

# 第 6 回授業補足資料

ロジックから型付きラムダ計算へ  
(命題論理だけでハリー・ハワード同型が分かる)

岡田光弘 担当

計算論理学授業11月9日

岡田光弘「慶應義塾大学計算論理学 補足教材」

2020年11月9日



なお、基本教材には©表示を付しています。



次回第 6 回(正しくは 7 回でした) 終了までの範囲で第 2 回レポート課題を提出します。

提出締め切りまでは 2 週間くらいの間隔をとります。

レポート課題の内容は来週 7 回目終了までの基本的事項についてだけです。

一方で、これまで授業で触れた「問い」のいくつかをボーナス点対象としてレポート課題に加える予定です。

証明論 1 8 ページの二つの残りの規則については来週解説します。(これは、音声では書き違えたように言いましたが、来週扱います。)



タイプ (型) = 論理式 (定義3.2 に対応)  
ラムダ項 = 証明 (前節の $\rightarrow$ だけを使った証明に対応: ただし得  
時限の証明木を一次元にエンコードしたのがラムダ項です。)

## 4.1 単純タイプ付ラムダ計算

### 定義 4.1 (タイプ)

1.  $P, Q, P_1, P_2 \dots$  等を**基底タイプ (base types)** と呼ぶ (これは第1章では原子命題と呼んだ). 基底タイプはタイプである.
2.  $A, B$  がタイプするとき,  $A \rightarrow B$  もタイプである ( $\rightarrow$  は第1章では含意と呼んだ).

上の1, 2でタイプと分かるものだけがタイプである (前はタイプを命題と呼んだ).

### 定義 4.2 (ラムダ項 ( $\lambda$ -terms))

1. 各タイプ  $A$  に対して, 変項として  $x^A, y^A, x_1^A, x_2^A$  を用いる.
2. もし  $t$  がタイプ  $A$  のラムダ項で,  $x^B$  がタイプ  $B$  の変項なら,  $\lambda x^B t$  はタイプ  $B \rightarrow A$  のラムダ項である. このとき,  $t$  に現われる  $x^B$  は束縛されると言われる.
3. もし  $t$  がタイプ  $B \rightarrow A$  のラムダ項で  $s$  がタイプ  $B$  のラムダ項なら,  $(ts)$  はタイプ  $A$  のラムダ項である.



タイプ (型) = 論理式 (定義3.2 に対応)  
ラムダ項=証明 (前節の $\rightarrow$ だけを使った証明に対応：ただし得  
時限の証明木を一次元にエンコードしたのがラムダ項です。)

## 4.1 単純タイプ付ラムダ計算

### 定義 4.1 (タイプ)

1.  $P, Q, P_1, P_2 \dots$  等を**基底タイプ (base types)** と呼ぶ (これは第1章では原子命題と呼んだ). 基底タイプはタイプである.
2.  $A, B$  がタイプするとき,  $A \rightarrow B$  もタイプである ( $\rightarrow$  は第1章では含意と呼んだ).

上の1, 2でタイプと分かるものだけがタイプである (前はタイプを命題と呼んだ).

### 定義 4.2 (ラムダ項 ( $\lambda$ -terms))

1. 各タイプ  $A$  に対して, 変項として  $x^A, y^A, x_1^A, x_2^A$  を用いる.
2. もし  $t$  がタイプ  $A$  のラムダ項で,  $x^B$  がタイプ  $B$  の変項なら,  $\lambda x^B t$  はタイプ  $B \rightarrow A$  のラムダ項である. このとき,  $t$  に現われる  $x^B$  は束縛されると言われる.
3. もし  $t$  がタイプ  $B \rightarrow A$  のラムダ項で  $s$  がタイプ  $B$  のラムダ項なら,  $(ts)$  はタイプ  $A$  のラムダ項である.

# 型（タイプ）とは

CやJavaやその他の多くのプログラミング言語でも使われる方（タイプ）概念と同じですが、より抽象的に基本型（ベース型）を与えることにより、論理—数理的な基盤的計算モデルを与えるのが型付きラムダ計算と考え方です。新しいプログラミング言語設計においても、型付きラムダ計算モデルの型理論は参考にされることが多いです。

ラムダ計算を基盤として一般計算モデル理論は、計算の基本をソフトウェア的観点から検討する理論と言えます。

本授業でも後ほど紹介するチューリングマシンモデルは、やはり一般計算モデル研究の基盤として位置づけられますが、計算の基盤をハードウェア的思考実験モデルの観点から検討する理論と言えます。

このソフトウェア的基本モデルとハードウェア的基本モデルは計算論として同地であることが数学的に証明されています。



## 4.1 単純タイプ付ラムダ計算

### 定義 4.1 (タイプ)

1.  $P, Q, P_1, P_2 \dots$  等を**基底タイプ (base types)** と呼ぶ (これは第 1 章では原子命題と呼んだ). 基底タイプはタイプである.
2.  $A, B$  がタイプするとき,  $A \rightarrow B$  もタイプである ( $\rightarrow$  は第 1 章では含意と呼んだ).

