Cheng Zhang

Research Statement

czhang03@bu.edu
cs-people.bu.edu/czhang03/
czhang03
Curriculum vitae

> *"Simplicity is a great virtue." - Edsger W. Dijkstra (1984)*

Computer science as a field is facing many complex problems in today's ever-developing world. Instead of directly attacking these convoluted questions, I believe in building a wealth of mathematical knowledge on simple foundational problems, and compose their solutions to conquer real-world challenges in a clean and efficient manner.

Guided by these goals, my PhD study focuses on three fundamental theories in computer science: Kleene algebra, coalgebra, and automata theory. I was fascinated not only by their elegant mathematical structures and simple formulations, but also by their vast applications across seemingly unrelated domains. Hence, theory and practice always go hand-in-hand in my research: whenever we discover an application, we seek to perfect its theory; and similarly, when a new theory is developed, we will thoroughly explore its use cases in the real-world.

TopKAT: A Unified View Of Program Logic

Incorrectness logic [12], although simple, has shown great potential for bug detection across various semantical domains [15, 10, 22].

Our work [21] aims to use Kleene algebra with tests (KAT) to provide a simple and abstract semantical foundations for incorrectness logic, unifying its theory in different domains. This task was proven to be impossible: we showed that the theory of KAT is insufficient to encode incorrectness logic, mainly because KAT lacks the relational domain operator. Thus, we devised a simple extension of KAT, named TopKAT, to soundly encode all the propositional proof rules of Hoare and incorrectness logic. TopKAT contains only one more axiom than KAT: there exists a largest element; and unlike KA with domain [17], TopKAT preserves the complexity class of KAT [21].

However, when we dived into the theory of TopKAT, we discovered an unexpected limitation of TopKAT. Despite its power to subsume both propositional Hoare and incorrectness logic, the algebra is incomplete with respect to its relational model. Our followup work [20] resolves this weakness by only looking at the inequalities used to encode incorrectness and Hoare logic, which we named "domain-comparison inequalities". In this work, we used techniques in universal algebra to streamline the definition of reduction [14, 8], enabling us to greatly simplify previous completeness proofs, and also allowing us to prove the relational completeness with regard domain-comparison inequalities. This result has not only demonstrated the effectiveness of reasoning about incorrectness and Hoare logics using TopKAT, but also other logics like reachability logic [11] as well.

In the future, we plan to extend the universal algebra techniques in this work to prove more complicated completeness results. We hope this will yield another compositional framework for completeness proof in Kleene Algebra.

Kleene Algebra With Commutativity Hypothesis

Commutativity hypothesis has long been recognized for its importance in control-flow analysis [6], yet recent work [1] has also established its vital role in relational verification. Contrary to its broad applications, the theory of KA with commutativity hypothesis remains stale; specifically, the decidability of the theory has made no progress since the question was raised by Kozen [6].

Independently, Kuznetsov [9] has shown that Kleene Algebra with commutativity is indeed undecidable. We, on the other hand, has shown the same result without using the induction or right unfolding rule [2]. Our result exhibits a large class of equational theories that are all undecidable when extended commutativity hypothesis, generalizing the result of Kuznetsov.

This work settles a long-standing open problem in Kleene algebra, and also demonstrates the limitation of relational reasoning with algebra of alignment [1]. In fact, we envision a decidable, yet less robust, extension of Kleene Algebra useful for reasoning about alignment problems in relational verifications.

Ongoing Works

CF-GKAT, control flow verification in nearly linear time

One of the hardship to fully verify current and legacy software is to build trust-worthy compilers and decompilers. While numerous ad-hoc techniques made their way into (de)compilers, the verification of these techniques are only viable by experts using proof assistants. Our framework, on the other hand, can not only serve as a theory for formal verification of control-flow manipulation algorithms, but also can be invoked within (de)compilers to verify the correctness of control-flow restructuring process on-the-fly.

These use cases are enabled by guraded Kleene algebra with tests (GKAT) [18], the theoretical foundation of our work. We extended GKAT with common control structures, including break, return, goto, and indicator variables, while preserving its efficiency, soundness, and completeness. These extensions enable us to verify a large class of control-flow restructuring algorithms [19, 3, 5, 7], and its efficiency allows on-the-fly verifications without consuming excessive time or system resources on users' computers. Indeed, we were able to verify decompilations of programs with millions of commands in mere seconds, on a laptop.

Theory and practice of symbolic GKAT

Although GKAT is extremely efficient in some use cases, its efficiency can be improved in many other scenarios. For example, when there is a large amount of primitive test (primitive conditional statements used in if-statement and while-loops), the memory usage and runtime of the original algorithm [18] will blowup exponentially. The large memory usage is typically resolved using derivatives to produce the automaton on-the-fly [4, 16], whereas the long runtime can be optimized using symbolic automaton [13].

Our latest work marries these two ideas, and built a theory of symbolic guarded Kleene coalgebra with tests (sGKCT), where we use category theory to streamline some languages in previous works of symbolic automata, and designed an efficient derivative-based symbolic decision procedure for GKAT. Unlike similar works on KAT [13], the structure of GKAT enables us to export the complex boolean logic into a fast and reliable solvers like z3; further improving the efficiency of our implementation.

This work also characterized the exact complexity of GKAT.

References

[1] Timos Antonopoulos et al. “An Algebra of Alignment for Relational Verification”. In: *Proceedings of the ACM on Programming Languages* 7.POPL (Jan. 2023), 20:573–20:603. DOI: 10.1145/3571213.

[2] Arthur Azevedo de Amorim, Cheng Zhang, and Marco Gaboardi. “Kleene Algebra with Commutativity Conditions Is Undecidable”. Apr. 2024.

[3] Zion Leonahenahe Basque et al. “Ahoy SAILR! There Is No Need to DREAM of C: A Compiler-Aware Structuring Algorithm for Binary Decompilation”. In: ().

[4] Janusz A. Brzozowski. “Derivatives of Regular Expressions”. In: *Journal of the ACM* 11.4 (Oct. 1964), pp. 481–494. ISSN: 0004-5411. DOI: 10.1145/321239.321249.

[5] A.M. Erosa and L.J. Hendren. “Taming Control Flow: A Structured Approach to Eliminating Goto Statements”. In: *Proceedings of 1994 IEEE International Conference on Computer Languages (ICCL’94)*. May 1994, pp. 229–240. DOI: 10.1109/ICCL.1994.288377.

[6] Dexter Kozen. “Kleene Algebra with Tests and Commutativity Conditions”. In: *Tools and Algorithms for the Construction and Analysis of Systems*. Ed. by Gerhard Goos et al. Vol. 1055. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 14–33. ISBN: 978-3-540-61042-7 978-3-540-49874-2. DOI: 10.1007/3-540-61042-1_35.

[7] Dexter Kozen and Maria-Cristina Patron. “Certification of Compiler Optimizations Using Kleene Algebra with Tests”. In: *Computational Logic — CL 2000*. Ed. by G. Goos et al. Vol. 1861. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 568–582. ISBN: 978-3-540-67797-0 978-3-540-44957-7. DOI: 10.1007/3-540-44957-4_38.

[8] Dexter Kozen and Frederick Smith. “Kleene Algebra with Tests: Completeness and Decidability”. In: *Computer Science Logic*. Ed. by Gerhard Goos et al. Vol. 1258. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 244–259. ISBN: 978-3-540-63172-9 978-3-540-69201-0. DOI: 10.1007/3-540-63172-0_43.

[9] Stepan L. Kuznetsov. “On the Complexity of Reasoning in Kleene Algebra with Commutativity Conditions”. In: *Theoretical Aspects of Computing – ICTAC 2023*. Ed. by Erika Ábrahám, Clemens Dubslaff, and Silvia Lizeth Tapia Tarifa. Cham: Springer Nature Switzerland, 2023, pp. 83–99. ISBN: 978-3-031-47963-2. DOI: 10.1007/978-3-031-47963-2_7.

[10] Quang Loc Le et al. “Finding Real Bugs in Big Programs with Incorrectness Logic”. In: *Proceedings of the ACM on Programming Languages* 6.OOPSLA1 (Apr. 2022), pp. 1–27. ISSN: 2475-1421. DOI: 10.1145/3527325.

