Kleene Algebra

Tobias Kappé

University College London

Séminaire de l'équipe Méthodes Formelles — May 21, 2019

Joint work with. . .

UCL

Marcello Bonsangue

Paul Brunet

Fredrik Dahlqvist

Nate Foster

Justin Hsu

Dexter Kozen

Bas Luttik

Jurriaan Rot

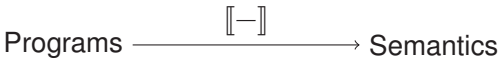
Alexandra Silva

Steffen Smolka

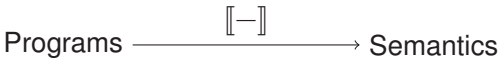
Jana Wagemaker

Fabio Zanasi

Equational theories for programs



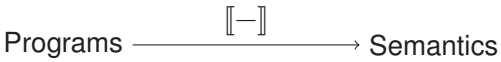
Equational theories for programs



Question

Can we characterise which programs are semantically equivalent?

Equational theories for programs



Question

Can we characterise which programs are semantically equivalent?

Question

Can we decide whether two programs have the same semantics?

Equational theories for programs

Rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e^*$

Equational theories for programs

Rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e^*$

Nondeterministic composition

Equational theories for programs

Rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e^*$

Sequential composition

Equational theories for programs

Rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e^*$

Nondeterministic repetition

Equational theories for programs

Rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e^*$

$$[\![0]\!]_{\mathsf{R}} = \emptyset$$

Equational theories for programs

Rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e^*$

$[\![0]\!]_{\mathsf{R}} = \emptyset$ $\qquad$ $[\![1]\!]_{\mathsf{R}} = \{\varepsilon\}$

Equational theories for programs

Rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e^*$

$[\![0]\!]_{\mathsf{R}} = \emptyset$ $[\![1]\!]_{\mathsf{R}} = \{\varepsilon\}$ $[\![a]\!]_{\mathsf{R}} = \{a\}$

Equational theories for programs

Rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e^*$

$$[\![0]\!]_{\mathsf{R}} = \emptyset \qquad [\![1]\!]_{\mathsf{R}} = \{\varepsilon\} \qquad [\![a]\!]_{\mathsf{R}} = \{a\} \qquad [\![e + f]\!]_{\mathsf{R}} = [\![e]\!]_{\mathsf{R}} \cup [\![f]\!]_{\mathsf{R}}$$

Equational theories for programs

Rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e^*$

$$[\![0]\!]_{\mathrm{R}} = \emptyset \qquad [\![1]\!]_{\mathrm{R}} = \{\varepsilon\} \qquad [\![a]\!]_{\mathrm{R}} = \{a\} \qquad [\![e + f]\!]_{\mathrm{R}} = [\![e]\!]_{\mathrm{R}} \cup [\![f]\!]_{\mathrm{R}}$$

$$[\![e \cdot f]\!]_{\mathrm{R}} = \left\{ wx : w \in [\![e]\!]_{\mathrm{R}}, x \in [\![f]\!]_{\mathrm{R}} \right\}$$

Equational theories for programs

Rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e^*$

$$[\![0]\!]_{\mathrm{R}} = \emptyset \qquad [\![1]\!]_{\mathrm{R}} = \{\varepsilon\} \qquad [\![a]\!]_{\mathrm{R}} = \{a\} \qquad [\![e + f]\!]_{\mathrm{R}} = [\![e]\!]_{\mathrm{R}} \cup [\![f]\!]_{\mathrm{R}}$$

$$[\![e \cdot f]\!]_{\mathrm{R}} = \left\{ wx : w \in [\![e]\!]_{\mathrm{R}}, x \in [\![f]\!]_{\mathrm{R}} \right\} \qquad [\![e^*]\!]_{\mathrm{R}} = \left\{ w_1 \cdots w_n : w_1, \ldots, w_n \in [\![e]\!]_{\mathrm{R}} \right\}$$

Equational theories for programs

$$e + e \equiv_{\mathsf{KA}} e$$

Equational theories for programs

$$e + e \equiv_{\mathsf{KA}} e \qquad\qquad e \cdot (f \cdot g) \equiv_{\mathsf{KA}} (e \cdot f) \cdot g$$

Equational theories for programs

$$e + e \equiv_{\mathsf{KA}} e \qquad e \cdot (f \cdot g) \equiv_{\mathsf{KA}} (e \cdot f) \cdot g$$

$$1 + e \cdot e^* \equiv_{\mathsf{KA}} e^*$$

Equational theories for programs

$$e + e \equiv_{\mathsf{KA}} e \qquad e \cdot (f \cdot g) \equiv_{\mathsf{KA}} (e \cdot f) \cdot g$$

$$1 + e \cdot e^* \equiv_{\mathsf{KA}} e^* \qquad \frac{e \cdot f + g \leqq_{\mathsf{KA}} f}{e^* \cdot g \leqq_{\mathsf{KA}} f} \qquad \ldots$$

Equational theories for programs

$$e + e \equiv_{\mathsf{KA}} e \qquad e \cdot (f \cdot g) \equiv_{\mathsf{KA}} (e \cdot f) \cdot g$$

$$1 + e \cdot e^* \equiv_{\mathsf{KA}} e^* \qquad \frac{e \cdot f + g \leq_{\mathsf{KA}} f}{e^* \cdot g \leq_{\mathsf{KA}} f} \qquad \ldots$$

Theorem (Kozen 1994; Salomaa 1966; Boffa 1990; Krob 1990)

For all rational expressions e and f, we have $e \equiv_{\mathsf{KA}} f$ if and only if $[\![e]\!]_{\mathsf{R}} = [\![f]\!]_{\mathsf{R}}$.

