



兰州大学
LANZHOU UNIVERSITY

语义分析器

授课老师: 燕昊
课程: 编译原理实验
班级: 计算机基地班
姓名: 王子豪
学号: 320200945691

June 5, 2023

目录

1	源语言及目标语言的介绍	2
1.1	源语言介绍	2
1.2	目标语言介绍	2
2	关键算法	2
2.1	运算表达式	3
2.2	跳转语句	3

1 源语言及目标语言的介绍

1.1 源语言介绍

- 源语言文法定义如下：

```

1  PROG      →   { DECLS STMTS }
2
3  DECLS     →   DECLS DECL | empty
4  DECL      →   int NAMES; | bool NAMES;
5  NAMES     →   NAMES , NAME | NAME
6  NAME      →   id
7
8  STMTS     →   STMTS STMT | STMT
9  STMT      →   id = EXPR; | id := BOOL;
10 STMT      →   if id then STMT
11 STMT      →   if id then STMT else STMT
12 STMT      →   while id do STMT
13 STMT      →   { STMTS STMT }
14 STMT      →   read id;
15 STMT      →   write id;

```

- 其中算术表达式使用运算符优先级，并没有严格区分 *bool* 形与 *int* 形。

1.2 目标语言介绍

目标语言使用四元组的形式

$$(result, num1, operator, num2)$$

- result* 表示结果存放的位置
- num1* 表示第一个运算数
- operator* 表示操作类型
- num2* 表示第二个运算符

一些特殊操作四元表达式形式如下：

- $(a, none, read, none) \rightarrow read\ a;$
- $(cond1, =, t1, none) \rightarrow cond1 = t1;$
- $(15, cond1, je, 0) \rightarrow \text{如果 } cond1 == 0, \text{ 跳转到 } 15。$
- $(4, none, jmp, none) \rightarrow \text{立即跳转到 } 4。$
- $(t1, a, >=, b) \rightarrow \text{若 } a \geq b, \text{ 则 } t1=1 \text{ 或 } t1 \text{ 为真。}$

2 关键算法

根据源语言的特性，关键算法主要体现在运算表达式及跳转语句上（while, if）

2.1 运算表达式

在中缀转后缀的过程中直接进行运算，由于这些运算都是中间运算且为基本运算，直接输出每次运算对应的四元表达式即可。每次运算的结果都存储在 t_n 中，并将 t_n 入栈。

代码体现如下：

```
1 string num2 = st2.top(); st2.pop();
2 string num1 = st2.top(); st2.pop();
3 string res = "t" + to_string(++cnt_st2);
4 st2.push(res);
5 Node temp3 = op.top(); op.pop();
6 quat.push_back({res, num1, temp3.val, num2});
7 pfix.push_back(temp3);
```

2.2 跳转语句

采用链表回填的方法，由于每个判断条件都为 id ，所以实现起来并不复杂。

每次进入 $while$ 或 if 语句时，开两个数组分别记录所有需要跳转到真假出口的位置，对于 $while$ 语句还需记录入口编号。读到真假出口时回填两个数组即可。

代码体现如下：

```
1 vector<int> T, F; // 真假出口
2 int ent = quat.size(); // 入口编号
3 ...
4 ...
5 string res, num1(Val), op("je"), num2("0"); // 等于0跳转到假出口
6 quat.push_back({res, num1, op, num2}); // 此时的res为空值，等待回填
7 F.push_back(quat.size() - 1); // 记录需要回填的编号
8 ...
9 ...
10 开始回填
11 for (int i = 0; i < F.size(); ++i)
12 {
13     quat[F[i]].result = to_string(num);
14 }
```

3 测试结果

3.1 样例 1

- input1

```
1 //program 1: add two numbers.
2 {
3     int a, b, c ;
4     a = 1;
5     b = 2;
6     c = a + b ;
7 }
```

- output1

```

0: (a, =, 1, none)
1: (b, =, 2, none)
2: (t1, a, +, b)
3: (c, =, t1, none)

```

3.2 样例 2

- input2

```

1 //program 2: read in and add two numbers,
2 //then print the result .
3 {
4     int a, b, c ;
5     read a ;
6     read b ;
7     c = a + b ;
8     write c ;
9 }

```

- output2

```

0: (a, none, read, none)
1: (b, none, read, none)
2: (t1, a, +, b)
3: (c, =, t1, none)
4: (c, none, write, none)

```

3.3 样例 3

- input3

```

1 /*program 3: add numbers from 1 to 100
2 *and print the result.
3 */
4 {
5     int a , sum ;
6     bool b ;
7     a = 1 ;
8     sum = 0 ;
9     b := a <= 100 ;
10    while b do
11    {
12        sum = sum + a ;
13        a = a + 1 ;
14        b := a <= 100 ;
15    }
16    write sum ;
17 }

```

- output3

```
0: (a, =, 1, none)
1: (sum, =, 0, none)
2: (t1, a, <=, 100)
3: (b, =, t1, none)
4: (12, b, je, 0)
5: (t1, sum, +, a)
6: (sum, =, t1, none)
7: (t1, a, +, 1)
8: (a, =, t1, none)
9: (t1, a, <=, 100)
10: (b, =, t1, none)
11: (4, none, jmp, none)
12: (sum, none, write, none)
```

