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 [33, 34, 40], semantics and logics of programs [19, 35], automata and formal language theory [27, 28], and the design and analysis of algorithms [1, 17, 21, 30]. Many authors have contributed to the development of Kleene algebra over the years: [2, 4, 5, 7–9, 12, 15, 18–20, 22, 26, 28, 36–39], 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) [13], 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 will define later 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. Quite recently, KAT has been adapted to form NetKAT, a language and logic for reasoning about packet-switching networks [3, 14].

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 [10] and concurrency control [11]. In addition, Kleene algebra has been used for verifying low-level compiler optimizations [25], data restructuring operations in parallelizing compilers [6, 29], pointer analysis [31, 32], and static analysis [24].

Much of the basic algebraic theory of KA was developed by John Horton Conway in his 1971 monograph [12]. This volume was originally published by Chapman and Hall and was out of print for many years, but due to recent renewed interest in the topic, it reappeared in a paperback edition by Dover in 2012.

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.

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.

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:

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

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

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

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 sequence of strings is ε; this is the case $n = 0$ in (5). Thus ε is always a member of A^* for any A, including $A = \varnothing$. 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 [18] 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 [16, 23].

Relational Algebras

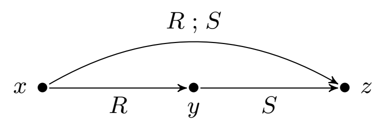
Another useful interpretation involves binary relations on a set X. Recall that a *binary relation* on a set X is just a set of ordered pairs of elements of X. Thus a binary relation on X is a subset of

$$X \times X = \{(x, y) \mid x, y \in X\}.$$

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*

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

(A common alternative notation is $S \circ R$. Note that with this notation, the order of R and S is reversed. This is for consistency with the usage of the same notation for functional composition $(g \circ f)(x) = g(f(x))$.) If we view R as a set of labeled directed edges on X, then there is an edge from x to z labeled $R\ ;S$ iff there exists a node y, an edge from x to y labeled R, and an edge from y to z labeled S.



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

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

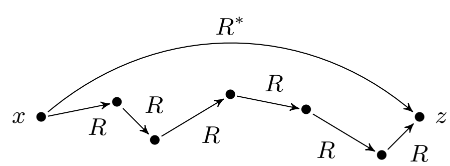
The operation * gives the *reflexive transitive closure* of a relation. Recall that a relation S is *reflexive* if $(x,x) \in S$ for all $x \in X$; that is, if S includes the identity relation as a subset. The relation S is *transitive* if $(x,z) \in S$ whenever $(x,y) \in S$ and $(y,z) \in S$; in other words, S is transitive if $S\ ;S \subseteq S$. 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 the sum of all finite powers of R, as with language models.

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

where

$$R^0 \stackrel{\text{def}}{=} \{(x,x) \mid x \in X\} \qquad\qquad R^{n+1} \stackrel{\text{def}}{=} R\ ;R^n.$$

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.



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.

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 semiring.* The domain is the set $\mathbb{R}_+ \cup \{\infty\}$ of nonnegative reals with an additional 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:

$$x + \infty \;=\; \infty + x \;=\; \infty + \infty \;=\; \infty.$$

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.

The $(\max, \cdot)$ Algebra

The domain of this algebra is the unit interval $[0, 1]$. The Kleene algebra operation $+$ is interpreted as max giving the maximum of two elements in the natural order on $[0, 1]$. The operation $\cdot$ is interpreted as ordinary multiplication. The Kleene algebra constants 0 and 1 are the real numbers 0 and 1, respectively. This algebra is useful in error-correcting codes and hidden Markov models using an algorithm known as the *Viterbi algorithm*.

The $(\max, \cdot)$ algebra and the $(\min, +)$ algebra are isomorphic under the map $x \mapsto -\log x$.

Axioms of Kleene Algebra

Now we give the formal definition of a Kleene algebra and derive some basic consequences.

