2004 年度夏学期・計算モデルの数理

ラムダ計算 (λ-Calculus)

黒木 裕介

計数工学科数理情報工学コース 4 **年**

t30501@mail.ecc.u-tokyo.ac.jp

2004 **年** 5 **月** 9 **日**

Ver.α-20040509

概要

計算可能性を正確に捉えるモデルのひとつに上げられる，λ 計算は，計算可能な関数を定義する λ **項**と，計算を定義する**簡約**によって構成される．この講義においては，λ 項の定義のあと，わき道にそれるように見えることが続くが，すべては簡約をきちんと定義するために行われていると思うと見通しがよい．また，λ 項の定義は，意味があるかどうか（semantics）は考えず，表記・形式・構文（syntax）を味気なく定義していると思うと飲み込みが早い．

なお本書は，東京大学工学部講義「計算モデルの数理」のうち，最初の 3 回のトピック「ラムダ計算」（担当：胡 振江 助教授）の内容を基にしている．

/* しかし，講義において，定義と命題の差があいまいにされていたり暗黙の定義が多すぎたりすることに疑問を持ち，自らの疑問を解消すべく筆を取り始めたので，ほかの人から見たら回りくどいほどの説明がついている可能性もある．その一方で，草稿（Ver.α-20040506）を見てもらった非受講者の感想を聞くと，初学者が理解するためには，いま以上に何に役立つのかを示したり，懇切丁寧に説明したりする必要があるようにも感じられる．*/

1　計算可能性を正確に捉えるモデル

- 関数モデル
  - λ 計算
  - 帰納モデル
- 論理モデル
- 機械モデル
  - チューリング機械
- 代数モデル
- 書換モデル

チューリング機械と並列されていることから分かるとおり，$\mathcal{P} \neq \mathcal{NP}$ 問題を，λ 計算を用いて表現することができる．

2　歴史的背景

- Church（数学，logist）：1930 年代 →1951 論文
- McCarchy：LISP$\cdots$symbolic，人工知能
- Landin：Algol60$\cdots$ キレイ，完璧に，初めて formal で形式的（⇔engineering, ad hoc）．意味を与えた →Pascal
- Scott（CMU，米）プログラムを操作するプログラム．表示的意味論
- John Backus（1977 年 Turing Award=Computer 科学最大の賞）受賞講演で，「関数型言語でプログラムしなければならない」．→ML，Haskell

3　基本的なアイディア

C などの言語では「定義」ベース．λ 計算は「expression」として示す．

4　λ 項とは

ここからいきなり λ 項の定義を始めるが，定義 1 から例 1 までは，途中で止まることなく，一気に読むのがよい．最初は戸惑うと思うが，意味があるかどうか（semantics）は考えず，表記・形式・構文（syntax）を味気なく定義していると思うのがよい．

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

1. $x \in \mathcal{V} \Rightarrow x \in \Lambda$

2. $M, N \in \Lambda \Rightarrow (M \sqcup N) \in \Lambda$ (**関数適用**)
3. $M \in \Lambda, x \in \mathcal{V} \Rightarrow (\lambda x.M) \in \Lambda$ (**抽象化**)
4. それ以外の構成は λ 項の集合ではない. (最小性)」

注意.

- 関数適用に出てくる ␣ は，空白のことを表す．明らかな場合，明示的には示さず，$(M\ N)$ と記すことも多い.
- 抽象化における「下の点 “.”」[*1]には重要な意味があるので，他の記号とまぎれないようにきちんと記す必要がある．λ という記号がいきなり出てきて不思議に思うかもしれないが，変数でもなんでもない，ただの記号であり，そういう記法だと思うのがよい.

  メモ．そういう意味においては，定義の 1 行目は，「., λ, (,) を記号とし」から始めるのがよいか.
- 抽象化の定義において，x を**引数**，M を *definition body* という.
- λ 項の構造は，変数，関数適用，抽象化という 3 本立てであることが特徴であり，以後の
  - ほとんどの定義は，この λ 項の構造を利用した再帰的な定義であり，
  - ほとんどの証明は，この λ 項の構造を利用した帰納法を用いることとなる. 」

例 1 (λ 項). $x, y, z \in \mathcal{V}$ とする.

- $x \in \Lambda$ (変数)
- $((x\,y)\,z) \in \Lambda$ (関数適用を 2 回)
- $(x\,(\lambda y.z)) \in \Lambda$ (変数と抽象化を関数適用)
- $(\lambda x.(\lambda y.(x\,y))) \in \Lambda$ (多重抽象化)
- $((\lambda x.(x\,x))\,(\lambda x.(x\,x))) \in \Lambda$ (抽象化をしたもの同士を関数適用)
- $(\lambda x.(x.y)) \notin \Lambda$ ($(x.y)$ の部分がおかしい：. を書くときには必ず λ[変数]. の形が必要. もしくは，$(x \sqcup y)$ とすれば正しい)
- $(\lambda.(x\,y)) \notin \Lambda$ (λ と . のあいだには変数が必要)
- $(y\,(\lambda y)) \notin \Lambda$ (λ のあとには [変数].[λ項] の形が必要)

λ 項の定義を素直に書き記すと，括弧が多くなり，煩雑な表現となってしまう．そこで，略記規則を定める.

定義 2 (λ 項の略記規則).

- 最外の括弧は省略可能

  例. $((\lambda x.(x\,x))(\lambda x.(x\,x))) =: (\lambda x.(x\,x))(\lambda x.(x\,x))$

[*1] “ 下の点 ” という言い方は，『JIS Z 8201-1981 数学記号解説』（日本規格協会，1981）による.

