Open Universiteit

Institute for Logic, Language and Computation

Completeness and the FMP for KA, revisited

Tobias Kappé

LIACS seminar, January 22, 2024

Prelude

- The main theorems in this talk are not new, but the proofs are.

Prelude

- The main theorems in this talk are not new, but the proofs are.
- Even if the contents are technical, the techniques are elementary.

Prelude

- The main theorems in this talk are not new, but the proofs are.
- Even if the contents are technical, the techniques are elementary.
- I learned most constructions as an undergraduate, here in Leiden.

Motivation

- Laws of Kleene algebra (KA) model equivalence of regular expressions.

  ☞ Salomaa 1966; Conway 1971; Boffa 1990; Krob 1990; Kozen 1994

Motivation

- Laws of Kleene algebra (KA) model equivalence of regular expressions.

  ☞ Salomaa 1966; Conway 1971; Boffa 1990; Krob 1990; Kozen 1994

- They are also useful when reasoning about programming languages.

  ☞ Kozen and Patron 2000; Anderson et al. 2014; Smolka et al. 2015

Motivation

- Laws of Kleene algebra (KA) model equivalence of regular expressions.

  ☞ Salomaa 1966; Conway 1971; Boffa 1990; Krob 1990; Kozen 1994

- They are also useful when reasoning about programming languages.

  ☞ Kozen and Patron 2000; Anderson et al. 2014; Smolka et al. 2015

- When is something true *only by the laws of KA*?

Motivation

- Laws of Kleene algebra (KA) model equivalence of regular expressions.

  ☞ Salomaa 1966; Conway 1971; Boffa 1990; Krob 1990; Kozen 1994

- They are also useful when reasoning about programming languages.

  ☞ Kozen and Patron 2000; Anderson et al. 2014; Smolka et al. 2015

- When is something true *only by the laws of KA*?
- How can we concisely show that something is *not* provable in KA?

Kleene algebra

Definition

Definition (Kleene algebra; c.f. Kozen 1994)

A *Kleene algebra* is a tuple $(K, +, \cdot, {}^*, 0, 1)$ where

Definition (Kleene algebra; c.f. Kozen 1994)

A *Kleene algebra* is a tuple $(K, +, \cdot, {}^*, 0, 1)$ where

(1) The "usual" laws for $+$ and $\cdot$ hold (associativity, distributivity, etc. . .)

Definition (Kleene algebra; c.f. Kozen 1994)

A *Kleene algebra* is a tuple $(K, +, \cdot, ^*, 0, 1)$ where

(1) The "usual" laws for $+$ and $\cdot$ hold (associativity, distributivity, etc. . .)

(2) For all $x, y, z \in K$, the following are true:

$$x + x = x \qquad 1 + x \cdot x^* = x^* \qquad 1 + x^* \cdot x = x^*$$

$$\frac{x + y \cdot z \leq z}{y^* \cdot x \leq z} \qquad \frac{x + y \cdot z \leq y}{x \cdot z^* \leq y}$$

Here, $x \leq y$ is a shorthand for $x + y = y$.

Kleene algebra

Languages

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

Example (KA of languages)

The KA of *languages over* Σ 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., $L^* = \{w_1 \cdots w_n : w_1, \ldots, w_n \in L\}$;
- ϵ is the empty word.

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

Example (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.
- Δ is the identity relation.

Kleene algebra

Reasoning example

Claim

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

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:

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

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 $u + u \cdot v \cdot u \cdot (v \cdot u)^* \leq u \cdot (v \cdot u)^*$;

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:

$$u + u \cdot v \cdot u \cdot (v \cdot u)^*$$

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:

$$u + u \cdot v \cdot u \cdot (v \cdot u)^* = u \cdot (1 + v \cdot u \cdot (v \cdot u)^*)$$

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

$$e, f ::= 0 \mid 1 \mid \mathtt{a} \in \Sigma \mid e + f \mid e \cdot f \mid e^*$$

Kleene algebra

Expressions

Definition

Exp is the set of *regular expressions*, generated by

$$e, f ::= 0 \mid 1 \mid \mathtt{a} \in \Sigma \mid e + f \mid e \cdot f \mid e^*$$

