Vol. 30 No. 6　　情　報　処　理　　June 1989

解　説

直観主義論理と型理論†

桜井貴文††

1. はじめに

構成的数学をプログラムの理論に適用してプログラム検証や合成を行おう，という考え方は1970年代に起こり近年に至り盛んに研究されるようになってきた．本稿では構成的論理とプログラムの理論のそれぞれの基礎的な部分を占める直観主義論理と型理論を取り上げ両者の関連について述べる．

直観主義論理は，表面的には古典論理から排中律を除いたものであり古典論理より証明できる論理式の範囲が狭くなるだけである．では狭くなったにもかかわらずどういう意義があるのかというと，排中律を除くことによって論理式を構成的に解釈できるという古典論理にはない重要な性質をもつようになるのである．一方プログラムというものは本来構成的なものであるのでプログラムがもつ型の解釈が構成的な性格をもつのはなんら不思議なことではない．本稿で扱うのはプログラムといっても型付ラムダ計算の項であってプログラムとしては単純なものではあるが，直観主義論理と型理論を適当に選ぶと直観主義論理においてある証明図がある論理式を証明しているという関係とあるプログラムがある型をもつという関係には対応を付けることができる．この対応は formulas-as-types 原理とか Curry-Howard isomorphism と呼ばれており，構成的数学とプログラムの理論を結ぶ重要な原理である．本稿の目的はこの原理を説明することにより両者の関係を説明することにある．

なお型理論についてはすでに本誌でも林による解説[8]がある．そこで扱われている型理論は本稿で述べるものとは少し違う体系であるが，型理論と関数型言語の関連や型理論の意味論など本稿では扱っていない話題も解説されているのでぜひ合わせて読んで欲しい．

† Intuitionistic Logic and Type Theory by Takafumi SAKURAI (Department of Mathematics, Tokyo Metropolitan University).

†† 都立大学数学科

2. 直観主義論理

論理体系を定義するスタイルには一般に sequent calculus と natural deduction という二つの種類がある．両者ともに一長一短があるが，本稿の目的である論理体系と型理論の対応について述べるという立場からは natural deduction のスタイルを用いたほうが分かりやすいので，本稿では natural deduction のスタイルを用いて直観主義論理の説明を行う．また扱う論理式や推論規則の種類によってもいくつか論理体系が考えられる．**2.1** でそれらを定義し，**2.2** で直観主義論理の構成的解釈について少し触れる．

2.1 直観主義論理

まず直観主義命題論理 (intuitionistic propositional logic) を定義する．

定義 2.1 直観主義命題論理の論理式 (formula) は次の三つの節によって決まる．

(**F 1**) 原子論理式 (atomic formula) は論理式である．(原子論理式はあらかじめいくつか用意されているとする．)

(**F 2**) $\bot$ は論理式である．

(**F 3**) A と B が論理式ならば，$A\wedge B$, $A\vee B$, $A\supset B$ は論理式である．

□

以降で $\neg A$ は $A\supset\bot$ の省略形とする．また厳密にいえば用意された原子論理式の集合ごとに一つ論理体系が定まるのであるが，本稿ではその集合による違いが問題になる場面はないので気にしないことにする．

直観主義命題論理における証明木とは，大ざっぱにいえば**図-1** の推論規則を組み合わせて作られる木であるが，正確には以下のように定義される．まず natural deduction における証明を一般的な形で定義する．

定義 2.2 natural deduction のスタイルの推論規則は一般に

$$\frac{\begin{matrix}[A_1]\ \ [A_n]\\ B_1\ \cdots\ B_n\end{matrix}}{C}(R)$$

$$\frac{A\quad B}{A\wedge B}(\wedge I)\qquad \frac{A\wedge B}{A}(\wedge E_0)\qquad \frac{A\wedge B}{B}(\wedge E_1)$$

$$\frac{A}{A\vee B}(\vee I_0)\qquad \frac{B}{A\vee B}(\vee I_1)\qquad \frac{A\vee B\quad \overset{[A]}{C}\quad \overset{[B]}{C}}{C}(\vee E)$$

$$\frac{\overset{[A]}{B}}{A\supset B}(\supset I)\qquad \frac{A\supset B\quad A}{B}(\supset E)$$

$$\frac{\bot}{A}(\bot E)$$

図-1　推論規則（命題）

という形をしている．(A_i, B_i, C は論理式，(R)は規則の名前である．）ただし $[A_i]$ の部分がないような i があってもよく，また推論規則の適用に条件が付くこともある．

B_i を (R) の前提，C を (R) の結論という．

論理式を証明する証明木 (proof tree) は次のように定義され，証明木の葉にある論理式のうち [ ] で囲まれていない論理式をその証明木の仮定という．

(PT 1)　A は A を証明する証明木．

(PT 2)　Π_i が B_i を証明する証明木で，(R) に条件が付いていれば Π_i はその条件を満たしているならば

$$\frac{\Pi_1'\cdots\Pi_n'}{C}(R)$$

は C を証明する証明木である．ただし，$[A_i]$ をもつ i について Π_i' は Π_i の仮定の A_i のうちいくつか（0個でもよい）を [ ] で囲んだもの（このとき [ ] で囲まれた A_i を (R) によって discharge された仮定という），そうでない i について Π_i' は Π_i．

ただし証明木ができあがった段階である論理式がどの推論規則によって discharge されたか分からなくなる場合がある．その場合は対応する論理式と推論規則に適当に番号か何かをふって区別できるようにする．

証明木 Π が論理式 A を証明しており，Π の仮定の列を Γ とする．このとき A は仮定 Γ の下で証明されるといい，$\Gamma\vdash A$ と書く．□

