Tentamen 1 (6 högskolepoäng) i Programkonstruktion och datastrukturer (1DL201)

Lars-Henrik Eriksson och Pierre Flener

Fredag 17 december 2010, kl 14:00 – 17:00, i Gimogatan 4, sal 1

Hjälpmedel: Inga. ***Inte*** heller elektronisk utrustning.

Hjälp: En av huvudlärarna kommer att besöka skrivsalen kl. 15:00 – 16:00 (senast).

Anvisningar: Markera i tabellen nedan ***inte mer än ett*** svar per fråga genom att kryssa över bokstaven för det svarsalternativ som du väljer.

Riv av denna sida och lämna bara in den. Det är inte meningen att du skall lämna kommentarer till dina svar (annat än om du anser att en fråga är fel – se nästa stycke).

Om du tycker att någon fråga är oklar eller felaktig, markera den med en $\star$ på ***den här*** sidan, och förklara ***på detta blad*** (på baksidan om det behövs) vad du menar att problemet är och vilka antaganden du gjort för att kunna svara på frågan.

	Fråga	Svar					Fråga	Svar				
betyg **3 U**	1	A	B	C	D	E	2	A	B	C	D	E
	3	A	B	C	D	E	4	A	B	C	D	E
	5	A	B	C	D	E	6	A	B	C	D	E
	7	A	B	C	D	E	8	A	B	C	D	E
	9	A	B	C	D	E	10	A	B	C	D	E
betyg **4**	11	A	B	C	D	E	12	A	B	C	D	E
	13	A	B	C	D	E	14	A	B	C	D	E
	15	A	B	C	D	E						
betyg **5**							16	A	B	C	D	E
	17	A	B	C	D	E	18	A	B	C	D	E
	19	A	B	C	D	E	20	A	B	C	D	E

Identitet: Din tentakod (eller namn och personnummer om du saknar kod):

. . . . . .

Frågeunderlag

En del av frågorna kommer att behandla ett problem som liknar det från den första lektionen. Man har en tidtabell för en busslinje, tågförbindelse eller liknande och vill hitta den förbindelse som avgår från en viss plats efter ett visst klockslag. Ta följande utdrag ur en tidtabell för tågen Uppsala–Stockholm som exempel.

Uppsala	Knivsta	Märsta	Stockholm
08:09	08:20	08:28	08:49
12:09	12:20	12:28	12:49
16:09	16:20	16:28	16:49
20:09	20:20	20:28	20:49

Tidtabellen anger alltså bl.a. att det går ett tåg från Uppsala kl. 08:09 som passerar Knivsta 08:20, Märsta 08:28 och kommer till Stockholm 08:49.

Tidtabellen representeras som en lista av förbindelser. Varje förbindelse representeras som en lista av tider för varje station. Tiderna representeras som heltal med timmar som hundratal. 1220 representerar alltså klockan 12:20. En fördel med denna representation är att man kan jämföra tider genom att göra en heltalsjämförelse. Dessutom finns en separat "rubriklista" som innehåller namnen på de olika platserna i förbindelserna. Tidtabellen i exemplet representeras alltså av rubriklistan

```
["Uppsala","Knivsta","Märsta","Stockholm"]
```

och förbindelselistan

```
[[0809,0820,0828,0849],[1209,1220,1228,1249],
 [1609,1620,1628,1649],[2009,2020,2028,2049]]
```

Man skall nu skriva ett program `lookUpTimetable` som tar som argument

- En rubriklista
- En förbindelselista
- Ett platsnamn
- En tidpunkt

och som tar reda på tiden för den första förbindelsen samma dag från den angivna platsen räknat från och med den angivna tidpunkten. Svaret skall ges som `SOME x` där x är tiden. Om det inte finns någon sådan förbindelse (den sista förbindelsen har redan passerats) skall svaret vara `NONE`.

Om man till exempel frågar efter första förbindelse från Knivsta från och med klockan 11:30 med ovanstående exempeltidtabell så skall svaret bli `SOME 1220`. Frågar man efter första förbindelse från Uppsala från och med klockan 20:15 så skall svaret bli `NONE`.

För att förenkla programmet så förutsätter vi att alla förbindelser passerar alla platser och att förbindelserna kommer i samma tidsordning vid varje plats (inga "omkörningar").

Nedan finns funktionsspecifikation och programkod för `lookUpTimetable`. Båda är ofullständiga – det finns luckor i dem, markerade `?1?`, `?2?` etc. Din uppgift blir att tala om vad som skall finnas i luckorna.

