



東南大學  
SOUTHEAST UNIVERSITY

# 编译原理实验

## 报告二

实验名称: Syntax Parser

学生姓名: 柳沿河

学生学号: 71117230

东南大学计算机科学与工程学院、软件学院

School of Computer Science & Engineering College of Software Engineering

Southeast University

二 0 一 九 年 十 二 月

## 一、 目的

- 了解语法分析的原理
- 掌握通过文法构造 LR 项集簇的方法
- 掌握通过 LR 项集簇编写语法分析程序的方法

## 二、 内容描述

编写一个语法分析程序，要求如下：

1. 输入：字符流、CFG
2. 输出
  - a) 自顶向下分析：推导序列
  - b) 自底向上分析：规约序列
3. 句子类别自定义
4. 考虑错误处理

任选一下三种种实现方式实现：

1. LL(1)分析
2. LR(1)分析
3. 基于 Yacc

## 三、 方法

本次实验采用第二种实现方式，即通过将 CFG 构造相应的 LR(1)分析表来编写语法分析程序。

## 四、 假设

- 分析语言：类 C 语言语法

- CFG:

$S \rightarrow id = E ; S$

$S \rightarrow id = E ;$

$S \rightarrow E ; S$

$S \rightarrow E ;$

$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow id \mid number \mid ( E )$

## 五、 相关 FA 描述

根据 CFG 构造的 LR(1)项集见: [LR\(1\)项集.pdf](#)

根据 LR1 项集构造的分析预测表如下:

state	Action											Goto			
	id	number	+	-	*	/	=	(	)	:	\$	S	E	T	F
0	S7	S1						S6				2	4	3	5
1			R12	R12	R12	R12				R12					
2											Accept				
3			R7	R7	S8	S9				R7					
4			S11	S12						S10					
5			R10	R10	R10	R10				R10					
6	S18	S13						S17					15	14	16
7			R11	R11	R11	R11	S19			R11					
8	S21	S1						S6							20
9	S21	S1						S6							22
10	S7	S1						S6			R4	23	4	3	5
11	S21	S1						S6						24	5
12	S21	S1						S6						25	5
13			R12	R12	R12	R12			R12						
14			R7	R7	S26	S27			R7						
15			S29	S30					S28						
16			R10	R10	R10	R10			R10						
17	S18	S13						S17					31	14	16
18			R11	R11	R11	R11			R11						
19	S21	S1						S6					32	3	5
20			R8	R8	R8	R8				R8					
21			R11	R11	R11	R11				R11					
22			R9	R9	R9	R9				R9					
23											R6				
24			R5	R5	S8	S9				R5					
25			R6	R6	S8	S9				R6					
26	S18	S13						S17						35	16
27	S18	S13						S17						36	16
28			R13	R13	R13	R13				R13					
29	S18	S13													
30	S18	S13													
31			S29	S30					S37						
32			S11	S12						S38					
33			R8	R8	R8	R8			R8						
34			R9	R9	R9	R9			R9						
35			R5	R5	S26	S27			R5						
36			R6	R6	S26	S27			R6			39	4	3	5
37			R13	R13	R13	R13			R13						
38	S7	S1						S6			R2				
39											R1				

六、 重要的数据结构描述

- LR1PPT 类：预测分析表对象
  - 成员

- ◆ `private Map<String, Action[]> actions`: 存放动作表的成员对象

- ◆ `private Map<String, int[]> gotos`: 存放转换表的成员对象

- 方法

- ◆ `private void buildActionTable()`: 构造动作表

- ◆ `private void buildGotoTable()`: 构造转换表

- ◆ `public Action[] ACTION(String terminalSymbol)`: 获取读入的终结符对应的所有动作

- ◆ `public int[] GOTO(String nonterminalSymbol)`: 获取读入的非终结符对应的所有状态转换

- Production 类: 产生式对象

- 成员

- ◆ `private String left`: 存放产生式左部

- ◆ `private String[] right`: 存放产生式右部

- 方法

- ◆ `get` 方法: 获取产生式的左部和右部

- ◆ 重载 `toString` 方法: 用于向文件中输出规约

- Action 类: 动作对象

- 成员

- ◆ `ActionType type`: 动作类型, 有 `ActionType.Shift`, `ActionType.Reduce`, `ActionType.Accept` 三种类型

