Completeness and the FMP for KA, revisited

Tobias Kappé

(i)Po(m)set Project Online Seminar, June 16, 2023
Laws of Kleene algebra apply to many programming language semantics. This means we can use KA to reason about program semantics. What can we (not) prove using these laws? When is something not true only by the laws of KA?
Definition (Kleene algebra)

A Kleene algebra is a tuple \((K, +, \cdot, *, 0, 1)\) where for all \(x, y, z \in K\), we have:

\[
\begin{align*}
x + x &= x & 1 + x \cdot x^* &= x^* & 1 + x^* \cdot x &= x^* \\
x + y \cdot z &\leq z & y^* \cdot x &\leq z & x \cdot z^* &\leq y
\end{align*}
\]

in addition to the “usual” laws for \(+\) and \(\cdot\) — associativity, distributivity, etc.

Here, \(x \leq y\) is a shorthand for \(x + y = y\).
Kleene algebra

Languages

Fix a (finite) set of letters $\Sigma$, and write $\Sigma^*$ for the set of words over $\Sigma$.

Example (KA of languages)
The KA of languages over $\Sigma$ is given by $(\mathcal{P}(\Sigma^*), \cup, \cdot,* , \emptyset, \{\epsilon\})$, where
- $\mathcal{P}(\Sigma^*)$ is the set of sets of words (languages);
- $\cdot$ is pointwise concatenation, i.e., $L \cdot K = \{wx : w \in L, x \in K\}$;
- $*$ is the Kleene star, i.e., $L^* = \{w_1 \cdots w_n : w_1, \ldots, w_n \in L\}$;
- $\epsilon$ is the empty word.
Kleene algebra

Relations

Fix a (not necessarily finite) set of states $S$.

Example (KA of relations)
The KA of relations over $S$ is given by $(\mathcal{P}(S \times S), \cup, \circ, \ast, \emptyset, \Delta)$, where

- $\mathcal{P}(S \times S)$ is the set of relations on $S$;
- $\circ$ is relational composition.
- $\ast$ is the reflexive-transitive closure.
- $\Delta$ is the identity relation.
Kleene algebra

Reasoning example

Claim

In every KA $K$ and for all $u, v \in K$, it holds that $(u \cdot v)^* \cdot u \leq u \cdot (v \cdot u)^*$.  

Proof. First, let’s recall the fixpoint rule:

\[
\frac{x + y \cdot z \leq z}{y^* \cdot x \leq z}
\]

It suffices to prove that $u + u \cdot v \cdot u \cdot (v \cdot u)^* \leq u \cdot (v \cdot u)^*$; we derive:

\[
u + u \cdot v \cdot u \cdot (v \cdot u)^* = u \cdot (1 + v \cdot u \cdot (v \cdot u)^*) = u \cdot (v \cdot u)^*
\]
Kleene algebra

Expressions

Definition
Exp is the set of regular expressions, generated by

\[ e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e^* \]

Definition
Given a KA \((K, +, \cdot, *, 0, 1)\) and \(h : \Sigma \to K\), we define \(\hat{h} : \text{Exp} \to K\) by

\[
\begin{align*}
\hat{h}(0) &= 0 & \hat{h}(e + f) &= \hat{h}(e) + \hat{h}(f) \\
\hat{h}(1) &= 1 & \hat{h}(e \cdot f) &= \hat{h}(e) \cdot \hat{h}(f) \\
\hat{h}(a) &= h(a) & \hat{h}(e^*) &= \hat{h}(e)^*
\end{align*}
\]
Let \( e, f \in \text{Exp} \); we write \( K \models e = f \) when \( \hat{h}(e) = \hat{h}(f) \) for all \( h \).

**Examples**

- We showed just now that \( K \models (a \cdot b)^* \cdot a \leq a \cdot (b \cdot a)^* \) for all KAs \( K \).
- \( \mathcal{P}(\Sigma^*) \models e = f \) when \( e \) and \( f \) denote the same regular language.
- \( \mathcal{P}(S \times S) \models (a + 1)^* = a^* \) because \( (R \cup \Delta)^* = R^* \) for all relations \( R \).
Let $e, f \in \text{Exp}$. We write . . .

- $\vdash e = f$ when $e = f$ follows from the axioms of KA.
- $\models e = f$ when $K \models e = f$ for every KA $K$.
- $\mathfrak{F} \models e = f$ when $K \models e = f$ holds in every finite KA $K$.
- $\mathfrak{R} \models e = f$ when $\mathcal{P}(S \times S) \models e = f$ for all $S$. 
Kleene algebra

Model theory

\[ \mathcal{F} \models e = f \quad \text{(Palka 2005)} \]

\[ \vdash e = f \quad \text{FMP} \]

\[ \mathcal{P}(\Sigma^*) \models e = f \quad \text{(Kozen 1994)} \]

\[ \models e = f \quad \text{Completeness} \]

\[ \mathcal{R} \models e = f \quad \text{(Pratt 1980)} \]
This talk

Palka’s proof of the FMP relies on Kozen’s completeness theorem.