定義 2.3　直観主義命題論理の推論規則は図-1 のものより成る．□

図-2, 3, 4 は直観主義命題論理の証明木の例である．

$$\frac{\overset{[A]}{B\supset A}(\supset I)}{A\supset(B\supset A)}(\supset I)$$

図-2　証明木の例

$$\frac{\dfrac{\overset{[A\wedge B]}{B}(\wedge E_1)\quad \overset{[A\wedge B]}{A}(\wedge E_0)}{B\wedge A}(\wedge I)}{(A\wedge B)\supset(B\wedge A)}(\subset I)$$

図-3　証明木の例

$$\frac{\dfrac{\dfrac{\dfrac{[A\supset(B\supset C)]\quad [A]}{B\supset C}(\supset E)\quad \dfrac{[A\supset B]\quad [A]}{B}(\supset E)}{C}(\supset I)}{A\supset C}(\supset I)}{(A\supset B)\supset(A\supset C)}(\supset I)}{(A\supset(B\supset C))\supset((A\supset B)\supset(A\supset C))}(\supset I)$$

図-4　証明木の例

これらの推論規則およびこれから出てくる推論規則は2種類つまり結論で論理記号が導入される規則と前提で論理記号が消去される規則に分類することができる．前者を introduction 規則，後者を elimination 規則という．

また $\Gamma\vdash A$ という関係だけに注目して次のように $\Gamma\vdash A$ を定義することもある．

(P 1)　Γ が A を含めば $\Gamma\vdash A$.

(P 2)　推論規則 (R) があり $\Gamma_1\vdash B_1,\cdots,\Gamma_n\vdash B_n$ (ただし $[A_i]$ をもつ i について Γ_i は Γ, A_i, そうでない i について Γ_i は Γ) ならば $\Gamma\vdash C$.

(P 1)，(P 2) は証明木を作るときに使う (PT 1)，(PT 2) に対応しているので，証明木があれば上の定義に従って $\Gamma\vdash A$ を導くことができることを確かめるのは容易であろう．

次に直観主義一階述語論理 (intuitionistic first-order predicate logic, しばしば **NJ** と呼ばれる) を定義する．述語論理になると個体 (individual) という概念が登場し，それは項 (term) によって表現される．

定義 2.4　一階述語論理の項は次の (T 1) と (T 2) によって定まる．

(T 1)　個体変数は項である．（個体変数は可算無限個用意されているとする．）

(T 2)　$t_1,\cdots,t_n$ が項ならば $f^n(t_1,\cdots,t_n)$ は項である．（ただし f^n は関数記号 (function symbol) であり，各自然数 n に対していくつか用意されているとする．）

□

定義 2.5　直観主義一階述語論理の論理式は次の (F 1′)，(F 4) と定義 2.1 の (F 2)，(F 3) によって定まる．

(F 1′)　$t_1,\cdots,t_n$ が項ならば $p^n(t_1,\cdots,t_n)$ は（原子）論理式である．（ただし p^n は述語記号 (predicate symbol) であり，各自然数 n に対していくつか用意されているとする．）

(F 4)　A が論理式で x が個体変数ならば，$\forall x.A$, $\exists x.A$ は論理式である．（$\forall x$, $\exists x$ は A に自由に出現する x を束縛する．）

□

（つまり (F 1) は (F 1′) の n を 0 に制限した場合で

$$\frac{A[a/x]}{\forall x.A}(\forall I) \qquad \frac{\forall x.A}{A[t/x]}(\forall E)$$

$$\frac{A[t/x]}{\exists x.A}(\exists I) \qquad \frac{\exists x.A \quad \overset{[A[a/x]]}{C}}{C}$$

図-5 推論規則（一階）

$$\cfrac{\cfrac{\cfrac{[\exists x.F \supset \bot] \quad \cfrac{\cfrac{}{F[a/x]}}{\exists x.F}(\exists I)}{\bot}(\supset E)}{F[a/x] \supset \bot}(\supset I)}{\forall x.F \supset \bot}(\forall I)}{\neg \exists x.F \supset \forall x.\neg F}(\supset I)$$

図-6 証明木の例

$$\frac{A[Q/P]}{\forall P.A}(\forall_2 I) \qquad \frac{\forall P.A}{A[B/P]}(\forall_2 E)$$

$$\frac{A[B/P]}{\exists P.A}(\exists_2 I) \qquad \frac{\exists P.A \quad \overset{[A[Q/P]]}{C}}{C}(\exists_2 E)$$

図-7 推論規則（二階）

$$\cfrac{\cfrac{\cfrac{[P \subset P] \quad [P]}{P}(\subset E)}{[P \supset P] \quad P}(\supset E)}{P}$$
$$\frac{P}{P \supset P}(\supset I)$$
$$\frac{(P \supset P) \supset (P \supset P)}{\forall P.(P \supset P) \supset (P \supset P)}(\forall_2 I)$$

図-8 証明木の例

ある.)

自由変数，束縛変数，自由出現，束縛出現は通常どおりに定義される．以降で変数を束縛するオペレータを導入するときも同様なのでいちいち断わらない．また $A[t/x]$ という表現は A に自由に出現する x を t でおきかえたものを表す.

定義 2.6 直観主義一階述語論理の推論規則は図-1と**図-5**の規則から成る．図-5で x と a は個体変数，t は項である．ただし，以下の eigen variable condition と呼ばれる条件が付く．($\forall I$) において $A[a/x]$ を証明する証明木の仮定に a が現れてはならない．($\exists E$) において $\exists x.A$, B および B を証明する証明木の仮定のうち $A[a/x]$ 以外のものに a が現れてはならない．□

図-6 は直観主義一階述語論理の証明木の例である.

また命題論理に戻るが今度は命題変数の限量(quantification) をもつ直観主義二階命題論理 (intuitionistic second-order propositional logic) を定義する.

定義 2.7 直観主義二階命題論理の論理式は (F 2), (F 3) および次の (F 1″), (F 5) によって定まる.

(F 1″) 原子論理式および命題変数は論理式である.(原子論理式はあらかじめいくつか，命題変数は可算無限個用意されているとする.)

(F 5) A が論理式で P が命題変数ならば，$\forall P.A$, $\exists P.A$ は論理式である．($\forall P$, $\exists P$ は A に自由に出現する P を束縛する.)

□

(念のため注意しておくが(F 1″)の原子論理式は (F 1) の意味のものである.)

定義 2.8 直観主義二階命題論理の推論規則は図-1と**図-7**の規則から成る．図-7で P と Q は命題変数，B は論理式である．ただし，($\forall_2 I$) と ($\exists_2 E$) には ($\forall I$) と ($\exists E$) と同様の eigen variable condition が付く．□

図-8 は直観主義二階命題論理の証明木の例である.

順番としてはここで直観主義二階述語論理を導入するのが当然であるが，本稿では触れない．(**5.** を参照せよ.)

最後に上記の論理体系の部分体系を二つ導入する．**4.** で型理論との比較に使うためである.

定義 2.9 最小⊃命題論理．論理式は (F 1) と (F 3′) で定まり，

(F 3′) A と B が論理式ならば，$A \supset B$ は論理式である.

推論規則は図-1 の ($\supset I$) と ($\supset E$) から成る．□

このように直観主義論理の部分体系で $\bot$ と ($\bot E$) を含まないものを最小論理という.

定義 2.10 直観主義二階⊃命題論理．論理式は(F 1″), (F 3′), (F 5′) で定まり，

(F 5′) A が論理式で P が命題変数ならば，$\forall P.A$ は論理式である.

推論規則は図-1 の ($\supset I$), ($\supset E$) と図-7 の ($\forall_2 I$), ($\forall_2 E$) から成る．□

この論理は $\bot$ と ($\bot E$) を含まないが直観主義論理と呼ばれる．なぜなら $\forall P.P$ が $\bot$ の役割をするからである.

2.2 直観主義論理の構成的解釈

証明 p が論理式 A を証明する，という関係を構成的に理解しようという場合，その関係は次のように説明 (定義ではない) される．(ここでいう証明とは必ずしも **2.1** で定義した具体的な証明木を指すのではなく，もっと広い意味で使っている.)

- p が $A \wedge B$ を証明するのは，p がペア (q, r) であって q が A を証明し r が B を証明している，というときである.
- p が $A \vee B$ を証明するのは，p がペア (n, q) で

あって $n=0$ ならば q が A を証明し $n=1$ ならば q が B を証明している，というときである．

• p が $A \supset B$ を証明するのは，p が次の規則：すなわち A を証明する q が与えられたとき q を B の証明に変換する規則である，というときである．

• p が $\perp$ を証明することはない．

• p が $\forall x.A$ を証明するのは，p が次の規則：すなわち t が与えられたとき $A[t/x]$ の証明を作りだす規則である，というときである．

• p が $\exists x.A$ を証明するのは，p がペア (t,q) であって q が $A[t/x]$ を証明している，というときである．

2番目の性質より，$A \vee B$ が証明されていればその証明には A の証明か B の証明のどちらかが含まれている，つまり $A \vee B$ が証明されていると A か B のどちらかが証明されている，ということがいえる．6番目の性質より，$\exists x.A$ が証明されていればその証明から $A[t/x]$ を成立させる t を取り出すことができる，ということがいえる．これらはそれぞれ disjunction property および existence property と呼ばれ，構成的論理を特徴付ける重要な性質である．しかしこれらだけでなく3番目のコに関する性質も重要である．つまり，$A \supset B$ の証明は A の証明から B の証明への関数となっている，いいかえればコは関数空間を構成するオペレータである，という性質である．このことは4.で型理論との比較を行うとき明らかになるであろう．

古典論理の場合は無条件に排中律を認めているため上のような説明ができないが，直観主義論理の場合上のような構成的説明を与える方法としては，natural deduction スタイルの証明木の normalization や realizability interpretation などが知られている．本稿では証明木の normalization による方法について4.で触れる．

3. 型理論（その1）

この章では二つの型理論つまりプログラムに型を与えるための体系を導入する．これらは型理論のうち型付ラムダ計算と呼ばれているものである．まず有限型付ラムダ計算 (finitely typed lambda calculus, simply typed lambda calculus ともいう) を定義する．

定義 3.1 有限型付ラムダ計算の型 (type) は次の二つの節によって定まる．

(Ty 1) 基本型 (base type) は型である．(基本型はいくつか用意されているとする．)

(Ty 2) σ と γ が型なら $\sigma \to \gamma$ は型である．

□

定義 3.2 有限型付ラムダ計算の項 (term) は次の二つの節によって定まる．

(Tm 1) 型 σ の型付変数は項である．(各型 σ につき型付変数は無限個用意されているとする．また型 σ の型付変数は x^{σ} のように上付き添字で型を表すことにする．)

(Tm 2) s, t が項，x^{σ} が型付変数ならば $\lambda x^{\sigma}.t$, st は項である．(λx^{σ} は t に自由に出現する x^{σ} を束縛する．)

□

定義 3.3 型変数の列 Γ を $x_1^{\sigma_1}, \cdots, x_n^{\sigma_n}$ とする．これらの型変数が互いに異なる（型の部分は同じでもよい）とき Γ は型宣言であるという．□

定義 3.4 有限型付ラムダ計算の型推論．Γ は型宣言，t は項，σ は型のとき，Γ の下で t は型 σ をもつという関係（$\Gamma \vdash t:\sigma$ と書く）を以下の規則によって定める．

(Ass) $\dfrac{}{\Gamma \vdash x^{\sigma}:\sigma}$ (Γ が x^{σ} を含んでいるとき)

$(\to I)$ $\dfrac{\Gamma, x^{\sigma} \vdash t:\gamma}{\Gamma \vdash \lambda x^{\sigma}.t:\sigma \to \gamma}$

$(\to E)$ $\dfrac{\Gamma \vdash t:\sigma \to \gamma \quad \Gamma \vdash s:\sigma}{\Gamma \vdash ts:\gamma}$

(今度は natural deduction の場合と違って単に上の規則を組み合わせて葉の部分が (Ass) である木を作ればよい．) □

続いて多相型付ラムダ計算 (polymorphic typed lambda calculus, second-order typed lambda calculus ともいう) を定義する．

定義 3.5 多相型付ラムダ計算の型は (Ty 2) および次の (Ty 1′), (Ty 3) によって定まる．

(Ty 1′) 基本型および型変数は型である．(基本型はいくつか，型変数は無限個用意されているとする．)

(Ty 3) σ が型，α が型変数ならば，$\forall \alpha.\sigma$ は型である．($\forall \alpha$ は σ に自由に出現する α を束縛する．)

□

定義 3.6 多相型付ラムダ計算の項は (Tm 1), (Tm 2) および次の (Tm 3), (Tm 4) によって定まる．

(Tm 3) t が項，α が型変数で，t に自由に出現する型付変数の型に α が自由に出現しないならば，$\Delta\alpha.t$ は項である．($\Delta\alpha$ は t に自由に出現する α を束縛する．)

$$\dfrac{\dfrac{}{x^\tau, y^\sigma \vdash x^\tau : \sigma}}{\dfrac{x^\tau \vdash \lambda y^\sigma.\, x^\tau : \sigma \to \tau}{\vdash \lambda x^\tau.\, \lambda y^\sigma.\, x^\tau : \tau \to (\sigma \to \tau)}}$$

図-9 型推論の例

$$\dfrac{\dfrac{\dfrac{\dfrac{}{f^{\alpha\to\alpha}, x^\alpha \vdash f^{\alpha\to\alpha} : \alpha \to \alpha} \quad \dfrac{}{f^{\alpha\to\alpha}, x^\alpha \vdash x^\alpha : \alpha}}{f^{\alpha\to\alpha}, x^\alpha \vdash f^{\alpha\to\alpha} x^\alpha : \alpha}}{f^{\alpha\to\alpha}, x^\alpha \vdash f^{\alpha\to\alpha}(f^{\alpha\to\alpha} x^\alpha) : \alpha}}{\dfrac{f^{\alpha\to\alpha} \vdash \lambda x^\alpha.\, f^{\alpha\to\alpha}(f^{\alpha\to\alpha} x^\alpha) : \alpha \to \alpha}{\dfrac{\vdash \lambda f^{\alpha\to\alpha}.\, \lambda x^\alpha.\, f^{\alpha\to\alpha}(f^{\alpha\to\alpha} x^\alpha) : (\alpha\to\alpha)\to(\alpha\to\alpha)}{\vdash \Delta\alpha.\, \lambda f^{\alpha\to\alpha}.\, \lambda x^\alpha.\, f^{\alpha\to\alpha}(f^{\alpha\to\alpha} x^\alpha) : \forall\alpha.\, (\alpha\to\alpha)\to(\alpha\to\alpha)}}$$

図-10 型推論の例

(Tm 4) t が項，σ が型ならば $t\sigma$ は項である．

□

定義 3.7 多相型付ラムダ計算の型推論の規則は有限型付ラムダ計算の型推論の規則に次の推論規則を加えたものから成る．

$$(\forall_r I)\quad \frac{\Gamma \vdash t : \sigma}{\Gamma \vdash \Delta\alpha.\, t : \forall\alpha.\, \sigma}\ \left(\begin{array}{l}\alpha \text{ は } \Gamma \text{ 中の型付変数の}\\ \text{型に自由に出現しない．}\end{array}\right)$$

$$(\forall_r E)\quad \frac{\Gamma \vdash t : \forall\alpha.\, \sigma}{\Gamma \vdash t\tau : \sigma[\tau/\sigma]}$$

□

図-9は有限型付ラムダ計算の型推論，図-10は多相型付ラムダ計算の型推論の例である．

以上の定義から明らかなように，有限/多相のいずれのラムダ計算でも項 t が与えられたとき $\Gamma \vdash t : \sigma$ を満たす Γ と σ があるかどうか判定でき，かつあるときは具体的に Γ と σ を求めることができしかも σ は唯一に定まる，という性質がある．これは t が十分な情報を含んでいるから，つまり t をみれば (Ass), $(\to I)$, $(\to E)$ などのどの規則の結論になっているのかが分かるからである．一方，型付ラムダ計算には，型は上と同様に定義するが（多相型付ラムダ計算に相当する場合でも）項は型付でないラムダ計算の項（つまり定義 3.2 の変数を型付でなくしたもの）を使って定式化するという流儀もある．(文献[8]ではこの流儀による型理論について述べてある．参照されたい．) この場合，項 t が与えられても t が型をもつかどうかは必ずしも明らかではない．実際，有限型付ラムダ計算に相当する体系では型を決定できるが，多相型付ラムダ計算に相当する体系では型を決定するアルゴリズムが存在するかどうかは分かっていない．このような型付ラムダ計算を Curry 流，本稿で定義した型付ラムダ計算を Church 流ということもある．本稿では Church 流を用いたのであるが，その理由は 4. でみるように直観主義論理との対応がつけやすいからである．

