A Stochastic λ-Calculus

Dana S. Scott
University Professor, Emeritus, Carnegie Mellon University
Visiting Scholar, UC Berkeley

As a formal theory, the λ-Calculus has equational rules for the ***explicit definition*** of functions in a type-free setting. The rules can be sketched as follows:

α-conversion

$$\lambda X.[...X...] = \lambda Y.[...Y...]$$

β-conversion

$$(\lambda X.[...X...])(T) = [...T...]$$

η-conversion

$$\lambda X.F(X) = F$$

Background and historical references can be found in the recent survey *Cardone-Hindley [2009]* cited below. ***The Graph Model*** for this theory satisfies the first two equations and modifies the third. The idea incorporates the notion of ***enumeration operators*** from Recursive Function Theory. (See the additional references cited.) The construction of the model proceeds as follows, where the elements of the model are just sets in the ***powerset*** of the integers, $\mathcal{P}\,\mathbb{N}$.

Define a ***pairing function*** on the integers by: $(n, m) = 2^n\,(2\,m + 1)$. This makes every positive integer uniquely the "Gödel number" of a pair of integers. Define numbering of ***finite sequences*** of integers by: $\langle\,\rangle = 0$ and $\langle n_0, n_1, \ldots, n_{k-1}, n_k\rangle = (\langle n_0, n_1, \ldots, n_{k-1}\rangle, n_k)$. That definition makes every integer uniquely the number of a finite sequence of integers. Further, define a numbering of ***finite sets*** of integers by: $\textbf{set}\,(\langle n_0, n_1, \ldots, n_{k-1}\rangle) = \{n_0, n_1, \ldots, n_{k-1}\}$. And for $X \in \mathcal{P}\,\mathbb{N}$, define $X^* = \{n \in \mathbb{N} \mid \textbf{set}\,(n) \subseteq X\}$.

The structure of the model is given by defining:

Application

$$F\,(X) = \{\,m \mid \exists\, n \in X^*.\,(n, m) \in F\,\}$$

Abstraction

$$\boldsymbol{\lambda} X.[\,...\,X\,...\,] = \{0\} \bigcup \{\,(n, m) \mid m \in [\,...\,\textbf{set}\,(n)\,...\,]\,\}$$

λ-Abstraction applies to any context defining a function $\Phi\,(X) = [\,...\,X\,...\,]$ which is ***continuous*** as a function $\Phi : \mathcal{P}\,\mathbb{N} \to \mathcal{P}\,\mathbb{N}$ in the ***topology*** on $\mathcal{P}\,\mathbb{N}$ with the sets $\mathcal{U}_n = \{\,X \subseteq \mathbb{N} \mid n \in X^*\,\}$ as a ***basis*** for the topology. The model satisfies the first two conversion rules and shows that ***any*** continuous function can be given by application with a suitable subset $F \subseteq \mathbb{N}$. The third conversion rule has to be weakened to $F \subseteq \boldsymbol{\lambda} X.F\,(X)$. The continuous functions ***are*** the general enumeration operators, and the ***computable*** ones are those given by r.e. sets F. There are many nice properties relating this λ-calculus structure to the lattice structure of $\mathcal{P}\,\mathbb{N}$. A rich ***type theory*** can be added to $\mathcal{P}\,\mathbb{N}$ by using ***partial equivalence relations*** (PERs) on $\mathcal{P}\,\mathbb{N}$ as types. There is no room to give the definitions in this abstract, however.

Probability can now be added to the model simply by using ***random variables***. For example, consider ***Borel functions*** $\mathbf{X} : [0, 1] \to \mathcal{P}\,\mathbb{N}$. These form a **λ**-calculus structure in the same way that real-valued random variables form a linear vector space. By using the standard measure on $[0, 1]$, probabilities can be assigned to logical formulas involving **λ**-calculus equations. Inasmuch as **λ**-calculus is a programming language, random variables can be introduced as parameters in defining ***randomized algorithms***. This gives us a ***denotational semantics*** for a stochastic **λ**-calculus.

There are many approaches to modeling λ-calculus, and expositions and historical references can be found in Cardone-Hindley [2009]. In 1972 Plotkin wrote an AI report at the University of Edinburgh entitled "A set-theoretical definition of application" which remained unpublished until it was incorporated into the more extensive paper Plotkin [1993]. Scott developed his model based on the powerset of the integers subsequently, but he only later realized it was basically the same as Plotkin's model. See Scott [1976] for further details where he called the idea ***The Graph Model****.*

• F. Cardone and J.R. Hindley. Lambda-Calculus and Combinators in the 20th Century. In: Volume 5, pp. 723-818, of Handbook of the History of Logic, Dov M. Gabbay and John Woods eds., North-Holland/Elsevier Science, 2009.

• Gordon D. Plotkin. Set-theoretical and other elementary models of the λ-calculus. Theoretical Computer Science, vol. 121 (1993), pp. 351–409.

• Dana S. Scott. Data types as lattices. SIAM Journal on Computing, vol. 5 (1976), pp. 522–587.

Much earlier, enumeration reducibility was introduced by Rogers in lecture notes and mentioned by Friedberg-Rogers [1959] as a way of defining a positive reducibility between sets. Enumeration degrees are discussed at length in Rogers [1967], and there is now a vast literature on the subject. Enumeration operators are also studied in Rogers [1967] as well. Earlier, Myhill-Shepherdson [1955] defined functionals on partial functions with similar properties. Neither of those teams saw that their operators possessed an algebra that would model λ-calculus, however.

• John Myhill and John C. Shepherdson, Effective operations on partial recursive functions, Zeitschrift für Mathematische Logik und Grundlagen der Mathematik, vol. 1 (1955), pp. 310–317.

• Richard M. Friedberg and Hartley Rogers jr., Reducibility and Completeness for Sets of Integers. Mathematical Logic Quarterly, vol. 5 (1959), pp. 117–125.   Some of the results of this paper are presented in abstract, Journal of Symbolic Logic, vol. 22 (1957), p. 107.

• Hartley Rogers, Jr., Theory of recursive functions and effective computability, McGraw-Hill, 1967, xix + 482 pp.

Some historical remarks on the notion of PER as an interpretation of types are given by Bruce et al. [1990], where we learn that they were introduced by Myhill and Shepherdson [1955] for types of first-order functions, and then extended to simple types by Kreisel [1959]. Scott took the use of partial equivalence relations from the work of Kreisel and collaborators.

• K. Bruce, A. A. Meyer, and J. C. Mitchell. The semantics of second-order lambda calculus.   In G. Huet, editor. Logical Foundations of Functional Programming, pp. 273–284. Addison-Wesley, 1990.

• G. Kreisel. Interpretation of analysis by means of constructive functionals of finite type. In A. Heyting, editor, Constructivity in Mathematics, pp. 101–128. North-Holland Co., Amsterdam, 1959.

Two papers about introducing random features in λ-calculus are Deliguoro-Piperno [1995] and Dal Lagoa-Zorzia [2012].
Both of those articles have many background references.

• U. Deliguoro and A. Piperno. Nondeterministic Extensions of Untyped λ-Calculus. Information and Computation, vol. 122 (1995), pp. 149–177.

• Ugo Dal Lagoa and Margherita Zorzia. Probabilistic operational semantics for the lambda calculus. RAIRO - Theoretical Informatics and Applications, vol. 46 (2012), pp. 413-450.