

编译原理大作业——构造算符优先分析表

叶增渝 519030910168

丁悠然 519030910178

(一) 基本知识与题目分析:

(1) 算符优先分析法

比较相邻运算符的优先级来确定句型的可归约串并指导归约进行的一种方法。

(2) 各类优先级

$a < b$ (a 的优先级小于 b): 文法中有形如 $A \rightarrow \dots aB \dots$ 的产生式且 $B \Rightarrow b \dots$ 或 $B \Rightarrow Cb \dots$

$a = b$ (a 的优先级等于 b): 文法中有形如 $A \rightarrow \dots ab \dots$ 或者 $A \rightarrow \dots aBb \dots$ 的产生式

$a > b$ (a 的优先级高于 b): 文法中有形如 $A \rightarrow \dots Bb \dots$ 的产生式且 $B \Rightarrow \dots a$ 或 $B \Rightarrow \dots aC$

实际上两个终结符之间的关系表示了它们的归约顺序。

(3) 算符优先分析表的基础

想要构造算符优先分析表, 首先需要为每一个非终结符寻找 Firstvt 与 Lastvt 集合, 表示在这个非终结符归约的过程中或之前会被归约的终结符集合, 并且在展开后出现上述一个非终结符被两个终结符夹在中间的情况。

Firstvt 集合三条规则:

(a) $A \rightarrow a \dots$, 即以终结符开头, 该终结符入 Firstvt

(b) $A \rightarrow B \dots$, 即以非终结符开头, 该非终结符的 Firstvt 入 A 的 Firstvt

(c) $A \rightarrow Ba \dots$, 即先以非终结符开头, 紧跟终结符, 则终结符入 Firstvt

Lastvt 集合三条规则:

(a) $A \rightarrow \dots a$, 即以终结符结尾, 该终结符入 Lastvt

(b) $A \rightarrow \dots B$, 即以非终结符结尾, 该非终结符的 Lastvt 入 A 的 Lastvt

(c) $A \rightarrow \dots aB$, 即先以非终结符结尾, 前面是终结符, 则终结符入 Lastvt

(4) 算符优先分析表的构造:

(a) 寻找终结符在左边, 非终结符在右边的符号对。如 $+T$ 则 $+ < \text{FIRSTVT}(T)$

(b) 寻找非终结符在左边, 终结符在右边的符号对。如 $T+$ 则 $\text{LASTVT}(T) > +$

(c) 终结符号 $\$$, 始终有 $\$ = \$$, 设初始终结符为 E 则 $\$ < \text{FIRSTVT}(E)$ 且 $\text{LASTVT}(E) > \$$

(5) 二义性的判断

一个文法有二义性, 这说明可以以两种不同的顺序进行归约得到相同结果, 那么在构造算符优先分析表时, 必然至少会有一次符号冲突产生 (代码上表现为符号覆盖), 那么此时我们可以断言文法具有二义性

(二) 实现功能说明

程序读入一系列产生式 (上下文无关), 其中用 \rightarrow 分隔左右式, 左边为一个非终结符, 右边为相关式子, 空格个数位置不影响, 非终结符以大写字母表示, 其余终结符以小写字母或字符的形式出现, 形如 $A \rightarrow B + s$, 可使用 | 分隔, 在同一行表示多个产生式

程序检测文法, 如果有二义性则在 cmd 中输出相关信息, 输出文件中也只有提示信息; 若没有二义性, 那么在输出文件中输出相应的算符优先分析表。

(三) 设计思路、具体实现与函数说明:

(1) 设计思路

根据上述基本知识, 我们首先处理输入, 将其分为左部与右部并进行整理, 找出所有出现的终结符与非终结符。

然后对每个非终结符，寻找其 firstvt 与 lastvt 集合，并寻找最终归约到的非终结符，为算符优先分析表的构造做准备

接着尝试构造算符优先分析表对应的矩阵，若在某一格有不同的二次填空，则说明有不同的归约顺序，存在二义性，没有则构造完成，进行输出即可。

(2) 使用库文件

<map> 一个键值对容器，其内部自建一颗红黑树，有对数据自动排序的功能。

<set>集合，一个内部自动有序而且不重复元素的容器

(3) 使用结构

MAX: 自定义常量

WordString 类: 一个含 char 类型的句子左部，string 类型的右部和两个简单函数

char source: 归约结束时的非终结符 (LR 分析的最终结果)

char relation[MAX][MAX]: 记录算符优先文法分析表信息的矩阵

vector<char> VT: 所有终结符的集合记录

vector<WordString> VN_set: 文法串输入的整理后的集合

map<char,int> VN_dic: char 为键，int 为对应的值 (方便查询使用)

