

CS 4240 Project Phase 3

David Benas, Sean Collins, Casey Evanish

December 7, 2014

Run/Compile Instructions

Running `TestTreeWalk.java` should output to console any intermediate steps. In root directory, this also outputs the MIPS.s file and any other necessary files. Please note that this report will be less detailed than the last couple, since things can be much more easily clarified in the demo.

1 Register Allocation

Simply put, for allocating registers, we get the blocks in the code, and use maps in order to appropriately substitute in the proper registers. The way we did this becomes clear in the file `CFGIntraBlockAllocation.java`.

2 Instruction Selection/Code Generation

In this portion of the project, we take the IR code that has the appropriate registers allocated to it and convert this into functional MIPS code. Essentially, what we did was use a Hashmap of (Function Name, Function Form), for example, add:

```
IR_MIPS_OP_MAPPINGS.put("add", "add <PARAM3>, <PARAM1>, <PARAM2>");
```

Following this, we looped through the entirety of the IR and put in the appropriate params. This got much more complicated for the floating point operations, since many of these had different instructions. So anytime one of the allocated registers was a \$f register, we would have to change the instruction from an add to an add.s. This got further complicated when we had to cast integers to floats, as the add.s operation would only take in the correct types. We did this by taking the non-float register and casting it to open float registers as such:

```
\\if the first argument is not a float register
"mtc1 " + components[2] + ", " + firstArg + "\\n" +
"cvt.s.w " + firstArg + ", " + firstArg + "\\n" +
insertParams(IR_MIPS_OP_MAPPINGS.get("add.s"), dest, firstArg, secondArg);
```

Essentially, if we detected a cast was necessary, we used the `mtc1` and `cvt.s.w` commands to do this for us.

After this we encountered a syntax error on our labels, since we did not realize that labels could not begin with integers or contain special characters (we had dashes). The fix for this was a simple grammar change, after which we solved all syntax error problems.