Research Statement — Christopher Hardin

Brief summary of research interests. I am interested in mathematical logic and theoretical computer science. Specifically, I am interested in program logics, particularly Kleene algebra with tests and dynamic logic. (Within Kleene algebra, I am most interested in describing the theory of relational Kleene algebras.) I am also interested in epistemic logic.

My planned research after graduation involves two areas: the Horn theory of relational Kleene algebra, and revelation in epistemic logic. I will discuss the Kleene algebra aspects first, which will require me to give some background information, and then discuss the epistemic logic, at a level that will not require any background.

Background and Motivation

Kleene algebra (KA) arises in many areas of computer science, such as automata theory, the design and analysis of algorithms, dynamic logic, and program semantics. Kleene algebra with tests (KAT) is essentially the result of combining KA with Boolean algebra, and this adaptation is particularly useful in dynamic logic and program semantics. For a detailed introduction see [6].

Before defining KA or KAT explicitly, I will give an example that is particularly important in motivating my research. Consider the set R of all binary relations on a set X. We will think of R as an algebra with constants $0, 1$, and operations $+, \cdot, ^*$, defined as follows.

$$\begin{aligned} 0 &= \emptyset \\ 1 &= \text{the identity relation on } X \\ p + q &= p \cup q \\ p \cdot q &= pq = \text{the relational composition of } p \text{ with } q \\ p^* &= \bigcup_{n \geq 0} p^n = \text{the reflexive transitive closure of } p \end{aligned}$$

An algebra of binary relations with operations defined as above is a *relational Kleene algebra* (RKA). Letting $B \subseteq R$ be those relations that are subsets of the identity, we can additionally define a unary operator $^-$ on B by

$$\overline{c} = 1 \setminus c.$$

One can verify that $(B, +, \cdot, ^-, 0, 1)$ forms a Boolean algebra. With this additional structure, R is an example of a *relational Kleene algebra with tests* (RKAT).

The above algebraic operations on R arise naturally in the study of program semantics. If we think of X as the set of states that some machine can be in, then a program p may be interpreted as a binary relation on X; specifically, we take $(x, y) \in p$ to mean, "If the machine is in state x when we run program p, then the machine could finish in state y." Under this interpretation, pq corresponds to "run p, then q". The other operations are needed to capture other common constructs; I must omit explanations, but here are some

examples: a "no-op" translates to 1; "fail" translates to 0; "if b then p else q" translates to $bp + \bar{b}q$; "while b do p" translates to $(bp)^*\bar{b}$.

Not every KA is required to be built from relations as R was. Formally, a KA is an algebraic structure $(K, +, \cdot, ^*, 0, 1)$ that forms an idempotent semiring under $+, \cdot, 0, 1$, and such that p^*q and qp^* are the $\leq$-least solutions to $q + px \leq x$ and $q + xp \leq x$, respectively, where $\leq$ is the partial order $p \leq q \stackrel{\text{def}}{\iff} p + q = q$. For our purposes here, all that must be understood about this definition is that it is first-order and captures many (but not all) of the properties of relational Kleene algebras—*e.g.*, $\cdot$ is associative and distributes over $+$. A KAT is a structure $(K, B, +, \cdot, ^*, ^{\bar{}}, 0, 1)$ where $(K, +, \cdot, ^*, 0, 1)$ is a KA and $(B, +, \cdot, ^{\bar{}}, 0, 1)$ is a Boolean subalgebra.

The most common intuition about * is that $p^* = \sup_{n \geq 0} p^n = 1 + p + p^2 + \cdots$. However, this property does not follow from the KA axioms, and must be postulated separately: we say that a KA is $*$-*continuous* if it satisfies the infinitary condition

$$pq^*r = \sup_{n \geq 0} pq^n r.$$

We use KA* (KAT*) to refer to $*$-continuous Kleene algebras (Kleene algebras with tests). While there are many Kleene algebras of interest that are not relational, all Kleene algebras that arise in practice are $*$-continuous (though non-$*$-continuous Kleene algebras do exist). In particular, all RKAs and RKATs are $*$-continuous. The drawback of $*$-continuity is that it is not a first-order axiomatizable condition, and accordingly introduces a great deal of complexity, as we shall see.

A *Horn formula* is a formula

$$(s_1 = t_1 \ \& \ \cdots \ \& \ s_k = t_k) \rightarrow s = t,$$

where s_i, t_i, s, t are terms. Given a class Γ of algebras (such as KA or RKAT), the *Horn theory* of Γ is the set of all Horn formulas in the language of Γ that are valid in every algebra in Γ.

Horn formulas are important in universal algebra, but are particularly important in KAT, where they are used to reason about program semantics: the hypotheses of a Horn formula capture (as much as possible or needed) the intended semantics of individual program fragments, while the conclusion expresses some kind of correctness condition. This, along with the fact that relational models are of primary interest in program semantics, gives the Horn theory of RKAT practical relevance. For example, Horn formulas with hypotheses of the form $pq = qp$ (which express that the order of execution of certain operations does not matter) have been used to verify compiler optimizations that eliminate redundant code from loops [7].

Existing Results

Unfortunately, the axioms of KAT are not strong enough to prove all Horn formulas that are valid in RKAT. In fact, I have shown that the Horn theory of RKAT is Π^1_1-complete (highly undecidable) [1]; the Horn theory of KAT* was already known to be Π^1_1-complete.

The Π^1_1-completeness of the Horn theory of RKAT yields a predicament: there are practical gains to be made by understanding this theory—*e.g.*, for what it tells us about program semantics—yet its complex-

ity forbids a thorough understanding. Fortunately, we can still manage to understand the theory in many significant ways, if not fully. This is what most of my research is about.

In [2], we identify certain Horn formulas whose validity (in any of KAT, KAT*, and RKAT) is equivalent to the validity of some equation, yielding an efficient reduction of a fragment of the Horn theory to the equational theory, which is decidable (requiring polynomial space). It was already known that one could eliminate hypotheses of the form $s = 0$ in this way; this paper shows how to also eliminate hypotheses $cp = c$ where c is a test and p is an atomic program symbol. Hypotheses of this form are common in practice, because they express the redundancy of code in certain situations (for example, the redundancy of "let $x := y$" when x already equals y), and subsequently are useful in eliminating redundant code and verifying compiler optimizations.

In [3], I define the notion of a *simple* Horn formula: one in which $*$ may appear only on the left-hand side of hypotheses $s_i \leq t_i$ and only on the right-hand side of the conclusion $s \leq t$. (We treat any equation $p = q$ as a pair of inequalities $p \leq q$ and $q \leq p$.) I show that the Horn theory of KAT*, restricted to simple formulas, is Σ^0_1-complete, and that the same holds for RKAT. If we weaken the condition to allow $*$ anywhere in the conclusion, the complexity in each case goes to Π^0_2-complete. Beyond that, it jumps to Π^1_1-complete.

More recently, I have developed an infinitary proof system for the Horn theory of RKAT that is sound and complete [4]. The proofs are well-founded infinitely branching trees, with each node being associated with a finite automaton. A very similar system, in which the automata essentially have their internal structure purged from them, is sound and complete for the Horn theory of KAT*. This "trees of automata" approach sheds some light on the relationship between RKAT and KAT*. In particular, it exhibits that relational validity has a strong combinatorial flavor (from how the internal structure of the automata evolves in the proof tree) that is absent from KAT*.

Further Questions

The further questions that I would like to look at are: Can this approach help us find a finitary axiomatization of the Horn theory of RKAT relative to KAT*? If we take the set of simple Horn formulas that are valid over RKAT, which is computably enumerable, and combine it with the KAT* axioms, does that give us the Horn theory of RKAT? If no finite or computably enumerable set of axioms suffices in this way, can we prove this fact?

Revelation in Epistemic Logic

I would also like to look at revelation in epistemic logic, namely, how knowledge evolves in systems where agents reveal facts to each other by sending messages. As a very straightforward example, let us assume that you don't know the current weather in Ithaca, New York. Now, suppose that I tell you, "It is cold and raining in Ithaca." Your knowledge has changed: you now know it is cold and raining in Ithaca. If I know that you heard me, then my knowledge has also changed: I now know that you know the weather in Ithaca, while I did not previously know this. A more subtle example: suppose Alice tells Bob, "There is a fly on

your back and you do not know this." While her statement may be truthful at the moment she says it, as soon as Bob hears it, he knows about the fly on his back, so it is no longer true that he does not know it. So the knowledge is not always incorporated in the most obvious way; in particular, just because someone you trust tells you something—"You do not know there is a fly on your back"—does not always mean that you will know it after you hear it.