Semigroups and Monoids

A *semigroup* is an algebraic structure $(S, \cdot)$, where S is a set and $\cdot$ is an associative binary operation on S, which means that $x \cdot (y \cdot z) = (x \cdot y) \cdot z$ for all $x, y, z \in S$. This allows us to write $x \cdot y \cdot z$ without ambiguity. A *monoid* is an algebraic structure $(M,\ \cdot,\ 1)$ where $(M,\ \cdot)$ is a semigroup and 1 is a distinguished element of M that is both a left and right identity for $\cdot$ in the sense that $1 \cdot x = x \cdot 1 = x$ for all $x \in M$.

For semigroups and monoids written multiplicatively, we often omit the operator $\cdot$ in expressions, writing xy for $x \cdot y$.

Example 1. *The following are common examples of monoids:*

1. $(\Sigma^*,\ \cdot,\ \varepsilon)$*, where* Σ^* *is the set of finite-length strings over an alphabet* Σ*,* $\cdot$ *is concatenation of strings, and* ε *is the null string;*
2. $(2^{\Sigma^*},\ \cdot,\ \{\varepsilon\})$*, where* $\cdot$ *is set concatenation* (2)*;*
3. $(2^{\Sigma^*},\ \cup,\ \varnothing)$*, where* 2^{Σ^*} *is the powerset or set of all subsets of* Σ^**,* $\cup$ *is set union, and* $\varnothing$ *is the empty set;*
4. $(\mathbb{N},\ +,\ 0)$*, where* $\mathbb{N}$ *is the set of natural numbers* $\{0, 1, 2, \ldots\}$*;*
5. $(\mathbb{N},\ \cdot,\ 1)$*;*
6. $(\mathbb{N}^n,\ +,\ \overline{0})$*, where* $\mathbb{N}^n$ *is the Cartesian product of* n *copies of* $\mathbb{N}$*,* $+$ *is vector addition, and* $\overline{0}$ *is the zero vector;*
7. $(\mathbb{R}_+ \cup \{\infty\},\ \min,\ \infty)$*, where* $\mathbb{R}_+$ *denotes the set of nonnegative real numbers,* ∞ *is a special infinite element greater than all real numbers, and* min *gives the minimum of two elements;*

8. $(R^{n\times n},\ \cdot,\ I)$, *where* $R^{n\times n}$ *denotes the set of* $n\times n$ *matrices over a ring* R, $\cdot$ *is ordinary matrix multiplication, and* I *is the identity matrix;*

9. $(X\to X,\ \circ,\ \mathsf{id})$, *where* $X\to X$ *denotes the set of all functions from a set* X *to itself,* $\circ$ *is function composition, and* id *is the identity function.*

Examples 3–7 are *commutative* monoids, which means that $xy = yx$ for all x, y. Example 8 is never commutative for any nontrivial ring R and $n \geq 2$. Example 9 is never commutative for any X with at least 2 elements.

Idempotent Semirings

A *semiring* is an algebraic structure $(S,\ +,\ \cdot,\ 0,\ 1)$ such that

- $(S,\ +,\ 0)$ is a commutative monoid,
- $(S,\ \cdot,\ 1)$ is a monoid,
- $\cdot$ distributes over $+$ on both the left and right in the sense that $x(y+z) = xy+xz$ and $(x+y)z = xz+yz$,
- 0 is an *annihilator* for $\cdot$ in the sense that $0\cdot x = x\cdot 0 = 0$ for all x.

A semiring is *idempotent* if $x + x = x$ for all x. We often abbreviate $x\cdot y$ to xy, and avoid parentheses by taking $\cdot$ to be higher precedence than $+$.

Collecting these axioms, we define an *idempotent semiring* to be any structure $(S,\ +,\ \cdot,\ 0,\ 1)$ satisfying the following identities for all $x, y, z \in S$:

$$\begin{array}{ll}
x + (y + z) = (x + y) + z & \qquad x + y = y + x \\
x + 0 = x & \qquad x + x = x \\
x(yz) = (xy)z & \qquad 1x = x1 = x \\
x(y + z) = xy + xz & \qquad (x + y)z = xz + yz \\
0x = x0 = 0. &
\end{array}$$

A *ring* is a semiring in which the additive monoid forms a group; that is, in which additive inverses exist. We cannot have additive inverses in an idempotent semiring unless the semiring is trivial, since $0 = -x + x = -x + x + x = 0 + x = x$.

