1. Recall from Homework 1 that you proved that functions consisting of any combination of set union, intersection, or constant sets in some domain \( S \) are monotonic. Suppose that we extend the grammar we used to include the complement operator, so functions \( f(x) = e \) are defined with \( e \) as

\[
e ::= x | s | e \cup e | e \cap e | !e
\]

The semantics of complement \(!T\) is the set \( S - T \) (or put another way, \( \text{eval} \ !e \ v = S - (\text{eval} \ e \ v) \)).

(a) Are functions \( f(x) = e \) still guaranteed to be monotonic? If so, justify your answer. If not, provide a counterexample.

(b) Explain why a transfer function defined by \( \text{Out}(\text{stmt}) = \text{Gen}(\text{stmt}) \cup (\text{In}(\text{stmt}) - \text{Kill}(\text{stmt})) \), which seems to include negation, is monotonic.

2. Let \( A \) be a lattice, with order \( \leq \). Define \( A \to A \) to be the set of all functions from \( A \) to \( A \), and define \( f \preceq g \) iff \( f(x) \leq g(x) \) for all \( x \in A \).

- Show that \( A \to A \) with order \( \preceq \) is also a lattice. That is, show that for all \( f, g \in A \to A \), \( f \sqcup g \) and \( f \sqcap g \) always exist.

- Suppose lattice \( A \) has height \( h \) and that \( A \) is finite with \( n \) elements. What is the height of the lattice \( (A \to A, \preceq) \)? (When counting height, count “edges” rather than “nodes,” e.g., if \( A \) were the lattice \( \{a, b\} \) with \( a < b \), then its height would be 1.)

3. A programming language ensures definite initialization when no program in the language is able to read a variable before it is initialized (written). Your task is to ensure that this property holds by designing a dataflow analysis on our language of commands (not including lists):

variables \( x, y, z \in V \)

integers \( i, j, k \in \mathbb{Z} \)

expressions \( e ::= x | i | e_1 + e_2 | e_1 \times e_2 \)

commands \( c, d ::= \text{skip} | x := e | \text{if}_{\text{not} 0} e \text{ then } c_1 \text{ else } c_2 | \text{while}_{\text{not} 0} e \text{ do } c | c_1 ; c_2 \)

Write down the following about your analysis:

(a) State precisely what your analysis aims to do. Explain how this will be used to ensure definite initialization.

(b) Give the direction of your analysis.

(c) Define the lattice, and define the meet operation.

(d) State how you would initialize the dataflow facts at each program point, including the entry or exit nodes, as appropriate.

(e) Give the transfer functions (e.g., in terms of \( \text{Gen} \) and \( \text{Kill} \) functions) for each statement, where for \( \text{if} \) and \( \text{while} \) the transfer function should be for the guards.