- ◆ `int stateRule`: 存放要移进到的状态或规约用的产生式编号

- StackItem 类: 分析栈中元素对象

## ■ 成员

- ◆ StackItemType type: 类型, 有 StackItemType.Token, StackItemType.State 两种
- ◆ String value: 若为 token, 则存放对应的字面值
- ◆ int state: 若为状态, 则存放对应的状态编号

## 七、 核心算法描述

算法: 类 C 语言语法分析算法 parse

输入: 类 C 词法分析得到的 token 序列, CFG, PPT

输出: 归约序列

执行过程:

1. 构造用于文件写入的 BufferedWriter 对象、token 序列对象的迭代器对象, 并初始化分析栈——添加状态为 0 的栈元素, 通过迭代器获取当前读入的 token 对象, 记为 curToken, 执行下一步
2. 根据 token 的字面值 and 栈顶的状态, 获取 curToken 对应的动作。若动作为 null, 则 token 不合法, 报错; 若动作为 Accept, 则提示语法分析成功, 结束分析; 否则执行下一步
3. 若动作为 Shift, 则用 curToken 构造新的 StackItem 对象并将其压入栈中, 再用该 Shift 动作 $S_i$ 对应的要移到的状态 $i$ 构造 StackItem 对象并将其压入栈中。最后用迭代器读取下一个 token 赋给 curToken, 返回 2; 否则执行下一步
4. 若动作为 Reduce, 则用该 Reduce 动作 $R_j$ 对应的产生式编号 $j$ 在产生式集合 productionSet 中获取对应的产生式 $p_j$ , 将其写入归约文件中。之后, 从最右端开始, 对于 $p_j$ 的右部的每一个符号 $s_k$ , 先将栈顶的状态项弹出, 然后将 $s_k$ 与栈顶元素的字面值进行对比, 若两者相等则弹出, 进行下一次循环; 否则报错。最后用 $p_j$ 左部的符号 $L$ 和当前栈顶的状态 $S$ 在 Goto 表中获得的下一个状态 $S'$ , 用 $L$ 构造 StackItem 对象并压入栈中, 再用 $S'$ 构造 StackItem 对象并压入栈中。返回 2

