What is Explicit Mathematics?

Dana S. Scott, FBA, FNAS, FAAAS
University Professor Emeritus
Carnegie Mellon University
Visiting Scholar
University of California, Berkeley

CMU Logic Colloquium
28 October 2017

To the memory of my great and inspiring friend
Solomon Feferman (1928 – 2016)

Check out: `http://math.stanford.edu/~feferman`

Solomon Feferman's
Operationally Based Axiomatic Programs

- The Explicit Mathematics Program
- The Unfolding Program
- A Logic for Mathematical Practice
- Operational Set Theory (OST)

- **Aim:** To have a straightforward and principled transfer of the notions of indescribable cardinals from set theory to admissible ordinals.
- **Problem:** The approach leaves open the question as to what is the proper analogue for admissible ordinals — if any — of a cardinal κ being Π^m_n-indescribable for m > 1.

"Advances in Proof Theory: In honor of Gerhard Jäger's 60th birthday"
Lecture at Bern, 13–14 December 2013.

On Mathematical Practice

- Most of current mathematics is based on ***non-constructive set-theoretical principles***, but in fact strikingly little of what is implicit in those principles is actually used (except, of course, in set theory itself).
- For example, the bulk of mathematical analysis may be developed within the ***finite type structure*** over the natural numbers N — and indeed within type level ***three***.
- ***Transfinite types*** appear in set theory by transfinite iteration of the powerset operation. But where such iteration is used at all in analysis, it is applied only to operations within a given type.
- Practice may be regarded as deficient in that it does not pursue the ***potential resources*** of transfinite types; this view is borne out by recent results concerning determinateness of Borel games (cf. the results of Donald A. Martin,1975).

Solomon Feferman. "Theories of finite type." In: J. Barwise (ed.), Handbook of Mathematical Logic, North-Holland, 1977, pp. 913–971.

The Role of Logic

- Viewed logically, the main existential principles within any given type **S** are ***comprehension axioms*** or ***choice axioms***.
- The former assert that for each property φ of elements of **S** there exists the set of all objects in **S** having the property φ.
- The class of properties considered may be described precisely within a formal language and, again quite strikingly, the defining properties which are actually used are of ***very low logical complexity*** (in several senses).
- This makes an ***informative logical analysis*** of practice even more feasible.

Solomon Feferman."Theories of finite type." p.914

Much more discussion can be read in that chapter.
Note that ***classical logic*** *is emphasized.*

Errett Bishop's Prolog to "Constructive Analysis" (1967)

- This book is a piece of ***constructivist propaganda*** designed to show that there does exist a satisfactory alternative (to classical mathematics). To this end, we develop a large portion of abstract analysis within a constructive framework.

- This development is carried through with ***an absolute minimum*** of philosophical prejudice concerning the nature of constructive mathematics.

- There are no ***dogmas*** to which we must conform. Our ***program*** is simple: to give ***numerical meaning*** to as much as possible of classical abstract analysis. Our ***motivation*** is the well-known scandal, exposed by Brouwer (and others) in great detail, that classical mathematics is deficient in numerical meaning.

Bishop's Book Prolog (Continued)

- The task of making analysis constructive is guided by three basic principles.
  - **First** , to make every concept ***affirmative***.
    (Even the concept of inequality is affirmative.)
  - **Second**, to avoid definitions that are not ***relevant***.
    (The concept of a pointwise continuous function is not relevant; a continuous function is one that is uniformly continuous on compact intervals.)
  - **Third**, to avoid ***pseudogenerality*** .
    (Separability hypotheses are freely employed.)
- The book thus has a threefold purpose:
  - **(1)** to present the ***constructive point of view***,
  - **(2)** to show that the constructive program can ***succeed***, and
  - **(3)** to lay a foundation for ***further work***.

These immediate ends tend to an ultimate goal to hasten the inevitable day when constructive mathematics will be the accepted norm.

Bishop's Book Prolog (Continued)

- We are not contending that idealistic mathematics is ***worthless*** from the constructive point of view.
- This would be as silly as contending that ***unrigorous mathematics*** is worthless from the classical point of view.
- Every theorem proved with idealistic methods presents a ***challenge***: to find a constructive ***version***, and to give it a constructive ***proof***.

