ラムダ計算と述語論理

Ziphil Aleshlas

2017 年 3 月 17 日

1. 初めに

何らかの論理体系においてある命題 Φ が真であるする．するとしばしば，適当なラムダ計算の型付け体系が存在して，その型付け体系で Φ を何らかの型だと解釈でき，Φ を型にもつ項 M が存在することが示せる．このとき，その M は命題としての Φ の証明と対応する．このような論理とラムダ計算との対応は Curry-Howard 対応と言われ，論理学と計算機科学を結びつけるものとして非常に重要である．このノートでは述語論理を取り上げ，型付きラムダ計算との対応を証明する．

2. 型付きラムダ計算

ここでは，型付きラムダ計算の体系として λP2 と呼ばれるものを扱う．通常の単純型付きラムダ計算 $\lambda\!\rightarrow$ では初めから項と型を区別するが，ここで扱う λP2 では項と型を区別せず扱い，型付け規則によって何が項で何が型であるかを決めるという方針をとる．

定義 1. 可算集合 Var を固定し，その元を**変項** (variable) という．

定義 2. 変項以外に記号 $\star$, $\square$ を用意し，これを**ソート** (sort) という．

定義 3. 集合 Term を以下によって再帰的に定義する．

$$
\begin{aligned}
x \in \mathrm{Var} &\implies x \in \mathrm{Term} \\
s \in \{\star, \square\} &\implies s \in \mathrm{Term} \\
M, N \in \mathrm{Term} &\implies (MN) \in \mathrm{Term} \\
A, M \in \mathrm{Term} \text{ AND } x \in \mathrm{Var} &\implies (\lambda x{:}\, A.\, M) \in \mathrm{Term} \\
A, B \in \mathrm{Term} \text{ AND } x \in \mathrm{Var} &\implies (\Pi x{:}\, A.\, B) \in \mathrm{Term}
\end{aligned}
$$

このとき，Term の元を**擬項** (pseudoterm) という．

以降，α-変換 (束縛変数の変換) で移り合う擬項は同一視する．

定義 4. 変項 x と擬項 A に対し，記号 $x\colon A$ を**型宣言** (type declaration) という．

定義 5. 型宣言の列 $\Gamma = \langle x_1\colon A_1, \cdots, x_n\colon A_n\rangle$ を**型環境** (type context) という．

型環境は型宣言の列であって集合ではない．したがって，重複した要素をもっている場合，それを 1 つにまとめたものとは異なる型環境であるとする．さらに，要素の順番を入れ替えたものも異なる型環境であるとする．例えば，$\langle x\colon A, x\colon A\rangle \neq \langle x\colon A\rangle$ および $\langle x\colon A, y\colon B\rangle \neq \langle y\colon B, x\colon A\rangle$ である．ただし，便宜上しばしば集合と同じように扱うことがある．例えば，2 つの型環境 Γ, Γ' に対して，$\Gamma \cup \Gamma'$ と書いて Γ の後に Γ' の要素を順番をそのままに付け加えた型環境を表す．

定義 6. 型環境 Γ と擬項 M, A に対し，記号 $\Gamma \vdash_{\lambda P2} M\colon A$ を以下の 7 個の推論規則に従って定める．

$$\frac{}{\vdash \star\colon \Box}\ \text{Axiom}$$

$$\frac{\Gamma \vdash A\colon s}{\Gamma \cup \langle x\colon A\rangle \vdash x\colon A}\ \text{Start}$$

$$\frac{\Gamma \vdash M\colon A \quad \Gamma \vdash B\colon s}{\Gamma \cup \langle x\colon B\rangle \vdash M\colon A}\ \text{Weak}$$

$$\frac{\Gamma \vdash M\colon (\Pi x\colon A.\, B) \quad \Gamma \vdash N\colon A}{\Gamma \vdash MN\colon B[x := N]}\ \text{App}$$

$$\frac{\Gamma \cup \langle x\colon A\rangle \vdash M\colon B \quad \Gamma \vdash (\Pi x\colon A.\, B)\colon s}{\Gamma \vdash (\lambda x\colon A.\, M)\colon (\Pi x\colon A.\, B)}\ \text{Abs}$$

$$\frac{\Gamma \vdash A\colon s \quad \Gamma \cup \langle x\colon A\rangle \vdash B\colon s'}{\Gamma \vdash (\Pi x\colon A.\, B)\colon s'}\ \text{Prod}$$

$$\frac{\Gamma \vdash M\colon A \quad \Gamma \vdash A'\colon s \quad A =_\beta A'}{\Gamma \vdash M\colon A'}\ \text{Conv}$$

なお，$\Gamma \cup \langle x\colon A\rangle$ などと書かれている箇所において，x は Γ の要素に含まれていないものとする．また，s は任意のソートを表すが，規則 Prod においては，

$$(s,\ s') \in \{(\star,\ \star),\ (\Box,\ \star),\ (\star,\ \Box)\}$$

とする[*1]．

定義 7. 擬項 M, A に対し，ある型環境 Γ が存在して $\Gamma \vdash_{\lambda P2} M\colon A$ が成り立つとする．このとき，M を**項** (term) といい，A を**型** (type) という．