ここまではプログラムの静的な面，つまり実行前の型チェックに相当する部分について述べてきたわけだが，プログラムであるからには動的な面，つまり計算規則に相当する部分もある．それがよく知られた β-簡約 (β-reduction) であり，有限型付ラムダ計算では

(β) $\quad (\lambda x^\tau.\, t)s \rhd t[s/x^\tau]$

多相型付ラムダ計算では (β) と

(β_2) $\quad (\Delta\alpha.\, t)\sigma \rhd t[\sigma/\alpha]$

によって定義される．そしてこれらの簡約に関していずれのラムダ計算においても次の定理が成り立つ．(ラムダ計算の用語を説明なしに用いるが．)

定理 3.8 $\Gamma \vdash t : \sigma$ かつ $t \rhd t'$ ならば $\Gamma \vdash t' : \sigma$. □

定理 3.9 $\Gamma \vdash t : \sigma$ ならば t は strongly normalizable，すなわちどのように簡約しても有限回で正規形になる．□

定理 3.8 の証明は明らかだが，定理 3.9 は（特に多相ラムダ計算の場合）難しい．興味のある読者は文献[5]を参照されたい．

4. Formulas-as-Types 原理：直観主義論理と型理論の対応

まず有限型付ラムダ計算と最小コ命題論理の対応について述べる．最初に有限型付ラムダ計算の基本型と最小コ命題論理の原子論理式が 1 対 1 に対応するような有限型付ラムダ計算と最小コ命題論理（つまり基本型と原子論理式の個数が同じ）を選びそれらについて以下の議論を行う．有限型付ラムダ計算の型 σ に対し，$\to$ をコに変え基本型を対応する原子論理式に変えて得られる論理式を $\hat{\sigma}$ で表し，型宣言 Γ が $x_1^{\sigma_1}, \cdots, x_n^{\sigma_n}$ であるとき $\hat{\Gamma}$ は $\hat{\sigma}_1, \cdots, \hat{\sigma}_n$ という論理式の列を表すことにする．このとき次の定理が成り立つ．

定理 4.1 有限型付ラムダ計算において $\Gamma \vdash t : \sigma$ が成り立てば，最小コ命題論理において $\hat{\sigma}$ を証明し仮定は $\hat{\Gamma}$ に含まれる証明木を作ることができる．

証明 この証明木を $\Pi(t)$ と表し $\Pi(t)$ を t の構造について帰納的に定める．(正確には $\Gamma \vdash t : \sigma$ の推論の長さについての帰納法であるが．)

(i) $t \equiv x^\tau$ のとき．型推論は

$$\frac{}{\Gamma_1, x^\tau, \Gamma_2 \vdash x^\tau : \tau}$$

となっているので

$$\Pi(x^\tau) \overset{\text{def}}{=} \hat{\tau}$$

と定義すれば定理の条件を満たす．

(ii) $t \equiv \lambda y^{\tau}.s$ のとき．型推論の最後のステップは

$$\frac{\Gamma, y^{\tau} \vdash s : \iota}{\Gamma \vdash \lambda y^{\tau}.s : \tau \to \iota}$$

となっている．帰納法の仮定により $\Pi(s)$ は $\tilde{\iota}$ を証明する証明木で仮定は $\tilde{\Gamma}, \tilde{\gamma}$ に含まれる．ここでは陽には書かないが(i)のときには $\tilde{\gamma}$ の横に変数 x^{τ} が書いてあるとする．そして $\Pi(s)'$ を $\Pi(s)$ の仮定のうち横に y^{τ} と書いてあるものを discharge して得られる木とする．このとき

$$\Pi(\lambda x^{\tau}.s) \overset{\text{def}}{=} \frac{\Pi(s)'}{\tilde{\gamma} \supset \tilde{\iota}}$$

と定義すれば $\Pi(\lambda x^{\tau}.s)$ は定理の条件を満たす．

(iii) $t \equiv rs$ のとき．型推論の最後のステップは

$$\frac{\Gamma \vdash r : \gamma \to \iota \quad \Gamma \vdash s : \gamma}{\Gamma \vdash rs : \iota}$$

となっている．帰納法の仮定により $\Pi(r)$ は $\tilde{\gamma} \supset \tilde{\iota}$ を証明する証明木で仮定は $\tilde{\Gamma}$ に含まれ，$\Pi(s)$ は $\tilde{\gamma}$ を証明する証明木で仮定は $\tilde{\Gamma}$ に含まれる．このとき

$$\Pi(rs) \overset{\text{def}}{=} \frac{\Pi(r) \quad \Pi(s)}{\tilde{\iota}}$$

と定義すれば $\Pi(rs)$ は定理の条件を満たす．□

少し説明を加えておく．($\supset I$)で discharge される仮定と λ による型付変数の束縛が対応している．すなわち ($\supset I$) で B を証明するとき局所的に使える仮定と局所化された型付変数が対応しているのである．(図-2と図-9を比較せよ．)

また逆に証明木から項を作ることもできる．(作り方は明らかであろう．図-4 の証明木から項を作ってみよ．) つまり項をそのまま証明木と思ってよいわけである．こう思うと $\Gamma \vdash t : \sigma$ を導く過程というのは，証明木がちゃんと定義に従って作られていることのチェックになっているのであり，(少しくどいがダメ押ししておくと) $\Gamma \vdash t : \sigma$ を導く過程において各 $\Gamma \vdash t : \sigma$ を $\tilde{\Gamma} \vdash \tilde{\sigma}$ に置き換えれば最小$\supset$命題論理で(P 1), (P 2) を使って $\tilde{\Gamma} \vdash \tilde{\sigma}$ を導く過程になっている．

