

# Programmering og Problemløsning

## Datalogisk Institut, Københavns Universitet

### Uge(r)seddel 5 – individuel opgave

Torben Mogensen

Deadline 6. oktober

I denne periode skal I arbejde individuelt. Formålet er at arbejde med lister, mængder (sets) og afbildninger (maps), og specielt med biblioteksfunktioner, der arbejder på disse typer.

Opgaverne i denne uge er delt i øve- og afleveringsopgaver.

#### Øveopgaverne er:

**Ø5.1.** HR: 5.1, 5.2.

**Ø5.2.** Definer en funktion `map` ækvivalent til `List.map` ved brug af `List.fold` eller `List.foldBack`.

**Ø5.3.** Givet funktionen `flip : ('a -> 'b - 'c) -> ('b -> 'a - 'c)` defineret ved

```
let flip f x y = f y x
```

lav definition af en funktion `fold` ækvivalent med `List.fold` ved at bruge funktionerne `List.rev` og `List.foldBack`.

**Ø5.4.** Brug `flip` fra opgave 5.3 til at definere en funktion `foldBack` ækvivalent til `List.foldBack` ved at bruge funktionerne `List.rev` og `List.fold`.

**Ø5.5.** Definer en funktion `ofList` ækvivalent til `Set.ofList` ved at bruge et udvalg af funktionerne `Set.add`, `List.fold` og `List.foldBack`.

**Ø5.6.** En snedig programmør definerer en sorteringsfunktion med definitionen

```
ssort xs = Set.toList (Set.ofList xs)
```

For eksempel giver `ssort [4; 3; 7; 2]` resultatet `[2; 3; 4; 7]`.

Diskuter, om programmøren faktisk er så snedig, som han tror.

**Ø5.7.** Undersøg, hvad der sker, hvis man bruger `Map.add` til at tilføje et nøgle/værdipar med en nøgle, der allerede findes i afbildningen. Diskuter fordele og ulemper ved denne opførsel.

#### Afleveringsopgaven er:

**A5.1.** Brug funktionerne fra Tabel 5.1 i HR (side 94) til at definere en funktion `concat : 'a list list -> 'a list`, der sammensætter en liste af lister til en enkelt liste.

F.eks. skal `concat [[2]; [6; 4]; [1]]` give resultatet `[2; 6; 4; 1]`.

**A5.2.** Brug funktionerne fra Tabel 5.1 i HR (side 94) til at definere en funktion `gennemsnit : float list -> float option`, der finder gennemsnittet af en liste af kommatall, såfremt dette er veldefineret, og `None`, hvis ikke. Forsøg at lave din løsning, så den kun laver et gennemløb af listen.

**A5.3.** Definer en funktion `toList` ækvivalent til `Set.toList` ved at bruge `Set.fold` eller `Set.foldBack`.

Afleveringsopgaven skal afleveres som både  $\LaTeX$ , den genererede PDF, samt en `fsx` fil med løsningen for hver delopgave, navngivet efter opgaven (f.eks. `A5-1.fsx`), som kan oversættes med `fsharp`, og hvis resultat kan køres med `mono`. Det hele samles i en zip-fil med sædvanlig navnekonvention (se tidligere ugesedler).

God fornøjelse

## Ugens nød 2

Vi vil i udvalgte uger stille særligt udfordrende og sjove opgaver, som interesserede kan løse. Det er helt frivilligt at lave disse opgaver, som vi kalder “Ugens nød”, men der vil blive givet en mindre præmie til den bedste løsning, der afleveres i Absalon.

Denne uges opgave omhandler afbildninger (maps).

Der er stor lighed mellem funktioner og *maps* som defineret i afsnit 5.3 i HR. Vi definerer derfor typen

```
type myMap<'a,'b> = ('a -> 'b) * Set<'a>
```

hvor funktionen definerer afbildningens funktion, og mængden bruges til at beskrive domænet for afbildningen.

**Nød 2.1.** Definer for `myMap<'a,'b>` alle de funktioner fra Table 5.4 i HR, der er defineret for `Map<'a,'b>`.

Mere præcist skal alle forekomster af `Map<'a,'b>` i typesignaturerne erstattes af `myMap<'a,'b>`, og funktionerne skal gøre det, der er beskrevet i tabellen.

Der skal uploades både en  $\text{\LaTeX}$ -fil, der beskriver fremgangsmåden, samt en fsx fil, der indeholder definitionerne. Navngivningen af filerne er ikke vigtig.