Ziyan Wang

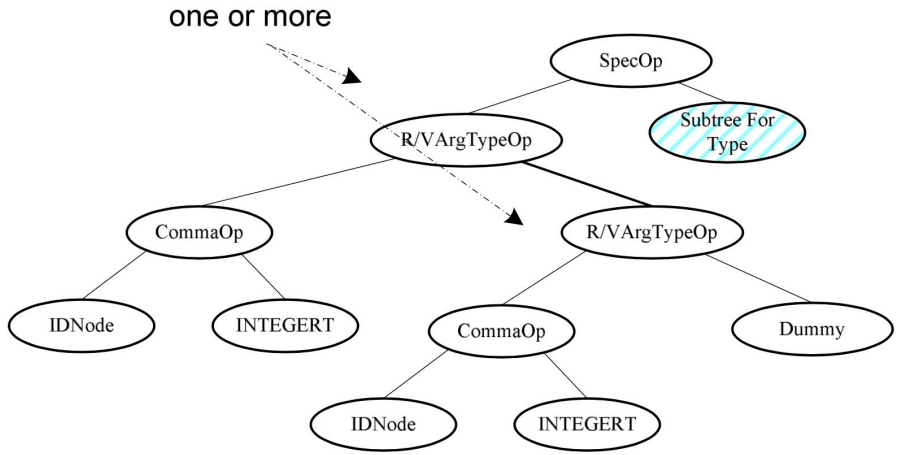
Project 4 Q&A

2021/06/10

Q: 生成的代码需要和标准输出一样吗?

- A: 不需要，只要运行结果正确即可

Q: 如何计算内存地址?



- A: 以函数参数为例

```
1   int handleFunctionParams(tree syntaxTree, int currentOffset, int currentParamCount) {
2       if (syntaxTree == NullExp()) {
3           return currentOffset;
4       }
5       if (RightChild(syntaxTree) != NullExp()) {
6           currentOffset = handleFunctionParams(RightChild(syntaxTree), currentOffset, currentParamCount + 1);
7       }
8       if (NodeOp(syntaxTree) == RArgTypeOp) {
9           SetAttr(IntVal(LeftChild(LeftChild((syntaxTree)))), OFFSET_ATTR, currentOffset);
10          fprintf(outputFile, "sw $a%d 0($sp)\n", currentParamCount);
11          fprintf(outputFile, "addi $sp $sp -4\n");
12          return currentOffset - 4;
13      } else {
14          // NodeOp(syntaxTree) == VArgTypeOp
15          // ...
16      }
17  }
18
19  handleFunctionParams(LeftChild(RightChild(LeftChild(syntaxTree))), 0, 0);
```

Q: 什么时候用Branch，什么时候用Jump?

- A: Branch是条件跳转，Jump是无条件跳转

4.4.3 Branch and Jump Instructions

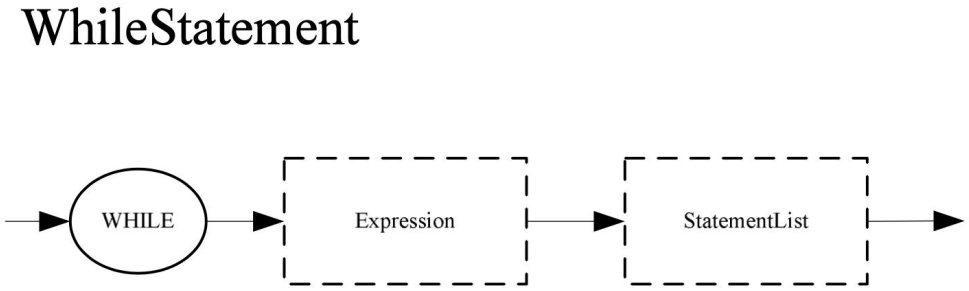
4.4.3.1 Branch

Op	Operands	Description
`b`	*lab*	Unconditional branch to *lab*.
`beq`	*src1, src2, lab*	Branch to *lab* if *src1* $\equiv$ *src2*.
`bne`	*src1, src2, lab*	Branch to *lab* if *src1* $\neq$ *src2*.

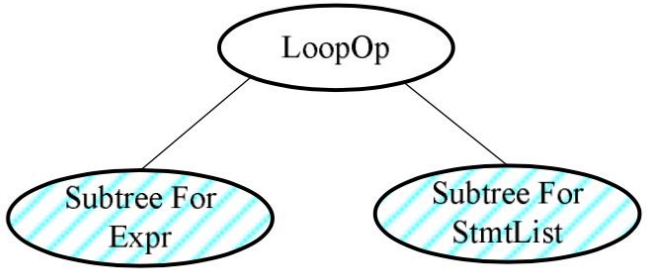
4.4.3.2 Jump

Op	Operands	Description
`j`	*label*	Jump to label *lab*.
`jr`	*src1*	Jump to location *src1*.
`jal`	*label*	Jump to label *lab*, and store the address of the next instruction in `$ra`.
`jalr`	*src1*	Jump to location *src1*, and store the address of the next instruction in `$ra`.

