Chapter 4 Homework:

12/4/2010

Q: Read Phil Wadler's Paper "Theorems for Free!" - www.mpi-sws.org/ dreyer/tor/papers/wadler.pdf

A: Finished. Hardly understood a thing. I need to get a better general idea of: lambda calculus (esp. notation), category theory, and most of the vocabulary used in this paper. Will place items on the to-do list.

Q: Define safetail, so that it acts just like tail except that it maps the empty list to the empty list rather than being undefined for that input. *Hint*: the function null can be used to test if a list is empty.

Define using each of the following:

1. A Conditional Expression
2. Guarded Equations
3. Pattern Matching

A: Here are all three versions, each of which use the following definition of the "unsafe" version of tail, called "mytail", though this could have easily been incorporated into each definition.

```
mytail (x:xs) = xs
```

1. Conditional:

```
csafetail xs = if xs == []
                  then []
                  else mytail xs
```

2. Guarded Equations:

```
gsafetail xs | xs == [] = []
             | otherwise =  tail xs
```

3. Pattern Matching:

```
psafetail [] = []
psafetail (x:xs) = xs
```

Q: Give three possible definitions for the logical (||) "or" operator using pattern matching.

A: The third seems to be preferable, and not sure that the other two are that distinct.

```
(||1) :: Bool → Bool → Bool
True ||1 True = True
True ||1 False = True
False ||1 True = True
False ||1 False = False

(||2) :: Bool → Bool → Bool
x ||2 x = x
x ||2 y = if x
             then True
             else y


(||3) :: Bool → Bool → Bool
True ||3 _ = True
False ||3 x = x
```

Q: Redefine the following version of "&&" using conditionals rather than pattern matching:

```
True && True = True
_ && _ = False
```

A:

```
(&&cond) x y = if x == y
                  then if x == True
                          then True
                          else False
                  else False
```

Q: Do the same for the following version:

```
True && b = b
False && _ = False
```

A:

```
(&&cond2) x y = if x == True
                   then y
                   else False
```