**编译原理：LR（1）文法实现**

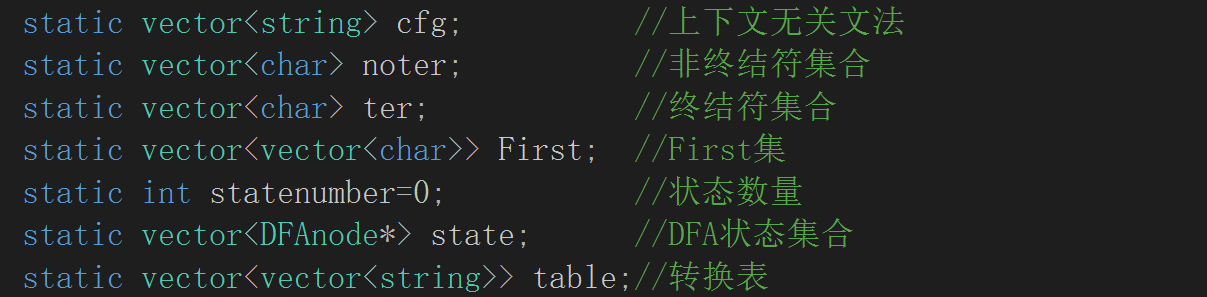
——111162 燕飞扬 20161003441

**一、程序输入规则**：

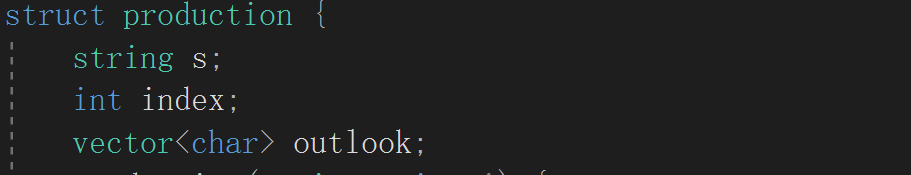
（1）不要输入带有|的产生式，将其分解为多条子产生式作为输入

（2）空字符用@代替输入

**二、数据结构：**

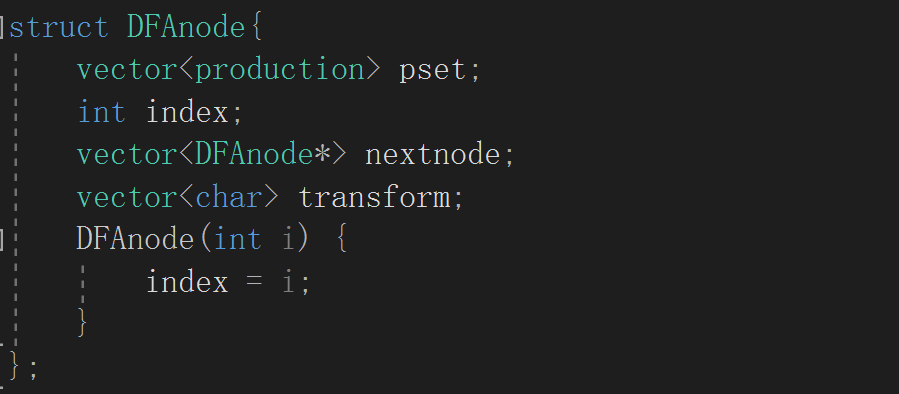


产生式结构体：



其中s表示产生式字符串，index表示产生式中.的位置，outlook表示展望字符集。

DFA状态结点结构体：



其中pset表示DFA结点中的产生式集合，index表示状态号，nextnode表示子结点集合，transform表示转换到子节点的转换字符。

**三、实现思路：**

1）找出产生式中的终结符集合和非终结符集合分别存入到数组ter和noter中。

2）计算非终结符的First集，提供给后面求展望字符用。

求First集思路（@表示空）：

如果产生式右部第一个字符为终结符，则将其计入左部first集

如果产生式右部第一个字符为非终结符执行以下步骤

求该非终结符的first集

将该非终结符的非空first集计入左部的first集

若存在@，则将指向产生式的指针右移

若不存在@，则停止遍历该产生式，进入下一个产生式

若已经到达产生式的最右部的非终结符，则将$加入左部的first集

处理数组中重复的first集中的终结符

3）求一个DFA状态内部产生式的扩展产生式，包括展望字符的求解。

一个新的DFA状态只有通过老状态产生式.移动产生新产生式，再通过这个新产生式扩展出这个新状态所有的产生式。

产生式扩展思路：

产生式扩展的过程是一个先进先出的过程，所以可以用队列来实现。

先将由老状态产生式通过.移动产生的新产生式入队列

While（队列不为空）{

取队头产生式front，出队列

判断front产生式中index所指向的（即.所指向的）字符a，

如果a是终结符，说明不需要扩展，continue

如果a是非终结符，说明需要扩展，将cfg中由a开始的产生式计算 展望字符后加入到状态集，然后入队列

}

计算展望字符的思路：

当产生新产生式的时候，需要计算其展望字符，

设新产生式的左部为A，遍历当前状态集合的每一条产生式p，

如果p当前index（即.）所指向的是A，