`lookUpTimetable(headers,connections,place,time)`
TYPE: `?1?`
PRE: `?2?`
POST: `SOME x` där `x` är tid för första förbindelsen vid `place` från och med tiden `time` och där `headers` respektive `connections` är en rubriklista respektive en förbindelselista enligt ovan.
EXAMPLE: `lookUpTimetable(["Uppsala","Knivsta",Märsta",Stockholm"], [[0809,0820,0828,0849],[1209,1220,1228,1249], [1609,1620,1628,1649],[2009,2020,2028,2049]],"Knivsta",1130) = SOME 1220`
VARIANT: `?6?`

```
fun lookUpTimetable(_,[],_,_) = NONE
  | lookUpTimetable(headers,connections,place,time) =
      let
        val thisTime = ?3?
      in
        if thisTime ?4? time then
          SOME thisTime
        else
          ?5?
     end;
```

`lookUpTimetable` använder en hjälpfunktion med följande funktionsspecifikation:

`parallelMember(L1,x,L2)`
TYPE: `string list*string*int list->int`
PRE: `x` finns i listan `L1`, `length L2` > positionen för `x` i listan `L1`
POST: Det element som finns på samma plats i `L2` som första förekomsten av `x` i `L1`.
EXAMPLE: `parallelMember(["A","B","C"],"B",[10,20,30]) = 20`

Frågor för betyg 3

Om du ger rätt svar på 7 av de 10 frågorna i detta avsnitt så blir du godkänd med minst betyg **3**, annars blir du underkänd (**U**). Du kan inte kompensera ett dåligt resultat i detta avsnitt med poäng från frågorna för betyg **4** eller **5**.

1. Vilken typ skall stå vid `?1?` ovan?
   (A) `int` (B) `int option` (C) `string list*int list*string*int->int`
   (D) `string list*int list*string*int->int option`
   (E) `string list*int list list*string*int->int option`

2. Vilket av följande skall ***inte*** vara en del av förvillkoret vid `?2?` ovan?
   (A) I varje element av `connections` skall värdena ligga i stigande ordning.
   (B) Varje element i `connections` skall vara en lika lång lista som listan `headers`
   (C) `place` måste finnas i `headers`
   (D) `time` måste finnas i ett element i `connections`
   (E) ***Alla*** villkoren (A)-(D) skall vara med i förvillkoret!

3. Vilken kod skall stå vid `?3?` ovan?
   (A) `parallelMember(headers,place,connections)`
   (B) `parallelMember(hd headers,place,connections)`
   (C) `parallelMember(headers,time,connections)`
   (D) `parallelMember(headers,time,hd connections)`
   (E) Något annat.
   Det som skall stå är: `parallelMember(headers,place,hd connections)`

4. Vilken kod skall stå vid `?4?` ovan?
   (A) `<=` (B) `>=` (C) `=` (D) `<>` (E) Något annat.

5. Vilken kod skall stå vid `?5?` ovan?
   (A) `lookUpTimetable(tl headers,connections,place,time)`
   (B) `lookUpTimetable(headers,tl connections,place,time)`
   (C) `lookUpTimetable(tl headers,tl connections,place,time)`
   (D) `hd connections::lookUpTimetable(headers,tl connections,place,time)`
   (E) Något annat.

6. Vilken variant skall stå vid `?6?` ovan?
   (A) `length headers` (B) `time` (C) `length connections` (D) `connections`
   (E) Något annat.

7. I något sammanhang arbetar man med kassakvitton som innehåller – i denna ordning:

   - Datum och klockslag
   - Ett antal rader med varor – för varje vara namn, styckepris och antal sålda exemplar
   - Den totala kostnaden för alla sålda varor

   Man vill definiera en ny datatyp som skall representera sådana kassakvitton:

   ```
   datatype kvitto = Kvitto of ...
   ```

   Vilket av följande alternativ är ***mest rimlig*** som fortsättning av deklarationen?
   (A) `string*int*string list*int list*int list*int`
   (B) `string*int*(string*string*string)list*int`
   (C) `(string*int*string*int*int*int)list`
   (D) `string*int*(string*int*int)list*int`
   (E) `string*int*(string*int list)list*int`

8. Vad är värdet av uttrycket `foo(10,3)` om `foo` är definierad så här?

   ```
   fun foo(x,y) = if x<y then x else foo(x-y,y);
   ```

   (A) 0 (B) 1 (C) 2 (D) 3 (E) 4

9. Vilken är en korrekt asymptotisk begränsning av polynomet $3 \cdot n^2 + 5 \cdot n + 4$?
   (A) $O(n)$ (B) $\Theta(n)$ (C) $O(n^3)$ (D) $\Theta(n^3)$ (E) $\Omega(n^3)$
   Enligt definitionen av O.