- 関数適用操作は左結合とする

  例. $((\cdots((M_1 \sqcup M_2) \sqcup M_3) \sqcup \cdots) \sqcup M_n) =: M_1 \sqcup M_2 \sqcup M_3 \sqcup \cdots \sqcup M_n$

- 多重抽象化

  例. $(\lambda x_1.(\lambda x_2.(\cdots.(\lambda x_n.M)))) =: \lambda x_1 x_2 \cdots x_n.M$

メモ. 上の定義だけでは，$(\lambda x.(x \sqcup y))$ といった項の略記は，$\lambda x.(x \sqcup y)$ までしか進まない．しかし，[5] などを読むと，上の定義 2 ではなく，別な定義を持って略記を定義し，$\lambda x.x\,y$ の形まで略記してよいことになっている．講義中においても $\lambda x.x\,y$ の形まで略記していたが，この場合，

- $M, N \in \Lambda, x \in \mathcal{V}$; $\lambda x.(M \sqcup N) =: \lambda x.M \sqcup N$
  つまり，抽象化の結合強度よりも，関数適用の結合強度のほうが強いということ．

も定義 2 に含める必要があるのではないだろうか．本書においては，与えられた問題や表現以外の自分で書き記す部分では，$\lambda x.(x \sqcup y)$ までしか略記できないという立場を取る．

定義 3 (部分項). λ 項の**部分項**を次のように定義する．

- x の部分項は
  - x

  である．
- $(M \sqcup N)$ の部分項は
  - $(M \sqcup N)$,
  - M の部分項，および
  - N の部分項

  である．
- $(\lambda x.M)$ の部分項は
  - $(\lambda x.M)$，および
  - M の部分項

  である．

例 2 (部分項の判定). $(\lambda xy.x)\,(\lambda y.x)\,z$ の部分項を求める．

注意. 意味は考えてはならない．

- $(\lambda xy.x)\,(\lambda y.x)\,z$
- $(\lambda xy.x)\,(\lambda y.x)$
- $(\lambda xy.x)$
- $(\lambda y.x)$ $\because (\lambda xy.x) = (\lambda x.(\lambda y.x))$
- $x_{(1)}$ 「前のほうの x」
- $(\lambda y.x)$

- $x_{(2)}$「後ろのほうの x」
- z

今の例を考えても，$(\lambda y.x)$ や x が複数箇所に現れていて，区別して示したいとき，不便である．そこで，各“ 部品 ”に“ 番地 ”を振ることを考える．アイディアは，木表現（図 1）から与えられる．

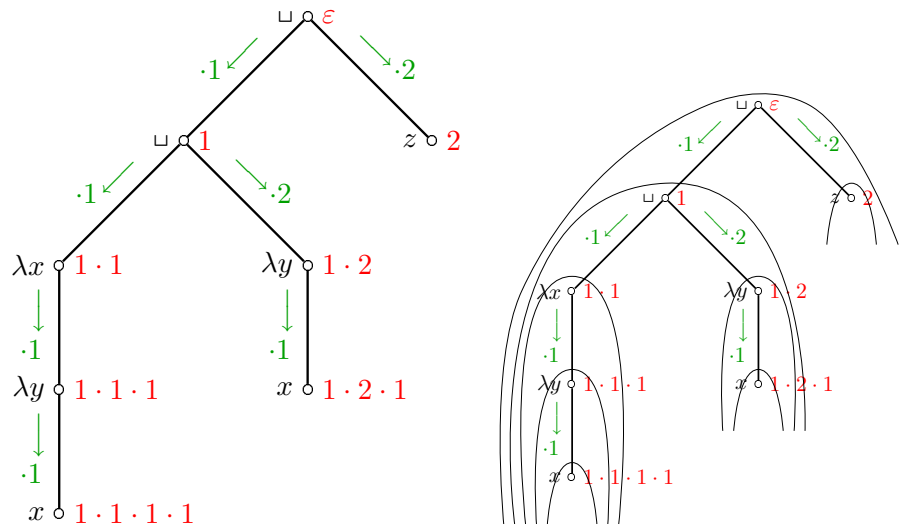


図 1　$(\lambda xy.x)\sqcup(\lambda y.x)\sqcup z$ の木表現

注意．関数適用に適当な記号がないので，図中では，$\sqcup$で示した．　」

木表現とは，与式全体を表現する root（根；ε と表記する）から始めて，

- 部分項に分解するたびに，左の（真に小さな部分）項を左に，右の項を右にという具合に 2 分の枝を張る．　(関数適用)
- 真に小さな部分項がひとつしかないときは，1 本だけ枝を張ることとする．　(抽象化)
- 分解するものがなくなったとき，そこを葉とし，枝を張らない．　(変数)

という具合に構成したものをいう．具体的には，図 1(a) のようにできあがる．このとき，図 1(b) のように，下から“ 覆いを被せていく ”と，部分項をもれなく数えることができる．

“ 番地 ”のつけ方は，左に枝を張ったときは，その子供に，自分の“ 番地 ”の後ろに $\cdot 1$ をつけ，逆に右に枝を張ったときは，その子供に，自分の“ 番地 ”の後ろに $\cdot 2$ をつけて表す．枝が 1 本しかないときは，その子供に，自分の“ 番地 ”の後ろに $\cdot 1$ をつけて表すこととする．ただし，$\varepsilon\cdot 1 =: 1, \varepsilon\cdot 2 =: 2$ と定義する．

別な言い方をすると，枝に $\{1,2\}$ が定まっていて，その番号の列として“ 番地 ”がついているともいえる．

このこと（**位置**）を改めて定義する． 部分項の定義（定義 3）に沿った形で示される．

定義 4 (項の出現位置).

位置の表現

ε もしくは，$\{1,2\}$ からなるリストとして表現する．

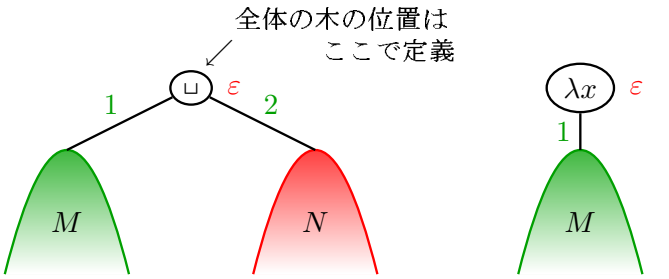
例. ε, 1, $1\cdot 1$, $1\cdot 2\cdot 1$, ...

