The main theorems in this talk are not new, but the proofs are.
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?
Definition (Kleene algebra; c.f. Kozen 1994)

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

Definition

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. . . )

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

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

Here, \(x \leq y\) is a shorthand for \(x + y = y\).
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 $(P(\Sigma^*), \cup, \cdot, *, \emptyset, \{\epsilon\})$, where
- $P(\Sigma^*)$ 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 $S$.

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.
- $\Delta$ is the identity relation.
Kleene algebra

Reasoning example

Claim

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

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

\[
u + u \cdot v \cdot u \cdot (v \cdot u)^*
\]
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:

$$x + y \cdot z \leq z$$

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

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

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

\[ e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e^* \]
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 \(h : \Sigma \rightarrow K\), we define \(\hat{h} : \text{Exp} \rightarrow K\) by

\[
\begin{align*}
\hat{h}(0) &= 0 \\
\hat{h}(a) &= h(a) \\
\hat{h}(e \cdot f) &= \hat{h}(e) \cdot \hat{h}(f) \\
\hat{h}(1) &= 1 \\
\hat{h}(e + f) &= \hat{h}(e) + \hat{h}(f) \\
\hat{h}(e^*) &= \hat{h}(e)^*
\end{align*}
\]

Example
If \(\ell : \Sigma \rightarrow \mathcal{P}(\Sigma^*)\) where \(\ell(a) = \{a\}\), then \(\hat{h}_\ell(e)\) is the regular language denoted by \(e\).
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 \(h : \Sigma \rightarrow K\), we define \(\hat{h} : \text{Exp} \rightarrow K\) by

\[
\begin{align*}
\hat{h}(0) &= 0 \\
\hat{h}(a) &= h(a) \\
\hat{h}(e \cdot f) &= \hat{h}(e) \cdot \hat{h}(f) \\
\hat{h}(1) &= 1 \\
\hat{h}(e + f) &= \hat{h}(e) + \hat{h}(f) \\
\hat{h}(e^*) &= \hat{h}(e)^*
\end{align*}
\]

Example
If \(\ell : \Sigma \rightarrow \mathcal{P}(\Sigma^*)\) where \(\ell(a) = \{a\}\), then \(\hat{\ell}(e)\) is the regular language denoted by \(e\).
Let $e, f \in \text{Exp}$. We write . . .

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

Kleene algebra
Model theory
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 $K$. 
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 \( K \).
- \( \models e = f \) when \( K \models e = f \) holds in every finite KA \( K \).
Kleene algebra
Model theory

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 \( S \).
Kleene algebra

Model theory

\[ \models e = f \]

(Kozen 1994)

\[ \mathcal{P}(\Sigma^*) \models e = f \]
Kleene algebra

Model theory

\[ \mathcal{K} \models e = f \]

(Pratt 1980)

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

(Kozen 1994)
Kleene algebra
Model theory

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

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

(Palka 2005)

(Pratt 1980)

(Kozen 1994)
Palka’s proof relies on Kozen’s completeness theorem.
Palka’s proof relies on Kozen’s completeness theorem. She writes:

\[\ldots\text{an independent proof of } [\text{the finite model property}] \text{ would provide a quite different proof of the Kozen completeness theorem, based on purely logical tools. We defer this task to further research.} \quad \text{(Palka 2005)}\]
Main result
In a nutshell

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

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

(Palka 2005)

We found such a proof — with many ideas inspired by Palka.
Main result
A roadmap

Need to show: if $\mathcal{F} \models e = f$, then $\models e = f$. 
Main result
A roadmap

Need to show: if $\mathcal{G} \models e = f$, 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 $\mathcal{F} \models e = f$, 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 $M_e$
Main result

A roadmap

Need to show: if $\exists \models e = f$, 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 $K_e$
Main result
A roadmap

Need to show: if $\mathcal{F} \models e = f$, 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 $\mathcal{F} \models e = f$, 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 $A = (Q, \to, I, F)$ 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 $L_A(q) = \{ a_1 \cdots a_n \in \Sigma^* : q \xrightarrow{a_1} \circ \cdots \circ a_n \xrightarrow{a_n} q' \in F \}$

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

Diagram:

```
 Automaton diagram with states q0 and q1 connected by transitions:
 q0 -> a -> q1
 q0 -> b -> q1
```

`a` and `b` are labels on the transitions.
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 \( q \xrightarrow{a} q' \) when \((q, a, q') \in \rightarrow\).