Order

Recall that a *partial order* is a binary relation on a set that is

- reflexive: for all x, $x \leq x$,
- antisymmetric: for all x, y, if $x \leq y$ and $y \leq x$, then $x = y$, and
- transitive: for all x, y, z, if $x \leq y$ and $y \leq z$, then $x \leq z$.

Any idempotent semiring has a naturally-defined partial order $\leq$ associated with it:

$$x \leq y \overset{\text{def}}{\Leftrightarrow} x + y = y. \tag{6}$$

The order relation $\leq$ is so central to the theory that one might take it as primitive, but we will consider it an abbreviation for the equation on the right-hand side of (6).

In the language-theoretic and relational models of the last lecture, $\leq$ is set inclusion $\subseteq$. But beware: in the $(\min, +)$ algebra, $\leq$ is the *reverse* of the natural order on $\mathbb{R}$ extended to $\mathbb{R}_+ \cup \{\infty\}$. By (6), $x \leq y$ iff $\min x, y = y$, but this occurs iff x is greater than or equal to y in the natural order on $\mathbb{R}$. (Note that we are carefully avoiding the use of the notation $\leq$ for the natural order on $\mathbb{R}$!)

That $\leq$ is a partial order follow easily from the definition (6) and the axioms of idempotent semirings. Reflexivity is just the idempotence axiom $x + x = x$. Antisymmetry follows from commutativity of $+$: if $x \leq y$ and $y \leq x$, then $y = x + y = y + x = x$. Finally, for transitivity, if $x \leq y$ and $y \leq z$, then

$$\begin{aligned} x + z &= x + (y + z) && \text{since } y + z = z \\ &= (x + y) + z && \text{associativity of } + \\ &= y + z && \text{since } x + y = y \\ &= z && \text{again since } y + z = z, \end{aligned}$$

therefore $x \leq z$.

In any idempotent semiring, the operators $+$ and $\cdot$ are *monotone* with respect to $\leq$ in the sense that for all $x, y, z \in S$,

$$\begin{aligned} x \leq y &\Rightarrow x + z \leq y + z \\ x \leq y &\Rightarrow z + x \leq z + y \\ x \leq y &\Rightarrow xz \leq yz \\ x \leq y &\Rightarrow zx \leq zy. \end{aligned}$$

(Monotonicity will also hold for the * operator of Kleene algebra, that is, $x \leq y \Rightarrow x^* \leq y^*$, when we get around to discussing it.) These properties follow easily from the axioms.

In any idempotent semiring, the operator $+$ gives the *least upper bound* or *supremum* of any pair of elements with respect to the natural order $\leq$, and 0 is the least element of the semiring with respect to $\leq$. To say that $x + y$ is the *least upper bound* of x and y says that

- $x + y$ is an upper bound: $x \leq x + y$ and $y \leq x + y$;
- $x + y$ is the *least* upper bound: if z is any other upper bound, that is, if $x \leq z$ and $y \leq z$, then $x + y \leq z$.

The proof that $x + y$ is an upper bound uses associativity and idempotence: $x + (x + y) = (x + x) + y = x + y$, and similarly for $y \leq x + y$. The proof that it is the least upper bound uses only associativity: if $x \leq z$ and $y \leq z$, then $(x + y) + z = x + (y + z) = x + z = z$.

One observation that is not difficult to check is that the $n \times n$ matrices over a semiring again form a semiring under the natural definitions of the matrix operations. Moreover, if the underlying semiring is idempotent, then so is the matrix semiring.

The * Operator

Now we turn to the * operator. This is the most interesting part of Kleene algebra, because it captures the notion of *iteration*. Because of this, it may seem that * is inherently infinitary. Indeed, there are several infinitary axiomatizations that we will consider. However, it is possible to derive most of the interesting parts of the theory in a purely finitary way.

The * operator is a unary operator written in postfix. Intuitively, x^* represents zero or more iterations of x. In relational models, this is reflexive transitive closure; in language models, the Kleene asterate.