Equational theories for programs

Theorem (Kleene 1956; Brzozowski 1964; Thompson 1968; Antimirov 1996)

For every rational expression e, there exists a finite automaton A such that $L(A) = [\![e]\!]_{\mathrm{R}}$.

Equational theories for programs

UCL

Theorem (Kleene 1956; Brzozowski 1964; Thompson 1968; Antimirov 1996)

For every rational expression e, there exists a finite automaton A such that $L(A) = [\![e]\!]_{\mathrm{R}}$.

Theorem (Hopcroft and Karp 1971; Bonchi and Pous 2013)

Language equivalence of finite automata is decidable (by checking bisimilarity).

Equational theories for programs

UCL

Theorem (Kleene 1956; Brzozowski 1964; Thompson 1968; Antimirov 1996)

For every rational expression e, there exists a finite automaton A such that $L(A) = [\![e]\!]_{\mathrm{R}}$.

Theorem (Hopcroft and Karp 1971; Bonchi and Pous 2013)

Language equivalence of finite automata is decidable (by checking bisimilarity).

Corollary

Given rational expressions e and f, we can decide whether $[\![e]\!]_{\mathrm{R}} = [\![f]\!]_{\mathrm{R}}$.

Equational theories for programs

NetKAT programs: $e, f ::= \texttt{drop} \mid \texttt{skip} \mid \texttt{dup} \mid \ell = v \mid \ell \leftarrow v \mid e + f \mid e \cdot f \mid e^*$

Equational theories for programs

NetKAT programs: $e, f ::= \texttt{drop} \mid \texttt{skip} \mid \texttt{dup} \mid \ell = v \mid \ell \leftarrow v \mid e + f \mid e \cdot f \mid e^*$

Denotational semantics: $[\![-]\!]_F$: Programs $\to$ Network Policies.

$$\ell \leftarrow v \cdot \ell = v \equiv_{PA} \ell \leftarrow v \qquad \ell \leftarrow v \cdot \ell' \leftarrow v' \equiv_{PA} \ell' \leftarrow v' \cdot \ell \leftarrow v \qquad \ldots$$

Write persistence

Equational theories for programs

NetKAT programs: $e, f ::= \texttt{drop} \mid \texttt{skip} \mid \texttt{dup} \mid \ell = v \mid \ell \leftarrow v \mid e + f \mid e \cdot f \mid e^*$

Denotational semantics: $[\![-]\!]_F$: Programs $\to$ Network Policies.

$$\ell \leftarrow v \cdot \ell = v \equiv_{PA} \ell \leftarrow v \qquad \ell \leftarrow v \cdot \ell' \leftarrow v' \equiv_{PA} \ell' \leftarrow v' \cdot \ell \leftarrow v \qquad \ldots$$

Write independence

Equational theories for programs

NetKAT programs: $e, f ::= \texttt{drop} \mid \texttt{skip} \mid \texttt{dup} \mid \ell = v \mid \ell \leftarrow v \mid e + f \mid e \cdot f \mid e^*$

Denotational semantics: $[\![-]\!]_{\mathsf{F}}$: Programs $\to$ Network Policies.

$$\ell \leftarrow v \cdot \ell = v \equiv_{\mathsf{PA}} \ell \leftarrow v \qquad \ell \leftarrow v \cdot \ell' \leftarrow v' \equiv_{\mathsf{PA}} \ell' \leftarrow v' \cdot \ell \leftarrow v \qquad \ldots$$

Theorem (Anderson et al. 2014)

For all NetKAT programs, $e \equiv_{\mathsf{PA}} f$ if and only if $[\![e]\!]_{\mathsf{F}} = [\![f]\!]_{\mathsf{F}}$.

Equational theories for programs

NetKAT programs: $e, f ::= \texttt{drop} \mid \texttt{skip} \mid \texttt{dup} \mid \ell = v \mid \ell \leftarrow v \mid e + f \mid e \cdot f \mid e^*$

Denotational semantics: $[\![-]\!]_{\mathsf{F}}$: Programs $\to$ Network Policies.

$$\ell \leftarrow v \cdot \ell = v \equiv_{\mathsf{PA}} \ell \leftarrow v \qquad \ell \leftarrow v \cdot \ell' \leftarrow v' \equiv_{\mathsf{PA}} \ell' \leftarrow v' \cdot \ell \leftarrow v \qquad \ldots$$

Theorem (Anderson et al. 2014)

For all NetKAT programs, $e \equiv_{\mathsf{PA}} f$ if and only if $[\![e]\!]_{\mathsf{F}} = [\![f]\!]_{\mathsf{F}}$.