There are existing frameworks for studying evolving knowledge; the method I propose is more syntactic than the approaches I have seen so far. I have done some work on this already, using propositional dynamic logic (PDL) as the base system. PDL is a multimodal logic in which the modalities—called *programs*—themselves form a KAT (in the semantics, they are in fact a RKAT); for more information see [5]. It is ordinarily used for reasoning about programs (for example, the PDL formula $[p]\varphi$ generally means "After running p, φ must hold") but can accommodate modalities of knowledge without modification (so that $[k_i]\varphi$ can mean "Agent i knows φ"—we generally think of the system as having some fixed set of agents numbered $1, \ldots, n$, two which we ascribe knowledge). My goal is to develop a logic of revelation in PDL that can be applied to new problems in a purely deductive fashion, without having to construct any models specific to the problem. Furthermore, since one can also work with program semantics in PDL, I would like to integrate the logic of revelation with the logic of programs, so that one would be able, say, to verify the correctness of a protocol that involves message passing, all within PDL.

The most primitive form of revelation is the *public announcement*, which essentially consists of some external agent (that is, an agent not formally treated as part of the system) telling some formula to the agents, such that every agent can hear it, and every agent knows that every agent can hear it, and every agent knows that every agents knows.... For any formula φ, we have a program a^φ such that that $[a^\varphi]\psi$ means "After φ is publically announced, ψ must hold." I have proposed an axiomatization for public announcements (which is different from the only treatment of public announcements that I have seen [8]), but I see this only as the starting point. A small modification allows us to make announcements which are restricted to a subset of the agents. Further modifications should hopefully be able to capture notions such as private communication between two agents, unreliable delivery, unreliable privacy, *etc*.

A concrete goal for this research would be to express and verify the standard solution to the "Dining Cryptographers" problem within the logic. (The dining cryptographers problem concerns the ability to send a message anonymously in a world where each physical transmission is not anonymous.) If this can be done, it will demonstrate that the logic has practical value to computer science, as a new tool for formal verification of protocols. A logic that can handle notions of knowledge in addition to ordinary program logic would be beneficial, since many protocols can be expressed naturally using knowledge (*e.g.*, "Send the bit until I know that the receiver knows the value of the bit").

Undergraduate Research

Epistemic logic has a flavor that I think can appeal to undergraduates. It could make an interesting topic for a Winter Term course, and I think there are questions that would be accessible to undergraduates, though they would be outside what I would consider the most elegant part of epistemic logic, which is well understood.

Kleene algebra is less accessible, but a particularly bright undergraduate at Cornell has recently done

original research in the area.

References

[1] Chris Hardin and Dexter Kozen. On the Complexity of the Horn Theory of REL. Technical Report 2003-1896, Computer Science Department, Cornell University, May 2003.

[2] Chris Hardin and Dexter Kozen. On the Elimination of Hypotheses in Kleene Algebra with Tests. Technical Report 2002-1879, Computer Science Department, Cornell University, October 2002.

[3] Chris Hardin. How the Location of * Influences Complexity in Kleene Algebra with Tests. In F. Baader and A. Voronkov, editors, *Proc. 11th Int. Conf. Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2004)*, volume 3452 of *Lecture Notes in Artificial Intelligence*, pages 224–239, Montevideo, Uruguay, March 2005. Springer-Verlag.

[4] Chris Hardin. Proof Theory for Kleene Algebra. To be presented at LICS 2005.

[5] David Harel, Dexter Kozen, and Jerzy Tiuryn. *Dynamic Logic*. MIT Press, Cambridge, 2000.

[6] Dexter Kozen. Kleene algebra with tests. Transactions on Programming Languages and Systems, May 1997, 427–443.

[7] Dexter Kozen and Maria-Cristina Patron. Certification of compiler optimizations using Kleene algebra with tests. In J. Lloyd, V. Dahl, U. Furbach, M. Kerber, K.-K. Lau, C. Palamidessi, L. M. Pereira, Y. Sagiv, and P. J. Stuckey, eds, *Proc. 1st Int. Conf. Computational Logic (CL2000)*, volume 1861 of Lecture Notes in Artificial Intelligence, Springer-Verlag, London, July 2000, 568–582.

[8] Hans van Ditmarsch, Wiebe van der Hoek and Barteld P. Kooi. Playing Cards with Hintikka: An introduction in Dynamic Epistemic Logic. Phinews, Volume 6, October 2004, 6–32.