Praktische Informatik 3: Funktionale Programmierung

Vorlesung 9 (13.12.2022): Übungen

Christoph Lüth

Wintersemester 2022/23

Weitere Eigenschaften endlicher Abbildungen

Übung 9.1: Was denkt ihr?

Überlegt mindestens **drei** weitere Eigenschaften endlicher Abbildungen!

1. Wenn etwas **gelesen** wird an einer **anderen** Stelle, an der etwas **gelöscht** worden ist, kann das Löschen vernachlässigt werden.

Weitere Eigenschaften endlicher Abbildungen

Übung 9.1: Was denkt ihr?

Überlegt mindestens **drei** weitere Eigenschaften endlicher Abbildungen!

1. Wenn etwas **gelesen** wird an einer **anderen** Stelle, an der etwas **gelöscht** worden ist, kann das Löschen vernachlässigt werden.
2. An unterschiedlichen Stellen **geschrieben** kommutiert.

Weitere Eigenschaften endlicher Abbildungen

Übung 9.1: Was denkt ihr?

Überlegt mindestens **drei** weitere Eigenschaften endlicher Abbildungen!

1. Wenn etwas **gelesen** wird an einer **anderen** Stelle, an der etwas **gelöscht** worden ist, kann das Löschen vernachlässigt werden.
2. An unterschiedlichen Stellen **geschrieben** kommutiert.
3. Wenn etwas **geschrieben** wird an der **gleichen** Stelle, an der etwas **gelöscht** worden ist, kann das Löschen vernachlässigt werden.

Weitere Eigenschaften endlicher Abbildungen

Übung 9.1: Was denkt ihr?

Überlegt mindestens **drei** weitere Eigenschaften endlicher Abbildungen!

1. Wenn etwas **gelesen** wird an einer **anderen** Stelle, an der etwas **gelöscht** worden ist, kann das Löschen vernachlässigt werden.
2. An unterschiedlichen Stellen **geschrieben** kommutiert.
3. Wenn etwas **geschrieben** wird an der **gleichen** Stelle, an der etwas **gelöscht** worden ist, kann das Löschen vernachlässigt werden.
4. An der **gleichen** Stelle **zweimal geschrieben** überschreibt der zweite den ersten Wert.

Weitere Eigenschaften endlicher Abbildungen

Übung 9.1: Was denkt ihr?

Überlegt mindestens **drei** weitere Eigenschaften endlicher Abbildungen!

1. Wenn etwas **gelesen** wird an einer **anderen** Stelle, an der etwas **gelöscht** worden ist, kann das Löschen vernachlässigt werden.
2. An unterschiedlichen Stellen **geschrieben** kommutiert.
3. Wenn etwas **geschrieben** wird an der **gleichen** Stelle, an der etwas **gelöscht** worden ist, kann das Löschen vernachlässigt werden.
4. An der **gleichen** Stelle **zweimal geschrieben** überschreibt der zweite den ersten Wert.
5. Wenn etwas **gelöscht** wird an der **gleichen** Stelle, an der etwas **geschrieben** worden ist, kann das Schreiben vernachlässigt werden.

Weitere Eigenschaften endlicher Abbildungen

Übung 9.1: Was denkt ihr?

Überlegt mindestens **drei** weitere Eigenschaften endlicher Abbildungen!

1. Wenn etwas **gelesen** wird an einer **anderen** Stelle, an der etwas **gelöscht** worden ist, kann das Löschen vernachlässigt werden.
2. An unterschiedlichen Stellen **geschrieben** kommutiert.
3. Wenn etwas **geschrieben** wird an der **gleichen** Stelle, an der etwas **gelöscht** worden ist, kann das Löschen vernachlässigt werden.
4. An der **gleichen** Stelle **zweimal geschrieben** überschreibt der zweite den ersten Wert.
5. Wenn etwas **gelöscht** wird an der **gleichen** Stelle, an der etwas **geschrieben** worden ist, kann das Schreiben vernachlässigt werden.
6. An der **gleichen** Stelle **zweimal gelöscht** ist das gleiche wie einmal löschen.

Weitere Eigenschaften endlicher Abbildungen

Übung 9.1: Was denkt ihr?

Überlegt mindestens **drei** weitere Eigenschaften endlicher Abbildungen!