Definition

Given a KA $(K, +, \cdot, ^*, 0, 1)$ and $h : \Sigma \to K$, we define $\widehat{h} : \mathrm{Exp} \to K$ by

$$\widehat{h}(0) = 0 \qquad \widehat{h}(\mathtt{a}) = h(\mathtt{a}) \qquad \widehat{h}(e \cdot f) = \widehat{h}(e) \cdot \widehat{h}(f)$$

$$\widehat{h}(1) = 1 \qquad \widehat{h}(e + f) = \widehat{h}(e) + \widehat{h}(f) \qquad \widehat{h}(e^*) = \widehat{h}(e)^*$$

Kleene algebra

Expressions

Definition

Exp is the set of *regular expressions*, generated by

$$e, f ::= 0 \mid 1 \mid \mathtt{a} \in \Sigma \mid e + f \mid e \cdot f \mid e^*$$

Definition

Given a KA $(K, +, \cdot, {}^*, 0, 1)$ and $h : \Sigma \to K$, we define $\widehat{h} : \mathrm{Exp} \to K$ by

$$\widehat{h}(0) = 0 \qquad \widehat{h}(\mathtt{a}) = h(\mathtt{a}) \qquad \widehat{h}(e \cdot f) = \widehat{h}(e) \cdot \widehat{h}(f)$$

$$\widehat{h}(1) = 1 \qquad \widehat{h}(e + f) = \widehat{h}(e) + \widehat{h}(f) \qquad \widehat{h}(e^*) = \widehat{h}(e)^*$$

Example

If $\ell : \Sigma \to \mathcal{P}(\Sigma^*)$ where $\ell(\mathtt{a}) = \{\mathtt{a}\}$, then $\widehat{\ell}(e)$ is the regular language denoted by e.

Kleene algebra

Model theory

Let $e, f \in \mathsf{Exp}$. We write ...

- $K, h \models e = f$ when K is a KA and $h : \Sigma \to K$ with $\widehat{h}(e) = \widehat{h}(f)$.

Let $e, f \in \mathrm{Exp}$. We write ...

- $K, h \models e = f$ when K is a KA and $h : \Sigma \to K$ with $\widehat{h}(e) = \widehat{h}(f)$.
- $K \models e = f$ when K is a KA and $K, h \models e = f$ for all h.

Kleene algebra

Model theory

Let $e, f \in \mathsf{Exp}$. We write ...

- $K, h \models e = f$ when K is a KA and $h : \Sigma \to K$ with $\widehat{h}(e) = \widehat{h}(f)$.
- $K \models e = f$ 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.

Kleene algebra

Model theory

Let $e, f \in \mathsf{Exp}$. We write ...

- $K, h \models e = f$ when K is a KA and $h : \Sigma \to K$ with $\widehat{h}(e) = \widehat{h}(f)$.
- $K \models e = f$ 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.

Kleene algebra

Model theory

Let $e, f \in \mathrm{Exp}$. We write ...

- $K, h \models e = f$ when K is a KA and $h : \Sigma \to K$ with $\widehat{h}(e) = \widehat{h}(f)$.
- $K \models e = f$ 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.

Kleene algebra

Model theory

$$\models e = f$$

$$\Updownarrow \quad \text{(Kozen 1994)}$$

$$\mathcal{P}(\Sigma^*) \models e = f$$

Kleene algebra

Model theory

$$\models e = f$$

$$\Updownarrow \quad \text{(Kozen 1994)}$$

$$\mathfrak{R} \models e = f \overset{\text{(Pratt 1980)}}{\Longleftrightarrow} \mathcal{P}(\Sigma^*) \models e = f$$

Kleene algebra

Model theory

(Palka 2005)

$$\mathfrak{F} \models e = f \iff \models e = f$$

(Kozen 1994)

$$\models e = f \iff \mathcal{P}(\Sigma^*) \models e = f$$

(Pratt 1980)

$$\mathfrak{R} \models e = f \iff \mathcal{P}(\Sigma^*) \models e = f$$

Main result

In a nutshell

Palka's proof relies on Kozen's completeness theorem.

Main result

In a nutshell

Palka's proof relies on Kozen's completeness theorem. She writes:

> *. . . 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)

Main result

In a nutshell

Palka's proof relies on Kozen's completeness theorem. She writes:

> *...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.

Main result

A roadmap

Need to show: if $\mathfrak{F} \models e = f$, then $\models e = f$.

