**编译实习面测**

错误样例分析:

一、Minijava-piglet

Test15.java 执行错误：空指针

1. 比对java和pg代码，发现错误位置：

Java:

if (i < 0)

k = 0;

else {

System.out.println(k);

...

}

Piglet:

CJUMP LT TEMP 34 0 Label0

HLOAD TEMP 35 TEMP 0 8

MOVE TEMP 35 0

HSTORE TEMP 0 8 TEMP 35

JUMP Label1

Label0

PRINT **TEMP 35**

1. 分析错误原因：

在生成的pg代码中，在iftrue语句块中，类的字段k与TEMP35绑定了；但是，在else分支中，TEMP35未初始化，所以print会产生空指针报错。

1. 解决办法：

每次遇到类的字段，不管有没有绑定，都生成一个“HLOAD”指令。

在public MType visit(Identifier n, MType argu) 方法中将相关语句修改为：

if (((MVar) id).isClassVar == true) {

MClass varClass = ((MMethod)argu).methodClass;

String temp;

String varOffset = Integer.toString(4 \* (varClass.getVarIndex(name) + 1));

if(currentFunctionUsedVarList.containsKey(name) == false) {

temp = getTemp();

currentFunctionUsedVarList.put(name, temp);

}

else {

temp = currentFunctionUsedVarList.get(name);

}

pigletCode += "\nBEGIN\n";

pigletCode += "HLOAD " + temp + " TEMP 0 " + varOffset + "\n";

pigletCode += "RETURN " + temp + "\n";

pigletCode += "END\n";

}

（之前只考虑了contains==false的情况）

修改后生成的pg代码：

PRINT

BEGIN

HLOAD TEMP 42 TEMP 0 8

RETURN TEMP 42

END

二、Piglet-spiglet

Test20.pg 输出结果错误

Test22.pg

（一）比对pg和spg代码，发现错误位置：

Piglet:

PLUS

BEGIN HLOAD TEMP 100 TEMP 0 8 RETURN TEMP 100 END

BEGIN HLOAD TEMP 100 TEMP 0 12 RETURN TEMP 100 END

Spiglet:

HLOAD **TEMP 100** TEMP 0 8

HLOAD **TEMP 100** TEMP 0 12

MOVE TEMP 130 PLUS **TEMP 100 TEMP 100**

（二）分析错误原因：

在piglet代码中，不同的begin-return-end语句可以返回同一个temp，对应不同的值，而spiglet不同。

（三）解决办法：

添加一个move语句，将return对象返回到一个新的temp上，避免重复。

/\* f0 -> "BEGIN" f1 -> StmtList() f2 -> "RETURN" f3 -> Exp() f4 -> "END"\*/

public String visit(StmtExp n) {

n.f1.accept(this);

String exp = n.f3.accept(this);

String temp = getTemp();//add

spigletCode += "MOVE " + temp + " " + exp + "\n";//add

return temp;//return exp;

}

1. 进阶测试分析
2. 没有实现短路与

解决办法：应该把简单的一行“LT ”代码，改为先判断左边的值是否为0，如果为0，则直接进入到else语句中；不然再判断右边的值是否0.

1. 加载溢出的参数到寄存器时，分配代码出现了bug。

有一行判断v1寄存器是否使用过的代码被误删了，在特殊情况下运行到该段代码会出现计算错误（将不同的溢出变量加载到了同一个v0寄存器中，导致了v1的值计算错误，进而导致下面在计算数组下标时计算错误，出现了访问栈溢出现象）。

解决办法：恢复被误删的代码即可。