[11] Nico Naus et al. *Reachability Logic for Low-Level Programs*. Mar. 2022. DOI: 10.48550/arXiv.2204.00076. arXiv: 2204.00076 [cs].

[12] Peter W. O’Hearn. “Incorrectness Logic”. In: *Proceedings of the ACM on Programming Languages* 4.POPL (Jan. 2020), pp. 1–32. ISSN: 2475-1421, 2475-1421. DOI: 10.1145/3371078.

[13] Damien Pous. “Symbolic Algorithms for Language Equivalence and Kleene Algebra with Tests”. In: *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL ’15. New York, NY, USA: Association for Computing Machinery, Jan. 2015, pp. 357–368. ISBN: 978-1-4503-3300-9. DOI: 10.1145/2676726.2677007.

[14] Damien Pous, Jurriaan Rot, and Jana Wagemaker. “On Tools for Completeness of Kleene Algebra with Hypotheses”. In: *Relational and Algebraic Methods in Computer Science: 19th International Conference, RAMiCS 2021, Marseille, France, November 2–5, 2021, Proceedings*. Berlin, Heidelberg: Springer-Verlag, Nov. 2021, pp. 378–395. ISBN: 978-3-030-88700-1. DOI: 10.1007/978-3-030-88701-8_23.

[15] Azalea Raad et al. “Local Reasoning About the Presence of Bugs: Incorrectness Separation Logic”. In: *Computer Aided Verification*. Ed. by Shuvendu K. Lahiri and Chao Wang. Vol. 12225. Cham: Springer International Publishing, 2020, pp. 225–252. ISBN: 978-3-030-53290-1 978-3-030-53291-8. DOI: 10.1007/978-3-030-53291-8_14.

[16] Todd Schmid et al. *Guarded Kleene Algebra with Tests: Coequations, Coinduction, and Completeness*. May 2021. DOI: 10.4230/LIPIcs.ICALP.2021.142. arXiv: 2102.08286 [cs].

[17] Igor Sedlár. “On the Complexity of Kleene Algebra with Domain”. In: *Relational and Algebraic Methods in Computer Science: 20th International Conference, RAMiCS 2023, Augsburg, Germany, April 3–6, 2023, Proceedings*. Berlin, Heidelberg: Springer-Verlag, Apr. 2023, pp. 208–223. ISBN: 978-3-031-28082-5. DOI: 10.1007/978-3-031-28083-2_13.

[18] Steffen Smolka et al. “Guarded Kleene Algebra with Tests: Verification of Uninterpreted Programs in Nearly Linear Time”. In: *Proceedings of the ACM on Programming Languages* 4.POPL (Jan. 2020), pp. 1–28. ISSN: 2475-1421. DOI: 10.1145/3371129.

[19] Khaled Yakdan et al. “No More Gotos: Decompilation Using Pattern-Independent Control-Flow Structuring and Semantics-Preserving Transformations”. In: *Proceedings 2015 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2015. ISBN: 978-1-891562-38-9. DOI: 10.14722/ndss.2015.23185.

[20] Cheng Zhang, Arthur Azevedo de Amorim, and Marco Gaboardi. *Domain Reasoning in TopKAT*. Apr. 2024. DOI: 10.4230/LIPIcs.ICALP.2024.133. arXiv: 2404.18417 [cs].

[21] Cheng Zhang, Arthur Azevedo de Amorim, and Marco Gaboardi. “On Incorrectness Logic and Kleene Algebra with Top and Tests”. In: *Proceedings of the ACM on Programming Languages* 6.POPL (Jan. 2022), 29:1–29:30. DOI: 10.1145/3498690.

[22] Linpeng Zhang and Benjamin Lucien Kaminski. “Quantitative Strongest Post: A Calculus for Reasoning about the Flow of Quantitative Information”. In: *Proceedings of the ACM on Programming Languages* 6.OOPSLA1 (Apr. 2022), 87:1–87:29. DOI: 10.1145/3527331.