

072: Integral の定義をただしく

↑ up

- issued: 2020-05-11
- 分類: 分類: C 改善項目
- status: Closed (2020-05-14)

関連: 068

概要

ペアへの bind (例 `let (q, r) = divMod n d` など) がまだできないため、Integral クラス、インスタンス定義は問題を回避した記述になっている。

```
class (Enum a) => Integral a where
  quot, rem      :: a -> a -> a
  div, mod       :: a -> a -> a
  quotRem, divMod :: a -> a -> (a, a)
  toInteger      :: a -> Integer
  -- Minimal complete definition: quotRem, toInteger
  n 'quot' d = fst $ quotRem n d
  n 'rem'  d = snd $ quotRem n d
  n 'div'  d = fst $ divMod n d
  n 'mod'  d = snd $ divMod n d
{-
  n 'quot' d = q where (,) q r = quotRem n d
  n 'rem'  d = r where (,) q r = quotRem n d
  n 'div'  d = q where (,) q r = divMod n d
  n 'mod'  d = r where (,) q r = divMod n d
  divMod n d = if sinum r == - signum d then (q-1, r+d) else (q, r)
    where (,) q r = quotRem n d
-}
```

これを、Langage Report と同じ書き方に戻したい。

調査ログ

2020-05-14 (Thu)

068 対処により、Language Report と同じ書きかたが可能となった。

```
class (Enum a) => Integral a where
  quot, rem      :: a -> a -> a
  div, mod       :: a -> a -> a
  quotRem, divMod :: a -> a -> (a, a)
  toInteger      :: a -> Integer
  -- Minimal complete definition: quotRem, toInteger
  n `quot` d = q where (q,r) = quotRem n d
  n `rem` d = r where (q,r) = quotRem n d
  n `div` d = q where (q,r) = divMod n d
  n `mod` d = r where (q,r) = divMod n d
  divMod n d = if signum r == - signum d then (q-1, r+d) else qr
    where qr@(q,r) = quotRem n d
```

2021-11-01 (Mon)

↑これで Close なんじゃないの？おそらく閉め忘れということで 2020-05-14 クローズとする。