Prof. Dr. Gerd Stumme, Folke Eisterlehner, Dominik Benz
Fachgebiet Wissensverarbeitung

16/18.06.2009

9. Präsenzübung „Algorithmen und Datenstrukturen“

Sommersemester 2009

1 Binärbäume

Aufgabe 1 (Binärbäume)

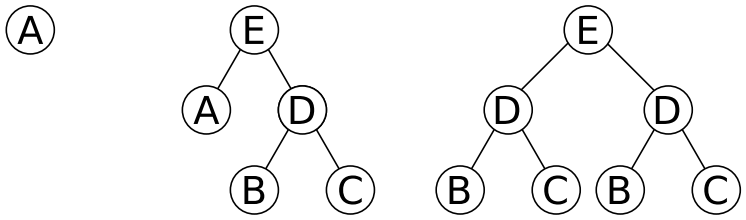
Sei $\mathcal{S}$ eine abzählbare Menge von Schlüsselwerten. Die Menge $\mathsf{BB}_{\mathcal{S}}$ der *Binärbäume* über $\mathcal{S}$ ist wie folgt definiert:

(B) Jeder Schlüssel $k \in \mathcal{S}$ ist ein Binärbaum T der Höhe $\mathrm{h}(T) := 1$ mit Wurzel k (k ist ein *äußerer Knoten* oder auch *Blatt*).

(R) Sind T_1, T_2 Binärbäume der Höhen h_1 bzw. h_2 und ist $\ell \in \mathcal{S}$ ein Schlüssel, so ist $T := (\ell, T_1, T_2)$ ein Binärbäum der Höhe $\mathrm{h}(T) := 1 + \max\{\mathrm{h}(T_1), \mathrm{h}(T_2)\}$ mit Wurzel ℓ (ℓ ist ein *innerer Knoten* und *direkter Vorgänger der Wurzelknoten von T_1 und T_2*).

(A) Nichts sonst ist ein Binärbaum.

Beispiele für Binärbäume sind A, $(\mathsf{E}, \mathsf{A}, (\mathsf{D}, \mathsf{B}, \mathsf{C}))$ und $(\mathsf{E}, (\mathsf{D}, \mathsf{B}, \mathsf{C}), (\mathsf{D}, \mathsf{B}, \mathsf{C}))$. Analog zu Bäumen aus Termen, lassen sich Binärbäume graphisch darstellen:



Als *Tiefe* eines Knotens v in einem Binärbaum T bezeichnet man die Anzahl von „übergeordneten Knoten“ von v, d.h. die Anzahl von Vorgängern. Ein Binärbaum T heißt *vollständig*, falls alle Blätter in T die gleiche Tiefe haben. Im obigen Beispiel hat jeder mit B beschriftete Knoten die Tiefe 2 und jeder mit D beschriftete Knoten die Tiefe 1. Der erste und der letzte Binärbaum sind vollständig.

Sei eine beliebige natürliche Zahl $h > 0$ gegeben.

a) Definieren Sie in Pseudocode mittels Rekursion eine zweistellige Funktion **`isEqual`**, so dass für alle Binärbäume T_1 und T_2 gilt:

$$\texttt{isEqual}(T_1, T_2) = \begin{cases} \texttt{true}, & \text{falls } T_1 \text{ und } T_2 \text{ gleich sind,} \\ \texttt{false}, & \text{sonst} \end{cases}$$

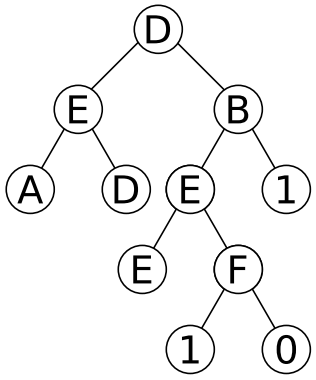
b) Wie viele Blätter hat ein *vollständiger* Binärbaum der Höhe h?

c) Wie viele innere Knoten hat ein *vollständiger* Binärbaum der Höhe h?

d) Wie viele Knoten (innere und äußere) hat ein beliebiger Binärbaum *maximal*?

Aufgabe 2 (Traversierung von Bäumen)

Betrachten Sie folgenden graphisch dargestellten Baum:



a) Geben Sie die Ausgabe der *Preorder*-Traversierung des Baumes an.

b) Geben Sie die Ausgabe der *Postorder*-Traversierung des Baumes an.

c) Geben Sie die Ausgabe der *Inorder*-Traversierung des Baumes an.

Aufgabe 3 (Blattbäume)

Meist wird bei der Definition von Binärbäumen lediglich gefordert, dass jeder Knoten *maximal* zwei Nachfolger besitzt. Nach obiger Definition hat jeder innere Knoten *genau zwei* Nachfolger. Wieso ist dies keine Einschränkung?

Beschreiben Sie, wie sich ein beliebiger Binärbaum mit *maximal zwei* Nachfolgern pro Knoten durch einen Baum mit *genau zwei* Nachfolgern pro innerem Knoten darstellen lässt.

2 Generics

Gegeben seien folgende Klassen:

```
class Bike {}
class MotorBike extends Bike {}
class Trike extends Bike {}

class Garage<T> {}
class BasementGarage<T> extends Garage<T>
class ParkingLot<T extends Bike>
```

Welche der folgenden Anweisungen sind in Java zulässig, welche nicht? Begründen Sie für die *nicht* zulässigen Anweisungen, warum sie nicht zulässig sind.

```
1. Garage<Bike> petersGarage = new Garage<Bike>();

2. Garage<Bike> johnsGarage = new Garage<MotorBike>();

3. Garage marysGarage = new Garage();

4. Garage<?> paulsGarage = new BasementGarage<Integer>();

5. BasementGarage<? extends Bike> mortensGarage = new Garage<Trike>();

6. ParkingLot joesParkingLot = new ParkingLot<MotorBike>();

7. ParkingLot<Object>  susisParkingLot = joesParkingLot; // aus vorheriger Zeile
```