$\varepsilon\cdot u = u\cdot\varepsilon = u$ (ε は $\cdot$ の単位元である)
$(u\cdot v)\cdot w = u\cdot(v\cdot w)$
$\exists w\in\{1,2\} : u\cdot w = v \overset{\text{def}}{\Longleftrightarrow} u \preccurlyeq v$．このことを「$u$ が v の親である」という．

部分項の出現位置の集合 $\mathcal{O}(\cdot)$

ある λ 項が与えられたときに，その部分項の出現位置 $\mathcal{O}(\cdot)$ は，以下のように帰納的に与えられる．

- $\mathcal{O}(x) = \{\varepsilon\}$
- $\mathcal{O}(M\sqcup N) = \{\varepsilon\}\cup\{1\cdot u \mid u\in\mathcal{O}(M)\}\cup\{2\cdot u \mid u\in\mathcal{O}(N)\}$
- $\mathcal{O}(\lambda x.M) = \{\varepsilon\}\cup\{1\cdot u \mid u\in\mathcal{O}(M)\}$



(a) $\mathcal{O}(M\sqcup N)$　　(b) $\mathcal{O}(\lambda x.M)$

図 2　木表現による部分項の理解

部分木を取り出す演算 M/u

λ 項 M の中から，ある位置 u のところの部分木を取り出す演算 M/u を以下のように定める．

- $M/\varepsilon = M$
- $(M\sqcup N)/1\cdot u = M/u$
- $(M\sqcup N)/2\cdot u = N/u$
- $(\lambda x.M)/1\cdot u = M/u$

注意．本書においては，「ある位置 u のところの（に）」を略して，「@u（に）」と書くことがある．

λ 計算を理解するうえで，重要な点は，変数を組み合わせて λ 項をつくっているという認識である．いままでは，意味を無視してきたが，ここで少しだけ説明しよう．

数学の世界で，関数というと，$f(x) = \text{const.}$, $f(x) = x$, $f(x) = x^2$, $\ldots$, $f(x, y) = x + y$, $f(x, y, z) = xy^2z$, $\ldots$, $f(x) = \sin x$, $f(x, y) = y \cos x$, $f(x) = \mathrm{e}^x$, $f(x) = \log x$, $\ldots$ などいろいろなものが考えられるが[*2]，実用的な場面においては，これらの関数の独立変数の部分に，具体的な数値を入れて（代入して）その値を利用したいことが多い．たとえば，

$$f(x, y) = x^2 + 2y + 1$$

が与えられたときに，x に 1 を，y に 2 を，代入するとは，

$$f(1, 2) = \left[x^2 + 2y + 1\right]_{\substack{x=1\\y=2}} = \left[1^2 + 2y + 1\right]_{y=2} = \left[2y + 2\right]_{y=2} = 2 \cdot 2 + 2 = 6$$

といった具合に行われる（いまの例では x から代入を始めたが，y から代入を始めても良いはずであることに注意しておく）．λ 計算においても，同様なことが可能で[*3]，たとえば，

$$(((\lambda xy.(x \sqcup y)) \sqcup (\lambda z.z + 1)) \sqcup 3)$$

といった λ 項が与えられると，

$$\begin{aligned}
&(((\lambda x.(\lambda y.(x \sqcup y))) \sqcup \underbrace{(\lambda z.z + 1)}_{x}) \sqcup 3) \\
&\to ((\lambda y.(\underbrace{(\lambda z.z + 1)}_{x} \sqcup y)) \sqcup \underbrace{3}_{y}) \to (\underbrace{(\lambda z.z + 1)}_{x} \sqcup \underbrace{3}_{y}) \\
&\to ((\lambda z.z + 1) \sqcup \underbrace{3}_{z}) \\
&\to \underbrace{3}_{z} + 1 \\
&\to 4.
\end{aligned}$$

といった具合にして，計算できるということがいいたいのである（先の例でどんな順番で代入するかに言及したが，λ 項の演算に関しても同様の点に注意する必要があることはあとに述べる）．

また，先に挙げた関数でいえば，$f(x, y) := x^2 + 2y + 1$ という関数は，独立変数を $x \to w$ として，$f(w, y) = w^2 + 2y + 1$ とすることでは f の性質は変わらない．しかし，$x \to y$ とすると，$f(x, y)$ という 2 変数関数が $f(y) = (y + 1)^2$ という 1 変数関数となって，関数の性質が変わってしまう．他の例で言えば，積分 $\int_x g(x)\,\mathrm{d}x$ は，独立変数が x である必要はなく，w に交換して，$\int_w g(w)\,\mathrm{d}x$ としても，性質は変わらない．

代入や変数変換についてきちんと考えなければ，計算をモデル化することはできないと言っても過言ではなかろう．つまりは，λ 計算を定義するにあたっても，きちんとしなければならない．

そこでまず，**変数**についてさらに細かく見ることにする．

[*2] f と $f(x)$ は違うということを強調することもあるが，ここでは，引数をもって定義するような関数を考えることにする．

[*3] **代入**という操作を定義していないのに「同様なことが可能だ」というのは苦しいが，先取りして説明していると思ってほしい．p.11 あたりで定義されて初めて正確なことが言えるようになる．ここでは，$f(x) = M$ を λ 計算の表現において，$\lambda x.M$ と書いていると思えばよく，$(\lambda x.M) \sqcup N$ が $f(x)|_{x=N}$ を表していると思っていればよい．

5　変数

定義 5 (自由変数，free variable). λ 項 M，$@u$ に出現した変数 $x = M/u\,;\, u \in \mathcal{O}(M)$ が M における**自由変数**である（M において**自由**である）$\overset{\text{def}}{\iff}$

$$\forall v \in \mathcal{O}(M) : v \prec u \Rightarrow M/v \neq \lambda x.(M/v \cdot 1)$$

この定義が意味するところは，u の任意の親 $v \in \mathcal{O}(M)$ に対して，$@v$ が (λx) であってはならないということである．$@v$ が (λx) のとき，v 直下の部分木の位置は $v \cdot 1$ と表現されることに注意する．

定義 6 (自由変数の集合). $M \in \Lambda$ におけるすべての自由変数の集合を，$\mathcal{FV}(M)$ とかく．

命題 1 (自由変数の集合).

- $\mathcal{FV}(x) = \{x\}$
- $\mathcal{FV}(M \sqcup N) = \mathcal{FV}(M) \cup \mathcal{FV}(N)$
- $\mathcal{FV}(\lambda x.M) = \mathcal{FV}(M) \setminus \{x\}$

定義 7 (束縛変数，bound variable). λ 項 M，$@u$ に出現した変数 $x = M/u\,;\, u \in \mathcal{O}(M)$ が M における**束縛変数**である（M において**束縛**である）$\overset{\text{def}}{\iff}$

$$\exists v \in \mathcal{O}(M) : v \prec u \Rightarrow M/v = \lambda x.(M/v \cdot 1)$$