... an independent proof of [the finite model property] would provide a quite different proof of the Kozen completeness theorem, based on purely logical tools. We defer this task to further research. (Palka 2005)

We found such a proof — with many ideas inspired by Palka.

Roadmap: Given $e, f \in \text{Exp}$ we do the following:
1. Turn expressions $e, f$ into a finite automaton $A$
2. Turn the finite automaton $A$ into a finite monoid $M$
3. Turn the finite monoid $M$ into a finite KA $K$
Expressions to automata

Definition
An automaton is a tuple \((Q, \to, I, F)\) where

- \(Q\) is a finite set of states; and
- \(\to \subseteq Q \times \Sigma \times Q\) is the transition relation; and
- \(I \subseteq Q\) is the set of initial states
- \(F \subseteq Q\) is the set of accepting states

We write \(q \xrightarrow{a} q'\) when \((q, a, q') \in \to\).
Expressions to automata

Definition
Let \((Q, \rightarrow, F)\) be an automaton. A solution is a function \(s : Q \rightarrow \mathbf{Exp}\) such that

\[
\vdash F(q) + \sum_{q \xrightarrow{a} q'} a \cdot s(q') \leq s(q)
\]

\[
F(q) = \begin{cases} 
1 & q \in F \\
0 & q \not\in F 
\end{cases}
\]

Example
For the automaton on the right, a solution satisfies

\[
\vdash 1 + a \cdot s(q_0) + b \cdot s(q_1) \leq s(q_0)
\]

\[
\vdash 0 + a \cdot s(q_1) + b \cdot s(q_0) \leq s(q_1)
\]

E.g., \(s(q_0) = (a + b \cdot a^* \cdot b)^*\) and \(s(q_1) = a^* \cdot b \cdot s(q_0)\).
Expressions to automata

Theorem (Kleene 1956; see also Conway 1971)

Every automaton admits a least solution (unique up to equivalence).

When $A$ is an automaton, we write $A(q)$ for its least solution at $q$.

Lemma (c.f. Kleene 1956; Antimirov 1996; Kozen 2001; Jacobs 2006)

For every $e$, we can construct an automaton $A_e = (Q_e, \rightarrow_e, l_e, F_e)$ such that

$$
\vdash e = \sum_{q \in l_e} A_e(q)
$$
Automata to monoids

Let $A = (Q, \rightarrow, I, F)$ be an automaton.

**Definition (Transition monoid; McNaughton and Papert 1968)**

$(M_A, \circ, \Delta)$ is the monoid where $M_A = \{a_1 \circ \cdots \circ a_n : a_1, \ldots, a_n \in \Sigma\}$.

**Example**

The transition monoid for the automaton $A$ on the right is carried by $M_A = \{\rightarrow_a, \rightarrow_b\}$, where

$\rightarrow_a = \{(q_0, q_0), (q_1, q_1)\} \quad \rightarrow_b = \{(q_0, q_1), (q_0, q_1)\}$
Automata to monoids

Definition (Transition automaton; McNaughton and Papert 1968)
Let $R \in M_A$. We write $A[R]$ for the transition automaton $(M_A, \rightarrow_o, \Delta, \{R\})$ where

$$P \xrightarrow{a} Q \iff P \circ a \xrightarrow{} = Q$$

Lemma (Solving transition automata)

$$\vdash A(q) = \sum_{qRq_f \in F} A[R](\Delta)$$
Monoids to Kleene algebras

Lemma (Palka 2005)

Let \((M, \cdot, 1)\) be a monoid. Now \((\mathcal{P}(M), \cup, \otimes, \circ, \emptyset, \{1\})\) is a KA, where

\[ T \otimes U = \{ t \cdot u : t \in T \land u \in U \} \quad T^{\circ} = \{ t_1 \cdots t_n : t_1, \ldots, t_n \in T \} \]

Lemma

Let \(A\) be an automaton, and let \(h : \Sigma \to \mathcal{P}(M_A)\) where \(h(a) = \{ \overrightarrow{a} \}\). Now

\[ R \in \hat{h}(A(q)) \iff q R q_f \in F \]
Putting it all together

In the sequel, fix $e, f \in \text{Exp}$, and:

- Let $A_{e,f} = (Q_{e,f}, \rightarrow_{e,f}, l_{e,f}, F_{e,f})$ be the disjoint union of $A_e$ and $A_f$.
- Let $M_{e,f} = (M_{A_{e,f}}, \circ, \Delta)$ be the monoid of $A_{e,f}$.

Lemma (Normal form)

Let $e, f \in \text{Exp}$ and $h : \Sigma \to \mathcal{P}(M_{e,f})$ be given by $h(a) = \{ \rightarrow_{e,f} \}$. The following hold:

\[ \vdash e = \sum_{R \in \hat{h}(e)} A_{e,f}[R](\Delta) \]

\[ \vdash f = \sum_{R \in \hat{h}(f)} A_{e,f}[R](\Delta) \]
Putting it all together
Finite model property

Theorem (Finite model property)

If $\mathcal{G} \models e = f$ then $\vdash e = f$.

