APPLIED CATEGORY THEORY IN CHEMISTRY, COMPUTING, AND SOCIAL NETWORKS

JOHN BAEZ, SIMON CHO, DANIEL CICALA, NINA OTTER, VALERIA DE PAIVA

1. Introduction

Society is increasingly complex and connected through the internet and social media, planetary climate and ecological challenges, transnational organization and global supply chains. To navigate and thrive in this networked world, we rely on scientific advances to help us manage this complexity by enabling robust communication, cooperation, and collaboration.

Within about the past decade, a growing number of researchers have realized that the aspects of category theory that make it useful in certain pure mathematical contexts also make it useful for the study of the underlying structure of physical and conceptual systems. From this realization, a new field has emerged called Applied Category Theory (ACT). Some major themes currently found in the ACT literature include compositionality, functorial semantics, and implementing these structures into user-friendly software. Indeed, engineers and scientists should benefit from the fruits of ACT, ideally without having to first study category theory which is why producing user-friendly software is a north star of ACT research. In this note, we provide a bird's eye view of these major themes, describe a road map to relevant literature, and highlight the essence and intuition of the central ideas as well as the payoffs that a category theoretic approach can bring. Into this narrative, we fit a brief description of specific research projects to be undertaken by participants of the 2022 Mathematical Research Community in Applied Category Theory.

2. Compositionality

To a category theorist, it is not the mathematical objects, but the morphisms between objects that are held to be fundamental. This viewpoint necessarily lifts composition to the fore of mathematical operations. When considering examples of a morphism, many may conjure functions between sets, homomorphisms between rings, or continuous maps between spaces. These examples are certainly important, however, morphisms can truly be *anything* satisfying the axioms for a category. One main thread of research in applied category theory is to model open systems by arranging them as morphisms in some category. "Open" here means that the systems are equipped with an interface that can interact with other compatible systems.

2.1. Structured Cospans. One method of encoding open systems as morphisms is to consider them as 'cospans' [5]. The idea is to create a category where we interpret each object as a system of some sort, and then define a *cospan* to be an object S with two morphisms into it

$$X \to S \leftarrow Y$$

that select which parts of S serve as inputs and outputs. Composition of cospans,

$$X \to S \leftarrow Y \text{ followed by } Y \to T \leftarrow Z,$$

is given by a purely categorical construction known as a pushout, which connects the outputs of S to the inputs of T. In applied category theory we often need *structured* cospans

$$X \to U(A) \leftarrow Y$$

where the object A lives in a different category from X and Y, related by a functor U. For example, X and Y could be sets, and A could be a graph.

The structured cospan approach has been applied to chemical reaction networks [9], Markov processes [6] (see Figure 1, and electrical circuits [7, 10]. Petri nets, typically found in chemistry and computer science, are a graphical formalism to describe distributed systems. These too can be realized as structured cospans thus offering

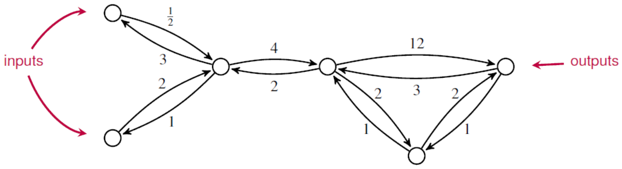


Figure 1. Open Markov Process

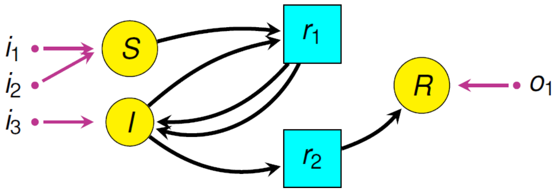


Figure 2. Open SIR Model as a Petri Net

a way to categorically build complex processes [8]. Figure 2 shows an open Petri net as a structured cospan encoding a simple model of infectious disease. Here S stands for a population of 'susceptible' people, I stands for 'infected' and R stands for 'resistant'.

It turns out that standard ways to manipulate a system—for example connecting outputs of one system to the inputs of another, turning an output into an input, considering multiple systems as a single system—are all realized with purely category theoretic operations. The payoff is that many different systems can be described in the same language: category theory. With different systems on equal footing, comparisons are more readily available. Rigorous, not simply heuristic, diagrammatic languages exist to assist in reasoning about systems of various kinds, and a structural analysis of systems may commence.