この型付け規則は以下に述べるように単純型の規則の拡張になっている．

[*1] 規則 Prod において $(s,\ s') = (\star,\ \Box)$ の場合を除いたものは λ2 もしくは System F と呼ばれ，Haskell に代表される多くの関数型プログラミング言語の基盤となっている．また，規則 Prod に $(s,\ s') = (\Box,\ \Box)$ の場合をさらに加えたものは λC もしくは calculus of constructions と呼ばれ，これをさらに拡張したものは Coq の型システムとして用いられている．このように，規則 Prod でどのような $(s,\ s')$ のパターンを許すかによって様々な型付け規則が定まる．

定義 8. 擬項 A, B に対し，

$$A \to B \equiv \Pi t{:}A.\, B$$

と書く．ただし，t は $t \notin \mathrm{FV}(B)$ を満たす変項とする．

このように定義すると，規則 App によって，

$$\frac{\Gamma \vdash M{:}(A \to B) \quad \Gamma \vdash N{:}A}{\Gamma \vdash MN{:}B}\ \mathrm{App}$$

が成立し，規則 Abs によって，

$$\frac{\Gamma \cup \langle x{:}A \rangle \vdash M{:}(A \to B) \quad \Gamma \vdash (A \to B){:}\star}{\Gamma \vdash (\lambda x{:}A.\, M){:}(A \to B)}\ \mathrm{Abs}$$

が成立する．さらに，$x \notin \mathrm{FV}(B)$ なる適当な変項 x を選び，規則 Prod の $(s,\, s') = (\star,\, \star)$ の場合を用いれば，

$$\frac{\Gamma \vdash A{:}\star \qquad \dfrac{\Gamma \vdash B{:}\star \quad \Gamma \vdash A{:}\star}{\Gamma \cup \langle x{:}A \rangle \vdash B{:}\star}\ \mathrm{Weak}}{\Gamma \vdash (A \to B){:}\star}\ \mathrm{Prod}$$

が成立する．以上により，$\Gamma \vdash A{:}\star$ が成り立つとき A を (単純型付きラムダ計算における) 型だと考えれば，単純型には全て $\star$ の型をつけることができ，単純型の型付け規則をそのまま行うことができる．

さて，規則 Prod について少し補足をしておく．まず，$(s,\, s') = (\star,\, \star)$ の場合は，すでに述べたように単純型を構成するのに用いられる．例えば，

$$\alpha{:}\star \vdash_{\lambda P2} (\alpha \to \alpha){:}\star \tag{2.1}$$

$$\alpha{:}\star \vdash_{\lambda P2} (\lambda x{:}\alpha.\, x){:}(\alpha \to \alpha) \tag{2.2}$$

$$\alpha{:}\star,\, \beta{:}\star,\, y{:}\beta,\, z{:}\alpha \vdash_{\lambda P2} ((\lambda x{:}\alpha.\, y)z){:}\beta$$

$$\alpha{:}\star,\, \beta{:}\star \vdash_{\lambda P2} (\lambda x{:}\alpha.\, \lambda y{:}(\alpha \to \beta).\, yx){:}(\alpha \to (\alpha \to \beta) \to \beta)$$

などが成り立つ．

$(s,\, s') = (\square,\, \star)$ の場合は，型に関する量化をするために用いられる．例えば，式 2.1 と規則 Prod を用いることで，

$$\vdash_{\lambda P2} (\Pi\alpha{:}\star.\, \alpha \to \alpha){:}\star$$

が得られる．これと式 2.2 を合わせれば，規則 Abs によって，

$$\vdash_{\lambda P2} (\lambda\alpha{:}\star.\, \lambda x{:}\alpha.\, x){:}(\Pi\alpha{:}\star.\, \alpha \to \alpha)$$

が導出できる．ここで出てくる $\lambda\alpha{:}\star.\, \lambda x{:}\alpha.\, x$ は，型 α を引数にとって α 上の恒等関数を返す項である．これは単純型付き計算の範囲では成立し得ない項である．

$(s,\ s') = (\star,\ \Box)$ の場合は，いわゆる依存型を構成するために用いられる．これにより，例えば，

$$\alpha{:}\star \vdash_{\lambda P2} (\alpha \to \star){:}\Box$$

が成り立つので，

$$\alpha{:}\star,\ p{:}(\alpha \to \star),\ x{:}\alpha \vdash_{\lambda P2} px{:}\star$$

が得られる．依存型の解釈は様々あるが，ここではこのノートのテーマである型と命題の関係に注目する．まず，$\star$ は命題と集合を同時に表す型であると考える．すると，$\alpha{:}\star$ は α が何らかの集合であることを宣言していると見なせる．さらに，$\alpha \to \star$ は α の元を受け取り命題を返す型であると考えられる．すなわち，それは α 上の述語である．したがって，$p{:}(\alpha \to \star)$ によって p は α 上の述語だと宣言されている．最後に，$x{:}\alpha$ によって x は α の元だと宣言されている．上に示した式は，これらの宣言のもとで px は 1 つの命題だということを述べている．

