プログラムの数理
Mathematical Structures in Programs

胡　振江

平成16年度冬学期

講義の概要

目的

本講義では算法**言語の基本概念**を関数プログラミングを通して修得する。

関数プログラミングは算法設計・プログラミングを**数学的な活動**としてとらえる考え方であり、本講義ではそれをプログラミング言語Haskellを用いて具体的に示すとともに、厳密な**科学・工学としてのプログラミングのあり方**を学ぶ。

内容

- 関数プログラミング言語Haskellの学習
  - プログラム：関数の定義
  - プログラムの実行：式の簡約
- 関数プログラミングの特徴の理解
  - 問題の記述：抽象的
  - プログラム構造：構成的
  - プログラム間の関係：推論，操作しやすい
  - プログラム性質：証明しやすい

他講義との関係

プログラムの数理（3年）

計算モデルの数理（4年）

教科書

- 武市正人訳、「関数プログラミング」, 近代科学社, 1994年. ISBN4-7649-0181-1
(R. Bird and P. Wadler, Introduction to Functional Programming, Prentice Hall, 1988)

参考書など

- Richard Bird. Introduction to Functional Programming in Haskell, Prentice Hall, 1998.
- 講義ページ:
  http://www.ipl.t.u-tokyo.ac.jp/~hu/pm04/

日程

- 10月:4, 11(祝日), 18, 25
- 11月:1, 8, 15(実習), 22, 29
- 12月:6, 13, 20
- 1月:10(祝日), 17, 24, 31(質疑)

欠席，遅刻しないよう

評価・成績

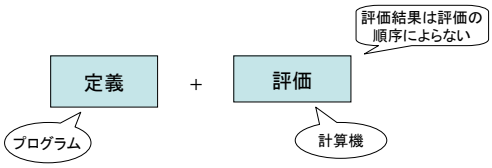
出席	20%
レポート	20%
期末試験	60%

学習方法

- 講義で内容を理解すること。
- 練習問題をやること。
- プログラムを書くこと。

関数プログラミングの基本的な概念

関数プログラミング



セッション (session)

```
Prelude> 2+5*3
17
Prelude> sin(1) + cos(1)
1.3817732906760363
Prelude> pi
3.141592653589793
Prelude> 7/2
3.5
```

スクリプト (Script): 関数定義

プログラム名: xxx.hs

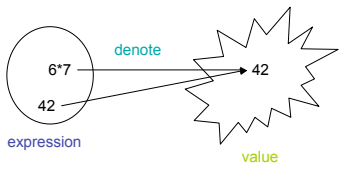
定義の並び

```
square x = x * x
min x y | x<=y      = x
        | otherwise = y
```

Prelude.hs: よく使われている関数定義の集まり

式と値

「式」は「値」を「表す」



参照透明性(reference transparency):

-- 変数の定義と同一の文脈中で変数の表す値は変らない。

簡約 (reduction)

```
square (3+4)
⇒ square 7
⇒ 7*7
⇒ 49
```

```
square (3+4)
⇒ (3+4) * (3+4)
⇒ 7*(3+4)
⇒ 7*7
⇒ 49
```

- 標準形 (canonical form)、正規形 (normal form)
  -- それ以上簡約できない式のこと。
- 底要素 (bottom value): ⊥
  -- 定義されない「値」（例: 1/0 ）

型 (type)

- 基本型: Int, Bool, Char, String
- リスト型: [t]
- 組型: (t1, t2, …, tn)
- 関数型: t1 -> t2
- 代数型、抽象型

強い型決め (strong typing)

関数と定義

f :: A -> B

- 関数定義

  自動導出可能

  ```
  double :: Int -> Int
  double x = x * x
  ```

- 関数適用

  double 5 ==> 25

- 多様型関数 (polymorphic function)

  ```
  id :: a -> a
  id x = x
  ```

定義の形式

場合分け (case analysis)

guard

```
min x y | x <= y  = x
        | x > y   = y
```

局所的な定義 (local definition)

local definition

```
f x y | x >= 0  = x + a
      | otherwise =  x - a
  where a = square (y+1)
```

カリー化 (currying)

```
min' (x, y) | x <= y  = x
            | x > y   = y
```

1引数関数

カリー化

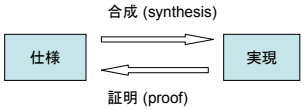
```
min x y | x <= y  = x
        | x > y   = y
```

2引数関数

```
min :: Int -> (Int -> Int)
```

仕様と実現

- 仕様 (specification)
  - 問題の数学的記述
- 実現 (implementation)
  - 仕様を満たすプログラム



Running Haskell Programs

Hugs をインストールする。

http://www.haskell.org/hugs/

(ECC has Hugs installed)

Primitive Library: Prelude.hs

Extended Library: Char.hs, List.hs, System.hs, …

Your Program: Test.hs, …

宿題

- 教科書を購入し，第一章を読む．
- Hugs をインストールする．
- Hugsを使ってみる．
  - http://cvs.haskell.org/Hugs/pages/hugsman/basics.html を読む
  - （スライド中の）例を確認する