例 3. $\lambda xy.(xyz)$ の木表現と自由・束縛

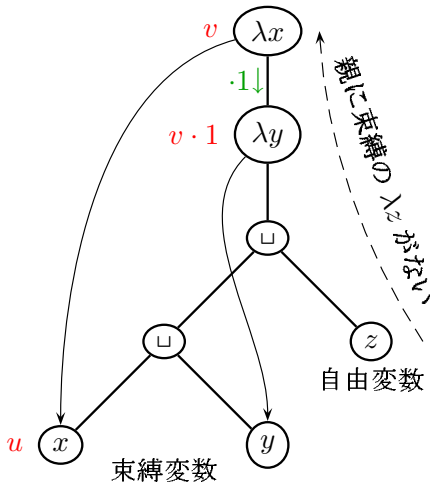


図 3　自由変数と束縛変数

例 4 (自由変数と束縛変数の判定). 例 2 において，

- x @1 · 1 · 1 · 1 は束縛変数
- x @1 · 2 · 1 は自由変数
- z @2 は自由変数

プログラムを書くにあたっても，scoping（範囲）や local 性は重要な注目点であり，束縛変数と自由変数の取扱いによってすべてを表現できるので，これからは，変数の取扱いについて見ていくことにする．

6 束縛変数の性質と演算

定義 8 (α 変換，renaming). 束縛変数の記号を，部分項に含まれない記号によって置き換えること．ある項 M を α 変換して N にすることを，$M \to_\alpha N$ と表す．変数の変換 (**例.** $x \to y$) を明示したいときは $M \xrightarrow{x \to y}_\alpha M_{y|x} = N$ とする．$M_{y|x}$ は，M のなかの変数 x をすべて変数 y に置き換えることを表す記号として定める．

注意．例 2 において，

- $\begin{cases} \lambda x \ @1 \cdot 1 \\ \quad x \ @1 \cdot 1 \cdot 1 \cdot 1 \end{cases} \to \begin{cases} \lambda z \ @1 \cdot 1 \\ \quad z \ @1 \cdot 1 \cdot 1 \cdot 1 \end{cases}$ は認められるが，
- $\begin{cases} \lambda x \ @1 \cdot 1 \\ \quad x \ @1 \cdot 1 \cdot 1 \cdot 1 \end{cases} \to \begin{cases} \lambda y \ @1 \cdot 1 \\ \quad y \ @1 \cdot 1 \cdot 1 \cdot 1 \end{cases}$ は認められない．なぜなら，変換後の変数 y が，λy @1·1 と λy @1·1·1 の両方から束縛されてしまい，性質が保存されないからである．

準備 (β 変換のための準備).

定義 9 (自由変数の代入，substitution). λ 項 M に含まれるすべての自由変数 x に，（表現すなわち λ 項）N を**代入**して得られる新たな λ 項を $M[x := N]$ と表し，以下で定義する．

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ないように定義したい．

- $y[x := N] \stackrel{\text{def}}{=\!=} \begin{cases} N & \text{if } y = x \\ y & \text{otherwise} \end{cases}$
- $(M_1 \sqcup M_2)[x := N] \stackrel{\text{def}}{=\!=} M_1[x := N] \sqcup M_2[x := N]$

  注意．代入の binding> 関数適用の binding とする．
- $(\lambda y.M)[x := N] \stackrel{\text{def}}{=\!=} \lambda y.M[x := N] \quad \text{if } y \neq x \text{ and } y \notin \mathcal{FV}(N)$

  注意．もしも，$y \in \mathcal{FV}(N)$ のときには，$(\lambda y.M)$ の方を先に α 変換する．つまり，$w \notin \mathcal{FV}(M)$ and $w \notin \mathcal{FV}(N)$ なる変数 w を新たに設定して，$(\lambda y.M) \xrightarrow{y \to w}_\alpha (\lambda w.M_{w|y})$ としてから，代入 $(\lambda w.M_{w|y})[x := N] = \lambda w.M_{w|y}[x := N]$ する．

例 5 (抽象化と代入). $M = \lambda y.(x\,y)$, $N = \lambda w.(y\,w)$ のとき，$M[x := N]$ を計算する．

$$\begin{aligned} M[x := N] &= (\lambda y.(x\,y))[x := \lambda w.(y\,w)] \\ &\xrightarrow{y \to z}_{\alpha} (\lambda z.(x\,z))[x := \lambda w.(y\,w)] \\ &= \lambda z.(\lambda w.(y\,w)\,z). \end{aligned}$$

補題 2 (代入補題). $x \neq y$ *and* $x \notin \mathcal{FV}(L)$ のとき，

$$\big(M[x := N]\big)[y := L] \stackrel{\text{def}}{=\!=} M[x := N][y := L] = M[y := L][x := N[y := L]]$$

Proof. λ 項の構造に関する帰納法で行う．
(I) $M = z$(変数) のとき：

$$[\text{左辺}] = z[x := N][y := L] = \begin{cases} N[y := L] & z = x \\ z[y := L] & \text{otherwise} \end{cases} = \begin{cases} L & z = y \\ z & z \neq y \end{cases}$$

[右辺] については，

- if $z = x$ $(\because z \neq y)$;
  $[\text{右辺}] = z[y := L][x := N[y := L]] = z[x := N[y := L]] = N[y := L]$.
- if $z \neq x$, $z = y$;
  $[\text{右辺}] = z[y := L][x := N[y := L]] = L[x := N[y := L]] = L \quad \because x \notin \mathcal{FV}(L)$.
  注意. $x \notin \mathcal{FV}(L) \Rightarrow L[x := N] = L$ を認める．
- if $z \neq x$, $z \neq y$;
  $[\text{右辺}] = z[y := L][x := N[y := L]] = z[x := N[y := L]] = z$

となるので，$M = z$(変数) のとき [左辺] = [右辺] が示せた．
(II) $M = (M_1 \sqcup M_2)$(関数適用) のとき：帰納法の仮定として，M_1, M_2 に関しては同値性が示されているとする．

$$\begin{aligned} [\text{左辺}] &= (M_1 \sqcup M_2)[x := N][y := L] \\ &\stackrel{\text{def}}{=\!=} \big((M_1 \sqcup M_2)[x := N]\big)[y := L] \\ &= \big(M_1[x := N] \sqcup M_2[x := N]\big)[y := L] \\ &= M_1[x := N][y := L] \sqcup M_2[x := N][y := L] \\ &= M_1[y := L][x := N[y := L]] \sqcup M_2[y := L][x := N[y := L]] \quad \because \text{帰納法の仮定より} \\ &= \big(M_1[y := L] \sqcup M_2[y := L]\big)[x := N[y := L]] \quad \because \text{代入の定義を逆向きに用いた} \\ &= (M_1 \sqcup M_2)[y := L][x := N[y := L]] = [\text{右辺}] \end{aligned}$$

