Categorical Semantics for Array-Based Programming Languages

Miles Gould
University of Glasgow

December 17, 2007

One of the most important concepts in computing is *polymorphism*, from the Greek for "many shapes": a piece of code is said to be polymorphic if it can be applied to many different types of data without change. Polymorphism allows programmers to solve problems with shorter, simpler and more maintainable code, and to re-use existing code in unforeseen situations. Polymorphism comes in several different flavours, each of which allows code to be applied across a different range of types. Some of these flavours have been better studied than others by computer scientists and mathematicians.

Particularly little studied is *shape-based* polymorphism. This can best be explained by means of an example. Consider ordinary addition of numbers: $a, b \mapsto a + b$. This allows us to define an operation of addition on arrays of numbers:

$$(a_1, \ldots, a_n) + (b_1, \ldots, b_n) := (a_1 + b_1, \ldots, a_n + b_n)$$

This also works for multi-dimensional arrays: simply add each element of the first array to the corresponding element in the second array. Addition is thus polymorphic in the shape and dimension of the arrays concerned. There's nothing special about addition in this example: we could have used any two-argument function. This idea is at the core of the design of so-called *array programming languages*, of which the progenitor is Iverson's language APL. APL and its derivatives (such as MATLAB) are widely used in the worlds of finance and engineering, but have been largely ignored by academic computer science.

Category theory can be thought of as the mathematical study of structure in the most general sense: it provides powerful tools for constructing larger mathematical objects out of small and simple ones, regardless of the precise nature of those objects. Consequently, it is the preferred language for discussing the semantic structure of programming languages. Category theory also provides us with tools to study the connections between different categorical structures - indeed, category theory starts with the assumption that the connections between objects are at least as important as the objects themselves.

The aim of this project would be to describe the shape-based polymorphism exhibited by the APL family of languages in a categorical way, and thence

- to construct a formal semantics for a simple APL-like language;
- to explore the connections between APL-style polymorphism and more conventional kinds;
- to apply the current body of knowledge on programming language semantics in this new setting, and prove new and hopefully useful results;
- to expose shape-based polymorphism to a wider academic audience;
- to expose the fruits of modern computer science to the APL community.