上の 1, 2 でタイプと分かるものだけがタイプである (前はタイプを命題と呼んだ).

- 今  $\text{Nat}$  を自然数の型とします (後にこれも基底タイプとしてではなく、より複雑なタイプとして定義しますが、ここでは暫定的に基底タイプと考えていただいても結構です。)
- この時、上の定義 4.1 の 2 のタイプ (型) に対する生成文法規則により、次のような表現もタイプ (型) であることが分かります。
- 例：
  - $\text{Nat} \rightarrow \text{Nat}$       自然数の入力を受けて自然数を入力する関数型サブプログラムの型
  - $(\text{Nat} \rightarrow \text{Nat}) \rightarrow (\text{Nat} \rightarrow \text{Nat})$       そのような型の関数型サブプログラムの入力を受けて、そのような型の関数型サブプログラムを入力する関数型サブプログラムの型
  - $((\text{Nat} \rightarrow \text{Nat}) \rightarrow (\text{Nat} \rightarrow \text{Nat})) \rightarrow \text{Nat}$       上のような型の関数型サブプログラムの入力を受けて、自然数を入力する関数型サブプログラムの型



上級者へ；定義4.2のラムダ項の生成規則は、前節の証明の形成規則に対応しています。（2次元の証明木をいわば1次元表現にエンコード（又は圧縮）したものです。デコードして元の証明木表現に戻すこともできます。）最初は「ならば」だけでこれを行います。この対応関係の説明は本授業の特徴的なことの一つです。

- 以下に、ラムダ項の生成規則（定義4.2）と前節の証明形成のための推論規則との対応関係を推論規則風に紹介します。この推論規則風の表記の仕方で定義4.2を表現したものを「型推論」と呼ぶことがあります。

大事な注意；ただし、定義4.2のラムダ項の生成規則と証明形成規則との関係が分からなくても、今後の授業についてこれます。成績評価Sを目指している人には、この対応関係が変わるようにしていただきたいですが、成績評価Aまでの評価を目標としている場合は、証明とラムダ項の関係について完全に理解していなくても大丈夫です。

ほとんどのレポート課題は、定義4.2のラムダ項の定義さえ分かっている場合、証明との対応を説明しなくても評価Aまでの特典が可能です。



#### 定義 4.2 (ラムダ項 ( $\lambda$ -terms))

1. 各タイプ  $A$  に対して, 変項として  $x^A, y^A, x_1^A, x_2^A$  を用いる.
  2. もし  $t$  がタイプ  $A$  のラムダ項で,  $x^B$  がタイプ  $B$  の変項なら,  $\lambda x^B t$  はタイプ  $B \rightarrow A$  のラムダ項である. このとき,  $t$  に現われる  $x^B$  は束縛されると言われる.
  3. もし  $t$  がタイプ  $B \rightarrow A$  のラムダ項で  $s$  がタイプ  $B$  のラムダ項なら,  $(ts)$  はタイプ  $A$  のラムダ項である.
- 上の 3 は次のように「ならば」の消去規則と対応がつく。次ページは 2 が「ならば」の導入規則に対応することを示している。

$$\frac{t : B \rightarrow A \quad s : B}{(ts) : A} \rightarrow -E$$



## 定義 4.2 (ラムダ項 ( $\lambda$ -terms))

1. 各タイプ  $A$  に対して, 変項として  $x^A, y^A, x_1^A, x_2^A$  を用いる.
2. もし  $t$  がタイプ  $A$  のラムダ項で,  $x^B$  がタイプ  $B$  の変項なら,  $\lambda x^B t$  はタイプ  $B \rightarrow A$  のラムダ項である. このとき,  $t$  に現われる  $x^B$  は束縛されると言われる.
3. もし  $t$  がタイプ  $B \rightarrow A$  のラムダ項で  $s$  がタイプ  $B$  のラムダ項なら,  $(ts)$  はタイプ  $A$  のラムダ項である.

$$\frac{t : A}{\lambda x^B . t : B \rightarrow A} \rightarrow -I$$

ここで,  $t$  に自由に現れる  $x^B$  は束縛されるという.