Main result

A roadmap

Need to show: if $\mathfrak{F} \models e = f$, then $\models e = f$.

Given $e, f \in \text{Exp}$ we do the following:

1. Turn expressions e into a finite automaton A_e

Main result

A roadmap

Need to show: if $\mathfrak{F} \models e = f$, then $\models e = f$.

Given $e, f \in \mathsf{Exp}$ we do the following:

1. Turn expressions e into a finite automaton A_e
2. Convert the finite automaton A_e into a finite monoid M_e

Main result

A roadmap

Need to show: if $\mathfrak{F} \models e = f$, then $\models e = f$.

Given $e, f \in \mathrm{Exp}$ we do the following:

1. Turn expressions e into a finite automaton A_e
2. Convert the finite automaton A_e into a finite monoid M_e
3. Translate the finite monoid M_e into a finite KA K_e

Main result

A roadmap

Need to show: if $\mathfrak{F} \models e = f$, then $\models e = f$.

Given $e, f \in \text{Exp}$ we do the following:

1. Turn expressions e into a finite automaton A_e
2. Convert the finite automaton A_e into a finite monoid M_e
3. Translate the finite monoid M_e into a finite KA K_e
4. Prove something about interpretations inside K_e

Main result

A roadmap

Need to show: if $\mathfrak{F} \models e = f$, then $\models e = f$.

Given $e, f \in \text{Exp}$ we do the following:

1. Turn expressions e into a finite automaton A_e
2. Convert the finite automaton A_e into a finite monoid M_e
3. Translate the finite monoid M_e into a finite KA K_e
4. Prove something about interpretations inside K_e
5. Apply the premise that $\models e = f$

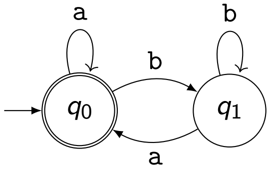
Expressions to automata

Definition

An automaton is a tuple $A = (Q, \rightarrow, I, F)$ where

- Q is a finite set of *states*; and
- $\rightarrow \subseteq Q \times \Sigma \times Q$ is the *transition relation*;
- $I \subseteq Q$ is the set of *initial states*
- $F \subseteq Q$ is the set of *accepting states*

We write $q \xrightarrow{\mathtt{a}} q'$ when $(q, \mathtt{a}, q') \in \rightarrow$.



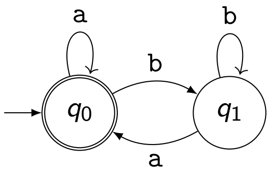
Expressions to automata

Definition

An automaton is a tuple $A = (Q, \to, I, F)$ where

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

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



The *language* of $q \in Q$ is $L_A(q) = \{\mathtt{a}_1 \cdots \mathtt{a}_n \in \Sigma^* : q \xrightarrow{\mathtt{a}_1} \circ \cdots \circ \xrightarrow{\mathtt{a}_n} q' \in F\}$

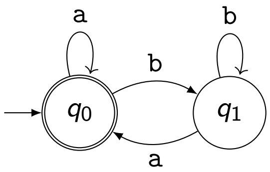
Expressions to automata

Definition

An automaton is a tuple $A = (Q, \to, I, F)$ where

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

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



The *language* of $q \in Q$ is $L_A(q) = \{a_1 \cdots a_n \in \Sigma^* : q \xrightarrow{a_1} \circ \cdots \circ \xrightarrow{a_n} q' \in F\}$

The language of A is given by $\bigcup_{q \in I} L_A(q)$.

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.

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 $M_A = \{\xrightarrow{\mathtt{a}_1} \circ \cdots \circ \xrightarrow{\mathtt{a}_n} : \mathtt{a}_1 \cdots \mathtt{a}_n \in \Sigma^*\}$.

Monoids to Kleene algebras

Lemma (Palka 2005)

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

$$T \otimes U = \{t \cdot u : t \in T \wedge u \in U\} \qquad T^{\circledast} = \{t_1 \cdots t_n : t_1, \ldots, t_n \in T\}$$

Putting it all together

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

Putting it all together

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

Lemma

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

Putting it all together

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

Lemma

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

Theorem (Finite model property)

If $\mathfrak{F} \models e = f$ *then* $\models e = f$.

