

AlgebraicSystems: Compositional Verification for Autonomous System Design

Georgios Bakirtzis

The University of Texas at Austin
bakirtzis@utexas.edu

Ufuk Topcu

The University of Texas at Austin
utopcu@utexas.edu

ABSTRACT

Autonomous systems require the management of several model views to assure properties such as safety and security among others. A crucial issue in autonomous systems design assurance is the notion of emergent behavior; we cannot use their parts in isolation to examine their overall behavior or performance. Compositional verification attempts to combat emergence by implementing model transformation as structure-preserving maps between model views. AlgebraicDynamics relies on categorical semantics to draw relationships between algebras and model views. We propose AlgebraicSystems, a conglomeration of algebraic methods to assign semantics and categorical primitives to give computational meaning to relationships between models so that the formalisms and resulting tools are interoperable through vertical and horizontal composition.

1 MOTIVATION

Ensuring that autonomous systems will behave as expected based on their requirements is often achieved by modeling. Model-based design is different from science (Figure 1). In science, we deal with systems that we have no control over, and we attempt to create formal models of, for example, how things move based on assumptions about the environment, inching with different paradigms closer to reality. Instead, in engineering, we have the benefit of amalgamating systems *from* models; that is, the value of our realized systems is how well we can conform them to our understanding captured in models [14]. Here too, we deal with different paradigms. In autonomous systems, the different paradigms can be viewed explicitly or implicitly as residing within distinct algebras.

Viewing models as algebras allows us to reason compositionally between them. However, most engineering work in compositionality centers around a particular formalism. For example, hybrid systems [16] and timed automata [10] model behavior, linear temporal logic specifications [1] and contracts [12] model requirements. We focus on the composition of individual algebras rather than their relation. Call this horizontal-type composition; within one mathematical model, we compose the same types of models to produce larger ones. Horizontal composition is generally an accepted line of work within one field. Still, a perhaps more interesting rule would be vertical composition, which would relate or otherwise be able to enforce a hierarchy among multiple such formalisms (Figure 2). One way of achieving vertical composition is by using tools from category theory.

Compositional verification refers to the rules that connect individual parts to construct a whole in a way that the behavior determines the whole. This is to say that compositionality is about *refinement* and *abstraction* [13]. Refinement is about augmenting

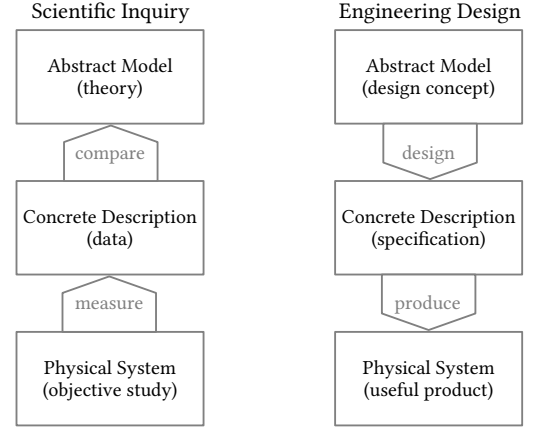


Figure 1: The value of an engineered system is how well it matches its associated models (adapted from Drexel [11]).

or recovering more information about a particular component. But abstraction is equally important in designing increasingly complex autonomous systems, which require us to put components together that form larger systems via black-boxing every component in such a way that, when composed, defines the desired system’s behavior.

One way to relate the algebras of requirement, behaviors, and architectures is to carve a common research trajectory with applied category theory. In engineering, category theory can give precise meaning to the transformations associated with remembering and forgetting between model views. To date, these advances are predominantly theoretical. To be fruitful in the program of compositional systems theory [2, 6], we must develop a computational interpretation of algebras and their associated horizontal and vertical composition rules. The eventual user of these tools should not have to be an expert in category theory but rather be able to leverage (1) warnings raised when composition fails and (2) have interoperability between theories, tools, and consequently analyses and synthesis techniques. AlgebraicSystems is an envisioned comprehensive program that implements these ideas in the high-performance Julia programming language [8].

