

Dual Calculus の Coterm とは何者か

角田健太郎 (Kentaro Tsunoda)

東京都立大学

Dummett は、推論の連鎖を段階的な論拠構築（あるいは正当化）と捉えることのできる直観主義論理を出発点として、命題の正当性やそこに現れる語の意味に関する体系的説明を展開した。そして同時に、論理的推論の妥当性を真理保存性に訴えることで説明する古典論理を無意味として切り捨てた。というのも、二値的解釈を許す古典論理は推論の連鎖が段階的な正当化だとする彼の証明に対する要求を満たさず、結局古典論理的証明は心理学的説得のような認識論的な言語活動に過ぎないと彼は考えたためである。

しかし Dummett ののち、1990 年代から 2000 年代にかけて、継続呼び出し機構を持つ非決定的なプログラムを構築する計算体系と古典論理が対応することが計算機科学の分野で明らかとなった。つまり、古典論理の推論の連鎖も一非決定的プログラムの構築という一ある種の段階的な合理化であると見なせることが明らかとなった。では、古典論理的証明が持つそうした合理化（あるいは証明項の形）の哲学的意義とは何だろうか？これが本発表の問いである。

さて、古典論理的証明の証明項を解釈する方法としては、Gentzen の古典シーケント計算 LK についての Wadler の研究がある。Wadler はまず LK と対応する計算体系 dual calculus を開発し、LK 証明図が持つ証明項の形を明らかにした。そして dual calculus から型付きラムダ計算への双対的な二つの翻訳（値呼び CPS 変換と名前呼び CPS 変換と呼ばれる）を与え、LK 証明図の構成的解釈法とその双対性を明らかにした。Wadler の研究は、Dummett の直観主義哲学と調和するような仕方で古典論理に意味を与える道を開いていると発表者は考える。

しかし発表者は今回、Dummett の哲学と調和するような古典論理的証明が持つ合理化概念ではなく、むしろ Dummett が真に切り捨ててしまうであろう古典論理的証明が持つ合理化概念とは何かという問題に焦点を合わせる。つまり、CPS 変換前に LK 証明図の証明項が持ち、かつその CPS 変換像が持たないものとは何かという問題である。

このような見通しのもと本発表は、dual calculus の証明項に現れる一方で CPS 変換後に姿を消してしまう coterm—ここでは「純粋に値の消費者として働く継続」の意味で用いている—が何者であるかを報告する。具体的には、call/cc 等の継続呼び出し演算

子を含む実際のプログラミングにおいて、*coterm* が概念上はどのような仕方で非関数的に含まれていると言えるのかを報告する。そして、純粋な値の消費者・非関数的な存在者を含む計算体系として古典論理を見た時、改めて古典論理的命題はどのように解釈することができるのかを検討する。

参考文献

- Dummett, M (1973) The Philosophical Basis of Intuitionistic Logic. *Truth and Other Enigmas*. Cambridge: Harvard UP. pp.215-247
- Gentzen, G (1935) Investigations into Logical Deduction. *Mathematische Zeitschrift 39*, pp.176-210,405-431. Reprinted in M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, North-Holland, 1969, pp.68-131
- Reynolds, J (1993) The discoveries of continuations. *Lisp and Symbolic Computation*, 6(3/4): pp.233-248
- Wadler, P (2003) Call-by-value is dual to call-by-name. *Proceedings of the eighth ACM SIGPLAN International Conference on Functional Programming (ICFP' 03)*, pp.189-201
- Wadler, P (2005) Call-by-value is dual to call-by-name, Reloaded. *Proceedings of Rewriting Techniques and Applications (RTA)*, Lecture Notes in Computer Science, 3467: pp.185-203