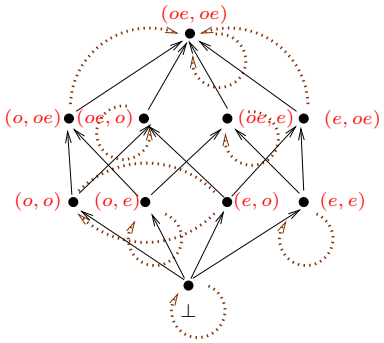
Program Analyis and Verification
Assignment 1
Lattices and Abstract Interpretation Due date: Aug. 30th, 11.59 pm.

Problem 1. Prove or disprove: A lattice is complete iff it has both a least and a greatest element.

Problem 2. Consider the function f on the lattice below. The solid arrows are the edges in the Hasse diagram of the lattice, while the dotted arrows depict the action of the function on the elements of the lattice.



Tell whether the given function on the given lattice is:

(a) Monotonic

(b) Continuous

(c) Distributive.

Justify your answers.

Problem 3. Let us call a function f on a lattice $(D, \leq)$ *increasing* if for each $d \in D$, we have $d \leq f(d)$. Prove or disprove:

(a) Every increasing function is monotonic.

(b) Every monotonic function is increasing.

Problem 4. Consider a programming language wherein each assignment statement s has a statement number (a natural number), accessible using the function $stmtNum(s)$. In a certain run of a program we say that a *definition* s of the variable `x` (i.e., s is an assignment statement whose lhs is `x`) *reaches* a program point t if at least once during the run control reaches t with a value in `x` that was placed into it earlier by s.

(a) Let us define a domain $ExtState = State \times VDef$. $State$ is the set of all concrete states (as in the class lectures). $VDef$ is defined as $Var \to Num$, where Var is the set of variables in the given program, and Num represents the set of statement numbers in the program. Define a concrete transfer function $nstateExt$ with signature $ExtState \to 2^{ExtState}$, such that from the collecting semantics at any program point t in the given program obtained using $nstateExt$, one can identify precisely the definitions that reach t across *all possible runs* of the program (i.e., a definition that reaches t in even one run must be part of identified set).

You need to define $nstateExt$ for both assignment statements and conditional nodes. You may use the given $nstate$ to transfer the $State$ component of each $ExtState$.

Also, indicate how the set of definitions that reach t can be identified from the collecting semantics at t.

(b) Let D be equal to 2^{Num}. Each element of D represents a set of statements in the given program. Define transfer functions f_n with signature $D \to D$ for assignment statements and conditional nodes, and specify the join operation on D, such that from the abstract JOP at any program point t in the given program, one can read-off an over-approximation of the set of definitions that reach t across all possible runs of the program. Subject to the over-approximation requirement just mentioned, you should aim for maximal precision.

(c) Define a function γ_D with signature $D \to 2^{ExtState}$ such that the γ-image of the abtract JOP at any point gives an over-approximation of the collecting semantics at that point. The over-approximation should be as tight as possible.

(d) Show an example program such that at some point in the program the abstract JOP computed using the abstract interpretation you have specified above is a strict superset of the actual set of definitions that can reach that point across all possible runs.

(e) What change is required to your solution in Part 2 above such that the abstract JOP at any program point t is an under-approximation of the set of definitions that happen to reach t in *every run* of the program. Your under-approximation should be as tight as possible.