Introduction to Kleene Algebra	Lecture 17
CS786 Spring 2004	April 12, 2004

Complexity of Kleene Algebra with Tests

In this lecture we show that the equational theory of KAT is *PSPACE*-complete. Thus KAT, while considerably more expressive than KA without tests, is no more difficult to decide. The results of this lecture are from [7, 2].

We have shown (Lecture **??**) that the Hoare theory of KAT (Horn formulas with premises of the form $r = 0$) reduces efficiently to the equational theory. We have also argued (Lecture **??**) that the equational theories of KAT and KAT* (star-continuous KAT) coincide, and that these theories are complete over certain language-theoretic and relational models.

Our *PSPACE* algorithm makes use of $\mathsf{Reg}_{\mathsf{P,B}}$, the free language-theoretic model involving sets of guarded strings introduced in Lecture **??**, and matrices over Kleene algebras with tests.

In contrast, propositional dynamic logic (PDL) is *EXPTIME*-complete [3], which indicates that some savings can be achieved by using KAT in applications where PDL would previously have been used.

We will show later that star-continuous KA in the presence of extra commutativity conditions of the form $pq = qp$, even for primitive p and q, is undecidable. This was observed by Cohen [1]. In fact, the universal Horn theory of KAT* is Π^1_1-complete [6].

Matrix Algebras

Let K be a Kleene algebra with tests B. As argued in Lecture **??**, the structure

$$(\mathrm{Mat}(n,K),\ \Delta(n,B),\ +,\ \cdot,\ ^*,\ ^-,\ 0_n,\ I_n)$$

again forms a Kleene algebra with tests, where $\mathrm{Mat}(n,K)$ denotes the family of $n \times n$ matrices over K, the operations $+$ and $\cdot$ are the usual operations of matrix addition and multiplication, respectively, 0_n is the $n \times n$ zero matrix, and I_n the $n \times n$ identity matrix. The operation * on matrices is defined inductively:

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]^* = \left[\begin{array}{c|c} (A+BD^*C)^* & (A+BD^*C)^*BD^* \\ \hline D^*C(A+BD^*C)^* & D^*+D^*C(A+BD^*C)^*BD^* \end{array}\right] \tag{17.1}$$

The distinguished Boolean subalgebra is the set $\Delta(n,B)$ of $n \times n$ diagonal matrices with entries from the distinguished Boolean algebra B. The operation $^-$ on $\Delta(n,B)$ just complements the diagonal elements, leaving the off-diagonal elements 0.

KAT Homomorphisms and Finitary Algebras

Definition 17.1 *Let K, K' be Kleene algebras with tests B, B', respectively. A* KAT-homomorphism *is a Kleene algebra homomorphism $h : K \to K'$ whose restriction to B is a Boolean algebra homomorphism $h : B \to B'$.*

Lemma 17.2 *Let $h : K \to K'$ be a KAT-homomorphism, and let $H : \mathrm{Mat}(n, K) \to \mathrm{Mat}(n, K')$ be its componentwise extension to matrices. Then H is a KAT-homomorphism.*

Proof. By definition, $H(E)_{ij} = h(E_{ij})$. It is immediate that $H(E + F) = H(E) + H(F)$, $H(EF) = H(E)H(F)$, $H(0) = 0$, and $H(I) = I$. For *, we can use the inductive definition (17.1) to give a straightforward inductive proof that $H(E^*) = H(E)^*$. Finally, for $E \in \Delta(n, B)$,

$$\begin{array}{rcl}
H(\overline{E})_{ij} & = & h(\overline{E}_{ij}) \\
& = & \left\{ \begin{array}{ll} h(\overline{E_{ij}}), & \text{if } i = j, \\ h(0), & \text{if } i \neq j \end{array} \right. \\
& = & \left\{ \begin{array}{ll} \overline{h(E_{ij})}, & \text{if } i = j, \\ 0, & \text{if } i \neq j \end{array} \right. \\
& = & \left\{ \begin{array}{ll} \overline{H(E)_{ij}}, & \text{if } i = j, \\ 0, & \text{if } i \neq j \end{array} \right. \\
& = & \overline{H(E)}_{ij} \ ,
\end{array}$$

so $H(\overline{E}) = \overline{H(E)}$. □

A Kleene algebra or Kleene algebra with tests is called *finitary* if for all $a \in K$ there exists an $m \geq 0$ such that $a^* = (1 + a)^m$. Any finite algebra is finitary, and any finitary algebra is star-continuous.

