CSC2226 - Assignment 1

Due: March 9, 2016 (in class) *

1. Recall the Hoare logic proof calculus:

\[
\begin{align*}
\text{Consequence} & \quad \varphi \vdash \varphi' & \vdash \{\varphi'\} S \{\psi'\} & \psi' \vdash \psi \\
& \vdash \{\varphi\} S \{\psi\} \\
\text{Sequencing} & \quad \vdash \{\varphi\} S_1 \{\varphi'\} & \vdash \{\varphi'\} S_2 \{\varphi''\} & \vdash \{\varphi\} S_1; S_2 \{\varphi''\}
\end{align*}
\]

\[
\begin{align*}
\text{Conditional} & \quad \vdash \{\varphi \land \psi\} S \{\varphi'\} & \vdash \{\varphi \land \neg \psi\} S' \{\varphi'\} \\
& \vdash \{\varphi\} \text{ if } \psi \text{ then } S \text{ else } S' \{\varphi'\} \\
\text{Loop} & \quad \vdash \{\varphi \land \psi\} S \{\varphi\} \\
& \vdash \{\varphi\} \text{ while } \psi \text{ do } S \{\varphi \land \neg \psi\}
\end{align*}
\]

\[
\text{Assignment} \quad \vdash \{\varphi[v \mapsto t]\} v := t \{\varphi\}
\]

Derive

\[
\vdash \{x = 0 \land y = 0 \land c = 3\} S \{2x \leq y\}
\]

where \(S\) is the program

\[
\text{while } z > 0 \text{ do (if } c = 0 \text{ then } (c := 3; x := x + 1) \text{ else } (c := c - 1; y := y + 1); z := z - 1)\]

Please use the \(S, S_1, S_2, S_3, S_4\) as short-hand to make the derivation shorter. You do not need to show the derivation for logical entailments in the consequence rule (but you should be confident that they are valid!)

2. Recall Kleene iteration, the high-level algorithm used by constant propagation (and other analyses):

*If your solution is typed up and in the form of a PDF file, then you may submit as late as March 11 (end of the day) by email (azadeh@cs.toronto.edu). Handwritten solutions should be submitted in class on March 9th.
Input : CFA \((V, E, r)\)

Complete lattice \(\langle D, \sqsubseteq, \sqcup, \bot, \top \rangle\)

\([\cdot]^\# : \text{Instr} \to (D \to D)\) such that \([\cdot]^\#\) is monotone for any \(\text{instr} \in \text{Instr}\)

Output: Inductive annotation \(\Phi : V \to D\)

For all \(v \in V\), initialize \(\Phi(v) \leftarrow \bot\);
\(\Phi(r) \leftarrow \top\);

repeat
\[\Phi' \leftarrow \Phi;\]
foreach \(u \xrightarrow{\text{instr}} v \in E\) do
\[\Phi(v) \leftarrow \Phi(v) \sqcup [\text{instr}]^\#(\Phi'(u));\]
end
until \(\Phi = \Phi'\);
return \(\Phi\)

Algorithm 1: Kleene iteration

(a) Suppose that \(D\) is instantiated to the constant propagation domain, that there are \(N\) variables in the program, every vertex is incident to \(O(1)\) edges, and that evaluating \([\cdot]^\#(\Phi'(u))\) takes \(O(1)\) time. What is the worst-case time complexity of this algorithm, expressed in \(N\) and \(|V|\)?

(b) Give an algorithm which improves upon this algorithm by a factor of \(|V|\). (Hint: for a given vertex \(v\), the algorithm always computes a new value for \(\Phi(v)\), even in cases where we know that it will be the same as \(\Phi'(v)\). How can that be avoided?)

3. Let \(P\) be a propositional signature. Let \(\text{Formula}(P)\) denote the set of all propositional formulas over \(P\), and let \(\text{Val}(P) : P \to \{\text{true, false}\}\) denote the set of valuations over \(P\).

Define a function \(f : 2^{\text{Formula}(P)} \to 2^{\text{Val}(P)}\) as

\[f(\Phi) \triangleq \{M \in \text{Val} : \forall \varphi \in \Phi. M \models \varphi\}\]

Exhibit a function \(g\) such that either
\[\left(2^{\text{Formula}(P)}, \sqsubseteq \right) \xleftarrow{g} \left(2^{\text{Val}(P)}, \sqsubseteq \right)\]

or
\[\left(2^{\text{Val}(P)}, \sqsubseteq \right) \xleftarrow{g} \left(2^{\text{Formula}(P)}, \sqsubseteq \right)\]

is a Galois connection, and prove that it is a Galois connection (Hint: only one direction is possible. Which one?)

4. Use the framework of abstract interpretation to derive an algorithm for shortest paths.

Suppose that \(G = \langle V, E \rangle\) is a directed graph and \(v \in V\) is a vertex. The (complete) lattice \((2^{E^*}, \sqsubseteq)\) consists of sets of sequences of edges (a superset of the paths in \(G\)), ordered by inclusion.
(a) Define a monotone function $f : 2^{E^*} \to 2^{E^*}$ such that $\text{lfp}(f)$ is the set of all paths in $G$ which start at $v$. You may find it helpful to use a function $\text{tgt} : 2^{E^*} \to V$ defined as

$$\text{tgt}((u_1,v_1)\langle u_2,v_2 \rangle \ldots \langle u_n,v_n \rangle) \triangleq \begin{cases} v & \text{if } n = 0 \\ v_n & \text{otherwise} \end{cases}$$

(b) A distance function is a function $d : V \to \mathbb{N}^\infty$ (where $\mathbb{N}^\infty = \mathbb{N} \cup \{\infty\}$) which assigns each vertex in $V$ its distance from $v$.

i. Define an order relation $\sqsubseteq$ on $V \to \mathbb{N}^\infty$, under which it is a complete lattice. What is the join operation?

ii. Exhibit a Galois connection $2^{E^*} \xleftarrow{\alpha} V \to \mathbb{N}^\infty$.

iii. Define a function $f^\sharp : (V \to \mathbb{N}^\infty) \to (V \to \mathbb{N}^\infty)$, and show that for all $d : V \to \mathbb{N}^\infty$, we have $\alpha(f(\gamma(d))) \sqsubseteq f^\sharp(d)$

(c) Using the fixed point transfer theorem, argue that $\text{lfp}(f^\sharp)$ overapproximates shortest paths ($d = \text{lfp}(f^\sharp)$ is such that the distance between $v$ and $u$ is at most $d(u)$ for all $u$).

Recall the Fixed point transfer theorem: Let $\langle C, \leq \rangle$ and $\langle A, \sqsubseteq \rangle$ be complete lattices, let $C \xleftarrow{\gamma} A$ is a Galois connection, and let $f : C \to C$ and $f^\sharp : A \to A$ be monotone function such that for all $a \in A$, we have $\alpha(f(\gamma(a))) \sqsubseteq f^\sharp(a)$. Then $\text{lfp}(f) \sqsubseteq \gamma(\text{lfp}(f^\sharp))$. 

3