Using $\equiv_{\mathsf{PA}}$, one can "compile" a NetKAT program into a format interpretable by network hardware [Smolka et al. 2015].

Equational theories for programs

NetKAT programs: $e, f ::= \texttt{drop} \mid \texttt{skip} \mid \texttt{dup} \mid \ell = v \mid \ell \leftarrow v \mid e + f \mid e \cdot f \mid e^*$

Denotational semantics: $[\![-]\!]_{\mathsf{F}}$: Programs $\to$ Network Policies.

$$\ell \leftarrow v \cdot \ell = v \equiv_{\mathsf{PA}} \ell \leftarrow v \qquad \ell \leftarrow v \cdot \ell' \leftarrow v' \equiv_{\mathsf{PA}} \ell' \leftarrow v' \cdot \ell \leftarrow v \qquad \ldots$$

Theorem (Anderson et al. 2014; Foster et al. 2015)

Given two NetKAT programs, it is decidable whether $[\![e]\!]_{\mathsf{F}} = [\![f]\!]_{\mathsf{F}}$.

Equational theories for programs

NetKAT programs: $e, f ::= \texttt{drop} \mid \texttt{skip} \mid \texttt{dup} \mid \ell = v \mid \ell \leftarrow v \mid e + f \mid e \cdot f \mid e^*$

Denotational semantics: $[\![-]\!]_{\mathsf{F}}$: Programs $\to$ Network Policies.

$$\ell \leftarrow v \cdot \ell = v \equiv_{\mathsf{PA}} \ell \leftarrow v \qquad \ell \leftarrow v \cdot \ell' \leftarrow v' \equiv_{\mathsf{PA}} \ell' \leftarrow v' \cdot \ell \leftarrow v \qquad \ldots$$

Theorem (Anderson et al. 2014; Foster et al. 2015)

Given two NetKAT programs, it is decidable whether $[\![e]\!]_{\mathsf{F}} = [\![f]\!]_{\mathsf{F}}$.

$$(\mathsf{sw} = A) \cdot p \cdot (\mathsf{sw} = B) \equiv_{\mathsf{PA}} \texttt{drop} \iff \text{switch } A \textit{ cannot} \text{ reach switch } B$$

Equational theories for programs

Parallel composition $\Rightarrow$ *concurrent* Kleene algebra

Lock-step behaviour $\Rightarrow$ *synchronous* Kleene algebra

Conditional flow $\Rightarrow$ *guarded* Kleene algebra

Concurrent Kleene Algebra

Parallel composition $\Rightarrow$ ***concurrent*** **Kleene algebra**

Lock-step behaviour $\Rightarrow$ *synchronous* Kleene algebra

Conditional flow $\Rightarrow$ *guarded* Kleene algebra

Concurrent Kleene Algebra

Series-rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e \parallel f \mid e^*$

Concurrent Kleene Algebra

Series-rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e \parallel f \mid e^*$

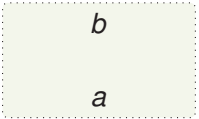
$$a \cdot b \quad \approx \quad a \longrightarrow b$$

Concurrent Kleene Algebra

Series-rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e \parallel f \mid e^*$

$a \cdot b \quad \approx \quad a \longrightarrow b$

$a \parallel b \quad \approx$

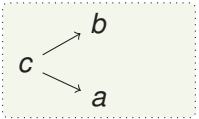


Concurrent Kleene Algebra

Series-rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e \parallel f \mid e^*$

$a \cdot b \quad \approx \quad a \longrightarrow b$

$c \cdot (a \parallel b) \quad \approx$

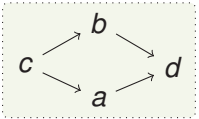


Concurrent Kleene Algebra

Series-rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e \parallel f \mid e^*$

$a \cdot b \quad \approx \quad a \longrightarrow b$

$c \cdot (a \parallel b) \cdot d \quad \approx$

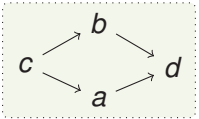


Concurrent Kleene Algebra

Series-rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e \parallel f \mid e^*$

$a \cdot b \quad \approx \quad a \longrightarrow b$

$c \cdot (a \parallel b) \cdot d \quad \approx$



$[\![0]\!]_{\mathrm{SR}} = \emptyset \qquad [\![1]\!]_{\mathrm{SR}} = \{\varepsilon\} \qquad [\![a]\!]_{\mathrm{SR}} = \{a\} \qquad [\![e + f]\!]_{\mathrm{SR}} = [\![e]\!]_{\mathrm{SR}} \cup [\![f]\!]_{\mathrm{SR}}$

$[\![e \cdot f]\!]_{\mathrm{SR}} = \{wx : w \in [\![e]\!]_{\mathrm{SR}}, x \in [\![f]\!]_{\mathrm{SR}}\} \qquad [\![e \parallel f]\!]_{\mathrm{SR}} = \{w \parallel x : w \in [\![e]\!]_{\mathrm{SR}}, x \in [\![f]\!]_{\mathrm{SR}}\}$

$$[\![e^*]\!]_{\mathrm{SR}} = \{w_1 \cdots w_n : w_1, \ldots, w_n \in [\![e]\!]_{\mathrm{SR}}\}$$

