LOOKING BACKWARD — LOOKING FORWARD

Appeal to Future: *Make the Models Do More Work!*

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

Domains XIII
Oxford, 7–8 July 2018
FLoC 2018: Federated Logic Conference 2018

Axioms for λ-Calculus

Constants: none

Variables: `x, y, z, ...`

Terms: expressions built up from variables using a binary ***application operation*** `M(N)` and a variable-binding operation of **λ**-***abstraction*** `(λx.M)`.

Substitutions: `M[N/x]` is defined for each variable `x` by replacing all ***free*** occurrences of `x` in `M` by a copy of `N` — ***provided that*** no free variables in `N` get captured by a variable binder in `M` or confused with other free variables.

Axioms: *(provided the substitutions are defined)*

(α) `(λx.M) = (λy.M[y/x]),` `y` not free in `M`.

(β) `(λx.M)(N) = M[N/x]`

(η) `(λx.f(x)) = f`

Question: Would a better notation have been $(x \mapsto M)$ **?**

Some Tools for Models

(1) Countably based algebraic lattices (with tops).

(2) Dcpos, esp. topologically closed subsets of (1) (no tops).

(3) Various kinds of PCAs (partial combinatory algebras).

Notes: **(a)** Both in (1) and (2) the partial orderings have natural topologies.

(b) For (3), Kleene's $\mathbf{K_2}$ has a good topology also.

(c) Both in (1) and (2) there are ***universal models.***

(d) Both universal models retract to their own continuous ***function spaces.***

(e) Both universal models (and $\mathbf{K_2}$) have a good notions of ***computability*****.**

The Powerset of the Integers

The powerset $\mathcal{P}(\mathbb{N}) = \{\, \mathtt{X} \mid \mathtt{X} \subseteq \mathbb{N} \,\}$ becomes a ***T_0-topological space*** with the sets of the form $\{\, \mathtt{X} \subseteq \mathbb{N} \mid \mathtt{E} \subseteq \mathtt{X} \,\}$ as a ***neighborhood base***, where $\mathtt{E}$ is taken as a ***finite*** set.

Note 1: The ***open*** subsets of $\mathcal{P}(\mathbb{N})$ in this topology are exactly those collections where a set belongs iff some finite subset belongs.

Note 2: The ***continuous*** functions $\boldsymbol{F}\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ are those where for all $\mathtt{X} \in \mathcal{P}(\mathbb{N})$ and all finite $\mathtt{E} \in \mathcal{P}(\mathbb{N})$ we have $\mathtt{E} \subseteq \boldsymbol{F}(\mathtt{X})$ iff there is a finite $\mathtt{D} \subseteq \mathtt{X}$ with $\mathtt{E} \subseteq \boldsymbol{F}(\mathtt{D})$.

Note 3: $\mathcal{P}(\mathbb{N})$ can be shown to be ***universal*** for countably based algebraic lattices in several senses.

Embedding Spaces as Subspaces

Theorem. Every countably based T_0-space $\mathcal{X}$ is homeomorphic to a ***subspace*** of $\mathcal{P}(\mathbb{N})$.

Proof Sketch: Let a subbasis for the topology of $\mathcal{X}$ be $\{ \mathcal{O}_n \mid n \in \mathbb{N} \}$.
Define $\varepsilon : \mathcal{X} \to \mathcal{P}(\mathbb{N})$ by $\varepsilon(x) = \{ n \in \mathbb{N} \mid x \in \mathcal{O}_n \}$.
By the T_0-axiom, this mapping is one-one onto a subspace of $\mathcal{P}(\mathbb{N})$.
Check first that the ***inverse image*** of opens of $\mathcal{P}(\mathbb{N})$ are open in $\mathcal{X}$.
Notice next that $\varepsilon(\mathcal{O}_n) = \varepsilon(\mathcal{X}) \cap \{ S \in \mathcal{P}(\mathbb{N}) \mid n \in S \}$.
Hence, the ***image*** of an open of $\mathcal{X}$ is an open of the subspace.
Therefore, ε is a homeomorphism to a subspace. Q.E.D.

Reference: P. Alexandroff. *Zur Theorie der topologischen Raume*. C.R. (Doklady) Acad. Sci. URSS, vol. 11 (1936), pp, 55-58.