3.4 样例 4

- input4

```
1 //program 4: input 3 numbers, find the largest
2 //one, and output it .
3 {
4     int a,b,c;
5     int lg;
6     bool cond;
7
8     read a;      read b;      read c;
9
10    cond := a > b ;
11    if cond then lg = a ;
12    else lg = b ;
13
14    cond := lg < c ;
15    if cond then lg = c ;
16
17    write lg ;
18 }
```

- output4

```

0: (a, none, read, none)
1: (b, none, read, none)
2: (c, none, read, none)
3: (t1, a, >, b)
4: (cond, =, t1, none)
5: (8, cond, je, 0)
6: (lg, =, a, none)
7: (9, none, jmp, none)
8: (lg, =, b, none)
9: (t1, lg, <, c)
10: (cond, =, t1, none)
11: (14, cond, je, 0)
12: (lg, =, c, none)
13: (14, none, jmp, none)
14: (lg, none, write, none)

```

3.5 样例 5

- input5

```

1  /* program 5: find all numbers which is
2  * divisible by 3 between 1 and 12 .
3  */
4  {
5      int number, res ;
6      bool cond1, cond2 ;
7
8      number = 1 ;
9      cond1 := number <= 12 ;
10     while cond1 do
11     {
12         res = number - ( number / 3 ) * 3 ;
13         cond2 := res == 0 ;
14         if cond2 then write number ;
15         number = number + 1 ;
16         cond1 := number <= 12 ;
17     }
18 }

```

- output5

```
0: (number, =, 1, none)
1: (t1, number, <=, 12)
2: (cond1, =, t1, none)
3: (18, cond1, je, 0)
4: (t1, number, /, 3)
5: (t2, t1, *, 3)
6: (t3, number, -, t2)
7: (res, =, t3, none)
8: (t1, res, ==, 0)
9: (cond2, =, t1, none)
10: (13, cond2, je, 0)
11: (number, none, write, none)
12: (13, none, jmp, none)
13: (t1, number, +, 1)
14: (number, =, t1, none)
15: (t1, number, <=, 12)
16: (cond1, =, t1, none)
17: (3, none, jmp, none)
```

3.6 样例 6

- input6

```
1 //program 6: no decls in program .
2 //NO errors in Syntax Checking.
3 //Undeclared id errors should be reported in Symantic Checking.
4 {
5     a = -1 ;
6     b = a - 1;
7     wirte a;
8 }
```

- output6


```
欢迎使用词法分析器，请选择输入方式：
1:从文件中输入  2:从终端中输入  3:退出
1
<{, delimiter>
<a, identifier1>
<=, op>
<-1, integer>
<;, delimiter>
<b, identifier2>
<=, op>
<a, identifier1>
<-, op>
<1, integer>
<;, delimiter>
<wirte, identifier3>
<a, identifier1>
<;, delimiter>
<}, delimiter>
未声明标识符a
PS E:\C++\compilation_principle> █
```

3.7 样例 7

- input7

```
1 //program 7: read in two numbers and do some calculations.
2 {
3     int a, b, c ;
4     read a ;
5     read b ;
6
7     //Test Expr().
8     a = a + 1 + 2 * 3 * 4 ;
9     b = b * 8 / 2 / 4 ;
10    c = ( a + b ) * ( a - b ) ;
11
12    write c ;
13 }
```

- output7

```

0: (a, none, read, none)
1: (b, none, read, none)
2: (t1, a, +, 1)
3: (t2, 2, *, 3)
4: (t3, t2, *, 4)
5: (t4, t1, +, t3)
6: (a, =, t4, none)
7: (t1, b, *, 8)
8: (t2, t1, /, 2)
9: (t3, t2, /, 4)
10: (b, =, t3, none)
11: (t1, a, +, b)
12: (t2, a, -, b)
13: (t3, t1, *, t2)
14: (c, =, t3, none)
15: (c, none, write, none)

```

3.8 样例 8

- input8

```

1  /* program8: input three numbers ,output the largest one.
2  * Test Statement : If_then and If_Then_Else .
3  */
4  {
5      int a,b,c ;
6      bool cond1,cond2,cond3;
7
8      read a; read b; read c;
9
10     cond1 := a >= b ;
11     cond2 := a >= c ;
12     cond3 := b >= c ;
13
14     if cond1 then
15         if cond2 then write a ;
16         else write c ;
17     if cond1 then
18         if cond3 then write b ;
19         else write c ;
20 }

```

- output8

```
0: (a, none, read, none)
1: (b, none, read, none)
2: (c, none, read, none)
3: (t1, a, >=, b)
4: (cond1, =, t1, none)
5: (t1, a, >=, c)
6: (cond2, =, t1, none)
7: (t1, b, >=, c)
8: (cond3, =, t1, none)
9: (15, cond1, je, 0)
10: (13, cond2, je, 0)
11: (a, none, write, none)
12: (14, none, jmp, none)
13: (c, none, write, none)
14: (15, none, jmp, none)
15: (21, cond1, je, 0)
16: (19, cond3, je, 0)
17: (b, none, write, none)
18: (20, none, jmp, none)
19: (c, none, write, none)
20: (21, none, jmp, none)
```