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

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

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* $e$. 
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 \((\mathcal{P}(M), \cup, \otimes, \circledast, \emptyset, \{1\})\) is a KA, where

\[
T \otimes U = \{t \cdot u : t \in T \land u \in U\} \\
T^\circledast = \{t_1 \cdots t_n : t_1, \ldots, t_n \in T\}
\]
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 $K_e = \mathcal{P}(M_{Ae})$.

Lemma
Let $e, f \in \text{Exp}$. 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 $K_e = \mathcal{P}(M_{Ae})$.

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 $\mathcal{M} \models e = f$ 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

\[
\begin{align*}
\models F(q) + \sum_{q \xrightarrow{a} q'} a \cdot s(q') & \leq s(q) \\
F(q) &= \begin{cases} 
1 & q \in F \\
0 & q \notin F
\end{cases}
\end{align*}
\]
Example

For the automaton on the right, a solution satisfies

\[ 1 + a \cdot s(q_0) + b \cdot s(q_1) \leq s(q_0) \]
\[ 0 + a \cdot s(q_1) + b \cdot s(q_0) \leq s(q_1) \]
Example (Continued)

We start with the second condition:

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

We start with the second condition:

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

We can rewrite this as

\[ a \cdot s(q_1) + b \cdot s(q_0) \leq s(q_1) \]
Example (Continued)

We start with the second condition:

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

We can rewrite this as

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

which by the fixpoint rule implies

\[ a^* \cdot b \cdot s(q_0) \leq s(q_1) \]
Example (Continued)

Now we look at the second condition

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

Now we look at the second condition

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

Substituting \( a^* \cdot b \cdot s(q_0) \leq s(q_1) \) we get

\[ 1 + a \cdot s(q_0) + b \cdot a^* \cdot b \cdot s(q_0) \leq s(q_0) \]
Example (Continued)

Now we look at the second condition

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

Substituting \( a^* \cdot b \cdot s(q_0) \leq s(q_1) \) we get

\[ 1 + a \cdot s(q_0) + b \cdot a^* \cdot b \cdot s(q_0) \leq s(q_0) \]

which rewrites to

\[ 1 + (a + b \cdot a^* \cdot b) \cdot s(q_0) \leq s(q_0) \]
Peeling the onion
Solving automata

Example (Continued)

Now we look at the second condition

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

Substituting \( a^* \cdot b \cdot s(q_0) \leq s(q_1) \) we get

\[ 1 + a \cdot s(q_0) + b \cdot a^* \cdot b \cdot s(q_0) \leq s(q_0) \]

which rewrites to

\[ 1 + (a + b \cdot a^* \cdot b) \cdot s(q_0) \leq s(q_0) \]

By the fixpoint rule

\[ (a + b \cdot a^* \cdot b)^* \leq s(q_0) \]
Example (Continued)

We now have two lower bounds:

\[(a + b \cdot a^* \cdot b)^* \leq s(q_0)\]
\[a^* \cdot b \cdot (a + b \cdot a^* \cdot b)^* \leq s(q_1)\]
Example (Continued)

We now have two lower bounds:

\[(a + b \cdot a^* \cdot b)^* \leq s(q_0)\]
\[a^* \cdot b \cdot (a + b \cdot a^* \cdot b)^* \leq s(q_1)\]

It turns these are also solutions to $A$ — thus we found the least solution.
Theorem (Kleene 1956; see also Conway 1971)

Every automaton admits a least solution (unique up to equivalence).
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 $q \in I$
Theorem (Kleene 1956; see also Conway 1971)

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

When $A$ is an automaton, we write

- $\bar{A}(q)$ for the least solution to $A$ at $q$
- $|A|$ for the sum of $\bar{A}(q)$ 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 $(M_A, \rightarrow_o, \{\Delta\}, \{R\})$ where

\[
P \xrightarrow{\rightarrow_o} Q \iff P \circ \xrightarrow{a} = Q
\]