Extending Continuous Functions

Theorem. If a T_0-space $\mathcal{X}$ is a subspace of a space $\mathcal{Y}$, then any continuous function $\mathbf{F}: \mathcal{X} \to \mathcal{P}(\mathbb{N})$ can be extended to a continuous function $\mathbf{G}: \mathcal{Y} \to \mathcal{P}(\mathbb{N})$.

Note: We could say that P is an ***injective*** space.

Note: Continuous functions ***between*** subspaces of $\mathcal{P}(\mathbb{N})$ come from the continuous function on $\mathcal{P}(\mathbb{N})$ into itself.

For proofs see:

Reference: Martín Hötzel Escardó. *Properly injective spaces and function spaces*. Topology and its Applications, vol. 89 (1998), pp. 75-120.

Embedding Algebraic Lattices

Theorem. Every countably based algebraic lattice $\mathcal{L}$ is isomorphic to a ***sub-algebraic lattice*** of $\mathcal{P}(\mathbb{N})$.

Proof Sketch: Let the non-zero finite elements of $\mathcal{L}$ be $\{ e_n \mid n \in \mathbb{N} \}$.
Define $\varepsilon : \mathcal{L} \to \mathcal{P}(\mathbb{N})$ by $\varepsilon(x) = \{ n \in \mathbb{N} \mid e_n \leqq x \}$.
By the properties of algebraic lattices, this mapping is one-one onto a sublattice of $\mathcal{P}(\mathbb{N})$.

Comment: ***Finite*** elements are often called ***compact***.

Comment: This is the topological embedding and only preserves ***intersections*** and ***directed unions***.

Reference: *Compact element.* Wikipedia, the free encyclopedia.
Note: This entry is incomplete, as is *Algebraic Lattice.*

The Space of Continuous Functions

Theorem. The space of continuous functions between two countably based algebraic lattices is again a countably based algebraic lattice under the point-wise ordering.

Theorem. The countably based algebraic lattices and continuous functions form a ***cartesian closed category***.

Theorem. The space **Cont**[$\mathcal{P}(\mathbb{N})$, $\mathcal{P}(\mathbb{N})$] of continuous functions from $\mathcal{P}(\mathbb{N})$ into itself is a ***continuous retract*** of $\mathcal{P}(\mathbb{N})$.

Corollary: $\mathcal{P}(\mathbb{N})$ can become a $\boldsymbol{\lambda}$-calculus model.

Enumeration Operators

Definitions. (1) ***Pairing:*** $(n,m) = 2^n(2m+1)-1$.

(2) ***Sequence numbers:*** $\langle\rangle = 0$ and

$\langle n_0, n_1, \ldots, n_{k-1}, n_k\rangle = (\langle n_0, n_1, \ldots, n_{k-1}\rangle, n_k)+1$.

(3) ***Sets:*** $\mathbf{set}(0) = \varnothing$ and $\mathbf{set}((n,m)+1) = \mathbf{set}(n) \cup \{m\}$.

(4) ***Kleene star:*** $X^* = \{n \mid \mathbf{set}(n) \subseteq X\}$, for sets $X \subseteq \mathbb{N}$.

Note: X^* **consists of all the sequence numbers representing all the finite subsets of the set** X**.**

Definition. An ***enumeration operator*** $F : \mathcal{P}(\mathbb{N}) \longrightarrow \mathcal{P}(\mathbb{N})$ is a mapping ***determined by*** a given subset $F \subseteq \mathbb{N}$ by the formula:

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

Exercise: Show that the enumeration operators on $\mathcal{P}(\mathbb{N})$ are ***exactly*** the continuous functions.

The λ-Calculus Model

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

Abstraction: $\boldsymbol{\lambda}\mathtt{X.[..X..] = \{(n,m) \mid m \in [..}\textbf{set}\mathtt{(n)..]\},}$
where $\mathtt{X} \mapsto \mathtt{[..X..]}$ is ***continuous***.

Note 1: Application is a continuous function of ***two*** variables.

Note 2: If $\boldsymbol{F}\mathtt{(X)}$ is continuous, then $\boldsymbol{\lambda}\mathtt{X.}\boldsymbol{F}\mathtt{(X)}$ is the ***largest*** *set* $\mathtt{F}$ where for all sets $\mathtt{T}$, we have $\mathtt{F(T)} = \boldsymbol{F}\mathtt{(T)}$.

Warning: Generally we only have $\mathtt{F} \subseteq \boldsymbol{\lambda}\mathtt{X.F(X)}$.

