Übungsblatt 9

Aufgabe 1. Zeigen Sie, dass wenn $(D_1, \sqsubseteq_{D_1}), \dots, (D_n, \sqsubseteq_{D_n})$ für $n \geq 0$ CPOs sind, dann ist auch $(D_1 \oplus \dots \oplus D_n, \sqsubseteq_{D_1 \oplus \dots \oplus D_n})$ eine CPO.

Lösung. Wir schreiben $\sqsubseteq$ für $\sqsubseteq_{D_1 \oplus \dots \oplus D_n}$ und $\bot$ für $\bot_{D_1 \oplus \dots \oplus D_n}$. Des Weiteren sei $D'_i = D_i \setminus \{\bot_{D_i}\}$ für $1 \leq i \leq n$.
Wir zeigen zunächst, dass $\sqsubseteq_{D_1 \oplus \dots \oplus D_n}$ eine partielle Ordnung ist.
Reflexivität: Sei $d \in D_1 \oplus \dots \oplus D_n$. Im Fall, dass $d = \bot$ ist, gilt $d \sqsubseteq d$. Im Fall, dass $d = (e, i) \in D'_i \times \{i\}$ ist, gilt $e \sqsubseteq_{D_i} e$ und deshalb auch dass $d \sqsubseteq d$.
Antisymmetrie: Seien $d, d' \in D_1 \oplus \dots \oplus D_n$ mit $d \sqsubseteq d'$ und $d' \sqsubseteq d$. Wenn $d = \bot$, dann folgt aus $d' \sqsubseteq d$ auch, dass $d' = \bot$, also $d = d'$. Der Fall, dass $d' = \bot$, ist analog. Wenn $d \neq \bot$, dann ist $d = (e, i) \in D'_i \times \{i\}$. Wegen $d \sqsubseteq d'$ muss auch $d' = (e', i) \in D'_i \times \{i\}$ sein mit $e \sqsubseteq_{D_i} e'$. Aus $d' \sqsubseteq d$ folgt analog, dass $e' \sqsubseteq_{D_i} e$. Also gilt wegen Antisymmetrie von $\sqsubseteq_{D_i}$, dass $e = e'$ und somit, dass $d = d'$.
Transitivität: Seien $d, d', d'' \in D_1 \oplus \dots \oplus D_n$ mit $d \sqsubseteq d'$ und $d' \sqsubseteq d''$. Wenn $d = \bot$, dann gilt auch $d \sqsubseteq d''$. Sei nun $d \neq \bot$, also $d = (e, i) \in D'_i \times \{i\}$. Wegen $d \sqsubseteq d'$ muss also $d' = (e', i) \in D'_i \times \{i\}$ sein mit $e \sqsubseteq_{D_i} e'$. Wegen $d' \sqsubseteq d''$ muss also auch $d'' = (e'', i) \in D'_i \times \{i\}$ sein mit $e' \sqsubseteq_{D_i} e''$. Nach Transitivität von $\sqsubseteq_{D_i}$ folgt dann $e \sqsubseteq_{D_i} e''$ und damit auch $d \sqsubseteq d''$.
Kleinstes Element: Für alle $d \in D_1 \oplus \dots \oplus D_n$ gilt nach Definition bereits, dass $\bot \sqsubseteq d$.
Kleinste obere Schranke: Sei $c \colon \mathbb{N} \to D_1 \oplus \dots \oplus D_n$ eine Kette. Wenn $c(n) = \bot$ für alle $n \in \mathbb{N}$ gilt, dann ist $c' \colon \mathbb{N} \to \{\bot\}$ mit $c'(n) = c(n)$ eine Kette und es gilt $\sqcup c' = \sqcup c$. Wenn es ein $k \in \mathbb{N}$ gibt mit $c(k) \in D'_i \times \{i\}$, dann muss auch für alle $k' > k$ gelten, dass $c(k') \in D'_i \times \{i\}$. Damit ist $c_k \colon \mathbb{N} \to (D_i \times \{i\})$ mit $c_k(n) = c(n + k)$ eine Kette und es gilt $\sqcup c_k = \sqcup c$.

Aufgabe 2. Zeichnen Sie jeweils die Domains zu folgenden Data-Deklarationen. Sie können die „Tags“ von den Summen weglassen.

(a) $\mathcal{D}(\text{Void})$ für

```
data Void
```

Lösung. Hier haben wir bloß $\bot$.

(b) $\mathcal{D}(\text{Unit})$ für

```
data Unit = U
```

Lösung.

$$\begin{tikzcd}
\underline{\mathrm{U}} \arrow[d, no head] \\
\perp
\end{tikzcd}$$

(c) $\mathcal{D}(\text{Test})$ für

```
data Test = A Unit Unit | B Bool
```

Lösung.

$$\begin{tikzcd}[column sep=small]
 & (\underline{\mathrm{A}}, \underline{\mathrm{U}}, \underline{\mathrm{U}}) \arrow[dl, no head] \arrow[dr, no head] & & & & & \\
(\underline{\mathrm{A}}, \underline{\mathrm{U}}, \perp) \arrow[dr, no head] & & (\underline{\mathrm{A}}, \perp, \underline{\mathrm{U}}) \arrow[dl, no head] & & (\underline{\mathrm{B}}, \underline{\mathrm{True}}) \arrow[dr, no head] & & (\underline{\mathrm{B}}, \underline{\mathrm{False}}) \arrow[dl, no head] \\
 & (\underline{\mathrm{A}}, \perp, \perp) \arrow[drr, no head] & & & & (\underline{\mathrm{B}}, \perp) \arrow[dll, no head] & \\
 & & & \perp & & &
\end{tikzcd}$$

Aufgabe 3. Betrachten wir wieder

```
data Test = A Unit Unit | B Bool
```

Geben Sie jeweils Haskell-Ausdrücke zu folgenden Semantiken an:

(a) $\perp$

Lösung.

```
a :: Test
a = undefined
```

(b) $(\underline{\mathrm{A}}, \perp, \underline{\mathrm{U}})$

Lösung.

```
b :: Test
b = A undefined U
```

(c) $(\underline{\mathrm{B}}, \perp)$

Lösung.

```
c :: Test
c = B undefined
```