看A的下一个字符的First集，

1：将除空以外的First集加入到展望字符集；

如果First集不含空，结束此次遍历

如果First集中含有空，继续下一个字符重复1过程

如果已经遍历到p的最后，则将p的展望字符加入到展望字符集

4）求所有DFA状态并建立状态间的联系

由初始状态产生新状态的过程是一个广度遍历的过程，所以可以用队列实现

具体思路：

若S->a是开始产生式，则初始化其index为3，展望字符为$，

即S->.a ,$

将开始产生式扩展生成一个开始0状态，将0状态加入总状态集合state

将开始0状态入队列

While（队列不为空）{

取队头状态front，front出队列

For循环遍历front状态中每一条产生式p，

如果index已经指向p最后，结束此条产生式遍历

否则，设p当前index指向字符b

将p中index指向后移一位，即index++ 生成一个新产生式作为子 状态的开始产生式，然后扩展出产生式集合生成一个新状态

将新状态加入到front的子状态集中，转换字符为b

将新状态加入到总状态集合state中

}

5）建立转换表table

建立一张二维表，行数为状态总数statenumber，列数为终结符数量加终止符（$） 数量加非终结符数量，即ter.size+1+noter.size。表内元素初始化为空。

For循环遍历总状态数组state，i表示当前遍历状态号

For循环遍历当前状态的所有子状态，

转换字符为t，

如果t为终结符，则将s加上当前子状态编号加入到表中对应位置（i行 t列），表示动作为移进shift。

如果t为非终结符，则将g加上当前子状态编号加入到表中对应位置（I 行t列），表示goto。

For循环遍历当前状态的产生式集合，p表示当前遍历产生式

如果p中index指向产生式最后，说明需要规约，则将r加上此产生式的 编号填入表中对应位置，表示规约reduce

6）验证字符串，用栈分析

按照课上分析过程，建立2个栈，一个状态栈，一个符号栈，待验证字符串为s

先将状态栈初始化为0，符号栈初始化为$，

根据状态栈栈顶元素和s中当前指向字符s[i]在转换表中查找动作 ACTION

如果ACTION是s开头，则移进

如果ACTION是r开头，则规约，栈出栈规约产生式右部字符串的字 符数量，左部字符入符号栈，然后再看状态栈栈顶元素和符号栈栈顶 元素在table中查看goto状态，将goto状态入状态栈。