Peeling the onion

Solving automata

Definition

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

$$\models F(q) + \sum_{q \xrightarrow{\mathtt{a}} q'} \mathtt{a} \cdot s(q') \leq s(q) \qquad\qquad F(q) = \begin{cases} 1 & q \in F \\ 0 & q \notin F \end{cases}$$

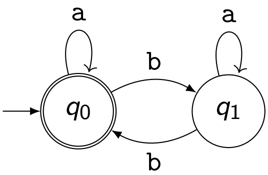
Peeling the onion

Solving automata

Example

For the automaton on the right, a solution satisfies

$$\models 1 + \mathtt{a} \cdot s(q_0) + \mathtt{b} \cdot s(q_1) \leq s(q_0)$$
$$\models 0 + \mathtt{a} \cdot s(q_1) + \mathtt{b} \cdot s(q_0) \leq s(q_1)$$



Peeling the onion

Solving automata

Example (Continued)

We start with the second condition:

$$0 + \mathtt{a} \cdot s(q_1) + \mathtt{b} \cdot s(q_0) \leq s(q_1)$$

Peeling the onion

Solving automata

Example (Continued)

We start with the second condition:

$$0 + \mathtt{a} \cdot s(q_1) + \mathtt{b} \cdot s(q_0) \leq s(q_1)$$

We can rewrite this as

$$\mathtt{a} \cdot s(q_1) + \mathtt{b} \cdot s(q_0) \leq s(q_1)$$

Example (Continued)

We start with the second condition:

$$0 + \mathtt{a} \cdot s(q_1) + \mathtt{b} \cdot s(q_0) \leq s(q_1)$$

We can rewrite this as

$$\mathtt{a} \cdot s(q_1) + \mathtt{b} \cdot s(q_0) \leq s(q_1)$$

which by the fixpoint rule implies

$$\mathtt{a}^* \cdot \mathtt{b} \cdot s(q_0) \leq s(q_1)$$

Peeling the onion

Solving automata

Example (Continued)

Now we look at the second condition

$$1 + \mathtt{a} \cdot s(q_0) + \mathtt{b} \cdot s(q_1) \leq s(q_0)$$

Peeling the onion

Solving automata

Example (Continued)

Now we look at the second condition

$$1 + \mathtt{a} \cdot s(q_0) + \mathtt{b} \cdot s(q_1) \leq s(q_0)$$

Substituting $\mathtt{a}^* \cdot \mathtt{b} \cdot s(q_0) \leq s(q_1)$ we get

$$1 + \mathtt{a} \cdot s(q_0) + \mathtt{b} \cdot \mathtt{a}^* \cdot \mathtt{b} \cdot s(q_0) \leq s(q_0)$$

Peeling the onion

Solving automata

Example (Continued)

Now we look at the second condition

$$1 + \mathtt{a} \cdot s(q_0) + \mathtt{b} \cdot s(q_1) \leq s(q_0)$$

Substituting $\mathtt{a}^* \cdot \mathtt{b} \cdot s(q_0) \leq s(q_1)$ we get

$$1 + \mathtt{a} \cdot s(q_0) + \mathtt{b} \cdot \mathtt{a}^* \cdot \mathtt{b} \cdot s(q_0) \leq s(q_0)$$

which rewrites to

$$1 + (\mathtt{a} + \mathtt{b} \cdot \mathtt{a}^* \cdot \mathtt{b}) \cdot s(q_0) \leq s(q_0)$$

Example (Continued)

Now we look at the second condition

$$1 + \mathtt{a} \cdot s(q_0) + \mathtt{b} \cdot s(q_1) \leq s(q_0)$$

Substituting $\mathtt{a}^* \cdot \mathtt{b} \cdot s(q_0) \leq s(q_1)$ we get

$$1 + \mathtt{a} \cdot s(q_0) + \mathtt{b} \cdot \mathtt{a}^* \cdot \mathtt{b} \cdot s(q_0) \leq s(q_0)$$

which rewrites to

$$1 + (\mathtt{a} + \mathtt{b} \cdot \mathtt{a}^* \cdot \mathtt{b}) \cdot s(q_0) \leq s(q_0)$$

By the fixpoint rule

$$(\mathtt{a} + \mathtt{b} \cdot \mathtt{a}^* \cdot \mathtt{b})^* \leq s(q_0)$$

Example (Continued)