2.2. Open Reaction Networks. Reaction networks are a widely used method of describing chemical reactions. There is a standard method of turning a reaction network into a collection of differential equations describing the time evolution of the concentration of various chemicals in solution. Starting in the 1970s, mathematical chemists formulated a number of deep theorems [17] and conjectures [1] saying how the qualitative behavior of these differential equations depend on topological features of the reaction network.

More recently, structured cospans have been used to describe "open" reaction networks— where chemicals can flow in and out—as morphisms in a category [9]. We can build larger reaction networks by composing smaller open ones, and the map sending an open reaction network to its differential equation is a functor. In the 2022 MRC in Applied Category Theory, participants will use this framework to study the qualitative behavior of chemical reactions.

2.3. Lenses. Lenses offer another method to connect systems together and are particularly useful to model a scenario involving a bidirectional flow of information between connected systems. A helpful, if rough, approximation of a lens is two interacting systems, each encoded as a set of states X, Y together with one map $f\colon X \to Y$ that "sends information forward" and a second map $b\colon X \times Y \to X$ that "sends information backwards". To illustrate, imaging that X is the set of behaviors of an individual named James and Y is the set of behaviors of the Category Cafe, James' favorite coffee house. The forward function captures how the Category Cafe behaves $f(x) \in Y$ in reaction to each of James' behavior $x \in X$. For instance, perhaps $x =$ 'James orders a coffee' maps to $f(x) =$ 'an employee pours a coffee'. The backwards function captures how each state of the cafe y affects each of James' behavior $b(-, y)\colon X \to X$. If $y =$ 'the cafe is busy', then $b(-, y)\colon X \to X$ might update $x_1 =$ 'orders a coffee' to $b(x_1, y) =$ 'leaves the cafe', hold $x_2 =$ 'uses the restroom' constant so $b(x_2, y) = x_2$ and, update $x_3 =$ 'sit and check emails' to $b(x_3, y) =$ 'stand in the corner and wait until customers leave'.

While this toy example imparts the flavor of a lens, it does not impart the lens' full majesty when the appropriate rigor and generality is considered. Indeed, lenses are so useful that people continue to rediscover them in seemingly unconnected situations. Gödel's Dialectica interpretation [18], a model of intuitionistic arithmetic, offers an early discovery of lenses, though without the term. Later de Paiva placed Gödel's logical framework into a category whose morphisms are generalized lenses [21]. This *Dialectica construction* has come to establish much of the current understanding about lenses. Contemporary lens applications include database theory [14], a structural perspective on functional learning [12], domain theory [23], and open game theory [16] with an emphasis on economic models. This same structure appearing in so many places excited category theorists who in turn began to study lenses on their own terms, starting with the category of *lenses in the category of sets*. The objects of this category are sets and the morphisms $(f, b)\colon X \to Y$ are lenses, so a pair of functions $f\colon X \to Y$ and $b\colon X \times Y \to X$ subject to several compatibility laws, that is commuting diagrams

$$\begin{tikzcd}
X \times Y \arrow[r, "b"] \arrow[dr, "\pi_2"'] & X \arrow[d, "f"] \\
& Y
\end{tikzcd}
\qquad
\begin{tikzcd}
X \arrow[r, "\langle \mathrm{id}_X, f \rangle"] \arrow[dr, "\mathrm{id}_X"'] & X \times Y \arrow[d, "b"] \\
& X
\end{tikzcd}$$

$$\begin{tikzcd}
X \times Y \times Y \arrow[r, "b \times \mathrm{id}_Y"] \arrow[d, "\pi_{1,3}"'] & X \times Y \arrow[d, "b"] \\
X \times Y \arrow[r, "b"'] & X
\end{tikzcd}$$

A composite of lenses

$$(f_1, b_1)\colon X \to Y \text{ and } (f_2, b_2)\colon Y \to Z$$

comprises the functions

$$X \to Z \text{ and } Z \times X \to X,$$

respectively defined by $x \mapsto f_2(f_1(x))$ and

$$(z, x) \mapsto b_1(b_2(z, f_1(x)), x)$$

and are depicted in Figure 3 using a string diagram. This category generalizes in various directions, for instance by taking different permutations of the compatibility laws, by taking lenses in various categories or, repeatedly, by replacing the Cartesian product with another monoidal product.

