

PA7 Overview

CS 471
November 14, 2007

UNIVERSITY OF VIRGINIA

The Main Idea for this Assignment

Convert Abstract Syntax Tree into an Intermediate Representation Tree

- Fairly straightforward, but lots of cases to handle
- Using techniques discussed in
 - IR lectures
 - Chapter 7 (will be VERY helpful)

You are given:

- tree.c, tree.h, printtree.c

You will fill in:

- translate.c, translate.h
- semant will now call on translate appropriately

Key Concepts

Ex – an “expression” represented as Tr_Ex
Nx – has “no result” represented as Tr_Nx
Cx – is “conditional” represented as Tr_Cx

- Some “expressions” in your AST should be converted to “statements”, e.g. while
- Other “expressions” should be converted to “conditionals”, e.g. boolean

Conditionals

$a > b \mid c < d$

Recall: `T_Cjump(T_binOp, L, R, Tlabel, Flabel)`

```
Temp_label z = Temp_newlabel()
T_stm s1 = T_Seq(T_Cjump(T_gt, a, b, ■, z),
                T_Seq(T_Label(z),
                    T_Cjump(T_lt, c, d, ■, ■)));
```

Maintain a “patch list” of label locations that we’ll fill in later...

Trues ■ will all jump to same place
 Falses ■ will too (at least the “holes” will)

While Loops

```
while (condition) body
```

Will become:

```
test:
    if not (condition) goto done
    body
    goto test
done:
```

Handling Variables

Must determine whether they’re local

- If so, emit code referencing the value as an offset from FP
- If not, emit code that traverses the Static Link in our Frame
 - This is why we kept track of the “level” of nesting ... so we know how many times to traverse the static link

Output

Call `printStmList` (in `printtree.c`)

This is due after break

- Nov 30 (midnight)
- Three weeks since PA6
- Two days after Test 2

