計算モデルの数理 I: λ 計算

胡振江
東京大学計数工学科

2007 年度

復習: 3 年の「プログラムの数理」

関数プログラミング：

- プログラム：式 + 関数の定義
- 計算：式の簡約
- 計算の結果：式の正規形

その理論背景は？

参考資料

- 井田哲雄,「計算モデルの基礎理論」(第 4 章), 岩波講座ソフトウェア科学１２, 岩波書店, １９９１年、3700 円. (ISBN4-00-010352-0)
- H.P.Barendregt, ”The lambda calculus: its syntax and semantics”, Studies in logic and the foundations of mathematics, v.103, North-Holland, 1984. (ISBN 044487 5085). `http://www.andrew.cmu.edu/user/cebrown/notes/barendregt.html` から無料で入手可能.
- 高橋正子,「計算論 - 計算可能性とラムダ計算」, コンピュータサイエンス大学講座 24, 近代科学社, 1991 年, 3500 円. (ISBN 4-7649-0184-6)

計算モデル

$\boxed{\text{計算を理論的・抽象的に考察するための数理モデル}}$

- 機械モデル（Turing 機械）
- **関数モデル**
- 論理モデル
- 書き換えモデル
- 代数モデル
- オートマトン

関数モデル

$\boxed{\text{関数を用いて計算の概念を正確に捉えるモデル}}$

- **帰納的関数の理論**
  計算可能な関数 ＝ もっとも基本的な関数 ＋ これらの関数の組み合わせ
- **ラムダ計算**
  - 計算可能な関数：ラムダ項
  - 計算：ラムダ項の簡約

λ 計算の歴史的背景

1930 年代、Church

- ラムダ計算の基礎が与えた.
- 計算の概念を明確にし，計算可能性に対する一つの答えを与えた.

1950 年代、McCarthy ラムダ計算に基づくプログラミング言語 Lisp を提案した.

1960 年代、Landin
ラムダ計算によって Algol60 の意味が与えられた。

1970 年代、Scott
表示的意味論：ラムダ計算と、ラムダ計算を介してプログラム集合的意味を与え、計算機科学の全般に大きな影響を与えている。

1970 年代の後半
関数型言語に関する研究が盛んになった

現在: 標準的な関数型言語：ML, Haskell

基本的なアイデア

- 抽象化による関数の定義:

$$\lambda x.x+1$$
$$\lambda x.(\lambda y.x+y)$$

- 関数適用:

$$(\lambda x.x+1)\ 5\ \Rightarrow\ 5+1$$

λ計算の言語

Definition 1 (λ項) $\mathcal{V}$ を可算無限個の変数の集合とする． λ項の集合 Λ を次の条件 (1)〜(3) を満たす最小の集合と定義する．

(1) $x \in \mathcal{V} \;\Rightarrow\; x \in \Lambda$ (変数)

(2) $M, N \in \Lambda \;\Rightarrow\; (MN) \in \Lambda$ (関数適用)

(3) $M \in \Lambda, x \in \mathcal{V} \;\Rightarrow\; (\lambda x.M) \in \Lambda$ (抽象化)

$\square$

```
<λ項> ::= <変数> | (<λ項> <λ項>) | (λ<変数>. <λ項>)
```

問題: 正しい λ 項はどれ？

$$(\lambda x.(\lambda y.(x\ y)))$$
$$(\lambda.(x\ y))$$
$$x$$
$$((x\ y)\ z)$$
$$(\lambda x.(x.y))$$
$$((\lambda x.(x\ x))\ (\lambda x.(x\ x)))$$
$$(x\ (\lambda y.z))$$
$$(y\ (\lambda y))$$

λ 項の略記規則

- 最外側の括弧ははずしてよい.

$$((\lambda x.(x\ x))\ (\lambda x.(x\ x))\ \Rightarrow\ (\lambda x.(x\ x))(\lambda x.(x\ x))$$

- 関数適用操作は左結合.

$$((\ldots(M_1\ M_2)\ldots)\ M_n)\ \Rightarrow\ M_1\ M_2\ \ldots\ M_n$$

- 多重抽象化.

$$(\lambda x_1.(\ldots(\lambda x_n.M)\ldots))\ \Rightarrow\ \lambda x_1\ \ldots\ x_n.M$$

問題：$(\lambda x.(\lambda y.((x\ y)\ (z\ u))))$ の略記表現は？

部分項

λ 項の構造を調べるときによく用いる概念である.

Definition 2 (部分項) λ 項の部分項を次のように帰納的に定義する.

1. $x \in \mathcal{V}$ の部分項は x である.
2. $(M\ N)$ の部分項は M の部分項, N の部分項, および $(M\ N)$ である.
3. $(\lambda x.M)$ の部分項は M の部分項と $(\lambda x.M)$ である.