Concurrent Kleene Algebra

$$e \parallel 0 \equiv_{\mathsf{BKA}} 0 \qquad e \parallel 1 \equiv_{\mathsf{BKA}} e \qquad e \parallel f \equiv_{\mathsf{BKA}} f \parallel e$$

$$e \parallel (f \parallel g) \equiv_{\mathsf{BKA}} (e \parallel f) \parallel g \qquad e \parallel (f + g) \equiv_{\mathsf{BKA}} e \parallel f + e \parallel g$$

Concurrent Kleene Algebra

$$e \parallel 0 \equiv_{\mathsf{BKA}} 0 \qquad e \parallel 1 \equiv_{\mathsf{BKA}} e \qquad e \parallel f \equiv_{\mathsf{BKA}} f \parallel e$$

$$e \parallel (f \parallel g) \equiv_{\mathsf{BKA}} (e \parallel f) \parallel g \qquad e \parallel (f + g) \equiv_{\mathsf{BKA}} e \parallel f + e \parallel g$$

Theorem (Laurence and Struth 2014)

For sr-expressions e and f, we have $e \equiv_{\mathsf{BKA}} f$ if and only if $[\![e]\!]_{\mathsf{SR}} = [\![f]\!]_{\mathsf{SR}}$.

Concurrent Kleene Algebra

$$e \parallel 0 \equiv_{\mathsf{BKA}} 0 \qquad e \parallel 1 \equiv_{\mathsf{BKA}} e \qquad e \parallel f \equiv_{\mathsf{BKA}} f \parallel e$$

$$e \parallel (f \parallel g) \equiv_{\mathsf{BKA}} (e \parallel f) \parallel g \qquad e \parallel (f + g) \equiv_{\mathsf{BKA}} e \parallel f + e \parallel g$$

Theorem (Laurence and Struth 2014)

For sr-expressions e and f, we have $e \equiv_{\mathsf{BKA}} f$ if and only if $[\![e]\!]_{\mathsf{SR}} = [\![f]\!]_{\mathsf{SR}}$.

Theorem (Laurence and Struth 2014; Brunet et al. 2017; K. et al. 2018b)

Given sr-expressions e and f, it is decidable whether $[\![e]\!]_{\mathsf{SR}} = [\![f]\!]_{\mathsf{SR}}$.

Concurrent Kleene Algebra

Concurrent Kleene Algebra

Concurrent Kleene Algebra

$$[\![e]\!]_{\mathrm{SR}}{\downarrow} = \left\{ U : \exists V \in [\![e]\!]_{\mathrm{SR}}.\ U \sqsubseteq V \right\}$$

Concurrent Kleene Algebra

$$[\![e]\!]_{\mathrm{SR}}{\downarrow} = \{U : \exists V \in [\![e]\!]_{\mathrm{SR}}.\ U \sqsubseteq V\}$$

Theorem (Brunet et al. 2017)

Given sr-expressions e and f, it is decidable whether $[\![e]\!]_{\mathrm{SR}}{\downarrow} = [\![f]\!]_{\mathrm{SR}}{\downarrow}$.

Concurrent Kleene Algebra

$$(e \parallel f) \cdot (g \parallel h) \leq_{\mathsf{CKA}} (e \cdot g) \parallel (f \cdot h)$$

Concurrent Kleene Algebra

$$(e \parallel f) \cdot (g \parallel h) \leq_{\mathsf{CKA}} (e \cdot g) \parallel (f \cdot h)$$

Theorem (K. et al. 2018a; Laurence and Struth 2017)

For an sr-expression e, we can construct an spr-expression $e{\downarrow}$, such that:

$$e \equiv_{\mathsf{CKA}} e{\downarrow} \qquad\qquad [\![e]\!]_{\mathsf{SR}}{\downarrow} = [\![e{\downarrow}]\!]_{\mathsf{SR}}$$

Concurrent Kleene Algebra

$$(e \parallel f) \cdot (g \parallel h) \leq_{\mathsf{CKA}} (e \cdot g) \parallel (f \cdot h)$$

Theorem (K. et al. 2018a; Laurence and Struth 2017)

For an sr-expression e, we can construct an spr-expression $e{\downarrow}$, such that:

$$e \equiv_{\mathsf{CKA}} e{\downarrow} \qquad\qquad [\![e]\!]_{\mathsf{SR}}{\downarrow} = [\![e{\downarrow}]\!]_{\mathsf{SR}}$$

Corollary

For sr-expressions e and f, we have $e \equiv_{\mathsf{CKA}} f$ if and only if $[\![e]\!]_{\mathsf{SR}}{\downarrow} = [\![f]\!]_{\mathsf{SR}}{\downarrow}$.