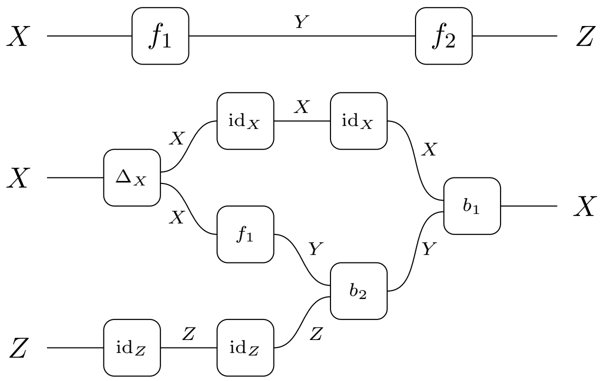


Figure 3. Lens Composition as a String Diagram

2.4. Dialectica Interpretation. When combing the above cited literature about lenses, one would notice that there are actually variations of lenses just as there are for any mathematical object. In fact, one of the variants of the lenses discussed by both [12] and [16] seems to be a certain restriction of de Paiva's Dialectica construction, although it is not immediately obvious to what degree such a restriction preserves the logical structure of the construction. In the 2022 MRC for Applied Category Theory, participants will construct a framework that clarifies in which precise sense the concept of lens as embodied by the Dialectica construction generalizes the variations of lenses discussed above.

3. Functorial Semantics

We have two formal and rigorous methods of building systems from their constituent parts: structured cospans and lenses. The categories we build from structured cospans or from lenses offer a syntax that we can use to reason about the structure of systems. However, we would also like to understand their behavior. Given our interest in composite systems, a natural question to ponder is: how much of a system's behavior is explained by the behavior of its component parts? To answer this question, we can borrow ideas from one of category theory's giants.

In his PhD thesis, William Lawvere introduced a category theoretic perspective on universal algebra called functorial semantics [19]. The idea is to encode the theory for an algebraic object into a category. For example, the category for the theory of a group will have its objects generated by taking all finite products of a single object G, giving objects

$$1,\ G,\ G\times G,\ G\times G\times G,$$

and so on. The morphisms of this category are generated, via composition and products, by the structure maps $e\colon 1\to G$, $(-)^{-1}\colon G\to G$, and $*\colon G\times G\to G$. The resulting morphisms are then quotiented by equations between morphisms that describe the properties of identity, invertibility, and associativity. This construction provides a unique morphism for every possible way to turn a string of group elements into a single element, for instance

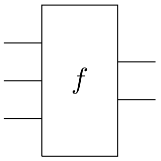
$$(g_1,g_2,g_3,g_4)\mapsto((g_1^{-1}*g_2)*g_3)^{-1})*g_4$$

has a dedicated morphism of type

$$G\times G\times G\times G\to G.$$

Note, there are no actual elements here, we are just using generalized symbols to describe the morphism. The resulting category **Th**(Group) is not a group; it is the syntax for groups. This is directly in line with our categories constructed using structured cospans or lenses to capture the syntax of various systems. Then, once we have a syntax, we can use a functor out of that syntax and into another category to realize the semantics. For example, every group is a functor **Th**(Group) $\to$ **Set** to the category of sets and set functions. Here **Set** is the semantics of the group. By changing the semantics, we can obtain the many flavors of groups: each topological group is a functor **Th**(Group) $\to$ **Top** to the category of topological spaces and continuous maps, each Lie group is a functor **Th**(Group) $\to$ **Diff** into the category of differential manifolds and smooth maps.

Applied category theorists use this idea to study open systems using two categories. The first category has as morphisms the open systems, for example encoded as structured cospans. This category serves as the syntax for the system, governing how we can combine systems to make larger, more complex systems. The second category captures the behavior of these systems. This category serves as the semantics and is typically the category **Rel** whose objects are sets and morphisms are binary relations, though a category of stronger relations may be appropriate. Then a functor **Syntax** $\to$ **Semantics** assigns to each system (a morphism in the syntax category), the relationship between behaviors on the system's inputs and outputs. For example, there is a functor from the category whose morphisms are passive linear circuits to the category **LinRel** whose objects are $\mathbb{R}^n$ for each natural number n and morphisms $\mathbb{R}^m\to\mathbb{R}^n$ are linear relations, that is linear subspaces of $\mathbb{R}^m\oplus\mathbb{R}^n$. This functor assigns to a passive linear network



$$\mapsto\ R\subset\mathbb{R}^6\oplus\mathbb{R}^4$$

where R comprises the tuples $(v_k,i_k)_{k=1}^{6}$ that represent the realizable potential-current pairs that can exist on each port according to Kirchhoff's Circuit Laws.