(III) $M = (\lambda z.M')$(抽象化) のとき：帰納法の仮定として，M' に関しては同値性が示されている

とする.

$$
\begin{aligned}
[\text{左辺}] &= (\lambda z.M')[x := N][y := L] \\
&\overset{\text{def}}{=\!=} \big((\lambda z.M')[x := N]\big)[y := L] \quad ; z \neq x \text{ とする．さもなくばまったく代入できない.} \\
&\xrightarrow{z \to w}_{\alpha} \big(\lambda w.M'_{w|z}[x := N]\big)[y := L] \quad ; w \notin \mathcal{FV}(M') \text{ and } w \notin \mathcal{FV}(N) \\
&\xrightarrow{w \to v}_{\alpha} \lambda v.M'_{v|w|z}[x := N][y := L] \quad ; v \notin \mathcal{FV}(M'_{w|z}) \text{ and } v \notin \mathcal{FV}(N) \text{ and } v \notin \mathcal{FV}(L) \\
&= \lambda v.M'_{v|w|z}[y := L][x := N[y := L]] \quad \because \text{帰納法の仮定より} \\
&\xrightarrow{v \to u}_{\alpha} \big(\lambda u.M'_{u|v|w|z}[y := L]\big)[x := N[y := L]] \quad ; u \notin \mathcal{FV}(M'_{v|w|z}) \text{ and } u \notin \mathcal{FV}(L) \\
&\xrightarrow{u \to z}_{\alpha} (\lambda z.M')[y := L][x := N[y := L]].
\end{aligned}
$$

$\because z \to w, w \to v, v \to u$ の α 変換によって，z は M' から排除されて u に置き換わっていた（$M'_{u|v|w|z}$ となっていた；一般に $u \neq z$）のだから，$u \to z$ としても性質を変えない（$u = z$ であっても問題ない）.

よって，(I)(II)(III) から，代入補題は証明された. □

」

答案 1.

$$M[x := N][x := L] = M[x := N[x := L]]$$

を証明する.

λ 項の構造に関する帰納法で行う.

(I) $M = y$(変数) のとき：

$$[\text{左辺}] = y[x := N][x := L] = \big(y[x := N]\big)[x := L] = \begin{cases} N[x := L] & y = x \\ y & otherwise \end{cases}$$

[右辺] については,

- *if* $y = x$;
  [右辺] $= y[x := N[x := L]] = N[x := L]$.
- *if* $y \neq x$;
  [右辺] $= y$

となるので，$M = y$(変数) のとき [左辺] = [右辺] が示せた.

(II) $M = (M_1 \sqcup M_2)$(関数適用) のとき：帰納法の仮定として，M_1, M_2 に関しては同値性が示されているとする.

$$
\begin{aligned}
[\text{左辺}] &= (M_1 \sqcup M_2)[x := N][x := L] \\
&\overset{\text{def}}{=\!=} \big((M_1 \sqcup M_2)[x := N]\big)[x := L] \\
&= \big(M_1[x := N] \sqcup M_2[x := N]\big)[x := L] \\
&= M_1[x := N][x := L] \sqcup M_2[x := N][x := L] \\
&= M_1[x := N[x := L]] \sqcup M_2[x := N[x := L]] \quad \because \text{帰納法の仮定より} \\
&= (M_1 \sqcup M_2)[x := N[x := L]] = [\text{右辺}] \quad \because \text{代入の定義を逆向きに用いた}
\end{aligned}
$$

(III) $M = (\lambda y.M')$(抽象化) のとき：帰納法の仮定として，M' に関しては同値性が示されているとする．

$$
\begin{aligned}
[\text{左辺}] &= (\lambda y.M')[x := N][x := L] \\
&\overset{\text{def}}{=\!=} \big((\lambda y.M')[x := N]\big)[x := L] \quad ; y \neq x \text{ とする．さもなくばまったく代入できない．} \\
&\xrightarrow{y \to z}_{\alpha} \big(\lambda z.M'_{z|y}[x := N]\big)[x := L] \quad ; z \notin \mathcal{FV}(M') \ \textit{and}\ z \notin \mathcal{FV}(N) \\
&\xrightarrow{z \to w}_{\alpha} \lambda w.M'_{w|z|y}[x := N][x := L] \quad ; w \notin \mathcal{FV}(M'_{z|y}) \ \textit{and}\ w \notin \mathcal{FV}(N) \ \textit{and}\ w \notin \mathcal{FV}(L) \\
&= \lambda w.M'_{w|z|y}[x := N[x := L]] \quad \because \text{帰納法の仮定より} \\
&\xrightarrow{w \to y}_{\alpha} (\lambda y.M')[x := N[x := L]].
\end{aligned}
$$

$\because y \to z, z \to w$ の α 変換によって，y は M' から排除されて w に置き換わっていた（$M'_{w|z|y}$ となっていた；一般に $w \neq y$）のだから，$w \to y$ としても性質を変えない（$w = y$ であっても問題ない）．

よって，*(I)(II)(III)* から，証明された． □

定義 10 (β **変換，**β **簡約).** $(\lambda x.P) \sqcup Q$ に対して，$P[x := Q]$ と変換することを β **変換**といい，記号 $\to_\beta$ を用いて，

$$(\lambda x.P) \sqcup Q \to_\beta P[x := Q]$$

と書く．

例 6 (β **変換).** $(\lambda x.(x \sqcup y)) \sqcup z \to_\beta z \sqcup y$.

注意 *(*β 変換の不可逆性*).* M, N: λ 項とする．一般に，$M \to_\beta N \not\Rightarrow N \to_\beta M$.

定義 11 (0 回以上の β **変換).** 0 回以上，β 変換を繰り返すことを，$\overset{*}{\to}_\beta$ で表す．

定義 12 (β **可簡約項).** 1 回以上 β 変換を施せる λ 項を，β **可簡約項**という．

定義 13 (部分項の β **変換と全体の** β **変換).**

- $M \to_\beta N \Rightarrow L\,M \to_\beta L\,N$
- $M \to_\beta N \Rightarrow M\,L \to_\beta N\,L$

- $M \to_\beta N \Rightarrow \lambda x.M \to_\beta \lambda x.N$

と定める．標語的には，「大きな項の中で小さな部分項を β 変換したら，全体も β 変換したという」こと．

例 7 (部分項の β 変換と全体の β 変換). $(\lambda x.(x \sqcup y)) \sqcup z \sqcup w \to_\beta z \sqcup y \sqcup w$.