別の例として，

$$\alpha{:}\star,\ p{:}(\alpha \to \star),\ x{:}\alpha \vdash_{\lambda P2} (\Pi x{:}\alpha.\ px \to px){:}\star$$

を挙げておく．$\Pi x{:}\alpha$ を x に関する全称量化だと考えることにする．すると，上の式は，集合 α とその上の述語 p について，任意の α の元 x に対し px ならば px が成り立つという主張は 1 つの命題であることを述べている．この命題は真であるが，実は，

$$\alpha{:}\star,\ p{:}(\alpha \to \star),\ x{:}\alpha \vdash_{\lambda P2} (\lambda x{:}\alpha.\lambda y{:}pa.\ x){:}(\Pi x{:}\alpha.\ px \to px)$$

が成り立つので，$\Pi x{:}\alpha.\ px \to px$ を型とする項が存在する．このノートの目標は，上の例のように，型を命題だと解釈したとき，その命題が真ならばその型をもつ項が実際に存在することを示すことである．

λP2 の詳しい性質についてはここでは省略する．適宜，Barendregt[1] の 5 章もしくは Hindley[2] の 13 章を参照してほしい．

3. 多類述語論理

ここでは，述語論理に型 (ここでは類と呼ばれる) の概念を加えた多類述語論理について考える．これは，以下のように定式化される．

定義 9. 空でない有限集合 Sort を用意し，その元を**類** (sort) という．

定義 10. 有限集合 Pred を用意し，その元を**述語** (predicate) という．それぞれの述語 P に対し，記号 $A_1 \times \cdots \times A_n$ を結び付け，この記号を P の**アリティ** (arity) という．

定義 11. 有限集合 Fun を用意し，その元を**関数** (function) という．それぞれの関数 f に対し，記号 $A_1 \times \cdots \times A_n \to A$ を結びつけ，この記号を f の**アリティ** (arity) という．

定義 12. 有限集合 Con を用意し，その元を**定数** (constant) という．それぞれの定数 c に対し，類 A を 1 つ結びつける．

定義 13. 可算集合 Var を用意し，その元を**変数** (variable) という．それぞれの変数 x に対し，類 A を 1 つ結びつける．なお，各類 A に対し，A と結び付けられた変数が可算個存在するようにしておく．

定義 14. 類, 述語, 関数, 定数, 変数から成る集合の組 $\mathcal{S}$ = (Sort, Pred, Fun, Con, Var) を**多類構造** (many sorted structure) という．

なお，Pred, Fun, Con, Var の右下にアリティや類を明記することで，そのアリティや類をもつもの全体の集合を表すことにする．例えば，$\mathrm{Fun}_{A_1\times\cdots\times A_n\to A}$ はアリティ $A_1 \times \cdots \times A_n \to A$ をもつ関数全体の集合を表し，Con_A は類 A をもつ定数全体の集合を表す．

さらに，述語, 関数, 定数, 変数の右上にアリティや類を書くことで，それがそのアリティや類と結び付けられていることを示すことがある．例えば，述語 p に対して $p^{A_1\times\cdots\times A_n}$ と書くことで，p のアリティが $A_1 \times \cdots \times A_n$ であることを明示する．

定義 15. 多類構造 $\mathcal{S}$ をとる．各類 A に対し，集合 Term_A を以下によって再帰的に定義する．

$$\begin{aligned} x \in \mathrm{Var}_A &\implies x \in \mathrm{Term}_A \\ c \in \mathrm{Con}_A &\implies c \in \mathrm{Term}_A \\ f \in \mathrm{Fun}_{A_1\times\cdots\times A_n\to A} \text{ AND } T_i \in \mathrm{Term}_{A_i} &\implies (fT_1\cdots T_n) \in \mathrm{Term}_A \end{aligned}$$

このとき，Term_A の元を A の**論理項** (term) という．

定義 16. 多類構造 $\mathcal{S}$ をとる．集合 Form を以下によって再帰的に定義する．

$$\begin{aligned} &\bot \in \mathrm{Form} \\ p \in \mathrm{Pred}_{A_1\times\cdots\times A_n} \text{ AND } T_i \in \mathrm{Term}_{A_i} &\implies (pT_1\cdots T_n) \in \mathrm{Form} \\ \Phi \in \mathrm{Form} &\implies (\neg\Phi) \in \mathrm{Form} \\ \Phi, \Psi \in \mathrm{Form} &\implies (\Phi \to \Psi) \in \mathrm{Form} \\ \Phi, \Psi \in \mathrm{Form} &\implies (\Phi \wedge \Psi) \in \mathrm{Form} \\ \Phi, \Psi \in \mathrm{Form} &\implies (\Phi \vee \Psi) \in \mathrm{Form} \\ \Phi \in \mathrm{Form} \text{ AND } x \in \mathrm{Var}_A &\implies (\forall x{:}A.\, \Phi) \in \mathrm{Form} \\ \Phi \in \mathrm{Form} \text{ AND } x \in \mathrm{Var}_A &\implies (\exists x{:}A.\, \Phi) \in \mathrm{Form} \end{aligned}$$

