Exercise sheet 1

Max Wisniewski, Alexander Steen

Task 1

What is the fold function corresponding to the |Either| datatype and what is its prelude correspondence?

Solution:

```
data Either a b = Left a | Right b

foldEither :: (a -> c) -> (b -> c) -> Either a b -> c
foldEither f _ (Left a)   = f a
foldEither _ g (Right b)  = g b
```

The corresponding function is called ‘either‘.

Task 2

Express the function |pow m n = m^n| (|m| to the power of |n|) as a fold.
Solution:

```
data Nat = O | S Nat

foldNat :: (a -> a) -> a -> Nat -> a
foldNat _ e O      = e
foldNat f e (S n)  = f ( foldNat f e n )

plus :: Nat -> Nat -> Nat
plus a b = foldNat S a b

mult :: Nat -> Nat -> Nat
mult a b = foldNat (plus a) b

pow :: Nat -> Nat -> Nat
pow a b = foldNat (mult a) b
```

PROOF??

Task 3

Express the predecessor function |pred :: Nat -> Maybe Nat| as a fold.
Solution:

In the following we use ‘maybe‘ which is the fold for Maybe in the haskell prelude.

```
> pred :: Nat -> Maybe Nat
> pred  = foldNat (maybe (Just O) (Just . S)) Nothing
```

We strip of one of the successors by a double application of the fold. The zero value ‘O‘ is mapped to ‘Nothing‘ in the anchor. In the first application ‘Nothing‘ is mapped to ‘Just O‘. In the following the value in the Monad will be incremented, but the first application no increment was done. Therefore we have done one increment less then the original number required and this is therefore the predecessor.

Task 4

Complete the proof that |‘plus‘| is commutative
Solution:

For completeness we will do the whole proof here.

claim 1. *The function ‘plus‘ is commutative.*

```
plus m n = plus n m
```

⌋

To proof this claim we first proof some lemma.

lemma 1. *Adding a zero to a number will return the number again.*

```
plus O n = n
```

⌋

lemma 2. *We can increment the first parameter instead of the second without changing the result.*

```
plus m (S n) = plus (S m) n
```

⌋

Proof lemma 1.
Induction on n.

I.A. $n = 0$

```
plus O O = O
```

holds.

I.S. $n \rightsquigarrow Sn$

$$\begin{aligned} \textit{plus 0 (S n)} &\stackrel{\text{Def. plus}}{=} \textit{S (plus 0 n)} \\ &\stackrel{\text{ind. hyp.}}{=} \textit{S n} \end{aligned}$$

By induction the claim holds.

□

Proof lemma 1.
Induction on n.

I.A. $n =$O

```
plus m (S O) = S (plus m O)
             = S m
             = plus (S m) O
```

I.S. $n \rightsquigarrow Sn$

$$\begin{aligned} plus\ m\ (S\ (S\ n)) &\overset{\text{Def. plus}}{=} S\ (plus\ m\ (S\ n)) \\ &\overset{\text{ind. hyp.}}{=} S\ (plus\ (S\ m)\ n) \\ &\overset{\text{Def. plus}}{=} plus\ (S\ m)\ (S\ n) \end{aligned}$$

By induction the claim holds.

□

Proof claim 1.
Let m be arbitrary but fixed.
Induction on n.

I.A. $n =$O

$$\begin{aligned} plus\ m\ O &\overset{\text{Def. plus}}{=} m \\ &\overset{\text{lemma 1}}{=} plus\ O\ m \end{aligned}$$

I.S. $n \rightsquigarrow Sn$

$$\begin{aligned} plus\ m\ (S\ n) &\overset{\text{Def. plus}}{=} S\ (plus\ m\ n) \\ &\overset{\text{ind. hyp.}}{=} S\ (plus\ n\ m) \\ &\overset{\text{Def. plus}}{=} plus\ n\ (S\ m) \\ &\overset{\text{lemma 2}}{=} plus\ (S\ n)\ m \end{aligned}$$

By induction the claim holds.

□

We concluded that the function ‘plus‘ is commutative.