$\square$

λ 項 M の部分項とは, M を構成する (M 自体を含む) λ 項のことをいう. M 自体を除いた M の部分項を M の**真部分項**という.

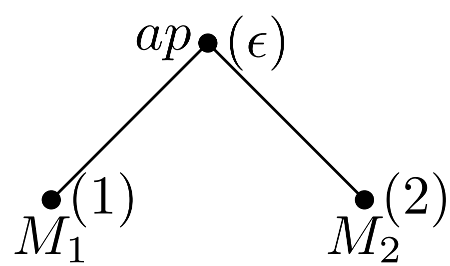
問題: $(\lambda x\ y.x)\ (\lambda y.x)\ z$ のすべての部分項を求めよ.

λ 項を表わすラベル付き木

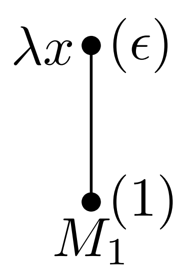
変数:

$$x \bullet (\epsilon)$$

関数適用:



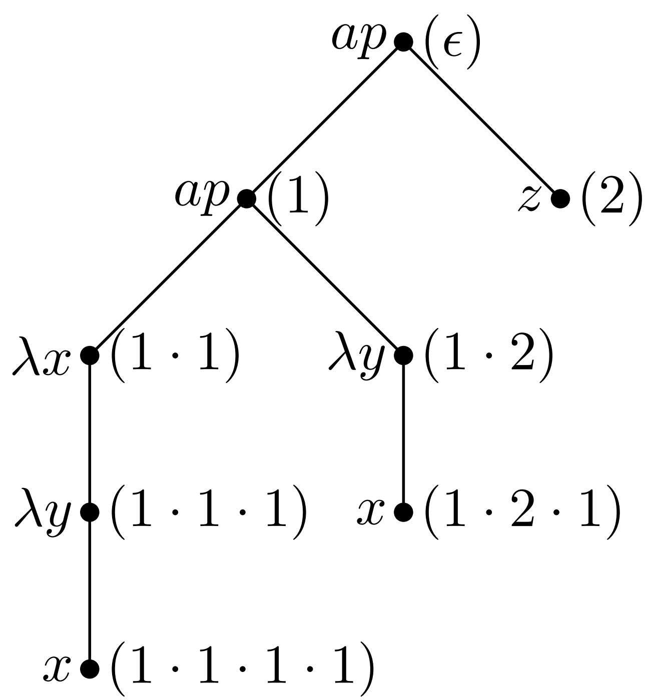
抽象化:



出現位置

ノードの emph 出現位置はルートからの数字列で表現される．例：$(1 \cdot 1 \cdot 2)$.
ルートの位置が ϵ で表現される．$\epsilon \cdot i = i \cdot \epsilon$ が成立する．

例： $(\lambda x\, y.x)\ (\lambda y.x)\ z$ **のラベル付き木**



出現位置の集合

Definition 3 (出現位置の集合) $\forall M \in \Lambda$ に対して，出現位置の集合 $\mathcal{O}(M)$ を次のように帰納的に定義する．

1. $M = x$ のとき
   $\mathcal{O}(M) = \{\epsilon\}$
2. $M = M_1\ M_2$ のとき
   $\mathcal{O}(M) = \{\epsilon\} \cup \{i \cdot u \mid u \in \mathcal{O}(M_i), i = 1, 2\}$
3. $M = \lambda x.M_1$ のとき
   $\mathcal{O}(M) = \{\epsilon\} \cup \{1 \cdot u \mid u \in \mathcal{O}(M_1)\}$

$\square$

M/u: 出現位置 $u \in \mathcal{O}(M)$ における部分項を表わす.

$$
\begin{array}{lcll}
M/\epsilon & = & M & \\
(M_1\ M_2)/i \cdot u & = & M_i/u, & \{ \text{ i=1,2 } \} \\
(\lambda x.M_1)/1 \cdot u & = & M_1/u &
\end{array}
$$

位置上の順序 $u \leq v$: v を根とする木が u を根とする木に含まれている.