このとき，Form の元を**論理式** (formula) という．

以降，α-変換 (束縛変数の変換) で移り合う論理式は同一視する．

多類構造 $\mathscr{H}$ を，

$$\begin{aligned}\mathrm{Sort} &= \{\mathtt{Nat}\}\\ \mathrm{Pred} &= \{\mathtt{eq}^{\mathtt{Nat}\times\mathtt{Nat}}\}\\ \mathrm{Fun} &= \{\mathtt{s}^{\mathtt{Nat}\to\mathtt{Nat}}, \mathtt{plus}^{\mathtt{Nat}\times\mathtt{Nat}\to\mathtt{Nat}}\}\\ \mathrm{Con} &= \{\mathtt{0}^{\mathtt{Nat}}\}\end{aligned}$$

によって定義すると，これによって Heyting 算術を行うことができる．この設定のもとでは，例えば，

$$\forall x\colon \mathtt{Nat}.\, \forall y\colon \mathtt{Nat}.\, (\mathtt{eq}(\mathtt{s}x)(\mathtt{s}y)) \to (\mathtt{eq}\, xy)$$
$$\forall x\colon \mathtt{Nat}.\, \mathtt{eq}(\mathtt{plus}\, x\mathtt{0})x$$
$$\forall x\colon \mathtt{Nat}.\, \forall y\colon \mathtt{Nat}.\ \mathtt{eq}(\mathtt{plus}\, x(\mathtt{s}y))(\mathtt{s}(\mathtt{plus}\, xy))$$

は正しい論理式である．これは Heyting 算術の公理の一部を表している．

定義 17. 多類構造 $\mathcal{S}$ をとる．論理式の集合 Δ と論理式 Φ に対し，Δ から Φ が導出できることを $\Delta \vdash_{\mathcal{S}} \Phi$ と書く．これを以下の 17 個の推論規則によって定義する．

$$\frac{\Phi \in \Delta}{\Delta \vdash \Phi}\ \mathrm{Start}$$
$$\frac{\Delta \vdash \Phi}{\Delta \cup \{\Psi\} \vdash \Phi}\ \mathrm{Weak}$$
$$\frac{\Delta \vdash \bot}{\Delta \vdash \Phi}\ \bot\mathrm{E}$$
$$\frac{\Delta \cup \{\Phi\} \vdash \bot}{\Delta \vdash \neg\Phi}\ \neg\mathrm{I}$$
$$\frac{\Delta \vdash \Phi \quad \Delta \vdash \neg\Phi}{\Delta \vdash \bot}\ \neg\mathrm{E}$$
$$\frac{\Delta \cup \{\Phi\} \vdash \Psi}{\Delta \vdash \Phi \to \Psi}\ \to\mathrm{I}$$
$$\frac{\Delta \vdash \Phi \to \Psi \quad \Delta \vdash \Phi}{\Delta \vdash \Psi}\ \to\mathrm{E}$$
$$\frac{\Delta \vdash \Phi \quad \Delta \vdash \Psi}{\Delta \vdash \Phi \wedge \Psi}\ \wedge\mathrm{I}$$
$$\frac{\Delta \vdash \Phi \wedge \Psi}{\Delta \vdash \Phi}\ \wedge\mathrm{E}_1$$
$$\frac{\Delta \vdash \Phi \wedge \Psi}{\Delta \vdash \Psi}\ \wedge\mathrm{E}_2$$
$$\frac{\Delta \vdash \Phi}{\Delta \vdash \Phi \vee \Psi}\ \vee\mathrm{I}_1$$
$$\frac{\Delta \vdash \Psi}{\Delta \vdash \Phi \vee \Psi}\ \vee\mathrm{I}_2$$
$$\frac{\Delta \vdash \Phi \vee \Psi \quad \Delta \cup \{\Phi\} \vdash X \quad \Delta \cup \{\Psi\} \vdash X}{\Delta \vdash X}\ \vee\mathrm{E}$$

$$\frac{\Delta \vdash \Phi}{\Delta \vdash \forall x{:}A.\,\Phi}\,\forall\mathrm{I}$$
$$\frac{\Delta \vdash \forall x{:}A.\,\Phi}{\Delta \vdash \Phi[x := T]}\,\forall\mathrm{E}$$
$$\frac{\Delta \vdash \Phi[x := T]}{\Delta \vdash \exists x{:}A.\,\Phi}\,\exists\mathrm{I}$$
$$\frac{\Delta \vdash \exists x{:}A.\,\Phi \quad \Delta \cup \{\Phi[x := y]\} \vdash \Psi}{\Delta \vdash \Psi}\,\exists\mathrm{E}$$

なお，規則 ∀I では $x \notin \mathrm{FV}(\Delta)$ とし，規則 ∃E では $y \notin \mathrm{FV}(\Delta) \cup \mathrm{FV}(\Phi) \cup \mathrm{FV}(\Psi)$ とする．また，T は任意の項を表し，$\Phi[x := T]$ において x と T に結びついた類は同じであるとする．これによって定義される論理体系を，$\mathcal{S}$ が定める**多類述語論理** (many sorted predicate logic) という．

