Kleene Algebra with Dynamic Tests

Igor Sedlár

Institute of Computer Science of the Czech Academy of Sciences

LLAMA Seminar
15. 11. 2023

Motivation: Reasoning about programs

Reasoning about...

- program equivalence $\rightarrow$ shorter code
- halting / divergence $\rightarrow$ code runs
- correctness $\rightarrow$ code does what it should

Motivation: Reasoning about programs

Reasoning about...

- program equivalence $\rightarrow$ shorter code
- halting / divergence $\rightarrow$ code runs
- correctness $\rightarrow$ code does what it should

...programs...

$$E, F := \mathsf{a} \in \Sigma \mid \mathbf{skip} \mid \mathbf{abort} \mid E;F \mid \mathbf{if}\ B\ \mathbf{then}\ E\ \mathbf{else}\ F \mid \mathbf{while}\ B\ \mathbf{do}\ F$$

where B is a Boolean formula over a set of variables Π.

Motivation: Reasoning about programs

Reasoning about...

- program equivalence $\rightarrow$ shorter code
- halting / divergence $\rightarrow$ code runs
- correctness $\rightarrow$ code does what it should

...programs...

$$E, F := \mathsf{a} \in \Sigma \mid \textbf{skip} \mid \textbf{abort} \mid E; F \mid \textbf{if } B \textbf{ then } E \textbf{ else } F \mid \textbf{while } B \textbf{ do } F$$

where B is a Boolean formula over a set of variables Π.

...using "logics of programs".

Outline

1. Kleene algebra with tests (KAT)
2. KA with <u>dynamic tests</u> (aKAT) and fragments (KA with domain)
3. aKAT and PDL $\rightarrow$ relational aKAT is EXPTIME-complete
4. Language completeness for aKAT
   - $\rightarrow$ aKAT = language aKAT
5. Relational completeness for aKAT
   - $\rightarrow$ relational aKAT = language aKAT = aKAT
   - $\rightarrow$ aKAT is EXPTIME complete
6. Completeness and complexity for fragments
   - $\rightarrow$ completeness for all
   - $\rightarrow$ aKA is EXPTIME-complete

See arxiv.org/abs/2311.06937

1. Kleene algebra with tests

Kleene algebra with tests (KAT) – 1

(Cohet et al. 1996; Kozen 1996, 1997; Kozen and Smith, 1997)

Syntax (KA + tests)

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

Kleene algebra with tests (KAT) – 1

(Cohet et al. 1996; Kozen 1996, 1997; Kozen and Smith, 1997)

Syntax (KA + tests)

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

Axioms

$$(e \cdot f) \cdot g \equiv e \cdot (f \cdot g) \quad (1)$$
$$e \cdot \mathtt{1} \equiv e \equiv \mathtt{1} \cdot e \quad (2)$$
$$(e + f) + g \equiv e + (f + g) \quad (3)$$
$$\mathtt{0} + e \equiv \mathtt{0} \equiv e + \mathtt{0} \quad (4)$$
$$e \equiv e + e \quad (5)$$
$$e \cdot (f + g) \equiv (e \cdot f) + (e \cdot g) \quad (6)$$
$$(e + f) \cdot g \equiv (e \cdot g) + (f \cdot g) \quad (7)$$

↑ idempotent semirings

$$\mathtt{1} + (e \cdot e^*) \leqq e^* \quad (8)$$
$$\mathtt{1} + (e^* \cdot e) \leqq e^* \quad (9)$$
$$f + (e \cdot g) \leqq g \implies e^* \cdot f \leqq g \quad (10)$$
$$f + (g \cdot e) \leqq g \implies f \cdot e^* \leqq g \quad (11)$$

↑ Kleene algebra (KA)

$$\mathtt{p}^\perp \cdot \mathtt{p} \equiv \mathtt{0} \qquad \mathtt{p}^\perp + \mathtt{p} \equiv \mathtt{1}$$
$$b \cdot b \equiv b \qquad b \cdot c \equiv c \cdot b$$

where $e \leqq f$ means $e + f \equiv f$ and $b, c \in \Lambda = \Pi \cup \Pi^\perp$. (Kappé 2022, 2023)

Kleene algebra with tests (KAT) – 2

Encoding programs...

- Boolean formulas: DFN over Λ

  e.g. $\neg(\mathrm{p} \leftrightarrow \neg\mathrm{q})$ is $\mathrm{p} \cdot \mathrm{q} + \mathrm{p}^{\perp} \cdot \mathrm{q}^{\perp}$ if $\Pi = \{\mathrm{p}, \mathrm{q}\}$
- $E; F$ is $E \cdot F$
- **skip** as 1 and **abort** as 0
- **if** B **then** E **else** F is $(B \cdot E) + (\neg B \cdot F)$
- **while** B **do** E is $(B \cdot E)^* \cdot (\neg B)$

...and their properties

- equivalence: $e \equiv f$
- divergence $e \equiv 0$, halting $e \not\equiv 0$
- correctness $(B \cdot e) \cdot (\neg C) \equiv 0$

Kleene algebra with tests (KAT) – 2

Encoding programs...

- Boolean formulas: DFN over Λ

e.g. $\neg(\mathrm{p} \leftrightarrow \neg\mathrm{q})$ is $\mathrm{p} \cdot \mathrm{q} + \mathrm{p}^{\perp} \cdot \mathrm{q}^{\perp}$ if $\Pi = \{\mathrm{p}, \mathrm{q}\}$

- $E; F$ is $E \cdot F$
- **skip** as 1 and **abort** as 0
- **if** B **then** E **else** F is $(B \cdot E) + (\neg B \cdot F)$
- **while** B **do** E is $(B \cdot E)^* \cdot (\neg B)$

...and their properties

- equivalence: $e \equiv f$
- divergence $e \equiv 0$, halting $e \not\equiv 0$
- correctness $(B \cdot e) \cdot (\neg C) \equiv 0$

Properties of KAT:

- Sound and complete for relational models and language models based on guarded strings
- Eq. theory PSPACE-complete
- Quasi-eq. theory Σ^0_1-complete; fragment with assumptions $e \equiv 0$ reduces to eq. th.

2. Kleene algebra with dynamic tests

Kleene algebra with dynamic tests – 1

Syntax:

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

Read $e^\perp$ as "e diverges" and $e^\top$ as "e halts".

Kleene algebra with dynamic tests – 1

Syntax:

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

Read $e^\perp$ as "e diverges" and $e^\top$ as "e halts".

$\rightarrow$ $^\perp$ is antidomain and $^\top$ is domain of KA with domain (Desharnais et al. 2006, 2011)

Kleene algebra with dynamic tests – 1

Syntax:

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

Read $e^\perp$ as "e diverges" and $e^\top$ as "e halts".

$\rightarrow$ $^\perp$ is antidomain and $^\top$ is domain of KA with domain (Desharnais et al. 2006, 2011)

Axioms: KA +

$$e^\perp \cdot e \equiv \mathtt{0} \quad (12)$$

$$(e \cdot f)^\perp \equiv (e \cdot f^\top)^\perp \quad (13)$$

$$e^\perp + e^\top \equiv \mathtt{1} \quad (14)$$

$$\mathtt{p}^\top \equiv \mathtt{p} \quad (15)$$

$$e^\top \equiv e^{\perp\perp} \quad (16)$$

This is "antidomain KAT", or aKAT.

Kleene algebra with dynamic tests – 2

Let $\mathbb{E}$ be the set of all expressions.

Definition 1

The set $\mathbb{F}$ of formulas (over Σ and Π) is defined by the following grammar:

$$\phi, \psi := \mathrm{p} \in \Pi \mid 0 \mid 1 \mid \phi + \psi \mid \phi \cdot \psi \mid e^{\perp} \mid e^{\top} \,,$$

where $e \in \mathbb{E}$.

Definition 2

An expression $e \in \mathbb{E}$ is testable iff it does not contain occurrences of $^{\top}$. A test is an expression $e^{\top}$ where e is testable. A parameter is either a test or an element of Π.

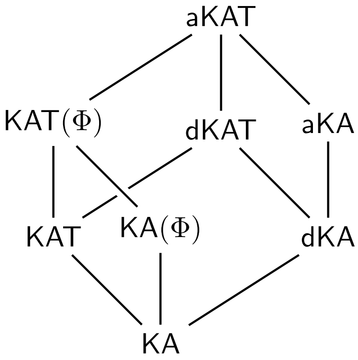
Notation: $\Phi^{\perp} = \{\phi^{\perp} \mid \phi \in \Phi\}$ and $\Phi^{\pm} = \Phi \cup \Phi^{\perp}$.

Syntactic fragments of aKAT – 1

K	Base	$\top$ applies to	$\perp$ applies to
KA	Σ	nothing	nothing
KA(Φ)	$\Sigma \cup \Phi$	nothing	nothing
dKA	Σ	everything	noting
aKA	Σ	everything	everything
KAT	$\Sigma \cup \Pi$	nothing	Π
KAT(Φ)	$\Sigma \cup \Phi$	nothing	Φ
dKAT	$\Sigma \cup \Pi$	everything	Π
aKAT	$\Sigma \cup \Pi$	everything	everything

$\Phi \subseteq \mathbb{E}$ is a set of parameters. $\mathbb{E}_{\mathsf{KA}(\Phi)}$ ($\mathbb{E}_{\mathsf{KAT}(\Phi)}$) is the set of regular expressions over Φ ($\Phi^{\pm}$); notation: $\mathbb{RE}(\Phi)$ ($\mathbb{RE}(\Phi^{\pm})$).

Syntactic fragments of aKAT – 2



We write $e \stackrel{\mathsf{K}}{\equiv} f$ if $e \equiv f$ and $e, f \in \mathbb{E}_{\mathsf{K}}$.

Relational models – 1

Definition 3

A ***relational model*** *(for Σ and Π) is a structure $M = \langle X, \mathsf{rel}_M, \mathsf{sat}_M \rangle$ where*

- *X is a set*
- $\mathsf{rel}_M : \Sigma \to 2^{X \times X}$
- $\mathsf{sat}_M : \Pi \to 2^X$.

Intuition: $\langle x, y \rangle \in \mathsf{rel}_M(\mathtt{a})$, or

$$x \xrightarrow{\mathtt{a}} y \,,$$

if action $\mathtt{a}$ may lead from state x to state y; $x \in \mathsf{sat}_M(\mathtt{p})$, or

$$(M, x) \models \mathtt{p} \,,$$

if proposition $\mathtt{p}$ is satisfied in state x.

Note: This is a Kripke frame for Σ as the “basis” of the modal signature.

Relational models – 2

Definition 4

For each relational model $M = \langle X, \mathsf{rel}_M, \mathsf{sat}_M \rangle$ *(for* Σ *and* Π*) we define the* M*-interpretation of* $\mathbb{E}$ *as the function* $[\![-]\!]_M : \mathbb{E} \to 2^{X \times X}$ *such that:*

$$[\![\mathsf{a}]\!]_M = \mathsf{rel}_M(\mathsf{a}) \qquad [\![\mathsf{p}]\!]_M = 1_{\mathsf{sat}_M(\mathsf{p})} \qquad [\![0]\!]_M = \emptyset \qquad [\![1]\!]_M = 1_X$$

$$[\![e + f]\!]_M = [\![e]\!]_M \cup [\![f]\!]_M \qquad [\![e \cdot f]\!]_M = [\![e]\!]_M \circ [\![f]\!]_M$$

