PK, 27. maj 2003

Løsningsforslag til DM22 eksamen juni 2003

Opgave 1 (25 %)

1.a: Det ses let at:

$$[a_0] = a_0, \quad \text{og} \quad [a_0/a_1] = \frac{a_0 a_1 + 1}{a_1} = a_0 + \frac{1}{a_1},$$

samt

$$\tfrac{p_i}{q_i} = [a_0/a_1/\cdots/a_{i-1}/a_i] = [a_0/a_1/\cdots/a_{i-1} + \tfrac{1}{a_i}].$$

Vi har da induktions-skridtet:

$$\begin{aligned}
[a_0/a_1/\cdots/a_i/a_{i+1}] &= [a_0/a_1/\cdots/a_i + \tfrac{1}{a_{i+1}}] \\
&= \frac{\left(a_i + \frac{1}{a_{i+1}}\right) p_{i-1} + p_{i-2}}{\left(a_i + \frac{1}{a_{i+1}}\right) q_{i-1} + q_{i-2}} \\
&= \frac{a_{i+1}(a_i p_{i-1} + p_{i-2}) + p_{i-1}}{a_{i+1}(a_i q_{i-1} + q_{i-2}) + q_{i-1}} \\
&= \frac{a_{i+1} p_i + p_{i-1}}{a_{i+1} q_i + q_{i-1}} = \frac{p_{i+1}}{q_{i+1}}.
\end{aligned}$$

1.b: En løsning der anvender list comprehension er:

```
f :: Int -> (Int,Int) -> (Int,Int)
f a (p,pp) = (a*p+pp,p)

co :: [Int] -> [(Int,Int)] -> [(Int,Int)]
co (a:as) l@((p,pp):ps) = co as (ff:l)
                          where ff = f a (p,pp)
co [] x = x

convergents :: [Int] -> [(Int,Int)]
convergents as = reverse (init (zip pp qq))
                 where
                      pp = [ fst h | h <- co as [(1,0)]]
                      qq = [ fst h | h <- co as [(0,1)]]
```

som med Hugs f.eks. giver

```
Main> convergents [1,1,1,1,1,1,1,1]
[(1,1),(2,1),(3,2),(5,3),(8,5),(13,8),(21,13),(34,21)]
Main> convergents [0,2,3,1]
[(0,1),(1,2),(3,7),(4,9)]
Main>
```

Den indsatte `reverse` er blot for at få konvergenterne ud i den samme rækkefølge som partialkvotienterne er givet i.

1.c En mulig løsning er:

```
contFrac :: (Int,Int) -> [Int]
contFrac (p,q) = reverse (fst (cf [] (p,q)))

cf :: [Int] -> (Int,Int) -> ([Int],(Int,Int))
cf as (p,0) = (as,(p,0))
cf as (p,q) = cf (a:as) (q,pp)
              where a  = p `div` q
                    pp = p `mod` q
```

som f.eks. giver:

```
Main> contFrac (34,21)
[1,1,1,1,1,1,2]
Main> contFrac (-17,3)
[-6,3]
Main>
```

Opgave 2 (25 %)

2.a: Mulig løsning:

```
msort([],[]).
msort([X],[X]):-!.
msort(X,Y) :- split(X,S1,S2),
              msort(S1,SS1),
              msort(S2,SS2),
              merge(SS1,SS2,Y).

split([],[],[]).
split([X],[],[X]).
split([X,Y|Z],[X|V],[Y|W]) :- split(Z,V,W).

merge([X],[],[X]).
merge([],[X],[X]).
merge([X|A],[Y|B],[X|C]) :- X<Y,  merge(A,[Y|B],C).
merge([X|A],[Y|B],[Y|C]) :- X>=Y, merge([X|A],B,C).
```

2.b: Haskell-bogen arbejder uformelt, her er argumentet at der er tale om $\log n$ dybde, hvor hvert niveau kræver $\Theta(n)$ prædikatkald (`split` og `merge`) plus noget konstant, dvs. $\Theta(n \log n)$. Formelt har vi $T(n) = 2T(n/2) + \frac{3}{2}n$ der har løsningen $T(n) \in \Theta(n \log n)$.

Opgave 3 (25 %)

3.a: Der er tale om den velkendte

```
partition      :: (a -> Bool) -> [a] -> ([a],[a])
partition p xs  = foldr select ([],[]) xs
                  where select x (ts,fs) | p x       = (x:ts,fs)
                                         | otherwise = (ts,x:fs)
```

3.b: Funktionen `powS` har lineær tids- og pladskompleksitet, $\Theta(n)$, og `pow` har logaritmisk plads og tid, $\Theta(\log n)$.

3.c: Kan for $n \geq 0$ gøres ved induktion baseret på basetilfældet $x^0 = 1$, samt anvendelse af at $x^{2n} = (x^n)(x^n)$ og $x^{2n+1} = x(x^n)(x^n)$, ved induktion i eksponenten.

Basetilfældet: `pow x 0 = 1` ved definitionen af `pow`.

Induktionsantagelsen: `pow x i = x`^{i}, for $i = 0, \cdots, n$.

Induktion: Her er to tilfælde:

$\underline{n+1 = 2k}$

`pow` x (2k)=square(pow x (2k) ‘div‘ 2) = square(pow x k) = x^{2k},
per induktionsantagelsen.

$\underline{n+1 = 2k+1}$

`pow` x (2k+1)= x*square(pow x ((2k+1)–1)‘div‘ 2) = x*square(pow x k) = x^{2k+1},
per induktionsantagelsen.

Tilfældet $n < 0$ følger af definitionen $x^{-n} = \frac{1}{x^n}$.

Opgave 4 (25 %)

4.a: En mulig løsning er:

```
between(X,X,Z) :- X =< Z.
between(X,Y,Z) :- X < Z, W is X+1, between(W,Y,Z).
```

4.b:

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

2: Her er der ingen løsning.

3: $X = g(a)$, $Y = a$ og $Z = g(g(a))$.

4.c:

```
| ?- clausify(all(X,all(Z,s(X,Z)->(m(X)#exists(Y,(n(X,Y)&m(Y))))))).

implout: all(_15,all(_16,~s(_15,_16)#m(_15)#exists(_22,n(_15,_22)&m(_22))))

negin: all(_15,all(_16,~s(_15,_16)#m(_15)#exists(_22,n(_15,_22)&m(_22))))

skolem: all(_15,all(_16,~s(_15,_16)#m(_15)#n(_15,f2(_16,_15))&m(f2(_16,_15))))

univout: ~s(_15,_16)#m(_15)#n(_15,f2(_16,_15))&m(f2(_16,_15))

conjnf: (~s(_15,_16)#m(_15)#n(_15,f2(_16,_15)))& ~s(_15,_16)#m(_15)#m(f2(_16,_15))

clausify: [cl([m(_15),n(_15,f2(_16,_15))],[s(_15,_16)]),
                              cl([m(_15),m(f2(_16,_15))],[s(_15,_16)])|_235]

m(_15); n(_15,f2(_16,_15)) :- s(_15,_16).
m(_15); m(f2(_16,_15)) :- s(_15,_16).
```

Svarende til

```
m(X) ; n(X,f2(Y,X)) ; m(f2(Y,X)) :- s(X,Y).
```