定義 14 (Combinator). 自由変数を含まない λ 項を Combinator と呼ぶ．特に，以下のものはよく用いられるので，特別な記号を定める．

- $\mathrm{I} \stackrel{\text{def}}{=\!=} \lambda x.x$
- $\mathrm{K} \stackrel{\text{def}}{=\!=} \lambda xy.x$ 第 2 番目の引数を無視し，第 1 番目の引数を定数として？扱う．
- $\mathrm{S} \stackrel{\text{def}}{=\!=} \lambda xyz.(x\,z\,(y\,z))$ 分配関数
- $\Omega \stackrel{\text{def}}{=\!=} (\lambda x.(x\,x))\,(\lambda x.(x\,x))$ 無限ループ

注意．K, S を用いれば，すべての計算可能なものを表せる．

例 8 (Combinator をもちいた簡約).

- $\mathrm{I}\,\mathrm{I} \to_\beta \mathrm{I}$
- $\mathrm{K}\,\mathrm{I}\,(\mathrm{I}\,\mathrm{I}) \stackrel{\text{def}}{=\!=} (\lambda xy.x)\,\underset{x}{\mathrm{I}}\,(\mathrm{I}\,\mathrm{I}) \to_\beta (\lambda y.\mathrm{I})\,\underset{y}{(\mathrm{I}\,\mathrm{I})} \to_\beta \mathrm{I}$

  注意．$\stackrel{*}{\to_\beta}$ の定義から $\mathrm{K}\,\mathrm{I}\,(\mathrm{I}\,\mathrm{I}) \stackrel{*}{\to_\beta} \mathrm{I}$ と書く．

答案 2-1. $M = (\lambda y.(\lambda x.y\,(x\,x))\,(\lambda x.y\,(x\,x)))\,a$ とする．M を β 簡約して得られる λ 項をすべて求める．

$$
\begin{aligned}
M &= (\lambda y.(\lambda x.y \sqcup (x \sqcup x)) \sqcup (\lambda x.y \sqcup (x \sqcup x))) \sqcup \underset{y}{a} \\
&\to_\beta \boxed{(\lambda x.a \sqcup (x \sqcup x)) \sqcup (\lambda x.a \sqcup (x \sqcup x))}\,; \\
&= (\lambda y.(\lambda x.y \sqcup (x \sqcup x)) \sqcup \underset{x}{(\lambda x.y \sqcup (x \sqcup x))}) \sqcup a \\
&\to_\beta (\lambda y.(y \sqcup ((\lambda x.y \sqcup (x \sqcup x)) \sqcup (\lambda x.y \sqcup (x \sqcup x))))) \sqcup \underset{y}{a} \\
&\to_\beta \boxed{a \sqcup ((\lambda x.a \sqcup (x \sqcup x)) \sqcup (\lambda x.a \sqcup (x \sqcup x)))}\,; \\
&\to_\beta (\lambda y.(y \sqcup ((\lambda x.y \sqcup (x \sqcup x)) \sqcup \underset{x}{(\lambda x.y \sqcup (x \sqcup x))}))) \sqcup a \\
&\to_\beta (\lambda y.(y \sqcup y \sqcup ((\lambda x.y \sqcup (x \sqcup x)) \sqcup (\lambda x.y \sqcup (x \sqcup x))))) \sqcup \underset{y}{a} \\
&\to_\beta \boxed{a \sqcup a \sqcup ((\lambda x.a \sqcup (x \sqcup x)) \sqcup (\lambda x.a \sqcup (x \sqcup x)))}\,;
\end{aligned}
$$

$$\begin{aligned}
&\to_\beta^* (\lambda y.(y\sqcup y\sqcup((\lambda x.y\sqcup(x\sqcup x))\sqcup\underbrace{(\lambda x.y\sqcup(x\sqcup x))}_{x})))\sqcup a\\
&\to_\beta (\lambda y.(y\sqcup y\sqcup y\sqcup((\lambda x.y\sqcup(x\sqcup x))\sqcup(\lambda x.y\sqcup(x\sqcup x)))))\sqcup\underbrace{a}_{y}\\
&\to_\beta \boxed{a\sqcup a\sqcup a\sqcup((\lambda x.a\sqcup(x\sqcup x))\sqcup(\lambda x.a\sqcup(x\sqcup x)))};\cdots
\end{aligned}$$

となり，すべてを示すことは不可能であるが，項の形が，

$$\underbrace{a_\sqcup\, a_\sqcup\cdots a_\sqcup}_{0\text{ 個以上}}((\lambda x.a_\sqcup(x_\sqcup x))\sqcup(\lambda x.a_\sqcup(x_\sqcup x)))$$

であることが分かる．

7　簡約と正規形

α 変換，β 変換という 2 つの変換を定義したが，次に気になるのは，いつまで変換を続ければ「美しい形」になるのかや，いつでも・どうやっても美しい形になるのかといったことである．同値関係を導入し，代数的に取り扱うことを目標とする．

定義 15 (β 同値関係). M, N: λ 項とする．ある関係 $\sim$ について，

- $M \to_\beta^* N \Rightarrow M \sim N$　[β 変換から導かれる関係][反射律]
  (β 変換を 0 回施す分には λ 項は変化しないので，$M \sim M$ は常に満たす．)
- $M \sim N \Rightarrow N \sim M$　[対称律]
- $M \sim N, N \sim L \Rightarrow M \sim L$　[推移律]

$\sim$ が上の 3 つの性質（つまり，同値関係の性質）をいつでも同時に満たすとき，$\sim$ のことを β **同値関係**といい，$=_\beta$ で表す．

注意．$M =_\beta N \Rightarrow M \to_\beta^* N$ は言えていないので，β 同値関係を β 変換に立ち返って調べることができず，気持ち悪いが，$M \to_\beta^* N$ or $N \to_\beta^* M \overset{\text{def}}{\Longleftrightarrow} M =_\beta N$ と定義すると，今度は推移律を示すのが難しくなり，$=_\beta$ が同値関係であることを証明するのに苦労する．

答案 2-2. 答案 *2-1* における M が，$M =_\beta a\, M$ であることを示す．

$$\forall n = \{0,1,2,\ldots\} : M \to_\beta^* \underbrace{a_\sqcup\, a_\sqcup\cdots a_\sqcup}_{n\ times}((\lambda x.a_\sqcup(x_\sqcup x))\sqcup(\lambda x.a_\sqcup(x_\sqcup x)))$$