There are several different competing axiomatizations of *, and in part our study will be to understand the relationships among them. For now, we shall pick a particular one as our official definition for the purposes of this course. Thus we define a *Kleene algebra* to be a structure $(K, +, \cdot, ^*, 0, 1)$ that is an idempotent semiring under $+, \cdot, 0, 1$ satisfying the axioms (7)–(10) below for *. We assign precedence $^* > \cdot > +$ to the operators to avoid unnecessary parentheses.

The axioms for * consist of two equations and two equational implications or *Horn formulas*. (Note that up to now, the axioms have been purely equational.) The two equational axioms for * are

$$1 + xx^* \le x^* \tag{7}$$
$$1 + x^*x \le x^* \tag{8}$$

and the two equational implications are

$$b + ax \le x \Rightarrow a^*b \le x \tag{9}$$
$$b + xa \le x \Rightarrow ba^* \le x. \tag{10}$$

Of course, these are all considered to be implicitly universally quantified, so that (9) and (10) are assumed to hold for all a, b, and x in any Kleene algebra.

The significance of (7)–(10) concerns the solution of linear inequalities. As we shall see, much of the theory of Kleene algebra is concerned with the solution of finite systems of linear inequalities. For example, a finite automaton is essentially such a system. Axioms (7) and (9) provide for the existence of a unique least solution to a certain single linear inequality in a single variable, namely

$$b + aX \le X, \tag{11}$$

where X is a variable ranging over elements of the Kleene algebra. Axioms (7) and (9) together essentially say that a^*b is a solution to (11), and moreover, it is the unique least solution among all solutions in the Kleene algebra. First, (7) says that a^*b is a solution, since by monotonicity of multiplication and distributivity,

$$\begin{aligned} 1 + aa^* \le a^* &\Rightarrow (1 + aa^*)b \le a^*b \\ &\Rightarrow b + a(a^*b) \le a^*b\ ; \end{aligned}$$

and (9) says exactly that a^*b is less than or equal to any other solution, therefore it is the unique least solution. Dually, the axioms (8) and (10) say that ba^* is the unique least solution to $b + Xa \le X$.

Let us illustrate the use of (9) and (10) to show that (7) and (8) can be strengthened to equalities. We show this for (7); the result for (8) is symmetric. We already have that $1 + xx^* \le x^*$, so by antisymmetry, it suffices to show the reverse inequality; equivalently,

$$x^*1 \le 1 + xx^*.$$

This is the right-hand side of (9) with 1 substituted for b, x substituted for a, and $1 + xx^*$ substituted for x, so it suffices to show that the left-hand side of (9) holds under the same substitution, or

$$1 + x(1 + xx^*) \le 1 + xx^*.$$

But this is immediate from (7) and monotonicity.

Recall from the last lecture that in relational models, R^* was defined to be the reflexive transitive closure of the relation R. To be reflexive means that $\mathsf{id} \subseteq R^*$, where id is the identity relation; to be transitive means that $R^* \,;\, R^* \subseteq R^*$; and to contain R means that $R \subseteq R^*$. Abstractly, these properties are expressed by the inequalities

$$1 \le x^* \tag{12}$$
$$x^*x^* \le x^* \tag{13}$$
$$x \le x^*, \tag{14}$$

respectively. Equivalently,

$$1 + x^*x^* + x \le x^*. \tag{15}$$

We might interpret this inequality as saying that x^* is a reflexive and transitive element dominating x. It does not, however, say that it is the reflexive transitive closure of x; for that we need the equational implication

$$1 + yy + x \le y \Rightarrow x^* \le y, \tag{16}$$

which says that x^* is the *least* reflexive and transitive element dominating x.

Now in the presence of the other axioms, (15) is equivalent to (7) (and, by symmetry, to (8) as well). To prove that (7) implies (15), it suffices to show that (7) implies (12)–(14). The inequality (12) is immediate from (7). Also, multiplying (12) on the left by x, by monotonicity we have $x \le xx^*$; then (14) is immediate from this and (7). The last inequality (13) requires either (9) or (10).