Lemma 17.3 *If K is finitary, then so is* $\mathrm{Mat}(n, K)$.

Proof. We proceed by induction on n. For the basis, the algebras K and $\mathrm{Mat}(1, K)$ are isomorphic, so there is nothing to prove. Now suppose $n \geq 2$. Break up $E \in \mathrm{Mat}(n, K)$ arbitrarily into submatrices

$$E = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right]$$

where A and D are square. Using the denesting rule,

$$
\begin{aligned}
\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]^* &= \left(\left[\begin{array}{c|c} A & 0 \\ \hline 0 & D \end{array}\right] + \left[\begin{array}{c|c} 0 & B \\ \hline C & 0 \end{array}\right]\right)^* \\
&= \left(\left[\begin{array}{c|c} A & 0 \\ \hline 0 & D \end{array}\right]^* \left[\begin{array}{c|c} 0 & B \\ \hline C & 0 \end{array}\right]\right)^* \left[\begin{array}{c|c} A & 0 \\ \hline 0 & D \end{array}\right]^* \\
&= \left(\left[\begin{array}{c|c} A^* & 0 \\ \hline 0 & D^* \end{array}\right] \left[\begin{array}{c|c} 0 & B \\ \hline C & 0 \end{array}\right]\right)^* \left[\begin{array}{c|c} A^* & 0 \\ \hline 0 & D^* \end{array}\right] \\
&= \left[\begin{array}{c|c} 0 & A^*B \\ \hline D^*C & 0 \end{array}\right]^* \left[\begin{array}{c|c} A^* & 0 \\ \hline 0 & D^* \end{array}\right]
\end{aligned}
$$

By the induction hypothesis, there exists an $m \geq 0$ such that

$$
\begin{aligned}
\left[\begin{array}{c|c} A^* & 0 \\ \hline 0 & D^* \end{array}\right] &= \left[\begin{array}{c|c} (I+A)^m & 0 \\ \hline 0 & (I+D)^m \end{array}\right] \\
&= \left[\begin{array}{c|c} I+A & 0 \\ \hline 0 & I+D \end{array}\right]^m \\
&\leq (I+E)^m
\end{aligned}
$$

Also, using the KA theorem $x^* = (xx)^*(1+x)$,

$$
\begin{aligned}
\left[\begin{array}{c|c} 0 & A^*B \\ \hline D^*C & 0 \end{array}\right]^* &= \left[\begin{array}{c|c} A^*BD^*C & 0 \\ \hline 0 & D^*CA^*B \end{array}\right]^* \left[\begin{array}{c|c} I & A^*B \\ \hline D^*C & I \end{array}\right] \\
&= \left[\begin{array}{c|c} (A^*BD^*C)^* & 0 \\ \hline 0 & (D^*CA^*B)^* \end{array}\right] \left[\begin{array}{c|c} I & A^*B \\ \hline D^*C & I \end{array}\right]
\end{aligned}
$$

By the induction hypothesis, there exists a $k \geq 0$ such that

$$
\begin{aligned}
&\left[\begin{array}{c|c} (A^*BD^*C)^* & 0 \\ \hline 0 & (D^*CA^*B)^* \end{array}\right] \\
&= \left[\begin{array}{c|c} (I+A^*BD^*C)^k & 0 \\ \hline 0 & (I+D^*CA^*B)^k \end{array}\right] \\
&= \left[\begin{array}{c|c} I+A^*BD^*C & 0 \\ \hline 0 & I+D^*CA^*B \end{array}\right]^k \\
&= \left(I + \left[\begin{array}{c|c} A^* & 0 \\ \hline 0 & D^* \end{array}\right] \left[\begin{array}{c|c} 0 & B \\ \hline C & 0 \end{array}\right] \left[\begin{array}{c|c} A^* & 0 \\ \hline 0 & D^* \end{array}\right] \left[\begin{array}{c|c} 0 & B \\ \hline C & 0 \end{array}\right]\right)^k \\
&\leq (I+(I+E)^m E (I+E)^m E)^k \\
&\leq (I+E)^{2k(m+1)}
\end{aligned}
$$

Similarly,

$$
\begin{array}{rcl}
\left[\begin{array}{c|c} I & A^*B \\ \hline D^*C & I \end{array}\right] & = & I + \left[\begin{array}{c|c} A^* & 0 \\ \hline 0 & D^* \end{array}\right] \left[\begin{array}{c|c} 0 & B \\ \hline C & 0 \end{array}\right] \\
& \leq & I + (I+E)^m E \\
& \leq & (I+E)^{m+1}
\end{array}
$$

Putting these all together, we have

$$E^* \quad \leq \quad (I+E)^{2km+2k+2m+1}$$

□

Matrices over a Boolean Algebra

In this section we establish some special properties of matrices over a Boolean algebra that will prove useful in the subsequent development.

If B is the distinguished Boolean algebra of a Kleene algebra with tests K, then the algebra $\mathrm{Mat}(n, B)$ is a subalgebra of $\mathrm{Mat}(n, K)$. (Note that it is not the distinguished Boolean algebra of $\mathrm{Mat}(n, K)$; in fact, it is not even a Boolean algebra in general). The algebra $\Delta(n, B)$ of diagonal matrices over B is the distinguished Boolean algebra of both $\mathrm{Mat}(n, K)$ and $\mathrm{Mat}(n, B)$.

Since $b^* = 1$ for any $b \in B$, it follows immediately from Lemma 17.3 that $\mathrm{Mat}(n, B)$ is finitary. In fact, it can be established by combinatorial means that if $A \in \mathrm{Mat}(n, B)$, then $A^* = (I + A)^{n-1}$, but we will not need this tighter bound.

Let B denote the free Boolean algebra on generators B. Given a matrix $J \in \mathrm{Mat}(n, B)$ and an atom α, let J_α be the 0-1 matrix

$$(J_\alpha)_{ij} \quad = \quad \begin{cases} 1, & \text{if } \alpha \leq J_{ij} \\ 0, & \text{otherwise.} \end{cases}$$

Lemma 17.4

$$\alpha \leq (J^*)_{ij} \quad \Leftrightarrow \quad (J^*_\alpha)_{ij} = 1$$

In particular, one can determine whether $\alpha \leq (J^*)_{ij}$ in linear time.

Proof. The first statement is a direct application of Lemma 17.2, using the Boolean homomorphism $h_\alpha : B \to \{0, 1\}$ defined by

$$h_\alpha(b) \quad = \quad \begin{cases} 1, & \text{if } \alpha \leq b \\ 0, & \text{otherwise.} \end{cases}$$

Then $J_\alpha = H_\alpha(J)$, where H_α is the componentwise extension of h_α to matrices. The condition to be proved is equivalent to the statement $H_\alpha(J^*) = H_\alpha(J)^*$.

The entries of J_α can be determined by testing whether $\alpha \le b$, which essentially amounts to evaluating a Boolean expression on a given truth assignment. The matrix J_α is a 0-1 matrix, so $(J_\alpha^*)_{ij}$ can be determined in linear time by depth first search on the corresponding directed graph. The entire matrix J_α^* can be computed efficiently using any standard transitive closure algorithm. $\square$

Matrix Representation of Terms

We eventually want to give an algorithm for deciding whether $\mathsf{KAT} \models p = q$. By Theorem **??** of Lecture **??**, it suffices to decide whether $G(p) = G(q)$, where G is the canonical interpretation $G : \mathsf{RExp}_{\mathsf{P,B}} \to \mathsf{Reg}_{\mathsf{P,B}}$, as defined in Lecture **??**.

One possible approach, exploited in Lecture **??**, is to construct from $p \in \mathsf{RExp}_{\mathsf{P,B}}$ a regular expression $\widehat{p} \in \mathsf{RExp}_{\mathsf{P,B}}$ such that

$$G(p) \quad = \quad R(\widehat{p}). \tag{17.2}$$

Then deciding whether $G(p) = G(q)$ reduces to deciding whether $R(\widehat{p}) = R(\widehat{q})$, which we know how to do in *PSPACE*.

Unfortunately, the construction of $\widehat{p}$ from p as given in Lecture **??** involves an exponential blowup, which the following example shows to be unavoidable. Suppose $k = 2m$. Consider the expression

$$p \quad = \quad (b_1 b_{m+1} + \overline{b}_1 \overline{b}_{m+1})(b_2 b_{m+2} + \overline{b}_2 \overline{b}_{m+2}) \cdots (b_m b_k + \overline{b}_m \overline{b}_k)$$

This expression represents the set of atoms in which the i^{th} and $m + i^{\text{th}}$ literal have the same parity. Any nondeterministic finite automaton accepting $G(p)$ must store in its state the first half of the string so that it can verify that the second half is correct. Therefore the automaton must have at least 2^m states. Since the translation between regular expressions and nondeterministic automata is linear, any regular expression $\widehat{p}$ such that $R(\widehat{p}) = G(p)$ must be exponentially longer than p.