Synchronous Kleene Algebra

Parallel composition $\Rightarrow$ *concurrent* Kleene algebra

Lock-step behaviour $\Rightarrow$ *synchronous* Kleene algebra

Conditional flow $\Rightarrow$ *guarded* Kleene algebra

Synchronous Kleene Algebra

Synchronous rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e \times f \mid e^*$

Synchronous Kleene Algebra

Synchronous rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e \times f \mid e^*$

$$[\![0]\!]_{\text{Syn}} = \emptyset \qquad [\![1]\!]_{\text{Syn}} = \{\varepsilon\} \qquad [\![a]\!]_{\text{Syn}} = \{\{a\}\}$$

$$[\![e + f]\!]_{\text{Syn}} = [\![e]\!]_{\text{Syn}} \cup [\![f]\!]_{\text{Syn}} \qquad [\![e \cdot f]\!]_{\text{Syn}} = \{wx : w \in [\![e]\!]_{\text{Syn}}, x \in [\![f]\!]_{\text{Syn}}\}$$

$$[\![e \times f]\!]_{\text{Syn}} = \{w \times x : w \in [\![e]\!]_{\text{Syn}}, x \in [\![f]\!]_{\text{Syn}}\} \qquad [\![e^*]\!]_{\text{Syn}} = [\![e]\!]^*_{\text{Syn}}$$

Synchronous Kleene Algebra

Synchronous rational expressions: $e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e \times f \mid e^*$

$$[\![0]\!]_{\text{Syn}} = \emptyset \qquad [\![1]\!]_{\text{Syn}} = \{\varepsilon\} \qquad [\![a]\!]_{\text{Syn}} = \{\{a\}\}$$

$$[\![e + f]\!]_{\text{Syn}} = [\![e]\!]_{\text{Syn}} \cup [\![f]\!]_{\text{Syn}} \qquad [\![e \cdot f]\!]_{\text{Syn}} = \{wx : w \in [\![e]\!]_{\text{Syn}}, x \in [\![f]\!]_{\text{Syn}}\}$$

$$[\![e \times f]\!]_{\text{Syn}} = \{w \times x : w \in [\![e]\!]_{\text{Syn}}, x \in [\![f]\!]_{\text{Syn}}\} \qquad [\![e^*]\!]_{\text{Syn}} = [\![e]\!]^*_{\text{Syn}}$$

Let $w, x \in \mathcal{P}(\Sigma)^*$ and $A, B \subseteq \Sigma$; then

$$w \times \varepsilon = \varepsilon \times w = w \qquad Aw \times Bx = (A \cup B)(w \times x)$$

Synchronous Kleene Algebra

$$e \times f \equiv_{\mathsf{SKA}} f \times e \qquad e \times (f \times g) \equiv_{\mathsf{SKA}} (e \times f) \times g \qquad e \times (f + g) \equiv_{\mathsf{SKA}} e \times f + e \times g$$

$$e \times 0 \equiv_{\mathsf{SKA}} 0 \qquad e \times 1 \equiv_{\mathsf{SKA}} e \qquad (a \cdot e) \times (b \cdot f) \equiv_{\mathsf{SKA}} (a \times b) \cdot (e \times f) \qquad (a, b \in \Sigma)$$

Synchronous Kleene Algebra

$$e \times f \equiv_{\mathsf{SKA}} f \times e \qquad e \times (f \times g) \equiv_{\mathsf{SKA}} (e \times f) \times g \qquad e \times (f + g) \equiv_{\mathsf{SKA}} e \times f + e \times g$$

$$e \times 0 \equiv_{\mathsf{SKA}} 0 \qquad e \times 1 \equiv_{\mathsf{SKA}} e \qquad (a \cdot e) \times (b \cdot f) \equiv_{\mathsf{SKA}} (a \times b) \cdot (e \times f) \qquad (a, b \in \Sigma)$$

Claim (Prisacariu 2010)

For syn. rational expressions e and f, we have $e \equiv_{\mathsf{SKA}} f$ if and only if $[\![e]\!]_{\mathsf{Syn}} = [\![f]\!]_{\mathsf{Syn}}$.

Synchronous Kleene Algebra

$e \times f \equiv_{\mathsf{SKA}} f \times e$ $\quad e \times (f \times g) \equiv_{\mathsf{SKA}} (e \times f) \times g$ $\quad e \times (f + g) \equiv_{\mathsf{SKA}} e \times f + e \times g$

$e \times 0 \equiv_{\mathsf{SKA}} 0$ $\quad e \times 1 \equiv_{\mathsf{SKA}} e$ $\quad (a \cdot e) \times (b \cdot f) \equiv_{\mathsf{SKA}} (a \times b) \cdot (e \times f)$ $\quad (a, b \in \Sigma)$

Claim (Prisacariu 2010)

For syn. rational expressions e and f, we have $e \equiv_{\mathsf{SKA}} f$ if and only if $[\![e]\!]_{\mathsf{Syn}} = [\![f]\!]_{\mathsf{Syn}}$.

Counterexample (Wagemaker et al. 2019)

Let $a \in \Sigma$; now $[\![a^* \times a^*]\!]_{\mathsf{Syn}} = [\![a^*]\!]_{\mathsf{Syn}}$, but $a^* \times a^* \not\equiv_{\mathsf{SKA}} a^*$.

Synchronous Kleene Algebra

$$\frac{e \cdot f + g \leq_{\mathsf{KA}} f}{e^* \cdot g \leq_{\mathsf{KA}} f} \qquad \frac{e \cdot f + g \leq_{\mathsf{KA}} e}{g \cdot f^* \leq_{\mathsf{KA}} e}$$

Synchronous Kleene Algebra

$$\frac{E(e) = 0 \qquad e \cdot f + g \equiv_{\mathrm{SF}_1} f}{e^* \cdot g \equiv_{\mathrm{SF}_1} f}$$

$$(e + 1)^* \equiv_{\mathrm{SF}_1} e^*$$

Synchronous Kleene Algebra

$$\frac{E(e) = 0 \qquad e \cdot f + g \equiv_{\mathrm{SF}_1} f}{e^* \cdot g \equiv_{\mathrm{SF}_1} f} \qquad (e + 1)^* \equiv_{\mathrm{SF}_1} e^*$$

Theorem (Wagemaker et al. 2019)

For syn. rational expressions e and f, we have $e \equiv_{\mathrm{SF}_1} f$ if and only if $[\![e]\!]_{\mathrm{Syn}} = [\![f]\!]_{\mathrm{Syn}}$.

Synchronous Kleene Algebra

$$\frac{E(e) = 0 \qquad e \cdot f + g \equiv_{\mathrm{SF}_1} f}{e^* \cdot g \equiv_{\mathrm{SF}_1} f} \qquad (e + 1)^* \equiv_{\mathrm{SF}_1} e^*$$

Theorem (Wagemaker et al. 2019)

For syn. rational expressions e and f, we have $e \equiv_{\mathrm{SF}_1} f$ if and only if $[\![e]\!]_{\mathrm{Syn}} = [\![f]\!]_{\mathrm{Syn}}$.

Theorem (Broda et al. 2015)

For syn. rational expressions e and f, it is decidable whether $[\![e]\!]_{\mathrm{Syn}} = [\![f]\!]_{\mathrm{Syn}}$.

Guarded Kleene Algebra

Parallel composition $\Rightarrow$ *concurrent* Kleene algebra

Lock-step behaviour $\Rightarrow$ *synchronous* Kleene algebra

Conditional flow $\Rightarrow$ *guarded* Kleene algebra

Guarded Kleene Algebra

Guarded rational expressions: $e, f ::= b \in \mathcal{B} \mid a \in \Sigma \mid e \cdot f \mid e +_b f \mid e^b$

Guarded Kleene Algebra

Guarded rational expressions: $e, f ::= b \in \mathcal{B} \mid a \in \Sigma \mid e \cdot f \mid e +_b f \mid e^b$

$$[\![b]\!]_{\mathsf{GR}} = \{\alpha \in \mathsf{At} : \alpha \leqslant b\}$$

Guarded Kleene Algebra

Guarded rational expressions: $e, f ::= b \in \mathcal{B} \mid a \in \Sigma \mid e \cdot f \mid e +_b f \mid e^b$

$$[\![b]\!]_{\mathrm{GR}} = \{\alpha \in \mathrm{At} : \alpha \leqslant b\} \qquad [\![a]\!]_{\mathrm{GR}} = \{\alpha a \beta : \alpha, \beta \in \mathrm{At}\}$$

Guarded Kleene Algebra

Guarded rational expressions: $e, f ::= b \in \mathcal{B} \mid a \in \Sigma \mid e \cdot f \mid e +_b f \mid e^b$

$$[\![b]\!]_{\mathrm{GR}} = \{\alpha \in \mathrm{At} : \alpha \leqslant b\} \qquad [\![a]\!]_{\mathrm{GR}} = \{\alpha a \beta : \alpha, \beta \in \mathrm{At}\}$$

$$[\![e \cdot f]\!]_{\mathrm{GR}} = \{w\alpha x : w\alpha \in [\![e]\!]_{\mathrm{GR}}, \alpha x \in [\![f]\!]_{\mathrm{GR}}\}$$

Guarded Kleene Algebra

Guarded rational expressions: $e, f ::= b \in \mathcal{B} \mid a \in \Sigma \mid e \cdot f \mid e +_b f \mid e^b$

$$[\![b]\!]_{\mathrm{GR}} = \{\alpha \in \mathrm{At} : \alpha \leqslant b\} \qquad [\![a]\!]_{\mathrm{GR}} = \{\alpha a \beta : \alpha, \beta \in \mathrm{At}\}$$

$$[\![e \cdot f]\!]_{\mathrm{GR}} = \{w\alpha x : w\alpha \in [\![e]\!]_{\mathrm{GR}}, \alpha x \in [\![f]\!]_{\mathrm{GR}}\}$$

$$[\![e +_b f]\!]_{\mathrm{GR}} = \{\alpha w \in [\![e]\!]_{\mathrm{GR}} : \alpha \leqslant b\} \cup \{\alpha x \in [\![f]\!]_{\mathrm{GR}} : \alpha \leqslant \overline{b}\}$$

