Semantics of Programming Languages Comprehensive
January 7, 1998

INSTRUCTIONS. Do all questions. You have 4 hours.

Question 1: All of the variables in the following command are integer variables.
Give a formal proof (or natural deduction style proof-tree) that the following specification is partially correct:

$$\{N \geq 0\}\ X := N;\ Y := N;\ \textbf{while}\ Y > 0\ \textbf{do}\ (X := X - 1;\ Y := Y - 2)\ \{X = N \text{ div } 2\}$$

Question 2: There are many parts to this question. Read all of the parts before you try to start answering. In most cases the questions have concise answers that can quickly be written down. (Part(i) is the exception; that may be a little longer.) When you are asked to give an example, be concise.

Definition: A *domain* is an ω-complete partial order. (If you have forgotten the definition of an ω-complete partial order, you can infer it from part(a).)

Part a: Give an example of a partially ordered set in which every subset has an upper bound, but for which there is at least one chain which does not have a least upper bound. (Hence, your example would be an example of a partially ordered set in which every subset has an upper bound, but which is not a domain because it is not ω-complete.)

Part b: Consider the simple imperative language, as in *Tennent*. What is a state?

Part c: Given a nonempty set S, we may consider the *discrete* partially ordered set $\ddot{S} = (S, \sqsubseteq)$ whose partial order is defined by

$$s_1 \sqsubseteq s_2 \ \text{ iff } \ s_1 = s_2\,.$$

Show that $\ddot{S}$ is a domain.

Part d: Show that if D is a domain, then so is the *raised* set $D_\perp$.

Part e: Let D be the discrete domain on three elements a, b, and c. Describe the ordering on $D_\perp$. (Just draw a diagram.)

Part f: Let D be a domain and S just an ordinary nonempty set. Show that the set $S \longrightarrow D$, ordered by

$$f \sqsubseteq g \ \text{ iff } \ \text{for all } s \in S : f(s) \sqsubseteq_D g(s)$$

is a domain.

Definition: Let D_1 and D_2 be domains with partial orders $\sqsubseteq_1$ and $\sqsubseteq_2$, respectively. Let $f : D_1 \longrightarrow D_2$. f is *monotone* iff $x \sqsubseteq_1 y$ implies $f(x) \sqsubseteq_2 f(y)$ for all x, y in D_1. Also, f is *ω-compact* iff for every chain $\{x_i \mid i \in \mathbf{N}\}$ in D_1, the least upper bound, $\bigsqcup_{i=0}^{\infty} f(x_i)$ is defined in D_2 and $f(\bigsqcup_{i=0}^{\infty} x_i) \sqsubseteq_2 \bigsqcup_{i=0}^{\infty} f(x_i)$. f is *ω-continuous* iff f is monotone and ω-compact. (This definition agrees with Tennent's definition of continuity on ω-complete partial orders.

Part g: Give an example of an ω-compact function $f : D \longrightarrow D$, where D is a domain, which is not monotone.

Part h: Let D be a domain with a bottom element $\perp$ and let $f : D \longrightarrow D$ be ω-continuous. Show that $\bigsqcup_{i=0}^{\infty} f^i(\perp)$ is the least fixed point of f. i.e. if $a = \bigsqcup_{i=0}^{\infty} f^i(\perp)$ then $f(a) = a$ and for all d such that $f(d) = d$, $a \sqsubseteq d$.

Part i: Let S be the set of states of the simple imperative language. The set of all partial functions from S to S, denoted by $S \rightharpoonup S$, is identifiable with the set of functions

$$S \longrightarrow \ddot{S}_{\perp}$$

Let β be a function from S to $\{\mathit{true}, \mathit{false}\}$. Let κ be a function in $\ddot{S}_{\perp} \longrightarrow \ddot{S}_{\perp}$ Now, define

$$\Gamma_{\beta,\kappa} : (S \longrightarrow \ddot{S}_{\perp}) \longrightarrow (S \longrightarrow \ddot{S}_{\perp})$$

by

$$\Gamma(c) = \lambda s. \begin{cases} c(\kappa(s)) & \text{if } \beta(s) = \mathit{true} \\ s & \text{if } \beta(s) = \mathit{false} \end{cases}$$

Show that $\Gamma_{\beta,\kappa}$ is ω-continuous.

Part j: The preceding development applies elementary techniques about domains to obtain the semantics of the while-command. Explain how. (Note that by part(h), $\Gamma_{\beta,\kappa}$ has a least fixed point. Let β be the meaning of a Boolean expression B, and let κ be the meaning of a command C. Explain how $\Gamma_{\beta,\kappa}$ is related to the semantics of **while** B **do** C.)

Question 3: Compare and contrast states vs. environments. (Think of States with regard to the semantics of the simple imperative language and think of environments with regard to the semantics of the simple recursive applicative language.)