

SI-möte #5, Programkonstruktion och datastrukturer

Elias Castegren
elca7381@student.uu.se

10 december 2010

Begrepp

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

- i)* Iteration (svansrekursion) *ii)* Ackumulator *iii)* datatype-deklarationer
iv) Uppräkningsdatatyper *v)* Rekursiva datatyper *vi)* option-datatypen

Övningar

1.

- i)* Skriv en icke-svansrekursiv funktion som givet ett heltal n beräknar $\sum_{i=1}^n \frac{1}{i}$.

Visa steg för steg hur ett anrop med $n = 4$ beräknas. Hur ser det delvis beräknade uttrycket ut när basfallet ska beräknas (alltså vad är det som “ligger och väntar” på att beräknas)?

- ii)* Skriv nu en svansrekursiv variant av samma funktion. Gör samma steg för steg-beräkning igen. Hur ser det delvis beräknade uttrycket ut vid basfallet nu?

2.

Skriv en datatype `fruitNVeg` (med dokumentation enligt kodstandarden) som representerar frukter och grönsaker. Varje frukt och grönsak ska ha ett namn och vikten i kilo. Varje frukt ska också ha en av smakerna söt, sur och besk (använd en uppräkningsdatatyp!). “Skapa” sedan ett par frukter och grönsaker.

3.

Skriv följande funktioner till ovanstående datatyp:

- i)* `getWeight(l) : fruitNVeg list -> real`, som beräknar den totala vikten av alla frukter och grönsaker i listan `l`
- ii)* `getVegitables(l) : fruitNVeg list -> fruitNVeg list`, som returnerar alla grönsaker i listan `l`
- iii)* `filterFlavour(l, f) : fruitNVeg list * flavour -> fruitNVeg list` som returnerar en ny lista med alla frukter från `l` som har smaken `f`. Skriv både en "vanligt" rekursiv funktion och en svansrekursiv version

4.

Skriv en funktion `getLongest(s) : string -> string * int` som plockar ut det längsta ordet i strängen `s` och returnerar det i en tupel tillsammans med ordets längd. För enkelhets skull kan man anta att strängen bara innehåller bokstäver och mellanslag.

5.

Utgå från nedanstående datatypsdeklaration

```
datatype 'a tree = Tree of 'a * ('a tree) list;
```

och skriv en funktion `distance(v, t) : 'a * ('a tree) -> int option` som returnerar avståndet från roten i trädet `t` till noden med värdet `v` (alla värden i `t` kan antas vara unika). Avståndet ska vara taggat med `SOME`. Om det inte finns någon nod med värdet `v` skall `NONE` returneras.

6.

Betrakta nedanstående datatyp:

```
datatype 'a nested = Elem of 'a | List of 'a nested list;
```

Typen kan användas som ett alternativ för att hantera nästlade listor som vanligtvis inte tillåts i ML. Listan `[1, 2, [3, 4], 3]` skulle kunna representeras som:

```
List([Elem(1), Elem(2), List([Elem(3), Elem(4)]), Elem(3)])
```

Skriv en funktion `flatten(l) : 'a nested -> 'a list` som tar en nästlad lista och plattar till den, alltså "tar bort" alla listklamrar utom de yttersta. Ett anrop till `flatten` med den nästlade listan ovan ska ge resultatet `[1, 2, 3, 4, 3]`.

Lycka till!