Lab 06

October 5, 2016

Control Flow Graphs

Control flow graphs (CFGs) are flow diagrams for computer programs. We have actually been writing basic blocks the whole time so far without formally describing them. From Wikipedia: "... a basic block is a straight-line code sequence with no branches in except to the entry and no branches out except at the exit. This restricted form makes a basic block highly amenable to analysis. Compilers usually decompose programs into their basic blocks as a first step in the analysis process. Basic blocks form the vertices or nodes in a control flow graph."

With MIPS instructions, when we encounter a `beq` instruction for example, we either jump to the label or go to the next instruction. In this lab, we are going to use `branch/jump` pairs to force explicit flow control. In the event that we don't follow our `branch` condition, we will jump to another basic block. Therefore, the following code will arrive at either the `next` label or the `after` label.

```
bne $s0, $t0, next
j after
```

This is very helpful when dealing with loops. Loops have three primary basic blocks: the loop header, the latch, and then one or more basic blocks composing the loop body. The loop header determines whether or not to continue the loop body (for example, whether `i < 5` evaluates to true). The latch is where the iterator variable is modified (e.g., `i++`). Inside the body we perform our computation including other looping constructs. The loop body performs some operations and jumps directly to the latch block; the latch block modifies the loop variable and jumps directly to the loop header.

Construct a program following the CFG on the following page that requests an integer n and outputs 0, 1, ... n-1 on its own line.

MIPS Triangle (part 1)

In Lab 01 we had a problem that asks you to program a triangle:

```
What is n? 4
*
**
***
****
```

Draw the CFG corresponding to this program using the guidelines given above and the example CFG.

MIPS Triangle (part 2)

Implement this following the CFG you constructed from the previous checkpoint.

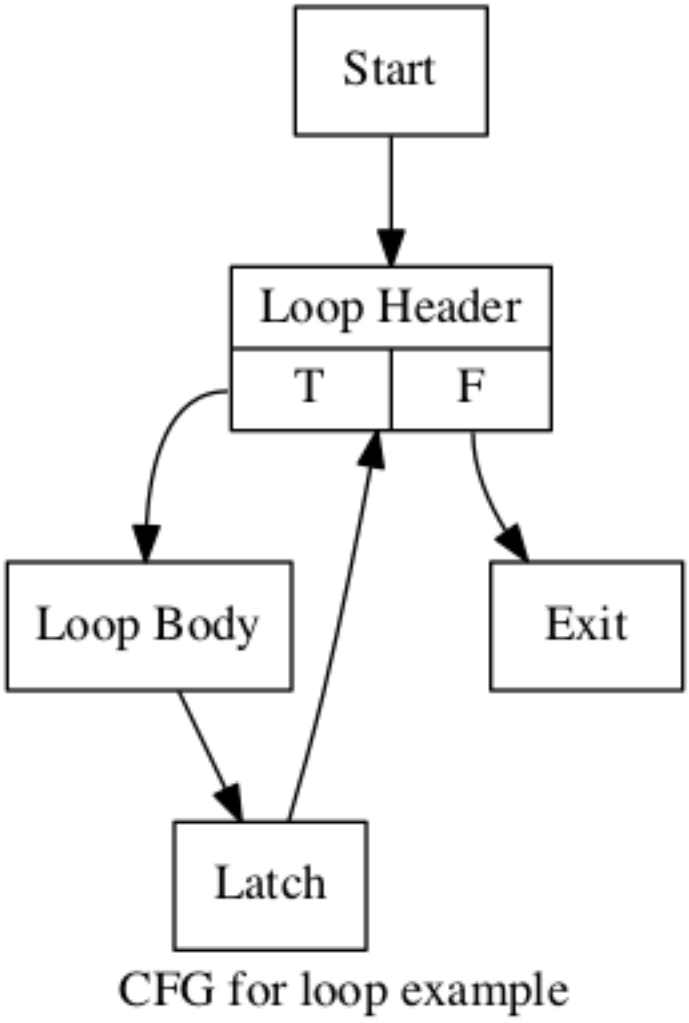
Start
Loop Header
T
F
Loop Body
Exit
Latch
CFG for loop example

Figure 1: CFG