set<char> first[MAX]: 记录所有非终结符对应的 firstvt 集合元素的矩阵

set<char> last[MAX]: 记录所有非终结符对应的 lastvt 集合元素的矩阵

int used[MAX]: 形成 VT 的辅助集合

int vis[MAX]: 表示对应键是否进行过 dfs 寻找，防止多次查询

bool flag = false: 表示文法是否具有二义性，初始化为无二义性

(4) 函数说明

void handle_input(istream &inFile): 处理输入文本，使用 usd，完成 VT，VN_set 的记录

void find_origin(): 寻找最终归约到的非终结符 source,若循环，我们规定最后的非终结符是第一个语句的左侧的非终结符

void first_dfs(int x): 将 x 对应下标的终结符的 FIRSTVT 集合进行完善的递归函数

void make_first(): 寻找所有终结符 FIRSTVT 集合并记录到 first[MAX]中，通过调用 first_dfs()进行深度优先遍历实现

void last_dfs(int x): 将 x 对应下标的终结符的 LASTVT 集合进行完善

void make_last(): 寻找所有终结符 LASTVT 集合并记录到 last[MAX]中,通过调用 last_dfs()进行深度优先遍历实现

void make_table(ofstream &outFile): 根据基本知识中提到的规则，完善算符优先分析表 relation[MAX][MAX]，通过是否有冲突填写判断是否有二义性，若有则输出相关信息；没有则将文法的算符优先分析表输出至指定路径文件中

int main(): 主函数，接受输入信息，打开输入文件和创建输出文件，然后分别调用 find_origin()、make_first()、make_last()、make_table()函数对全局变量进行计算存储，并最终输出

(四) 实验环境

操作系统: Windows 10

编程语言及调式环境: C++ (Mingw-w64)

(五) 实验过程

首先，了解了算符优先分析法的相关基础知识，了解内容

然后，首先尝试构造了不含\$的算符优先分析表构造，再将\$的判断纳入其中

接着，了解文法二义性原因，尝试着判断如何在构造过程中确定二义性存在与否，向程序中添加二义性判断

最后，测试两个 input 文件，查看实现情况，测试结果

(六) 实验结果

(1) input1.txt (无二义性，可算符优先构造的文法)

```
input1.txt ×
编译原理 > input1.txt
1 E -> E + T | T
2 T -> T * F | F
3 F -> (E) | i
4
```

运行结果：

```
PS D:\VS code\C C++\编译原理> cd "d:\VS code\C C++\编译原理\" ; if ($?) { g++ analyTable.cpp -o analyTable } ; if ($?) { .\analyTable }
Input the name of the input file name
input1.txt
Input the name of the ouput file name
output1.txt
```

输出文件 output1.txt:

```
output1.txt ×
编译原理 > output1.txt
1 -----operator precedence table-----
2 |      | + | * | ( | ) | i | $ |
3 -----
4 | + | > | < | < | > | < | > |
5 -----
6 | * | > | > | < | > | < | > |
7 -----
8 | ( | < | < | < | = | < |
9 -----
10 | ) | > | > |   | > |   | > |
11 -----
12 | i | > | > |   | > |   | > |
13 -----
14 | $ | < | < | < |   | < | = |
15 -----
```

(2) input2.txt (有二义性的文法)

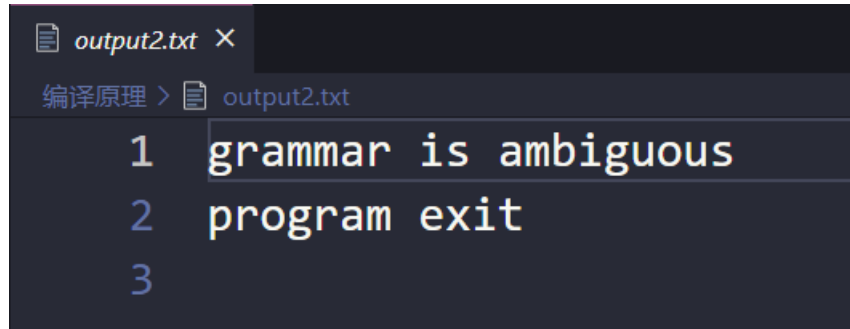
```
input2.txt ×
编译原理 > input2.txt
1 E -> E + E
2 E -> E * E
3 E -> (E)
4 E -> i
```

运行结果：

```
PS D:\VS code\C++\编译原理> cd "d:\VS code\C++\编译原理\" ; if ($?) { g++ analyTable.cpp -o analyTable } ; if ($?) { .\analyTable }
Input the name of the input file name
input2.txt
Input the name of the output file name
output2.txt

grammar is ambiguous
program exit
```

输出文件 output2.txt:



```
output2.txt X
编译原理 > output2.txt
1 grammar is ambiguous
2 program exit
3
```

总结：经过给定输入文法与其他一些简单文法的检测，能够正常地判断二义性与构造算符优先分析表

Tips:由于采用 UTF-8 编码,算符优先分析表在 txt 文档中直接打开可能会产生些许错位,但不影响结果,代码中注释了一些例如 firstvt、lastvt 等集合、记录的输出,有需要可以去注释查看分析过程。

(七) 实验心得

作为课程设计的第一部分内容,其实除了需要了解基础知识外没有什么重要的难点,但是暂时还只支持 char 类型作为终结符与非终结符,没有扩展功能,而且由于算符优先分析法本身的能力的局限,其实还是有相当多的文法无法检测,功能比较局限。

但是作为理论的实践的一部分,还是能收获到一些对编译语法分析的一些知识,从实践的角度了解一个编译器的工作流程,深有体会,收获很大。