

TENTAMEN I DVA 229 FUNKTIONELL PROGRAMMERING MED F#

Torsdagen den 24 mars 2022, kl 14:30 – 18:30

Kurslitteratur är inte tillåten, och inte heller andra hjälpmedel som på något sätt kan ersätta kurslitteraturen (t.ex. egna anteckningar, andra böcker i ämnet, kopior av OH-bilder, datorer eller räknare med dito lagrad information). Endast generella hjälpmedel är tillåtna, som räknare utan lagrad information av betydelse för kursen, ordbok, allmän formelsamling och liknande. För godkänt krävs 15 poäng, max är 30 poäng. Resultatet offentliggörs senast torsdagen den 14 april 2022.

Vänligen observera följande:

- Motivera alltid dina svar. Bristande motivering kan ge poängavdrag. Omvänt kan även ett felaktigt svar ge poäng, om det framgår av motiveringen att tankegången ändå är riktig.
- Skriv tydligt!
- Varje blad skall vara försedd med uppgiftsnummer och bladnummer.
- Endast en uppgift på ett och samma blad.
- Skriv enbart på ena sidan av ett blad.
- Uppgifterna är inte nödvändigtvis sorterade i svårighetsgrad. Om du kör fast kan det löna sig att gå vidare till nästa uppgift.
- Lösningförslag kommer att finnas på kursens hemsida efter att tentan är slut.

Frågor: Björn Lisper på 0739-607199.

UPPGIFT 1 (6 POÄNG)

Deklarera en funktion som tar en lista av flyttal (`float`) och returnerar det största icke-negativa talen i listan! Om inga positiva tal finns i listan ska funktionen returnera `0.0`.

a) Deklarera en version som använder rekursion! (4p)

b) Gör en alternativ deklaration av funktionen som använder sig av någon eller några inbyggda högre ordningens listfunktioner i F# på ett vettigt sätt. (2p)

UPPGIFT 2 (4 POÄNG)

`Array.tryFind : ('a -> bool) -> 'a [] -> 'a option` är en funktion i Array-modulen.

`Array.tryFind p a` returnerar `Some x` för det första elementet `x` i arrayen `a` där `p x` blir sant, eller `None` om `a` inte innehåller något sådant element. T.ex. ska gälla att:

```
Array.tryFind (fun x -> x < 0) [|0;-2;-1|] = Some -2
Array.tryFind (fun x -> x < 0) [|0;2;1|] = None
```

Ge en deklaration av `Array.tryFind` som fungerar på detta sätt! Funktionen ska använda rekursion och får inte begagna inbyggda F#-funktioner.

UPPGIFT 3 (2 POÄNG)

Betrakta följande kod:

```
let x = lazy (printf "Hello world"; sqrt 9.0)
let y = x.Force()
let z = x.Force()
```

Vad händer när man kör koden, och varför?

UPPGIFT 4 (4 POÄNG)

Deklarera en funktion som tar en lista av heltal (`int`) som argument och returnerar produkten av elementen. Är listan tom ska 1 returneras. Beräkningen ska ske genom att produkten successivt ackumuleras i en muterbar referenscell. Din lösning ska använda rekursion.

UPPGIFT 5 (2 POÄNG)

Följande funktion summerar ASCII-koderna för tecknen i en sträng:

```
let rec asciiSum s = if s = ""
                     then 0
                     else int(s.[0]) + asciiSum (s.[1..])
```

Denna kod gör rätt sak men har ett problem. Vilket? Förklara varför problemet uppstår!

UPPGIFT 6 (2 POÄNG)

Följande kod är ett försök att deklarerar två ömsesidigt rekursiva funktioner `f` och `g`:

```
let rec f x = g (x-1)
let rec g x = if x = 0 then 1 else f x
```

Detta kommer dock ej att fungera. Förklara varför!

UPPGIFT 7 (6 POÄNG)

En variant av binära träd ser ut som följer:

- En nod kan ha noll, en eller två söner. (De med noll söner är löv.)
- Varje nod (även löv) innehåller data av någon typ `'a` som är samma för alla noder i trädet, men för övrigt kan vara vilken typ som helst.

a) Deklarera en datatyp för denna typ av binära träd! (2p)

b) Deklarera en funktion `treeMap`, som tar en funktion `f` och ett träd `t` av ovanstående typ som argument, och returnerar ett träd av samma form där varje data `x` har ersatts av `f x`. (`treeMap` ska alltså implementera en `map`-funktion för dessa träd.) (4p)

UPPGIFT 8 (4 POÄNG)

Visa att den listalgebraiska lagen

```
map f >> tail = tail >> map f
```

gäller. Du behöver inte göra ett helt formellt induktionsbevis, utan det räcker med ett lite mer informellt resonemang i den stil som vi gjort på föreläsning. (4p)

Lycka till! Björn