以上のような型理論と論理体系の対応を formulas-as-types 原理 (formulas-as-types principle) とか Curry-Howard isomorphism と呼び，多相型付ラムダ計算と直観主義二階$\supset$命題論理に対しても同様の対応が存在する．基本型と原子論理式に1対1の対応が付くようにそれぞれの体系を選び，型変数と命題変数に1対1の対応を付け (どちらも可算無限個なのでこのような対応は存在する)，上と同様に型 σ に対し論理式 σ を定める．そして証明木 $\Pi(t)$ を定理 4.1 での定義に次の場合を加えて定める．

(iv) $t \equiv \Delta\alpha.s$ のとき．

$$\Pi(\Delta\alpha.s) \overset{\text{def}}{=} \frac{\Pi(s)}{\forall \tilde{\alpha}.\tilde{\gamma}}$$

ただし $\Pi(s)$ は $\tilde{\gamma}$ を証明しており，eigen variable condition を満たしていることは $(\forall, I)$ の条件より分かる．

(v) $t \equiv s\iota$ のとき．

$$\Pi(s\iota) \overset{\text{def}}{=} \frac{\Pi(s)}{\tilde{\gamma}[\tilde{\iota}/\tilde{\alpha}]}$$

ただし $\Pi(s)$ は $\forall\tilde{\alpha}.\tilde{\gamma}$ を証明している．

これにより多相型付ラムダ計算と直観主義二階$\supset$命題論理についても定理 4.1 が成り立つことが分かる．(図-8と図-10を比較せよ．)

以上はいわば静的な面での対応であるが，ラムダ計算には β-簡約という動的な面もある．論理体系でこの動的な面に対応するのがこれから説明する証明木の簡約という概念である．証明木の中で introduction 規則に elimination 規則が続く部分があれば簡約することができる．具体的には，直観主義二階命題論理の簡約規則を示したのが図-11 である．(直観主義命題論理や直観主義一階述語論理の簡約規則はこれから明らかであろう．) ここで

$$\begin{matrix}[A] & \Pi \\ \Sigma\,, & A \\ & \Sigma\end{matrix}$$

という二つの表記はそれぞれ Σ の A という仮定が

$$\frac{\dfrac{\Pi_0 \quad \Pi_1}{\dfrac{A_0 \quad A_1}{A_0 \wedge A_1}(\wedge I)}}{A_i}(\wedge E_i) \quad \Rightarrow \frac{\Pi_i}{A_i} \quad (i=0,1)$$

$$\frac{\dfrac{\dfrac{\Sigma}{A_i}}{A_0 \vee A_1}(\vee I_i) \quad \dfrac{\overset{[A_0]}{\Pi_0}}{C} \quad \dfrac{\overset{[A_1]}{\Pi_1}}{C}}{C}(\vee E) \quad \Rightarrow \frac{\dfrac{\Sigma}{A_i}}{C} \quad (i=0,1)$$

$$\frac{\dfrac{\overset{[A]}{\dfrac{\Sigma}{B}}}{A \supset B}(\supset I) \quad \dfrac{\Pi}{A}}{B}(\supset E) \quad \Rightarrow \frac{\dfrac{\Pi}{A}}{\dfrac{\Sigma}{B}}$$

$$\frac{\dfrac{\dfrac{\Pi}{A[Q/P]}(\forall, I)}{\forall P.A}}{A[B/P]}(\forall, E) \quad \Rightarrow \frac{\Pi[B/Q]}{A[B/P]}$$

$$\frac{\dfrac{\dfrac{\Sigma}{A[B/P]}}{\exists P.A}(\exists, I) \quad \dfrac{\overset{[A[Q/P]]}{\Pi}}{C}}{C}(\exists, E) \quad \Rightarrow \frac{\dfrac{\Pi}{A[B/P]}}{\dfrac{\Pi[B/Q]}{C}}$$

図-11 簡約規則

discharge された木と Σ の仮定の A の上に Π をのせてできる木を表している．(この規則の場合どの A の上にのせればよいかは明らかであろう．) そして証明木が⇒の左の形をしている部分木をもつときそれを⇒の右の木で置き換える，という操作により証明木の簡約が行われる．証明木にこの操作が施せないときその証明木は正規形であるという．(formulas-as-types 原理による項と証明木の対応を考えれば (β) は $(\supset red)$ に，(β_2) は $(\forall red)$ に対応することは明らかであろう．) このとき上記の三つの体系で定理 3.9 の拡張である '証明木は strongly normalizable である' という性質が成り立つ．(証明は文献[6] を見よ．本質的には定理 3.9 の証明と同じであるが．)ここまでくれば逆に有限型付ラムダ計算や多相型付ラムダ計算を直観主義命題論理や直観主義二階命題論理に対応するように拡張しようと考えるのは当然である．拡張の仕方は明らかであるが，問題はそのように拡張された型付ラムダ計算がどのように使われるか，である．一つの例が SOL[13] である．これは直観主義二階命題論理に対応する型付ラムダ計算であり，(詳しくは文献[13] を参照して欲しいが) いわゆる抽象型を二階の∃で解釈する，という主張をしている．その他，たとえば $(\wedge I)$ はペアを作る操作であり，$(\forall_2 I)$ が多相型のプログラムを作る操作であるといったようなことは容易に理解できるであろう．(recursion もしくは loop に相当するものがないことに気が付いた読者もいると思うが，本稿ではそれについては触れない．文献[12]を参考にして考えよ．)

