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 \( \text{Reg}_{P,B} \), 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 \( \Pi_1^1 \)-complete [6].

Matrix Algebras

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

\[
(\text{Mat}(n,K), \Delta(n,B), +, \cdot, *, \neg, 0_n, I_n)
\]

again forms a Kleene algebra with tests, where \( \text{Mat}(n,K) \) 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:

\[
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix}^* = \begin{bmatrix}
(A + BD^*C)^* D^* + D^*C(A + BD^*C)^* BD^* \\
D^*C(A + BD^*C)^* & (A + BD^*C)^* BD^*
\end{bmatrix}
\]  \( (17.1) \)

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 $H : \text{Mat}(n, K) \to \text{Mat}(n, K')$ 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 $\ast$, we can use the inductive definition (17.1) to give a straightforward inductive proof that $H(E^\ast) = H(E)^\ast$. Finally, for $E \in \Delta(n, B)$,

$$H(\overline{E})_{ij} = \begin{cases} h(E_{ij}), & \text{if } i = j, \\ h(0), & \text{if } i \neq j \end{cases} = \begin{cases} H(E_{ij}), & \text{if } i = j, \\ 0, & \text{if } i \neq j \end{cases} = H(\overline{E})_{ij},$$

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

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 $a^\ast = (1 + a)^m$. Any finite algebra is finitary, and any finitary algebra is star-continuous.

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

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

$$E = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$
where $A$ and $D$ are square. Using the denesting rule,

$$
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix}^* = \left( \begin{bmatrix}
A & 0 \\
0 & D
\end{bmatrix} + \begin{bmatrix}
0 & B \\
C & 0
\end{bmatrix} \right)^*
$$

$$
= \left( \begin{bmatrix}
A & 0 \\
0 & D
\end{bmatrix}^* \begin{bmatrix}
0 & B \\
C & 0
\end{bmatrix} \right) \begin{bmatrix}
A & 0 \\
0 & D
\end{bmatrix}^*
$$

$$
= \left( \begin{bmatrix}
A^* & 0 \\
0 & D^*
\end{bmatrix} \begin{bmatrix}
0 & B \\
C & 0
\end{bmatrix} \right) \begin{bmatrix}
A^* & 0 \\
0 & D^*
\end{bmatrix}
$$

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

$$
\begin{bmatrix}
A^* & 0 \\
0 & D^*
\end{bmatrix} = \left( I + A \right)^m \begin{bmatrix}
0 & (I + D)^m \\
0 & I + D
\end{bmatrix}
$$

$$
\leq (I + E)^m
$$

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

$$
\begin{bmatrix}
0 & A^*B \\
D^*C & 0
\end{bmatrix}^* = \begin{bmatrix}
A^*BD^*C & 0 \\
0 & D^*CA^*B
\end{bmatrix}^* \begin{bmatrix}
I & A^*B \\
D^*C & I
\end{bmatrix}
$$

$$
= \begin{bmatrix}
(A^*BD^*C)^* & 0 \\
0 & (D^*CA^*B)^*
\end{bmatrix} \begin{bmatrix}
I & A^*B \\
D^*C & I
\end{bmatrix}
$$

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

$$
\begin{bmatrix}
(A^*BD^*C)^* & 0 \\
0 & (D^*CA^*B)^*
\end{bmatrix}
$$

$$
= \left( I + A^*BD^*C \right)^k \begin{bmatrix}
0 & (I + D^*CA^*B)^k \\
0 & I + D^*CA^*B
\end{bmatrix}
$$

$$
= \left( I + \left[ \begin{bmatrix}
A^* & 0 \\
0 & D^*
\end{bmatrix} \begin{bmatrix}
0 & B \\
C & 0
\end{bmatrix} \begin{bmatrix}
A^* & 0 \\
0 & D^*
\end{bmatrix} \begin{bmatrix}
0 & B \\
C & 0
\end{bmatrix} \right]^k
$$

$$
\leq (I + (I + E)^m E(I + E)^m E)^k
$$

$$
\leq (I + E)^{2(k(m+1))}
$$
Similarly,
\[
\begin{bmatrix}
I & A^*B \\
D^*C & I
\end{bmatrix} = I + \begin{bmatrix}
A^* & 0 \\
0 & D^*
\end{bmatrix} \begin{bmatrix}
0 & B \\
C & 0
\end{bmatrix}
\leq I + (I + E)^m E
\leq (I + E)^{m+1}
\]

Putting these all together, we have
\[E^* \leq (I + E)^{2km + 2k + 2m + 1}\]

\[\square\]

**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 $\text{Mat}(n, B)$ is a subalgebra of $\text{Mat}(n, K)$. (Note that it is not the distinguished Boolean algebra of $\text{Mat}(n, K)$; 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 $\text{Mat}(n, K)$ and $\text{Mat}(n, B)$.

Since $b^* = 1$ for any $b \in B$, it follows immediately from Lemma 17.3 that $\text{Mat}(n, B)$ is finitary. In fact, it can be established by combinatorial means that if $A \in \text{Mat}(n, B)$, 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 $J \in \text{Mat}(n, B)$ and an atom $\alpha$, let $J_\alpha$ be the 0-1 matrix
\[
(J_\alpha)_{ij} = \begin{cases} 
1, & \text{if } \alpha \leq J_{ij} \\
0, & \text{otherwise.}
\end{cases}
\]

**Lemma 17.4**

\[\alpha \leq (J^*)_{ij} \iff (J^*_\alpha)_{ij} = 1\]

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

4
Then $J_\alpha = H_\alpha(J)$, where $H_\alpha$ is the componentwise extension of $h_\alpha$ to matrices. The condition to be proved is equivalent to the statement $H_\alpha(J^*) = H_\alpha(J)^*$. The entries of $J_\alpha$ can be determined by testing whether $\alpha \leq b$, which essentially amounts to evaluating a Boolean expression on a given truth assignment. The matrix $J_\alpha$ 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 $J_\alpha^*$ can be computed efficiently using any standard transitive closure algorithm.

\section*{Matrix Representation of Terms}

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

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

$$G(p) = R(\hat{p}).$$

Then deciding whether $G(p) = G(q)$ reduces to deciding whether $R(\hat{p}) = R(\hat{q})$, which we know how to do in $PSPACE$. Unfortunately, the construction of $\hat{p}$ from $p$ as given in Lecture ?? involves an exponential blowup, which the following example shows to be unavoidable. Suppose $k = 2m$. Consider the expression

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

This expression represents the set of atoms in which the $i^{th}$ and $m+i^{th}$ 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 $2^n$ states. Since the translation between regular expressions and nondeterministic automata is linear, any regular expression $\hat{p}$ such that $R(\hat{p}) = G(p)$ 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 $P \in \text{Mat}(n, \mathcal{F})$ with small entries and 0-1 vectors $u, v$ of length $n$ such that

$$R(p) = R(u^T P^* v) ,$$

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
The matrix $P$ is the transition matrix of the automaton equivalent to the regular expression $p$ over the input alphabet $\mathcal{P} \cup \mathcal{B} \cup \overline{\mathcal{B}}$. 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 $\mathcal{B} \cup \overline{\mathcal{B}}$ and the entries of $A$ are sums of elements of $\mathcal{P}$. Using the denesting rule of $\text{KA}$, we can then write

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

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 $\alpha_i$ being handled by $J^*$ and the symbols $p_i$ by $A$.

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

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

**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

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

**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

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

we have that

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

$$\Leftrightarrow$$

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

Furthermore, by the definition of matrix multiplication, this occurs iff there exist $s_0$, $t_0$, $s_1$, $t_1$, $s_2$, $t_2$, $\ldots$, $s_m$, $t_m$ such that

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

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

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

By the definition of Boolean matrix multiplication, this occurs iff

\[
    u^T J_{\alpha_0}^* A_{p_1} J_{\alpha_1}^* \cdots J_{\alpha_m-1}^* A_{p_m} J_{\alpha_m}^* v = 1
\]

\( \square \)

A \textit{PSPACE} Algorithm

Now we give a \textit{PSPACE} algorithm for deciding whether 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

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

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

\[
    R(p) = R(u^T P^* v) \\
    R(q) = R(y^T Q^* z).
\]

By the fact that

\[
    G(p) = \bigcup_{x \in R(p)} G(x)
\]

proved in Lecture ??, we also have

\[
    G(p) = G(u^T P^* v) \\
    G(q) = G(y^T Q^* z)
\]
Writing $P = J + A$ and $Q = K + B$ where the entries of $J$ and $K$ are sums of elements of $\mathcal{B} \cup \mathcal{B}$ and the entries of $A$ and $B$ are sums of elements of $P$, we have

\[
G(p) = G(u^T(J^*A)J^*v) \\
G(q) = G(y^T(K^*B)K^*z)
\]

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

\[
\begin{align*}
&u^T J_{\alpha_0}^* A_{p_1}^* J_{\alpha_1}^* \cdots J_{\alpha_{m-1}}^* A_{p_m}^* J_{\alpha_m}^* v = 1 \quad \text{and} \\
y^T K_{\alpha_0}^* B_{p_1}^* K_{\alpha_1}^* \cdots K_{\alpha_{m-1}}^* B_{p_m}^* K_{\alpha_m}^* z = 0
\end{align*}
\]

Let $u_0 = u$ and $y_0 = y$. We guess $\alpha_0, p_1, \alpha_1, p_2, \alpha_2, \ldots$ in that order. After guessing $\alpha_i$, $i \geq 0$, we calculate $J_{\alpha_i}$ and $K_{\alpha_i}$ 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

\[
\begin{align*}
w_i^T &= u_i^T J_{\alpha_i}^* \\
x_i^T &= y_i^T K_{\alpha_i}^*
\end{align*}
\]

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

\[
\begin{align*}
u_i^T &= w_{i-1}^T A_{p_i} \\
y_i^T &= x_{i-1}^T B_{p_i}
\end{align*}
\]

It follows inductively that

\[
\begin{align*}
w_i^T &= u_0^T J_{\alpha_0}^* A_{p_1}^* J_{\alpha_1}^* \cdots J_{\alpha_{i-1}}^* A_{p_i}^* J_{\alpha_i}^* \\
x_i^T &= y_0^T K_{\alpha_0}^* B_{p_1}^* K_{\alpha_1}^* \cdots K_{\alpha_{i-1}}^* B_{p_i}^* K_{\alpha_i}^*
\end{align*}
\]

We halt and accept if at any point $w_i^Tv = 1$ and $x_i^Tz = 0$.

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 $\text{KAT}$ is $PSPACE$-complete.

In the next lecture, we will show that $\text{PHL}$ is also $PSPACE$-hard, thus there is no benefit to $\text{PHL}$ over $\text{KAT}$.
References