To circumvent this exponential blowup, we work with a matrix representation of expressions. The construction of Kleene's theorem as given in Lecture **??** produces a matrix $P \in \mathrm{Mat}(n, \mathcal{F})$ with small entries and 0-1 vectors u, v of length n such that

$$R(p) \quad = \quad R(u^T P^* v)\ , \tag{17.3}$$

where n is approximately the size of p and $\mathcal{F}$ is the free Kleene algebra with tests on generators P and B. The construction of P is by induction on the structure of p, and corresponds to the combinatorial construction of an automaton from a regular expression as found for example in [4, 5]. The matrix P is the transition matrix of the automaton equivalent to the regular expression p over the input alphabet $\mathsf{P} \cup \mathsf{B} \cup \overline{\mathsf{B}}$. The vectors u and v specify the start and final states of the automaton, respectively. The elements of P are 0, 1, and sums of primitive symbols. This construction is given in its entirety in Lecture **??**, so we do not repeat it here.

Since the entries of P are sums of primitive symbols, we can write $P = J + A$, where the entries of J are sums of elements of $\mathsf{B} \cup \overline{\mathsf{B}}$ and the entries of A are sums of elements of P. Using the denesting rule of KA, we can then write

$$P^* \quad = \quad (J^* A)^* J^*$$

This form is particularly well suited to the treatment of guarded strings $\alpha_0 p_1 \alpha_1 \cdots \alpha_{m-1} p_m \alpha_m$, the guards α_i being handled by J^* and the symbols p_i by A.

We extend the definition of J_α above to general matrices. For $p \in \mathsf{P}$, define the 0-1 matrix

$$(A_p)_{ij} \quad = \quad \begin{cases} 1, & \text{if } p \leq A_{ij} \\ 0, & \text{otherwise.} \end{cases}$$

Lemma 17.5 *$\alpha_0 p_1 \alpha_1 \cdots \alpha_{m-1} p_m \alpha_m \in G(u^T (J^* A)^* J^* v)$ if and only if*

$$u^T J^*_{\alpha_0} A_{p_1} J^*_{\alpha_1} \cdots J^*_{\alpha_{m-1}} A_{p_m} J^*_{\alpha_m} v \quad = \quad 1$$

Proof. Because of the restricted form of the entries in J^* and A, all guarded strings in $G(u^T (J^* A)^k J^* v)$ are of the form $\alpha_0 p_1 \alpha_1 \cdots \alpha_{k-1} p_k \alpha_k$. Since

$$G(u^T (J^* A)^* J^* v) \quad = \quad \bigcup_{k \geq 0} G(u^T (J^* A)^k J^* v) \ ,$$

we have that

$$\begin{gathered} \alpha_0 p_1 \alpha_1 \cdots \alpha_{m-1} p_m \alpha_m \in G(u^T (J^* A)^* J^* v) \\ \Leftrightarrow \\ \alpha_0 p_1 \alpha_1 \cdots \alpha_{m-1} p_m \alpha_m \in G(u^T (J^* A)^m J^* v) \end{gathered}$$

Furthermore, by the definition of matrix multiplication, this occurs iff there exist s_0, t_0, s_1, t_1, ..., s_m, t_m such that

- $u_{s_0} = 1$
- $\alpha_i \leq (J^*)_{s_i t_i}$, $0 \leq i \leq m$
- $p_i \leq A_{t_{i-1} s_i}$, $1 \leq i \leq m$

- $v_{t_m} = 1$.

By Lemma 17.4 and the definition of A_{p_i}, this occurs iff there exist s_0, t_0, s_1, t_1, $\ldots$, s_m, t_m such that

- $u_{s_0} = 1$
- $(J^*_{\alpha_i})_{s_i t_i} = 1$, $0 \leq i \leq m$
- $(A_{p_i})_{t_{i-1} s_i} = 1$, $1 \leq i \leq m$
- $v_{t_m} = 1$.

