

Module Title: Informatics 1a
Exam Diet (Dec/April/Aug): Aug 2007
Brief notes on answers:

```
import Char

-- 1a

f :: [Int] -> Int
f xs = product [ x+1 | x <-xs, x >= 2]

x1a = f [1,2,0,3,4,-25] == 60

-- 1b

f' :: [Int] -> Int
f' [] = 1
f' (x:xs) | x >= 2 = (x+1) * f' xs
           | otherwise = f' xs

x1b = f' [1,2,0,3,4,-25] == 60

-- 1c

f'' :: [Int] -> Int
f'' xs = foldr (*) 1 (map (+1) (filter (>= 2) xs))

x1c = f'' [1,2,0,3,4,-25] == 60

x1 = x1a && x1b && x1c

-- 2a

combine :: Char -> Char -> Char
combine '*' '*' = '*'
combine '*' y | isAlpha y = y
combine x '*' | isAlpha x = x
combine x y | isAlpha x && isAlpha y = '*'

x2a = combine 'a' '*' == 'a'&& combine '*' 'b' == 'b' &&
      combine '*' '*' == '*'&& combine 'a' 'b' == '*'

-- 2b

combines :: String -> String -> String
combines xs ys | length xs == length ys = [ combine x y | (x,y) <- zip xs ys ]
```

```

x2b = combines "ab**" "**cd" == "abcd" && combines "*bc*" "**cd" == "*b*d"

-- 2c

combines' :: String -> String -> String
combines' [] [] = []
combines' (x:xs) (y:ys) = combine x y : combines' xs ys

x2c = combines' "ab**" "**cd" == "abcd" && combines' "*bc*" "**cd" == "*b*d"

x2 = x2a && x2b && x2c

-- 3a

hide :: Int -> Int -> String -> String
hide m n xs = replicate m '*' ++
              drop m (take n xs) ++
              replicate (length xs - n) '*'

x3a = hide 2 5 "abcdefgh" == "***cde***"

-- 3b

hide' :: Int -> Int -> String -> String
hide' m n xs = h 0 xs
  where
    h i [] = []
    h i (x:xs) | m <= i && i < n = x : h (i+1) xs
               | otherwise      = '*' : h (i+1) xs

x3b = hide' 2 5 "abcdefgh" == "***cde***"

x3 = x3a && x3b

x = x1 && x2 && x3

```

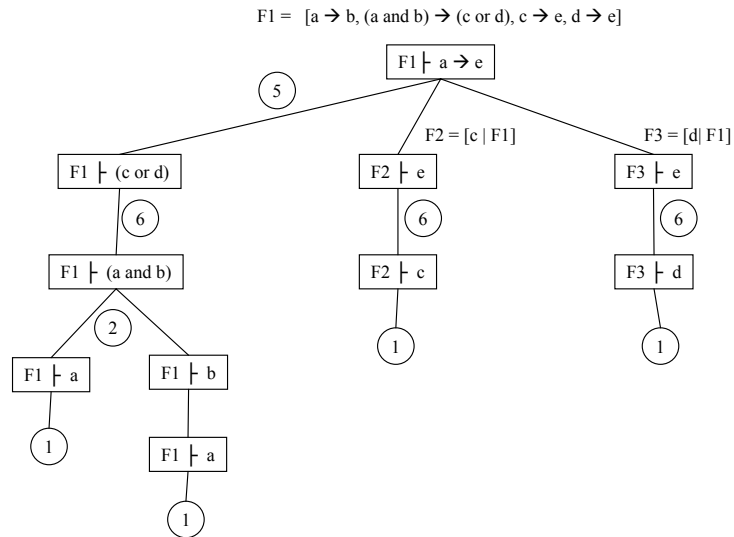
4(a) An appropriate truth table (where columns 7 and 11 are our two expressions) is:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|-----|-----|-----------------|-----------------|-------------------|---------|-------------------|------------------------|---------|---------|------------------|
| a | b | $\text{not}(a)$ | $\text{not}(b)$ | $a \rightarrow b$ | 5 and 4 | $6 \rightarrow 3$ | $\text{not}(a)$ or b | 8 and 4 | 9 and 1 | $\text{not}(10)$ |
| t | t | f | f | t | f | t | t | f | f | t |
| t | f | f | t | f | f | t | f | f | f | t |
| f | t | t | f | t | f | t | t | f | f | t |
| f | f | t | t | t | t | t | t | t | f | t |

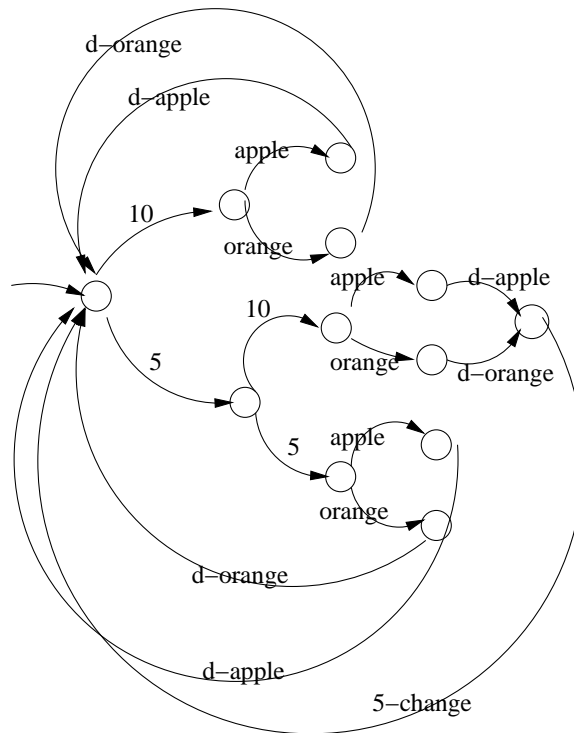
4(b) An appropriate sequence of transformations is:

$$\begin{array}{c}
 ((a \rightarrow b) \text{ and } \text{not}(b)) \rightarrow \text{not}(a) \\
 \Downarrow \\
 ((\text{not}(a) \text{ or } b) \text{ and } \text{not}(b)) \rightarrow \text{not}(a) \\
 \Downarrow \\
 \text{not}(((\text{not}(a) \text{ or } b) \text{ and } \text{not}(b)) \text{ and } \text{not}(\text{not}(a))) \\
 \Downarrow \\
 \text{not}(((\text{not}(a) \text{ or } b) \text{ and } \text{not}(b)) \text{ and } a)
 \end{array}$$

5 Students are expected to draw a proof tree or some other similar form of representation like the one below:



6(a) An appropriate FSM is as follows:



6(b) The difficulty with this extension to the FSM is that, if the maximum number of coins the machine can retain is unbounded, the FSM has to count to an arbitrary high number which it cannot do. One correct answer, therefore, is “no” for the reason given. The other correct answer is “yes” under the assumption that we limit the number of coins retained so that we have a finite counting problem. We then could construct the necessary FSM.

7 An appropriate FSM is as follows:

