Abstract Computability

J.R.B. Cockett *
Department of Computer Science, University of Calgary,
Calgary, T2N 1N4, Alberta, Canada

March 7, 2016

This reports on work with Ximo Boils, Jonathan Gallagher, Pieter Hofstra, and Pavel Hrubeš.

1 Introduction

Over the last few decades a fascinating new direction in computability has been taking root.

We teach our students that the Church-Turing thesis makes all notions of computation equivalent. But what do we mean by equivalent? The usual interpretation is that the functions which can be produced – in that great leveller the category of sets – by any given (reasonable) notion of computation are the same. Yet we know that a model of computation as given by the λ-calculus is a completely a different beast from that given by a Turing machine ... and this should suggests to any sane person that the notion of equivalence used in the Church-Turing thesis is completely destroying some very significant distinctions.

Recently various other notions of computability have appeared on the stage to challenge our preconceptions. At the level of machines we now are facing the real possibility of Quantum machines: should we really view the notion of computability of these machines as being equivalent to that of classical computation? We know there are significant differences. It would be a poor theory indeed, which could not see these differences. One of my personal – perhaps less practical – favourites is the differential λ calculus: in this notion of computation all computable maps can be differentiated. Considering that the notion of differentiation is nowhere in sight in for either Turing machines or the untyped λ-calculus, it begins to get even harder to believe that this notion should be written-off as being equivalent to the usual Church-Turing notion of computation.

These examples are sent to awaken us from a slumber in which we may have dreamed that the foundations of Computer Science had been entirely settled. We should not reject the advances that have been made but rather we should just quietly acknowledge that, as our horizons and understandings expand, that it is increasingly useful to view these foundations through a telescope which has a much higher resolution than hitherto ... and which allows us to distinguish fine details which, perhaps reasonably, before we were happy to simply ignore.

The talk will present what I feel is the "modern" view of computability: in it, far from having everything nicely sewn up, almost everything is open. It is based on Turing Categories [1, 7] which are conceptually very simple but allow a huge variation in models. I am of course a category theorist so it is natural for me to organize things through a categorical lens. However, I would claim this is not just gratuitous abstraction but an important step in opening up the subject again. The categorical language allows one to see generalities, to compare models, and to modularize constructions – far from obscuring it provides a powerful and unifying telescope through which to recognize and organize distinctions.

*Partially supported by NSERC, Canada.

In the talk I will concentrate on the models of computability whose total functions belong to particular functional complexity classes (such as Linear Time, Polynomial Time, Polynomial Space, etc.) and the techniques used to produce these models. Of particular interest, I feel, are the more concrete models which arise from reformulating – categorically – what complexity theorists do anyway [3, 5]. There is here an unexpected confluence of techniques, which, on the one hand arises form how complexity theorists measure complexity and on the other how category theorists formulate partiality.

Below I include some of the basic definitions on which these ideas hinge together with some references.

2 Partiality in category theory: restriction categories

A **restriction category** [2], is a category with a restriction combinator which, to any map, assigns an endomorphism of its domain:

$$\frac{f : A \to B}{\overline{f} : A \to A}$$

This must satisfy just four identities:

$$\textbf{[R.1]}\ \overline{f}\, f = f \qquad \textbf{[R.2]}\ \overline{f}\ \overline{g} = \overline{g}\ \overline{f}$$
$$\textbf{[R.3]}\ \overline{f}\ \overline{g} = \overline{\overline{g}\ f} \qquad \textbf{[R.4]}\ f\ \overline{g} = \overline{fg}\ f$$

As $\overline{f}\ \overline{f} = \overline{\overline{f}\ f} = \overline{f}$, the restriction of a map is always an idempotent, called a **restriction idempotent**, which should be thought of as a partial identity. In a restriction category a map $f : A \to B$ is said to be **total** in case $\overline{f} = 1_A$: the total maps always form a subcategory.

Since restriction maps are idempotent and commute the restriction idempotents associated to an object A form a semilattice, $\mathcal{O}(A)$. We think of this semilattice topologically as specifying the open sets of A. Each map $f : A \to B$ then, acting like a continuous map, carries open sets backward

$$f^* : \mathcal{O}(B) \to \mathcal{O}(A); e \mapsto \overline{fe}$$

this map preserves the meet operation but does not preserve the top element: it is, therefore, a **stable** map of semilattices.

Every restriction category is order enriched: given two parallel maps $f, g : X \to Y$ in a restriction category we shall write

$$f \leq g \Leftrightarrow \overline{f}\, g = f.$$

Two parallel maps $f, g : X \to Y$ in a restriction category are **compatible**, written $f \smile g$, in case $\overline{f}\, g = \overline{g}\, f$. Intuitively this means that they agree where they are both defined.

In restriction categories there is a weaker notion of "isomorphism": a map $f : A \to B$ is a **partial isomorphism** if there is a map $f^{(-1)} : B \to A$ such that $ff^{(-1)} = \overline{f}$ and $f^{(-1)}f = \overline{f^{(-1)}}$. The notation suggests that the partial inverse is unique and this is the case. Restriction categories in which all maps are partial isomorphisms are called **inverse categories**. Inverse categories are to restriction categories what groupoids are to ordinary categories.