Conversely, to show that (15) implies (7), assume (15). Then $1 \le x^*$ from (12). Also, by (13), (14), and monotonicity we have $xx^* \le x^*x^* \le x^*$. Since $1 + xx^*$ is the least upper bound of 1 and xx^*, we have (7).

Each of (9) and (10) alone implies (16). The converse does not hold: later, we will construct a "left-handed" Kleene algebra that is not "right-handed" (one satisfying (9) but not (10)). To show that (9) implies (16), suppose that (9) holds for all a, b, and x, and assume the left-hand side of (16). To show the right-hand side of (16) holds, by (9) it suffices to show that $1 + xy \le y$. But this follows easily from the left-hand side of (16): we have $x \le y$, and by monotonicity, $1 + xy \le 1 + yy \le y$.

In the presence of the other axioms, the implications (9) and (10) are equivalent to

$$ax \le x \Rightarrow a^*x \le x \tag{17}$$
$$xa \le x \Rightarrow xa^* \le x, \tag{18}$$

respectively. These alternative forms are quite useful in some contexts.

Finally, as promised, we show that * is monotone. Suppose $x \le y$. We wish to show that $x^* \le y^*$. By (9), it suffices to show that $1 + xy^* \le y^*$. But since $x \le y$, by monotonicity and (7) we have $1 + xy^* \le 1 + yy^* \le y^*$.

Next time we will look at some alternative axiomatizations of *.

References

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

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

[3] 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.

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

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

[6] Adam Barth and Dexter Kozen. Equational verification of cache blocking in LU decomposition using Kleene algebra with tests. Technical Report TR2002-1865, Computer Science Department, Cornell University, June 2002.

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

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

[9] Ernie Cohen. Hypotheses in Kleene algebra. Technical Report TM-ARH-023814, Bellcore, 1993. `http://citeseer.nj.nec.com/1688.html`.

[10] Ernie Cohen. Lazy caching in Kleene algebra, 1994. `http://citeseer.nj.nec.com/22581.html`.

[11] Ernie Cohen. Using Kleene algebra to reason about concurrency control. Technical report, Telcordia, Morristown, N.J., 1994.

[12] John Horton Conway. *Regular Algebra and Finite Machines.* Chapman and Hall, London, 1971. Dover edition, 2012.

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

[14] Nate Foster, Dexter Kozen, Mae Milano, Alexandra Silva, and Laure Thompson. A coalgebraic decision procedure for NetKAT. In *Proc. 42nd ACM SIGPLAN-SIGACT Symp. Principles of Programming Languages (POPL'15)*, pages 343–355, Mumbai, India, January 2015. ACM.

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

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

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

[18] 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.

[19] 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.

[20] Dexter Kozen. On Kleene algebras and closed semirings. In Rovan, editor, *Proc. Math. Found. Comput. Sci. (MFCS'90)*, volume 452 of *Lecture Notes in Computer Science*, pages 26–47, Banska-Bystrica, Slovakia, 1990. Springer-Verlag.

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

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

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

[24] Dexter Kozen. Kleene algebras with tests and the static analysis of programs. Technical Report TR2003-1915, Computer Science Department, Cornell University, November 2003.

[25] 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 (CL'00)*, volume 1861 of *Lecture Notes in Artificial Intelligence*, pages 568–582, London, July 2000. Springer-Verlag.

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

[27] 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.

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

[29] Nikolay Mateev, Vijay Menon, and Keshav Pingali. Fractal symbolic analysis. In *Proc. 15th Int. Conf. on Supercomputing*, pages 38–49. ACM, ACM Press, 2001.

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

[31] B. Möller. Calculating with pointer structures. In R. Bird and L. Meertens, editors, *Algorithmic Languages and Calculi. Proc. IFIP TC2/WG2.1 Working Conference*, pages 24–48. Chapman and Hall, February 1997.

[32] B. Möller. Safer ways to pointer manipulation. Technical Report 2000-4, Institut für Informatik, Universität Augsburg, 2000.

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

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

[35] 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.

[36] 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.

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

[38] 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.

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

[40] A. Tarski. On the calculus of relations. *J. Symbolic Logic*, 6(3):73–89, 1941.