上の定義には二重否定除去則が含まれないことに注意すること．ラムダ計算と対応するのは直観主義論理であり，古典論理とは綺麗に対応しない．

4. 述語論理とラムダ計算の関係

4.1. 論理と型の対応

初めに，多類述語論理の論理式をラムダ計算の擬項として解釈する枠組みを定める．

定義 18. ラムダ計算の擬項 Φ, Ψ に対し，

$$\begin{aligned}
\bot &\equiv \Pi\theta{:}\star.\,\theta \\
\neg\Phi &\equiv \Phi \to \bot \\
\Phi \to \Psi &\equiv \Pi t{:}\Phi.\,\Psi \\
\Phi \wedge \Psi &\equiv \Pi\theta{:}\star.\,(\Phi \to \Psi \to \theta) \to \theta \\
\Phi \vee \Psi &\equiv \Pi\theta{:}\star.\,(\Phi \to \theta) \to (\Psi \to \theta) \to \theta \\
\forall x{:}A.\,\Phi &\equiv \Pi x{:}A.\,\Phi \\
\exists x{:}A.\,\Phi &\equiv \Pi\theta{:}\star.\,(\Pi x{:}A.\,\Phi \to \theta) \to \theta
\end{aligned}$$

と定める．なお，t は $t \notin \mathrm{FV}(\Psi)$ を満たす変項とする．

この定義により，多類構造の Sort, Pred, Fun, Con, Var をラムダ計算の Var の部分集合としてとっておけば，多種述語論理の項や式は全てラムダ計算の擬項で表現される．以降，多類述語論理の論理項や論理式とラムダ計算の擬項は特に区別しない．

4.2. 標準環境

多類述語論理では，初めから使える定数が多類構造として定まっている．例えば，Heyting 算術を定める多類構造 $\mathscr{H}$ では，述語として `eq` が使え，関数として `s` や `plus` が使える．しかし，ラムダ計算

ではそのような初めから使える定数はないので，型を導く前提条件すなわち型環境として，多類構造をラムダ計算に落とし込む必要がある．

定義 19. 多類構造 $\mathcal{S}$ に対し，

$$\begin{aligned}
[\![\mathcal{S}]\!]_{\mathrm{Sort}} &= \langle A\colon \star \mid A \in \mathrm{Sort}\rangle \\
[\![\mathcal{S}]\!]_{\mathrm{Pred}} &= \langle p\colon (A_1 \to \cdots \to A_n \to \star) \mid p^{A_1\times\cdots\times A_n} \in \mathrm{Pred}\rangle \\
[\![\mathcal{S}]\!]_{\mathrm{Fun}} &= \langle f\colon (A_1 \to \cdots \to A_n \to A) \mid f^{A_1\times\cdots\times A_n\to A} \in \mathrm{Fun}\rangle \\
[\![\mathcal{S}]\!]_{\mathrm{Con}} &= \langle c\colon A \mid c^A \in \mathrm{Con}\rangle
\end{aligned}$$

とおき，

$$[\![\mathcal{S}]\!] = [\![\mathcal{S}]\!]_{\mathrm{Sort}} \cup [\![\mathcal{S}]\!]_{\mathrm{Pred}} \cup [\![\mathcal{S}]\!]_{\mathrm{Fun}} \cup [\![\mathcal{S}]\!]_{\mathrm{Con}}$$

と定義する．$[\![\mathcal{S}]\!]$ を $\mathcal{S}$ の**標準環境** (canonical context) という．

なお，$[\![\mathcal{S}]\!]_{\mathrm{Sort}}$, $[\![\mathcal{S}]\!]_{\mathrm{Pred}}$, $[\![\mathcal{S}]\!]_{\mathrm{Fun}}$, $[\![\mathcal{S}]\!]_{\mathrm{Con}}$ それぞれにおいて，その元の順序は任意にとって良い．実際，例えば $[\![\mathcal{S}]\!]_{\mathrm{Sort}}$ と $[\![\mathcal{S}]\!]'_{\mathrm{Sort}}$ を異なる順序を入れた型環境とすると，

$$[\![\mathcal{S}]\!]_{\mathrm{Sort}} \vdash_{\lambda P2} M\colon A \iff [\![\mathcal{S}]\!]'_{\mathrm{Sort}} \vdash_{\lambda P2} M\colon A$$

が成り立つから，順序を気にする必要はない．

例として，Heyting 代数に対応する多類構造 $\mathscr{H}$ については，

$$[\![\mathscr{H}]\!] = \langle \mathtt{Nat}\colon \star,\ \mathtt{eq}\colon (\mathtt{Nat} \to \mathtt{Nat} \to \star),\ \mathtt{s}\colon (\mathtt{Nat} \to \mathtt{Nat}),\ \mathtt{plus}\colon (\mathtt{Nat} \to \mathtt{Nat} \to \mathtt{Nat}),\ \mathtt{0}\colon \mathtt{Nat}\rangle$$