10. Vilken är en strikt ("tight") asymptotisk begränsning av körtiden för funktionen `g` nedan? Antag att `g` endast är definierad för ickenegativa heltal. Antag att `fakt`(n) ***alltid*** beräknar fakulteten $n!$ av n, i tid $\Theta(n)$.

    ```
    fun g(0) = 0
      | g(n) = if n > 1000 then g(n-1) + fakt(7) else g(n-1) - fakt(7)
    ```

    (A) $\Theta(n)$ (B) $\Theta(n \cdot \lg n)$ (C) $\Theta(n^2)$ (D) $\Theta(n^2 \cdot \lg n)$ (E) $\Theta(2^n)$
    Enligt teorem 1, med $a = 1$ och $b = \Theta(1)$.

Frågor för betyg 4

Om du fått minst betyg **3** genom dina svar på de föregående frågorna och dessutom svarar rätt på minst 3 av de 5 frågorna i detta avsnitt så blir du godkänd med minst betyget **4**. Du kan inte kompensera ett dåligt resultat i detta avsnitt med poäng från frågorna för betyg **3** eller **5**.

11. Vilket av följande påståenden om typer är ***inte*** sant?
    (A) Typer är mängder av data med gemensamma egenskaper.
    (B) Typen hos en funktion avgör vilka typer funktionens argument och värde har.
    (C) Typkontroll kan hitta fel i program.
    (D) I en tupel kan olika element ha olika typ.
    (E) Varje värde hör till en bestämd typ.
    `[]` och `NONE` är exempel på värden som hör till flera typer samtidigt.

12. Här är några olika testfall för `lookUpTimetable`:

    (a) `lookUpTimetable(["A","B"],[[1000,1030]],"A",1000) = SOME 1000`

    (b) `lookUpTimetable(["A","B"],[[1000,1030]],"A",1030) = NONE`

    (c) `lookUpTimetable([],[],"A",1030) = NONE`

    (d)
    ```
    lookUpTimetable(["A","B","C"],[[1000,1030,1100],[1100,1130,1200]],
                    "B",1045)  = SOME 1130
    ```

    Vilka av dessa testfall krävs för att få full kodtäckning i `lookUpTimetable` utan att ta med några onödiga testfall?
    (A) a (B) b (C) a och b (D) a, b och c (E) a, b, c och d

13. Här är fyra förslag till eftervillkor för funktionen `foo` från fråga 8. Alla är i någon mening korrekta (om vi antar att förvillkoren är rimliga), men vilka är ***acceptabla*** eftervillkor i en funktionsspecifikation enligt de principer vi gått igenom i kursen?

    (a) Man dividerar `x` med `y`. Svaret är resten.

    (b) `x` mod `y`

    (c) `x` om `x` är mindre än `y`. Annars rekursion med argumenten `x-y` och `y`.

    (d) Man subtraherar upprepade gånger `y` från `x` tills `x<y`. Då är svaret `x`.

    (A) a och b (B) a och c (C) b och d (D) a, b och d (E) Alla fyra.
    a och b svarar på frågan "vad?" medan c och d svarar på frågan "hur?"

14. Vilket är det totala antalet multiplikationer och divisioner som utförs vid beräkningen av `p`(n) med hjälp av definitionen av `p` nedan? Ett förvillkor för `p` är att argumentet är en positiv potens av 2.

```
fun p(2) = 1
  | p(n) = let val d = 4 * p(n div 2) in d * d end
```

(A) $\lg n$ (B) $\underline{3 \cdot \lg\left(\frac{n}{2}\right)}$ (C) $\frac{1}{6} \cdot n^2 - \frac{2}{3}$ (D) $\frac{1}{4} \cdot n^2 - 1$ (E) $n^2 \cdot \lg n$
Det är den slutna formen (visas med induktion) av följande "divide-and-conquer" rekursionsformel för antalet multiplikationer och divisioner, $MD(n)$, som görs när `p`(n) beräknas:

$$MD(n) = \begin{cases} 0 & \text{om } n = 2 \\ 1 \cdot MD(n/2) + 3 & \text{om } n = 2^k \text{ där } k > 1 \end{cases}$$