The completeness theorem for restriction category asserts:

Theorem 2.1 *Every restriction category occurs as a full subcategory of a partial map category.*

This is significant as is says that partiality has a purely algebraic expression. Reasoning algebraically is often much easier and more accurate than reason about partiallity by other methods.

A **cartesian restriction category** (see [8] who called them P-categories), $\mathbb{X}$, is a restriction category which has a (partial) final object and for which each pair of objects has a (partial) product.

To say that there is a **partial final object** means, explicitly, there is an object 1 which has from each object a total map $!_A : A \to 1$ so that any map $f : A \to 1$ has $f = \overline{f}\,!_A$ This, in turn, means the maps from $A \to 1$ are in bijective correspondence to the restriction idempotents on A, $\mathcal{O}(A)$. Clearly a partial terminal object becomes a true terminal object in the total map subcategory.

To say that $\mathbb{X}$ has binary (partial) products is to say that for each $A, B \in \mathbb{X}$ there is an object $A \times B$ with total maps $\pi_0 : A \times B \to A$ and $\pi_1 : A \times B \to B$ such that for any pair of maps $f : Z \to A$ and $g : Z \to B$ there is a unique map $\langle f, g\rangle : Z \to A \times B$ such that $\langle f, g\rangle \pi_0 = \overline{g}\, f$ and $\langle f, g\rangle \pi_1 = \overline{f}\, g$. This means that $\_ \times \_ : \times \times \mathbb{X} \to \mathbb{X}$ is a restriction functor and that : $\langle 1_A, 1_A\rangle = \Delta_A : A \to A \times A$ is total and a natural transformation.

Notice that the projections are lax natural rather than natural.in the sense that we have:

$$\begin{array}{ccc} A \times B & \xrightarrow{f \times g} & A' \times B' \\ {\scriptstyle \pi_0}\downarrow & \geq & \downarrow{\scriptstyle \pi_0} \\ A & \xrightarrow[f]{} & A' \end{array}$$

Also note that on the total category this means we have real products. This ensures that the cartesian structure is unique.

3 Computability in category theory: Turing categories

A Turing category may be described as a Cartesian restriction category with an special object T, called a **Turing object**, such that:

- Every object in the category is a retract of T.
- There is an **application map**, also called a **Turing morphism**, $\bullet : T \times T \to T$ such that for every (partial) map $f : T \times T \to T$ there is a *total map* $\widetilde{f} : A \to T$, called an **index** or a **code** of f, such that:

$$\begin{array}{ccc} T \times T & \xrightarrow{\bullet} & T \\ {\scriptstyle \widetilde{f} \times 1_T}\uparrow & \nearrow_{f} & \\ T \times T & & \end{array}$$

Turing categories provide a unifying formulation of abstract computability [1], and when the category has joins and is discrete, one can obtain many of the standard results from computability theory [4].

A key result for Turing categories – indeed, this essentially the completeness result for Turing categories – is that:

Theorem 3.1 *The computable maps of a partial combinator algebra in any Cartesian restriction category is a Turing category. Furthermore, up to splitting of idempotents, every Turing category arises in this manner.*

A reasonable question to ask is what classes of maps can be the total maps of a Turing category. The question is answered in [6]: what is slightly surprising is that the requirements are minimal. In fact, complexity classes such as the Linear Time and the Polynomial Time functions satisfy the requirements. This means that Turing categories can provide a unified approach both computability and complexity. In [5] concrete models which realized functional complexity classes were exhibited which will be discussed further in the talk.

References

[1] J.R.B Cockett, and P. Hofstra, *Introduction to Turing Categories.* Annals of Pure and Applied Logic, Vol. 156 (2008) 183–209.

[2] J. R. B. Cockett, and S. Lack, *Restriction Categories I: Categories of partial maps.* Theoretical Computer Science, Vol 270 (2002) 223-259.

[3] J.R.B. Cockett, and B. F. Redmond, *A Categorical Setting for Lower Complexity*. Electronic Notes in Theoretical Computer Science (ENTCS), Volume 265, 2010, Pages 277-300

[4] J.R.B. Cockett, *Categories and computability.* Lecture notes available at http://pages.cpsc.ucalgary.ca/ robin.

[5] J.R.B. Cockett, J. Diaz-Boïls, J. Gallagher, and P. Hrubeš, *Timed Sets, Functional Complexity, and Computability*. Electronic Notes in Theoretical Computer Science, Volume 286, September 2012, Pages 117-137.

[6] J.R.B. Cockett, P.J.W. Hofstra, and P. Hrubeš, *Total Maps of Turing Categories*. Electronic Notes in Theoretical Computer Science, Volume 308, 29 October 2014, Pages 129?146

[7] R.A. Di Paola, and A. Heller, *Dominical categories: recursion theory without elements.* Journal of Symbolic Logic, Vol. 52 (1987), 595-635.

[8] E. Robinson, and G. Rosolini, *Categories of partial maps.* Information and Computation, vol. 79 (1988) 94–130.