となる．$\star$ が命題を表す型と考えれば，この定義は妥当であろう．

次に，多類述語論理での論理式の導出の前提を型環境として解釈する．そのために，論理式 Φ それぞれに対して記号 k_Φ を 1 つ新しくとって固定し，ラムダ計算の変項に加える．型環境として $k_\Phi\colon \Phi$ を与えることで，Φ を型にもつ項を強制的に作ることができるので，Φ が真だと仮定したことになる．なお，k_Φ たちはラムダ項に含めることができるが，議論を簡単にするため，以降単にラムダ計算の変項と言った場合は，ここで固定した k_Φ たちとは異なるものであるとする．

定義 20. 多類構造 $\mathcal{S}$ における式の集合 $\Delta = \{\Phi_1, \cdots, \Phi_n\}$ に対し，

$$[\![\Delta]\!] = \langle k_{\Phi_1}\colon \Phi_1, \cdots, k_{\Phi_n}\colon \Phi_n\rangle$$

と定義する．$[\![\Delta]\!]$ を Δ の**標準環境** (canonical context) という．

$[\![\Delta]\!]$ の元の順序は問題にならないので，任意にとって良い．

さて，多類構造 $\mathcal{S}$ が定める多類述語論理において

$$\Delta \vdash_{\mathcal{S}} \Phi \qquad [4.1]$$

が証明できるならば，あるラムダ項 M によって

$$[\![\mathcal{S}]\!] \cup [\![\Delta]\!] \vdash_{\lambda P2} M : \Phi$$

が成り立つことを主張したいのであるが，Φ には $\forall$ や $\exists$ で束縛されていない変数が含まれている場合があるので，これは一般には正しくない．例えば，Heyting 代数を表す多類構造 $\mathscr{H}$ において，

$$\vdash_{\mathscr{H}} (\forall x{:}\,\mathtt{Nat}.\, \forall y{:}\,\mathtt{Nat}.\, \mathtt{eq}\, x^{\mathtt{Nat}} y^{\mathtt{Nat}}) \to \mathtt{eq}\, z^{\mathtt{Nat}} z^{\mathtt{Nat}}$$

が成り立つが，ここには自由変数として $z^{\mathtt{Nat}}$ が含まれている．一方ラムダ計算では，型付けられた擬項に含まれる自由変項は必ず型環境に含まれていなければならないので，

$$[\![\mathscr{H}]\!] \vdash_{\lambda P2} M : ((\forall x{:}\,\mathtt{Nat}.\, \forall y{:}\,\mathtt{Nat}.\, \mathtt{eq}\, x^{\mathtt{Nat}} y^{\mathtt{Nat}}) \to \mathtt{eq}\, z^{\mathtt{Nat}} z^{\mathtt{Nat}})$$

が成り立つなら $z{:}\,\mathtt{Nat}$ が $[\![\mathscr{H}]\!]$ に属しているはずだが，これは正しくない．

以上により，主張を弱めて，式 4.1 が成り立つときに，ある型環境 Γ とラムダ項 M が存在して，

$$[\![\mathcal{S}]\!] \cup \Gamma \cup [\![\Delta]\!] \vdash_{\lambda P2} M : \Phi \qquad [4.2]$$

が成り立つということを主張するのだが，Γ に何の制約も課さないとこれは自明である．実際，適当な変項 k をとって $k{:}\,\Phi$ を Γ に加えてしまえば，$M \equiv k$ として上の主張が成り立つ．Γ が必要になるのは Φ に含まれる変数が原因なのだから，Γ には多類構造の変数に関する型宣言のみを許すことにする．

定義 21. 多類構造 $\mathcal{S}$ と型環境 Γ をとる．Γ に属する任意の型宣言 $x{:}\,A$ に対して A が類であるとき，Γ を**変数環境** (variable context) という．

我々の主張は，式 4.1 が成り立つときに，ある変数環境 Γ とラムダ項 M が存在して式 4.2 が成り立つことである．

4.3. Curry-Howard 対応

まずは，多類構造の論理項と論理式が正しく型付けられることを示す．

命題 1. 多類構造 $\mathcal{S}$ をとる．類 A をもつ論理項 T に対し，変数環境 Γ が存在して，

$$[\![\mathcal{S}]\!] \cup \Gamma \vdash_{\lambda P2} T : A$$
$$\mathrm{FV}(\Gamma) \subseteq \mathrm{FV}(T)$$

が成り立つ．

命題 2. 多類構造 $\mathcal{S}$ をとる．論理式 Φ に対し，変数環境 Γ が存在して，

$$[\![\mathcal{S}]\!] \cup \Gamma \vdash_{\lambda P2} \Phi : \star$$
$$\mathrm{FV}(\Gamma) \subseteq \mathrm{FV}(\Phi)$$

が成り立つ．

T および Φ の構造に関する帰納法により，どちらも容易に証明ができる．

さて，多類述語論理における証明とラムダ計算における項が対応するわけだが，その対応する項を決定するために，論理式の導出過程を表現している擬項を注釈として付随させる．