## 定義 4.2 (ラムダ項 ( $\lambda$ -terms))

1. 各タイプ  $A$  に対して、変項として  $x^A, y^A, x_1^A, x_2^A$  を用いる.
  2. もし  $t$  がタイプ  $A$  のラムダ項で、 $x^B$  がタイプ  $B$  の変項なら、 $\lambda x^B t$  はタイプ  $B \rightarrow A$  のラムダ項である. このとき、 $t$  に現われる  $x^B$  は束縛されると言われる.
  3. もし  $t$  がタイプ  $B \rightarrow A$  のラムダ項で  $s$  がタイプ  $B$  のラムダ項なら、 $(ts)$  はタイプ  $A$  のラムダ項である.
- 上の 1 は  $A$  を前提にし、その前提自体を結論とする証明に対応する。(なお、 $x^A$  は、 $x$  を型  $A$  の変項としてつかうという型宣言することにあたります。) 節の「証明」の構成では前提が前提でなるとくきにカギカッコを付けて番号を付けたが、ラムダ項では前提が後になって前提でなくなる時のことを見越して、初めから番号を付けていることに相当します。番号は、数字ではなく、 $x, y, z, u, v$ , などの変更に表記する。
  - なお、同じラムダ項内では、一度型宣言が表記されている変項に対して、その変項がおなじ型であることが文脈上明らかな場合は、変項右肩の型表記は省略してよいこととします。



# 問い

- 次の表現が定義4.2の意味で型付きラムダ項となるように、なるべく単純な型を各変更に割り当ててください。またそのとき、全体の型が何になるかも、定義4.2のラムダ項の生成規則に沿って答えてください。どのような順序で定義4.2の規則を使ったかを示してください。
- $(xy)$
- $(\lambda zzy)$
- $(z \lambda y(yx))$



# 問いの答え 1

( $xy$ )の型付け

今  $y$  の型を例えば  $P$  とします。

定義4.2の3によってしかこの表現がラムダ項となる可能性はありません。そして3に従えば、 $x$ の型は $y$ の型を入力型として、何かを出力する関数型（「ならば」の形）であることが分かります。ですので、 $x$ の型が $P \rightarrow A$ の形をしていけば、どんなに複雑な型 $A$ であっても、全体は型付きラムダ項となり、出来上がったラムダ項全体の型は、定義4.2の3で指定していると売りに、型 $A$ となるわけです。できりだけ簡単な $A$ ととえることにして、例えば $A$ として規定型 $R$ をとる場合は、全体の型は $R$ となるわけです。このように答えてください。

上級者の人用の別な答え方としては、型推論（証明の推論の「ならば」の消去規則の形をしている）を推論規則として用いて、日本語を使うことなく、答えることでも結構です。



# 問いの続き

3つ目の

- $(z \lambda y(yx))$

についても、同様な仕方で定義4.2に従って型付きラムダ項であることを示せます。示してみてください。

$(z \lambda y(yy))$

についてはどうでしょうか、



# ラムダ項の $\beta$ 簡約：証明の正規化ステップに対応する

任意のタイプ  $B$  と任意のラムダ項  $t$  と  $s$  に対して、次のラムダ項の書き換え規則は  $\beta$ -**簡約規則**と呼ばれる。

$$(\lambda x^B t s) \triangleright t[x^B := s]$$

ただし  $s$  はタイプ  $B$  とする。

ここで  $t : A$  とすると、 $\lambda x^B t : B \rightarrow A$ 、 $(\lambda x^B t s) : A$  かつ  $t[x^B := s] : A$  となることが分かる。特に  $\beta$ -簡約規則の左項と右項は同じタイプ  $A$  をもつ。



任意のタイプ  $B$  と任意のラムダ項  $t$  と  $s$  に対して、次のラムダ項の書き換え規則は  $\beta$ -簡約規則と呼ばれる。

$$(\lambda x^B t s) \triangleright t[x^B := s] \quad \text{ただし } s \text{ はタイプ } B \text{ とする。}$$

ここで  $t : A$  とすると、 $\lambda x^B t : B \rightarrow A$ 、 $(\lambda x^B t s) : A$  かつ  $t[x^B := s] : A$  となることが分かる。特に  $\beta$ -簡約規則の左項と右項は同じタイプ  $A$  をもつ。

- 例

- $(\lambda x^{\text{Nat}} ((+3)x) 2) \triangleright ((+3)2)$

- $(\lambda x^{\text{Nat}} ((+3)x) (\lambda y^{\text{Nat}} (+y) 2) 5))$

- 

- $\triangleright (\lambda x^{\text{Nat}} ((+3)x)((+5)2) ) \triangleright ((+3)((+5)2) )$

- ベータ簡約を赤の部分に先に適用し、そのあとで青の部分の簡約をすることもできます。どのような戦略（簡約可能な箇所[リデックスと呼ぶ]の選び方）によっても計算結果が同じとなります。  
 ‘（チャーチ・ロッサ性, または合流性を持つ、と言います。）’



# 問い

- 先ほどの問いで型を付けて、型付きラムダ項だと分かった次の項について、右肩に型名を記して、 $\beta$ 簡約してください。簡約はできない既約形になっている場合は、そのことがどこから分かるか答えてください。上級者は証明の正規化との対応を示してください。基本教材の次のsubsectionにも多数の練習問題あり。

- $(xy)$
- $(\lambda zzy)$
- $(z \lambda y(yx))$

