

2021 年 1 月

2021-01-14 (Thu)

[Bunny]

0.9.x 対応項目 より、[P2] Read [a] に対応。ついでに、Read () もぬけていたので足しておいた。いずれも、lib/Prelude.hs への追加。

対応項目リストから漏れているが、Show Char, Read Char などぬけているなあ。

なお、Read Double, Read Float も対応しようかなとおもい、Haskell98 より ReadFloat をもってきてうごかそうとしたが、それは失敗。

```
-- import          Data.Ratio
-- import          Numeric

readFloat' :: (RealFrac a) => ReadS a
readFloat' r = [(fromRational ((n%1)*10^(k-d)), t) | (n, d, s) <- readFix r
                                                    , (k, t) <- readExp s] ++
               [(0/0, t) | ("NaN", t) <- lex r] ++
               [(1/0, t) | ("Infinity", t) <- lex r]

where
  readFix r = [(read (ds ++ ds'), length ds', t)
              | (ds, d) <- lexDigits r
              , (ds', t) <- lexFrac d]

  lexFrac ('.':ds) = lexDigits ds
  lexFrac s       = [("", s)]

  readExp :: String -> [(Int, String)]
  readExp (e:s) | e `elem` "eE" = readExp' s
  readExp s       = [(0, s)]

  readExp' ('-' : s) = [(-k, t) | (k, t) <- readDec s]
  readExp' ('+' : s) = readDec s
  readExp' s         = readDec s

x :: [(Double, String)]
x = readFloat' "3.14"

y :: [(Double, String)]
```

```
y = readFloat' "1.23e-5"
```

```
main = do print x  
         print y
```

readExp に型シグネチャをつけないと、ランタイムにおいて assertion error が起こった。Int の引き算で型不一致を検出しているようだったので、猪口才 defaulting がわるさしているのかと思い、型シグネチャをつけてみたのだが、こんどはコンパイルでこける。

要調査。

Project Euler 4, 5

息抜きがてら Project Euler 4^{*1} や 5 を Bunny で動かしてみる。これらは問題なし。

*1 https://wiki.haskell.org/Euler_problems/1_to_10#Problem_4