Introduction to Kleene Algebra	**Lecture 1**
CS786 Spring 2002	**January 22, 2002**

1 Course Roadmap

Kleene algebra (KA) is an algebraic system that captures axiomatically the properties of a natural class of structures arising in logic and computer science. It is named for Stephen Cole Kleene (1909–1994), who among his many other achievements invented finite automata and regular expressions, structures of fundamental importance in computer science. Kleene algebra is the algebraic theory of these objects, although it has many other natural and useful interpretations.

Kleene algebras arise in various guises in many contexts: relational algebra [26, 33, 27], semantics and logics of programs [16, 28], automata and formal language theory [23, 24], and the design and analysis of algorithms [1, 25, 14, 18]. Many authors have contributed to the development of Kleene algebra over the years [2, 3, 4, 5, 6, 7, 10, 12, 15, 16, 17, 19, 22, 24, 29, 30, 31, 32], to name a few. There are various competing axiomatizations, and one topic of our study will be to understand the relationships between these definitions.

In semantics and logics of programs, Kleene algebra forms an essential component of Propositional Dynamic Logic (PDL) [11], in which it is mixed with Boolean algebra and modal logic to give a theoretically appealing and practical system for reasoning about computation at the propositional level. From a practical point of view, many simple program manipulations, such as loop unwinding and basic safety analysis, do not require the full power of PDL, but can be carried out in a purely equational subsystem using the axioms of Kleene algebra. The Boolean algebra component is an essential ingredient, since it is needed to model conventional programming constructs such as conditionals and **while** loops that rely on Boolean tests. However, for many applications, the modal component is not essential. We define later on a variant of Kleene algebra, called *Kleene algebra with tests* (KAT), for reasoning equationally with these constructs. We will show that KAT provides an equational approach to program verification that subsumes traditional approaches such as Hoare logic.

Cohen has studied Kleene algebra in the presence of extra Boolean and commutativity conditions. He has given several practical examples of the use of Kleene algebra in program verification, such as lazy caching [8] and concurrency control [9]. In addition, Kleene algebra has been used for verifying low-level compiler optimizations [21], data restructuring operations in parallelizing compilers [], and pointer analysis [].

Much of the basic algebraic theory of KA was developed by John Horton Conway in his 1971 monograph [10]. Unfortunately, this delightful little volume is out of print. What is worse, when Chapman and Hall went out of business, they sold the rights to all their titles to two publishing companies, but neither one claims any knowledge of this title. Thus there is little chance that Conway's monograph will be reprinted anytime soon, despite significant renewed interest in the topic.

We begin our study by describing several concrete examples of Kleene algebras. These will serve as motivating examples to provide intuition about the properties we are trying to capture axiomatically with the formal definition. We will conclude this lecture with the formal definition of a Kleene algebra and derive some basic properties that follow from these axioms.

2 Examples of Kleene Algebras

A Kleene algebra consists of a set K with distinguished binary operations $+$ and $\cdot$, unary operation *, and constants 0 and 1 with certain properties. The intuitive meaning of the operations depends on the model; however, we can at least say that the operator * typically involves some notion of *iteration*. The * operator is the most interesting aspect of Kleene algebra. For example, it allows us to express and reason about properties of simple looping constructs in programming languages.

Here are three classes of models that motivate the definition of Kleene algebra.

2.1 Language-Theoretic Models

Let Σ^* denote the set of finite-length strings over a finite alphabet Σ, including the null string ε. Define the following constants and operations on subsets of Σ^*:

$$A + B \stackrel{\text{def}}{=} A \cup B \tag{1}$$

$$A \cdot B \stackrel{\text{def}}{=} \{xy \mid x \in A,\ y \in B\} \tag{2}$$

$$0 \stackrel{\text{def}}{=} \varnothing \tag{3}$$

$$1 \stackrel{\text{def}}{=} \{\varepsilon\}. \tag{4}$$

Thus the operation $\cdot$, applied to two sets of strings A and B, produces the set of all strings obtained by concatenating a string from A with a string from B, in that order. The operator symbol $\cdot$ is often omitted, and we just write AB for $A \cdot B$.

These operations have several agreeable properties. For example, $\cdot$ distributes over $+$ on both sides, in the sense that $A(B+C) = AB+AC$ and $(A+B)C = AC+BC$; the element 0 is a both a left and right identity for $+$ in the sense that $0 + A = A + 0 = A$; and the element 1 is both a left and right identity for $\cdot$ in the sense that $1A = A1 = A$.

Now define the powers of A with respect to $\cdot$ inductively:

$$A^0 \stackrel{\text{def}}{=} \{\varepsilon\}$$
$$A^{n+1} \stackrel{\text{def}}{=} A \cdot A^n.$$

The unary operation * on sets of strings is defined as follows:

$$\begin{aligned} A^* &\stackrel{\text{def}}{=} \bigcup_{n \geq 0} A^n \\ &= \{x_1 \cdots x_n \mid n \geq 0 \text{ and } x_i \in A,\ 1 \leq i \leq n\}. \end{aligned} \tag{5}$$

Thus A^* is the union of all powers of A; equivalently, A^* consists of all strings obtained by concatenating together any finite collection of strings from A in any order. By convention, the concatenation of the empty set of strings is ε; this is the case $n = 0$ in (5). Thus ε is always a member of A^* for any A, including $\varnothing$. The operation * is known as *Kleene asterate*.