By the definition of Boolean matrix multiplication, this occurs iff

$$u^T J^*_{\alpha_0} A_{p_1} J^*_{\alpha_1} \cdots J^*_{\alpha_{m-1}} A_{p_m} J^*_{\alpha_m} v \quad = \quad 1$$

□

A *PSPACE* Algorithm

Now we give a *PSPACE* algorithm for deciding whether $\mathsf{KAT} \models p \leq q$, or equivalently by Theorem **??** of Lecture **??**, whether $G(p) \subseteq G(q)$. The algorithm will nondeterministically guess a guarded string

$$\alpha_0 p_1 \alpha_1 \cdots \alpha_{m-1} p_m \alpha_m \quad \in \quad G(p) - G(q).$$

We first produce the matrices u, P, v and y, Q, z such that

$$\begin{aligned} R(p) &= R(u^T P^* v) \\ R(q) &= R(y^T Q^* z). \end{aligned}$$

By the fact that

$$G(p) \quad = \quad \bigcup_{x \in R(p)} G(x)$$

proved in Lecture **??**, we also have

$$\begin{aligned} G(p) &= G(u^T P^* v) \\ G(q) &= G(y^T Q^* z) \end{aligned}$$

Writing $P = J + A$ and $Q = K + B$ where the entries of J and K are sums of elements of $\mathsf{B} \cup \overline{\mathsf{B}}$ and the entries of A and B are sums of elements of P, we have

$$\begin{aligned} G(p) &= G(u^T (J^* A) J^* v) \\ G(q) &= G(y^T (K^* B) K^* z) \end{aligned}$$

By Lemma 17.5, it suffices to guess $\alpha_0 p_1 \alpha_1 \cdots \alpha_{m-1} p_m \alpha_m$ such that

$$\begin{aligned} u^T J^*_{\alpha_0} A_{p_1} J^*_{\alpha_1} \cdots J^*_{\alpha_{m-1}} A_{p_m} J^*_{\alpha_m} v &= 1 \quad \text{and} \\ y^T K^*_{\alpha_0} B_{p_1} K^*_{\alpha_1} \cdots K^*_{\alpha_{m-1}} B_{p_m} K^*_{\alpha_m} z &= 0 \end{aligned}$$

Let $u_0 = u$ and $y_0 = y$. We guess α_0, p_1, α_1, p_2, α_2, ... in that order. After guessing α_i, $i \geq 0$, we calculate J_{α_i} and K_{α_i} and their reflexive transitive closures $J^*_{\alpha_i}$ and $K^*_{\alpha_i}$, then calculate the 0-1 column vectors w_i and x_i such that

$$\begin{aligned} w_i^T &= u_i^T J^*_{\alpha_i} \\ x_i^T &= y_i^T K^*_{\alpha_i} \end{aligned}$$

After guessing p_i, $i \geq 1$, we calculate A_{p_i} and B_{p_i}, then calculate the 0-1 column vectors u_i and y_i such that

$$\begin{aligned} u_i^T &= w_{i-1}^T A_{p_i} \\ y_i^T &= x_{i-1}^T B_{p_i} \end{aligned}$$

It follows inductively that

$$\begin{aligned} w_i^T &= u_0^T J^*_{\alpha_0} A_{p_1} J^*_{\alpha_1} \cdots J^*_{\alpha_{i-1}} A_{p_i} J^*_{\alpha_i} \\ x_i^T &= y_0^T K^*_{\alpha_0} B_{p_1} K^*_{\alpha_1} \cdots K^*_{\alpha_{i-1}} B_{p_i} K^*_{\alpha_i} \end{aligned}$$

We halt and accept if at any point $w_i^T v = 1$ and $x_i^T z = 0$.

The correctness of this algorithm follows from Lemma 17.5. It uses at most polynomial space, since in each stage of the computation only the vectors w_i and x_i need be remembered.

The algorithm can be made deterministic using Savitch's Theorem (see [4]). The problem is *PSPACE*-hard, as shown in Lecture **??**. We have thus shown

Theorem 17.6 *The equational theory of* KAT *is PSPACE-complete.*

In the next lecture, we will show that PHL is also *PSPACE*-hard, thus there is no benefit to PHL over KAT.

References

[1] Ernie Cohen, February 1994. Personal communication.

[2] Ernie Cohen, Dexter Kozen, and Frederick Smith. The complexity of Kleene algebra with tests. Technical Report 96-1598, Computer Science Department, Cornell University, July 1996.

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

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

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

[6] Dexter Kozen. On the complexity of reasoning in Kleene algebra. *Information and Computation*, 179:152–162, 2002.

[7] Dexter Kozen and Frederick Smith. Kleene algebra with tests: Completeness and decidability. In D. van Dalen and M. Bezem, editors, *Proc. 10th Int. Workshop Computer Science Logic (CSL'96)*, volume 1258 of *Lecture Notes in Computer Science*, pages 244–259, Utrecht, The Netherlands, September 1996. Springer-Verlag.