

# 構成的プログラムと型

日比野 啓

2013-05-29

# 本日の概要

- 型とプログラム
- 関数と構成的プログラム
- 値と型付け
- 論理パズルの例

質問は随時受けつけます

# 本日の概要

- 型とプログラム
- 関数と構成的プログラム
- 値と型付け
- 論理パズルの例

# 型とプログラム

あらためて型とは ？

- 型とはその型を持つ値の性質を示すもの

# 型とプログラム

あらためて型とは ？

- 型とはその型を持つ値の性質を示すもの

プログラムとは ？

- 目的の値を作り出すもの

# 型とプログラム

あらためて型とは ？

- 型とはその型を持つ値の性質を示すもの

プログラムとは ？

- 目的の値を作り出すもの

正しいプログラムは

正しく型が付いた値を作り出す

- 正しいプログラムは作り出した値の性質を証明する

# 型とプログラム - 例 - Even0.hs

値を検査し、  
それが正しい場合においてのみ `Even` を作る

```
-- data Maybe a = Just a
--                  / Nothing

data Even = Even' Int

mayEven :: Int -> Maybe Even
mayEven i
  | i `rem` 2 == 0 = Just (Even' i)
  | otherwise      = Nothing
-- 例えば Even' は他からは隠蔽する
```



※ 画面切り替え練習

# 実習

```
% ghci Even0.hs
```

```
...
```

```
> mayEven 10
```

```
...
```

```
> mayEven 5
```

※ Tab で補完できる

# 型とプログラム - 例 - Even1.hs

検査しなくても Even を作るくことができる公理系の例

```
data Even = Double Int
```

```
double :: Int -> Even
```

```
double i = Double i
```

```
mayEven :: Int -> Maybe Even
```

```
mayEven i
```

```
  | rm == 0    = Just (double qt)
```

```
  | otherwise = Nothing
```

```
  where (qt, rm) = i `quotRem` 2
```

# 実習

```
% ghci Even1.hs
...
> double 5
...
> mayEven 10
...
> mayEven 5
```

# 型とプログラム - まとめ

- 型とはその型を持つ値の性質を示すもの
- プログラムとは目的の値を作り出すもの
- 正しいプログラムは正しく型が付いた値を作り出し、その性質を証明する

# 本日の概要

- 型とプログラム
- 関数と構成的プログラム
- 値と型付け
- 論理パズルの例

# 関数

関数とは？

- 値の持つ性質から値の持つ性質を導き出す

# 関数

関数とは？

- 値の持つ性質から値の持つ性質を導き出す
- 関数は証明における推論規則



# 関数

関数とは？

- 値の持つ性質から値の持つ性質を導き出す
- 関数は証明における推論規則

プログラムは推論規則の組み合わせで証明を構成する

# 関数 - 例 - Even2.hs

さらに別の公理系を考える

```
data EvenPrime = Even' Int
```

```
zero' :: EvenPrime
```

```
zero' = Even' 0
```

```
plus2' :: EvenPrime -> EvenPrime
```

```
plus2' (Even' i) = Even' (i + 2)
```

# 関数 - 例 - Even3.hs

拡張すると、 Even を作り出す加算ができるように

```
data Even = Even
           (EvenPrime -> EvenPrime)
```

```
zero :: Even
zero = Even id
```

```
two :: Even
two = Even plus2'
```

```
(<+>) :: Even -> Even -> Even
(Even f) <+> (Even g) = Even (f . g)
```

# 実習

```
% ghci useEven3.hs  
> zero '  
> plus2' (plus2' (plus2' zero'))  
> zero  
> two  
> four  
> six  
> four <+> six
```

# 関数

- ① 0 と 2 の繰り返し加算による公理系 (EvenPrime) を定義した
- ② EvenPrime を拡張した Even の定義から、Even と Even の加算が Even になることを導出できた

質問

# 本日の概要

- 型とプログラム
- 関数と構成的プログラム
- 値と型付け
- 論理パズルの例

# 値と型付け

型とはその型を持つ値の性質を示すもの

- 値の性質  $\doteq$  型の意味  
→ 値の性質  $\leq$  型の意味



# 値と型付け

型とはその型を持つ値の性質を示すもの

- 値の性質  $\models$  型の意味  
→ 値の性質  $\leq$  型の意味

幽霊型 - Phantom Type

# 値と型付け - 幽霊型 - 例 - Maybe.hs

```
noInt :: Maybe Int
noInt =  Nothing
```

```
noString :: Maybe String
noString =  Nothing
```

```
-- noInt == noString
```

# 実習

```
% ghci Maybe.hs  
> justOne  
> justHello  
> noInt  
> noString  
> justOne == justOne  
> noInt == noString
```

# 値と型付け - 幽霊型 - 例 - IntUnit.hs

```
data Int' u = Int' Int
```

```
(<+>) :: Int' u -> Int' u -> Int' u  
Int' a <+> Int' b = Int' (a + b)
```

```
data YenUnit = Yen  
type Yen = Int' YenUnit
```

```
data GramUnit = Gram  
type Gram = Int' GramUnit
```

# 実習

```
% ghci IntUnit.hs
> twentyYen
> :t twentyYen
> thirtyGram
> :t thirtyGram
> twentyYen <+> twentyYen
> twentyYen <+> thirtyGram
```

# 実習

```
> :i Yen
> :t impose
> :i Imposed
> :t impose twentyYen
> :t twentyImposed
> twentyImposed <+> twentyImposed
> twentyYen <+> twentyImposed
```

質問

# 本日の概要

- 型とプログラム
- 関数と構成的プログラム
- 値と型付け
- 論理パズルの例



# 論理パズル

<http://ja.wikipedia.org/wiki/ロジックパズル>  
から持ってきた問題

# 論理パズル - 力技の解法 - L0.hs

```
% ghci Even4.hs  
...  
> good
```

# 論理パズル - 推論規則を利用した解法

## - solve.hs

```
% ghci solve.hs  
...  
> solve  
> :t solve
```

質問

# 参考文献

- Software Foundations  
<http://www.cis.upenn.edu/~bcpierce/sf/>
- Software Foundations (日本語訳)  
<http://github.com/sfja/sfja>