

recursion

Jon Sparring

October 18, 2018

1 Lærervejledningn

Emne Rekursion

Sværhedsgrad Middel

2 Introduktion

3 Opgave(r)

1. Skriv en funktion, `fac : n:int -> int`, som udregner fakultetsfunktionen $n! = \prod_{i=1}^n i$ vha. rekursion.
2. Skriv en funktion, `sum : n:int -> int`, som udregner summen $\sum_{i=1}^n i$ vha. rekursion. Lav en tabel som i Opgave 3i.0 og sammenlign denne implementation af sum med `while`-implementation og `simpleSum`.
3. Skriv en funktion, `sum : int list -> int`, som tager en liste af heltal og returnerer summen af alle tallene. Funktionen skal traversere listen vha. rekursion.
4. Den største fællesnævner mellem 2 heltal, t og n , er det største heltal c , som går op i både t og n med 0 til rest. Euclids algoritme¹ finder den største fællesnævner vha. rekursion:

$$\text{gcd}(t, 0) = t, \quad (1)$$

$$\text{gcd}(t, n) = \text{gcd}(n, t \% n), \quad (2)$$

hvor `%` er rest operatoreren (som i F#).

- (a) Implementer Euclids algoritme, som den rekursive funktion

`gcd : int -> int -> int`

- (b) lav en white- og black-box test af den implementerede algoritme,

- (c) Lav en håndkøring af algoritmen for `gcd 8 2` og `gcd 2 8`.

¹https://en.wikipedia.org/wiki/Greatest_common_divisor

5. Lav dine egen implementering af `List.fold` og `List.foldback` ved brug af rekursion.
6. Benyt `List.fold` og `List.foldback` og dine egne implementeringer fra Opgave 5 til at udregne summen af listen `[0 .. n]`, hvor `n` er et meget stort tal, og sammenlign tiden, som de fire programmer tager. Diskutér forskellene.