

SI-möte #3, Programkonstruktion och datastrukturer

Elias Castegren

elca7381@student.uu.se

17 november 2010

Begrepp

- i)* Vad menas med en lista i ML
- ii)* Vad menas med en polymorf typ
- iii)* Förklara skillnaden mellan listoperatorerna `::` och `@`. Ge exempel!
- iv)* Varför är det oftast bättre att använda `::` än `@` när man ska bygga upp en lista?

Övningar

1.

Vilka värden har följande uttryck i ML? Ange dessutom värdenas typer.

- i)* `hd ["D", "M", "V"]`
- ii)* `3::2::[2, 4, 6]`
- iii)* `tl ((hd (tl [1, 2, 3, 4, 5, 6]))::[42])`
- iv)* `[]::[]`
- v)* `[("a" ^ "b" ^ "c", size "abc")] @ [("", length ([]@[]))]`

2.

Skriv följande ML-funktioner och ange deras typer. Använd inga inbyggda funktioner från List-biblioteket (förutom `::` och `@`).

- i)* `third(l)` som beräknar det tredje elementet i listan `l`
- ii)* `length(l)` som beräknar längden av listan `l`
- iii)* `addToList(l, n)` som adderar `n` till alla element i heltalslistan `l`
- iv)* `removeSign(l)` som returnerar flyttalslistan `l` men med alla element omgjorda till motsvarande absolutbelopp

3.

Skriv en funktion `sortedMerge(a, b)` som tar två heltalslistor sorterade i stigande ordning och slår ihop dem till en sorterad lista. Vad har funktionen för typ?

4.

Skriv en funktion `containsDuplicate(l)` som tar en lista och returnerar `true` om den innehåller en eller flera dubletter, annars `false`. Hur skulle funktionen kunna göras enklare om man bara tillåter sorterade listor?

5.

Funktionerna `explode` och `implode` kan användas för att omvandla värden mellan strängar och `char`-listor:

```
explode("Boom") → ["B", "o", "o", "m"],  
implode(["m", "o", "o", "B"]) → "mooB"
```

Använd dessa funktioner för att implementera en egen version av den inbyggda funktionen `String.substring` (den kan till exempel heta `mySubstring`). Använd inga inbyggda funktioner från `String`-biblioteket. Funktionen behöver inte kunna hantera ogiltiga värden.

6.

Skriv en funktion `zip(a, b)` som tar listorna $a = [a_1, a_2, \dots, a_n]$ och $b = [b_1, b_2, \dots, b_m]$ och returnerar $[(a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)]$. Om det blir några element över i någon lista (dvs om $m \neq n$) skall dessa ignoreras.

7.

Skriv en funktion `unzip(l)` som tar en lista $l = [(a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)]$ och returnerar $([a_1, a_2, \dots, a_n], [b_1, b_2, \dots, b_n])$

(Både `zip` och `unzip` finns inbyggda i `ML`-biblioteket `ListPair`)

Lycka till!