$$[\![e^*]\!]_M = ([\![e]\!]_M)^* = \bigcup_{n \geq 0} [\![e]\!]_M^n$$

$$[\![e^\perp]\!]_M = \mathsf{a}\,([\![e]\!]_M) = \bigcup_{R \subseteq 1_X} (R \circ [\![e]\!]_M = \emptyset)$$

$$[\![e^\top]\!]_M = \mathsf{d}\,([\![e]\!]_M) = 1_X \setminus \mathsf{a}([\![e]\!]_M)$$

where, for $R \subseteq 2^{X \times X}$, $R^0 = 1_X$ *and* $R^{n+1} = R^n \circ R$*. Expressions* e *and* f *are relationally equivalent) iff* $[\![e]\!]_M = [\![f]\!]_M$ *for all* M*.*

Relational models – 3

Note that

$$[\![e^{\perp}]\!]_M = \{\langle x, x\rangle \mid \neg\exists y : \langle x, y\rangle \in [\![e]\!]_M\}$$

$$[\![e^{\top}]\!]_M = \{\langle x, x\rangle \mid \exists y : \langle x, y\rangle \in [\![e]\!]_M\}$$

In other words, $^{\perp}$ is dynamic negation of Dynamic Predicate Logic (Groenendijk and Stokhof, 1991) and $^{\top}$ is dynamic double negation.

Relational models – 3

Note that

$$[\![e^{\perp}]\!]_M = \{\langle x, x\rangle \mid \neg\exists y : \langle x, y\rangle \in [\![e]\!]_M\}$$

$$[\![e^{\top}]\!]_M = \{\langle x, x\rangle \mid \exists y : \langle x, y\rangle \in [\![e]\!]_M\}$$

In other words, $^{\perp}$ is dynamic negation of Dynamic Predicate Logic (Groenendijk and Stokhof, 1991) and $^{\top}$ is dynamic double negation.

If $[\![e]\!]_M$ is the start-halt relation for (program) e, then

- $[\![e^{\perp}]\!]_M$ represents the set of states where e diverges
- $[\![e^{\top}]\!]_M$ represents the set of states where e halts

3. aKAT and PDL

Satisfaction of formulas – 1

It is easy to prove by induction on the structure of ϕ that $[\![\phi]\!]_M \subseteq 1_X$ for all formulas ϕ and all M. We will write

$$(M, x) \vDash \phi \iff \langle x, x \rangle \in [\![\phi]\!]_M$$

Satisfaction of formulas – 1

It is easy to prove by induction on the structure of ϕ that $[\![\phi]\!]_M \subseteq 1_X$ for all formulas ϕ and all M. We will write

$$(M, x) \vDash \phi \iff \langle x, x \rangle \in [\![\phi]\!]_M$$

Observation 1

For all ϕ and all M:

1. $(M, x) \vDash \phi$ *iff* $\langle x, x \rangle \in [\![\phi^\top]\!]_M$
2. $(M, x) \nvDash 0$ *and* $(M, x) \vDash 1$ *for all* x
3. $(M, x) \vDash \phi^\perp$ *iff* $(M, x) \nvDash \phi$
4. $(M, x) \vDash \phi + \psi$ *iff* $(M, x) \vDash \phi$ *or* $(M, x) \vDash \psi$
5. $(M, x) \vDash \phi \cdot \psi$ *iff* $(M, x) \vDash \phi$ *and* $(M, x) \vDash \psi$
6. $(M, x) \vDash (e \cdot \phi)^{\perp\perp}$ *iff there is* y *such that* $\langle x, y \rangle \in [\![e]\!]_M$ *and* $(M, y) \vDash \phi$
7. $(M, x) \vDash (e \cdot \phi^\perp)^\perp$ *iff for all* y*,* $\langle x, y \rangle \in [\![e]\!]_M$ *implies that* $(M, y) \vDash \phi$

