Hyperfunctions

Sava Krstić[1], John Launchbury[1], and Duško Pavlović[2]

[1] Oregon Graduate Institute {krstic,jl}@cse.ogi.edu
[2] Kestrel Institute dusko@kestrel.edu

Hyperfunctions from A to B are elements of the infinitely nested function type

$$[A, B] = (((\cdots \Rightarrow A) \Rightarrow B) \Rightarrow A) \Rightarrow B.$$

Such types, for cardinality reasons, do not exist in the category of sets, but we have them in any category of domains that allows standard (Plotkin-Smyth) construction of canonical solutions of domain equations [AJ94]. In some programming languages, types $[A, B]$ can be explicitly declared, and [LKS00] contains examples of computing with hyperfunctions in *Haskell.*

Coalgebraic, rather than domain-theoretic framework seems more appropriate for a study of hyperfunctions, so let us assume that $\mathcal{C}$ is a cartesian closed category such that, for every objects A, B, the functor $H_{A,B}X = (X \Rightarrow A) \Rightarrow B$ has a final coalgebra $[A, B]$. One can then define units $\theta_A : I \longrightarrow [A, A]$ and composition $(\#) : [B, C] \times [A, B] \longrightarrow [A, C]$ as suitable anamorphisms, and then prove that coalgebras $[A, B]$ are hom-objects of a $\mathcal{C}$-enriched *category of hyperfunctions* $\mathcal{H}$. We have described this in [KLP01], where a general construction of coalgebra enriched categories is given, motivated mostly by applications in the semantics of processes. (Categories of resumptions and hyperfunctions are the simplest examples of the construction.)

While the semantic relevance and programming potential of hyperfunctions are still unclear, their conceptual simplicity and mathematical attraction call for a further investigation. In what follows we summarize what we have discovered so far.

The first thing to look at is the relationship between $[A, B] \Rightarrow A$ and $[B, A]$. One would expect an isomorphism, but it does not follow automatically. Instead, there is only a map in one direction

$$\phi_{AB} : ([B, A] \Rightarrow B) \longrightarrow [A, B]$$

—the anamorphism corresponding to a simple $H_{A,B}$-coalgebra structure on $[B, A] \Rightarrow B$ [KLP01]. To proceed, we need to assume that the maps ϕ_{AB} are isomorphisms. For convenience, let us assume also that the category $\mathcal{C}$ is well-pointed, so that we can identify every object of $\mathcal{C}$ with the set of its global elements. These assumptions seem sufficient to derive just about everything we can prove about hyperfunctions in a category of domains. How close $\mathcal{C}$, with the conditions we imposed on it, is to a category of domains is not clear at the moment. For example, we do have a restricted form of algebraic compactness:

Theorem 1. $[A, B]$ *is both an initial algebra and a final coalgebra for the functor* $H_{A,B}$.

Most of the algebraic structure in the category $\mathcal{H}$ of hyperfunctions can be conveniently expressed in terms of the *hyperfunction application operator*

$$(\cdot)\colon [A,B] \times [B,A] \longrightarrow B$$

defined just by transposing ϕ_{AB}. For example, hyperfunction units θ_A and composition $\#$, originally defined as anamorphisms, are characterized by equations

$$(x \# y) \cdot z = x \cdot (y \# z)$$
$$\theta_A \cdot x = x \cdot \theta_A$$

There are then maps $\mathsf{lift}\colon (A \Rightarrow B) \longrightarrow [A,B]$, originally also defined as anamorphisms, that allow us to regard ordinary functions as hyperfunctions [KLP01]. One can show that $\widetilde{f} = \mathsf{lift}(f)$ is characterized by another simple equation

$$\widetilde{f} \cdot x = f(x \cdot \widetilde{f}).$$

The hyperfunction unit θ_A is, of course, the lift of id_A. Acting on itself, it defines the *bottom element* $\bot_A = \theta_A \cdot \theta_A$. More generally, acting on lifts of ordinary functions, it defines a fixpoint operator.

Theorem 2. *The equation*

$$Y(f) = \widetilde{f} \cdot \theta$$

defines a uniform fixpoint operator on $\mathcal{C}$.

