  }  }  finalResult.push(eval(arr.join('').replace(/F/g, 'false').replace(/T/g, 'true')))  }  $('.result').each((index, cur) => {  cur.innerText = finalResult[index]  })  // finalArgument = '((┐┐┐P∨Q)⇔(A∧B))→(R∨(┐B∧C))'  $(".truthTable table").find("th:last-of-type")[0].innerText = finalArgument  }  /\*  创建并利用真值表的总函数入口  \*/  function truthTable() {  conditionalCombination()  createTruthTable()  fillTruthTable()  $('.container').css('transform', 'translateY(100vh)')  $('.truthTableBox').css('transform', 'translateY(-98vh)')  if ($('.truthTable').height() > '450')  $('.truthTable').css('overflowY', 'scroll').css('overflowX', 'hidden')  }  /\*  主析取范式  \*/  function mainDisjunctiveNormalForm() {  let mainDisjunctiveNormalFormList = [],  mainDisjunctiveNormalForm = [];  $('.result').each((index, cur) => {  if (cur.innerText == 'true') {  for (let i = 0; i < resultList[index].length; i++) {  resultList[index][i] == 'T' ?  mainDisjunctiveNormalFormList.push(argumentList[i]) :  mainDisjunctiveNormalFormList.push(`┐${argumentList[i]}`)  }  }  })  for (let i = 0; i < mainDisjunctiveNormalFormList.length; i + argumentList.length - 1) {  mainDisjunctiveNormalForm.push(mainDisjunctiveNormalFormList.splice(i, i + argumentList.length - 1))  }  mainDisjunctiveNormalForm.map(item => {  item[0] = `(${item[0]}`  item[item.length - 1] = `${item[item.length-1]})∨`  })  MainDisjunctiveNormalForm = mainDisjunctiveNormalForm.join('').replace(/,/g, '∧')  MainDisjunctiveNormalForm = MainDisjunctiveNormalForm.substring(0, MainDisjunctiveNormalForm.length - 1)  $('#MainDisjunctiveNormalForm').text(`${MainDisjunctiveNormalForm}`)  }  /\*  主合取范式  \*/  function principalConjunctiveNormalForm() {  let principalConjunctiveNormalFormList = [],  principalConjunctiveNormalForm = [];  $('.result').each((index, cur) => {  if (cur.innerText == 'false') {  for (let i = 0; i < resultList[index].length; i++) {  resultList[index][i] == 'F' ?  principalConjunctiveNormalFormList.push(argumentList[i]) :  principalConjunctiveNormalFormList.push(`┐${argumentList[i]}`)  }  }  })  for (let i = 0; i < principalConjunctiveNormalFormList.length; i + argumentList.length - 1) {  principalConjunctiveNormalForm.push(principalConjunctiveNormalFormList.splice(i, i + argumentList.length - 1))  }  principalConjunctiveNormalForm.map(item => {  item[0] = `(${item[0]}`  item[item.length - 1] = `${item[item.length-1]})∧`  })  PrincipalConjunctiveNormalForm = principalConjunctiveNormalForm.join('').replace(/,/g, '∨')  PrincipalConjunctiveNormalForm = PrincipalConjunctiveNormalForm.substring(0, PrincipalConjunctiveNormalForm.length - 1)  $('#PrincipalConjunctiveNormalForm').text(`${PrincipalConjunctiveNormalForm}`)  }  })  测试结果： | | | | | |

**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）  中文五号宋体，英文五号Times new roman字体，1.25倍行距  **说明：这部分内容主要包括：在编程、调试或测试过程中遇到的问题及解决方法、本次实验的心得体会、进一步改进的设想等。**  (一)实验中遇到的主要问题及解决方法  主要问题：在条件组合列真值表的时候没有合适算法  解决办法：最后利用行数与变量个数关系进行真值表条件的排列组合  （二）实验心得  遇到问题应当耐心，不可急躁，越急躁越解决不了问题。  （三）意见与建议（没有可省略） | | | | | |
| 1. **支撑毕业要求指标点**   支撑毕业要求的指标点为：   * 1-4掌握计算机科学与技术领域的专业知识，能将专业知识用于分析和解决计算机领域复杂工程问题。   √   * 2-1能够应用数学、自然科学和工程科学的基本知识，识别和分析计算机领域复杂工程问题的特征。 | | | | | |
| **六、指导教师评语 (含学生能力达成度的评价)** | | | | | |
| **成 绩** |  | **批阅人** |  | **日 期** |  |

如果不太想写太多字，“指导教师评语”也可以设计为如下的各选择项用打勾形式（仅仅作为一个简单示例，请各课程负责人根据课程和实验情况以及支撑的指标点来自行设定选择项，同一门课程的不同实验评分细则项允许存在不同）：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **评 分 细 则** | **评分项** | **优秀** | **良好** | **中等** | **合格** | **不合格** |
| **遵守实验室规章制度** |  |  |  |  |  |
| **学习态度** |  |  |  |  |  |
| **算法思想准备情况** |  |  |  |  |  |
| **程序设计能力** |  |  |  |  |  |
| **解决问题能力** |  |  |  |  |  |
| **课题功能实现情况** |  |  |  |  |  |
| **算法设计合理性** |  |  |  |  |  |
| **算法效能评价** |  |  |  |  |  |
| **回答问题准确度** |  |  |  |  |  |
| **报告书写认真程度** |  |  |  |  |  |
| **内容详实程度** |  |  |  |  |  |
| **文字表达熟练程度** |  |  |  |  |  |
| **其它评价意见** |  | | | | |
| **本次实验能力达成评价（总成绩）** |  |  |  |  |  |