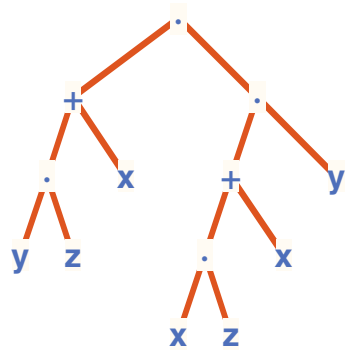
1. a) (2p) Vilket aritmetiskt uttryck beskrivs av nedanstående binära träd?



b) (3p) Räkna upp trädets noder i preordning, inordning och postordning.
c) (1p) Konstruera med hjälp av *Add* och *Mult* en funktion som beräknar uttrycket i a)

LÖSNING

a) $(y\,z + x)\,(x\,z\, +\, x)\,y$

b) Preordning: $\cdot\ +\ \cdot\ y\ z\ x\ \cdot\ +\ \cdot\ x\ z\ x\ y$

Inordning: $y\ \cdot z\ +\ x \cdot x\ \cdot\ z + x \cdot y$

Postordning: $y\,z\,\cdot\ x\ +\ x\ z\ \cdot\ x\ +\ y\ \cdot\ \cdot$

c) $Mult(Add(Mult(y, z), x), Mult(Add(Mult(x, z), x), y))$

2. Vilket innehåll får s då följande procedur körs med
a) (2p) $x = 0, 1, 2, 3$
b) (3p) godtyckligt x. (Glöm inte bort att motivera ditt svar!)

$y \longleftarrow 0$
$k \longleftarrow 0$
Sålänge $y < x$ {
Multiplicera 3 och k *i* y
Öka k}
$y = x$? *i* s

LÖSNING

a) s får innehåll 1,0,0,1 Visas med provkörningar.

b) Proceduren undersöker om x är delbart med 3. Dvs s får innehållet 1 om x är något av talen $\{0,\ 3,\ 6,\ 9,\ 12,\ \ldots\}$, och 0 annars.
MOTIVERING: Innan första besöket i slingan är y:s innehåll lika med 0 som är det minsta talet i treans tabell. Vid varje besök i slingan ändras y-högens innehåll till att bli nästkommande tal i treans tabell. Om vi bortser från sista besöket i slingan produceras på detta sätt samtliga tal i treans tabell som är mindre än x. Vid sista besöket blir y endera lika stor som x eller större än x. (Om inte det vore fallet så skulle ju nämnda besök inte vara det sista, eller hur!) Sammantaget innebär detta att om x råkar vara ett tal i treans tabell, så stannar slingan efter att detta tal har producerats i y-högen, och om x inte är ett sådant tal, så stannar slingan efter att ha producerat det tal som är det närmsta större talet än x i treans tabell. När slingan är färdigkörd placerar likhetstestaren en 1.a i s om likhet gäller, och en 0:a i s annars.

3. (6p) Beskriv (med motiveringar förstås) vad funktionen g returnerar.

$$g(\ddot{O}ka(x)) = h(\ddot{O}ka(x), x)$$
$$\begin{cases} h(x, 0) = 0 \\ h(x, \ddot{O}ka(y)) = Om(Delbar(x, \ddot{O}ka(y)),\ Add(h(x, y), \ddot{O}ka(y)),\ h(x, y)) \end{cases}$$

LÖSNING

g returnerar summan av alla naturliga tal som är mindre än input och som går jämnt upp i input. T ex returneras summan $1+2+3$ då input är 6 och summan $1+3$ då input är 9.

MOTIVERING: När g körs på tex 9 så startar h med anropet $h(9, 8)$. Hjälpfunktionen h har sitt andra argument som primitivt rekurserande argument. Det betyder att när $h(9, 8)$ anropas, så bottnar rekursionen efter 8 anrop. Vid varje rekurivt anrop testas (i detta fall) om 9 är delbart med det högra argumentet. Om så är fallet adderas högra argumentet:

$$h(9, 8) = h(9, 7) = h(9, 6) = h(9, 5) = h(9, 4) = h(9, 3) = h(9, 2) + 3$$
$$= h(9, 1) = h(9, 0) + 1 + 3 = 0 + 1 + 3$$

4. (5p) Konstruera en listfunktion som för varje enkel inputlista av tal returnerar samma lista bakvänd.
Tex skall funktionen returnera [3 5 1] för det fall att inputlistan är [1 5 3].

LÖSNING

Se i kursboken EXEMPEL 3.8 samt ÖVNING 3.8.

5. (5p) Vad beräknar följande funktion vid körning på ett godtyckligt träd? Glöm inte bort att motivera!

$$Legnis([rot \mid skog]) = Eller(Ett?(Längd(skog)),\ g(skog))$$

där $\begin{cases} g([\,]) = 0 \\ g([träd \mid skog]) = Eller(Legnis(träd),\ g(skog)) \end{cases}$

LÖSNING

Funktionen undersöker för ett godtyckligt träd om det innehåller någon nod som har *ett* enda barn, dvs som har exakt *en* utgående båge (nedåt).

MOTIVERING: Ett träd har nämnda egenskap omm roten är en nod som har exakt ett barn eller om någon av de träd som ligger i skogen under roten har egenskapen ifråga. Den givna funktionen är byggd på just detta faktum.
"Skogspecialisten" g testar skogen under roten.

ANM. Läs på sidan 76 i kursboken om nodernas släktskapsrelationer.

6. (6p) Konstruera en funktion $Förekomster(x, L)$ som returnerar antalet förekomster av x i en enkel lista L.

LÖSNING

$Förekomster(x, [\,]) = 0$
$Förekomster(x, [nod \mid L]) =$
$\quad Om(Lika(x, nod),\ Öka(Förekomster(x, L)),\ Förekomster(x, L))$

7. (7p) Tillverka en funktion som givet ett godtyckligt binärt träd och en godtycklig atom söker upp varje förekomst av atomen i trädet, och som returnerar en lista av samtliga vägar från trädets rot ner till de funna atomförekomsterna.
Om ex.vis det binära trädet är det som förekommer i uppgift 1 och om atomen är $+$, så skall förslagsvis $[[\cdot\ +]\ [\cdot\ \cdot\ +]]$ returneras.
Om atomen istället är x så är $[[\cdot\ +\ x]\ [\cdot\ \cdot\ +\ \cdot\ x]\ [\cdot\ \cdot\ +\ x]]$ en korrekt outputlista.

LÖSNING

Vägar([ ], x)=[ ]
Vägar([*rot*, *vä*, *hö*], x) =
  Om(*Lika*(*rot*, x),
    FogaIn({*rot*}, *Utöka*(*FogaSamman*(*Vägar*(x, *vä*), *Vägar*(x, *hö*)), *rot*)),
    Utöka(*FogaSamman*(*Vägar*(x, *vä*), *Vägar*(x, *hö*)), *rot*))

Utöka([ ], nod) = [ ]
Utöka([*väg* | *vägar*] , *nod*) =
  FogaIn(*UtökaVägen*(*väg*, *nod*), *Utöka*(*vägar*, *nod*))

där
UtökaVägen(*väg*, *nod*) = *FogaIn*(*nod*, *väg*)