《编译原理》实验报告

实验课程：编译原理

实验项目：文法问题处理器

指导老师：黄煜廉

专　　业：计算机科学与技术

班　　级：2019 级计算机科学与技术 4 班

姓　　名：陆泓相

学　　号：20192131076

文法问题处理器

一、 实验目的

设计一个应用软件，以实现文法的化简及各种问题的处理。

二、 实验内容

（1）系统需要提供一个文法编辑界面，让用户输入文法规则（可保存、打开存有文法规则的文件）

（2）化简文法：检查文法是否存在有害规则和多余规则并将其去除。系统应该提供窗口以便用户可以查看文法化简后的结果。

（3）检查该文法是否存在着左公共因子（可能包含直接和间接的情况）。如果存在，则消除该文法的左公共因子。系统应该提供窗口以便用户可以查看消除左公共因子的结果。

（4）检查该文法是否存在着左递归（可能包含直接和间接的情况），如果存在，则消除该文法的左递归。系统应该提供窗口以便用户可以查看消除左递归后的结果。

（5）求出经过前面步骤处理好的文法各非终结符号的 first 集合与 follow 集合，并提供窗口以便用户可以查看这些集合结果。【可以采用表格的形式呈现】

（6）对输入的句子进行最左推导分析，系统应该提供界面让用户可以输入要分析的句子以及方便用户查看最左推导的每一步推导结果。【可以采用表格的形式呈现推导的每一步结果】

三、 运行环境及注意事项

本项目的编程环境为：Qt 5.14.2

注意事项：在 Qt Creator 打开项目时，项目的路径不能包含中文，可先把 GrammarProcessor 文件夹放到不包含中文的路径下再打开。

使用步骤：打开 Qt Creator，然后打开 GrammarProcessor 文件夹中的 GrammarProcessor.pro 文件，即可运行程序。

四、 设计思路

(一)化简文法

化简文法主要去除文法规则中的多余规则，也就是文法中任何句子的推导都不会用到的规则，换句话来说就是文法中不能含有不可到达和不可终结的非终结符。

1) 消除含有不可到达的非终结符的文法：用一个集合 reachable 来存储可到达的非终结符，用一个队列来存储非终结符，队列里的这些非终结符是位于即将要访问的文法的左部。算法描述如下：

```
1. 初始化reachable只包含开始符号，开始符号进队；
2. while(队列不为空)
3.     当前非终结符号出队；
4.     访问以当前非终结符号为左部的文法；
5.     把右部包含的所有非终结符存储到reachable中，未被访问过的非终结符逐一进队；
6. 删除左部或右部包含不在reachable里的非终结符的文法；
```

2) 消除含有不可终止的非终结符的文法：用一个集合 terminable 来存储可终结的非终结符。算法描述如下：

```
1. 把所有文法右部只包含终结符或 ε (用@来表示)的非终结符存入terminable；
2. while(terminable有变化)
3.     对于左部为不在terminable里的非终结符的文法：
4.         如果文法右部只包含终结符和在terminable里的非终结符，则把左部的非终结符加入terminable；
5. 删除左部或右部包含不在terminable里的非终结符的文法；
```

(二)提取左公共因子

假定 A 的规则是：

A→δβ1|δβ2|…|δβn|γ1|γ2|…|γm（其中，每个γ不以δ开头）

那么这些规则可以改写为：

A→δA'|γ1|γ2|…|γm

A'→β1|β2|…|βn

经过反复提取左公共因子，就能够把每个非终结符（包括新引进者）的所有候选首符集便成为两两不相交。由于提取左公共因子之后可能会产生多余规则，因此最后还需要消除多余的文法规则。