The Revised Book:

Errett Bishop and Douglas Bridges. "Constructive Analysis." Springer-Verlag, Grundlehren der mathematischen Wissenschaften, vol. 279, 1985, xii + 477 pp. ***Softcover reprint 2011****.*

Note: *A Google Scholar search for* ***bishop bridges constructive*** *turns up a truly vast literature.*

Martin-Löf's Intuitionistic Theory of Types

- The theory of types with which we shall be concerned is intended to be a ***full scale system*** for formalizing intuitionistic mathematics as developed, for example, in the book by Bishop.
- The language of the theory is richer than the languages of traditional intuitionistic systems in permitting ***proofs*** to appear as parts of propositions so that the propositions of the theory can express properties of proofs — and not only individuals — like in first order predicate logic.
- This makes it possible to strengthen the ***axioms for existence***, disjunction, absurdity and identity.
- In the case of existence, this possibility seems first to have been indicated by ***William Howard***.

Per Martin-Löf. "An intuitionistic theory of types: Predicative part." In: Logic Colloquim '73, H. E. Rose and J. C. Shepherdson, eds., North-Holland, 1975, pp. 73-118.

B. Nordström, K. Petersson and J. M. Smith. "Martin-Löf's Type Theory." In: Handbook of Logic in Computer Science, vol. 5, Oxford University Press, 2000, pp.1-37.

The Question of Universes

- The present theory was first based on the strongly impredicative axiom that there is a ***type of all types***, in symbols, **V**∈**V**, which is at the same time a type and an object of that type.
- This axiom had to be abandoned, however, after it had been shown to lead to a ***contraction*** by Jean-Yves Girard. (And there is a related, independent result of John Reynolds.)
- The incoherence of the idea of a type of all types whatsoever made it necessary to distinguish — like in category theory — between ***small*** and ***large*** types.

✻ ✻ ✻

Gerhard Jäger "The Operational Penumbra: Some Ontological Aspects", 2017, in preparation.

Informally speaking, universes play a similar role in explicit mathematics as admissible sets in weak set theory and sets $V_\varkappa$ (for regular cardinals) in full classical set theory.

Russell & Church's Strict Typing

- ***All*** variables and operations must be given ***types***, as in:

$$\boldsymbol{\lambda}\, \mathtt{x}{:}\mathcal{A}.\mathtt{F(x)} : \mathcal{A} \to \mathcal{B}.$$

Suppose $\mathtt{F} = \boldsymbol{\lambda}\,\mathtt{x}{:}\mathbb{R}.(((\mathtt{x}\cdot_{\mathbb{R}}\mathtt{x})+_{\mathbb{R}}\mathtt{x})+_{\mathbb{R}}1_{\mathbb{R}})$ and so $\mathtt{F} : \mathbb{R} \to \mathbb{R}$,

Then $\mathtt{F(5)} = 31$ and $\mathtt{F(-1)} = 1$ but

$\mathtt{F(\mathbf{i})}$ = ***undefined*** and $\mathtt{F(\mathbf{j})}$ = **??**

Suppose $\mathtt{F} = \boldsymbol{\lambda}\,\mathtt{x}{:}\,\mathbb{H}.(((\mathtt{x}\cdot_{\mathbb{H}}\mathtt{x})+_{\mathbb{H}}\mathtt{x})+_{\mathbb{H}}1_{\mathbb{H}})$ and so $\mathtt{F} : \mathbb{H} \to \mathbb{H}$,

Then $\mathtt{F(5)} = 31$ and $\mathtt{F(-1)} = 1$ and also

$\mathtt{F(\mathbf{i})} = \mathbf{i}$ and $\mathtt{F(\mathbf{j})} = \mathbf{j}$,

because $\mathbb{R} \subseteq \mathbb{C} \subseteq \mathbb{H}$.

