

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 \perp für $\perp_{D_1 \oplus \dots \oplus D_n}$. Des Weiteren sei $D'_i = D_i \setminus \{\perp_{D_i}\}$ 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 $d = \perp$ 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 $d = \perp$, dann folgt aus $d' \sqsubseteq d$ auch, dass $d' = \perp$, also $d = d'$. Der Fall, dass $d' = \perp$, ist analog. Wenn $d \neq \perp$, 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 $d = \perp$, dann gilt auch $d \sqsubseteq d''$. Sei nun $d \neq \perp$, 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 $\perp \sqsubseteq d$.

Kleinste obere Schranke: Sei $c: \mathbb{N} \rightarrow D_1 \oplus \dots \oplus D_n$ eine Kette. Wenn $c(n) = \perp$ für alle $n \in \mathbb{N}$ gilt, dann ist $c': \mathbb{N} \rightarrow \{\perp\}$ 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 $c_k: \mathbb{N} \rightarrow (D_i \times \{i\})$ 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

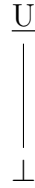
```
data Void
```

Lösung. Hier haben wir bloß \perp .

(b) $\mathcal{D}(\text{Unit})$ für

```
data Unit = U
```

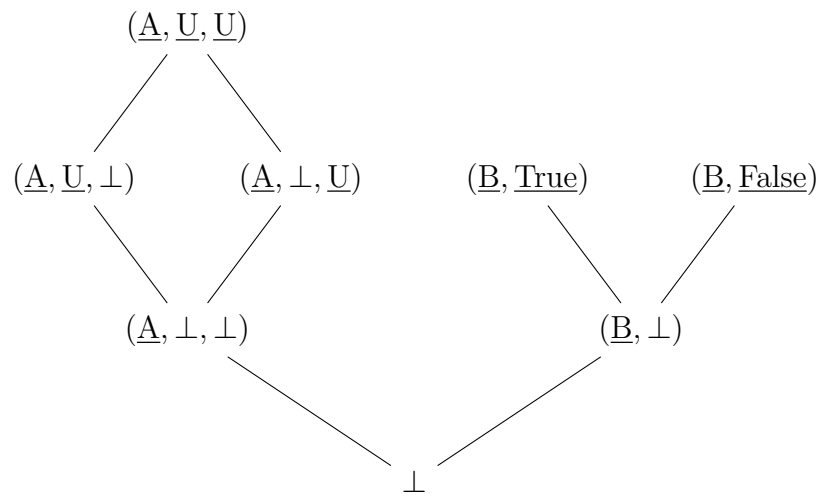
Lösung.



(c) $\mathcal{D}(\text{Test})$ für

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

Lösung.



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) (A, \perp , U)

Lösung.

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

(c) (B, \perp)

Lösung.

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