

# 「プログラムの数理」講義復習

胡 振江  
東京大学 計数工学科  
2006 年度

## 本講義の目的

- 算法言語の基本概念（文法，意味，型，計算モデル）を学習することによって，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．
- 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を理解する．計算機プログラムの数理的構造を捉え．それに基づいて構成的にアルゴリズムを記述する構成的アルゴリズム論 (constructive algorithmics) の基礎を扱う．

講義ページ:

<http://www.ipl.t.u-tokyo.ac.jp/~hu/pub/teach/pm06/>

## 講義内容 (1/2)

- **Part I: 関数プログラミング言語 Haskell の基本を学習する .**
  - ▶ 関数プログラム : 関数の定義
  - ▶ 関数プログラムの実行 : 式の簡約
  - ▶ 特徴 :
    - \* 抽象的 (algorithmic) な記述
    - \* 構成的なプログラム構造
    - \* 推論 ● 変換が容易

## 講義内容 (2/2)

- Part II: プログラムの演算法 (Program Calculation) を学習する .
  - ▶ Bird-Meertens Formalism: プログラムを構築するためのメタ理論
    - \* データ構造によるプログラムの構造化
    - \* プログラミング代数 (プログラミング理論の構築)

## 教科書 ● 参考書

- 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)
- Richard Bird, Lecture Notes on Constructive Functional Programming, Technical monograph 69; PRG, Oxford University, September 1988.

## 講義詳細

1. 関数プログラミングの基本概念
2. 基本データ型上の関数
3. 関数プログラムの設計
4. 合成型上の関数
5. 帰納法によるプログラムの証明
6. 構成的アルゴリズム論 1 (基本概念)
7. 構成的アルゴリズム論 2 (準同型)
8. 構成的アルゴリズム論 3 (方向付き簡約計算)
9. 構成的アルゴリズム論 4 (リストから行列への拡張)

## 関数プログラミングの基本概念

- 関数
- 関数プログラミング言語
- 関数プログラムの構成
  - ▶ 関数の定義
  - ▶ 型
- 関数プログラムの評価
- 関数プログラムの処理系

## 基本データ型上の関数

- 数型（整数型と浮動小数点数型）とその上の関数
- 論理型とその上の関数
- 文字型とその上の関数
- 文字列型とその上の関数

## 関数プログラムの設計

- 関数プログラムの設計法
  - ▶ 解きたい問題を理解する .
  - ▶ 関数の型を決め , 問題を解くための情報を集める .
  - ▶ 複雑問題を簡単な問題に分割して解く
- 再帰関数 : Loop の表現
- 例題 : 平方根の計算

## 合成型上の関数

- 組型とその上の関数
  - ▶ 基本関数 (構成子 + 分離関数)
  - ▶ 有理数上の計算
- 関数型とその上の関数
  - ▶ 関数合成
  - ▶ 逆関数
  - ▶ 正格関数・非正格関数
- リスト型とその上の関数
  - ▶ リスト上の再帰関数
  - ▶ リスト上の高階関数

## 帰納法によるプログラムの証明

- 自然数上の帰納法によるプログラムの証明
- リスト上の帰納法によるプログラムの証明
- 補助定理と一般化

## 構成的アルゴリズム論

A *calculus* of functions for deriving programs from their specifications:

- A range of concepts and notations for defining *functions* over various data types (including lists, trees, and arrays);
- A set of *algebraic laws* (rules, lemmas, theorems) for manipulating functions;
- A framework for *constructing new calculation rules* to capture principles of programming.

## Part I: Basic Concepts

- Notations on Functions
- Algebraic View of Lists
- List functions as Compositions of Homomorphisms
- Basic Calculation Rules for Derivation of Homomorphisms: Promotion Rules
- Horner's Rule
- Maximum Segment Sum Problem

## Part II: Homomorphisms

- Formalization of Homomorphism: Reduce after Map
- Program specification with Homomorphisms
- Point-free calculation rules for manipulating homomorphisms
- Longest All- $P$  Segment Problem

## Part III: Left Reductions

- General homomorphic equations and well-defined functions
- Left reduction: a sequential computation pattern
- Loops: implementation of left reduction
- Left-zeros
- The Minimax problem

## Part IV: Extension to Arrays

- Formalization of arrays as a binoid
- Formalization of well-defined functions on arrays as homomorphisms
- General rules: promotion/fusion
- Special rules: Horner's rule
- Program derivation by calculation

## 期末試験について

- 問題：前半は 4 割，後半は 6 割（arrays に関する部分を含まない）。
- ポイント：教科書中の内容と講義資料を理解し，演習問題を解くこと。
- 試験：教科書・講義資料など持ち込み可。

ご協力どうもありがとうございました！