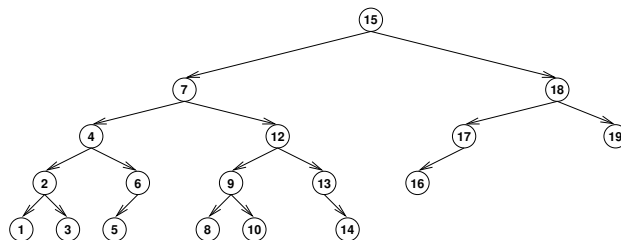


Übungen zu Einführung in die Informatik II

**Aufgabe 15**      **AVL-Bäume**

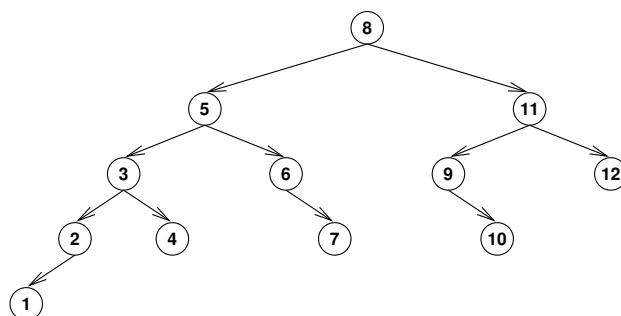
AVL-Bäume sind Datenstrukturen, die sich dadurch auszeichnen, dass sie Höhenbalancierung garantieren und damit eine logarithmische Komplexität für Suchen, Einfügen und Löschen gewährleisten. Bei den Operationen Einfügen und Löschen kann es allerdings zu einer kurzfristigen Störung der Balancierungeigenschaft kommen. Diese werden durch Rotationen bereinigt.

a) Gegeben sei folgender AVL-Baum:



- (i) In diesen Baum wird nun ein Knoten mit Inhalt 0 eingefügt. Balancieren Sie diesen Baum.
- (ii) Nehmen wir an, wir fügen in obigen Baum einen Knoten mit Inhalt 11 ein. An welcher Stelle im Baum tritt nun die Verletzung der AVL-Eigenschaft auf? Machen Sie sich klar, warum eine Einfachrotation hier nicht zum Erfolg führt. Welche Operation müssen wir zur Wiederherstellung der AVL-Eigenschaft stattdessen anwenden?

b) Gegeben sei nun folgender AVL-Baum:



- (i) Welche spezielle Eigenschaft besitzt dieser Baum?
- (ii) Aus diesem Baum soll nun der Knoten mit dem Inhalt 12 gelöscht werden. Welche Auswirkungen hat dies auf den Baum und wie kann die AVL-Eigenschaft wiederhergestellt werden?

**Aufgabe 16**      **AVL-Bäume**

Für den aus der Vorlesung bekannte Daten-Typ

```
type 'a avl =  
  Null  
| Pos of ('a avl * 'a * 'a avl)  
| Neg of ('a avl * 'a * 'a avl)  
| Eq of ('a avl * 'a * 'a avl)
```

zur Repräsentation von AVL-Bäumen sollen folgende Funktionen definiert werden:

- a) Eine Funktion `list2avl : 'a list -> 'a avl`, die eine Liste als Parameter erhält und einen AVL-Baum mit den in der Liste enthaltenen Elementen konstruiert und zurückliefert.
- b) Eine Funktion `find : 'a avl -> 'a -> bool`, die als Parameter einen AVL-Baum `avl` sowie ein Element `e` erhält und `true` zurückliefert, falls das Element `e` im AVL-Baum enthalten ist. Andernfalls soll `false` zurückgeliefert werden.
- c) Eine Funktion `extract_one : 'a * 'a -> 'a avl -> 'a option * 'a avl * bool`, die ein beliebiges Element `e` aus einem AVL-Baum `avl` extrahiert, wobei `e` aus dem als Parameter übergebenen Intervall `(a,b)` sein muss. Zurückgeliefert werden soll, ähnlich wie bei der Funktion `extract_min`, das unter Umständen extrahierte Element, der neue AVL-Baum sowie die Information, ob der neue AVL-Baum flacher ist als der alte.
- d) Eine Funktion `extract : 'a * 'a -> 'a avl -> 'a list * 'a avl`, die aus einem AVL-Baum `avl` alle Elemente eines Intervalls `(a,b)` extrahiert.
- e) (Optional) Eine Funktion `check : 'a avl -> bool`, die überprüft, ob ein Baum ein AVL-Baum ist.

Zur Lösung dieser Aufgabe werden die in der Vorlesung eingeführten Funktionen

```
rotateRight : 'a avl * 'a * 'a avl -> 'a avl * bool  
rotateLeft : 'a avl * 'a * 'a avl -> 'a avl * bool  
insert : 'a -> 'a avl -> 'a avl  
extract_min : 'a avl -> 'a option * 'a avl * bool  
extract_max : 'a avl -> 'a option * 'a avl * bool
```

benötigt. Eine Datei mit den entsprechenden Definitionen wird auf der Seite zur Übung zur Verfügung gestellt.