Løsningsforslag til DM22 eksamen juni 2004

Opgave 1 (25 %)

1.a Typesignaturerne fremgår af den fulde definition:

```
mystery :: [Int] -> Int
mystery xs  = snd (ms (0,0) xs)

ms :: (Int,Int) -> [Int] -> (a,a)
ms (c,s) (x:xs)
        = ms (cc,ss) xs
        where cc = max 0 (c + x)
              ss = max s cc
ms (c,s) [] = (c,s)
```

NB: Typen må være `Int`, idet der sammenlignes med 0.

1.b Kaldet `mystery [1,2,-3,4,5]` returnerer 9, og `mystery []` værdien 0.

1.c Der er tale om en halerekursion, dvs. iteration, og tidskompleksiteten er derfor $\Theta(n)$, da øvrigt arbejde er $\Theta(1)$.

1.d Det fremgår at `mystery` beregner det maksimale "segment", dvs. hvis listen består af $[a_0, \cdots, a_{n-1}]$, da beregnes $M_n = \max_{0\leq j\leq k\leq n}(S_{jk})$ hvor $S_{jk} = \sum_j^{k-1} a_i$ er et "segment", idet det bemærkes at $S_{jj} = 0$.

Induktionsbeviset må baseres på egenskaber ved `ms`, dvs. afbildningen `(c,s)` $\to$ `(cc,ss)`, og hypotesen må indeholde invarianter for c og s, udtrykt for given værdi af $n = h$.

Basistilfældet: Kaldet `ms (0,0) []` returnerer `(0,0)`, svarende til at $M_0 = S_{00} = 0$.

Hypotese $n = h$: (Invarianter)

$$\begin{aligned} s &= M_h \\ c &= \max_{0\leq j\leq h} S_{jh} \geq 0 \end{aligned}$$

hvilket holder i basistilfældet hvor $h = 0$ og $(c, s) = (0, 0)$, og udtrykket for s svarer til det forventede resultat.

Induktionsskridtet $n = h + 1$:
Her sættes cc til $\max(0, c + a_h)$. Vi betragter nu:

$$\begin{aligned} \max_{0\leq j\leq h+1} S_{jh+1} &= \max(\max_{0\leq j\leq h} S_{jh+1}, \max_{0\leq j=h+1} S_{jh+1}) \\ &= \max(\max_{0\leq j\leq h}(S_{jh} + a_h), S_{h+1h+1}) \\ &= \max((\max_{0\leq j\leq h} S_{jh}) + a_h, S_{h+1h+1}) \\ &= \max(c + a_h, 0) \;=\; cc \geq 0 \end{aligned}$$

dvs. opdateringen af c giver det korrekte resultat. Tilsvarende fås for s:

$$\begin{aligned} M_{h+1} &= \max_{0\leq j\leq k\leq h+1}(S_{jk}) \\ &= \max(\max_{0\leq j\leq k\leq h}(S_{jk}), \max_{0\leq j\leq k=h+1}(S_{jh})) \\ &= \max(M_h, \max_{0\leq j\leq h+1}(S_{jh+1})) \\ &= \max(s, cc) \;=\; ss. \end{aligned}$$

Opgave 2 (25 %)

Her er en løsning med et forsimplet Europakort:

```
naboer(tyskland,[luxenborg,frankrig,belgien,holland,danmark]).
naboer(belgien,[frankrig,luxenborg,tyskland,holland]).
naboer(luxenborg,[belgien,frankrig,tyskland]).
naboer(frankrig,[luxenborg,tyskland,belgien]).
naboer(holland,[belgien,tyskland]).
naboer(danmark,[tyskland]).

farv([]).
farv([Land/Farve|Rest]):-
        farv(Rest),write(Rest),nl,
        member(Farve,[rod,gul,gron,blaa]),
        \+ ','(member(Land1/Farve,Rest),nabo(Land,Land1)).

nabo(Land,Land1):-
        naboer(Land,Naboliste),
        member(Land1,Naboliste).

land(L):- naboer(L,_).

farvning(FarveListe):-
        findall(L/_,land(L),FarveListe),farv(FarveListe),!
```

hvor kaldet `farvning(X)`.giver flg. output:

```
 | ?- farvning(X).
[]
[danmark/rod]
[holland/rod,danmark/rod]
[frankrig/rod,holland/rod,danmark/rod]
[luxenborg/gul,frankrig/rod,holland/rod,danmark/rod]
[belgien/gron,luxenborg/gul,frankrig/rod,holland/rod,danmark/rod]

X = [tyskland/blaa,belgien/gron,luxenborg/gul,frankrig/rod,holland/rod,danmark/rod]
```

Opgave 3 (25 %)

