

课程实验报告

课 程 名 称： 编译原理

实验项目名称： 语法分析表的构造

专 业 班 级： 软件2003

姓 名： 曾佳宝

学 号： 202026010324

指 导 教 师： 黎文伟

完 成 时 间： 2023 年 4 月 12 日

信息科学与工程学院

|  |  |
| --- | --- |
| 实验题目：语法分析表的构造 | |
| 实验目的：   1. 学习和掌握消除左递归和提取左公因子的方法 2. 学习和掌握求解FIRST和FOLLOW函数的一般方法和步骤 3. 学习和掌握DFA的方法和步骤 4. 掌握判断LL(1)文法和SLR(1)文法的一般算法 5. 掌握构造LL(1)语法分析表和LR(1)语法分析表的一般算法 | |
| 实验环境：笔记本电脑、Win10、idea Java环境 | |
| 实验内容及操作步骤：  **一、基本数据结构**  1）文法符：作为终结符和非终结符的基类，其中SymbolType有三种类型：TERMINAL（终结符），NONTERMINAL（非终结符），NULL（ε）。     1. 终结符定义：其中LexemeCategory定义如下      1. 词法类型定义        1. 非终结符定义     4）产生式定义    5）LL(1)语法分析表中，每一格的定义    6）LR(0)项目定义    7）LR(0)项集定义    8）变迁边定义    9）DFA定义    10）LR(1)语法分析表中ACTION部分的定义，其中ActionCategory有三种类型：r（reduce规约，id为产生式）、s（shift移入，id为状态）、a（accept，接受）    11）LR(1)语法分析表中GOTO部分的定义    12）产生式概述表定义    **二、针对LL语法分析，实现如下函数**  1）判断是否存在左递归以及左递归的消除实现  判断左递归：判断非终结符的所有产生式，若存在产生式右部的头部和该非终结符相同，则表明存在左递归。  消除左递归：对于含左递归的文法符 X，将其产生式分成含左递归的和不含左递归的两个部分。把 X 的左递归变换成了 X’的右递归，并且新增一个ε产生式。具体过程如下：    实现函数：    函数测试：A->Aa,A->b消除左递归      101397c2c0a24853b5e98bd3631d53f    2）产生式有左公因子的判断，以及左公因子的提取实现  判断左公因子：保存非终结符的所有产生式右部的头部信息，若存在两个及以上产生式右部的头部信息相同，则表明存在左公因子。  提取左公因子：将含有公共头部信息的产生式写做一个公共头部信息+A’。再将剩下的部分由A’推导。具体过程如下：    实现函数：    函数测试：A->ab, A->ac      0660964f74b6093dc70ad92e00b5328  3）产生式的FIRST函数求解  实现方法：对于产生式X->Y1Y2......Yn-1Yn,FIRST(Y1)∈FIRST(X)显然成立。但如果从Y1至Yj，0<j<n，全为非终结符，且都含虚产生式，那么FIRST(Yj+1) 属于 FIRST(X)。算法思想如下：    实现函数：    函数测试：      8c97faf37323a20ac80af5dbf5e3e83  4）非终结符的FIRST函数求解  实现方法：对每个非终结符的产生式，求其FIRST函数，再将其合并即可。  实现函数：    函数测试：文法同上，输出每个非终结符的FIRST函数  8c97faf37323a20ac80af5dbf5e3e83  5）非终结符的FOLLOW函数求解  实现方法：穷举所有情形，找出跟在T’后面的终结符。产生式X->Y1Y2......Yn-1Yn蕴含有如下两个FOLLOW信息。  ①对于末尾符Yn，如果它为非终结符，那么FOLLOW(X)∈FOLLOW(Yn)。若Yi为终结符（0＜i＜n），且从Yi+1至Yn全为非终结符，且都含虚产生式，那么FOLLOW(X)属于FOLLOW(Yi)。  ②除了末尾符Yn之外，对于产生式右部中任一文法符Yi，其中0＜i＜n，如果Yi是一个非终结符，那么FIRST(Yi+1)-ε∈FOLLOW(Yi)。如果Yi为非终结符（0＜i＜n-1），且从Yi+1至Yj（i+1＜j＜n）全为非终结符，且都含虚产生式，那么FIRST(Yj+1)-ε∈FOLLOW(Yi)。  具体算法如下：    实现函数：    根据每个非终结符求出FOLLOW相关信息后，将依赖的非终结符的FOLLOW信息加入其中：    函数测试：文法同上，先求其FIRST函数，再求FOLLOW函数，最后进行求解        6）LL（1）文法的判断  实现方法：对于一个文法，其中的任一非终结符X，设其实产生式有X->α1，X->α2，……， X->αn，若满足 FIRST(X->αi)∩ FIRST(X->αj) =Φ，其中i≠j且0<i,j≤n。