Guarded Kleene Algebra

Guarded rational expressions: $e, f ::= b \in \mathcal{B} \mid a \in \Sigma \mid e \cdot f \mid e +_b f \mid e^b$

$$[\![b]\!]_{\mathrm{GR}} = \{\alpha \in \mathrm{At} : \alpha \leqslant b\} \qquad [\![a]\!]_{\mathrm{GR}} = \{\alpha a \beta : \alpha, \beta \in \mathrm{At}\}$$

$$[\![e \cdot f]\!]_{\mathrm{GR}} = \{w\alpha x : w\alpha \in [\![e]\!]_{\mathrm{GR}}, \alpha x \in [\![f]\!]_{\mathrm{GR}}\}$$

$$[\![e +_b f]\!]_{\mathrm{GR}} = \{\alpha w \in [\![e]\!]_{\mathrm{GR}} : \alpha \leqslant b\} \cup \{\alpha x \in [\![f]\!]_{\mathrm{GR}} : \alpha \leqslant \overline{b}\}$$

$$[\![e^b]\!]_{\mathrm{GR}} = \{\alpha_1 w_1 \alpha_2 w_2 \cdots \alpha_n w_n \alpha_{n+1} : \alpha_i w_i \alpha_{i+1} \in [\![e]\!], \alpha_1, \ldots, \alpha_n \leqslant b, \alpha_{n+1} \leqslant \overline{b}\}$$

Guarded Kleene Algebra

Guarded rational expressions: $e, f ::= b \in \mathcal{B} \mid a \in \Sigma \mid e \cdot f \mid e +_b f \mid e^b$

$$[\![b]\!]_{\mathrm{GR}} = \{\alpha \in \mathrm{At} : \alpha \leqslant b\} \qquad [\![a]\!]_{\mathrm{GR}} = \{\alpha a \beta : \alpha, \beta \in \mathrm{At}\}$$

$$[\![e \cdot f]\!]_{\mathrm{GR}} = \{w\alpha x : w\alpha \in [\![e]\!]_{\mathrm{GR}}, \alpha x \in [\![f]\!]_{\mathrm{GR}}\}$$

$$[\![e +_b f]\!]_{\mathrm{GR}} = \{\alpha w \in [\![e]\!]_{\mathrm{GR}} : \alpha \leqslant b\} \cup \{\alpha x \in [\![f]\!]_{\mathrm{GR}} : \alpha \leqslant \overline{b}\}$$

$$[\![e^b]\!]_{\mathrm{GR}} = \{\alpha_1 w_1 \alpha_2 w_2 \cdots \alpha_n w_n \alpha_{n+1} : \alpha_i w_i \alpha_{i+1} \in [\![e]\!], \alpha_1, \ldots, \alpha_n \leqslant b, \alpha_{n+1} \leqslant \overline{b}\}$$

Lemma

For guarded rational expressions e and f, it is decidable whether $[\![e]\!]_{\mathrm{GR}} = [\![f]\!]_{\mathrm{GR}}$.

Guarded Kleene Algebra

$$0 \cdot e \equiv_{\mathsf{GKA}} 0 \equiv_{\mathsf{GKA}} e \cdot 0 \qquad 1 \cdot e \equiv_{\mathsf{GKA}} e \equiv_{\mathsf{GKA}} e \cdot 1 \qquad e \cdot (f \cdot g) \equiv_{\mathsf{GKA}} (e \cdot f) \cdot g$$

$$e +_b (f +_c g) \equiv_{\mathsf{GKA}} (e +_b f) +_{b \vee c} g \qquad b \cdot c \equiv_{\mathsf{GKA}} b \wedge c \qquad e +_b 0 \equiv_{\mathsf{GKA}} b \cdot e \qquad e +_b e \equiv_{\mathsf{GKA}} e$$

$$e +_b f \equiv_{\mathsf{GKA}} f +_{\overline{b}} e \qquad (e +_b f) \cdot g \equiv_{\mathsf{GKA}} (e \cdot g) +_b (f \cdot g) \qquad e +_b f \equiv_{\mathsf{GKA}} b \cdot e +_b f$$

$$e \cdot e^b +_b 1 \equiv_{\mathsf{GKA}} e^b \qquad (e +_c 1)^b \equiv_{\mathsf{GKA}} (c \cdot e)^b \qquad \frac{E(e) = 0 \qquad f \cdot e +_b g \equiv_{\mathsf{GKA}} e}{f^b \cdot g \equiv_{\mathsf{GKA}} e}$$