We now have two lower bounds:

$$(\mathtt{a} + \mathtt{b} \cdot \mathtt{a}^* \cdot \mathtt{b})^* \leq s(q_0)$$
$$\mathtt{a}^* \cdot \mathtt{b} \cdot (\mathtt{a} + \mathtt{b} \cdot \mathtt{a}^* \cdot \mathtt{b})^* \leq s(q_1)$$

Example (Continued)

We now have two lower bounds:

$$(\mathtt{a} + \mathtt{b} \cdot \mathtt{a}^* \cdot \mathtt{b})^* \leq s(q_0)$$
$$\mathtt{a}^* \cdot \mathtt{b} \cdot (\mathtt{a} + \mathtt{b} \cdot \mathtt{a}^* \cdot \mathtt{b})^* \leq s(q_1)$$

It turns these are also solutions to A — thus we found the least solution.

Peeling the onion

Solving automata

Theorem (Kleene 1956; see also Conway 1971)

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

Peeling the onion

Solving 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

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

Peeling the onion

Solving 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

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

Lemma
If $e \in \mathrm{Exp}$, *then* $\models \lfloor A_e \rfloor \leq e$.

Definition (Transition automaton; McNaughton and Papert 1968)

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

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

Definition (Transition automaton; McNaughton and Papert 1968)

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

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

Intuition: $w \in L(A[R])$ means $q \; R \; q'$ iff w traces from q to q' in A.

Peeling the onion

Solving monoids

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

$$\models \lfloor A[R] \rfloor \leq \overline{A}(q)$$

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

$$\models \lfloor A[R] \rfloor \leq \overline{A}(q)$$

Let $h_e : \Sigma \to K_e$ be given by $h_e(\mathtt{a}) = \{\xrightarrow{\mathtt{a}}_e\}$.

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

$$\models \lfloor A[R] \rfloor \leq \overline{A}(q)$$

Let $h_e : \Sigma \to K_e$ be given by $h_e(\mathtt{a}) = \{\xrightarrow{\mathtt{a}}_e\}$.

Lemma

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

Peeling the onion

Solving Kleene algebras

Let $h_e : \Sigma \to K_e$ be given by $h_e(\mathtt{a}) = \{\xrightarrow{\mathtt{a}}_e\}$.

Peeling the onion

Solving Kleene algebras

Let $h_e : \Sigma \to K_e$ be given by $h_e(\mathtt{a}) = \{\xrightarrow{\mathtt{a}}_e\}$.

Lemma

Let $e, f \in \mathrm{Exp}$. We have that

$$\models f \leq \sum_{R \in \widehat{h_e}(f)} \lfloor A_e[R] \rfloor$$

Proof sketch.

By induction on f. □

Lemma

Let $e, f \in \mathrm{Exp}$. 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 $\widehat{h_e}(e) = \widehat{h_e}(f)$; we can then derive

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

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

Peeling the onion

The grand finale

Theorem
If $\mathfrak{F} \models e = f$, *then* $\models e = f$.

Proof.
Since K_e and K_f are finite KAs, we have that $K_e \models e = f$ and $K_f \models e = f$.

Peeling the onion

The grand finale

Theorem

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

Proof.

Since K_e and K_f are finite KAs, we have that $K_e \models e = f$ and $K_f \models e = f$.

The proof then follows by the previous lemma. □

Some thoughts

- The proof uses *Antimirov's* instead of *Brzozowski's construction*.

Some thoughts

- The proof uses *Antimirov's* instead of *Brzozowski's construction*.
- We do not rely on bisimilarity-based arguments at all (c.f. Jacobs 2006).
- We do not use the right-handed axioms for the star:

$$1 + x \cdot x^* = x^* \qquad \frac{x + y \cdot z \leq y}{x \cdot z^* \leq y}$$

Some thoughts

- The proof uses *Antimirov's* instead of *Brzozowski's construction*.
- We do not rely on bisimilarity-based arguments at all (c.f. Jacobs 2006).
- We do not use the right-handed axioms for the star:

$$1 + x \cdot x^* = x^* \qquad\qquad \frac{x + y \cdot z \leq y}{x \cdot z^* \leq y}$$

