**Name: Tan Lai Chian Alan**
**Student Number: A0174404L**
**Email: e0210437@u.nus.edu**

**Design for Task 1**

| | Initialized Variables | Remarks |
|---|---|---|
| $L$<br>$\sqsubseteq$<br>$\sqcup$<br>$\perp$ | $P(Var_* \times Lab_*^?)$<br>$\subseteq$<br>$\cup$<br>$\varnothing$ | Finite and satisfies Ascending Chain Condition<br>Partially ordered by subset inclusion<br>May analysis<br>Bottom is empty set |
| <br>E<br>F | $\{ (x, ?) \mid x \in FV(S_*) \}$<br>$\{ init(S_*) \}$<br>$flow(S_*)$ | Extremal value<br>Because forward analyses<br>Forward analyses |
| F<br>$f_l$ | $\{ f : L \rightarrow L \mid \exists\ l_k, l_g : f(l) = (l \setminus l_k) \cup l_g \}$<br>$f_l(l) = (l \setminus kill([B]^l)) \cup gen([B]^l)$ where $[B]^l \in blocks(S_*)$ | |

**LLVM and Clang Versions Used**
llvm-3.5-dev
clang-3.5

**Implementation of Task 2**
**Step 1:** Read the IR file to extract Module M from IR, extract Function main from Module M.
**Step 2:** Initialize Analysis Map and Traversal Stack, traverse the CFG in Depth First Order (Similar to StackSet.cpp)
**Step 3:** Upon entry (and popping) of each Basic Block from stack, iterate through the Basic Block's set of instructions to check for initialized variables using *isa<StoreInst>(I)*, push instruction to Analysis Map if it's a store instruction.
**Step 4:** Extract last instruction of Basic Block to get successor blocks
**Step 5:** For each successor block, increase value of depth by 1 and push to traversal stack, and then push all initialized variables of parent Basic Block to the successor block.
**Step 6:** Print result of Analysis map

**Implementation of Task 3**
Algorithm used is MOP. The implementation for tasks 2 and 3 are pretty much the same, with just a small tweak for Step 5 in Task 3
**Step 5:** For each successor block, iterate through parent Basic Block's set of initialized variables (instructions). For each instruction, check if the instruction exists in successor block's list of instructions in Analysis map. If does not exist, set *hasChanged* to true and push

instruction to successor's Analysis map. After iterating through all instructions, if *hasChanged* is true, then push successor block to traversal stack.

**Build and Run Pass**

Check out README.md in folder