# 八、 用例

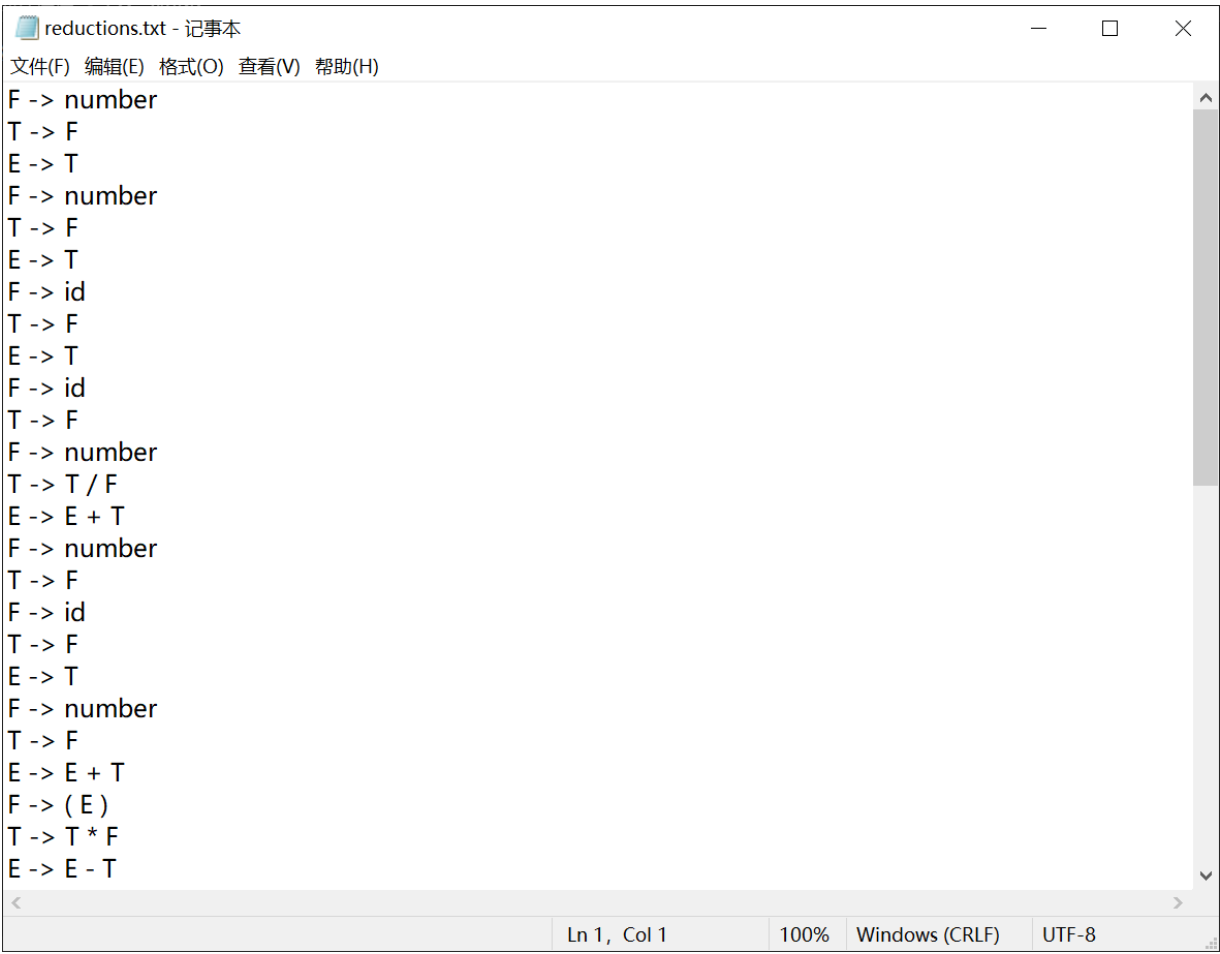
测试用例如下：

```
x = 10;
y = 100;
t = x + y / 3 - 2 * ( x + 6 );
t + 3 - x*5 / (y-1);
```

运行程序方法：

- java     SyntaxParser     test.txt     tokens.txt     reductions.txt
- IDE 中运行测试类 LR1ParserTest

运行结果如下：



## 九、 发生的问题和相关的解决方案

- 无法顺利结束语法分析：在从词法分析器获取的 token 序列最后添加字面值为\$的 token 对象
- 分析算法可读性较差，结构化不高：构造 Action 类和 StackItem 类，并用枚举变量表示动作类型和栈元素类型

## 十、 感受和评论

刚开始进行本次实验时，尝试采用了比较简单的 LL(1)分析法，但为了熟悉更为强大的 LR(1)分析法，最终还是采用了 LR(1)分析法。该方法分析能力强大，符合机器的运行方式，但同时代价非常高，需要构造非常多的 LR(1)项目集，人工实现方式几乎无法分析稍微复杂一点的 CFG。经过权衡，最终去除了一些含有关键字的句子，保留了常用的四则运算和变量的定义语句，即使如此还是有 40 个 LR(1)项集。构造过程要求十分的仔细，稍有不慎就会造成错误。尽管最终完成了这次试验，但花费的时间和精力较大，在以后的学习实践中会尝试 Yacc 来构造分析表和分析器。