$$u \leq v \ =_{def}\ \exists w,\ v = u \cdot w$$

自由変数・束縛変数

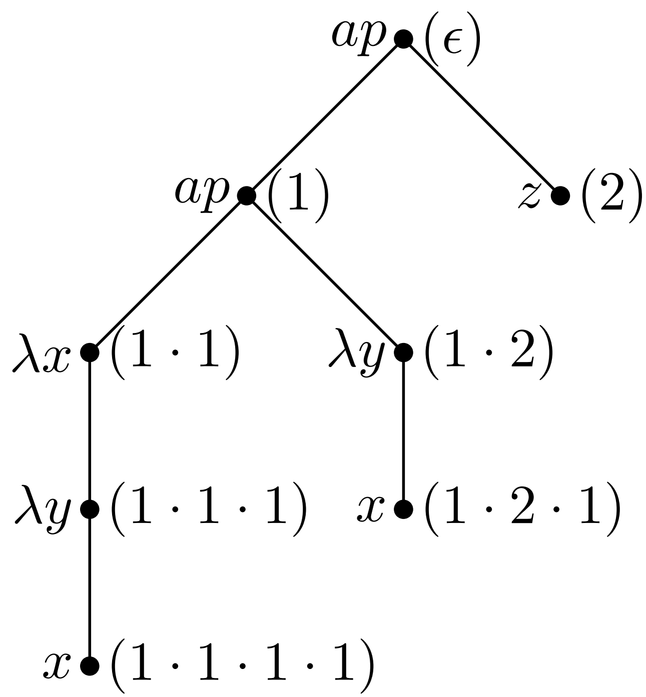
Definition 4 (自由変数) 任意の λ 項 M に含まれる変数 x，すなわち，
$\exists u \in \mathcal{O}(M), M/u = x$，次の条件を満たすとき，$M$ において自由であるという．

$$\forall v \in \mathcal{O}(M), [v < u \Rightarrow M/v \not\equiv \lambda x.(M/v \cdot 1)]$$

Definition 5 (束縛変数) 任意の λ 項 M に含まれる変数 x，すなわち，
$\exists u \in \mathcal{O}(M), M/u = x$，次の条件を満たすとき，$M$ において束縛であるという．

$$\exists v \in \mathcal{O}(M), [v < u \Rightarrow M/v \equiv \lambda x.(M/v \cdot 1)]$$

問題：$(\lambda x\, y.x)\ (\lambda y.x)\ z$ **の中のすべての自由変数と束縛変数を求めよ.**



問題*:* **λ 項** M **に含まれるすべての自由変数の集合を求める関数** $\mathcal{FV}(M)$ **を定義せよ.**

束縛変数の名前変更

$\lambda x.M$ を $f(x) = M$ となる関数を表現するものと考えられる．
$f(x) = x + 5$ も，$f(y) = y + 5$ も定義している関数 f は同じものである．
これに従えば，

$$\lambda x.M \equiv \lambda y.M'$$

ただし，M' は M の自由変数 x をすべて y で置き換えて得られた λ 項とする．

$$M' \equiv M[x := y]$$

例：

$$\begin{array}{lcl} \lambda x.x & \equiv & \lambda y.y \\ \lambda x.plus\ x\ x & \equiv & \lambda y.plus\ y\ y \end{array}$$

自由変数の代入

$M[x := N]$: M に含まれるすべての M における自由変数 x に N を代入して得られる項を表わす.

$$
\begin{array}{lcl}
y[x := N] & \equiv & \begin{cases} N & \text{if } x \equiv y \\ y & \text{otherwise} \end{cases} \\
(M_1\ M_2)[x := N] & \equiv & M_1[x := N]\ M_2[x := N] \\
(\lambda y.M_1)[x := N] & = & \lambda y.M_1[x := N] \quad \text{if } y \not\equiv x \wedge y \notin \mathcal{FV}(N) \\
 & & \qquad\qquad\qquad\quad \text{(by renaming)}
\end{array}
$$

注意：代入によって，N の自由変数が M の部分項で束縛されることがない.

問題: $(\lambda y.x\, y)[x := \lambda w.y\, w]$ **を求めよ.**

代入補題

Lemma 1 (代入補題) $\forall L, M, N \in \Lambda$ に対して，$x \not\equiv y$ かつ $x \notin \mathcal{FV}(L)$ のとき，

$$M[x := N][y := L] \;\equiv\; M[y := L][x := N[y := L]]$$

問題：λ 項の構造に関する帰納法で代入補題を証明せよ．

コンビネータ

これからの議論で重要な役割を果たす特別な λ 項.

Definition 6 (コンビネータ) 自由変数を含まない λ 項をコンビネータ (combinator) という．重要なコンビネータには以下のものがある.

$$
\begin{aligned}
\mathbf{I} &\equiv \lambda x.x \\
\mathbf{K} &\equiv \lambda x\,y.x \\
\mathbf{F} &\equiv \lambda x\,y.y \\
\mathbf{S} &\equiv \lambda x\,y\,z.x\,z\,(y\,z) \\
\mathbf{B} &\equiv \lambda x\,y\,z.x\,(y\,z) \\
\mathbf{C} &\equiv \lambda x\,y\,z.x\,z\,y \\
\mathbf{Q} &\equiv (\lambda x.x\,x)\,(\lambda x.x\,x)
\end{aligned}
$$