定義 22. 多類構造 $\mathcal{S}$ をとる．式の集合 Δ と式 Φ と擬項 M に対し，記号 $M\colon \Delta \vdash_{\mathcal{S}} \Phi$ を以下の推論規則に従って定める．

$$\frac{\Phi \in \Delta}{k_\Phi\colon \Delta \vdash \Phi}\ \mathrm{Start}$$

$$\frac{M\colon \Delta \vdash \Phi}{M\colon \Delta \cup \{\Psi\} \vdash \Phi}\ \mathrm{Weak}$$

$$\frac{M\colon \Delta \vdash \bot}{M\Phi\colon \Delta \vdash \Phi}\ \bot\mathrm{E}$$

$$\frac{M\colon \Delta \cup \{\Phi\} \vdash \bot}{(\lambda k_\Phi\colon \Phi.\, M)\colon \Delta \vdash \neg\Phi}\ \neg\mathrm{I}$$

$$\frac{M\colon \Delta \vdash \Phi \quad N\colon \Delta \vdash \neg\Phi}{NM\colon \Delta \vdash \bot}\ \neg\mathrm{E}$$

$$\frac{M\colon \Delta \cup \{\Phi\} \vdash \Psi}{(\lambda k_\Phi\colon \Phi.\, M)\colon \Delta \vdash \Phi \to \Psi}\ \to\mathrm{I}$$

$$\frac{M\colon \Delta \vdash \Phi \to \Psi \quad N\colon \Delta \vdash \Phi}{MN\colon \Delta \vdash \Psi}\ \to\mathrm{E}$$

$$\frac{M\colon \Delta \vdash \Phi \quad N\colon \Delta \vdash \Psi}{(\lambda\theta\colon \star.\, \lambda u\colon (\Phi \to \Psi \to \theta).\, uMN)\colon \Delta \vdash \Phi \wedge \Psi}\ \wedge\mathrm{I}$$

$$\frac{M\colon \Delta \vdash \Phi \wedge \Psi}{(M\Phi(\lambda u\colon \Phi.\, \lambda v\colon \Psi.\, u))\colon \Delta \vdash \Phi}\ \wedge\mathrm{E}_1$$

$$\frac{M\colon \Delta \vdash \Phi \wedge \Psi}{(M\Psi(\lambda u\colon \Phi.\, \lambda v\colon \Psi.\, v))\colon \Delta \vdash \Psi}\ \wedge\mathrm{E}_2$$

$$\frac{M\colon \Delta \vdash \Phi}{(\lambda\theta\colon \star.\, \lambda u\colon (\Phi \to \theta).\, \lambda v\colon (\Psi \to \theta).\, uM)\colon \Delta \vdash \Phi \vee \Psi}\ \vee\mathrm{I}_1$$

$$\frac{M\colon \Delta \vdash \Psi}{(\lambda\theta\colon \star.\, \lambda u\colon (\Phi \to \theta).\, \lambda v\colon (\Psi \to \theta).\, vM)\colon \Delta \vdash \Phi \vee \Psi}\ \vee\mathrm{I}_2$$

$$\frac{M\colon \Delta \vdash \Phi \vee \Psi \quad N\colon \Delta \cup \{\Phi\} \vdash X \quad P\colon \Delta \cup \{\Psi\} \vdash X}{(MX(\lambda k_\Phi\colon \Phi.\, N)(\lambda k_\Psi\colon \Psi.\, P))\colon \Delta \vdash X}\ \vee\mathrm{E}$$

$$\frac{M\colon \Delta \vdash \Phi}{(\lambda x\colon A.\, M)\colon \Delta \vdash \forall x\colon A.\, \Phi}\ \forall\mathrm{I}$$

$$\frac{M\colon \Delta \vdash \forall x\colon A.\, \Phi}{MT\colon \Delta \vdash \Phi[x := T]}\ \forall\mathrm{E}$$

$$\frac{M\colon \Delta \vdash \Phi[x := T]}{(\lambda\theta\colon \star.\, \lambda u\colon (\Pi x\colon A.\, \Phi \to \theta).\, uTM)\colon \Delta \vdash \exists x\colon A.\, \Phi}\ \exists\mathrm{I}$$

$$\frac{M\colon \Delta \vdash \exists x\colon A.\, \Phi \quad N\colon \Delta \cup \{\Phi[x := y]\} \vdash \Psi}{(M\Psi(\lambda y\colon A.\, \lambda k_\Phi\colon \Phi.\, N))\colon \Delta \vdash \Psi}\ \exists\mathrm{E}$$

各規則における条件などは定義 17 と同じである．

ここで定義した M が，まさに多類述語論理の証明に対応するラムダ項なのである．

定理 3. 多類構造 $\mathcal{S}$ をとる．論理式の集合 Δ と論理式 Φ と擬項 M が

$$M\colon \Delta \vdash_{\mathcal{S}} \Phi$$

を満たすならば，ある変数環境 Γ が存在して，

$$[\![\mathcal{S}]\!] \cup \Gamma \cup [\![\Delta]\!] \vdash_{\lambda P2} M\colon \Phi$$

が成り立つ．

証明は概略のみを述べる．$M\colon \Delta \vdash_{\mathcal{S}} \Phi$ の導出に関する帰納法による．以下では 1 つの場合だけを取り扱うが，他の場合も同様にして示せる．

