Completeness and the FMP for KA, revisited

Tobias Kappé

Prague Workshop on Kleene Algebra and Many-Valued Logic
Some context

- 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*?
Kleene algebra

Languages

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

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

Definition (KA of relations)
The KA of relations over $S$ is given by $(\mathcal{R}(S), \cup, \circ, *, \emptyset, \Delta)$, where
- $\mathcal{R}(S)$ is the set of relations on $S$;
- $\circ$ is relational composition.
- $*$ is the reflexive-transitive closure.
- $\Delta$ is the identity relation.
Let $e, f \in \text{Exp}$. We write . . .

- $K, h \models e = f$ when $K$ is a KA and $h : \Sigma \to K$ with $\hat{h}(e) = \hat{h}(f)$.
- $K \models e = f$ when $K$ is a KA and $K, h \models e = f$ for all $h$.
- $\models e = f$ 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 $\mathcal{R}(S) \models e = f$ for all $S$.
- $\mathfrak{F}\mathfrak{R} \models e = f$ when $\mathcal{R}(S) \models e = f$ for all finite $S$. 

Kleene algebra

Model theory
Kleene algebra

Model theory

(Palka 2005)

\[ \mathcal{F} \models e = f \quad \iff \quad \models e = f \]

(Kozen 1994)

(Pratt 1980)

\[ \mathcal{H} \models e = f \quad \iff \quad \mathcal{P}(\Sigma^*) \models e = f \]
Lemma

If $\mathfrak{M} \models e = f$, then $\models e = f$.

Proof sketch.
We show that $\mathfrak{M} \models e = f$ implies $\mathcal{P}(\Sigma^*) \models e = f$. If $\mathcal{P}(\Sigma^*) \not\models e = f$, then (w.l.o.g.) there exists some word $x$ in the language of $e$ but not in $f$. Choose

$$S = \{w \in \Sigma^* : |w| \leq |x|\} \quad h : \Sigma \rightarrow \mathcal{R}(S), a \mapsto \{(w, wa) : wa \in S\}$$

Now $\mathcal{R}(S), h \not\models e = f$. The claim then follows by Kozen's theorem. □
This talk

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

\[ \ldots \text{an independent proof of [the finite model property] would provide a quite different proof of the Kozen completeness theorem, based on purely logical tools. We defer this task to further research.} \]  

(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 \( \text{KA} \) \( K \)
Definition
An automaton is a tuple \((Q, \rightarrow, I, F)\) where
- \(Q\) is a finite set of states; and
- \(\rightarrow \subseteq Q \times \Sigma \times Q\) 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 \((q, a, q') \in \rightarrow\).
Expressions to automata

Lemma (c.f. Kleene 1956; Brzozowski 1964; Antimirov 1996)

For every $e$, we can construct an automaton $A_e$ that accepts the language of $e$. 
Let $A = (Q, \rightarrow, I, F)$ be an automaton.

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

$(M_A, \circ, \Delta)$ is the monoid where $M_A = \{a_1 \circ \cdots \circ a_n : a_1 \cdots a_n \in \Sigma^*\}$. 
Lemma (Palka 2005)

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

\[
T \otimes U = \{t \cdot u : t \in T \land u \in U\} \quad \quad \quad T^\ast = \{t_1 \cdots t_n : t_1, \ldots, t_n \in T\}
\]
Putting it all together

Given an expression $e$, we can now obtain a finite KA $K_e = \mathcal{P}(M_{Ae})$.

Lemma
Let $e, f \in \text{Exp}$. If $K_e \models e = f$ and $K_f \models e = f$, then $\models e = f$.

Theorem (Finite model property)
If $\mathcal{F} \models e = f$ then $\models e = f$. 
Peeling the onion
Solving automata

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

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

\[
F(q) = \begin{cases} 
1 & q \in F \\
0 & q \notin F 
\end{cases}
\]

Example
For the automaton on the right, a solution satisfies

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

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

E.g., \(s(q_0) = (a + b \cdot a^* \cdot b)^*\) and \(s(q_1) = a^* \cdot b \cdot s(q_0)\).
Theorem (Kleene 1956; see also Conway 1971)

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

When $A$ is an automaton, we write

- $\overline{A}(q)$ for the least solution to $A$ at $q$
- $|A|$ for the sum of $\overline{A}(q)$ for $q \in I$

Lemma

If $e \in \text{Exp}$, then $|A_e| \leq e$. 
Definition (Transition automaton; McNaughton and Papert 1968)

Let $R \in M_A$. We write $A[R]$ for the transition automaton $(M_A, \rightarrow, \{\Delta\}, \{R\})$ where

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

Lemma (Solving transition automata)

Let $A$ be an automaton, let $q \in Q$ and let $R \in M_A$ with $q R q_f \in F$. We have

$$|A[R]| \leq \overline{A}(q)$$
Let \( h_e : \Sigma \to K_e \) be given by \( h_e(a) = \overrightarrow{a}_e \).

**Lemma**

Let \( e \in \text{Exp} \) and let \( R \in h_e(e) \). Then \( \models \overline{A_e[R]} \leq e \).

**Lemma**

Let \( e, f \in \text{Exp} \). We have that

\[
\models f \leq \sum_{R \in h_e(f)} |A_e[R]|
\]

**Proof sketch.**

By induction on \( f \).

\[\square\]
Peeling the onion
Proving the main lemma

Lemma
Let $e, f \in \text{Exp}$. If $K_e \models e = f$ and $K_f \models e = f$, then $\models e = f$.

Proof.
Since $K_e \models e = f$, we have that $h_e(e) = h_e(f)$; we can then derive

$$\models f \leq \sum_{R \in h_e(f)} [A_e[R]] = \sum_{R \in h_e(e)} [A_e[R]] \leq e$$

By a similar argument, $\models e \leq f$; the claim then follows.
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

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

Definition (Bi-KA)

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

\(\triangleright (K, +, \cdot, *)\) and \((K, +, \parallel, ^+)\) are both KAs, and

\(\triangleright \parallel\) commutes, i.e., \(K \models e \parallel f = f \parallel e\).

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

Definition (Concurrent KA)

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

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

The bi-KA of pomset languages over $\Sigma$ is $(\mathcal{P}(\text{Pom}(\Sigma)), \cup, \cdot, \|, *, \uparrow, \emptyset, \{1\})$, 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 $L^\uparrow = \{1\} \cup L \cup L \| L \cup \cdots$. 
Example

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

- $I(\Sigma)$ contains the pomset languages downward-closed under $\sqsubseteq$; and
- the operators are as for bi-KA, but followed by downward closure under $\sqsubseteq$. 
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
   \[ \Rightarrow \] Pomset automata (K., Brunet, Luttik, et al. 2019)
   \[ \Rightarrow \] or HDAs (van Glabbeek 2004; Fahrenberg 2005; Fahrenberg et al. 2022)

2. Translate these automata to *ordered bimonoids* (Bloom and Õsik 1996)
   \[ \Rightarrow \] see also (Lodaya and Weil 2000; van Heerdt et al. 2021)

3. Translate bimonoids to concurrent KAs.
   \[ \Rightarrow \] 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


