

Justin Nguyen
CSC 4351
Professor Baumgartner
22 April, 2022

unCx()

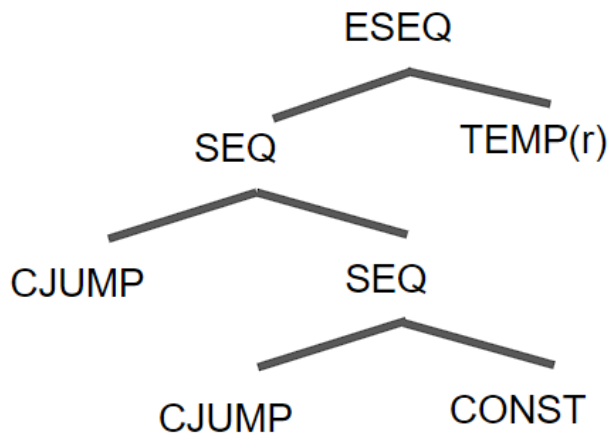
The program uses the native implementation provided by the skeleton packages.

unEx()

This function uses unCx() on the condition node and its child nodes a/b. After this, the function returns an ESEQ with a tree structure similar to the tree in the textbook.

```
Tree.Exp unEx() {  
    // You must implement this function  
    Temp r = new Temp();  
    Tree.Stm condStm = cond.unCx(t, f);  
    Tree.Stm aStm = a.unCx(join, join);  
    Tree.Stm bStm = b.unCx(join, join);  
    return new Tree.ESEQ(new Tree.SEQ(new Tree.SEQ(condStm, new Tree.SEQ(aStm, bStm)), unNx()), new Tree.TEMP(r));  
}
```

For the example conditional “if $a < b$ then $c < d$ else 0” a tree that looks like the following is produced.



unNx()

unNx() is just a copy of the provided unNx() method.

```
Tree.Stm unNx() {  
    // You must implement this function  
  
    return new Tree.SEQ(new Tree.SEQ(unCx(join, join), join);  
}
```

Output

I did very little in terms of modifying Translate. Translate very little was changed. I don't really understand the implementation so the output is missing all of the trees that are created by "IfThenElseExp.java"

```
tigermain:  
MOVE(  
    TEMP t0,  
    CONST 0)
```

This is all that the implemented functions would print