如果ACTION是ACC，则接受，结束

如果在查表时查到ACTION为空，则不接受，结束

**四、遇到的问题及解决方案**

问题一：开始时产生式扩展出新产生式的过程不知道怎么模拟

解决方案：产生式产生新产生式，新产生式又产生新产生式，是一个先进先出的过程， 可以用队列来模拟。

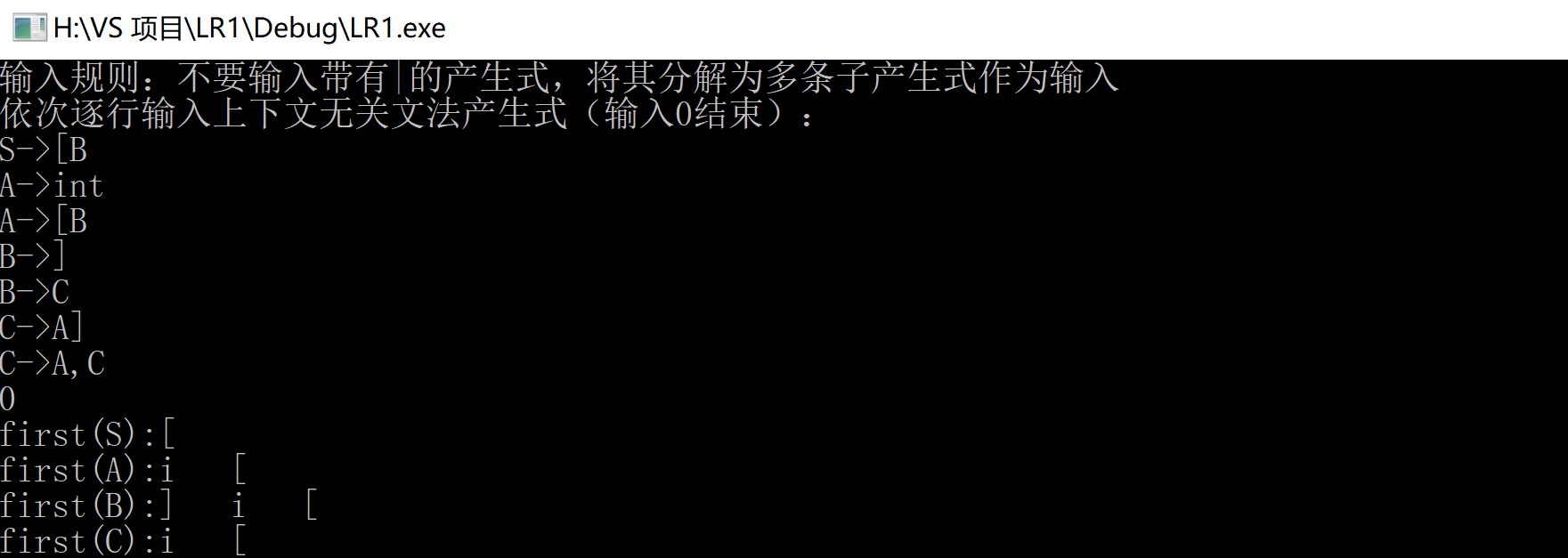
问题二：开始时，我没有将DFA结点存在一个总状态数组中，只存有一个头指针，遍历 DFA时，由于有环，进入死循环。

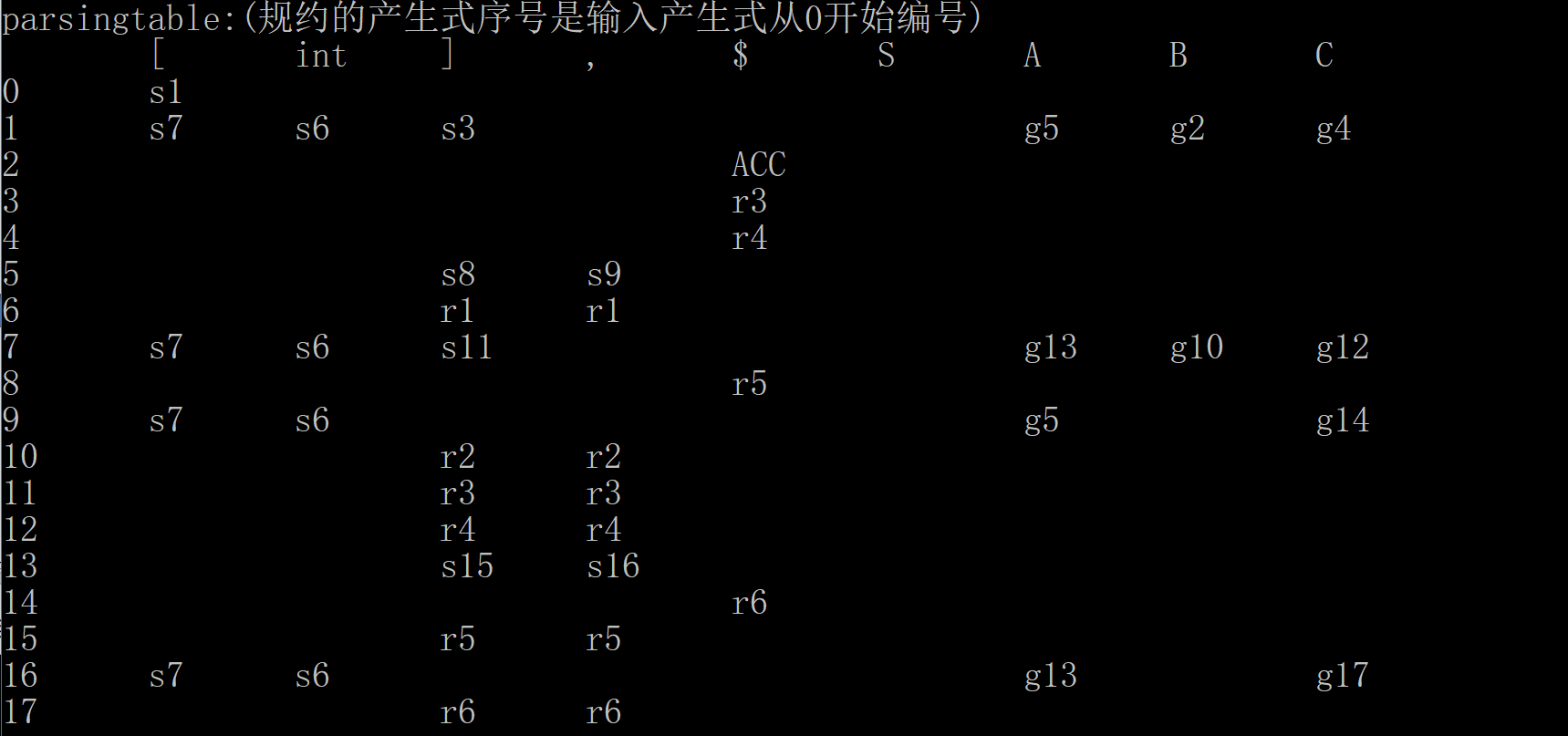
解决方案：将DFA结点存在一个总状态数组中，遍历DFA即遍历数组，就不会出现循环 的情况

