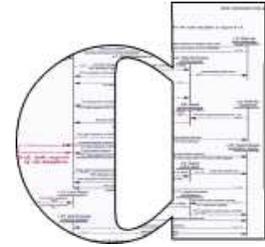


# DEPENDABLE SYSTEMS AND SOFTWARE

Fachrichtung 6.2 — Informatik  
Prof. Dr.-Ing. Holger Hermanns  
Dipl.-Inform. Lijun Zhang



## Übungsblatt 7 (Programmierung I)

---

Lesen Sie im Skript: Kapitel 7

---

### Aufgabe 7.1: (Baumdarstellungen)

Geben Sie Ausdrücke an, die Bäume mit den unten gezeigten grafischen Darstellungen beschreiben. Dabei können Sie die deklarierten Bezeichner  $t1$  und  $t2$  in §7.1 (Abbildung 7.1) verwenden.



### Aufgabe 7.2: (Baumeigenschaften)

Sei die grafische Darstellung eines Baums gegeben. Nehmen Sie an, dass die Darstellung  $n \geq 1$  Kanten enthält und beantworten Sie die folgenden Fragen:

- Wie viele Knoten enthält die Darstellung mindestens/höchstens?
- Wie viele Blätter enthält die Darstellung mindestens/höchstens?
- Wie viele innere Knoten enthält die Darstellung mindestens/höchstens?

### Aufgabe 7.3: (Lexikalische Ordnung)

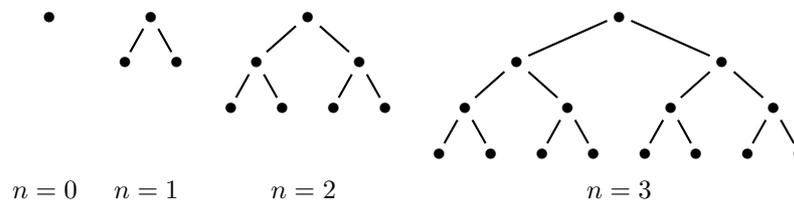
Zeichnen Sie die Darstellung der Bäume  $T[T[t1]]$ ,  $T[t1, T[T[t1]]]$ ,  $T[T[T[T[t1]]]]$ , und ordnen Sie sie und die Bäume in Aufgabe 7.1 gemäß der lexikalischen Ordnung an.

### Aufgabe 7.4: (Bäume)

Geben Sie einen Baum mit 5 Knoten an, der genau zwei Teilbäume hat. Wieviele solche Bäume gibt es?

### Aufgabe 7.5: (Binäre Bäume)

Schreiben sie eine Prozedur  $tree : int \rightarrow tree$  die für  $n \geq 0$  binäre Bäume wie folgt liefert:



Achten Sie darauf, dass die identischen Unterbäume der zweistelligen Teilbäume jeweils nur einmal berechnet werden. Das sorgt dafür, dass ihre Prozedur auch für  $n = 1000$  schnell ein Ergebnis liefert. Verwenden Sie die Prozedur `iter` aus §3.9

**Aufgabe 7.6: (Adressen bei Bäumen)**

Betrachten Sie die graphische Darstellung des Baums  $t_3$  in Abbildung 7.1. (§7)

- (a) Geben Sie die Adresse der Wurzel an.
- (b) Geben Sie die Adressen der inneren Knoten an (es gibt genau 4).
- (c) Geben Sie die Adressen der Auftreten des Teilbaums  $t_2$  an.

**Aufgabe 7.7: (Listenpräfix)**

Schreiben Sie eine Prozedur  $prefix : "a list \rightarrow "a list \rightarrow bool$ , die testet, ob eine Liste ein Prefix einer Liste ist.

**Aufgabe 7.8: (Rekursion in Bäumen)**

Schreiben Sie eine Prozedur

- (a)  $node : tree \rightarrow int list \rightarrow bool$ , die testet, ob es sich bei einer Adresse um einen Knoten eines Baums handelt.
- (b)  $leaf : tree \rightarrow int list \rightarrow bool$ , die testet, ob es sich bei einer Adresse um ein Blatt eines Baums handelt.
- (c)  $father : int list \rightarrow int list$ , die den Vater eines Knotens liefert. Wenn es sich bei dem Knoten um die Wurzel handelt, soll die Ausnahme Subscript geworfen werden.
- (d)  $son : tree \rightarrow int list \rightarrow int \rightarrow int list$ , die den  $k$ -ten Sohn eines Knotens liefert. Wenn der Knoten keinen  $k$ -ten Sohn hat, soll die Ausnahme Subscript geworfen werden.

Beachten Sie, dass der zugrunde liegende Baum bei den obigen Prozeduren genau dann als Argument übergeben wird, wenn er für die Bestimmung des Ergebnisses erforderlich ist.

**Aufgabe 7.9: (Breite eines Baums)**

Die Breite eines Baums ist die Anzahl seiner Blätter. Entwickeln Sie eine Prozedur  $breadth : tree \rightarrow int$ , die die Breite eines Baums bestimmt.