3.a Det ses let at

$$p = a^2 + b^2 \qquad q = b(a+b) + ab = b^2 + 2ab$$

3.b At multiplicere med x svarer her til at multiplicere resultatet i form af en matrix U (repræsenteret ved et talpar (u, v)) med matricen A som repræsentation af x, hvor

$$A = \left\{ \begin{array}{cc} a+b & b \\ b & a \end{array} \right\}, \quad U = \left\{ \begin{array}{cc} u+v & v \\ v & u \end{array} \right\}$$

dvs. produktet repræsenteres ved tuplet (x, y) hvor

$$x = bv + au, \qquad y := (a+b)v + bu$$

3.c For lige n skal der kun kvadreres, for ulige n både kvadreres og det hidtidige resultat multipliceres med A, dvs.:

```
fib :: Int -> Int
fib n = snd (fh (0,1) n)

fh :: (Int,Int) -> Int -> (Int,Int)
fh (a,b) n
       | n < 0          =  error "negativt argument"
       | n==1           =  (a,b)
       | n`mod`2 == 0   =  fh (a*a+b*b,(b+2*a)*b) (n `div` 2)
       | otherwise      =  (b*v+a*u,(a+b)*v+b*u)
                           where (u,v)=fh (a*a+b*b,(b+2*a)*b) ((n-1) `div` 2)
```

Med `listfib`:

```
listfib :: Int -> [Int] -> [Int]
listfib n list
        | n==1      = list
        | otherwise = listfib (n-1) (fib (n-1) : list)
```

fås:

```
listfib 20 []
[1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,1597,2584,4181]
```

En lidt mere tricky version er:

```
fh (a,b) n
       | n < 0          =  error "negativt argument"
       | n==1           =  (a,b)
       | n`mod`2 == 0   =  (u,v)
       | otherwise      =  (b*v+a*u,(a+b)*v+b*u)
                           where (u,v)=fh (a*a+b*b,(b+2*a)*b) (n `div` 2)
```

Opgave 4 (25 %)

4.a Der findes faktisk en indbygget `reverse`, men ellers kan man jo lave sin egen:

```
palin(A):-rev(A,X),A=X.
rev([],[]).
rev([H|T],L):-rev(T,RevT),append(RevT,[H],L).
```

En udførelse er så:

```
| ?- palin([1,2,1]).
      1    1  Call: palin([1,2,1]) ?
      2    2  Call: rev([1,2,1],_84) ?
      3    3  Call: rev([2,1],_108) ?
      4    4  Call: rev([1],_132) ?
      5    5  Call: rev([],_156) ?
      5    5  Exit: rev([],[]) ?
      6    5  Call: append([],[1],_184) ?
      6    5  Exit: append([],[1],[1]) ?
      4    4  Exit: rev([1],[1]) ?
      7    4  Call: append([1],[2],_213) ?
      7    4  Exit: append([1],[2],[1,2]) ?
```

```
3    3  Exit: rev([2,1],[1,2]) ?
8    3  Call: append([1,2],[1],_244) ?
8    3  Exit: append([1,2],[1],[1,2,1]) ?
2    2  Exit: rev([1,2,1],[1,2,1]) ?
1    1  Exit: palin([1,2,1]) ?

(10 ms) yes
```

Men der er også den helt simple: `palin(A) :- rev(A,A).`

4.b 1: $X = g(g(Y))$ og $Z = g(Y)$.

2: Her er der ingen løsning. ($Y = g(X)$ og $X = h(Y)$ kan kun løses hvis $g = h^{-1}$)

3: Ej heller her! (kræver $X = Z = a$ og $a = b$)

4: `W=[1|A], B=[1|A], X=1, C=[2,1,2]`

4.c Bogens transformationssystem giver følgende:

```
| ?- clausify(all(X,exists(Y,c(X,Y)->((~b(X))&exists(Z,a(Y,Z)&d(X)))))).

implout: all(_16,exists(_17,~c(_16,_17)# ~b(_16)&exists(_25,a(_17,_25)&d(_16))))

negin: all(_16,exists(_17,~c(_16,_17)# ~b(_16)&exists(_25,a(_17,_25)&d(_16))))

skolem: all(_16,~c(_16,f1(_16))# ~b(_16)&a(f1(_16),f2(_16))&d(_16))

univout: ~c(_16,f1(_16))# ~b(_16)&a(f1(_16),f2(_16))&b(_16)

conjnf: (~c(_16,f1(_16))# ~b(_16))&(~c(_16,f1(_16))#a(f1(_16),f2(_16)))
                                                  & ~c(_16,f1(_16))#d(_16)

clausify: [cl([],[c(_16,f1(_16)),b(_16)]),
           cl([a(f1(_16),f2(_16))],[c(_16,f1(_16))]),
           cl([d(_16)],[c(_16,f1(_16))])|_362]

:- c(_16,f1(_16)), b(_16).
a(f1(_16),f2(_16)) :- c(_16,f1(_16)).
d(_16) :- c(_16,f1(_16)).
```

Svarende til

```
?- c(X,f1(X)),b(X).

a(f1(X),f2(X)) :- c(X,f1(X)).

d(X) :- c(X,f1(X)).
```

med de originale variabelnavne: `X, Y=f1(X) og Z=f2(X).`