Fondaments de la programmation
Exercice série 2: Domaines Sémantiques

April 27, 2000

1. Soient les cpo's D et E. Montrer que, par la definition de fonction continue 3.2, si $f : D \to E$ est continue alors f est monotone ($2 \Rightarrow 1$).

2. Soient les cpo's D et E telles que toute chaîne est stationnaire, ç.à.d.

$$\forall \langle x_i \rangle_{i\in\omega} \subseteq D : x_i \sqsubseteq x_{i+1} \Rightarrow \exists k \in \omega : x_k = \sqcup \langle x_i \rangle_{i\in\omega}$$

   Montrer que si $f : D \to E$ est monotone alors f est continue ($1 \Rightarrow 2$).

3. La definition d'une fonction recursive peut etre donnée par la definition d'une relation binaire. Par example la fonction:

$$\begin{array}{rcl} f(0) & = & 1 \\ f(n) & = & n * f(n-1) \end{array}$$

   peut etre vue comme la relation $F \subseteq \mathbb{N} \times \mathbb{N}$ defini recursivement par

$$F = \{\langle 0, 1 \rangle\} \cup \{\langle n, n * y \rangle \,| n > 0 \wedge \langle n-1, y \rangle \in F\}.$$

   Montrer que l'ensemble

$$I = \{\left\langle n, \prod_{i=1}^{n} i \right\rangle |n > 0\}$$

   est le point fixe de F, donc la solution de l'equation recursive pour F. Montrer aussi que: $\langle n, y \rangle \in I$ ssi $f(n) = y$.

4. Soit le cpo $(D, \leq)$ et soit $([D \to D], \sqsubseteq)$ le cpo de fonctions continues telles que:

   - $\forall f, g \in D : f \sqsubseteq g \Leftrightarrow \forall d \in D : f(d) \leq g(d)$

   Montrer que $([D \to D], \sqsubseteq)$ est un cpo.

5. Soit les cpo's $(D_1, \leq_1)$ et $(D_2, \leq_2)$. Montrer que la structure $(D, \leq) = (D_1 + D_2, \leq)$, appelée *somme disjointe* et defini par:

   - $D = (D_1 \times \{1\}) \cup (D_2 \times \{2\}) \cup \{\perp_D\}$
   - $\forall \langle d, i \rangle, \langle d', j \rangle \in D : \langle d, i \rangle \leq \langle d', j \rangle \Leftrightarrow i = j \wedge d \leq_i d'$
   - $\forall \langle d, i \rangle \in D : \perp_D \leq \langle d, i \rangle$

   est un cpo.

6. Soit les cpo's $(D_1, \leq_1)$ et $(D_2, \leq_2)$. Montrer que la structure $(D, \leq) = (D_1 \oplus D_2, \leq)$, appelée *"smashed sum"* et defini par:

- $D = ((D_1 \setminus \{\perp_{D_1}\}) \times \{1\}) \cup ((D_2 \setminus \{\perp_{D_2}\}) \times \{2\}) \cup \{\perp_D\}$
- $\forall \langle d, i \rangle, \langle d', j \rangle \in D : \langle d, i \rangle \leq \langle d', j \rangle \Leftrightarrow i = j \wedge d \leq_i d'$
- $\forall \langle d, i \rangle \in D : \perp_D \leq \langle d, i \rangle$

est un cpo. Utiliser la definition de "smashed sum" pour construire le domain $Alist$ des listes de type A comme solution de l'equation des domaines recursifs:

$$Alist = (Nil \oplus (A \times Alist))$$

ou $Nil = \langle \perp, nil \rangle$ et $\perp \leq nil$.