Any subset of the full powerset of Σ^* containing $\varnothing$ and $\{\varepsilon\}$ and closed under the operations of $\cup$, $\cdot$, and * is a Kleene algebra, and is a subalgebra of the full powerset algebra. One such subalgebra of particular significance is the algebra of *regular sets*. This is the smallest subalgebra containing all sets $\{a\}$ for $a \in \Sigma$.

As is well known, the regular sets are also the sets of strings accepted by finite-state automata, or finite-state transition systems with an acceptance condition. The equivalence of these two representations was proved in Kleene's original paper [15] and is known in this context as *Kleene's theorem*. A proof of this result can be found in any introductory text in automata and computability; see for example [13, 20].

2.2 Relational Algebras

Another useful interpretation involves binary relations on a set X. Recall that a *binary relation* on a set X is just a set of ordered pairs of elements of X. Thus a binary relation on X is a subset of $X \times X$.

The set of all binary relations on a set X forms a Kleene algebra under the following definitions of the operators. We again interpret $+$ as set union. The multiplication operation $\cdot$ is interpreted as *relational composition*

$$R \circ S \stackrel{\text{def}}{=} \{(x, z) \mid \exists y \in X\ (x, y) \in R \text{ and } (y, z) \in S\}.$$

If we view R as a set of labeled directed edges on X, then there is an edge from x to z labeled $R \circ S$ iff there is a node y such that there is an edge from x to y labeled R and and edge from y to z labeled S.

(insert picture here)

The element 0 is the null relation $\varnothing$, and 1 as the identity relation

$$1 \stackrel{\text{def}}{=} \{(x,x) \mid x \in X\}.$$

The operation * gives the *reflexive transitive closure* of a relation. Recall that a relation R is *reflexive* if $(x,x) \in R$ for all $x \in X$; that is, if R includes the identity relation as a subset. The relation R is *transitive* if $(x,z) \in R$ whenever both $(x,y) \in R$ and $(y,z) \in R$; in other words, R is transitive if $R \circ R \subseteq R$. The smallest reflexive and transitive relation containing R is called the *reflexive transitive closure* of R and is denoted R^*. This notation fortuitously coincides with the definition of the * operation as giving the sum of all finite powers of R as above.

$$R^* = \bigcup_{n \geq 0} R^n,$$

where

$$\begin{aligned} R^0 &\stackrel{\text{def}}{=} \{(x,x) \mid x \in X\} \\ R^{n+1} &\stackrel{\text{def}}{=} R \circ R^n. \end{aligned}$$

Equivalently, there is an R^* edge from x to z iff there is an R-path of length 0 or more from x to z.

(insert picture here)

A *relational Kleene algebra* is any subset of $2^{X \times X}$ closed under these operations. These models are useful in programming language semantics, because they can be used to represent the input/output relations of programs.

2.3 The min,+ Algebra

Here is a rather unusual model that turns out to be useful in shortest path algorithms in graphs. This algebra is called the *min,+ algebra*, also known as the *tropical algebra*. The domain is the set $\mathbb{R}_+ \cup \{\infty\}$ of nonnegative reals with an added infinite element ∞. The Kleene algebra operation $+$ is interpreted as the operation min giving the minimum of two elements in the natural order on $\mathbb{R}_+ \cup \{\infty\}$. The Kleene algebra operation $\cdot$ is interpreted as $+$ in $\mathbb{R}_+ \cup \{\infty\}$; the usual definition of $+$ on $\mathbb{R}_+$ is extended to include ∞ in the natural way. The Kleene algebra constants 0 and 1 are interpreted as ∞ and the real number 0, respectively.

The * operation on this algebra is not very interesting: $x^* = 1$ (= the real number 0) for any x. However, the * of matrices over this algebra is quite interesting: it gives a way of calculating the shortest path between any two points in a finite directed graph.

References

[1] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, Mass., 1975.

[2] S. Anderaa. On the algebra of regular expressions. Appl. Math., Harvard Univ., 1965. Cambridge, Mass., 1–18.

[3] K. V. Archangelsky. A new finite complete solvable quasiequational calculus for algebra of regular languages. Manuscript, Kiev State University, 1992.

[4] Roland Carl Backhouse. *Closure Algorithms and the Star-Height Problem of Regular Languages*. PhD thesis, Imperial College, London, U.K., 1975.

[5] Stephen L. Bloom and Zoltán Ésik. Equational axioms for regular sets. *Math. Struct. Comput. Sci.*, 3:1–24, 1993.

[6] Maurice Boffa. Une remarque sur les systèmes complets d'identités rationnelles. *Informatique Théoretique et Applications/Theoretical Informatics and Applications*, 24(4):419–423, 1990.

[7] Ernie Cohen. Hypotheses in Kleene algebra. Available as ftp://ftp.telcordia.com/pub/ernie/research/homepage.html, April 1994.

[8] Ernie Cohen. Lazy caching. Available as ftp://ftp.telcordia.com/pub/ernie/research/homepage.html, 1994.

[9] Ernie Cohen. Using Kleene algebra to reason about concurrency control. Available as ftp://ftp.telcordia.com/pub/ernie/research/homepage.html, 1994.

[10] John Horton Conway. *Regular Algebra and Finite Machines*. Chapman and Hall, London, 1971.

[11] Michael J. Fischer and Richard E. Ladner. Propositional dynamic logic of regular programs. *J. Comput. Syst. Sci.*, 18(2):194–211, 1979.

[12] P. V. Gorshkov. Rational data structures and their applications. *Cybernetics*, 25(6):760–765, Nov.–Dec. 1989.

[13] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

[14] Kazuo Iwano and Kenneth Steiglitz. A semiring on convex polygons and zero-sum cycle problems. *SIAM J. Comput.*, 19(5):883–901, 1990.

[15] Stephen C. Kleene. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 3–41. Princeton University Press, Princeton, N.J., 1956.

[16] Dexter Kozen. On induction vs. *-continuity. In Kozen, editor, *Proc. Workshop on Logic of Programs*, volume 131 of *Lecture Notes in Computer Science*, pages 167–176, New York, 1981. Springer-Verlag.

[17] Dexter Kozen. On Kleene algebras and closed semirings. In Rovan, editor, *Proc. Math. Found. Comput. Sci.*, volume 452 of *Lecture Notes in Computer Science*, pages 26–47, Banska-Bystrica, Slovakia, 1990. Springer-Verlag.

[18] Dexter Kozen. *The Design and Analysis of Algorithms.* Springer-Verlag, New York, 1991.

[19] Dexter Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Infor. and Comput.*, 110(2):366–390, May 1994.

[20] Dexter Kozen. *Automata and Computability.* Springer-Verlag, New York, 1997.

[21] Dexter Kozen and Maria-Cristina Patron. Certification of compiler optimizations using Kleene algebra with tests. In John Lloyd, Veronica Dahl, Ulrich Furbach, Manfred Kerber, Kung-Kiu Lau, Catuscia Palamidessi, Luis Moniz Pereira, Yehoshua Sagiv, and Peter J. Stuckey, editors, *Proc. 1st Int. Conf. Computational Logic (CL2000)*, volume 1861 of *Lecture Notes in Artificial Intelligence*, pages 568–582, London, July 2000. Springer-Verlag.

[22] Daniel Krob. A complete system of B-rational identities. *Theoretical Computer Science*, 89(2):207–343, October 1991.

[23] Werner Kuich. The Kleene and Parikh theorem in complete semirings. In T. Ottmann, editor, *Proc. 14th Colloq. Automata, Languages, and Programming*, volume 267 of *Lecture Notes in Computer Science*, pages 212–225, New York, 1987. EATCS, Springer-Verlag.

[24] Werner Kuich and Arto Salomaa. *Semirings, Automata, and Languages.* Springer-Verlag, Berlin, 1986.

[25] Kurt Mehlhorn. *Data Structures and Algorithms 2: Graph Algorithms and NP-Completeness.* EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1984.

[26] K. C. Ng. *Relation Algebras with Transitive Closure.* PhD thesis, University of California, Berkeley, 1984.

[27] K. C. Ng and A. Tarski. Relation algebras with transitive closure, abstract 742-02-09. *Notices Amer. Math. Soc.*, 24:A29–A30, 1977.

[28] Vaughan Pratt. Dynamic algebras as a well-behaved fragment of relation algebras. In D. Pigozzi, editor, *Proc. Conf. on Algebra and Computer Science*, volume 425 of *Lecture Notes in Computer Science*, pages 77–110, Ames, Iowa, June 1988. Springer-Verlag.

[29] Vaughan Pratt. Action logic and pure induction. In J. van Eijck, editor, *Proc. Logics in AI: European Workshop JELIA '90*, volume 478 of *Lecture Notes in Computer Science*, pages 97–120, New York, September 1990. Springer-Verlag.

[30] V. N. Redko. On defining relations for the algebra of regular events. *Ukrain. Mat. Z.*, 16:120–126, 1964. In Russian.

[31] Jacques Sakarovitch. Kleene's theorem revisited: A formal path from Kleene to Chomsky. In A. Kelemenova and J. Keleman, editors, *Trends, Techniques, and Problems in Theoretical Computer Science*, volume 281 of *Lecture Notes in Computer Science*, pages 39–50, New York, 1987. Springer-Verlag.

[32] Arto Salomaa. Two complete axiom systems for the algebra of regular events. *J. Assoc. Comput. Mach.*, 13(1):158–169, January 1966.

[33] Alfred Tarski. On the calculus of relations. *J. Symb. Logic*, 6(3):65–106, 1941.