1. Wenn etwas **gelesen** wird an einer **anderen** Stelle, an der etwas **gelöscht** worden ist, kann das Löschen vernachlässigt werden.
2. An unterschiedlichen Stellen **geschrieben** kommutiert.
3. Wenn etwas **geschrieben** wird an der **gleichen** Stelle, an der etwas **gelöscht** worden ist, kann das Löschen vernachlässigt werden.
4. An der **gleichen** Stelle **zweimal geschrieben** überschreibt der zweite den ersten Wert.
5. Wenn etwas **gelöscht** wird an der **gleichen** Stelle, an der etwas **geschrieben** worden ist, kann das Schreiben vernachlässigt werden.
6. An der **gleichen** Stelle **zweimal gelöscht** ist das gleiche wie einmal löschen.
7. **Löschen** aus der **leeren** Abbildung ist die leere Abbildung.

Quick Question

Übung 9.2: Gleichheiten

Betrachtet die letzten beiden Fälle:

$$\texttt{put a w (put a v s)} = \texttt{put a w s}$$

$$\texttt{a} \neq \texttt{b} \implies \texttt{put a v (put b w s)} = \texttt{put b w (put a v s)}$$

Wiese müssen wir die Fälle $\texttt{a} = \texttt{b}$ und $\texttt{a} \neq \texttt{b}$, aber nicht $\texttt{w} = \texttt{v}$ und $\texttt{w} \neq \texttt{v}$ unterscheiden?

Quick Question

Übung 9.2: Gleichheiten

Betrachtet die letzten beiden Fälle:

$$\texttt{put a w (put a v s)} = \texttt{put a w s}$$

$$\texttt{a} \neq \texttt{b} \implies \texttt{put a v (put b w s)} = \texttt{put b w (put a v s)}$$

Wiese müssen wir die Fälle $\texttt{a} = \texttt{b}$ und $\texttt{a} \neq \texttt{b}$, aber nicht $\texttt{w} = \texttt{v}$ und $\texttt{w} \neq \texttt{v}$ unterscheiden?

Lösung: Im Gegensatz zu a und b gelten beide Axiome sowohl für $\texttt{w} = \texttt{v}$ als auch für $\texttt{w} \neq \texttt{v}$:

$$\texttt{put a w (put a w s)} = \texttt{put a w s}$$

$$\texttt{a} \neq \texttt{b} \implies \texttt{put a w (put b w s)} = \texttt{put b w (put a w s)}$$

Was stimmt hier nicht?

Übung 9.3: `Map` als balancierte Bäume.

Warum ist diese Implementierung von `Map` als binärer Baum falsch?

```
data Map α β = Empty
             | Node α β Int (Map α β) (Map α β)
             deriving Eq
```

Was stimmt hier nicht?

Übung 9.3: Map als balancierte Bäume.

Warum ist diese Implementierung von `Map` als binärer Baum falsch?

```
data Map α β = Empty
             | Node α β Int (Map α β) (Map α β)
             deriving Eq
```

Lösung: Weil die abgeleitete Gleichheit nicht die beobachtbare Gleichheit ist. Die Gleichheit darf nur prüfen, ob die gleichen Schlüssel/Wert-Paare enthalten sind:

```
toList :: Map α β → [(α, β)]
toList = fold (λk x l r → l++[(k,x)]++r) []
```

```
instance (Eq α, Eq β)⇒ Eq (Map α β) where
  t1 == t2 = toList t1 == toList t2
```

Implementation: Gleichheit

Übung 9.4:

Warum reicht für Gleichheit auf Schlangen nicht `derive Eq` und wie implementieren wir es dann?

Implementation: Gleichheit

Übung 9.4:

Warum reicht für Gleichheit auf Schlangen nicht `derive Eq` und wie implementieren wir es dann?

Lösung:

- Gegenbeispiel:

$$q_1 = \texttt{deq (enq 7 (enq 4 (enq 9 empty)))}, q_2 = \texttt{enq (7 (enq 4 empty))}$$

- Zwei Schlangen sind gleich, wenn der **Inhalt gleich** ist:

```
instance Eq α⇒ Eq (Qu α) where
  Qu xs1 ys1 == Qu xs2 ys2 =
    xs1 ++ reverse ys1 == xs2 ++ reverse ys2
```