最後に 2.2 で述べた構成的説明について触れるが，紙面の制限があるのでごく大ざっぱな説明にとどめる．証明木が正規形であるとき (いいかげんないい方であるが) 上のほうが elimination 規則で下のほうが introduction 規則という形になっている．なぜなら introduction 規則に elimination 規則が続く部分があればその部分は簡約可能であるからである．(ある論理記号に関する introduction 規則に続く elimination 規則は必ずその論理記号に関するものである．ただしここで続くといっているのは下図の $(\vee E)$ の場合

$$\dfrac{\begin{array}{ccc} & [A] & [B] \\ A\vee B & C & C \end{array}}{C}(\vee E)$$

$(\vee E)$ に続くのは $[A]$ または $[B]$ の下にある規則であり，$(\vee E)$ の上下にある C については素通しになっている，つまり間に推論規則がはいっていないとみなす．($(\exists E)$ と $(\exists_2 E)$ の場合も同様に考える．) したがってある証明木が $A\vee B$ を証明していればそれを正規化することにより $(\vee I_0)$ か $(\vee I_1)$ で終わるようにすることができる．したがって A か B のどちらかが証明されている．∃についても同様の議論ができる．これにより，disjunction property, existence property が今まで述べてきた三つの直観主義論理に対して成り立つことが分かる．さらに∧，⊃，∀についても同様の議論を適用できる．たとえば $A\supset B$ の証明木があるとする．それを正規化すれば $(\supset I)$ で終わる形となり，これに $(\supset E)$ が続けば (すなわち引数が apply されたら) 簡約が起こる形である．つまり $A\supset B$ の証明木は関数とみなすことができる．

したがって normalization によって構成的説明を与えることができるのである．

5. 型 理 論 (その 2)

これまではできるだけ話を簡単にするため述語論理に対応するような型理論について述べなかったが，もちろん無視することはできない重要な話題なのでこの章で少し触れておく．たとえば直観主義一階述語論理に対応する型理論を考えようという場合，一番安直な方法としては 2.1 で定義した直観主義一階述語論理をそのまま型付ラムダ計算とみなす，というやり方が考えられる．つまり証明木を項，論理式を型，証明木の簡約を計算規則としてラムダ計算を定めるのである．これはこれで特に文句をつける筋合もないのだが，単に論理体系をラムダ計算にコピーしただけでもう一つおもしろ味がない．(といってしまえば今まで述べてきた formulas-as-types 原理による命題論理と型付ラムダ計算の対応も自明なもの，ということになる．ある意味ではそうなのだが，独立して発展した直観主義論理と型付ラムダ計算の間に対応が発見されたのだし，この対応によって直観主義論理の構成的性格を浮き彫りにできるのだからやはり相応の価値はあるといってよいだろう．) ということで，以下 formulas-as-types 原理により述語論理とみなすことができる部分を含んでおりしかも上でいったような自明なものではない型理論として，Martin-Löf の直観主義型理論[12] (intuitionistic type theory) と Coquand-Huet の Calculus of Constructions[4] を紹介するが，完全に紹介する余裕はないので formulas-as-types 原理とどうかかわるか，という点に焦点をあてて述べる．

5.1 Martin-Löf の型理論

Martin-Löf の直観主義型理論（以下 ML と書く）は judgement と呼ばれる基本的な表現を四つもっており ML は judgement を導く規則から成る体系である．judgement の一つは $a \in A$ という形をしており これは 'a は型 A の要素である' と 'a は論理式 A の証明である' という2とおりの読み方をすることができる．（さらに別の読み方もできるが本稿では触れない．）ここでは ML の規則のうち上の二つの読みが交錯する規則を一つだけ取り上げる．

$$\frac{b \in \Pi x \in A.\, B \quad a \in A}{ap(b, a) \in B[a/x]}$$

これは Π-elimination と呼ばれる規則（$(\forall E)$ と比較せよ）である．$\Pi x \in A.\, B$ は依存型と呼ばれる型であるが論理式として解釈すると '型 A のすべての要素 x について B が成り立つ' と読める．この論理式が b によって証明されているとき型 A の要素 a が与えられれば $ap(b, a)$ は $B[a/x]$ の証明になる．もちろん Π-elimination を 'b が型 $\Pi x \in A.\, B$ をもち，…' と読んでもよいのだが，上のように読めば formulas-as-types 原理が現れていることが分かるであろう．

$a \in A$ が成り立つとき A を論理式とみなせば a は A の証明になっているのだが，これらは直観主義一階算術*を含む論理体系の証明木と論理式になっている．したがって ML はその一部にある種の一階述語論理を含む型理論であるということができる．

5.2 Calculus of Constructions

述語論理において $p(t)$ という原子論理式を考える．この p というものは1引数の述語記号であって，p が単独で（つまり $p(t)$ という形ではなく）出現することはない．p の働きを考えてみると，項と結びついて論理式を作り出す働きとみなせる．F が論理式であるということを F は '論理式という型' をもつ 'もの' と読み換えてみれば，この p の働きは型理論ふうに

$$\frac{\Gamma \vdash p : A \to *\quad \Gamma \vdash t : A}{\Gamma \vdash p(t) : *}$$