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

Peeling the onion
Solving monoids
Definition (Transition automaton; McNaughton and Papert 1968)
Let \( R \in M_A \). We write \( A[R] \) for the transition automaton \((M_A, \to_\circ, \{\Delta\}, \{R\})\) where

\[
P \xrightarrow{a} Q \iff P \circ a \to = Q
\]

Intuition: \( w \in L(A[R]) \) means \( q R q' \) iff \( w \) traces from \( q \) to \( q' \) in \( A \).
Lemma (Solving transition automata)

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

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

$$|A[R]| \leq \overline{A}(q)$$

Let $h_e : \Sigma \to K_e$ be given by $h_e(a) = \{ \overset{a}{\rightarrow} \}$. 
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

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

Let $h_e : \Sigma \to K_e$ be given by $h_e(a) = \{a\}^e$.

Lemma

Let $e \in \text{Exp}$ and let $R \in \hat{h}_e(e)$. Then $\models \overline{A_e}[R] \leq e$. 
Let $h_e : \Sigma \rightarrow K_e$ be given by $h_e(a) = \{\rightarrow_e^a\}$. 
Let $h_e : \Sigma \rightarrow K_e$ be given by $h_e(a) = \{a \rightarrow e\}$.

**Lemma**

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

$$\models f \leq \sum_{R \in \hat{h}_e(f)} [A_e[R]]$$

**Proof sketch.**

By induction on $f$. 

\[\square\]
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 $\hat{h}_e(e) = \hat{h}_e(f)$; we can then derive

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

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

If $\mathcal{S} \models e = f$, then $\models e = f$.

Proof.

Since $K_e$ and $K_f$ are finite KAs, we have that $K_e \models e = f$ and $K_f \models e = f$. 
Theorem
If $\mathcal{F} \models e = f$, 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:

\[ 1 + x \cdot x^* = x^* \]

\[ x + y \cdot z \leq y \]

\[ x \cdot z^* \leq y \]
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:

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

- 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:

\[
1 + x \cdot x^* = x^* \quad \text{and} \quad x + y \cdot z \leq y \\
\Rightarrow x \cdot z^* \leq y
\]

- 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”.

28 / 30
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 $\mathfrak{A} \models e = f$, then $\models e = f$. 

Proof sketch.
We show that $\mathfrak{A} \models e = f$ implies $P(\Sigma^\ast) \models e = f$. For $n \in \mathbb{N}$, choose $\Sigma_n = \{w \in \Sigma^\ast : |w| \leq n\}$, $h_n : \Sigma \to \mathbb{R}(\Sigma_n)$, $a \mapsto \{(w, wa) : wa \in \Sigma_n\}$. For all $w \in \Sigma_n$ and regular expressions $g$, we now have $w \in \hat{\ell}(f)$ iff $(\varepsilon, w) \in b\hat{h}_n(g)$. This means that $P(\Sigma^\ast) \models e = f$ if and only if $\models e = f$. Whence $P(\Sigma^\ast) \models e = f$. 

31 / 30
Lemma
If $\mathfrak{M} \models e = f$, then $\mathfrak{M} \models e = f$.

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

$$
\Sigma_n = \{ w \in \Sigma^* : |w| \leq n \} \quad h_n : \Sigma \to \mathcal{R}(\Sigma_n), \ a \mapsto \{(w, wa) : wa \in \Sigma_n\}
$$
Bonus: extending the model theory

Lemma

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

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

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

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

Bonus: extending the model theory

Lemma

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

Proof sketch.

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

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

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

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

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

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

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

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

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

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

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

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

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

**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 \(\dagger\).
**Bonus: 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

\[ (e \parallel g) \cdot (f \parallel h) \leq (e \cdot f) \parallel (g \cdot h) \]
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 $\Sigma$;
- $\text{1}$ 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 $(I(\Sigma), \cup, \cdot, \parallel, *, \dagger, \emptyset, \{1\})$, 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 $K$. 

Bonus: 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 \( K \).
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>
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 ´Esik 1996)
   - see also (Lodaya and Weil 2000; van Heerdt et al. 2021)

3. Translate bimonoids to concurrent KAs.
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)
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