In general, these semantics-assigning functors capture the external behavior of a system as a composite of the system's components. The generality of this approach favors the structural perspective and, by using category theory as a common language, allowing for a more readily-made comparison for systems of different types. In an era of increasing interdisciplinarity, the ability to translate knowledge across disciplines is crucial. Applied category theory is one approach towards building such a dictionary.

It is worth noting that functorial semantics as described above does does not capture any behavior that is emergent from composing systems. Research is underway in this direction by using so called *lax functors* [13].

3.1. Social Simplicial Complexes. The power of functors goes beyond their ability to describe the deconstruction of systems into their syntax and semantics. They are a powerful organizational tool that encompasses a staggeringly large number of the most famous mathematical operations. Indeed, computing some free algebraic object on a set, the fundamental group on a space, the homology and cohomology of spaces, the tangent or cotangent bundle of a smooth manifold are all functors. Even a space like a sheaf or presheaf can be represented as a functor. By thinking about a space as a functor, the higher-dimensional features can be studied using higher category theory, a perspective that offers new tools to classical subjects. In the field of Topological Data Analysis, functoriality of many constructions is a key ingredient in the study of their robustness [11].

In the 2022 MRC, participants will study social systems using functors and other category theoretic tools. Many of the methods currently used in network science were first developed by social network scientists, who use nodes to represent an agent of a social system, and (un)directed labelled edges to represent binary relations between agents (see Figure 4).

Two of the main properties of social systems that social scientists are interested in studying are positions and roles. For networks, positions are defined as equivalence classes of nodes, while roles are equivalence classes of compound relations [22]. Since the 1970s a lot of research has been done to develop these concepts in a rigorous way [15]. Otter and Porter developed methods to relate the analysis of roles and positions in social networks, using a functorial formulation [20]. At the 2022 MRC, we intend to extend this functorial framework to account for higher-order interactions between social agents by modeling social systems with simplicial complexes instead of mere graphs.

4. Software Development

A goal of ACT community is to bridge the gap between theorists using category theoretic modeling tools and those who want to *use* the models to say something useful and true about the world. One can cross their fingers and hope that the "users" will simply take it upon themselves to learn enough category theory to take advantage of ACT-styled models. A more proactive approach would be to build user-friendly (meaning, no category theory knowledge required) tools. Such tools will likely take the form of computer software with intuitive graphical interfaces where the category theory is programmed under-the-hood. A number of researchers are currently working on building such software tools, though this work is very much in its infancy. One example include Globular [3], a proof assistant that allows one to perform higher-dimensional calculations in categories via a graphical interface. Structured cospans of Petri nets were implemented in the software package Julia to develop an SIR model that is compositional in the sense that various cities can each have their own model that can be connected together to form a composite SIR model [2]. Users can set parameters and all the category theory remains underneath the hood. Private enterprise is also entering the picture. The organization Statebox[1] is blending an ACT approach to Petri Nets together with blockchain technology to develop a technology stack based on a visual programming language. In addition, they have built a software engine for compositional game theoretic modeling, a finite state machine oracle [4]. The company Conexus[2] uses applied categorical methods for data integration.

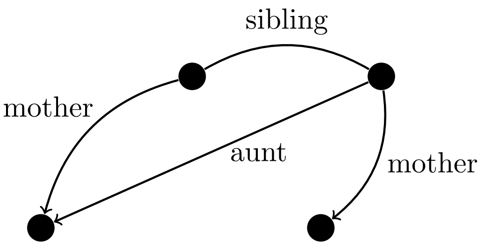


FIGURE 4. An example of social network modeling kinship relationships.

The success of ACT as a discipline largely hinges on its ability to be accessible and available to scientists and engineers, meaning the building of software is central to the ACT program.