Guarded Kleene Algebra

$$0 \cdot e \equiv_{\mathsf{GKA}} 0 \equiv_{\mathsf{GKA}} e \cdot 0 \qquad 1 \cdot e \equiv_{\mathsf{GKA}} e \equiv_{\mathsf{GKA}} e \cdot 1 \qquad e \cdot (f \cdot g) \equiv_{\mathsf{GKA}} (e \cdot f) \cdot g$$

$$e +_b (f +_c g) \equiv_{\mathsf{GKA}} (e +_b f) +_{b \vee c} g \qquad b \cdot c \equiv_{\mathsf{GKA}} b \wedge c \qquad e +_b 0 \equiv_{\mathsf{GKA}} b \cdot e \qquad e +_b e \equiv_{\mathsf{GKA}} e$$

$$e +_b f \equiv_{\mathsf{GKA}} f +_{\overline{b}} e \qquad (e +_b f) \cdot g \equiv_{\mathsf{GKA}} (e \cdot g) +_b (f \cdot g) \qquad e +_b f \equiv_{\mathsf{GKA}} b \cdot e +_b f$$

$$e \cdot e^b +_b 1 \equiv_{\mathsf{GKA}} e^b \qquad (e +_c 1)^b \equiv_{\mathsf{GKA}} (c \cdot e)^b \qquad \frac{E(e) = 0 \qquad f \cdot e +_b g \equiv_{\mathsf{GKA}} e}{f^b \cdot g \equiv_{\mathsf{GKA}} e}$$

Conjecture

For guarded rational expressions e and f, we have $e \equiv_{\mathsf{GKA}} f$ if and only if $[\![e]\!]_{\mathsf{GR}} = [\![f]\!]_{\mathsf{GR}}$.

Conclusion

UCL

1. Kleene algebra is a powerful tool for reasoning about program equivalence.

Conclusion

UCL

1. Kleene algebra is a powerful tool for reasoning about program equivalence.
2. Classical results for Kleene algebra extend to new operators.

Conclusion

1. Kleene algebra is a powerful tool for reasoning about program equivalence.
2. Classical results for Kleene algebra extend to new operators.
3. When done right, reductions for completeness and decidability coincide.

Conclusion

1. Kleene algebra is a powerful tool for reasoning about program equivalence.
2. Classical results for Kleene algebra extend to new operators.
3. When done right, reductions for completeness and decidability coincide.
4. Hardness of these proofs warrants this modular approach.

Thank you for your attention

https://coneco-project.org

For slides, see https://tobias.kap.pe

References I

- C. J. Anderson et al. [2014]. *NetKAT: semantic foundations for networks*. DOI: 10.1145/2535838.2535862.
- V. M. Antimirov [1996]. *Partial Derivatives of Regular Expressions and Finite Automaton Constructions*. DOI: 10.1016/0304-3975(95)00182-4.
- M. Boffa [1990]. “Une remarque sur les systèmes complets d’identités rationnelles”. In: *ITA* 24, pp. 419–428.
- F. Bonchi and D. Pous [2013]. *Checking NFA equivalence with bisimulations up to congruence*. DOI: 10.1145/2429069.2429124.
- S. Broda et al. [2015]. *Deciding Synchronous Kleene Algebra with Derivatives*. DOI: 10.1007/978-3-319-22360-5_5.
- P. Brunet et al. [2017]. *On Decidability of Concurrent Kleene Algebra*. DOI: 10.4230/LIPIcs.CONCUR.2017.28.
- J. A. Brzozowski [1964]. *Derivatives of Regular Expressions*. DOI: 10.1145/321239.321249.

References II

- N. Foster et al. [2015]. *A Coalgebraic Decision Procedure for NetKAT*. DOI: 10.1145/2676726.2677011.
- J. E. Hopcroft and R. M. Karp [1971]. *A linear algorithm for testing equivalence of finite automata*. URL: https://ecommons.cornell.edu/handle/1813/5958.
- T. Kappé et al. [2018a]. *Concurrent Kleene Algebra: Free Model and Completeness*. DOI: 10.1007/978-3-319-89884-1_30.
- T. Kappé et al. [2018b]. *Equivalence checking for weak bi-Kleene algebra*. eprint: abs/1807.02102.
- S. C. Kleene [1956]. "Representation of Events in Nerve Nets and Finite Automata". In: *Automata Studies*, pp. 3–41.
- D. Kozen [1994]. *A Completeness Theorem for Kleene Algebras and the Algebra of Regular Events*. DOI: 10.1006/inco.1994.1037.
- D. Krob [1990]. *A Complete System of B-Rational Identities*. DOI: 10.1007/BFb0032022.
- M. R. Laurence and G. Struth [2014]. *Completeness Theorems for Bi-Kleene Algebras and Series-Parallel Rational Pomset Languages*. DOI: 10.1007/978-3-319-06251-8_5.

References III

- M. R. Laurence and G. Struth [2017]. *Completeness Theorems for Pomset Languages and Concurrent Kleene Algebras*. eprint: abs/1705.05896.
- C. Prisacariu [2010]. *Synchronous Kleene Algebra*. DOI: 10.1016/j.jlap.2010.07.009.
- A. Salomaa [1966]. *Two Complete Axiom Systems for the Algebra of Regular Events*. DOI: 10.1145/321312.321326.
- S. Smolka et al. [2015]. *A fast compiler for NetKAT*. DOI: 10.1145/2784731.2784761.
- K. Thompson [1968]. *Regular Expression Search Algorithm*. DOI: 10.1145/363347.363387.
- J. Wagemaker et al. [2019]. *Completeness and Incompleteness of Synchronous Kleene Algebra*.