Interoperability between models and tools outputs assurance cases for properties we care about, such as safety and security, through compositional verification [4]. By designing systems compositionally, we precisely address the lack of interoperability between formalisms, an open problem [15], within AlgebraicSystems.

2 COMPOSITIONAL VERIFICATION

A combination of models is often used to assure metrics such as safety and security and dynamics and control (Figure 2). The categorical formalization of composition gives rise to the unification

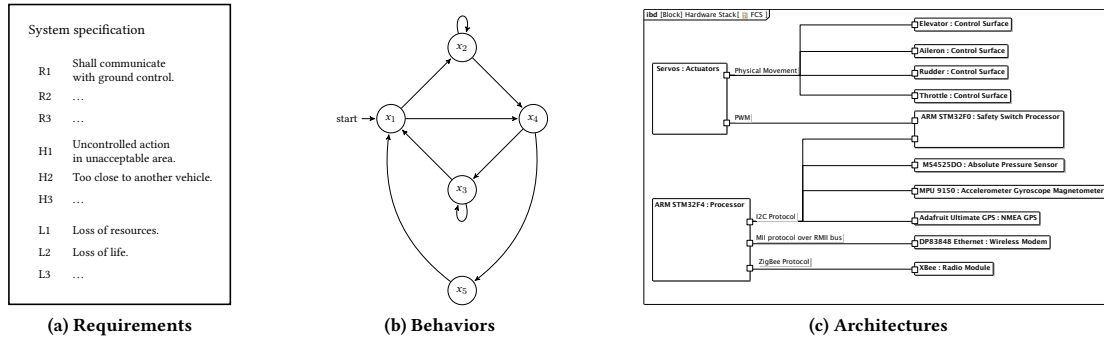


Figure 2: There is a semantic gap between types of models needed to assure safe and secure behavior of deployed systems.

of (a subset) of requirements, system behaviors, and system architectures in a traceable manner. The categorical primitive of functoriality—structure-preserving maps between categories—gives concrete meaning to abstraction and refinement in cyber-physical system models, which can assist with the specification (and eventually validation) of increasingly complex systems.

Several mathematical domains, such as graph theory, can implement compositional verification. In compositional systems theory and AlgebraicSystems, verification is not about prediction but rather abstracting and refining structure—organizing different models within categories for syntax and assigning algebras for semantics. Category theory is one context where the meaning of composition is formal and refers to something specific, namely the partial operation on morphisms of a category. We formalize the notion of composite systems in this work using the systems-as-algebras framework in the wiring diagram category.

For example, a controls model can have a syntactic flavor within the wiring diagram category, where we precisely construct formal interfaces between boxes [3, 7]. At the moment, there is no behavior, just an architectural arrangement of parts. It is then the job of the controls algebra to assign meaning to each of the boxes. The computation of the dynamic behavior of the control system is then the horizontal composition of boxes that are inhabited by controls algebras (semantics) and how those semantics give rise to the whole given the arrangement between those boxes with wires (syntax).

There are several models and assurance methods we would like to relate compositionally for autonomous systems. Types of models and assurance methods for autonomous systems that are not currently related or have compositional verification between them include models of Markov decision processes [17], control synthesis [18], contracts [5], and shielding [9]. AlgebraicSystems is to be the conglomeration of these models and methods.

AlgebraicSystems gives formal and computational meaning to relationships between formalisms, domain-specific models, and assurance methods. We can treat syntactic elements as categories and semantics as algebras by implementing compositional verification using category theory. The categorical interpretation of system theory engenders an understanding of model-based design as examining how formalisms are *related* to each other rather than how individual formalisms model systems in isolation.

