**深 圳 大 学 实 验 报 告**

**课程名称： 编译原理**

**实验项目名称：实验一 高级语言及其文法**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术**

**指导教师： 罗成文**

**报告人：谭嘉豪 学号： 2020152087 班级： 数计班**

**实验时间： 2023年3月16日**

**实验报告提交时间： 2023年3月27日**

**教务部制**

|  |
| --- |
| **一、实验目的与要求**   1. **使用 C、C++、或 Java 完成程序编写；** 2. **使用实验所提供的模板撰写实验报告，要求内容详实，有具体的设计描述、 关键的代码片段、及实验结果屏幕截图；** 3. **在截止日期前将代码、实验报告、测试文件（如有）等所有实验相关文件压缩到一个压缩包“姓名\_学号\_实验一.rar ”上传至 Blackboard。**   **二、实验内容**   1. **文法的定义与读取：现规定文法由 Grammar.txt 文件保存，请制定文法文件的具体存储格式。如 文法 G={{S,A,B,C}, {a,b,c}, {S->ABC,A->a,B->b,C->c}, S}在 Grammar.txt 文件中可由以下方式描述并存储：**   **--------------------------------------------------**  **S,A,B,C**  **a,b,c**  **S->ABC,A->a,B->b,C->c**  **S**  **--------------------------------------------------**  **文法的文本形式可根据自己需要自由定义，在此基础上，编程实现对任意文法文件的读取。**   1. **文法的分类：根据 Chomsky 的文法体系分类（课本 P40），文法分为四大种类。请在任务一的基础上，编程实现对 Grammar.txt 中存储的文法进行分类，自动判断其所属类别。例如任务一中所给出的文法G应被判定为2型文法，即上下文无关文法。请设计分类方法，并设计四类不同的测试文法测试分类结果的正确性。**   **三、实验步骤及说明**   1. **文法的定义与读取** 2. **文法的定义**   **在本次实验中，文法的描述方式如下：**   * **第一行：描述文法的变量，每个变量之间以一个逗号分隔。** * **第二行：描述文法的终结符，每个终结符之间以一个逗号分隔。** * **第三行：描述文法的所有产生式，每个产生式均具有形式，当多个产生式的左部相同式，可使用形式描述，产生式之间以一个逗号分隔。** * **第四行：一个字符描述文法的开始符号。**   **PS：空串以一个‘.’来表示。**     1. **文法的读取**   **为存储所读取的文法数据，定义了一个Grammar类，类的成员属性有四个，分别是：**   * **V：一个无序哈希表，用于存储变量的集合。增删查改操作的时间复杂度均是。** * **T：一个无序哈希表，用于存储终结符的集合。增删查改操作的时间复杂度均是。** * **P：一个无序哈希表，用于存储所有产生式的集合，对于唯一的产生式左部，映射到一个存储其所有产生式右部的vector。增删查改操作的时间复杂度均是。** * **S：一个字符，表示文法的开始符号。**     **接下来通过C++文件操作读取文法，使用getline()函数读取一行获取变量、终结符、产生式，由于开始符号只有一个字符，可以使用get()函数获得。**    **读取变量的代码如下，由于变量之间以一个逗号分割，因此读取时跳过逗号即可。**    **读取终结符的代码如下，由于终结符之间以一个逗号分割，因此读取时跳过逗号即可。**    **对于由于考虑到的情况，因此定义一个DecomposeProduction()函数，用于将一个产生式进行分解，将其分解成，，…，，并将其存储在哈希表中。**  **分解产生式的函数如下：**    **读取产生式的代码如下，注意最后一个产生式需要在跳出循环后分解。**     1. **文法的分类**   **四种文法之间的逐级“包含”关系如下图。**    **由于四种文法之间的关系是将产生式作进一步限制而定义的，因此对文法进行分类，可以分为两种思路：**   * **思路一：由于所有文法都是0型文法，因此可以先判断文法是否是1型文法，若不是1型文法，则文法为0型文法；若是1型文法，再判断是否是2型文法，若不是2型文法，则文法为1型文法；若是2型文法，再判断是否是3型文法，若不是3型文法，则文法为2型文法；若是3型文法，则文法最终为3型文法。** * **思路二：与思路一相反，首先判断文法是否是3型文法，若是3型文法，则文法为3型文法；若不是3型文法，再判断是否是2型文法；若是2型文法，则文法为2型文法；若不是2型文法，再判断是否是1型文法，若是1型文法，则文法为1型文法；若不是1型文法，则文法最终为0型文法。**   **PS：本次实验使用了思路二对文法进行分类，以及C++程序进行实现。**  **接下来介绍本次实验使用到的以下自定义函数。**   1. **isOfVariable**   **该函数用于判断字符串s是否属于变量的集合中。**     1. **check\_CSG**   **判断文法是否为CSG，遍历所有的产生式，若存在产生式右部为空串‘.’或产生式的右部长度严格小于左部，该文法不是CSG。**     1. **check\_CFG**   **判断文法是否为CFG，遍历所有的产生式，若存在产生式的左部不属于变量的集合或存在产生式右部为空串‘.’或产生式的右部长度严格小于左部，该文法不是CFG。**     1. **check\_RG**   **判断文法是否为RG，判断文法是否是左线性文法或右线性文法。**  **判断文法是否是左线性文法：遍历所有的产生式，判断所有产生式是否都为A->w或A->Bw的形式，其中A，B属于变量集合，w属于终结符的正闭包集合。**    **判断文法是否是右线性文法：遍历所有的产生式，判断所有产生式是否都为A->w或A->wB的形式，其中A，B属于变量集合，w属于终结符的正闭包集合。**     1. **printGrammar**   **打印文法的函数，用于将读取的文法进行打印输出。**     1. **check\_Grammar**   **调用以上函数对文法进行分类判断。**    **对实验要求的Grammar.txt文件所存储的文法进行分类，可以在终端观察到打印的文法格式及其判断结果--** **The current grammar is CFG，与预期结果一致。**     1. **进一步验证程序的正确性**   **最后我使用课本P41与课本P42的四种文法的例子按照上述定义的文法格式存储在记事本中，使用编写的程序进行判断，进一步验证程序的正确性。**  **RG.txt：**    **测试结果：**    **CFG.txt：**    **测试结果：**    **CSG.txt：**    **测试结果：**    **PSG.txt：**    **测试结果：**    **可以看到，四种文法的测试均通过。**  **四、感想及其他**  **通过本次实验，我编写了C++程序对文法进行定义与读取、对文法进行分类，并进一步利用课本上的例子对程序进行验证，对文法及文法的分类有了更加清晰的认识。** |

深圳大学学生实验报告用纸

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  2023年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。