

Plan for today

PA8 updates

- make sure to get an updated version of MipsFrame.java or fix the initArray implementation
- make sure to fix Translate so that it correctly translates while statements

Assem data structure

Tree patterns for MiniJava

CS453 Lecture

Tree Patterns

2

Assem intermediate representation

Assem.Instr

- "assembly language instruction without register assignments"

OPER(String assem, List<Temp> dst, List<Temp> src, List<Label> jumps)

- contains a string with holes for registers indicated by 'd#' and 's#' and holes for labels indicated by 'j#'
- dst and src are lists of Temps whose register assignment should fill holes
 - first entry in src is associated with 's0, second with 's1, etc.
 - first entry in dst is associated with 'd0, etc.
- jumps is a list of labels for filling in label holes

LABEL(String assem, Label label)

- a label statement in the target code

MOVE(String assem, Temp dst, Temp src)

- similar to OPER in that assem string contains holes, but ..
 - no jumps
 - only one src and dst Temp

CS453 Lecture

Tree Patterns

3

Tree Patterns

Approach

- organize them by Tree.Exp and Tree.Stm node and for each one figure out if munchNodeNAME is needed
- determine which nodes correspond to code generation for MIPS

Tree.Exp nodes

- ExpCONST(int i)
- ExpNAME(Label n) - parent node will do any code gen
- ExpTEMP(Temp t) - code gen not needed because result already in Temp
- ExpBINOP(int binop, Exp left, Exp right)
 - code gen based on?
 - how do we get Temps for left and right?
- ExpMEM(Exp exp)
 - where can this show up in Tree.Stm?
- ExpCALL(Exp func, List<Exp> args)
 - what should happen here?

CS453 Lecture

Tree Patterns

4

Tree Patterns

Tree.Exp nodes

- StmMOVE(Temp t, Exp rhs)
- StmMOVE(ExpMEM lhs, Exp rhs)
- StmEXP(Exp e)
- StmJUMP(Label targ)
- StmCJUMP(int relop, Exp lhs, Exp rhs, Label true, Label f)
- StmLABEL(Label l)

CS453 Lecture

Tree Patterns

5