

SI-möte #7, Programkonstruktion och Datastrukturer

Elias Castegren

elca7381@student.uu.se

13 December 2010

Begrepp

- i)* Vad är en anonym funktion i ML?
- ii)* Vad menas med en högre ordningens funktion i programmeringssammanhang?
- iii)* Vad menas med stuvade (*curried*) funktioner?
- iv)* Vad gör de inbyggda funktionerna `map`, `foldr` och `filter` i ML?
- v)* Vad är skillnaden på funktionerna `foldr` och `foldl`?

Övningar

1.

Betrakta nedanstående kod:

```
val x = 2;
fun foo x =
  let
    val f = fn x => x+1
    fun foo x = f(f x)
  in
    f(x) + foo x
  end;
foo(x);
```

Vad beräknas anropet i den sista raden till?

2.

Skriv följande funktioner i två versioner. En där du använder rekursion och en där du använder MLs inbyggda högre ordningens funktioner.

- i)* `addTags (l)` som returnerar stränglistan `l` med “<” och “>” runt varje element.
- ii)* `dropLess (l, k)` som returnerar heltalslistan `l` utan alla element mindre än `k`
- iii)* `getLongest (l)` som returnerar den längsta strängen i `l`.

3.

Skriv en funktion `bool` som tar två funktioner f och g och returnerar en tredje funktion $f \circ g$. Vad har `bool` för typ?

$f \circ g$ utläses “ f boll g ” och definieras som $(f \circ g)(x) = f(g(x))$. I ML kan man använda lilla `o` som en infix version av `bool`. “`hd o tl o tl`” beräknas till exempel till en funktion som givet en lista `l` beräknar `hd (tl (tl (l)))`.

4.

Vad gör följande stuvade högre ordningens funktion?

```
fun walrus [] lucy = lucy
  | walrus (eggman::eggmen) lucy = walrus eggmen (eggman lucy);
```

Vad har `walrus` för typ? Skriv ett lämpligt eftervillkor för funktionen.

5.

Skriv egna (rekursiva) versioner av de inbyggda funktionerna `map`, `foldr` och `filter` (med fullständiga funktionsspecifikationer). Skriv också `map` och `filter` med hjälp av `foldr`.

6.

En annan inbyggd högre ordningens funktion i ML är `List.tabulate (n, f)` som givet ett heltal `n` och en funktion `f` beräknar listan

```
[f(0), f(1), f(2), ..., f(n-1)]
```

Vad har `List.tabulate` för typ? Använd `List.tabulate` (eventuellt i kombination med andra funktioner) för att skriva följande funktioner:

- i)* `squares n` som beräknar en lista med kvadraten på alla naturliga tal som är mindre än `n`
- ii)* `range a b` som givet heltal `a` och `b` beräknar en lista med heltalen i intervallet $[a, b)$
- iii)* `primes n` som givet `n` beräknar en lista med alla primtal mindre än `n`

Lycka till!