Note 3: If the function $\boldsymbol{F}\mathtt{(X,Y)}$ is continuous, then the abstraction term $\boldsymbol{\lambda}\mathtt{X.}\boldsymbol{F}\mathtt{(X,Y)}$ is continuous in the ***other variable***.

Some Historical Background

The model could easily have been defined in 1957!!

John R. Myhill: Born: 11 August 1923, Birmingham, UK
Died: 15 February 1987, Buffalo, NY

John Shepherdson: Born: 7 June 1926, Huddersfield, UK
Died: 8 January 2015, Bristol, UK

Hartley Rogers, Jr.: Born: 6 July, 1926, Buffalo, NY
Died: 17 July, 2015, Waltham, MA

- 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 earlier results are presented in an abstract in **The 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.

Pairing and Relations

Recall. ***Pairing functions*** for sets in $\mathcal{P}(\mathbb{N})$ can be defined by these enumeration operators:

$$\textbf{Pair}(X)(Y)=\{2n \mid n\in X\}\cup\{2m+1 \mid m\in Y\}$$

$$\textbf{Fst}(Z)=\{n \mid 2n\in Z\} \text{ and } \textbf{Snd}(Z)=\{m \mid 2m+1\in Z\}.$$

Note: Under this definition we have $\mathcal{P}(\mathbb{N}) = \mathcal{P}(\mathbb{N})\times\mathcal{P}(\mathbb{N})$ in the category of topological spaces. From time to time we may write $\textbf{Pair}(X)(Y)=(X,Y)$ to save space.

Convention. Every subset of $\mathcal{P}(\mathbb{N})$ can be regarded as a ***binary relation***, where for all $\mathcal{A}\subseteq\mathcal{P}(\mathbb{N})$ we write $X\ \mathcal{A}\ Y$ iff $\textbf{Pair}(X)(Y)\in\mathcal{A}$.

Partial Equivalences as Types

Definition. By a ***type*** over $\mathcal{P}(\mathbb{N})$ we shall understand

a ***partial equivalence relation*** $\mathcal{A} \subseteq \mathcal{P}(\mathbb{N})$

where, for all $\mathtt{X},\mathtt{Y},\mathtt{Z} \in \mathcal{P}(\mathbb{N})$, we have

- $\mathtt{X}\ \mathcal{A}\ \mathtt{Y}$ implies $\mathtt{Y}\ \mathcal{A}\ \mathtt{X}$, and
- $\mathtt{X}\ \mathcal{A}\ \mathtt{Y}$ and $\mathtt{Y}\ \mathcal{A}\ \mathtt{Z}$ imply $\mathtt{X}\ \mathcal{A}\ \mathtt{Z}$.

We also write $\mathtt{X}:\mathcal{A}$ iff $\mathtt{X}\ \mathcal{A}\ \mathtt{X}$,

and say that $\mathcal{A}$ ***types*** $\mathtt{X}$.

Note: **Think of a type as a quotient space of a subspace of $\mathcal{P}(\mathbb{N})$.**

Taking quotients is a very common mathematical construction.

It is, however, better NOT to pass to using equivalence classes as points in order to make it easier to employ our λ-calculus.

The Category of Types

Definition. The ***exponentiation*** of types $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}(\mathbb{N})$ is defined as that relation where

$$\mathtt{F}(\mathcal{A} \to \mathcal{B})\mathtt{G} \text{ iff } \forall \mathtt{X},\mathtt{Y}.\ \mathtt{X}\ \mathcal{A}\ \mathtt{Y} \text{ implies } \mathtt{F(X)}\ \mathcal{B}\ \mathtt{G(Y)}.$$

Exercise: Show $(\mathcal{A} \to \mathcal{B})$ is a partial equivalence relation.

Exercise: Show $\mathtt{F}:\mathcal{A} \to \mathcal{B}$ implies $\forall \mathtt{X}:\mathcal{A}.\ \mathtt{F(X)}:\mathcal{B}$.

Exercise: Show $(\lambda \mathtt{X}.\lambda \mathtt{Y}.\mathtt{X}) : \mathcal{A} \to (\mathcal{B} \to \mathcal{A})$ for any types $\mathcal{A}$ and $\mathcal{B}$.

Theorem: The types form a ***category*** expanding the category of subspaces.

