

## Aufgabe 6 Tiefenentspannung mittels Funktor

[22 Punkte]

In dieser Aufgabe soll die Tiefensuche auf Bäumen und (sonstigen) Graphen mittels eines Funktors implementiert werden. Zur Erinnerung sei zunächst die Definition der Tiefensuche gegeben.

**Tiefensuche:** Die Tiefensuche besucht die Knoten eines Graphen ausgehend von einem Startknoten. Dazu folgt sie Kanten zu den Nachfolgern des Startknotens. Sie besucht dabei jeweils zunächst den Startknoten und anschließend rekursiv die Knoten, die vom ersten Nachfolger des Startknotens aus erreichbar sind. Anschließend fährt sie mit dem zweiten Nachfolger des Startknotens fort usw. Um Terminierung zu sichern, besucht die Tiefensuche jeden Knoten höchstens einmal.

Die Signatur eines Graphen, der Integer-Zahlen (ohne Duplikate) in seinen Knoten hält, habe hier die folgende Signatur:

```
1 module type Graph = sig
2   type node
3   type graph
4
5   (* Gibt die Liste direkter Nachfolger des gegebenen Knotens zurück
6      ↪ *)
7   val successors : graph -> node -> node list
end
```

Bearbeiten Sie nun folgende Teilaufgaben.

1. Implementieren Sie ein Modul **BinaryTree**, welches die obige Signatur umsetzt (Sie müssen nur die von der Signatur geforderten Typen und Funktionen implementieren). Das Modul soll als Graphstruktur einen binären Baum verwenden, wobei Teilbäume jeweils in den Knoten gespeichert werden.
2. Implementieren Sie ein Modul **GraphImpl**, welches die obige Signatur umsetzt (Sie müssen nur die von der Signatur geforderten Typen und Funktionen implementieren). Das Modul soll einen gerichteten Graphen implementieren, der Kanten zwischen beliebigen Knoten erlaubt. Nutzen Sie Listen, um Kanten bzw. Knoten zu speichern.
3. Implementieren Sie den Funktor **MakeGraphSearch**, der, gegeben ein Modul **G** der Signatur **Graph**, die Tiefensuche implementiert. Dazu soll der Funktor die Funktion **val dfs : G.graph -> G.node -> ('a -> G.node -> 'a) -> 'a -> 'a** anbieten, die über einen Graphen (1. Parameter), ausgehend von einem Startknoten (2. Parameter) eine Funktion **f** (3. Parameter) mittels einer Tiefensuche faltet. Der Startwert der Faltung ist dabei der letzte Parameter. Die Funktion **f** soll die Schlüssel der Knoten in der Reihenfolge ihrer Entdeckung bearbeiten. **Hinweis:** Speichern Sie bereits bekannte Knoten in einer Liste, um mehrfaches Besuchen auszuschließen!