CS611 Lecture 17 **Kleene Algebra** 6 October 2003

Scribe: Art Munson Lecturer: Dexter Kozen

1 Kleene Algebra (KA)

Kleene Algebra is the algebra of regular expressions. Since regular expressions define finite state automata (FSA), it gives us an algebra over FSA's. It has applications in automata and regular expressions, relational algebra, and the design / analysis of algorithms (particularly shortest paths, connectivity, and computational geometry).

1.1 Axioms

Kleene algebra is an *idempotent semiring* under $+, \cdot, 0, 1$.[1] This structure has the following properties:

$(p+q)+r = p+(q+r)$	$(pq)r = p(qr)$	($+, \cdot$ are associative)
$p+q = q+p$		($+$ is commutative)
$p1 = 1p = p$	$p0 = 0p = 0$	(1,0 are idempotent under $\cdot$)
$p+0 = p$		(0 is idempotent under $+$)
$p(q+r) = pq+pr$	$(p+q)r = pr+qr$	($\cdot$ is distributive)

$$\begin{aligned} p^*q &= \text{ least } x \text{ such that } q + px \le x \\ qp^* &= \text{ least } x \text{ such that } q + xp \le x \end{aligned}$$

In the algebra define $x \le y \overset{\text{def}}{\leftrightarrow} x + y = y$.

This is a universal Horn axiomatization.

1.2 Standard Interpretation of Kleene Algebra

The standard interpretation of Kleene algebra uses regular sets over Σ, a set of symbols, to define a set of strings. Let A, B be sets. The interpretation is as follows (notation in Kleene algebra on the left):

$$\begin{array}{lcll} A+B & = & A \cup B & \\ AB & = & \{xy \mid x \in A, y \in B\} & \text{(is this the cartesian product of } A, B\text{?)} \\ A^* & = & \bigcup_{n \ge 0} A^n = A^0 \cup A^1 \cup A^2 \cup \ldots & \\ 1 & = & \{\varepsilon\} & \\ 0 & = & \emptyset & \end{array}$$

The expression $p \in \Sigma$ is interpreted as $\{p\}$.

[1] "A *semiring* is an algebra $(A, \cdot, +, 0, 1)$ of a set A, where 0 and 1 are constants, $(A, \cdot, 1)$ is a monoid, $(A, +, 0)$ is a commutative monoid, $\cdot$ distributes over $+$ from the left and right, and 0 is both a left and right annihilator ($0a = a0 = 0$). Often $a \cdot b$ is written simply as ab, and the semiring $(A, \cdot, +, 0, 1)$ as simply A.

The relation $\le$ on a semiring A is defined as $a \le b$ if and only if there exists some $c \in A$ such that $a + c = b$, and is a quasiordering. If $+$ is idempotent over A (that is, $a + a = a$ for all $a \in A$), then $\le$ is a partial ordering.

Addition and (left and right) multiplication are monotonic operators with respect to $\le$, with 0 as the minimal element."
(*source:* http://planetmath.org/encyclopedia/semiring.html).

1.3 Interpretation for Binary Relations

Alternatively, Kleene algebra can be interpreted over binary relations. Let R, S be binary relations on a set X:

$$\begin{array}{lcl} R+S & = & R \cup S \\ RS & = & R \circ S = \{(u,v) \mid \exists w.(u,w) \in R, (w,v) \in S\} \\ R^* & = & \text{reflexive transitive closure of } R = \bigcup_{n \geq 0} R^n = R \cup R^1 \cup R^2 \cup \ldots \\ 1 & = & \text{identity relation} = \{(u,u) \mid u \in X\} \\ 0 & = & \emptyset \end{array}$$

This interpretation has some properties useful for KA proofs, particularly sliding and denesting:

$$\begin{array}{ll} 1 + pp^* = 1 + p^*p = p^* & \\ p^*p^* = p^* * = p^* & \\ (pq)^*p = p(qp)^* & \text{sliding} \\ (p^*q)^*p^* = (p+q)^* & \text{denesting} \\ px = xq \rightarrow p^*x = xq^* & \text{bisimulation, where } x \text{ is an arbitrary element} \\ qp = 0 \rightarrow (p+q)^* = p^*q^* & \text{loop distribution} \end{array}$$

1.4 Completeness and Complexity of Kleene Algebra

The following have been shown about Kleene algebra:

- Deductively complete for the equational theory of regular sets of strings and relational models.
- Complexity: equational theory is PSPACE-complete.
- Hoare theory (Horn theory with premises $p = 0$) is PSPACE-complete.
- Horn theory is Π^1_1-complete for star-continuous and relational models.

