Gandy Machines Made Easy via Category Theory

Joseph Razavi* Andrea Schalk†
University of Manchester, UK

30th June 2020

Abstract

Updating work reported in a previous conference paper [10], we describe how Gandy's principles for computation are naturally described by category theoretic concepts. Category theoretic tools provide invariance properties which require sophisticated treatment in other axiomatizations, and do most of the heavy lifting when arguing that the resulting model is computable, and that computations can be composed or decomposed in various ways.

1 Introduction

To clarify the foundations of the study of computation, Robin Gandy [7] described a very general model of computation, in which computations are causal processes constrained by two conceptual principles: first, that states of a computation are finite objects with a bounded amount of local detail; second, that in the course of computation changes propagate through states with a bounded velocity. He gave very general set theoretic axioms for these principles. This remarkable idea has been a source of inspiration in the intervening years [1, 2, 11], especially for those interested in the relationship between computability and the physical world [3, 5]. Later researchers often replace the set theoretic axioms with particular structures to simplify reasoning.

In this paper we see how a general interpretation of Gandy's principles can be given using category theory, in such a way that the resulting state update is always computable. Category theory has several advantages. Functoriality takes care of various 'relabelling invariance' properties which many models require. It also adds a kind of concreteness. Usually one wants to talk about parts of a certain shape occurring in various places inside states. This 'located part' notion gives rise to a category, and it is conceptually simpler to work in terms of this structure rather than the specifics of how states are encoded. The utility of tools like comma categories and Kan extensions is emphasized in [8, 6] which model a notion of locally deterministic global

*joseph.razavi@manchester.ac.uk
†andrea.schalk@manchester.ac.uk

rewriting. The present work can be thought of as identifying which categories are suitable, and showing how the causal neighbourhoods of Gandy can be defined in terms of the categorical structure.

2 Categorical Gandy Machines

To describe suitable categories of local states, we need to ensure the objects can be thought of as finite structures with a finite amount of local variation. We do this by considering locally finite categories $\mathbb{C}$ with a finite dense subcategory $\mathbb{B}$.

We model the update of local pieces of states as an endofunctor $\boldsymbol{U}$ on $\mathbb{C}$. Conceptually, we picture $\boldsymbol{U}$ updating the partial state based on the information it contains, which might involve removing a 'boundary region' where information could have come in from outside. An effect to be explained can be thought of as a part of an updated state; an object of the comma category $\mathbb{C}/\boldsymbol{U}$. Because of the propagation of changes through the state, we know that any effect must have been caused by the smallest region in the input containing all the changes which will give rise to it. Therefore, there should be an initial causal explanation for any effect. Thus, we demand that the obvious functor $\boldsymbol{U}^{\triangleleft} : \mathbb{C}/\mathbb{C} \to \mathbb{C}/\boldsymbol{U}$ which arises from applying $\boldsymbol{U}$ have a left adjoint $\boldsymbol{F}$. A priori, such a left adjoint could change the codomain of the morphism, explaining an effect by, as it were, causes from another universe! However, a general argument shows that if it has a left adjoint, it has one which preserves codomain.

The left adjoint models the idea that 'changes propagate...', to which we need add '...with finite velocity'. This is modelled by demanding that $\boldsymbol{F}$ preserves the property of being a subcategory with a finite image under the domain projection. In detail, we have the following.

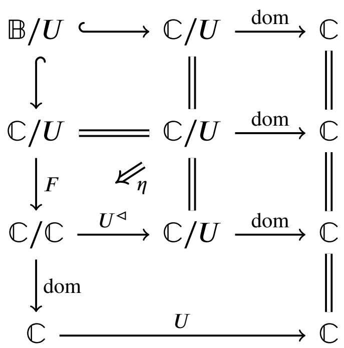
Definition 1. A **categorical Gandy machine** is a functor $\boldsymbol{U} : \mathbb{C} \to \mathbb{C}$, where $\mathbb{C}$ is a locally finite category with a finite dense subcategory, such that

- the induced functor $\boldsymbol{U}^{\triangleleft} : \mathbb{C}/\mathbb{C} \to \mathbb{C}/\boldsymbol{U}$ has a left adjoint $\boldsymbol{F}$, and
- for every subcategory $\mathbb{A}$ of $\mathbb{C}/\boldsymbol{U}$ where $\mathsf{dom}[\mathbb{A}]$ is finite, $(\mathsf{dom} \circ \boldsymbol{F})[\mathbb{A}]$ is also finite.