より，

$$\begin{aligned}
\forall n = \{0,1,2,\ldots\} : M \to_\beta^* \underbrace{a_\sqcup\, a_\sqcup\cdots a_\sqcup}_{n\ times}((\lambda x.a_\sqcup(x_\sqcup x))\sqcup(\lambda x.a_\sqcup(x_\sqcup x))) =: M'\\
M \to_\beta^* \underbrace{a_\sqcup\, a_\sqcup\cdots a_\sqcup}_{(n+1)\ times}((\lambda x.a_\sqcup(x_\sqcup x))\sqcup(\lambda x.a_\sqcup(x_\sqcup x))) =: M''.
\end{aligned}$$

このとき，0 回の β 変換で $a_{\sqcup}M' \xrightarrow{*}_{\beta} M''$ と移れるので，

$$M'' =_{\beta} a_{\sqcup}M'.$$

さらに，定義 13（部分項の β 変換）から，$M \xrightarrow{*}_{\beta} M' \Rightarrow a_{\sqcup}M \xrightarrow{*}_{\beta} a_{\sqcup}M'$ より，

$$a_{\sqcup}M =_{\beta} a_{\sqcup}M'.$$

対称律から，$a_{\sqcup}M =_{\beta} a_{\sqcup}M' \Rightarrow a_{\sqcup}M' =_{\beta} a_{\sqcup}M$ なので，推移律から

$$M =_{\beta} M'', M'' =_{\beta} a_{\sqcup}M', a_{\sqcup}M' =_{\beta} a_{\sqcup}M \Longrightarrow M =_{\beta} a_{\sqcup}M. \quad \square$$

定義 16 (β 正規形). 部分項に β 可簡約項をもたない λ 項を β **正規形**という．

例 9 (β 正規形).

- $\Omega \stackrel{\text{def}}{=\!=} (\lambda x.(x\,x))\,\underbrace{(\lambda x.(x\,x))}_{x} \to_{\beta} (\lambda x.(x\,x))\,(\lambda x.(x\,x)) \to_{\beta} \cdots$ となって，何度 β 簡約しても同じ項しか現れないので，β 正規形を持たない．無限ループといわれるゆえんである．
- $\mathrm{K\,I\,}\Omega \xrightarrow{*}_{\beta} \mathrm{I}$．不思議である．

定理 3 (Church-Rosser (CR 性)). $M, N \in \lambda$ 項，

$$M =_{\beta} N \Rightarrow \exists L \in \lambda \text{ 項} : M \xrightarrow{*}_{\beta} L \ \textit{and}\ N \xrightarrow{*}_{\beta} L$$

系 3.a (正規形の一意性). $\forall M \in \lambda$ 項は，β 正規形が存在するときには，一意である．

Proof. 対偶を示す．もし, $\forall M \in \lambda$ 項は, β 正規形が複数存在して, 任意の 2 つが $L_1, L_2\,(L_1 \neq L_2)$ と表せたとする．このとき，

$$\begin{aligned} M \xrightarrow{*}_{\beta} L_1, M \xrightarrow{*}_{\beta} L_2 &\Rightarrow M =_{\beta} L_1, M =_{\beta} L_2 \quad (\beta \text{ から導かれる同値関係}) \\ &\Rightarrow L_1 =_{\beta} M, M =_{\beta} L_2 \quad (\text{対称律}) \\ &\Rightarrow L_1 =_{\beta} L_2 \quad (\text{推移律}) \end{aligned}$$

ここで，Church-Rosser 定理から，$\exists L \in \lambda$ 項 : $L_1 \xrightarrow{*}_{\beta} L$ and $L_2 \xrightarrow{*}_{\beta} L$．しかしここで，$L_1, L_2$ はともに β 可簡約項を持たないのだから，どちらの項も β 変換は 0 回しか作用させることができない．よって，$L_1 = L = L_2$ でなければならないので，矛盾する．よって正規形は，存在するときには一意に定まる． $\square$

注意．λ 項に関する等値性 $=$ は定義していないが，syntax がまったく同じ項であるときに $=$ で 2 項を結び，等値であるということにする．答案 2-2 においては，苦し紛れに β 変換を 0 回作用すればよいと書いたが，$=$ があればもっと楽であった．しかし，実は p.8 から暗黙のうちに $=$ や $\neq$ を使っていて，よろしくなかった．

定理 4 (合流性).

簡約の方法には 2 種類ある．

- **最左簡約**
- **内部簡約**

方法ではなく，概念として，「**標準簡約**」がある．これらを定義する．

定義 17 (最左簡約).

例 10 (最左簡約).

例 11 (最左簡約).

例 12 (最左簡約).

定義 18 (内部簡約).

定義 19 (標準簡約).

命題 5 (最左簡約の非「標準簡約」性).

命題 6 (内部簡約の非「標準簡約」性).

定義 20 (正規化簡約戦略).

命題 7 (最左簡約戦略と正規化簡約戦略).

答案 3. $(\lambda xy.(\lambda w.w\,w)\,x\,y)\,(\mathrm{S}\,a)\,(\mathrm{K}\,\mathrm{I})$ を最左簡約戦略で β 正規形までの簡約を示す．

まず，関数適用と括弧を明示して書き直す．

$$
\begin{aligned}
&(\lambda xy.((\lambda w.(w\sqcup w))\sqcup x\sqcup y))\sqcup(\mathrm{S}\sqcup a)\sqcup(\mathrm{K}\sqcup\mathrm{I})\\
&=(\lambda xy.((\lambda w.(w\sqcup w))\sqcup x\sqcup y))\sqcup((\lambda xyz.(x\sqcup z\,(y\sqcup z)))\sqcup a)\sqcup((\lambda xy.x)\sqcup(\lambda x.x)).
\end{aligned}
$$