Products and Sums of Types

Definition. The ***product*** of two types $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}(\mathbb{N})$ is defined as that relation where

$$X(\mathcal{A} \times \mathcal{B})Y \text{ iff } \mathbf{Fst}(X)\mathcal{A}\,\mathbf{Fst}(Y) \text{ and } \mathbf{Snd}(X)\,\mathcal{B}\,\mathbf{Snd}(Y).$$

Exercise: The product of two types is again a type, and we have

$$X:(\mathcal{A} \times \mathcal{B}) \text{ iff } \mathbf{Fst}(X):\mathcal{A} \text{ and } \mathbf{Snd}(X):\mathcal{B}\ .$$

Definition. The ***sum*** of two types $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}(\mathbb{N})$ is defined as that relation where $X(\mathcal{A}+\mathcal{B})Y$ iff

either $\exists X_0, Y_0 [X_0 \mathcal{A} Y_0 \;\&\; X = (\{0\}, X_0) \;\&\; Y = (\{0\}, Y_0)]$
or $\exists X_1, Y_1 [X_1 \mathcal{B} Y_1 \;\&\; X = (\{1\}, X_1) \;\&\; Y = (\{1\}, Y_1)]$.

Exercise: The sum of two types is again a type, and we have

$X:(\mathcal{A} + \mathcal{B})$ iff either $\mathbf{Fst}(X) = \{0\} \;\&\; \mathbf{Snd}(X):\mathcal{A}$
or $\mathbf{Fst}(X) = \{1\} \;\&\; \mathbf{Snd}(X):\mathcal{B}$.

Note: Types form a (bi) cartesian closed category — whereas the topological category of subspaces does not.

Dependent Products & Sums

With simple logical definitions in $\mathcal{P}(\mathcal{P}(\mathbb{N}))$, one can study dependent type theory with the kind of rules used by Martin-Löf and de Bruijn.

But note that the λ-terms are type free.

With further simple logical definitions in $\mathcal{P}(\mathcal{P}(\mathcal{P}(\mathbb{N})))$, one can study functors between types as well as systems of types.

And this also leads to . . .

Polymorphic Types

Theorem. The class of all types is a ***complete lattice***, because it is closed under ***arbitrary intersections***.

Exercise: Show that $\lambda \mathtt{X}.\lambda \mathtt{Y}.(\mathtt{X},\mathtt{Y}) : \bigcap_{\mathcal{A},\mathcal{B}} (\mathcal{A} \to (\mathcal{B} \to (\mathcal{A} \times \mathcal{B})))$

Definition. The ***Scott numerals*** (1963) in λ-calculus are:

$\underline{0} = \lambda \mathtt{X}.\lambda \mathtt{F}.\mathtt{X}$, $\underline{1} = \lambda \mathtt{X}.\lambda \mathtt{F}.\mathtt{F}(\underline{0})$, $\underline{2} = \lambda \mathtt{X}.\lambda \mathtt{F}.\mathtt{F}(\underline{1})$, etc., and

$\underline{\text{succ}} = \lambda \mathtt{Y}.\lambda \mathtt{X}.\lambda \mathtt{F}.\mathtt{F}(\mathtt{Y})$, and

$\underline{\text{pred}} = \lambda \mathtt{Y}.\mathtt{Y}(\underline{0})(\lambda \mathtt{X}.\mathtt{X})$.

Note: Any ***monotone*** function on types has a ***least & greatest fixed point***.

Exercise: Show $\mathcal{S}cott = \bigcap_{\mathcal{A}} (\mathcal{A} \to ((\mathcal{S}cott \to \mathcal{A}) \to \mathcal{A}))$ types these numerals.

Some Closing Observations

- Enumeration operators over $\mathcal{P}(\mathbb{N})$ model λ-calculus and are characterized by a simple topology.
- The large category of types over $\mathcal{P}(\mathbb{N})$ inherits much topology.
- λ-calculus over $\mathcal{P}(\mathbb{N})$ plus the arithmetic combinators provides a basic notion of computability.
- The category of types over $\mathcal{P}(\mathbb{N})$ thus also inherits aspects of computability.
- Polymorphism for types then gives an abstract foundation for defining inductive and co-inductive data structures.
- Propositions-as-types then will enforce using constructive logic.

The model can in this way function as a laboratory for exploring these ideas in a very concrete fashion — which is also open to using Computer-Based Theorem Proving.