

SI-möte #2, Programkonstruktion

Elias Castegren

elca7381@student.uu.se

9 november 2010

Begrepp

Vad betyder följande begrepp? Beskriv och ge exempel! (Du behöver inte ge formella definitioner)

- i)* Rekursion *ii)* Basfall *iii)* Matchning *iv)* Lokala deklARATIONER

Övningar

1.

Det n :te triangeltalet definieras som antalet kulor som behövs för att bygga en triangel med n rader. På liknande sätt definieras det n :te pyramidtalet som antalet kulor som behövs för att bygga en tetraeder (en pyramid med triangulär bas) med n nivåer.

- i)* Skriv en rekursiv funktion `triangle` som beräknar det n :te triangeltalet.
ii) Skriv en rekursiv funktion `pyramid` som beräknar det n :te pyramidtalet. Fundera på om `triangle` kan användas som hjälpfunktion.

(Anm. Det finns slutna formler för både triangel- och pyramidtalet som förstås skulle vara mer effektiva att använda. Uppgiften är tänkt som en övning i att skriva rekursiva funktioner)

2.

Skriv nedanstående rekursiva funktioner. Uppgifterna kan lösas utan hjälpfunktioner men det är inte fel att använda sig av sådana vid behov.

- i)* `sumMultiples(a, m) : int*int->int` som returnerar summan av alla naturliga tal mindre än m som är multipler av a .
ii) `isPalindrome(s) : string->bool` som avgör ifall s är ett palindrom (alltså en sträng som läses likadant framlänges och baklänges).
iii) `contains(s1, s2) : string*string->bool` som avgör ifall $s2$ förekommer någonstans i $s1$. Anropet `contains("rekursion", "kur")` ska alltså beräknas till `true` medan `contains("rekursion", "kub")` blir `false`.

3. Betrakta nedanstående funktion:

```
fun cousinMary("", _) = 0
  | cousinMary(syeeda, naima) =
    if String.sub(syeeda, 0)=naima then
      cousinMary(String.substring(syeeda, 1, size syeeda - 1), naima)
      + 1
    else
      cousinMary(String.substring(syeeda, 1, size syeeda - 1), naima)
```

Vad gör funktionen? Vad har den för typ? Skriv en egen version av funktionen (med vettiga argument- och funktionsnamn). Låt funktionen använda lokala deklARATIONER för att göra koden ännu mer lättläst.

4.

Ett sätt att undersöka ifall ett tal är delbart med tre är att summera siffrorna i talet. Om summan är delbar med tre så är talet självt delbart med tre. T.ex. är talet 252414 delbart med tre eftersom $2 + 5 + 2 + 4 + 1 + 4 = 18$ (som i sin tur är delbart med tre eftersom $1 + 8 = 9$).

Skriv ett program som använder den här metoden för att undersöka om ett tal n är delbart med tre eller inte. Du får *inte* använda jämförelser som $n \bmod 3 = 0$. Fundera på vilka funktioner som behövs och vad man kan använda som basfall.

5.

Betrakta nedanstående kod:

```
val a = 5;
fun f(x) =
  let
    val a = a + x
    val x = a + x
  in
    a + x
  end;
```

Vad beräknas anropet $f(a)$ till? Får man ett annat resultat om man gör samma anrop en gång till?

6.

Skriv ett en funktion `longestWord(s) : string -> string*int` som plockar ut det längsta ordet i strängen s och returnerar det i en tupel tillsammans med ordets längd. Du kan anta att s bara innehåller bokstäver och blanksteg.

Lycka till!