

CMSC 631, Fall 2006

Homework 1

Due Wednesday, Sept. 20, in class

- Let S be a finite set, and let L be the lattice of subsets of S , with order \subseteq . Show that any function $f(x)$ constructed from union, intersection, and constant sets is monotonic. Here, I mean that $f(x) = e$ where e can be specified by the grammar

$$e ::= x \mid S' \mid e \cup e \mid e \cap e$$

where S' is any subset of S . Your proof should be by induction on the structure of e .

- Suppose that we extend the grammar for e from problem 1 to include the complement operator $!e$, where $!T = S - T$ for some set T . Is f still guaranteed to be monotonic? If it is, justify your answer. If it's not, provide a counterexample. In addition, explain why a transfer function defined by $Out(stmt) = Gen(stmt) \cup (In(stmt) - Kill(stmt))$, 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 a simplification of the 3-address code language from project 1 (doesn't include arrays):

```

lab_stmt ::= stmt
          | identifier : stmt
stmt      ::= id:=arg1 binop arg2;
          | id:=arg;
          | goto label;
          | if arg1 relop arg2 goto label;
binop     ::= + | - | * | /
relop     ::= == | != | < | > | >= | <=
id        ::= identifier
label    ::= identifier
arg       ::= identifier | constant
    
```

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 three-address code statement.