

## 2. Übungsblatt

Ausgabe: 09.12.09

Abgabe: 13.01.10

---

### 7 Bäume zum ersten

5 Punkte

Für den Datentyp der binären Bäume `'a tree` definieren Sie folgende Operationen:

```
map :: "('a => 'b) => 'a tree => 'b tree"
```

```
nodes :: "'a tree => 'a set"
```

```
inorder :: "'a tree => 'a list"
```

Zeigen Sie folgende Eigenschaften:

1. `map f` angewandt auf einen Baum `t` ist dasselbe wie die Funktion `f` angewandt auf jedes Element von `nodes t`.
2. Die Länge der durch Inorder-Traversierung eines Baumes `t` entstehenden Liste ist gleich der Größe von `t`.

### 8 Geordnete Bäume

15 Punkte

Ein binärer Baum `t` ist *geordnet*, wenn er entweder leer ist, oder wenn alle Knoten des linken Unterbaums kleiner (oder gleich) sind als der Wert des Wurzelknotens, der Wurzelknoten kleiner (oder gleich) allen Knoten des rechten Unterbaums ist, und linker und rechter Unterbaum wiederum geordnet sind.

1. Formalisieren Sie diese Definition als ein (rekursives) Prädikat

```
ordered :: "'a tree => bool"
```

2. Definieren Sie Operationen

```
ins :: "'a=> 'a tree=> 'a tree"
```

```
del :: "'a=> 'a tree=> 'a tree"
```

welche ein Element in einen geordneten Baum einfügen bzw. löschen.

3. Spezifizieren Sie die *Korrektheit* der Operationen: informell sollen beide die Ordnung des Baumes bewahren, und `ins` den Knoten zu der Menge der Knoten hinzugefügen, und `del` das Argument aus der Menge der Knoten entfernen.
4. Zeigen Sie die Korrektheit einer der beiden Operationen.