Categories for Collection Types

Eugenio Moggi*
moggi@unige.it
DIBRIS, Univ. di Genova, Italy

Abstract

Collection types have been proposed as a way to capture database query languages in a type-theoretic setting. In 1998 Manes introduced the notion of collection monad on the category of sets as a suitable semantics for collection types. The canonical example of collection monad is the finite powerset monad. In order to account for the algorithmic aspects, the category of sets is unsuitable, and should be replaced with other categories, whose arrows are functions computable by *low complexity algorithms*. We propose a class of categories more general than locoses, that include standard set-theoretic models and also models based on Kalmar elementary functions (i.e. level 3 of Grzegorczyk hierarchy). In them one can

- define an analogue of the finite powerset monad and recast the notion of collection monad;
- interpret a significant fragment of Martin-Lof extensional type theory.

Summary

In the context of database languages collection types have been proposed as a common abstraction for a variety of datatypes (e.g. sets, multisets, lists, trees) for storing finite collections of elements of a certain type. Core typed calculi based on collection types, like those in [1], provide a suitable framework for database query languages that go beyond traditional relational database. These calculi have a lot in common with metalanguages for computational types [4], but there are also important differences

- Equality of collections is *decidable*, while equality of programs is *undecidable*
- Functional types and arbitrary recursion are incompatible with decidable equality

they are interpreted in categories with finite products, and collection types are interpreted by strong monads. In [3] Manes identifies certain *finitary* monads on the category $\mathbb{S}$ of sets as a suitable semantics for collection types. These monads, called collection monads, have a well-behaved notion of membership, and a collection has only finitely many members. Manes provides two characterizations for a collection monad M

1. M is induced by a *balanced* algebraic theory;
2. M is equipped with a *taut* monad map τ to the *finite powerset* monad $\mathbb{P}_f$.

The 2nd characterization can be recast in any category $\mathcal{C}$ of classes [5] with a sub-category $\mathcal{S} \subset \mathcal{C}$ of *small maps*, provided the role of $\mathbb{P}_f A$ is taken by the object $\mathbb{P}_s(A)$ of *small subobjects* of A. When $\mathcal{C}$ is $\mathbb{S}$ and small maps are the $f\colon A \to I$ s.t. $\forall i \in I.\ A_i \stackrel{\Delta}{=} \{a | f(a) = i\}$ is finite, then $\mathbb{P}_s = \mathbb{P}_f$. This sub-category of small maps is *classified* by the projection $\pi\colon F \to \mathbb{N}$, where $\mathbb{N}$ is the set of natural numbers and $F \stackrel{\Delta}{=} \{(n, i) \in \mathbb{N}^2 | i < n\}$. We propose to take as $\mathcal{C}$ the category $\mathsf{P}[C]$ induced by a sub-monoid $C \subseteq \mathbb{N} \to \mathbb{N}$ of the monoid total functions on $\mathbb{N}$ (see [2] for a more general construction), namely the sub-category of $\mathbb{S}$ s.t.

*Project MIUR-PRIN 2010-2011 provided support for this research.

$$X \in \mathsf{P}[C] \overset{\Delta}{\iff} X \subseteq \mathbb{N} \text{ and } X \xrightarrow{f} Y \text{ in } \mathsf{P}[C] \overset{\Delta}{\iff} \exists f' \in C. \forall x \in X. f(x) = f'(x)$$

The categorical properties of $\mathsf{P}[C]$ clearly depends on the properties of C. Remarkable examples of C are

1. $\mathbb{N} \to \mathbb{N}$, then $\mathsf{P}[C]$ is equivalent to the category $\mathbb{S}_\omega$ of countable sets
2. $\mathcal{R}$ the set of *primitive recursive functions*, then $\mathsf{P}[C]$ is a *locos* and $\mathbb{N}$ is a natural number object
3. $\mathcal{E}_k$ level k of Grzegorczyk hierarchy, in these cases $\mathbb{N}$ is not a natural number object in $\mathsf{P}[C]$.

When C is one of the remarkable examples (provide we start from level 3 of Grzegorczyk hierarchy), then $\mathsf{P}[C]$ is a category of classes with small maps are classified $\pi \colon F \to \mathbb{N}$, where $F \overset{\Delta}{=} \{\langle n, i\rangle | i < n\}$ and $\langle -, - \rangle$ is Cantor's encoding of $\mathbb{N}^2$ in $\mathbb{N}$. In conclusion

- the choice of C determines what are the *computable* functions on $\mathbb{N}$ (in some domain specific language)
- the choice of small maps in $\mathsf{P}[C]$ captures *finiteness*

when $\mathsf{P}[C]$ with the chosen family of small maps is a category of classes, then $\mathbb{P}_s$ is well-defined and we can define *collection monads* on $\mathsf{P}[C]$ as those monads equipped with a *taut* monad map to $\mathbb{P}_s$.

References

[1] Peter Buneman, Shamim A. Naqvi, Val Tannen, and Limsoon Wong. Principles of Programming with Complex Objects and Collection Types. *Theoretical Computer Science*, 149:3–48, 1995.

[2] John Longley. Computability structures, simulations and realizability. *Mathematical Structures in Computer Science*, FirstView:1–49, 2013.

[3] Ernest G. Manes. Implementing Collection Classes with Monads. *Mathematical Structures in Computer Science*, 8:231–276, 1998.

[4] Eugenio Moggi. Notions of Computation and Monads. *Information and Computation/information and Control*, 93:55–92, 1991.

[5] Alex K. Simpson. Elementary Axioms for Categories of Classes. In *Logic in Computer Science*, pages 77–85, 1999.