Decaf 实验四

4.中间代码生成

Decaf 实验四

- 三地址码
  - The most basic instruction of three address code has the general form x=y op z
  - x,y,z are names, constants or compiler-generated temporary names
  - op stands for any arithmetic or logical operator, such as + ,'and'
  - "Three-address code" comes from this form of instruction, in general each of x,y and z represents an address in memory

Decaf 实验四

- 三地址码
  - 赋值语句

```
a   =   5   +   2   *   b;
```

  - 三地址码

```
_t1  =  2  * b;
_t2 = 5  +  _t1;
a  =  _t2;
```

Decaf 实验四

- if语句

```
{
    int x;int y; int z;
    if( x < y )
     z = x;
    else
     z = y;
    z = z + z;
}
```

- 三地址码

```
L1:     iffalse x < y goto L5
L4:     z = x
        goto L3
L5:     z = y
L3:     z = z + z
L2:
```

Decaf 实验四

- Decaf语言中间代码生成
- 主要在包inter中每个类的gen（对于while等控制语句）或jumping函数（对于控制语句中的条件表达式）
  - 将while、if等控制语句及其条件表达式翻译成if x goto L1等形式的三地址码

Decaf 实验四

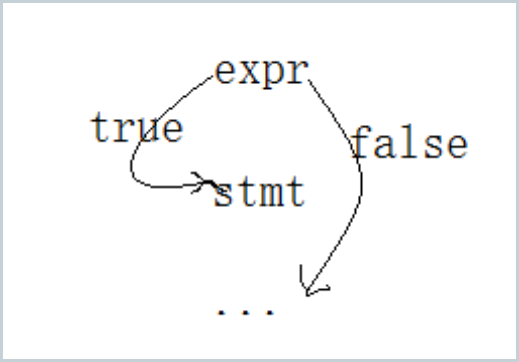
- 将控制语句中布尔表达式条件翻译成if x goto L1等形式三地址码的实现（inter包中Expr类的emitjumps函数）

```
public void emitjumps(String test, int t, int f){
if(t!=0&&f!=0){   //t: true   f: false
emit("if "+test+" goto L"+t);
emit("goto L"+f);
}
else if(t!=0) emit("if "+test+" goto L"+t);
else if(f!=0) emit("iffalse "+test+" goto L"+f);
else ;
}
```

Decaf 实验四

- if (expr) { stmt; }
- 其三地址码的实现对应inter包中If类的gen函数

```
public void gen(int b, int a) {  //b: begin   a:  after
     int label = newlabel(); // label for the code for stmt
     expr.jumping(0, a);     // fall through on true, goto a on false
     emitlabel(label); stmt.gen(label, a);
  }
```



Semantic Rules

E.true=newlabel;

E.false=S.next;

S.code=E.code || Label E.true || S1.code

Decaf 实验四

- 例子

```
{
    int a;int b; int c; int min;
    if(a<b)
    {
        min=a;
    }
    while(min+a<c)
    {
        a=a+1;
    }
}
```

- 例子-输出

```
L1:     iffalse a < b goto L3
L4:     min = a
L3:     t1 = min + a
        iffalse t1 < c goto L2
L5:     a = a + 1
        goto L3
L2:
```

Decaf 实验四

- 实验内容：
  - 1.参考第7章节中中间代码生成相关知识理解读懂提供的Decaf语言中间代码生成源代码
  - 2.提供的源码中没有实现对for语句的中间代码生成，参考源码中inter包中的while、 if等控制语句的中间代码生成，添加for语句的中间代码生成处理（主要是完善inter包中for类的gen函数），使得可为包含for语句的Decaf源程序生成中间代码
  - 3.在第三次语法分析实验报告的基础上继续添加撰写完成中间代码生成的实验报告，每位同学在2025年06月22日前在教学在线提交完整的实验报告与源代码。

for语句中间代码生成的参考属性文法

- S->for (stmt1; expr; stmt2) { stmt3 }

```
for(i = 100; i > 5+a/b; i=i+1)
{
    if c<d then
        x=y+z
}
write x;
```

```
        i = 100
L2:  _t1=a/b
     _t2=5+_t1
  if_false i>_t2 goto L1
  if_false c<d goto L3
     _t3=y+z
     x=_t3
L3: _t4 =i+1
     i= _t4
     goto L2
L1:  write x
```

```
stmt1.next = newlabel()
stmt3.next = newlabel()
  expr.true  = 0
  expr.false = S.next
S.code =      stmt1.code||
stmt1.next: expr.code||
                stmt3.code||
stmt3.next: stmt2.code||
                goto stmt1.next
```

Decaf 实验四

- 包含for语句的decaf代码例子：

```
{
    int a;int b; int c; int min; int i;
    if(a<b)
    {
        min=a;
    }
    for(i=1;i<100;i=i-1)
    {
        a=a+1;
    }
}
```