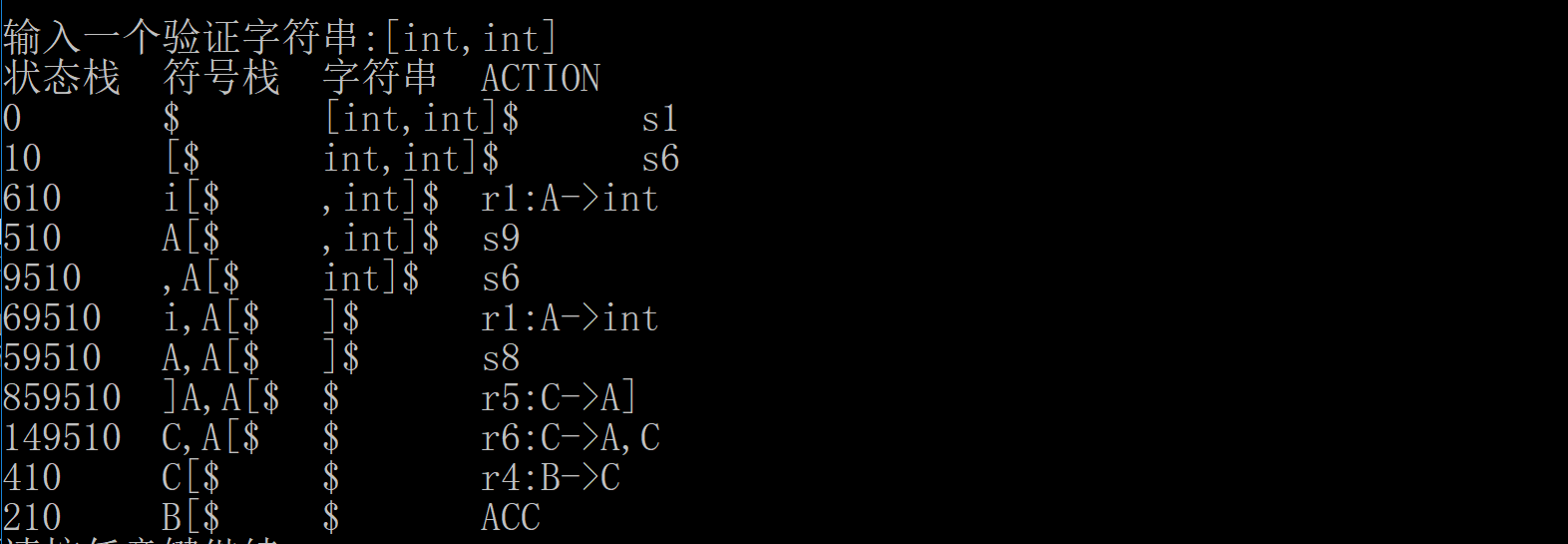
**五、总结**

通过这次实习，我对LR（1）文法的过程认识更加清晰。每一步该做什么，其中有什么需要注意，我都有了一个清晰的认知。其中，展望字符的求解一直是令我疑惑的，通过这次实习，展望字符求法已经完全掌握。栈的分析过程之前也是一知半解，自己实现后，已经完全清楚了它的过程，移进shift，goto，规约ruduce。总之，这次实习对我提升很大。

**六、测试结果截图**

****

****

****