One-sorted Program Algebras*

Igor Sedlár[1] and Johann J. Wannenburg[1]

Institute of Computer Science, Czech Academy of Sciences, Prague, Czech republic
sedlar@cs.cas.cz and wannenburg@cs.cas.cz

Kleene algebra with tests [6], KAT, is a simple algebraic framework for verifying properties of propositional while programs. KAT subsumes Propositional Hoare logic (PHL) [7] and it has been applied in a number of verification tasks. KAT is PSPACE-complete [2], has computationally attractive fragments [9], and its extensions have been applied beyond while programs, for instance in network programming languages [1].

KAT is two-sorted, featuring a Boolean algebra of tests embedded into a Kleene algebra of programs. For various reasons, a one-sorted alternative to KAT may be desirable. For instance, "one-sorted domain semirings are easier to formalise in interactive proof assistants and apply in program verification and correctness" [4, p. 576]. A one-sorted alternative called *Kleene algebra with antidomain* was introduced in [3]. The idea of KAA is to expand Kleene algebra with an *antidomain* operator a, such that $d(x) = a(a(x))$ is a *domain* operation, where the set of images of elements of the algebra under d forms a Boolean algebra in which the complement of $d(x)$ is $a(x)$. Hence, one obtains a Boolean algebra of tests in a one-sorted setting. Consequently, the equational theory of KAT embeds into the equational theory of KAA.

It is known that KAA is decidable in EXPTIME [8], and KAA can be used to create modal operators that invert the sequential composition rule of PHL. Such inversions are derivable from KAA but not KAT [10]. However, KAA has certain features that may be undesirable depending on the application. First, if $\mathbf{K}$ is a KAA, $d(\mathbf{K})$ is necessarily the maximal Boolean subalgebra of the negative cone of $\mathbf{K}$; see Thm. 8.5 in [3]. In a sense, then, every "proposition" is considered a test, contrary to some of the intuitions expressed in [6]. These intuitions also collide with the approach of taking KAT as KA with a Boolean negative cone [4, 5]. Second, not every Kleene algebra expands to a KAA, not even every finite

*This work was carried out within the project *Supporting the internationalization of the Institute of Computer Science of the Czech Academy of Sciences* (no. CZ.02.2.69/0.0/0.0/18_053/0017594), funded by the Operational Programme Research, Development and Education of the Ministry of Education, Youth and Sports of the Czech Republic. The project is co-funded by the EU.

one; see Prop. 5.3 in [3]. This is in contrast to the fact that every Kleene algebra expands to a KAT.

In this talk we generalize KAA to a framework we'll call *one-sorted Kleene algebra with tests*, KAt. We start by assuming equations that essentially state nothing more than that each KAt has a Boolean subalgebra in the negative cone. Already in this case KAt has most of the desired features of KAA: every KAt contains a Boolean subalgebra of tests and the equational theory of KAT embeds into the equational theory of KAt. In addition, every Kleene algebra expands into a KAt (ensuring that it is a conservative expansion), and the subalgebra of tests in KAt is not necessarily the maximal Boolean subalgebra of the negative cone. We then consider various extensions of KAt with axioms known from KAA to show which properties of the domain operator are still consistent with the desired features of KAt. In addition, we consider a variant of the KAt framework where test complementation is defined using a residual of Kleene algebra multiplication.

References

[1] Carolyn Jane Anderson, Nate Foster, Arjun Guha, Jean-Baptiste Jeannin, Dexter Kozen, Cole Schlesinger, and David Walker. NetKAT: Semantic foundations for networks. In *Proc. 41st ACM SIGPLAN-SIGACT Symp. Principles of Programming Languages (POPL'14)*, pages 113–126, San Diego, California, USA, January 2014. ACM.

[2] Ernie Cohen, Dexter Kozen, and Frederick Smith. The complexity of Kleene algebra with tests. Technical Report TR96-1598, Computer Science Department, Cornell University, July 1996.

[3] Jules Desharnais and Georg Struth. Internal axioms for domain semirings. *Science of Computer Programming*, 76(3):181–203, 2011. Special issue on the Mathematics of Program Construction (MPC 2008). URL: https://www.sciencedirect.com/science/article/pii/S0167642310000973, doi:10.1016/j.scico.2010.05.007.

[4] Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Domain semirings united. *Acta Cybernetica*, pages 575–583, 2022. doi:10.14232/actacyb.291111.

[5] Peter Jipsen. From semirings to residuated Kleene lattices. *Studia Logica*, 76(2):291–303, Mar 2004. doi:10.1023/B:STUD.0000032089.54776.63.

[6] Dexter Kozen. Kleene algebra with tests. *ACM Trans. Program. Lang. Syst.*, 19(3):427–443, May 1997. doi:10.1145/256167.256195.

[7] Dexter Kozen. On Hoare logic and Kleene algebra with tests. *ACM Trans. Comput. Logic*, 1(1):60–76, July 2000. doi:10.1145/343369.343378.

[8] Bernhard Möller and Georg Struth. Algebras of modal operator and partial correctness. *Theoretical Computer Science*, 351(2):221–239, 2006. URL: https://www.sciencedirect.com/science/article/pii/S0304397505006195, doi:10.1016/j.tcs.2005.09.069.

[9] Steffen Smolka, Nate Foster, Justin Hsu, Tobias Kappé, Dexter Kozen, and Alexandra Silva. Guarded Kleene algebra with tests: Verification of uninterpreted programs in nearly linear time. In *Proc. 47th ACM SIGPLAN Symp. Principles of Programming Languages (POPL'20)*, pages 61:1–28, New Orleans, January 2020. ACM.

[10] Georg Struth. On the expressive power of Kleene algebra with domain. *Information Processing Letters*, 116(4):284–288, 2016. URL: https://www.sciencedirect.com/science/article/pii/S002001901500191X, doi:10.1016/j.ipl.2015.11.007.