

3.5 研究性学习专题 5 基于 SLR(1)分析法的语法制导翻译及中间代码生成程序设计原理与实现

1. 理论传授

SLR(1) 分析法，语法指导的基本概念，生成四元式的理论和方法。

2. 目标任务

本次实验实现一个简单的编程语言的语法制导翻译和中间代码生成。该编程语言由一个名为 f 的“函数”构成，函数名固定为 f ，函数需要至少 1 个变量作为参数，变量名只由 1 个小写字母组成，例如 a 、 b 、 c 等都是可以作为参数的变量名，但是 f 不可以。函数通过参数列表说明所需要的参数，参数列表紧跟着函数名用方括号包裹列出，如果有多于 1 个的参数，中间使用逗号分开。参数列表后紧跟着一个等号，等号右边是作为函数参数的变量构成的计算表达式，表达式可以有 $+$ $-$ $*$ $/$ 四则运算，括号 $()$ ，其运算规则和 C 语言的运算规则一致。但是如果要表达取负数，则需要至少 1 层原括号包裹。举个例子：

```
f[x]=x  
f[i]=i+i*i-i/i  
f[x,y,z]=x*y+z-x/(z+x)
```

分别是使用该语言写成的三个合法的程序，而

```
f[x]=y  
f[x,y,z]=x++y  
f[x,y]=xy  
f[A]=A
```

是不合法的程序，第一个程序里表达式中出现了没有在参数列表中声明的变量 y ，而第二个、第三个和第四个程序则是出现了语法错误。

于是可以定义出这样的语法：

$V_t = \{+, -, *, /, ,, (,), [,], f, X\}$ 其中 X 表示除了 f 之外的任意小写字母

$V_n = \{S, A, E, T, F\},$

$G[S]:$

$S \rightarrow f[A] = E$

$A \rightarrow$ 请你补充 A 的定义，使其能够表示该编程语言的参数列表的内容（1 个或多个参数，如果有多个参数中间用逗号, 分开），并且符合 SLR(1) 文法。

$E \rightarrow E + T | E - T | T$

$T \rightarrow T * F | T / F | F$

$F \rightarrow (E) | X$

【设计要求】 1. 补充上面的文法，如果有需要你可以引入新的非终结符号，只要满足要求即可，在实验报告中写出你完善后的文法。 2. 构造该文法的 SLR(1) 分析表。 3. 设计语法制导翻译过程，可以给出每一个产生式对应的语义动作（如果有）。 4. 测试例子至少应当测试“目标任务”中的所有测试例，对于合法的程序给出程序执行结果的四元式序列，对于错误的程序应该能识别出错

误，最好能给出错误信息。5. 四元式可以参考下面的设计，也可以自行设计。

定义函数：

(DEF_FUNC, 函数名,,)

定义 1 个参数：

(DEF_ARG, 参数名,,)

并且，所有的 DEF_ARG 一定会紧跟在 DEF_FUNC 后面

加法：

(ADD, 源操作数 1, 源操作数 2, 目标操作数)

减法：

(SUB, 源操作数 1, 源操作数 2, 目标操作数)

乘法：

(MUL, 源操作数 1, 源操作数 2, 目标操作数)

除法：

(DIV, 源操作数 1, 源操作数 2, 目标操作数)

赋值：

(ASSIGN, 源操作数,, 目标操作数)

结束函数定义：

(END_FUNC,,)

函数的返回值在可以四元式中用特殊的变量 R 表示, 参数之外的临时变量可以用 T1, T2, T3...等表示。

例如对于程序 $f[x, y] = x * (x + y)$ 可以翻译为（这只是一种可能，也可以翻译为其它的等价的四元式序列）：

(DEF_FUNC, f,,)

(DEF_ARG, x,,)

(DEF_ARG, y,,)

(ADD, x, y, T1)

(MUL, x, T1, T2)

(ASSIGN, T2,, R)

(END_FUNC,,)

本次实验不要求对中间代码（四元式）优化，有兴趣的同学可以尝试一下。（例如在上面的例子中，MUL x T1 的结果可以直接给 R，而不需要 T2 中转）

参考测试例（仅供参考，助教的程序跑的结果）：

输入程序：

$f[x] = x$

输出四元式：

(DEF_FUNC,f,,)

```
(DEF_ARG,x,,)
(ASSIGN,x,,R)
(END_FUNC,,,)

```

输入程序:

$f[i]=i+i*i-i/i$

输出四元式:

```
(DEF_FUNC,f,,)
(DEF_ARG,i,,)
(MUL,i,i,T1)
(ADD,i,T1,T2)
(DIV,i,i,T3)
(SUB,T2,T3,T4)
(ASSIGN,T4,,R)
(END_FUNC,,,)

```

输入程序:

$f[x,y,z]=x*y+z-x/(z+x)$

输出四元式:

```
(DEF_FUNC,f,,)
(DEF_ARG,x,,)
(DEF_ARG,y,,)
(DEF_ARG,z,,)
(MUL,x,y,T1)
(ADD,T1,z,T2)
(ADD,z,x,T3)
(DIV,x,T3,T4)
(SUB,T2,T4,T5)
(ASSIGN,T5,,R)
(END_FUNC,,,)

```

输入程序:

$f[x]=y$

报错:

Error: Used undeclared variable y

输入程序:

$f[x,y]=x++y$

报错:

Error: In line 1, at position 1: Could not analyze the grammar.

Possible grammar rule:

$E \rightarrow E \text{ TK_ADD } T$

Near +

输入程序:

f[x,y]=xy

报错:

Error: In line 1, at position 1: Could not analyze the grammar.

Possible grammar rule:

F -> TK_X

Near y

输入程序:

f[A]=A

报错:

Error: In line 1 at position 1: Lexer error occurs after token [, the following token could not be recognized to be any type of token.