  - These were known not to be necessary

    ☞ Krob 1990; Boffa 1990; Das, Doumane, and Pous 2018; Kozen and Silva 2020

Some thoughts

- The proof uses *Antimirov's* instead of *Brzozowski's construction*.
- We do not rely on bisimilarity-based arguments at all (c.f. Jacobs 2006).
- We do not use the right-handed axioms for the star:

$$1 + x \cdot x^* = x^* \qquad\qquad \frac{x + y \cdot z \leq y}{x \cdot z^* \leq y}$$

  - These were known not to be necessary

    ☞ Krob 1990; Boffa 1990; Das, Doumane, and Pous 2018; Kozen and Silva 2020
  - Upshot: a proof-theoretic result for KA: "right-hand elimination".

Coq formalization

- All results formalized in the Coq proof assistant.

Coq formalization

- All results formalized in the Coq proof assistant.
- Trusted base:
  - Calculus of Inductive Constructions.
  - Streicher's *axiom K*.
  - Dependent functional extensionality.

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.

Further open questions

- Can we apply these ideas to *guarded Kleene algebra with tests*?

Further open questions

- Can we apply these ideas to *guarded Kleene algebra with tests*?
- Do these techniques extend to *KA with hypotheses*?

Further open questions

- Can we apply these ideas to *guarded Kleene algebra with tests*?
- Do these techniques extend to *KA with hypotheses*?
- Is there a representation theorem or duality for KA?

Further open questions

- Can we apply these ideas to *guarded Kleene algebra with tests*?
- 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`

Bonus: extending the model theory

Lemma
If $\mathfrak{FR} \models e = f$, *then* $\models e = f$.

Bonus: extending the model theory

Lemma
If $\mathfrak{FR} \models e = f$*, then* $\models e = f$*.*

Proof sketch.
We show that $\mathfrak{FR} \models e = f$ implies $\mathcal{P}(\Sigma^*) \models e = f$. For $n \in \mathbb{N}$, choose

$$\Sigma_n = \{w \in \Sigma^* : |w| \leq n\} \qquad h_n : \Sigma \to \mathcal{R}(\Sigma_n), \mathtt{a} \mapsto \{(w, w\mathtt{a}) : w\mathtt{a} \in \Sigma_n\}$$

Bonus: extending the model theory

Lemma
If $\mathfrak{F}\mathfrak{R} \models e = f$*, then* $\models e = f$*.*

Proof sketch.
We show that $\mathfrak{F}\mathfrak{R} \models e = f$ implies $\mathcal{P}(\Sigma^*) \models e = f$. For $n \in \mathbb{N}$, choose

$$\Sigma_n = \{w \in \Sigma^* : |w| \leq n\} \qquad h_n : \Sigma \to \mathcal{R}(\Sigma_n),\ \mathtt{a} \mapsto \{(w, w\mathtt{a}) : w\mathtt{a} \in \Sigma_n\}$$

For all $w \in \Sigma_n$ and regular expressions g, we now have $w \in \widehat{\ell}(g)$ iff $(\epsilon, w) \in \widehat{h_n}(g)$.

Bonus: extending the model theory

Lemma

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

Proof sketch.

We show that $\mathfrak{FR} \models e = f$ implies $\mathcal{P}(\Sigma^*) \models e = f$. For $n \in \mathbb{N}$, choose

$$\Sigma_n = \{w \in \Sigma^* : |w| \leq n\} \qquad h_n : \Sigma \to \mathcal{R}(\Sigma_n),\ \mathtt{a} \mapsto \{(w, w\mathtt{a}) : w\mathtt{a} \in \Sigma_n\}$$

For all $w \in \Sigma_n$ and regular expressions g, we now have $w \in \widehat{\ell}(g)$ iff $(\epsilon, w) \in \widehat{h_n}(g)$.

Thus $w \in \hat{\ell}(f)$ if and only if $w \in \widehat{h_{|w|}}(e) = \widehat{h_{|w|}}(f)$ if and only if $w \in \widehat{\ell}(f)$.

Bonus: extending the model theory

Lemma

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

Proof sketch.