REFERENCES

- [1] R. Alur, S. Moarref, and U. Topcu. 2018. Compositional and symbolic synthesis of reactive controllers for multi-agent systems. *Information and Computation* (2018). <https://doi.org/10.1016/j.ic.2018.02.021>
- [2] G. Bakirtzis. 2021. *Compositional Cyber-Physical Systems Theory*. Ph.D. Dissertation. University of Virginia. <https://doi.org/10.18130/xn8v-5d89>
- [3] G. Bakirtzis, C. H. Fleming, and C. Vasilakopoulou. 2021. Categorical Semantics of Cyber-Physical Systems Theory. *ACM Transactions on Cyber-Physical Systems* (2021). <https://doi.org/10.1145/3461669>
- [4] G. Bakirtzis, F. Genovese, and C. H. Fleming. 2021. Yoneda Hacking: The Algebra of Attacker Actions. *arXiv:2103.00044 [cs.CR]* (2021).
- [5] G. Bakirtzis and R. Gonzalez. 2022. bakirtzisg/AlgebraicContracts.jl. <https://doi.org/10.5281/zenodo.6166867>
- [6] G. Bakirtzis, E. Subrahmanian, and C. H. Fleming. 2021. Compositional Thinking in Cyberphysical Systems Theory. *Computer* (2021). <https://doi.org/10.1109/MC.2021.3085532>
- [7] G. Bakirtzis, C. Vasilakopoulou, and C. H. Fleming. 2020. Compositional Cyber-Physical Systems Modeling. In *Proceedings of the 2020 Applied Category Theory Conference (ACT 2020) (EPTCS)*. <https://doi.org/10.4204/EPTCS.333.9>
- [8] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. 2017. Julia: A Fresh Approach to Numerical Computing. *SIAM Rev.* (2017). <https://doi.org/10.1137/14100671>
- [9] B. Könighofer, M. Alshiekh, R. Bloem, L. R. Humphrey, R. Könighofer, U. Topcu, and C. Wang. 2017. Shield synthesis. *Formal Methods for System Design* (2017). <https://doi.org/10.1007/s10703-017-0276-9>
- [10] P. Bouyer and A. Petit. 1999. Decomposition and Composition of Timed Automata. In *Proceedings of the 26th International Colloquium on Automata, Languages and Programming (ICALP 1999) (LICS)*. https://doi.org/10.1007/3-540-48523-6_18
- [11] K. E. Drexler. 2013. *Radical abundance: How a revolution in nanotechnology will change civilization*. Public Affairs.
- [12] K. Ghasemi, S. Sadraadini, and C. Belta. 2020. Compositional synthesis via a convex parameterization of assume-guarantee contracts. In *Proceedings of the 23rd ACM International Conference on Hybrid Systems: Computation and Control (HSCC '20)*. <https://doi.org/10.1145/3365365.3382212>
- [13] J. M. Hedges. 2016. *Towards compositional game theory*. Ph.D. Dissertation. Queen Mary University of London.
- [14] Edward A. Lee. 2021. Determinism. *ACM Trans. Embed. Comput. Syst.* (2021). <https://doi.org/10.1145/3453652>
- [15] M. Luckcuck, M. Farrell, L. A. Dennis, C. Dixon, and M. Fisher. 2019. Formal Specification and Verification of Autonomous Robotic Systems: A Survey. *ACM Comput. Surv.* (2019). <https://doi.org/10.1145/3342355>
- [16] A. Nejati, S. Soudjani, and M. Zamani. 2021. Compositional abstraction-based synthesis for continuous-time stochastic hybrid systems. *European Journal of Control* (2021). <https://doi.org/10.1016/j.ejcon.2020.04.001>
- [17] D. Shiebler. 2021. Categorical stochastic processes and likelihood. *Compositionality* (2021). <https://doi.org/10.32408/compositionality-3-1>
- [18] P. Tabuada and G. J. Pappas. 2006. Linear Time Logic Control of Discrete-Time Linear Systems. *IEEE Trans. Automat. Control* (2006). <https://doi.org/10.1109/TAC.2006.886494>