($\mathbb{R}$ = **reals**, $\mathbb{C}$ = **complexes**, $\mathbb{H}$ = **quaternions**)

Curry's Polymorphic Typing

- Variables are not given ***types***, as in $\boldsymbol{\lambda}\mathtt{x.F(x)} : \mathcal{A} \to \mathcal{B}$, and we have to take care that $\mathtt{F}$ ***respects*** types $\mathcal{A}$ and $\mathcal{B}$.
- And it may turn out that also $\boldsymbol{\lambda}\mathtt{x.F(x)} : \mathcal{C} \to \mathcal{D}$, where $\mathcal{C}$ and $\mathcal{D}$ are quite ***different*** types. ***Prime example***: $\boldsymbol{\lambda}\mathtt{x.x} : \mathcal{A} \to \mathcal{A}$.

This was the approach in Martin-Löf's original presentation, and I was very puzzled as to how UNTYPED lambda expressions were expected to know how to BEHAVE with respect to different types of arguments.

As Martin-Löf showed, however, the formal theory was sound, but for me the SEMANTICS seemed questionable. But we shall now look at a specific MODEL.

Axiomatizing λ-Calculus

Definition. λ-calculus — as a formal theory — has rules for the ***explicit definition*** of functions *via* well known equational rules and axioms:

α-conversion

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

β-conversion

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

η-conversion

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

NOTE: The third axiom will be dropped in favor of a theory employing properties of a partial ordering.

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.

Using Gödel Numbering

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

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

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

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

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

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

The Powerset of the Integers

(1) The powerset $\mathcal{P}(\mathbb{N}) = \{\, X \mid X \subseteq \mathbb{N} \,\}$ is a ***topological space*** with the sets $\mathcal{U}_n = \{\, X \mid n \in X^* \}$ as a ***basis*** for the topology.

(2) Functions $\Phi : \mathcal{P}(\mathbb{N})^n \longrightarrow \mathcal{P}(\mathbb{N})$ are ***continuous*** iff, for all $m \in \mathbb{N}$, we have $m \in \Phi(X_0, X_1, \ldots, X_{n-1})^*$ iff there are $k_i \in X_i^*$ for each of the $i<n$, such that $m \in \Phi(\mathbf{set}(k_0), \mathbf{set}(k_1), \ldots, \mathbf{set}(k_{n-1}))$.

(3) The application operation $F(X)$, defined below, is continuous as a function of ***two*** variables.

Note: These basic facts are very easy to prove, and we will find that the powerset is a very rich space.

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 a open of $\mathcal{X}$ is an open of the subspace.

Therefore, ε is a homeomorphism to a subspace. Q.E.D.

Moreover: Continuous functions ***between*** subspaces come from those of $\mathcal{P}(\mathbb{N})$.

Note: This embedding theorem is originally due to:

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

Enumeration Operators Given as Sets

Application:

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

Abstraction:

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

$$\{0\} \cup \{ (n,m) \mid m \in [\ldots \textbf{set}(n) \ldots] \}$$

- ***Enumeration operators*** are the continuous functions on the powerset.
- If the function $\Phi(X_0, X_1, \ldots, X_{n-1})$ is continuous, then the abstraction term $\lambda X_0 . \Phi(X_0, X_1, \ldots, X_{n-1})$ is continuous in all of the ***remaining variables***.
- If $\Phi(X)$ is continuous, then $\lambda X. \Phi(X)$ is the ***largest set*** F such that for all sets T, we have $F(T) = \Phi(T)$. And, therefore, generally $F \subseteq \lambda X.F(X)$.

Enumeration Operators form the Model

This model clearly satisfies the rules of α, β-conversion (but not η) and 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.

Some Lambda Properties & Computability

Theorem. For all sets of integers F and G we have:

$$\lambda X.F(X) \subseteq \lambda X.G(X) \text{ iff } \forall X.F(X) \subseteq G(X),$$

$$\lambda X.(F(X) \cap G(X)) = \lambda X.F(X) \cap \lambda X.G(X),$$

and

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

Definition. A continuous operator $\Phi(X_0, X_1, \ldots, X_{n-1})$ is ***computable*** iff in the model this set is **RE**:

$$F = \lambda X_0 \lambda X_1 \ldots \lambda X_{n-1}.\Phi(X_0, X_1, \ldots, X_{n-1}).$$

Fixed Points and Recursion

Three Basic Theorems.