如果X还有虚产生式X->ε,若进一步满足 FIRST(X->αi)∩FOLLOW(X)=Φ,其中0<i≤n。具有这种特性的文法被称之为 LL(1)文法。  实现函数：    函数测试：文法同上，生成FIRST和FOLLOW函数后，对每个非终结符进行依次判断，若每个非终结符都满足条件，则其为LL（1）文法。    7）LL（1）语法分析表的填写  实现方法：对于 LL(1)文法，在自顶向下最左推导当中，设当前要推导的非终结符为X，当前词为w，如果w∈FIRST(X->αi)，就选择 X->αi进行推导。如果存在有X->ε且w∈FOLLOW(X)，就选择X->ε进行推导。  实现函数：    函数测试：文法同上，求出FIRST和FOLLOW函数判断其是否为LL（1）文法，再构造语法分析表      **三、针对LR语法分析，实现如下函数。**  1）void getClosure(ItemSet itemSet);  函数作用：基于LR（0）核心项的闭包求解。  实现方法：找到所有待约项目，根据待约项目推导出非核心项。  实现函数：    其中判断item是否重复的函数如下：    函数测试：求E’的闭包I0      2）void exhaustTransition(ItemSet itemSet)  函数作用：穷举一个LR（0）项集的变迁，其中中包括驱动符的穷举，下一项集的创建，下一项集中核心项的确定，下一项集是否为新项集的判断。  实现方法：首先找到所有驱动符，对每个驱动符创建一个项集，求该项集的核心项及其闭包，再判断该项集是否为新项集。最后创建一条变迁边连接两个项集。  实现函数：    其中变化如下：  1.新维护了一个链表，用于保存所有项集和之后判断新项集    2.新增了一个构造函数，使其在原项目基础上原点位置右移，且变为核心项    3.判断两项集是否相同：判断两者LR0项目是否相等    函数测试：以项目集I0的变迁为例，先求出其核心项，再求其闭包。  954f8976f7a7470001476eb4a1f5a3d    3）文法的LR（0）型DFA求解  实现方法：在符号栈中，从状态0开始，穷举所有变迁。对于每一变迁的驱动文法符，求下一状态（即核心项闭包）。如果下一状态是一个新状态，则使用相同策略穷举。如此迭代下去，直到把所有的状态变迁都穷举出来。  实现函数：    函数测试：文法如上，对每个项集求其变迁即可。以下是最终项集    以下是DFA的开始状态和对应边：    a9f191a686ab27589d20baa2807111e  4）SLR（1）文法的判断  实现思路：对于每个项集，找到它的移入终结符集合和规约项目集合，  ①规约项目FOLLOW集合与移入终结符集合有冲突 ==> 移入-规约冲突  ②规约项目FOLLOW集合之间有冲突 ==> 规约-规约冲突  上述两种情况都不发生，则为SLR（1）文法。  实现函数：    函数测试：分别用下面两个文法进行测试，第一个文法是SLR（1）文法，但第二个文法存在规约-规约冲突。   |  |  | | --- | --- | |  | 44e9e15e334b4b0b597afd8a7eb17d6 | |  |  |   5）LR语法分析表的填写  实现思路：是从 0 状态开始，逐行填写。对于DFA中的每个状态，它的每条出边都要在语法分析表中对应填写一格。  ①如果出边的驱动符为终结符，就填到ACTION 部分，在目标状态序号前加s，表示移入(shift)。  ②如果出边为非终结符，就填到GOTO 部分，直接填上目标状态序号即可。  ③如果包含规约项目，对该产生式头部非终结符的FOLLOW集合中的每个终结符，都要在其对应格中填上规约项的产生式序号，并在产生式序号前加 r，表示规约(reduce)。  ④如果包含接受项目，就填到ACTION 部分，在非终结符“#”下填a，表示接受。  实现函数：  1.语法分析表    2.产生式概述表    函数测试：文法如上，求出FIRST和FOLLOW集，求出项集及变迁边，创建DFA，基于DFA填写下列表格。  1.FIRST和FOLLOW集    2.产生式概述表      3.语法分析表      收获与体会：   1. 通过此处实验对Java语言进行了一次回忆，并且对链表、Map等数据结构的使用有了进一步的心得。 2. 通过此次练习，对LL语法分析有了进一步的理解，并且在编码实现的过程中可以逐渐熟练掌握消除左递归和提取左公因子的方法。 3. 对于求FIRST函数和FOLLOW函数也有了进一步的了解，但是对判断依赖环仍有一定挑战。 4. 对于LR(0)项目的闭包求解和变迁能够较好地掌握，其中分析核心项和判断新项集作为核心内容也有了更好地掌握。 5. 基于上一次实验，本次实验求解LR(0)型DFA，在回顾DFA知识的同时掌握了语法分析的基本步骤。 6. 对于判断LL(1)文法和SLR(1)文法有了更加深入的了解，能够区分LL(1)语法分析表和LR语法分析表的差异，并且完成构造。 | |
| 实验成绩 |  |