

Defining Parker's Concatenation

Eddie Antonio Santos

April 18, 2017

1 Implementation

```
{-# LANGUAGE MagicHash #-}  
import GHC.Integer.Logarithms ( integerLogBase# )
```

First, we wish to determine how many digits there are in a number. This will indicate how many times we should multiply a by ten.

Digits is defines as:

$$digits(x) = 1 + \lfloor \log_{10} x \rfloor$$

```
digits :: (Integral a) => a -> a  
digits i = succ . floor . log10 . fromIntegral $ i
```

Temporary:

```
pconcat :: Integral a => a -> a -> a  
pconcat a b = undefined
```

Appendix: Main function

```
main = putStrLn $ show (6 `pconcat` 1)
```

Appendix: \log_{10}

This one gets weird. In order to get decent accuracy for \log_{10} , we need to define double precision of $\ln 10$ elsewhere, then use it. See <http://stackoverflow.com/a/11293064/6626414>.

```
ln10 :: Double  
ln10 = log 10
```

Now we have to **explicitly** define \log_{10} as being defined on `Double`.

```
log10 :: Double -> Double  
log10 y = log y / ln10
```