Encore et toujours des dcpos

Exercice 1 :
(Booléens) On considère $\mathbf{Bool}_\perp = \{0, 1, \perp\}$ ordonné tel que avec $x < y$ si et seulement si $x = \perp$ et $y \neq \perp$.

1. Quels sont les ouverts de Scott de $\mathbf{Bool}_\perp$? Les fermés ?
2. Exhiber toutes les fonctions monotones de $\mathbf{Bool}_\perp$ dans $\mathbf{Bool}_\perp$.
3. Soit D un dcpo, et f une fonction monotone de $\mathbf{Bool}_\perp$ dans D. Montrer que f est Scott-continue.
4. Dessiner $\mathbf{Bool}_\perp \times \mathbf{Bool}_\perp$ (ordre produit).
5. Énumérer les fonctions Scott continues f telle que f restreinte à $\{0, 1\}$ définit la fonction booléenne « ou ».
6. En voyant $\perp$ comme « un calcul divergent », donner une interprétation calculatoire de chacun des prolongements à $\mathbf{Bool}_\perp$ de la fonction booléenne « ou ».

Exercice 2 :
(Réels à précision arbitraire) Soit $I = \mathbb{R}$ et $J = \{[x, y] \mid x, y \in I, x \leq y\}$ avec l'ordre $\supseteq$.

1. Montrer que J est un dcpo. Est-ce un treillis ? Complet ?
2. Donner une fonction monotone de J dans $\mathbf{Bool}_\perp$ qui n'est pas Scott-continue.
3. Quels sont les éléments maximaux de J ? Notons M l'ensemble des éléments maximaux.
4. Soit f une fonction continue de J dans $\mathbf{Bool}_\perp$. Montrer que $f^{-1}(\{1\})$ est ouvert.
5. Considérons l'application $I : x \mapsto \{x\}$ qui va de $[0, 1]$ dans J. Montrer que I est continue.

Bonus. Qu'est-ce que la topologie de Scott restreinte à l'ensemble des éléments maximaux ?

6. Montrer que M est connexe. C'est-à-dire qu'il n'existe pas d'ouverts U, V disjoints tous deux non vides tels que
$$U \cap M \uplus V \cap M = M \tag{1}$$
7. Soit g une fonction continue de J dans $\mathbf{Bool}_\perp$ telle que pour tout $x \in I$, $g(\{x\}) \neq \perp$. Montrer que g est constante sur I.
8. Imaginons un langage de programmation qui implémente l'arithmétique réelle de précision arbitraire en utilisant des intervalles. Comment se compotera la fonction d'égalité de ce langage ?

Un micro langage

Exercice 3 :
On considère le langage $\{\star, \bullet\}$ équipé de la sémantique à petits pas suivante :

$$X \star \bullet Y \to XY, \text{ si } \exists n \geq 0, X = \star^n \qquad X \bullet \star Y \to XY, \text{ si } \exists n \geq 0, X = \bullet^n$$

1. Énoncer puis prouver un théorème de déterminisme.

2. On considère le dcpo $\{0,1\}$ équipé de l'ordre plat et la sémantique suivante :
$$[\![\varepsilon]\!]_2 = 0 \qquad [\![aX]\!]_2 = 1 - [\![X]\!]_2,\text{ si } a \in \{\star, \bullet\}$$
Montrer que cette sémantique est correcte par rapport à la sémantique à petits pas.
3. Même question pour le dcpo des entiers relatifs équipé de l'ordre plat et la sémantique suivante
$$[\![\varepsilon]\!]_{\mathbb{Z}} = 0 \qquad [\![\star X]\!]_{\mathbb{Z}} = 1 + [\![X]\!]_{\mathbb{Z}} \qquad [\![\bullet X]\!]_{\mathbb{Z}} = -1\ + [\![X]\!]_{\mathbb{Z}}$$
4. On se donne la notion d'équivalence observationnelle suivante :
$$A \equiv B \triangleq \forall C[\cdot], C[A] \to^* \varepsilon \iff C[B] \to^* \varepsilon$$
   (a) Montrer que c'est une relation d'équivalence.
   (b) Les sémantiques dénotationnelles sont-elles complètement abstraites ?

Maintenant, on type

Ne surtout pas hésiter à utiliser le cours sans modération.

Exercice 4 :
(Un exercice nécessaire) Considérons l'expression PCF u suivante :

```
letrec f x = if x=0 then 0 else x + f (x-1) in f 6
```

Calculer la sémantique dénotationnelle de u.

Exercice 5 :
(La sémantique n'est pas adéquate) Considérons l'expression PCF u suivante :

```
letrec f (x) = 3 in
letrec g (x) = g (x) in
f (g 0)
```

1. Ce n'est pas une expression valide, car il manque les annotations de type. Les ajouter.
2. Calculer la sémantique dénotationnelle de u.
3. Montrer qu'il n'y a aucune dérivation $E \vdash u \Rightarrow v$ où E est un P-environnement, et v une valeur.

Exercice 6 :
(Boolean Function Calculus) On considère le langage suivant
$$M := x \mid \lambda x : \tau.M \mid MN \mid \textbf{let } x : \tau = M \textbf{ in } N \mid \textbf{ff} \mid \textbf{tt} \mid \textbf{if } M \textbf{ then } N \textbf{ else } P$$

1. Proposer un système de typage adapté.
2. Donner une dérivation de $\vdash (\lambda x.\textbf{if } x \textbf{ then ff else } x)\textbf{tt} : \textbf{bool}$
3. Montrer que le système de type proposé est déterministe.
4. Quel élément de la syntaxe du langage de programmation est crucial pour garantir le déterminisme du typage ? Expliciter avec un exemple.
5. Montrer que la construction `let` s'encode à l'aide des autres constructions de manière bien typée.