β 簡約を，最左簡約戦略で行う．

$$
\begin{aligned}
&(\lambda xy.((\lambda w.(w\sqcup w))\sqcup x\sqcup y))\sqcup\underset{x}{\underline{((\lambda xyz.(x\sqcup z\sqcup(y\sqcup z)))\sqcup a)}}\sqcup((\lambda xy.x)\sqcup(\lambda x.x))\\
&\xrightarrow[\alpha]{y\to v}(\lambda xv.((\lambda w.(w\sqcup w))\sqcup x\sqcup v))\sqcup\underset{x}{\underline{((\lambda xyz.(x\sqcup z\sqcup(y\sqcup z)))\sqcup a)}}\sqcup((\lambda xy.x)\sqcup(\lambda x.x))\\
&\to_\beta(\lambda v.(\lambda w.(w\sqcup w))\sqcup((\lambda xyz.(x\sqcup z\sqcup(y\sqcup z)))\sqcup a)\sqcup v)\sqcup\underset{v}{\underline{((\lambda xy.x)\sqcup(\lambda x.x))}}\\
&\to_\beta(\lambda w.(w\sqcup w))\sqcup\underset{w}{\underline{((\lambda xyz.(x\sqcup z\sqcup(y\sqcup z)))\sqcup a)}}\sqcup((\lambda xy.x)\sqcup(\lambda x.x))\\
&\to_\beta(\lambda xyz.(x\sqcup z\sqcup(y\sqcup z)))\sqcup\underset{x}{\underline{a}}\sqcup((\lambda xyz.(x\sqcup z\sqcup(y\sqcup z)))\sqcup a)\sqcup((\lambda xy.x)\sqcup(\lambda x.x))\\
&\to_\beta(\lambda yz.(a\sqcup z\sqcup(y\sqcup z)))\sqcup\underset{y}{\underline{((\lambda xyz.(x\sqcup z\sqcup(y\sqcup z)))\sqcup a)}}\sqcup((\lambda xy.x)\sqcup(\lambda x.x))\\
&\xrightarrow[\alpha]{z\to w}(\lambda yw.(a\sqcup w\sqcup(y\sqcup w)))\sqcup\underset{y}{\underline{((\lambda xyz.(x\sqcup z\sqcup(y\sqcup z)))\sqcup a)}}\sqcup((\lambda xy.x)\sqcup(\lambda x.x))\\
&\to_\beta(\lambda w.(a\sqcup w\sqcup(((\lambda xyz.(x\sqcup z\sqcup(y\sqcup z)))\sqcup a)\sqcup w)))\sqcup\underset{w}{\underline{((\lambda xy.x)\sqcup(\lambda x.x))}}
\end{aligned}
$$

$$\to_\beta a\sqcup((\lambda xy.x)\sqcup\underbrace{(\lambda x.x)}_{x})\sqcup(((\lambda xyz.(x\sqcup z\sqcup(y\sqcup z)))\sqcup a)\sqcup((\lambda xy.x)\sqcup(\lambda x.x)))$$

$$\to_\beta a\sqcup(\lambda y.(\lambda x.x))\sqcup\underbrace{(((\lambda xyz.(x\sqcup z\sqcup(y\sqcup z)))\sqcup a)\sqcup((\lambda xy.x)\sqcup(\lambda x.x)))}_{y}$$

$$\to_\beta a\sqcup(\lambda x.x).$$

8 λ 項による計算の coding

定義 21 (Bool 計算).

- $\text{true} \stackrel{\text{def}}{=\!=} \lambda tf.t$
- $\text{false} \stackrel{\text{def}}{=\!=} \lambda tf.f$
- $\text{test} \stackrel{\text{def}}{=\!=} \lambda lmn.(l\,m\,n)$
- $\text{and} \stackrel{\text{def}}{=\!=} \lambda bc.(b\,c\,\text{false})$
- or
- not

定義 22 (組).

- $\text{pair} \stackrel{\text{def}}{=\!=} \lambda fsb.(b\,f\,s)$
- $\text{fst} \stackrel{\text{def}}{=\!=} \lambda p.(p\,\text{true})$
- $\text{snd} \stackrel{\text{def}}{=\!=} \lambda p.(p\,\text{false})$

答案 4. $\text{snd}\,(\text{pair}\,v\,w) = w$ を証明する．まず，書き下す．

$$[\text{左辺}] = (\lambda p.(p\sqcup\lambda tf.f))\sqcup((\lambda fsb.(b\sqcup f\sqcup s))\sqcup v\sqcup w).$$

最左簡約戦略で簡約してゆく．

$$\begin{aligned}
[\text{左辺}] &= (\lambda p.(p\sqcup(\lambda tf.f)))\sqcup\underbrace{((\lambda fsb.(b\sqcup f\sqcup s))\sqcup v\sqcup w)}_{p}\\
&\to_\beta (\lambda fsb.(b\sqcup f\sqcup s))\sqcup\underbrace{v}_{f}\sqcup w\sqcup(\lambda tf.f)\\
&\to_\beta (\lambda sb.(b\sqcup v\sqcup s))\sqcup\underbrace{w}_{s}\sqcup(\lambda tf.f)\\
&\to_\beta (\lambda b.(b\sqcup v\sqcup w))\sqcup\underbrace{(\lambda tf.f)}_{b}\\
&\to_\beta (\lambda tf.f)\sqcup\underbrace{v}_{t}\sqcup w\\
&\to_\beta (\lambda f.f)\sqcup\underbrace{w}_{f}\\
&\to_\beta w = [\text{右辺}].\quad \square
\end{aligned}$$

定義 23 (Church numbers).

- $0 \stackrel{\text{def}}{=\!=} \lambda sz.z$

- $1 \stackrel{\text{def}}{=\!=} \lambda sz.(s\, z)$
- $n \stackrel{\text{def}}{=\!=} \lambda sz.(s\, (s\, \cdots (s\, z) \cdots))$
- $\text{succ} \stackrel{\text{def}}{=\!=} \lambda nsz.(s\, (n\, s\, z))$
- $\text{plus} \stackrel{\text{def}}{=\!=} \lambda mnsz.(m\, s\, (n\, s\, z))$
- $\text{times} \stackrel{\text{def}}{=\!=} \lambda mn.(m\, (\text{plus}\, n)\, 0)$
- $\text{isZero} \stackrel{\text{def}}{=\!=} \lambda m.(m\, (\lambda x.\text{false})\, \text{true})$

参考文献

[1] 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).

[2] J.R. Hindley: *Basic simple type theory*, Cambridge University Press, 1997.

[3] 胡 振江：「計算モデル：ラムダ計算 (1)」(http://www.ipl.t.u-tokyo.ac.jp/lecture/computation-model2004/lambda1.pdf)

[4] 胡 振江：「計算モデル：ラムダ計算 (2)」(http://www.ipl.t.u-tokyo.ac.jp/lecture/computation-model2004/lambda2.pdf)

[5] 井田哲雄：『計算モデルの基礎理論』, 岩波講座ソフトウェア科学 12, 岩波書店, 1991, (ISBN4-00-010352-0, 3700 円), 第 4 章.

[6] 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, (ISBN 4-7649-0184-6, 3500 円).