8　計算体系と論理体系の関係

本講義の主題は，プログラミング言語に関する研究と形式論理に関する研究が密接に関連することを学ぶことであった．本章では，いよいよ，計算体系と論理体系の関係を解き明かす．

8.1　型付きラムダ計算と直観主義命題論理の関係

これまでの演習を通して，型付きラムダ計算の体系と，直観主義命題論理の体系が極めて似ていることに気付いただろう．もちろん，それは，単に CAL システムを使ったために表記が似通っていたわけではないし，今回の講義のために，無理矢理似たようなシステムをもってきたわけではない．

たとえば，$A \to B$ という型は，$A \supset B$ という命題と非常によく似た規則を持っている．

- $\to$ の導入規則 (λ) と $\supset$ の導入規則 ($\supset I$)

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : A \to B}\ \lambda \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \supset B}\ \supset I$$

- $\to$ の除去規則 ($apply$) と $\supset$ の除去規則 ($\supset E$)

$$\frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B}\ apply \quad \frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}\ \supset E$$

計算体系の $M : A$ という形から $M :$ を取り除けば論理体系の対応する規則になることがわかる．

同じことは，$\times$ の規則 ($pair, left, right$) と $\wedge$ の規則 ($\wedge I, \wedge EL, \wedge ER$)，$+$ の規則 ($inl, inr, case$) と $\vee$ の規則 ($\vee IL, \vee IR, \vee E$) についても言える．また，変数の規則 (var) は仮定を導入する規則 ($assume$) に対応する．全ての場合において，計算体系における $M : A$ という形から項の部分 $M :$ を取り除けば論理体系の対応する規則になることがわかる．

これは偶然であろうか？そうではない．次章で見るようにこれは非常に本質的な現象である．ここでは，その準備として，計算体系と論理体系に若干の (非本質的な) 変更を加える．

- 型付きラムダ計算の体系に，空の型 (要素を持たない型) として $\bot$ を導入する．

  $\bot$ は以下の規則 ($\bot$ 型の除去規則) だけを持つ型である．

$$\frac{\Gamma \vdash M :\bot}{\Gamma \vdash abort(M) : A}\ abort$$

  直感的には，$\bot$ 型は「空の型」であり，要素を持たないものである．したがって，もし $\bot$ 型の値を計算する項 M があったとすれば，Γ 自体が矛盾した仮定であるので，そこからどんな型 A の要素も計算できてしまえる，というのがこの規則の意味である．このような型を導入しても，計算体系は本質的には何も変わらない，ということが直感的に想像されるであろう．すなわち，$abort$ を含む項 (プログラム) を実行したとき，$abort(M)$ という部分は決して実行されない (その部分に制御が行くことはない)．

  要素を持たない型は存在意義がないと考えるかもしれないが，Java 言語の try-catch, ML 言語の exception などで表現される例外機構の記述では有用である．それについては,「古典論理への拡張」の節で述べる．

- 直観主義命題論理の体系で $\neg$ は省略形と考える．すなわち，$\neg A$ は $A \supset \bot$ の省略形と見なし，$\neg$ という記号はないと考える．したがって，$\neg$ に関する 2 つの規則 $\neg I$ と $\neg E$ もないものと考える．

8.2 Curry(-Howard) の同型対応

Curry-Howard の同型対応は,「命題＝型」原理 (Propositions-as-Types Principle) とも呼ばれ, 命題と型が本質的に同じであることを主張するものである．しかし，それは，単に命題と型が同じ導出規則で生成されるという静的な性質の対応だけではない．命題の導出と型の導出とが対応し，さらに，命題の導出に対する計算と型の導出に対する計算が対応する，という動的な性質の対応を含むものである．

ここで,「命題の導出に対する計算」とは何であろうか？それは前章で見た「無駄を省く変形操作」のことである．では,「型の導出に対する計算」とは何であろうか？型の導出とは，何らかの項がある型を持つ，ということの導出のことであり,「型つきラムダ項の導出」と呼んでいたものであった．つまり,「型の導出に対する計算」とは，型つきラムダ項の計算に他ならない．

以上まとめると以下の表を得る．

論理体系	**計算体系**
命題	**型**
$\supset$	$\rightarrow$
$\wedge$	$\times$
$\vee$	$+$
$\bot$	$\bot$
命題の導出 ($\vdash A$ の導出)	**型の導出 ($\vdash A'$ の導出)**
$\supset I$ 規則, $\supset E$ 規則	λ 規則, $apply$ 規則
$\wedge I$ 規則, $\wedge EL$ 規則 $\wedge ER$ 規則	$pair$ 規則, $left$ 規則, $right$ 規則
$\vee IL$ 規則, $\vee IR$ 規則, $\vee E$ 規則	inl 規則, inr 規則, $case$ 規則
$\bot E$ 規則	$abort$ 規則
この導出に対する変形操作 (無駄を省く変形)	**この導出の計算 (項の計算)**
$\supset I - \supset E$ 変形	β 計算規則
$\wedge I - \wedge EL$ 変形	pair-left 計算規則
$\wedge I - \wedge ER$ 変形	pair-right 計算規則
$\vee IL - \vee E$ 変形	case-inl 計算規則
$\vee IR - \vee E$ 変形	case-inr 計算規則

この表の意味するところは以下の通りである．

- 命題 A は，それにあらわれる $\supset, \wedge, \vee, \bot$ をそれぞれ $\rightarrow, \times, +, \bot$ に読みかえることにより型 A' に対応する．
- A の導出 (正確には，$\vdash A$ という判断の導出) は，その中に現れる規則を，表の 2 番目に列挙した対応関係 ($\supset I$ を λ 規則で置きかえ，$\supset E$ を $apply$ 規則で置きかえる，等) で置きかえることにより，A' の導出 (正確には，適当な項 M に対して $\vdash M : A'$ の導出) になる．
- A の導出に対する計算 (変形操作の系列) は，各変形操作を，表の 3 番目に列挙した対応関係 ($\supset I - \supset E$ 変形を，β 計算規則で置きかえる，等) で置きかえることにより，A' の導出に対する計算となる．

すなわち，直観主義論理の命題の導出と計算は，型付きラムダ計算の項の導出と計算に完全に対応する．これらは，片方が与えられれば他方が機械的に (一意的に) 定まる，という全単射で対応付けられている．ところで，計算体系にしても論理体系にしても，我々は形式的な構文と導出

のみで定まるものとしてきたので，これらが完全に対応する，ということは，直観主義論理と型付きラムダ計算の体系は同じものである，ということになる．(もちろん，最初にこれらの体系を構築した際に意図した意味論は違うものであったはずだが，結果として形式化されたものはまったく同じである．)

これが Curry-Howard の同型対応である．Curry-Howard の同型対応は，計算と論理の形式的体系の間の深い関係を示すものである．なお，歴史的には，ここで述べたような命題論理の対応は Curry が発見し，後に少々触れる述語論理まで拡張した際の対応を Howard が発見したので，本節で述べた範囲での対応は正確には「Curry の同型対応」というべきである．

8.3 Curry-Howard の同型対応の周辺

前節の結果は理論的には美しい結果であるが，ただちにいくつかの疑問がわくだろう．

- そもそも何故 Curry-Howard の同型対応がなりたったのであろうか？
- Curry-Howard の同型対応が発見されたからといって，何が嬉しいのだろうか？ 2 つの独立に作られたものが同じことがわかると，何か本質的な進歩があるのだろうか？
- 型付きラムダ計算に対応するのは，古典論理ではなく，直観主義論理である．我々の日常的推論が古典論理でおこなわれていることを考えると，直観主義論理と対応することの意味は何だろうか？
- 形式的体系がたまたま同じだからといって，それは形式化のやり方に依存することであって,「計算」と「論理」が本質的に同じという結論を導いていいのだろうか？それぞれの意味も同じといっていいのだろうか？

これらの疑問について以下では考えてみる．

8.3.1 なぜ対応するのか？

直観主義論理の形式的体系に「ぴったり一致する」意味論として，構成的解釈があることを既に学んだ．これを使って，Curry-Howard 同型対応が「必然的」なものであることが説明できる．

- 構成的解釈において $A \supset B$ という命題の証拠は,「A の証拠をもらって B の証拠を返す関数」であると定められた.「命題 A の証拠」を「命題 A に対応する型 A' に族する要素」という対応関係でみると,「A の証拠をもらって B の証拠を返す関数」は「A 型の要素をもらって B 型の要素を返す関数」であり，そのような関数を集めた型は，ちょうど $A \to B$ という型になる．
- 構成的解釈において $A \wedge B$ の証拠は「A の証拠と B の証拠の対」であった．これを「命題 A の証拠」を「命題 A に対応する型 A' に族する要素」という対応関係でみると,「A' 型の要素と B' 型の要素の対」であり，これを集めた型は，ちょうど $A \times B$ という型になる．
- 構成的解釈において $A \vee B$ の証拠は,「A であるか B であるかをあらわす 1bit の情報と，A の証拠もしくは B の証拠の対」であった．これを「命題 A の証拠」を「命題 A に対応する型 A' に族する要素」という対応関係でみると,「A' 型であるか B' 型であるかを決める 1bit の情報」と,「A' 型の要素，もしくは，B' 型の要素」の対になる．これを集めた型は，ちょうど $A + B$ という型になる．

- 命題 $\bot$ には証拠がないが，これは，要素を持たない型 $\bot$ に対応している．

言い換えると，構成的解釈を表現するのに必要なだけの表現力をもった計算体系が型付きラムダ計算であるといえる．このように考えると，直観主義論理と型付きラムダ計算が対応するのは，むしろ当然のことと言える．

8.3.2 なにが嬉しいのか？

一般に，全く異なる生い立ちをもつ 2 つの理論の関係が解き明かされると，新たな発展が期待される．Curry-Howard の同型対応でも事情は同じであり，計算体系と論理体系で別々に知られていた事実の間の関係がわかったり，片方の結果を他方に移すことで新たな展開が知られたりする．このような観点を基礎とする研究は非常に盛んであり，**プログラミング言語論の研究の中心的テーマの 1 つ**である．

この章の後の方で，そのような話題を 4 つ紹介する．

8.3.3 直観主義論理と対応したことの意味は何か？

直観主義論理は，構成的解釈に対応する形式的体系であることからわかるように「計算」によって論理を解釈しようとする体系である．したがって，このような論理が計算体系と対応したからといって，あまり驚きはないかもしれない．いずれにせよ，**「直観主義論理＝計算の論理」**である．

では，純粋な論理として見たとき，直観主義論理はどんなものであろうか？既に見てきたように，本講義のような定式化をする限り，直観主義論理の証明は古典論理の証明よりはるかに自然である．命題 A が導出可能なとき，命題 A の部分命題しかあらわれないような導出が必ず存在する．このような論理は，純粋な論理体系としても非常に興味深いものであることが想像できよう．

さらに，近年の数学の流れとして，構成的な存在証明への関心があげられる．存在証明とは，$\exists x.A(x)$ という形の定理の証明のことである．一般に，数学の定理は排中律等を多用して証明されるため，$\exists x.A(x)$ が証明されたからといって，この x を具体的に計算することができる保証はまったくない．実際,「すべての実数を順番に並べる方法がある」という定理が成立するが，具体的な並べ方を紙に書くことは (どんな数学者といえども) できない．また,「どんな円でも半径と円周の長さの比率が一定である」という定理から円周率を計算することはできないだろう．しかし，近年の数学の傾向として，$\exists x.A(x)$ の証明は，構成的であることが望ましい，とされるようになってきた．構成的な存在証明とは，その証明の中に「x を具体的に計算する手続き」がはいっているものである．構成的証明は，直観主義論理で証明を作ることに相当し，一般の (非構成的かもしれない) 証明は，古典論理で証明を作ることに相当する．直観主義論理は，古典論理における排中律 (あるいはそれと同等な他の原理) を使うことができないので，証明を作る段階では苦労が多いが，証明できてしまえば,「計算手続きを含む」という非常に良い性質を持つものである．

