

## ソフトウェア開発の形式工学手法 Formal Engineering Method for Software

胡 振江(東京大学)  
HU, Zhenjiang (Univ. of Tokyo)



## 講義の概要



## 目的

- 本講義では**正しくて効率のよいプログラム**を開発するための基本手法を**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。



## 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)



## 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



## 具体的な内容

1. 関数プログラミング言語の基本概念
2. 基本データ型
3. リストの表現
4. リスト処理の基本関数
5. リスト処理の例
6. 再帰的な定義
7. 帰納法による証明
8. 簡約計算モデル
9. プログラムの効率
10. 無限リスト
11. 循環構造



### 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/FEM05/>



### 日程

- 4月: 13, 20, 27
- 5月: 4(祝日), 11, 18, 25
- 6月: 1, 8, 15, 22, 29
- 7月: 6(休講), 13(予備)

欠席, 遅刻しないよう



### 評価・成績

- |            |     |
|------------|-----|
| • レポート(2回) | 20% |
| • 中間テスト    | 20% |
| • 期末試験     | 60% |



### 学習方法

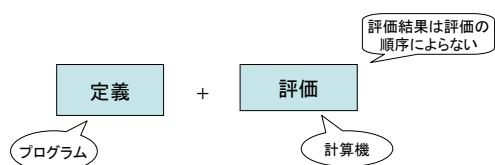
- 講義で内容を理解すること。
- 練習問題をやること。
- プログラムを書くこと。



### 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

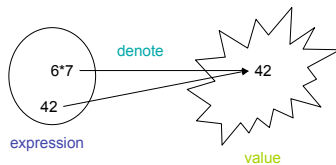
```
square x = x * x
min x y | x <= y    = x
        | otherwise = y
```

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わらない。



## 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):  $\perp$   
-- 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  
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)

$\text{min } x \ y \mid x \leq y = x$   
 $\mid x > y = y$

guard

局所的な定義 (local definition)

$f \ x \ y \mid x \geq 0 = x + a$   
 $\mid \text{otherwise} = x - a$   
 $\text{where } a = \text{square } (y+1)$

local definition



## カーリー化 (currying)

$\text{min}' \ (x, y) \mid x \leq y = x$   
 $\mid x > y = y$

1引数関数

↓ カーリー化

$\text{min } x \ y \mid x \leq y = x$   
 $\mid x > y = y$

2引数関数

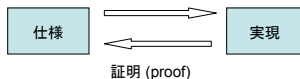
$\text{min} :: \text{Int} \rightarrow (\text{Int} \rightarrow \text{Int})$



## 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

合成 (synthesis)



## Running Haskell Programs

Hugs をインストールする。

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



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を試してみる。
  - <http://cvs.haskell.org/Hugs/pages/hugsman/basics.html> を読む
  - (スライド中の)例を確認する



## デモ

- Hugsの起動
- 式の計算
- Haskell プログラムの作成