Proof.
$\mathcal{P}(M_{e,f})$ is a finite KA; hence $\mathcal{P}(M_{e,f}) \models e = f$, i.e., $\hat{h}(e) = \hat{h}(f)$. But then:

$$\vdash e = \sum_{R \in \hat{h}(e)} A_{e,f}[R](\Delta) = \sum_{R \in \hat{h}(f)} A_{e,f}[R](\Delta) = f$$
Theorem (Completeness)

If $\mathcal{P}(\Sigma^\ast) \models e = f$ then $\vdash e = f$.

Proof.

Let $L : \Sigma \to \mathcal{P}(\Sigma^\ast)$ be given by $L(a) = \{a\}$.

We can show that $\hat{h}(e) = \{\overrightarrow{a_1}_{e,f} \circ \cdots \circ \overrightarrow{a_n}_{e,f} : a_1 \cdots a_n \in \hat{L}(e)\}$, and similarly for $f$.

If $\mathcal{P}(\Sigma^\ast) \models e = f$, then $\hat{L}(e) = \hat{L}(f)$, so $\hat{h}(e) = \hat{h}(f)$. The rest proceeds as before. \qed
Coq formalization

- All results formalized in the Coq proof assistant.
- Trusted base:
  - Calculus of Inductive Constructions.
  - Streicher’s *axiom K*.
  - Dependent functional extensionality.
- Some concepts are encoded differently; ideas remain the same.
Pomsets

Expressions in concurrent KA (CKA) are generated by

\[ e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e \parallel f \mid e^* \mid e^\dagger \]

Definition (Bi-KA)

A bi-KA is a tuple \((K, +, \cdot, \parallel, *, \dagger, 0, 1)\) where

- \((K, +, \cdot, *)\) and \((K, +, \parallel, \dagger)\) are both KAs, and
- \(\parallel\) commutes, i.e., \(K \models e \parallel f = f \parallel e\).

A weak bi-KA is a bi-KA without the \(\dagger\).

Definition (Concurrent KA)

A (weak) concurrent KA is a (weak) bi-KA \(K\) satisfying

\[ (e \parallel g) \cdot (f \parallel h) \leq (e \cdot f) \parallel (g \cdot h) \]
Pomsets

Example

The bi-KA of pomset languages over \( \Sigma \) is \((\mathcal{P}(\text{Pom}(\Sigma)), \cup, \cdot, \|, *, \dagger, \emptyset, \{1\})\), where

- \( \text{Pom}(\Sigma) \) denotes the set of pomsets over \( \Sigma \);
- 1 denotes the empty pomset;
- \( L \cdot L' = \{U \cdot V : U \in L, V \in L'\} \) and similarly for \( \| \); and
- \( L^* = \{1\} \cup L \cup L \cdot L \cup \cdots \) and \( L^\dagger = \{1\} \cup L \cup L \| L \cup \cdots \).
Example

The concurrent KA of pomset ideals over $\Sigma$ is $(\mathcal{I}(\Sigma), \cup, \cdot, \parallel, *, ^\dagger, \emptyset, \{1\})$, where

- $\mathcal{I}(\Sigma)$ contains the pomset languages downward-closed under $\sqsubseteq$; and
- the operators are as for bi-KA, but followed by downward closure under $\sqsubseteq$. 
Pomsets

Theorem (Laurence and Struth 2014)

Let $e$ and $f$ be (weak) concurrent KA expressions.

Now $\mathcal{P}(\text{Pom}(\Sigma)) \models e = f$ if and only if $K \models e = f$ for all (weak) bi-KAs $K$

Theorem (Laurence and Struth 2017; K., Brunet, Silva, et al. 2018)

Let $e$ and $f$ be weak concurrent KA expressions.

Now $\mathcal{I}(\Sigma) \models e = f$ if and only if $K \models e = f$ for all weak CKAs $K$
Conjecture

Let $e$ and $f$ be concurrent KA expressions.

Now $\mathcal{I}(\Sigma) \models e = f$ if and only if $K \models e = f$ for all CKAs $K$

Current techniques do not work!
<speculation>
Pomsets

The following roadmap might work:

1. Translate CKA expressions to automata
   ⇒ Pomset automata (K., Brunet, Luttik, et al. 2019)
   ⇒ or HDAs (van Glabbeek 2004; Fahrenberg 2005; Fahrenberg et al. 2022)

2. Translate these automata to ordered bimonoids (Bloom and Ésik 1996)
   ⇒ see also (Lodaya and Weil 2000; van Heerdt et al. 2021)

3. Translate bimonoids to concurrent KAs.
   ⇒ essentially the same recipe?
</speculation>
Further open questions

- Can we apply these ideas to guarded Kleene algebra with tests?
- Does KA have a finite relational model property?
- Do these techniques extend to KA with hypotheses?
- Is there a representation theorem or duality for KA?

https://kap.pe/slides  https://kap.pe/papers


References II


References IV


