***Aufgabe 1***

Code war gegeben. Man musste Signatur bestimmen.

***Aufgabe 2 (12 Punkte)***

Es sei die Liste xs = (list x1 x2 x3 … xn) gegeben. Wir betrachten hier den sog. Left scan (scan l), der bei einem Aufruf (scan z f xs) zu folgender Liste führt:

(Liste z

(f z x1)

(f (f z x1) x2))

…

(f (f (…(f z x1) …) xn-1) xn))

Bsp.:

(scan l z f empty) liefert (list z)

(scan l o t (list 1 2 3 4) liefert (list 0 1 2 3 6 1 0)

a) Bestimme die Signatur
b) Implementiere scan l

***Aufgabe 3 (10 Punkte)***

Induktion über Listen

P(xs) = (sum (map (lambda (x) 1) xs)) = (length xs)

***Aufgabe 4 (12 Punkte)***

a) Take, drop, welches ist nicht endrekursiv, unterstreiche den Kontext. (2 Punkte)
b) Welche Konsequenzen hat ein Aufruf einer nicht endrekursiven Funktion. (1 Punkt)
c) Implementiere eine enrekursive Funktion. (9 Punkte)

(:split (list-of %a) (list-of natural -> (list-of(list-of %a))))

Folgendes Verhalten:

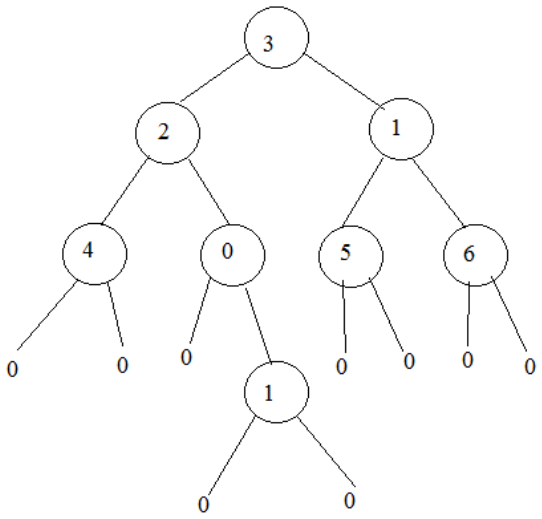
(split (list „a“ „b“ „c“ „d“ „e“) (list 203))
(list (list „a“ „b“) empty(list „c“ „d“ „e“))

(split (list „a“ „b“ „c“ „d“ „e“) empty)
Empty

Funktion über listen nutzen, take & drop.

***Aufgabe 5 (12 Punkte)***

a) Verhalten sich leaf und leaf1 gleich? Begründen Sie. (1 Punkt)
b) Cheapset-path (mit Signatur) welches aus einem Binärbaum der Signatur (btree of real) die Liste der Markierung des billigsten Pfades extrahiert. Ein Weg von der Wurzel zu einem Blatt ist der billigste Pfad, wenn die Summe der Knotenmarkierung auf dem Pfad minimal ist. Cheapset-path liefert also (list 3 2 0 1). (list 2 0 1) ist billigster Pfad im Teilbaum mit Wurzel 2 Signatur ist (btree-of %a). Benutze (:sum ((list-of number) -> number))



***<u>Aufgabe 6 (6 Punkte)</u>***

Rekursion

(FST ((PAIR A) B )) → ß A

PAIR = (λx.( λy.( λs.((s x ) y))))

FST = (λp.(p(λx.( λy.x))))