

CMSC 631, Fall 2007

Homework 4

Due Monday, Oct. 22, in class

- Recall from Homework 1 that you proved that functions consisting of any combination of set union, intersection, or constant sets s 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

$$e ::= x \mid s \mid e \cup e \mid e \cap e \mid !e$$

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

- Are functions $f(x) = e$ still guaranteed to be monotonic? If so, justify your answer. If not, provide a counterexample.
 - Explain why a transfer function defined by $\text{Out}(stmt) = \text{Gen}(stmt) \cup (\text{In}(stmt) - \text{Kill}(stmt))$, which seems to include negation, is monotonic.
- Let A be a lattice, with order \leq . Define $A \rightarrow A$ to be the set of all functions from A to A , and define $f \leq' g$ iff $f(x) \leq g(x)$ for all $x \in A$.

- Show that $A \rightarrow A$ with order \leq' is also a lattice. That is, show that for all $f, g \in A \rightarrow A$, $f \sqcup g$ 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 $(A \rightarrow A, \leq')$? (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.)

- 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):

<i>variables</i>	x, y, z	\in	V
<i>integers</i>	i, j, k	\in	\mathcal{Z}
<i>expressions</i>	e	$::=$	$x \mid i \mid e_1 + e_2 \mid e_1 * e_2$
<i>commands</i>	c, d	$::=$	skip $\mid x := e \mid \mathbf{if}_{not0} e \mathbf{then} c_1 \mathbf{else} c_2 \mid \mathbf{while}_{not0} e \mathbf{do} c \mid c_1; c_2$

Write down the following about your analysis:

- State precisely what your analysis aims to do. Explain how this will be used to ensure definite initialization.
- Give the direction of your analysis.
- Define the lattice, and define the meet operation.
- State how you would initialize the dataflow facts at each program point, including the entry or exit nodes, as appropriate.
- Give the transfer functions (e.g., in terms of Gen and Kill functions) for each statement, where for *if* and *while* the transfer function should be for the guards.