

Programmierung 1 (Wintersemester 2012/13)

Aufgaben aus den Übungsgruppen 7 (Kapitel 7)

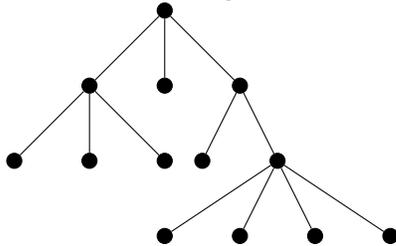
Hinweis: Dieses Übungsblatt enthält von den Tutoren für die Übungsgruppe erstellte Aufgaben.

Die Aufgaben und die damit abgedeckten Themenbereiche sind für die Klausur weder relevant, noch irrelevant.

1 Baumschule

Aufgabe 7.1 (*So viele Begriffe!*)

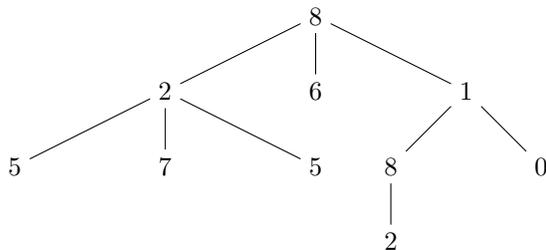
Betrachten Sie den folgenden Baum:



- Welche Größe hat dieser Baum? Welchen Grad, welche Tiefe, welche Stelligkeit, welche Breite? Wie viele innere Knoten besitzt dieser Baum?
- Malen Sie den gespiegelten Baum auf!
- Markieren Sie den zweiten Nachfolger der Wurzel! Welche Knoten sind der Wurzel übergeordnet?
- Wie viele Teilbäume besitzt dieser Baum? Wie viele Unterbäume?
- Geben Sie jeweils die Prä- und Postnummerierung des obigen Baumes an!
- Fast geschafft!! Wie lauten Prä- und Postlinearisierung?

Aufgabe 7.2 (*Markierte Bäume*)

Wir betrachten nun markierte Bäume.

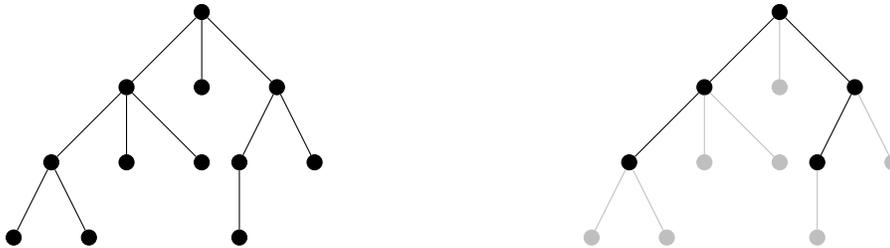


- Was ist die Gestalt dieses markierten Baumes?
- Wie lautet der Kopf des Baums?

- (c) Geben Sie Prä- und Postprojektion an!
- (d) Wie lautet die Grenze dieses Baums?
- (e) Geben Sie die erste Ebene an!

Aufgabe 7.3 (*Der Wald brennt*)

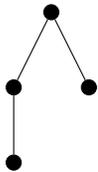
- (a) Der Baum brennt. Nach dem Feuer sind alle Blätter des Baumes verschwunden, nur noch die verkohlten Äste sind übrig. Schreiben sie eine Prozedur $feuer : tree \rightarrow tree$, die alle Blätter eines Baumes entfernt. Lediglich der atomare Baum darf unverändert bleiben. Beispiel:



- (b) Hat der Baum nach einer Anwendung von $feuer$ noch Blätter? Ist es möglich, dass der Baum nach einer Anwendung mehr Blätter als vorher hat?
- (c) Wie oft müssen sie $feuer$ anwenden, um den Baum komplett zu zerstören? Der Baum ist komplett zerstört, wenn nur noch die Wurzel übrig ist, also $T[]$. Welche wichtige Größe spielt hier mit?
- (d) Schreiben sie eine Prozedur, die mithilfe von $feuer$ diese Größe bestimmt!

Aufgabe 7.4 (*Pre- und Postordnung*)

- (a) Schreiben sie zwei Prozeduren $preorder : tree \rightarrow treelist$ und $postorder : tree \rightarrow treelist$, die alle Teilbäume eines Baumes in Pre- bzw. Postordnung ausgeben. Das Beispiel wird zu $[T[T[T[]], T[]] , T[T[]] , T[] , T[]]$ in Postordnung. Sie können sich an der Aufgabe zur Pre/Postlinearisierung orientieren.



- (b) Schreiben sie mithilfe dieser Prozeduren zwei Prozeduren $pre : tree \rightarrow int \rightarrow tree$ und $post : tree \rightarrow int \rightarrow tree$, die den n -ten Teilbaum eines Baumes in Pre- bzw. Postordnung liefern. Die Prozeduren sollen äquivalent zu denen aus Kapitel 7.6.1 (Teilbaumzugriff) sein.

2 Knochelecke

Aufgabe 7.5 (*Bäume bauen*)

Dieter hat gelernt, dass ein Baum durch die Menge seiner möglichen Adressen beschrieben werden kann. Da ihm diese Darstellung besser gefällt, sind alle seine Bäume vom Typ $int\ list\ list$. Damit sie mit Dieters Bäumen etwas anfangen können, müssen sie einen Baum aus einer Menge von Adressen konstruieren. Da Dieter ordentlich ist können sie davon ausgehen, dass sie nur gültige Bäume bekommen und dass die Adressen strikt sortiert sind. Beispiel für eine solche Liste:

[[], [1], [1,1], [1,2], [2]]

Schreiben sie eine Prozedur $buildTree : int\ list\ list \rightarrow tree$, mit der sie Dieters Baumdarstellung in die ihnen vertraute Darstellung konvertieren können!

Aufgabe 7.6 (*Baumschüler Dieter Schlau, heute mal ganz linear*)

- (a) Schreiben Sie eine Prozedur $postlin : tree \rightarrow int\ list$, die die Postlinearisierung eines Baumes bestimmt.
- (b) Und nun zurück: Schreiben Sie eine Prozedur $parsePostLin : int\ list \rightarrow tree$, die aus einer Postlinearisierung den Baum rekonstruiert.
- (c) Für Wahnsinnige:

Schreiben Sie Prozeduren $encode : tree \rightarrow int$ und $decode : int \rightarrow tree$, so dass für alle Bäume t gilt: $decode\ (encode\ t) = t$

Nutzen Sie dabei, dass es in SML keine obere Grenze für $ints$ gibt.

Aufgabe 7.7 (*Für Experten mit viel zu viel Zeit*)

Die Baumdarstellung des Interpreters ist für Menschen nur sehr schlecht lesbar. Schreiben sie eine Prozedur $printTree : tree \rightarrow unit$, die einen Baum lesbar im Interpreter ausgibt. Verwenden sie dazu die Prozedur $print : string \rightarrow unit$, die einen String ausgibt.