We show that $\mathfrak{FR} \models e = f$ implies $\mathcal{P}(\Sigma^*) \models e = f$. For $n \in \mathbb{N}$, choose

$$\Sigma_n = \{w \in \Sigma^* : |w| \leq n\} \qquad h_n : \Sigma \to \mathcal{R}(\Sigma_n),\ \mathsf{a} \mapsto \{(w, w\mathsf{a}) : w\mathsf{a} \in \Sigma_n\}$$

For all $w \in \Sigma_n$ and regular expressions g, we now have $w \in \widehat{\ell}(g)$ iff $(\epsilon, w) \in \widehat{h_n}(g)$.

Thus $w \in \hat{\ell}(f)$ if and only if $w \in \widehat{h_{|w|}}(e) = \widehat{h_{|w|}}(f)$ if and only if $w \in \widehat{\ell}(f)$.

This means that $\mathcal{P}(\Sigma^*), \ell \models e = f$, whence $\mathcal{P}(\Sigma^*) \models e = f$. □

Bonus: pomsets

Expressions in *concurrent KA* (CKA) are generated by

$$e, f ::= 0 \mid 1 \mid \mathtt{a} \in \Sigma \mid e + f \mid e \cdot f \mid e \parallel f \mid e^* \mid e^\dagger$$

Bonus: pomsets

Expressions in *concurrent KA* (CKA) are generated by

$$e, f ::= 0 \mid 1 \mid \mathtt{a} \in \Sigma \mid e + f \mid e \cdot f \mid e \parallel f \mid e^* \mid e^\dagger$$

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 †.

Bonus: pomsets

Expressions in *concurrent KA* (CKA) are generated by

$$e, f ::= 0 \mid 1 \mid \mathtt{a} \in \Sigma \mid e + f \mid e \cdot f \mid e \parallel f \mid e^* \mid e^\dagger$$

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 †.

Definition (Concurrent KA)

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

$$(e \parallel g) \cdot (f \parallel h) \leq (e \cdot f) \parallel (g \cdot h)$$

Bonus: pomsets

Example

The *bi-KA of pomset languages* over Σ is $(\mathcal{P}(\mathrm{Pom}(\Sigma)), \cup, \cdot, \parallel, {}^*, {}^\dagger, \emptyset, \{1\})$, where

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

Bonus: pomsets

Example

The *concurrent KA of pomset ideals* over Σ 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 $\sqsubseteq$.

Bonus: pomsets

Theorem (Laurence and Struth 2014)

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

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

Bonus: pomsets

Theorem (Laurence and Struth 2014)

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

Now $\mathcal{P}(\mathrm{Pom}(\Sigma)) \models e = f$ 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

Bonus: pomsets

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

Bonus: pomsets

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!

Bonus: pomsets

`<speculation>`

Bonus: pomsets

The following roadmap *might* work:

Bonus: 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)

Bonus: 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)

Bonus: 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?

Bonus: pomsets

</speculation>

References I

- Anderson, Carolyn Jane et al. (2014). "NetKAT: semantic foundations for networks". In: *POPL*, pp. 113–126. DOI: 10.1145/2535838.2535862.
- Antimirov, Valentin M. (1996). "Partial Derivatives of Regular Expressions and Finite Automaton Constructions". In: *Theor. Comput. Sci.* 155.2, pp. 291–319. DOI: 10.1016/0304-3975(95)00182-4.
- Bloom, Stephen L. and Zoltán Ésik (1996). "Free Shuffle Algebras in Language Varieties". In: *Theor. Comput. Sci.* 163.1&2, pp. 55–98. DOI: 10.1016/0304-3975(95)00230-8.
- Boffa, Maurice (1990). "Une remarque sur les systèmes complets d'identités rationnelles". In: *RAIRO Theor. Informatics Appl.* 24, pp. 419–423. DOI: 10.1051/ita/1990240404191.
- Brzozowski, Janusz A. (1964). "Derivatives of Regular Expressions". In: *J. ACM* 11.4, pp. 481–494. DOI: 10.1145/321239.321249.
- Conway, John Horton (1971). *Regular Algebra and Finite Machines*. Chapman and Hall, Ltd., London.

