EECS 665

HW7

Qixiang Liu

2856114

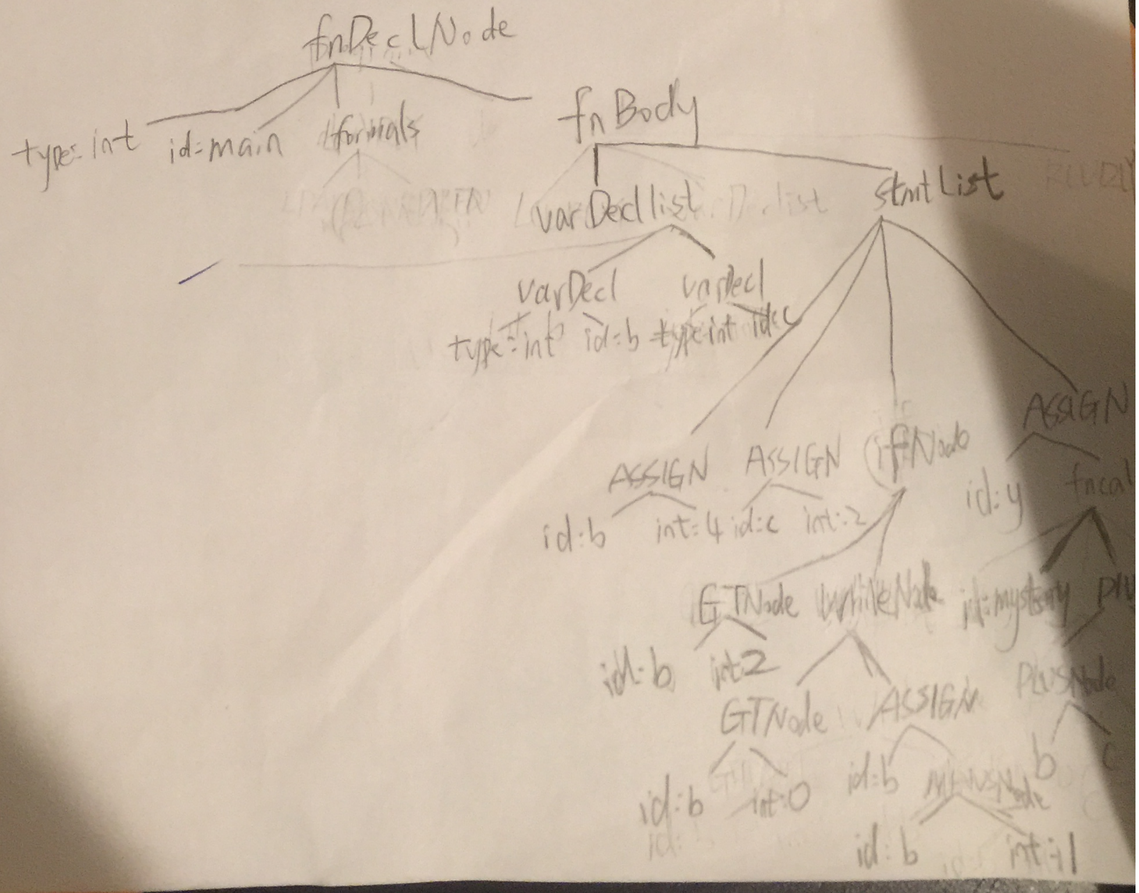
Q 1: IR help the compiler achieve its goals, and avoids dealing with target language directly. IR also decouples compiler frontend to backend, break down source language constructs into target language over several steps. Finally, IR optimize programs!

① AST can be a good target for analysis such as function inling, (type checking and constant fold!

② 3AC (linear): ① simple model of memory ② Highly constrained set of instructions ③ The most "complex" instruction has three operands and one operator

③ CFG (hybrid): CFG can maintain a linear order, primary for intraprocedural control-flow! CFG has domination tree!

Q2: part I AST



4 rules!

1) Syntactic details

2) operator move up

Part 2: 3AC

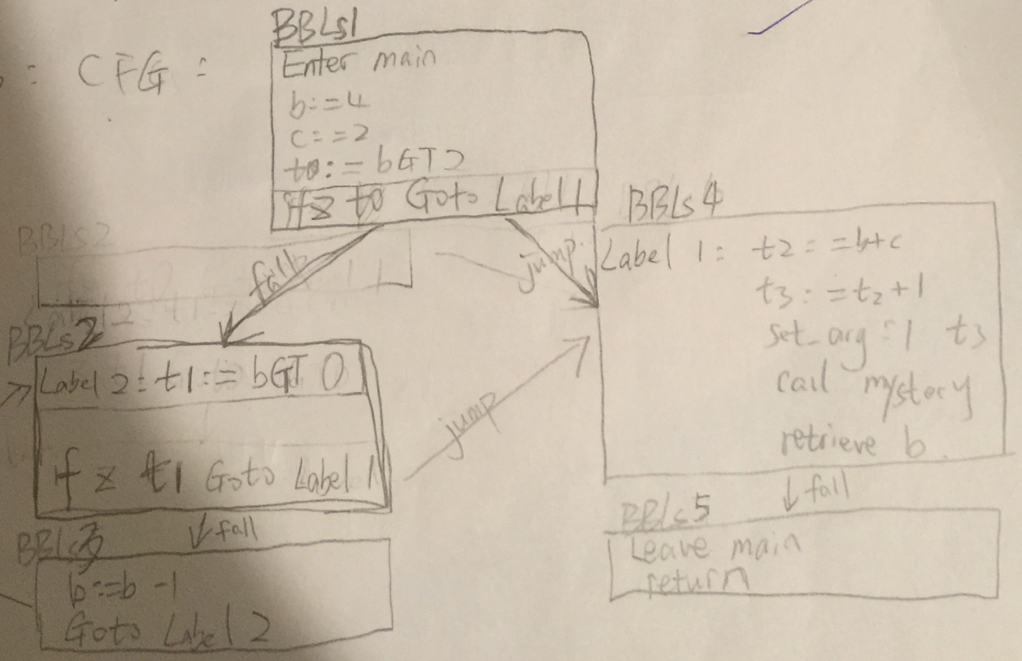
```
Enter main
b:= 4
c: =2
t0:= b GT 2
ifz t0 Goto Label 1
Label2: t1:=b GT 0
        ifz t1 Goto Label 1
                b:= b -1
                Goto labal 2

Label 1:        t2:= b+c
                t3:= t2+1
                set_arg : 1 t3
                call mystery
                retrieve b

leave main

return.
```

part 3: CFG:


BBLs1
Enter main
b:=4
c:=2
t0:=bGT2
ifz t0 Goto Label1
fall
jump
BBLs4
Label 1: t2:=b+c
t3:=t2+1
Set_arg :1 t3
call mystery
retrieve b
BBLs2
Label 2: t1:= bGT 0
ifz t1 Goto Label 1
jump
jump
fall
BBLs3
b:=b -1
Goto Label 2
fall
BBLs5
Leave main
return