[1] https://statebox.org/

[2] https://conexus.com/

5. Conclusion

The ACT community is continuing to grow and seeking early-career researchers, programmers, scientists, and engineers of all stripes to join us at the 2022 Mathematical Research Community. Those who enjoy a systems-thinking and structural perspective will find that category theory provides a rigorous and robust framework for reasoning about systems, processes, and relationships. Expertise in category theory is not required to join, just a desire to learn.

References

[1] Anderson, D., 2011. A proof of the global attractor conjecture in the single linkage class case, SIAM J. Appl. Math. 71, 1487–1508. Available at arXiv:1101.0761.

[2] Baas, A., Fairbanks, J., Libkind, S., and Patterson, E., 2021. Operadic modeling of dynamical systems: mathematics and computation. Available at arXiv:2105.12282.

[3] Bar, K., Kissinger, A. and Vicary, J., 2016. Globular: an online proof assistant for higher-dimensional rewriting. arXiv preprint arXiv:1612.01093. Available at http://globular.science/

[4] Statebox Team. The Mathematical Specification of the Statebox Language, 2018. Available at arXiv:1906.07629.

[5] Baez, J.C. and Courser, K., 2020. Structured cospans. Theory and Applications of Categories, 35, 1771–1822. Available at arXiv:1911.04630

[6] Baez, J.C., Fong, B. and Pollard, B.S., 2016. A compositional framework for Markov processes. Journal of Mathematical Physics, 57(3), p.033301. arXiv:1508.06448

[7] Baez, J.C., Coya, B. and Rebro, F., 2018. Props in network theory. Theory and Applications of Categories, 33, 727–783. Available at arXiv:1707.08321

[8] Baez, J.C. and Master, J., 2020. Open Petri nets. Mathematical Structures in Computer Science, 30(3), 314–341. Available at arXiv:1808.05415.

[9] Baez, J.C. and Pollard, B.S., 2017. A compositional framework for reaction networks. Reviews in Mathematical Physics, 29(09), p.1750028. Available at arXiv:1704.02051.

[10] Baez, J.C. and Fong, B., 2018. A compositional framework for passive linear networks, Theory and Applications of Categories, 33, 1158–1222. Available at arXiv:1504.05625.

[11] Carlsson, G., 2008. Topology and data. *Bulletin of the AMS*, **46**: 255–308.

[12] Fong, B., Spivak, D. and Tuyéras, R., 2019, June. Backprop as functor: A compositional perspective on supervised learning. In 2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) (pp. 1-13). IEEE. Available at arXiv:1711.10455.

[13] Fong, B. and Spivak, D.I., 2019. *An Invitation to Applied Category Theory: Seven Sketches in Compositionality*. Cambridge University Press. Available at arXiv:1803.05316.

[14] Foster, J.N., Greenwald, M.B., Moore, J.T., Pierce, B.C. and Schmitt, A., 2007. Combinators for bidirectional tree transformations: A linguistic approach to the view-update problem. ACM Transactions on Programming Languages and Systems (TOPLAS), 29(3), pp. 17-es.

[15] Freeman, L., 2004. The Development of Social Network Analysis, Empirical Press.

[16] Ghani, N., Hedges, J., Winschel, V. and Zahn, P., 2018, July. Compositional game theory. In Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (pp. 472–481).

[17] Horn, F. and Jackson, R., 1972. General mass action kinetics, *Arch. Rat. Mechanics and Analysis*, 50, 81–116.

[18] Gödel, K., Feferman, S., et. al., 1986. Kurt Gödel: Collected Works: Volume II: Publications 1983–1974 (Vol. 2). Oxford University Press.

[19] Lawvere, F.W., 1963. Functorial semantics of algebraic theories. Proceedings of the National Academy of Sciences of the United States of America, 50(5), p.869. Available at http://www.tac.mta.ca/tac/reprints/articles/5/tr5abs.html.

[20] Otter, N. and Porter, M.A., 2020. A unified framework for equivalences in social networks. Available at arXiv:2006.10733.

[21] de Paiva, V., 1989. The dialectica categories. Categories in Computer Science and Logic, 92, 47–62.

[22] Wasserman, S. and Faust, K., 1994. Social Network Analysis, Cambridge University Press.

[23] Winskel, G., 2021. Domain theory and Interaction. British computer Science FACS (Formal Aspects of Computer Science) Newsletter. British Computer Society. Available at https://www.bcs.org/media/7577/facs-jul21.pdf