CSCD70-Assignment2

peter.chou

March 2023

1 Programming Writeup

1.1 Liveness

For the liveness pass we treat the all the incoming variables that is part the of the ϕ instruction as uses. Since the definition of a live variable is that it holds a value that may be needed in the future, or equivalently if its value may be read before the next time the variable is written to. therefore we must treat all incoming values of a ϕ node as uses.

1.2 Sparse Conditional Constant Propagations

For sparse conditional propagation I've decided to forgo the use of the built in llvm::LatticeValueElement mostly because it was giving me unexpected behaviour and I've decided to roll my own Lattice struct. The Lattice data structure works exactly as described in lecture.

The Lattice can be in three states: **undefined** (representing variable is never reached), **constant**, **overdefined** (representing multiple definition).

The algorithmn for sparse conditional propogration works by assuming every variable is undefined at first and updating variables only when the instruction only has constant operand.

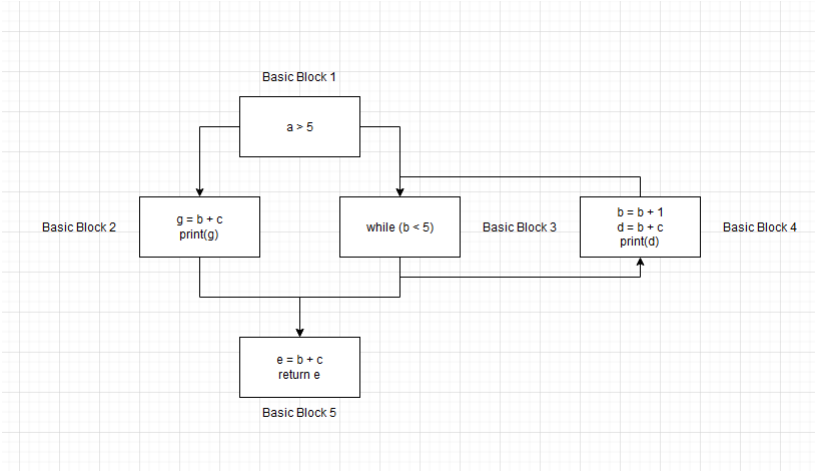
If the instruction is a phi instruction we evaluate if the incoming values are defined in the basic block they are coming from.

We also consider a phi instruction to be constant if it one of the incoming value is undefined but the other one is constant.

Based on computed values we progressively mark the entirely control flow graph as executable or not executable until we arrived at a fixed state where no changes are made.

2 Theortical Questions

1. Build Control Flow Graph



2. Will be Avaliable Expression
3. Postponable Expressions
4. Final pass