$M\colon \Delta \vdash_{\mathcal{S}} \Phi$ が規則 $\vee$E の帰結である場合．証明の最後のステップは，

$$\frac{M\colon \Delta \vdash \Phi \vee \Psi \quad N\colon \Delta \cup \{\Phi\} \vdash X \quad P\colon \Delta \cup \{\Psi\} \vdash X}{(MX(\lambda k_\Phi\colon \Phi.\, N)(\lambda k_\Psi\colon \Psi.\, P))\colon \Delta \vdash X}\ \vee\mathrm{E}$$

となっている．帰納法の仮定は，

$$[\![\mathcal{S}]\!] \cup \Gamma_1 \cup [\![\Delta]\!] \vdash_{\lambda P2} M\colon (\Phi \vee \Psi) \tag{4.3}$$

$$[\![\mathcal{S}]\!] \cup \Gamma_2 \cup [\![\Delta]\!] \cup \langle k_\Phi\colon \Phi\rangle \vdash_{\lambda P2} N\colon X \tag{4.4}$$

$$[\![\mathcal{S}]\!] \cup \Gamma_3 \cup [\![\Delta]\!] \cup \langle k_\Psi\colon \Psi\rangle \vdash_{\lambda P2} P\colon X \tag{4.5}$$

の 3 つであり，$\Gamma_1, \Gamma_2, \Gamma_3$ は全て変数環境である．

まず，X は論理式なので，適当な変数環境 Γ_4 によって，

$$[\![\mathcal{S}]\!] \cup \Gamma_4 \vdash_{\lambda P2} X\colon \star$$

が成り立つ．これと式 4.3 に規則 App を適用して，

$$[\![\mathcal{S}]\!] \cup \Gamma_1 \cup \Gamma_4 \cup [\![\Delta]\!] \vdash_{\lambda P2} MX\colon ((\Phi \to X) \to (\Psi \to X) \to X) \tag{4.6}$$

を得る．一方，$\Phi \to X$ も論理式だから，ある変数環境 Γ_5 が存在して

$$[\![\mathcal{S}]\!] \cup \Gamma_5 \vdash_{\lambda P2} (\Phi \to X)\colon \star$$

が成り立つから，これと式 4.4 に規則 Abs を適用すれば，

$$[\![\mathcal{S}]\!] \cup \Gamma_2 \cup \Gamma_5 \cup [\![\Delta]\!] \vdash_{\lambda P2} (\lambda k_\Phi\colon \Phi.\, N)\colon (\Phi \to X) \tag{4.7}$$

を得る．同様にして，式 4.5 から，ある変数環境 Γ_6 が存在して

$$[\![\mathcal{S}]\!] \cup \Gamma_3 \cup \Gamma_6 \cup [\![\Delta]\!] \vdash_{\lambda P2} (\lambda k_\Phi\colon \Psi.\, P)\colon (\Psi \to X) \tag{4.8}$$

が分かる．したがって，式 4.6, 4.7, 4.8 に規則 App を用いて，

$$[\![\mathcal{S}]\!] \cup \Gamma_7 \cup [\![\Delta]\!] \vdash_{\lambda P2} (MX(\lambda k_\Phi\colon \Phi.\, N)(\lambda k_\Psi\colon \Psi.\, P))\colon X$$

を得る．ここで，

$$\Gamma_7 = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4 \cup \Gamma_5 \cup \Gamma_6$$

とおいたが，これは変数環境であるから，定理の主張が示された．

4.4. 二階論理との対応

初めに定義した多類述語論理は，述語に関する量化を許さない一階論理である．しかし，型付け規則 λP2 では，式 2 が示すように型 $\star$ をもつ変項を量化することができる．さらに，

$$\vdash_{\lambda P2} (\Pi p\colon (\alpha \to \star).\, \Pi x\colon \alpha.\, px)\colon \star$$

のように型 $\alpha \to \star$ の変項を量化することもできる．$\star$ は論理式に対応し，$\alpha \to \star$ は α 上の述語に対応するのだったから，λP2 では論理式や述語に関する量化に相当することが可能なのである．実際，初めに定義した他類述語論理を二階論理に拡張しても，定理 3 と同様の主張が成立することが知られている．

A. 註

定理 3 において，$\mathrm{FV}(\Gamma) \subseteq \mathrm{FV}(\Phi) \cup \mathrm{FV}(M)$ を満たすように Γ をとれるような気がする．しかし，上の証明の M のとり方では，∧E, ∨E, ∃E の 3 つの規則において，$x \notin \mathrm{FV}(\Phi) \cup \mathrm{FV}(M)$ であっても $x \in \mathrm{FV}(\Gamma)$ であるような変項 x が現れる可能性がある．M を十分簡約すれば，そのような x を消すことができる気がするが，良い証明が思いつかない．

参考文献

[1] H. P. Barendregt (1992)「Lambda calculi with types」『Handbook of Logic in Computer Science』2:117–309

[2] J. R. Hindley, J. P. Seldin (2008)『Lambda-Calculus and Combinators: an Introduction』Cambridge University Press