- All pure λ-terms define ***computable*** operators.
- If $\Phi(X)$ is continuous and if we let $\nabla = \lambda X.\Phi(X(X))$, then the set $P = \nabla(\nabla)$ is the ***least fixed point*** of Φ.
- The least fixed point of a ***computable*** operator is computable.

A Principal Theorem. These computable operators:

$$\textbf{Succ}(X)=\{n+1 \mid n \in X\},$$

$$\textbf{Pred}(X)=\{n \mid n+1 \in X\}, \text{ and}$$

$$\textbf{Test}(Z)(X)(Y)= \{n \in X \mid 0 \in Z\} \cup \{m \in Y \mid \exists k.k+1 \in Z\},$$

together with λ-calculus, suffice for defining all RE sets.

Pairing and Relations

Definition. ***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. However, the isomorphisms $\mathcal{P}(\mathbb{N}) \cong \mathcal{P}(\mathbb{N}) + \mathcal{P}(\mathbb{N})$ and $\mathcal{P}(\mathbb{N}) \cong \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ are not true, and they need more discussion.

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

Partial Equivalences as Types

Definition. By a ***type*** over $\mathcal{P}(\mathbb{N})$ we understand a ***partial equivalence relation*** $\mathcal{A} \subseteq \mathcal{P}(\mathbb{N})$ where, for all $X, Y, Z \in \mathcal{P}(\mathbb{N})$, we have

$X \mathcal{A} Y$ implies $Y \mathcal{A} X$, and

$X \mathcal{A} Y$ and $Y \mathcal{A} Z$ imply $X \mathcal{A} Z$.

Additionally we write $X : \mathcal{A}$ iff $X \mathcal{A} X$.

Note: **It is better NOT to pass to equivalence classes and the corresponding quotient spaces. But we can THINK in those terms if we like, as this is a very common mathematical construction.**

Definition For subspaces $\mathcal{X}\ .\ \mathcal{P}(\mathbb{N})$, we write

$$[\mathcal{X}] = \{(X, X) \mid X \in \mathcal{X}\},$$

so that we may regard ***subspaces as types*.**

The Category of Types

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

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

Theorem. The exponentiation (= function space) of two types is again a type, and we have

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

Theorem. Types do form a category — expanding the topological category of subspaces.

Definition. For each type $\mathcal{A}$ the ***identity type*** on $\mathcal{A}$ is defined as that relation such that $\mathtt{Z}(\mathtt{X} \equiv_{\mathcal{A}} \mathtt{Y})\mathtt{W}$ iff $\mathtt{Z}\ \mathcal{A}\ \mathtt{X}\ \mathcal{A}\ \mathtt{Y}\ \mathcal{A}\ \mathtt{W}$.

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

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

Theorem. 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 $\exists X_1, Y_1 [X_1 \mathcal{B} Y_1 \;\&\; X = (\{1\}, X_1) \;\&\; Y = (\{1\}, Y_1)]$.

Theorem. 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 $\mathbf{Fst}(X) = \{1\} \;\&\; \mathbf{Snd}(X): \mathcal{B}$.

Isomorphism of Types

Definition. Two types $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}(\mathbb{N})$ are ***isomorphic***, in symbols $\mathcal{A} \cong \mathcal{B}$, provided there are mappings

$$\mathtt{F}:\mathcal{A} \to \mathcal{B} \text{ and } \mathtt{G}:\mathcal{B} \to \mathcal{A} \text{ where}$$

$$\forall \mathtt{X}:\mathcal{A}.\ \mathtt{X}\ \mathcal{A}\ \mathtt{G(F(X))} \text{ and } \forall \mathtt{Y}:\mathcal{B}.\ \mathtt{Y}\ \mathcal{B}\ \mathtt{F(G(Y))}.$$