と書ける．ここで $*$ は '論理式という型' を表現している．（A は適当な型とする．）このような設定では p は単独で出現できる．さらに，二階述語論理や高階述語論理では論理式の abstract という概念が自然に導入される．これも型理論ふうに書くと

$$\frac{\Gamma, x^A \vdash B : *}{\Gamma \vdash \lambda x^A.\, B : A \to *}$$

となる．（$(\to I)$ の特別な場合である．）

このように考えれば高階の概念を自然に表現できる．Calculus of Constructions はこの考え方を実現した型理論であり，上のような規則に加え

$$\frac{\Gamma \vdash A : *\quad \Gamma \vdash B : *}{\Gamma \vdash A \wedge B : *}$$

のような論理式を構成する規則を含んでいる．（実は $\wedge$ を primitive にする必要はない．上の規則は単に説明の便宜上のものである．）そして型 $*$ をもつものが論理式で，論理式を型にもつものがその論理式の証明である，という formulas-as-types 原理に従った読みを可能にしており，このときの論理体系は直観主義高階述語論理となっている．

6. おわりに

直観主義論理についても型理論についても formulas-as-types 原理を説明するのに最低限必要な程度の説明にとどまったので，もう少し進んで勉強したい読者のために参考文献を紹介して本稿を終えることにする．

natural deduction のスタイルを用いた論理体系（古典論理，直観主義論理，その他）について書かれた最初の本として 14) は有名である．この本は簡潔であり証明論入門的な性格ももっているので大変手頃であるが残念ながら現在手に入らない．6) は sequent calculus を使った証明論の入門書であるが付録に natural deduction についても詳しく書いてある．ただしこの本は入門書といっても3章以降は急速に難しくなるので初学者は取りあえず1章と2章を読めばよいであろう．

すでに証明論の素養がある読者が natural deduction を使った体系の性質（特に証明木の normalization）について知りたいというのであれば，6) か 15) の概当する部分（2. A, 4. B, 4. C）を読むのが適当であろう．15) が掲載されている proceedings にはこれに関連した論文がいくつか収められている．興味のある読者はそれらも参照するとよいだろう．

2.2 で述べた直観主義論理の構成的説明についてももっと詳しく知りたければ 1) を参照せよ．直観主義論理だけでなくその他のさまざまな構成的論理について知りたい場合も 1) が適当であろう．

10) は formulas-as-types 原理の発見者の一人 Howard による formulas-as-types 原理の解説である．その他 5), 13) にも解説がある．

☆ 厳密な定義はここでは不要なのでしないが，要するに直観主義一階述語論理の一種で自然数を扱うための体系であり加算や乗算に関する公理や帰納法の規則が加わっている．

3. で述べた型付ラムダ計算がプログラミング言語としてどれぐらいの表現能力があるのかについては本稿ではまったく触れなかった．データ構造や制御構造という観点から述べているのが 2)，計算の複雑さという観点から述べているのが 5) である．その他型理論の話題については 9) の文献案内を参照されたい．

Martin-Löf の直観主義型理論については基本的な文献は 12) であり，プログラミングとの関連についても触れてあるのが 11) である．さらに他の論理体系との関連なども知りたければ 1) のXI章 (Martin-Löf の直観主義型理論の入門的解説も含んでいる) が適当であろう．

Coquand-Huet の Calculus of Constructions については 4) が基本的な文献である．彼らがこの理論を発表したのは 1984 年であるが，古いレポートなどには誤りも含まれているので 4) を読むのがよい．3) は 4) とは少し体系が異なり体系の説明も少ない．(しかもより強い体系の存在を示唆しているがそれは誤っている.) しかし自力で変換して読むことができれば多くの興味深い例を発見できるであろう．また Calculus of Constructions によく似た体系に 7) の Logical Framework がある．こちらも参照されたい．

参 考 文 献

1) Beeson, M. J.: Foundations of Constructive Mathematics, Springer-Verlag (1986).
2) Böhm, C. and Berarducci, A.: Automatic Synthesis of Typed Λ-programs on Term Algebra, Theoretical Computer Science, Vol. 39 (1985).
3) Coquand, T. and Huet, G.: Constructions: A Higher Order Proof System of Mechanizing Mathematics, In EUROCAL 85, Linz, Lecture Notes in Computer Science 205 (1985).
4) Coquand, T. and Huet, G.: The Calculus of Constructions, Information and Computation, Vol. 76 (1988).
5) Fortune, S., Leivant, D. and O'Donnell, M.: The Expressiveness of Simple and Second-order Type Structures, J. ACM, Vol. 30, No. 1 (1983).
6) Girard, J.-Y.: Proof Theory and Logical Complexity, volume I. Bibliopolis (1987).
7) Harper, R., Honsell, F. and Plotkin, G.: A Framework for Defining Logics, In Proceedings of the 2nd Annual Symposium on Logic in Computer Science (1987).
8) 林　晋：関数型言語における型理論，情報処理，Vol. 29, No. 8 (1988).
9) 林　晋：文献案内型理論，コンピュータソフトウェア，Vol. 5, No. 1 (1988).
10) Howard, W. A.: The Formulae-as-types Notion of Constructions, In Seldin, J. and Hindley, J. editors, To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, Academic Press (1980).
11) Martin-Löf, P.: Constructive Mathematics and Computer Programming, In Logic, Methodology and Philosophy of Science, VI, North-Holland (1980).
12) Martin-Löf, P.: Intuitionistic Type Theory, Bibliopolis (1984).
13) Mitchell, J. and Plotkin, G.: Abstract Types Have Existential Type: In Proceedings of the 12th Annual Symposium on Principles of Programming Languages (1985).
14) Prawitz, D.: Natural Deduction, Almqist and Wiksell (1965).
15) Prawitz, D.: Ideas and Results in Proof Theory, In Fenstad, J. editor, Proceedings of the Second Scandinavian Logic Symposium, North-Holland (1971).

(平成元年2月17日受付)