**Aufgabe 7.10: (Grad eines Baumes)**

Der Grad eines Baumes ist die maximale Stelligkeit seiner Teilbäume. Beispielsweise hat der Baum  $t_3$  den Grad 3. Entwickeln Sie eine Prozedur  $degree : tree \rightarrow int$ , die den Grad eines Baumes bestimmt.

**Aufgabe 7.11: (Arithmetische Ausdrücke)**

Schreiben Sie für die arithmetischen Ausdrücke aus § 6.4 Prozeduren  $size : exp \rightarrow int$  und  $depth : exp \rightarrow int$ , die die Größe und die Tiefe von Ausdrücken liefern. Beispielsweise sollen  $size$  und  $depth$  für den Ausdruck  $x(2y + 3)$  die Ergebnisse 7 und 3 liefern.

**Aufgabe 7.12: (Ausdrücke)**

Die arithmetischen Ausdrücke aus §6.4 kann man als binäre Bäume auffassen. Dabei liefern die Konstruktoren  $C$  und  $V$  atomare und die Konstruktoren  $A$  und  $M$  zusammengesetzte Bäume.

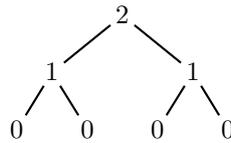
- (a) Schreiben Sie eine Prozedur  $prelin : exp \rightarrow int list list$ , die die Adressen eines Audrucks in Präordnung liefert.
- (b) Schreiben Sie eine Prozedur  $postlin : exp \rightarrow int list list$ , die die Adressen eines Audrucks in Postordnung liefert.

**Aufgabe 7.13: (Prä- und Postordnung)**

Schreiben Sie zwei Prozeduren  $prelin$  und  $postlin$  des Typs  $tree \rightarrow int list list$ , die die Adressen eines Baums in der Reihenfolge der Prä- beziehungsweise Postordnung liefern.

**Aufgabe 7.14: (Binärbäume mit Tiefe)**

Schreiben Sie eine Prozedur  $ltr : int \rightarrow int$  *ltr*, die zu  $n \geq 0$  einen balancierten Binärbaum der Tiefe  $n$  liefert, dessen Teilbäume mit ihrer Tiefe markiert sind. Für  $n = 2$  soll *tree* den Baum



liefern. Achten Sie darauf, dass die identischen Unterbäume der zweistelligen Teilbäume jeweils nur einmal berechnet werden. Verwenden Sie die Prozedur *iter* aus §3.9.

**Aufgabe 7.15: (lmap)**

Schreiben Sie eine Prozedur  $lmap : (\alpha \rightarrow \beta) \rightarrow \alpha$  *ltr*  $\rightarrow \beta$  *ltr*, die eine Prozedur auf alle Marken eines Baumes anwendet. Verwenden Sie die Prozedur *map* für Listen.

**Aufgabe 7.16: (Find)**

Schreiben Sie eine Prozedur  $find : (\alpha \rightarrow bool) \rightarrow \alpha$  *ltr*  $\rightarrow \alpha$  *option*, die zu einer Prozedur und einem Baum die gemäß der Präordnung erste Marke des Baums liefert, für die die Prozedur *true* liefert. Orientieren Sie sich dabei an der Prozedur *prest* in §7.5.1.

**Aufgabe 7.17: (Spiegeln von Bäumen)**

Spiegeln reversiert die Ordnung der Unterbäume der Teilbäume eines Baums:



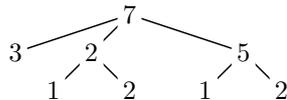
Schreiben Sie eine Prozedur  $mirror : \alpha$  *ltr*  $\rightarrow \alpha$  *ltr*, die markierte Bäume spiegelt.

**Aufgabe 7.18: (Grenze eines Baums)**

Die Grenze eines markierten Baums ist die Liste der Marken seiner Blätter, in der Ordnung ihres Auftretens von links nach rechts und mit Mehrfachauftreten. Die Grenze des Baums *t3* ist  $[2, 7, 7, 4, 2, 7, 7]$ . Schreiben Sie eine Prozedur  $frontier : \alpha$  *ltr*  $\rightarrow \alpha$  *list*, die die Grenze eines Baums liefert.

**Aufgabe 7.19: (Ebenen)**

Die  $n$ -te Ebene eines markierten Baums ist die Liste der Marken der Knoten der Tiefe  $n$ , angeordnet entsprechend der Anordnung der Knoten. Als Beispiel betrachten wir den Baum



Die nullte Ebene ist  $[7]$ , die erste Ebene ist  $[3, 2, 5]$ , die zweite Ebene ist  $[1, 2, 1, 2]$ , und die dritte und alle nachfolgenden Ebenen sind  $[\ ]$ . Schreiben Sie eine Prozedur  $level : \alpha$  *ltr*  $\rightarrow int \rightarrow \alpha$  *list*, die die  $n$ -te Ebene eines Baums liefert.