1.5 Matrices over a Kleene Algebra

We can use matrices of regular expressions to represent state transitions in FSA. Once we define the matrix operations, then Kleene algebra can be applied to the state transitions. The matrix operations are defined as:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} + \begin{pmatrix} e & f \\ g & h \end{pmatrix} \stackrel{\text{def}}{=} \begin{pmatrix} a+e & b+f \\ c+g & d+h \end{pmatrix}$$

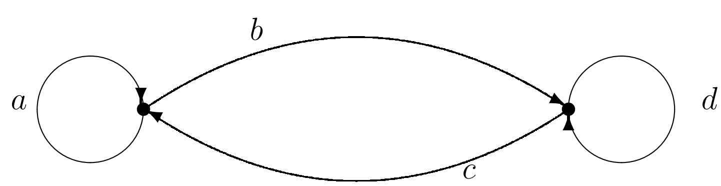
$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} e & f \\ g & h \end{pmatrix} \stackrel{\text{def}}{=} \begin{pmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{pmatrix}$$

$$0 \stackrel{\text{def}}{=} \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \qquad 1 \stackrel{\text{def}}{=} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^* \stackrel{\text{def}}{=} \begin{pmatrix} (a+bd^*c)^* & (a+bd^*c)^*bd^* \\ (d+ca^*b)^*ca^* & (d+ca^*b)^* \end{pmatrix}$$

For larger matrices (*e.g.* 3-by-3), the operations can be defined inductively by considering each a, b, c, d as block matrices.

Where did the last equation come from? Well, it was derived with the below picture in mind:



Besides representing finite automata, matrices over a Kleene algebra (KA) let us do construction of regular expressions, solve linear equations over a KA, and lead to connectivity and shortest path algorithms.

2 Kleene Algebra with Tests (KAT)

Kleene algebra with tests adds a boolean algebra $(B, +, \cdot, -, 0, 1)$ to a KA to get $(K, B, +, \cdot, *, -, 0, 1)$, where $B \subseteq K$.[2] The convention in this section will be that $p, q, r \in K$ and $a, b, c \in B$.

Adding tests lets us apply KAT to language-theoretic models, where K = regular sets of guarded strings over Σ, B and B = a finite free boolean algebra. Other possibilities are relational models and matrices over a KAT.

2.1 Completeness and Complexity of KAT

The following have been shown about KAT:

- Deductively complete for the equational theory of regular sets of guarded strings and relational models.
- Complexity: equational theory is PSPACE-complete.
- Hoare theory (Horn theory with premises $p = 0$) is PSPACE-complete.
- Full Horn theory is still Π^1_1-complete.

2.2 Modeling Programs

Here is an example of how to model program constructs to get a flavor of how KAT can model programs:

$$\begin{aligned} &\textbf{if}\, b\, \textbf{then}\, p\, \textbf{else}\, q &\equiv\quad& bp + \overline{b}q \\ &\textbf{while}\, b\, \textbf{do}\, p &\equiv\quad& (bp)^*\overline{b} \end{aligned}$$

Because KAT subsumes propositional Hoare logic (PHL), we can also encode partial correctness assertions (PCA's) using KAT. The translation of Hoare logic rules to KAT is straight-forward, and can be found in Dexter's slide deck link referenced in Section 3.

KAT includes a completeness theorem: all relationally valid rules of the form

$$\frac{\{b_1\}p_1\{c_1\}, \ldots, \{b_n\}p_n\{c_n\}}{\{b\}p\{c\}}$$

are derivable in KAT. This is not the case for PHL; consider the following valid rule that is unprovable in Hoare logic:

$$\frac{\{c\}\textbf{if}\, b\, \textbf{then}\, p\, \textbf{else}\, p\{c\}}{\{c\}p\{c\}}$$

In contrast, the corresponding KAT expression $c(bp + \overline{b}p)\overline{c} = 0$ is easily provable in KAT.

2.3 Applications of KAT

KAT can be used in compiler optimizations, static analysis and verification, complexity of substructural logics, and unique ordered binary decision diagram (BDD).

3 More Information

The slides for the lecture can be found at:

http://www.cs.cornell.edu/kozen/611-10-6-03.ppt

Of course, you can always talk to Dexter!

[2]The operator $-$ is for complement, not minus. In use it is placed above a boolean expression, *e.g.* $\overline{a}$.