3 Computability

We now turn to establishing that every process described by a categorical Gandy machine is computable. As a first approximation, we show that $\boldsymbol{U}$ is given by a pointwise Kan extension. This roughly corresponds to some of Gandy's reasoning about reconstructing the result of computation from overlapping parts, and brings us closer to the framework of [8] in which being given by a Kan extension is part of the definition.

Proposition 3.1. *Let $U : \mathbb{C} \longrightarrow \mathbb{C}$ be a categorical Gandy machine. Let $F_{\triangleleft}$ be the left adjoint for $U^{\triangleleft}$ stipulated by the definition and let η be the unit of that adjunction. Let $\mathbb{B}$ be a dense subcategory of $\mathbb{C}$ and $i : \mathbb{B}/U \hookrightarrow \mathbb{C}/U$ be the induced inclusion. Then the following diagram exhibits U as the pointwise (left) Kan extension of* $(\mathrm{dom} \circ i)$ *along* $(\mathrm{dom} \circ F_{\triangleleft} \circ i)$.



Proof. The top row is a pointwise Kan extension because of density of $\mathbb{B}$. The others are absolute Kan extensions, hence pointwise, and the result follows since pointwise Kan extensions are closed under vertical pasting. □

It is worth noting that the proof is a simple argument about pasting diagrams, making use of standard results about pointwise Kan extensions. The result holds not only in **Cat**, but in any 2-category with comma objects.

Although this Proposition gives *some* representation of UX for any input X, it does not quite suffice to establish that U is computable, because it might not be possible to compute the canonical diagram for UX from an arbitrary colimit diagram. However, a very similar proof shows that the Kan extension has an extra property which makes this possible.

Proposition 3.2. *Let $U : \mathbb{C} \longrightarrow \mathbb{C}$ be a categorical Gandy machine as in Proposition 3.1. Then the Kan extension given is preserved by the functors* $\mathbb{C}[b, -]$ *for all objects b of $\mathbb{B}$.*

Finally, given an input X in $\mathbb{C}$, it is not clear from the pasting that the colimit diagram for UX is finite, as $\mathbb{B}/U$ is in general an infinite category. However, straightforward reasoning shows that one need only consider morphisms of the form $\eta_g : b \to \mathrm{dom}(UFg)$. Then since $\mathbb{B}$ is finite, the finiteness condition on F guarantees that the diagram is finite. Computing this finite diagram amounts to calculating hom-sets in $\mathbb{C}$ which is done by using the fact that $\mathbb{B}$ is dense, yielding the following.

Corollary 3.3. *Let $U : \mathbb{C} \longrightarrow \mathbb{C}$ be a categorical Gandy machine as in Proposition 3.1. Then U is computable.*

4 Forthcoming Work: Composing Gandy Machines

One of the important reasons to use category theory is that it ought to provide tools for building up examples out of simple pieces. In this section we sketch some results in this direction which will appear in forthcoming work.

Proposition 4.1. *Categorical Gandy machines on a category* $\mathbb{C}$ *form a monoid.*

This follows from general properties of pullbacks, which avoids having to define the composed adjunction by hand. The analogous result holds in the model of [2], where at first causal antecedents were calculated for the composite explicitly. However, in [4] a topological definition is given—a vast generalization of Hedlund's theorem for cellular automata—which achieves a similar simplification of the analogous result. We view the category theoretic version as living somewhere in between: one avoids manually describing the causal neighbourhoods, but at the same time reasons concretely in the sense of providing an explicit description of them.

We now turn to another way of building more complicated categorical Gandy machines out of simpler ones. First, a ready supply of simple categorical Gandy machines is given by functors which have a left adjoint 'on the nose'.

Proposition 4.2. *Let* $\mathbb{C}$ *be a category, and* $\boldsymbol{U} : \mathbb{C} \to \mathbb{C}$ *have a left adjoint. Then* $\boldsymbol{U}^{\lhd}$ *has a left adjoint.*

The examples given by this Proposition are usually uninteresting. For instance, consider the subcategory of graphs consisting only of paths, and the functor $\boldsymbol{U}$ which removes the outer two nodes from a path. Such examples can be thought of as 'spaces', over which we might like to overlay more complicated dynamics by adding local state. This is achieved using discrete fibrations.