Thus, the existence of hyperfunction types implies the existence of a nicely behaved fixpoint operator. Recall that *uniform* here means that $Y(g) = h(Y(f))$, for any $f\colon A \longrightarrow A$, $g\colon B \longrightarrow B$, and $h\colon A \longrightarrow B$ such that h is strict (i.e. $h(\bot_A) = \bot_B$) and $g \circ h = h \circ f$ [SP00].

Hyperfunctions can be approximated by ordinary functions and that fact leads to an interesting induction principle. Define first $[A,B]_n$ inductively by $[A,B]_0 = I$ and $[A,B]_{n+1} = ([B,A]_n \Rightarrow B)$. Then we have "embedding-projection" pairs

$$\begin{tikzcd} {[A,B]_n} \arrow[r, shift left, "\mathsf{lift}_n^{AB}"] & {[A,B]} \arrow[l, shift left, "\mathsf{proj}_n^{AB}"] \end{tikzcd}$$

defined inductively by

$$\mathsf{lift}_0 = \bot \qquad\qquad \mathsf{proj}_0 = \,!$$
$$\mathsf{lift}_{n+1} = \phi \circ (\mathsf{proj}_n \Rightarrow B) \qquad \mathsf{proj}_{n+1} = (\mathsf{lift}_n \Rightarrow B) \circ \psi,$$

where ϕ, ψ is the isomorphism pair

$$\begin{tikzcd} {[B,A] \Rightarrow B} \arrow[r, shift left, "\phi"] & {[A,B].} \arrow[l, shift left, "\psi"] \end{tikzcd}$$

Let $e_n^{AB} \colon [A, B] \longrightarrow [A, B]$ be the idempotent $\mathsf{lift}_n \circ \mathsf{proj}_n$ and let also u_n be the shorthand for $e_n(u)$. The image of e_n is a copy of $[A, B]_n$ inside $[A, B]$ and u_n is the approximation of u that resides in that image.

Theorem 3. *(a) If $u, v \in [A, B]$ and $u_n = v_n$ for every n, then $u = v$.*
(b) For every $u \in [A, B]$ and $v \in [B, A]$ and every $n \geq 0$,

$$u_{n+1} \cdot v = u \cdot v_n.$$

This is a restatement in our coalgebraic setting of the *minimal invariant property* of canonical solutions to domain equations; see [Pit96,AJ94,AC98].

Theorem 3 is the basis for inductive proofs of hyperfunction equalities. It logically comes before Theorem 2 and equations characterizing θ and $\#$, and is used to derive them.

The most common inductive pattern of reasoning about hyperfunctions is the following.

Theorem 4. ***(Hyperfunction Induction Principle)*** *To prove the equality of hyperfunctions x and y, it suffices to check that, for every z, the equality $x \cdot z = y \cdot z$ follows from $z \cdot x = z \cdot y$.*

We close with a result about the structure of hyperfunction categories; it establishes hyperfunctions as a "notion of computation" in the sense of [PR97]. The proof is a substantial exercise in the just stated induction principle.

Theorem 5. *The category $\mathcal{H}$ of hyperfunctions is symmetric premonoidal and the functor $\mathsf{lift} \colon \mathcal{C} \longrightarrow \mathcal{H}$ is strict premonoidal.*

References

[AC98] R. M. Amadio and P.-L. Curien. *Domains and Lambda-Calculi.* Cambridge University Press, 1998.

[AJ94] S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science, volume 3.* Clarendon Press, 1994.

[KLP01] S. Krstić, J. Launchbury, and D. Pavlović. Categories of processes enriched in final coalgebras. In F. Honsel and M. Miculan, editors, *Foundations of Software Science and Computation Structures (FOSSACS 2001)*, volume 2030 of *Lecture Notes in Computer Science*. Springer, 2001.

[LKS00] J. Launchbury, S. Krstić, and T. E. Sauerwein. Zip fusion with hyperfunctions. Technical report, Oregon Graduate Institute, 2000. Preprint available on `http://www.cse.ogi.edu/~krstic`.

[Pit96] A. Pitts. Relational propreties of domains. *Information and Computation*, 127:66–90, 1996.

[PR97] J. Power and E. Robinson. Premonoidal categories and notions of computation. *Mathematical Structures in Computer Science*, 7:453–468, 1997.

[SP00] A. Simpson and G. Plotkin. Complete axioms for categorical fixed-point operators. In *15th Symposium on Logic in Computer Science (LICS 2000)*. IEEE Computer Society, 2000.