15. Vilken är en strikt ("tight") asymptotisk begränsning av körtiden för funktionen `t` nedan? Använd "Master Theorem" (MT)! Låt n vara längden av listargumentet. Observera att `t`(L) beräknar en lista av ***samma*** längd som listan L. Antag att `d`(L) beräknar två dellistor till listan L av ungefär ***halva*** längden ($|L|/2$) av listan L och att den beräkningen ***alltid*** görs i tid $\Theta(|L|/2)$. Antag att `u`(L) beräknar en lista med samma längd ($|L|$) som listan L och att den beräkningen ***alltid*** görs i tid $\Theta(|L| \cdot \lg |L|)$.

```
fun t([ ]) = [ ]
  | t([x]) = [x]
  | t(xxs) = let val (A,B) = d(xxs) in u(t(A) @ t(B)) end
```

(A) $\Theta(n)$ (B) $\Theta(n \cdot \lg n)$ (C) $\Theta(n^2)$ (D) $\Theta(n^2 \cdot \lg n)$ (E) $\underline{\text{MT är inte tillämpbar}}$
Eftersom $f(n) = \Theta(n/2 + n \cdot \lg n)$ inte dominerar $n^{\log_b a} = n$, då $a = 2 = b$.

Frågor för betyg 5

Om du fått minst betyg **4** genom dina svar på de föregående frågorna och dessutom svarar rätt på minst 3 av de 5 frågorna i detta avsnitt så blir du godkänd med betyg **5**. Du kan inte kompensera ett dåligt resultat i detta avsnitt med poäng från frågorna för betyg **3** eller **4**.

16. Man kan koda funktionen `parallelMember` så här:

```
fun parallelMember(l::ll,x,m::mm) =
  if l = x then
    m
  else
    parallelMember(ll,x,mm);
```

När ett ML-system behandlar definitionen av `parallelMember` så får den ***inte*** samma typ som i funktionsspecifikationen som gavs tidigare i tentan. Vilken typ får den?

(A) `'a list*'a*'b list->'b`
(B) `''a list*''a*''a list->''a`
(C) `''a list*''a*string list->string`
(D) `''a list*''a*'b list->'b`
(E) `''a list*''a*''b list->''b`

17. Man matar in följande deklarationer till ett ML-system:

```
val x = 1;
fun bar y =
  let
    val x = x+1
  in
    x+y
  end;
val x = 3;
```

Därefter ger man – i tur och ordning – ML-systemet uttrycken `bar x` och `x` att beräkna. Vilka värden beräknas?

(A) 5 och 3 (B) 5 och 4 (C) 7 och 3 (D) 7 och 4 (E) 8 och 3

18. Datastrukturdefinitionen i fråga 7 behöver en datastrukturinvariant. Vilken del av datastrukturen ***behöver inte*** (rimligen) nämnas i datastrukturinvarianten? (Dvs förekomster av denna del kan ha vilket värde som helst – även oberoende av resten av datastrukturen.)
(A) Datumet (B) Tidpunkten (C) Namn på varor (D) Summan (E) Antal varor
Fälten för datum och tidpunkt måste ha ett innehåll som verkligen beskriver ett datum eller en tidpunkt. Summan måste stämma med styckepris och antal hos alla varor. Antalet av en vara kan inte vara negativt (eller noll). Däremot finns inget som begränsar vad en vara kan heta.

19. Vilket är det minst restriktiva förvillkor (det som tillåter flest argument) som funktionen `foo` i fråga 8 kan ha om den alls skall beräkna något värde? (Alltså inte nödvändigtvis ett värde som stämmer med eftervillkoren i fråga 13).
(A) $y \neq 0$ (B) $y > 0$ (C) $x \geq 0$ ***och*** $y > 0$ (D) $x < y$ ***eller*** $y > 0$ (E) Något annat.

20. Författaren till *Da Vinci-koden*, Dan Brown, skriver i kapitel 4 och 5 av sin bok *Gåtornas palats* att den påhittade datorn TRANSLTR rutinmässigt hittar 64 bitars lösenord för krypterade texter på ungefär 10 minuter. TRANSLTR har tre miljoner processorer som arbetar parallellt. Den hittade ett visst lösenord med en miljon bitar på tre timmar. Vad är den ***genomsnittliga*** tiden som krävs för att hitta 70-bitars lösenord?
(A) $\approx$ 11 min. (B) $\approx$ 12 min. (C) $\approx$ 13 min. (D) $\approx$ 11 timmar (E) Något annat
Att gissa n bitar tar tid $\Theta(2^n)$ i värsta fall, alltså kommer ytterligare 6 bitar att öka den genomsnittliga körtiden med en faktor $2^6 = 64$.