Theorem. If types $\mathcal{A}_0 \cong \mathcal{B}_0$ and $\mathcal{A}_1 \cong \mathcal{B}_1$, then

$$(\mathcal{A}_0 \times \mathcal{A}_1) \cong (\mathcal{B}_0 \times \mathcal{B}_1), \text{ and}$$

$$(\mathcal{A}_0 + \mathcal{A}_1) \cong (\mathcal{B}_0 + \mathcal{B}_1), \text{ and}$$

$$(\mathcal{A}_0 \to \mathcal{A}_1) \cong (\mathcal{B}_0 \to \mathcal{B}_1).$$

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

Checking Isomorphisms

Theorem. We have these ***algebraic laws*** for all types $\mathcal{A}, \mathcal{B}, \mathcal{C}$:

$$(\mathcal{A} \times \mathcal{B}) \cong (\mathcal{B} \times \mathcal{A}),$$

$$(\mathcal{A} + \mathcal{B}) \cong (\mathcal{B} + \mathcal{A}),$$

$$((\mathcal{A} \times \mathcal{B}) \times \mathcal{C}) \cong (\mathcal{A} \times (\mathcal{B} \times \mathcal{C})),$$

$$((\mathcal{A} + \mathcal{B}) + \mathcal{C}) \cong (\mathcal{A} + (\mathcal{B} + \mathcal{C})),$$

$$(\mathcal{A} \times (\mathcal{B} + \mathcal{C})) \cong ((\mathcal{A} \times \mathcal{B}) + (\mathcal{A} \times \mathcal{C})),$$

$$((\mathcal{A} \times \mathcal{B}) \rightarrow \mathcal{C}) \cong (\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C})),$$

$$(\mathcal{A} \rightarrow (\mathcal{B} \times \mathcal{C})) \cong ((\mathcal{A} \rightarrow \mathcal{B}) \times (\mathcal{A} \rightarrow \mathcal{C})), \text{ and}$$

$$((\mathcal{A} + \mathcal{B}) \rightarrow \mathcal{C}) \cong ((\mathcal{A} \rightarrow \mathcal{C}) \times (\mathcal{B} \rightarrow \mathcal{C})).$$

Roberto Di Cosmo. ***"Isomorphisms of Types: from λ-calculus to information retrieval and language design."*** Progress in Theoretical Computer Science, Birkhäuser, 1995, 235 pp. Softcover reprint 2011.

Dependent Products

Definition. Let $\mathcal{T}$ be the class of all types.

For each $\mathcal{A} \in \mathcal{T}$, an ***$\mathcal{A}$-indexed family of types***

is a function $\mathcal{B}: \mathcal{P}(\mathbb{N}) \to \mathcal{T}$, such that

$\forall X_0, X_1.\ X_0\ \mathcal{A}\ X_1$ implies $\mathcal{B}(X_0) = \mathcal{B}(X_1)$.

In words: Equivalent parameters produce equivalent types.

Definition. The ***dependent product*** of an $\mathcal{A}$-indexed family of types, $\mathcal{B}$, is this equivalence relation:

$F_0 (\prod X{:}\mathcal{A}.\mathcal{B}(X)) F_1$ iff

$\forall X_0, X_1.\ X_0\ \mathcal{A}\ X_1$ implies $F_0(X_0)\ \mathcal{B}(X_0)\ F_1(X_1)$.

Note: $(\mathcal{A} \to \mathcal{B}) = \prod X{:}\mathcal{A}.\mathcal{B}$.

Dependent Sums

Definition. The ***dependent sum*** of an $\mathcal{A}$-indexed family of types, $\mathcal{B}$, is this equivalence relation:

$$Z_0(\sum X:\mathcal{A}.\mathcal{B}(X))Z_1 \text{ iff}$$

$$\exists X_0,Y_0,X_1,Y_1[X_0\mathcal{A}X_1 \ \& \ Y_0\mathcal{B}(X_0)Y_1 \ \&$$

$$Z_0 = (X_0,Y_0) \ \& \ Z_1 = (X_1,Y_1)]$$

Theorem. The dependent products and dependent sums of indexed families of types are always again types.

Note: $(\mathcal{A}\times\mathcal{B}) = \sum X:\mathcal{A}.\mathcal{B}$.