Q: 如何处理循环语句?

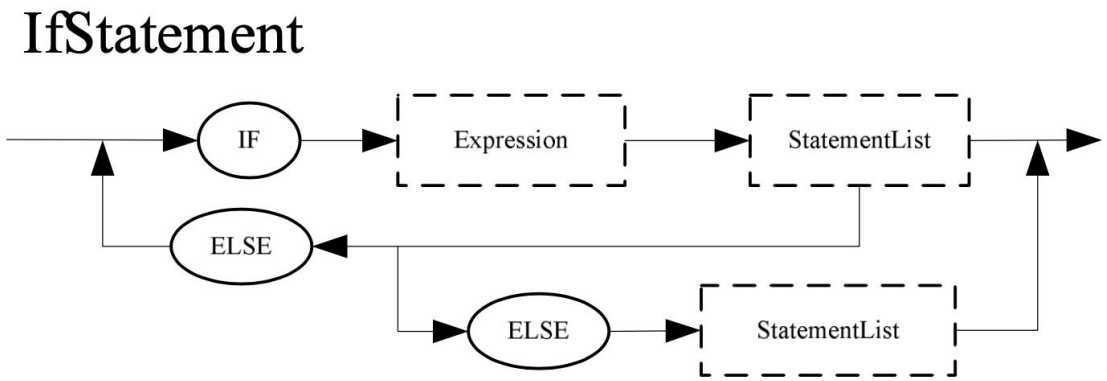


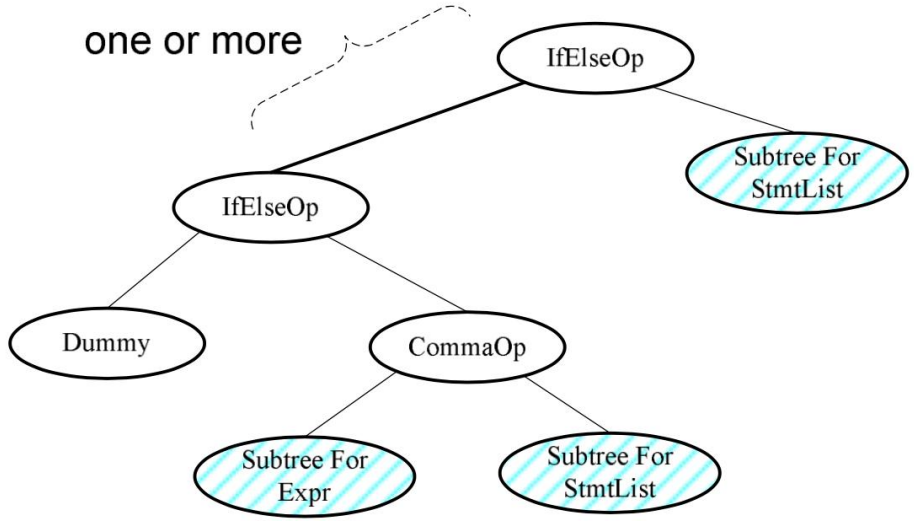

```
1   void emitLoop(tree syntaxTree) {
2       static int loopCount = 0;
3
4       fprintf(outputFile, "while_start_%d:\n", loopCount);
5
6       process(LeftChild(syntaxTree));
7
8       fprintf(outputFile, "beq $0 $t0 while_end_%d\n", loopCount);
9
10      process(RightChild(syntaxTree));
11
12      fprintf(outputFile, "j while_start_%d\n", loopCount);
13      fprintf(outputFile, "while_end_%d:\n", loopCount);
14
15      loopCount++;
16  }
```



Q: 如何处理if语句?

```
1   void emitIf(tree syntaxTree, int endIfCount) {
2       static int ifCount = 0;
3
4       tree leftIfTree = LeftChild(syntaxTree);
5       if (leftIfTree != NullExp()) {
6           emitIf(leftIfTree, endIfCount);
7       }
8
9       tree rightIfTree = RightChild(syntaxTree);
10      if (NodeOp(rightIfTree) != CommaOp) {
11          fprintf(outputFile, "     # else\n");
12          process(rightIfTree);
13          fprintf(outputFile, "j if_endif_%d\n", endIfCount);
14      } else {
15          fprintf(outputFile, "     # if stmt\n");
16          process(LeftChild(rightIfTree));
17          fprintf(outputFile, "beq $0 $t0 if_false_%d\n", ifCount);
18          process(RightChild(rightIfTree));
19          fprintf(outputFile, "j if_endif_%d\n", endIfCount);
20          fprintf(outputFile, "if_false_%d:\n", ifCount);
21          ifCount++;
22      }
23  }
24
25  void emitIfStart(tree syntaxTree) {
26      static int endIfCount = 0;
27      emitIf(syntaxTree, endIfCount);
28      fprintf(outputFile, "if_endif_%d:\n", endIfCount);
29      endIfCount++;
30  }
```





Thanks

Ziyan Wang