References II

- Das, Anupam, Amina Doumane, and Damien Pous (2018). "Left-Handed Completeness for Kleene algebra, via Cyclic Proofs". In: *LPAR*, pp. 271–289. DOI: 10.29007/hzq3.
- Fahrenberg, Uli (2005). "A Category of Higher-Dimensional Automata". In: *FoSSaCS*, pp. 187–201. DOI: 10.1007/978-3-540-31982-5_12.
- Fahrenberg, Uli et al. (2022). "A Kleene Theorem for Higher-Dimensional Automata". In: *CONCUR*, 29:1–29:18. DOI: 10.4230/LIPIcs.CONCUR.2022.29.
- Jacobs, Bart (2006). "A Bialgebraic Review of Deterministic Automata, Regular Expressions and Languages". In: *Algebra, Meaning, and Computation, Essays Dedicated to Joseph A. Goguen on the Occasion of His 65th Birthday*, pp. 375–404. DOI: 10.1007/11780274_20.
- Kappé, Tobias, Paul Brunet, Bas Luttik, et al. (2019). "On series-parallel pomset languages: Rationality, context-freeness and automata". In: *J. Log. Algebr. Meth. Program.* 103, pp. 130–153. DOI: 10.1016/j.jlamp.2018.12.001.

References III

- Kappé, Tobias, Paul Brunet, Alexandra Silva, et al. (2018). "Concurrent Kleene Algebra: Free Model and Completeness". In: *ESOP*, pp. 856–882. DOI: 10.1007/978-3-319-89884-1_30.
- Kleene, Stephen C. (1956). "Representation of Events in Nerve Nets and Finite Automata". In: *Automata Studies*, pp. 3–41.
- Kozen, Dexter (1994). "A Completeness Theorem for Kleene Algebras and the Algebra of Regular Events". In: *Inf. Comput.* 110.2, pp. 366–390. DOI: 10.1006/inco.1994.1037.
- Kozen, Dexter and Maria-Christina Patron (2000). "Certification of Compiler Optimizations Using Kleene Algebra with Tests". In: *CL*, pp. 568–582. DOI: 10.1007/3-540-44957-4_38.
- Kozen, Dexter and Alexandra Silva (2020). "Left-handed completeness". In: *Theor. Comput. Sci.* 807, pp. 220–233. DOI: 10.1016/j.tcs.2019.10.040.
- Krob, Daniel (1990). "A Complete System of B-Rational Identities". In: *ICALP*, pp. 60–73. DOI: 10.1007/BFb0032022.

References IV

- Laurence, Michael R. and Georg Struth (2014). “Completeness Theorems for Bi-Kleene Algebras and Series-Parallel Rational Pomset Languages”. In: *RAMiCS*, pp. 65–82. DOI: 10.1007/978-3-319-06251-8_5.
- — (2017). *Completeness Theorems for Pomset Languages and Concurrent Kleene Algebras*. arXiv: 1705.05896.
- Lodaya, Kamal and Pascal Weil (2000). “Series-parallel languages and the bounded-width property”. In: *Theor. Comp. Sci.* 237.1, pp. 347–380. DOI: 10.1016/S0304-3975(00)00031-1.
- McNaughton, Robert and Seymour Papert (1968). “The syntactic monoid of a regular event”. In: *Algebraic Theory of Machines, Languages, and Semigroups*, pp. 297–312.
- Palka, Ewa (2005). “On Finite Model Property of the Equational Theory of Kleene Algebras”. In: *Fundam. Informaticae* 68.3, pp. 221–230. URL: http://content.iospress.com/articles/fundamenta-informaticae/fi68-3-02.

References V

- Pratt, Vaughan R. (1980). "Dynamic Algebras and the Nature of Induction". In: *STOC*, pp. 22–28. DOI: 10.1145/800141.804649.
- Salomaa, Arto (1966). "Two Complete Axiom Systems for the Algebra of Regular Events". In: *J. ACM* 13.1, pp. 158–169. DOI: 10.1145/321312.321326.
- Smolka, Steffen et al. (2015). "A fast compiler for NetKAT". In: *ICFP*, pp. 328–341. DOI: 10.1145/2784731.2784761.
- van Glabbeek, Rob J. (2004). "On the Expressiveness of Higher Dimensional Automata: (Extended Abstract)". In: *EXPRESS*, pp. 5–34. DOI: 10.1016/j.entcs.2004.11.026.
- van Heerdt, Gerco et al. (2021). "Learning Pomset Automata". In: *FoSSaCS*, pp. 510–530. DOI: 10.1007/978-3-030-71995-1_26.