以上のように，直観主義論理，あるいは，構成主義は，純粋な論理や数学の世界でも注目されている重要な論理であるといえる．

8.3.4 では，計算と論理は本当に同じか？

この問に対する答えは 1 つではない．論理体系や計算体系の定式化 (形式的体系を作ること) は，一意的ではなく，多様な形式的体系があり得るからである．

論理体系の形式化は，3 つのスタイルがある．

- 自然演繹 (natural deduction), 本講義で採用したスタイル

- シーケント計算 (sequent calculus)

$$\frac{}{\Gamma, A \vdash \Delta, A}\ intial \qquad \frac{\Gamma, A \vdash \Delta, B}{\Gamma \vdash \Delta, A \supset B}\ \supset R \qquad \frac{\Gamma_1, B \vdash \Delta_1 \quad \Gamma \vdash A, \Delta_2}{\Gamma_1, \Gamma_2, A \supset B \vdash \Delta_1, \Delta_2}\ \supset L$$

  自然演繹における I(導入規則) と E(除去規則) のかわりに，このように，L(⊢ の左側に導入) 規則と R(⊢ の右側に導入) 規則とがある．

- Hilbert 流 (Hilbert が考案したスタイル)

$$\frac{\Gamma \vdash A \supset B \quad \Gamma \vdash A}{\Gamma \vdash B}\ modus\ ponens$$

$$\frac{}{\Gamma \vdash (A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))}\ S \qquad \frac{}{\Gamma \vdash A \supset (B \supset A)}\ K$$

  推論規則は modus ponens だけで，あとは全て S, K のような公理である．

古典論理や直観主義論理など多くの論理体系は，これら 3 つのスタイルのいずれでも形式化でき，どのスタイルでの形式化も同等 (証明できる論理式は同じ) であることが知られている．

Curry-Howard の同型対応は，このうち，自然演繹スタイルと Hilbert スタイルに対して成立するものである．シーケント体系スタイルに対しては Curry-Howard の同型対応に相当するものはない．したがって，シーケント計算スタイルでしか与えられていない論理体系は，対応する計算体系がない，ということになる．

一方，プログラミング言語は非常に複雑なものが多い (ほとんどの実用的なプログラミング言語は，その厳密な定義を書くと 1 冊の本になる．) 本講義で述べた型付きラムダ計算であれば，それに対応する論理体系を考えることは容易であったが，たとえば，C 言語に対応する論理体系を厳密に定義するのはほとんど不可能である．

このような観点から，計算と論理が「どんな場合にも完全に一致する」ということは言えないことがわかる．

しかしながら，以下のようなことは言える．

- 論理体系のみ，あるいは，計算体系のみを考察対象としているときに比べて，Curry-Howard の同型対応が成立するような論理体系と計算体系がある場合，(後に述べる 4 つの例からもわかるように) 体系に関する種々の良い性質を証明する手だてが用意され，見通しの良い議論を展開することができる．

- したがって，論理体系のみ，あるいは，計算体系のみが与えられた場合,「Curry-Howard の同型対応が成立するような相棒はなにか？」を考えることは有意義である．そのような相棒が存在しない場合，多くの場合は，もとの体系が適切に構成されていないことを意味する．

- すなわち，C-H の同型対応が成立するように論理/計算体系を設計すべき，という設計の指針となる．