

15-745: Spring 2011

Homework 1

Salil Joshi
salilj@andrew.cmu.edu

Cyrus Omar
cyrus@cmu.edu

1 Using FunctionInfo Results

1. Function signatures, as used for forward references, are represented as functions within the LLVM IR. Since signatures have no body, they have zero basic blocks.
2. The number of call sites a function has is important in determining whether inlining it will increase the size of the assembly unacceptably. If this number is small, inlining can occur safely. The threshold is higher for functions with fewer instructions.
3. The number of call sites can be a heuristic indicator of the relative importance of a function in a program. The number of basic blocks can similarly serve as a heuristic measure of the complexity of control flow through a function.

If the number of basic blocks is small, it may be fruitful to use asymptotically inefficient optimization techniques because the actual magnitude of time it takes remains small.

The number of instructions can be useful in estimating the amount of time it may take to run a pass on the function.

2 Implementation

We created a per-function pass (to facilitate constant propagation within the function) and iterated through each instruction, dispatching on its instruction type with a `switch` statement.

1. **Constant Propagation** When a constant was stored in a variable, we found places where it was being loaded into a register, and replaced all uses of the register with the constant by going through its use list.
2. **Algebraic Identities**
3. **Constant Folding** stuff
4. **Strength Reduction** stuff

3 Source Listing

stuff