Proposition 4.3. *Let* $\boldsymbol{U} : \mathbb{C} \to \mathbb{C}$ *be a categorical Gandy machine, with* $\boldsymbol{U}^{\lhd}$ *right adjoint to* $\boldsymbol{F}$*. Let* $\mathbb{D}$ *be a locally finite category with a finite dense subcategory, and* $\boldsymbol{P} : \mathbb{D} \to \mathbb{C}$ *be a discrete fibration. Now suppose we have a functor* $\boldsymbol{W} : \mathbb{D} \to \mathbb{D}$ *making the diagram*

$$\begin{array}{ccc} \mathbb{D} & \xrightarrow{\boldsymbol{W}} & \mathbb{D} \\ {\scriptstyle \boldsymbol{P}}\downarrow & & \downarrow{\scriptstyle \boldsymbol{P}} \\ \mathbb{C} & \xrightarrow{\boldsymbol{U}} & \mathbb{C} \end{array}$$

commute. Then $\boldsymbol{w}$ *is a categorical Gandy machine.*

The utility of this result is that $\boldsymbol{W}$ can be specified in a manner similar to a cellular automaton: for an object $\boldsymbol{D}$ of $\mathbb{D}$, applying $\boldsymbol{U}\boldsymbol{P}$ gives a 'blank template' which determines the 'shape' of $\boldsymbol{W}\boldsymbol{D}$. If one can fill these blank templates in functorially, then the resulting functor must be a categorical Gandy machine.

5 Future Work

One direction of future work would be to investigate other ways of composing categorical Gandy machines. It is likely that allowing a finite amount of local nondeterminism would help with this, and the present axiomatization suggests standard ways in which this might be approached. Similarly, one might wish to extend to a stochastic analogue, especially if one wished to study naturally occurring systems. These however are not our immediate concerns.

Our original motivation to investigate these axioms is to view computations over a fixed space, such as those generated by Proposition 4.3 for a suitably nice base category, as *continuous flows of information* over the base. This programme is sketched in a rather inchoate form in the first author's thesis [9], but we lacked a suitable axiomatization. For this investigation, it is crucial to be able to get at the causal neighbourhoods in a way which provides reasoning principles for them. We believe that the present work has reached this point.

Finally, the connection to the topological model of [4] certainly deserves attention. A tantalizing hint is given by asking whether Proposition 4.3 has an analogue there. If our current understanding of that work is correct, very simple 'unlabelled' spaces in our sense should correspond to *one point* spaces in that setting, since there is only one possible configuration. However, they may well have interesting structure *as locales*, suggesting that topos theory might provide a link. As is well known, the continuous function will induce an adjunction between toposes, but the connection with the present work is certainly not immediate and remains speculative.

Indeed, Gandy, reflecting on the use of 'restrictions' in his axioms, wondered whether 'a treatment using concepts analogous to those of sheaf theory or topos theory might be worth developing' but worried that 'the concepts from category theory which would be necessary would be too abstract' to justify the thesis he was defending. It may yet be that his vision can be doubly vindicated: we have endeavoured here to give a concrete conceptual account of his ideas using category theory, and in its relationship to [4] may be the germ of the abstract account he foresaw.

References

[1] Pablo Arrighi, Clément Chouteau, Stefano Facchini, and Simon Martiel. Causal dynamics of discrete manifolds. *arXiv preprint arXiv:1805.10051*, 2018.

[2] Pablo Arrighi and Gilles Dowek. Causal graph dynamics. In Artur Czumaj, Kurt Mehlhorn, Andrew Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming*, pages 54–66, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[3] Pablo Arrighi and Gilles Dowek. The physical Church-Turing thesis and the principles of quantum theory. *International Journal of Foundations of Computer Science*, 23(05):1131–1145, 2012.

[4] Pablo Arrighi, Simon Martiel, and Vincent Nesme. Generalized Cayley graphs and cellular automata over them. *arXiv preprint arXiv:1212.0027*, 2012.

[5] B. Jack Copeland and Oron Shagrir. Physical computation: How general are Gandy's principles for mechanisms? *Minds and Machines*, 17:217–231, 2007.

[6] Alexandre Fernandez, Luidnel Maignan, and Antoine Spicher. Lindenmayer systems and global transformations. In *International Conference on Unconventional Computation and Natural Computation*, pages 65–78. Springer, 2019.

[7] Robin Gandy. Church's thesis and principles for mechanisms. In *The Kleene Symposium*, pages 123–148. North-Holland, 1980.

[8] Luidnel Maignan and Antoine Spicher. Global graph transformations. 2015.

[9] Joseph Razavi. *Information Flow in Spatial Models of Computation*. PhD thesis, The University of Manchester, August 2017.

[10] Joseph Razavi and Andrea Schalk. A category theoretic interpretation of Gandy's principles for mechanisms. *arXiv preprint arXiv:1904.10109*, 2019.

[11] Wilfried Sieg and John Byrnes. ***K**-graph machines: generalizing Turing's machines and arguments*, volume 6 of *Lecture Notes in Logic*, pages 98–119. Springer-Verlag, Berlin, 1996.