(三)消除左递归

左递归包括直接左递归和间接左递归，对于直接左递归，按如下方法进行消除：

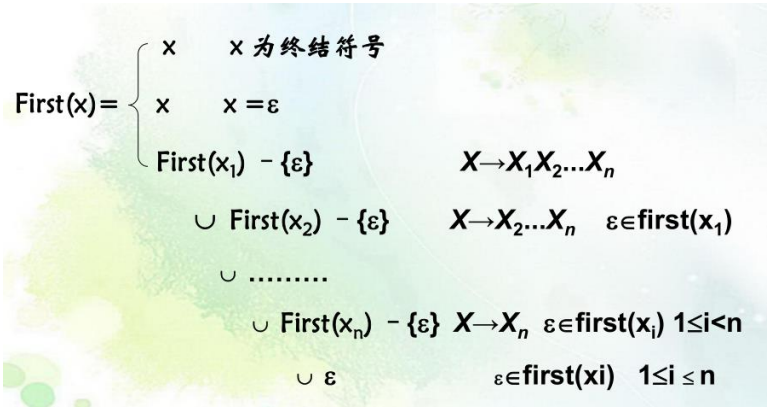
A→Aa|b　改写为　A→bA'　和　A'→aA'|ε

对于间接左递归，解决思路为：逐个对非终结符进行解决，将干净非终结符代入未解决的非终结符中，并将其消除干净，反复实施。算法描述如下：

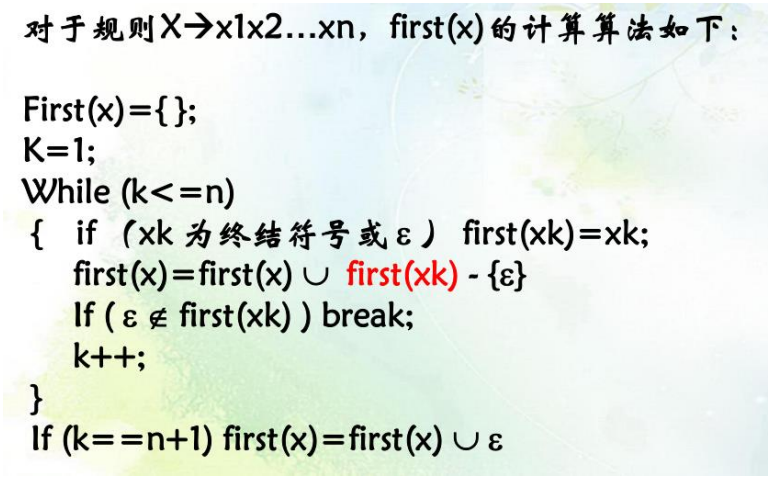
```
1. 将文法的所有非终结符号按某种顺序排列A1，…，Am;
2. for (i=1; i<=m; i++)
3.     for (j=1; j<i; j++)
4.         将规则Ai→Ajα改写;
           //改写方法如下：如果Aj→β1 | β2 | ... | βk
           //Ai→β1α | β2α | ... | βkα
5.         消除Ai规则中得非直接左递归;
6. 化简所得文法，即消除多余规则;
```

(四)求非终结符号的 first 集合

First 集合计算方法的归纳如下：



算法描述如下：



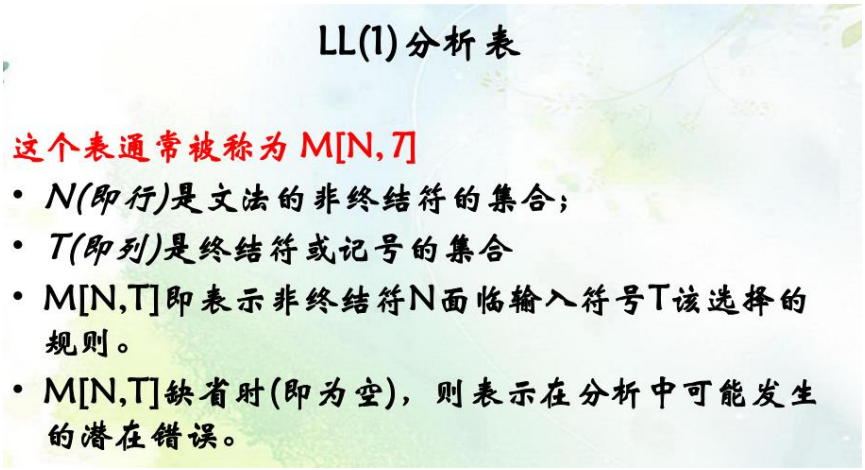
(五)求非终结符号的 follow 集合

对于 Follow 集合的定义如下：

算法描述如下：

(六)对输入的字符串进行最左推导分析

采用 LL(1)文法分析法进行最左推导分析，LL(1)分析表的定义及构造思路如下：



LL(1)分析表的构造步骤

为每个非终结符A和规则A→α 重复以下两个步骤：

1)对于First(α)中的每个记号a，都将A→α添加到项目M[A, a]中。

2)若ε在First(α)中，则对于Follow (A) 的每个元素a(记号或是$)，都将A→α 添加到M[A, a]中。

3)把分析表A中每个未定义元素置为ERROR。

注意：通常用空白表示即可

五、 实验结果分析

对于实验结果的演示，我另外录了一个视频，这里就只放一张整体的结果图。