

TENTAMEN I DVA 229 FUNKTIONELL PROGRAMMERING MED F#

Fredagen den 3 juni 2016, kl 14:10 – 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 23 juni 2016.

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 021-151709.

UPPGIFT 1 (5 POÄNG)

Inom vissa numeriska tillämpningar är det vanligt att man har arrayer där de allra flesta elementen är noll. Det kan då löna sig att ha en speciell representation, där man bara lagrar de element som är skilda från noll. Man måste då också lagra deras index. Ett sätt att göra detta är att ha en lista av par (index,värde) som är sorterad m.a.p. index: t.ex. arrayen

```
[ | 1.0; 0.0; 0.0; 0.0; 3.5; 0.0 | ]
```

lagras som

```
[ (0, 1.0) ; (4, 3.5) ]
```

a) deklarera en funktion som summerar elementen i en array av `float` representerad på detta sätt! Summeringen ska göras med direkt rekursion. (3p)

b) samma fråga, men implementeringen ska istället göras med någon eller några av de inbyggda högre ordningens funktionerna i F#. (2p)

UPPGIFT 2 (2 POÄNG)

Betrakta följande F#-deklarationer:

```
let f x = x + 17
let g x = f (x : float)
```

Kommer `f` respektive `g` att få någon typ och i så fall vilken? Förklara vad som händer!

UPPGIFT 3 (5 POÄNG)

Deklarera en funktion som tar en sträng av numeriska tecken ('0' - '9') och returnerar motsvarande heltal! Exempelvis ska strängen "123" översättas till heltalet 123, av typ `int`. Om strängen är tom ska heltalet 0 returneras. Om det finns något icke-numeriskt tecken i strängen ska ett felavbrott ske.

UPPGIFT 4 (3 POÄNG)

Betrakta följande F#-deklarationer:

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

- a) Vad blir resultatet av anropet `g (f -1) 1` om vi använder ivrig evaluering (call-by-value)? (1p)
- b) Vad blir resultatet av anropet `g (f -1) 1` om vi använder lat evaluering (call-by-need)? (1p)
- c) Använder F# lat eller ivrig evaluering? (1p)

UPPGIFT 5 (4 POÄNG)

Deklarera en funktion `maxarray` som tar en array av flyttal (`float`) som argument och returnerar det största av dess element. Du får anta att arrayen inte är tom.

Din funktion ska utifrån sett inte ha några sidoeffekter, men ska internt använda en lokalt deklarerad muterbar referenscell där det största elementet successivt ackumuleras. Funktionen ska använda rekursion – imperativa F#-loopar är inte tillåtna. Du får heller inte använda någon av de inbyggda funktionerna i F# som skulle kunna användas för att göra en direkt lösning.

UPPGIFT 6 (6 POÄNG)

Inom datorgrafiken representerar man ibland 3D-objekt med mängder av enkla geometriska 3D-objekt, som sfärer, som tillsammans approximerar det objekt som man vill representera. Det är vanligt att man organiserar dessa mängder av objekt som träd, där objekten finns i löven av träden.

a) Deklarera en datatyp för binära träd av sfärer enligt ovan, där varje sfär karakteriseras av dess mittpunkt som är en 3D-koordinat (tre flyttal) och dess radie (flyttal). (2p)

b) Deklarera en funktion som tar ett träd av typen som deklarerats i a), och som utifrån representationen som en mängd sfärer beräknar volymen av det objekt som representeras. Du får göra det förenklande antagandet att inga sfärer överlappar i rummet. (4p)

Ledning: volymen av en sfär med radien r är $4\pi r^3/3$.

UPPGIFT 7 (5 POÄNG)

Funktionen `pair` definieras av

```
let rec pair l =
  match l with
  | [] -> []
  | x::xs -> (x,x) :: pair xs
```

a) Vad beräknar funktionen `pair`? (1p)

b) Härled en typ för `pair`! För full poäng ska den härledda typen vara den mest generella. Ordentlig motivering krävs. (4p)

Ledning: man kan betrakta `cons (::)` och parbildning `(,)` som två funktioner med följande typer:

```
(::) : 'a -> 'a list -> 'a list
(,) : 'b -> 'c -> ('b * 'c)
```

Lycka till! Björn