Systems of Dependent Types

Definition. We say that $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ form a ***system of dependent types*** iff

- $\forall X_0, X_1 . [X_0 \, \mathcal{A} \, X_1 \Rightarrow \mathcal{B}(X_0) = \mathcal{B}(X_1)]$, and
- $\forall X_0, X_1, Y_0, Y_1 . [X_0 \, \mathcal{A} \, X_1 \; \& \; Y_0 \, \mathcal{B}(X_0) \, Y_1 \Rightarrow \mathcal{C}(X_0, Y_0) = \mathcal{C}(X_1, Y_1)]$, and
- $\forall X_0, X_1, Y_0, Y_1, Z_0, Z_1 . [X_0 \, \mathcal{A} \, X_1 \; \& \; Y_0 \, \mathcal{B}(X_0) \, Y_1 \; \& \; Z_0 \, \mathcal{C}(X_0, Y_0) \, Z_1 \Rightarrow \mathcal{D}(X_0, Y_0, Z_0) = \mathcal{D}(X_1, Y_1, Z_1)]$,

provided that $\mathcal{A} \in \mathcal{T}$, and $\mathcal{B}, \mathcal{C}, \mathcal{D}$ are functions on $\mathcal{P}(\mathbb{N})$ to $\mathcal{T}$ of the indicated number of arguments.

Theorem. Under the above assumptions on the system $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$, we will always have

$$\prod X : \mathcal{A} . \sum Y : \mathcal{B}(X) . \prod Z : \mathcal{C}(X, Y) . \; \mathcal{D}(X, Y, Z) \in \mathcal{T}.$$

Polymorphic Types

Theorem. The class $\mathcal{T}$ of all types is a ***complete lattice***, because it is closed under ***arbitrary intersections***.

Example: $\lambda X.\lambda Y.(X,Y) : \bigcap_{\mathcal{A},\mathcal{B}} (\mathcal{A} \to (\mathcal{B} \to (\mathcal{A} \times \mathcal{B})))$

Theorem. Any ***monotone*** $\Phi : \mathcal{T} \to \mathcal{T}$ has a ***least & greatest fixed point***.

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

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

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

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

Example: $\mathcal{Scott} = \bigcap_{\mathcal{A}} (\mathcal{A} \to ((\mathcal{Scott} \to \mathcal{A}) \to \mathcal{A}))$ **types the numerals.**

Propositions as Types

Definition. Every type $\mathcal{P} \in \mathcal{T}$ can be regarded as a ***proposition***, where ***asserting*** (or ***proving*** $\mathcal{P}$) means finding ***evidence*** $\mathtt{E}:\mathcal{P}$.

Convention: Under this interpretation of logic,

asserting $(\mathcal{P} \times \mathcal{Q})$ means asserting a ***conjunction***,

asserting $(\mathcal{P} + \mathcal{Q})$ means asserting a ***disjunction***,

asserting $(\mathcal{P} \rightarrow \mathcal{Q})$ means asserting an ***implication***,

asserting $(\prod \mathtt{X}:\mathcal{A}.\mathcal{P}(\mathtt{X}))$ means asserting a ***universal quantification***, and

asserting $(\sum \mathtt{X}:\mathcal{A}.\mathcal{B}(\mathtt{X}))$ means asserting an ***existential quantification***.

Example: **Given** $\mathtt{F}:(\mathcal{A} \rightarrow (\mathcal{A} \rightarrow \mathcal{A}))$**, then asserting**

$$\prod \mathtt{X}:\mathcal{A}.\prod \mathtt{Y}:\mathcal{A}.\prod \mathtt{Z}:\mathcal{A}.\ \mathtt{F}(\mathtt{X})(\mathtt{F}(\mathtt{Y})(\mathtt{Z})) \equiv_{\mathcal{A}} \mathtt{F}(\mathtt{F}(\mathtt{X})(\mathtt{Y}))(\mathtt{Z})$$

is the same as asserting that $\mathtt{F}$ **is an associative binary operation.**

A Possible New Area for Application

Asymptotic Differential Algebra and
Model Theory of Transseries
by
Matthias Aschenbrenner, Lou van den Dries, and
Joris van der Hoeven

Princeton University Press, 2017, xxi + 833 pp.

Preface: We develop here the algebra and model theory of the differential field of transseries, a fascinating mathematical structure obtained by iterating a construction going back more than a century to Levi-Civita and Hahn. It was introduced about thirty years ago as an exponential ordered field by Dahn and Göring in connection with Tarski's problem on the real field with exponentiation, and independently by Écalle in his proof of the Dulac Conjecture on plane analytic vector fields.

Some Conclusions

- 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.