Satisfaction of formulas – 2

$$
\begin{aligned}
\langle x, x\rangle \in [\![\overbrace{(e \cdot \phi)^{\perp\perp}}^{\langle e\rangle \phi}]\!]_M &\iff \neg\exists y : \langle x, y\rangle \in [\![(e \cdot \phi)^{\perp}]\!]_M \\
&\iff \langle x, x\rangle \notin [\![(e \cdot \phi)^{\perp}]\!]_M \\
&\iff \exists y : \langle x, y\rangle \in [\![e \cdot \phi]\!]_M \\
&\iff \exists y : \langle x, y\rangle \in [\![e]\!]_M \ \& \ \langle y, y\rangle \in [\![\phi]\!]_M
\end{aligned}
$$

$$
\begin{aligned}
\langle x, x\rangle \in [\![\overbrace{(e \cdot \phi^{\perp})^{\perp}}^{[e] \phi}]\!]_M &\iff \neg\exists y : \langle x, y\rangle \in [\![(e \cdot \phi^{\perp})]\!]_M \\
&\iff \forall y, z : \langle x, y\rangle \in [\![e]\!]_M \implies \langle y, z\rangle \notin [\![\phi^{\perp}]\!] \\
&\iff \forall y : \langle x, y\rangle \in [\![e]\!]_M \implies \langle y, y\rangle \in [\![\phi]\!]_M
\end{aligned}
$$

aKAT = “PDL in disguise”

We obtain the usual semantics of (programs and formulas) of PDL within aKAT. PDL programs form a specific fragment $\mathbb{E}_{\mathsf{PDL}}$ of $\mathbb{E}$. aKAT is “one sorted test algebra”.

aKAT = "PDL in disguise"

We obtain the usual semantics of (programs and formulas) of PDL within aKAT. PDL programs form a specific fragment $\mathbb{E}_{\mathsf{PDL}}$ of $\mathbb{E}$. aKAT is "one sorted test algebra".

Theorem 1

The problem of deciding relational equivalence between arbitrary expressions in $\mathbb{E}$ is $\mathrm{EXPTIME}$-complete.

Proof. Lower bound: the ($\mathrm{EXPTIME}$-hard) membership problem for polynomial-space alternating Turing machines reduces to the problem of satisfiability of PDL-formulas in relational models: For each machine A and input t, there is a PDL-formula $\mathsf{F}_{A,t}$ such that A accepts t iff $\mathsf{F}_{A,t}$ is satisfiable.

Upper bound: for every $e \in \mathbb{E}$ there is a poly. computable equivalent $e' \in \mathbb{E}_{\mathsf{PDL}}$. □

Question: Do we need the full aKAT language for arbitrary $\mathsf{F}_{A,t}$?

4. Language completeness

Guarded strings – 1

Disclaimer: We draw heavily on Hollenberg's (1998) relational completeness proof for an equational axiomatization of Test Algebras (relationally valid equations between PDL-programs), itself combining modal logic with (Kozen and Smith, 1997). We simplify and generalize.

Let Φ be a finite set of parameters $\phi_1, \ldots, \phi_n$. An atom over Φ is a sequence $\psi_1 \ldots \psi_n$ where $\psi_i \in \{\phi_i, \phi_i^{\perp}\}$. Notation $G \triangleleft \phi$ means "ϕ appears in atom G". $\mathbb{A}(\Phi)$ is the set of all atoms over Φ.

A guarded string over Φ is any sequence of the form

$$G_1 \mathsf{a}_1 G_2 \ldots \mathsf{a}_{n-1} G_n$$

where each $G_i \in \mathbb{A}(\Phi)$ and $\mathsf{a}_j \in \Sigma$. $\mathbb{GS}(\Phi)$ is the set of all guarded strings over Φ.

Guarded strings – 2

Fusion product is a partial binary operation on $\mathbb{GS}(\Phi)$ defined as follows:

$$xG \diamond Hy = \begin{cases} xGy & G = H \\ \text{undefined} & G \neq H \end{cases}$$

Fusion product is lifted to Φ-guarded languages $K, L \subseteq \mathbb{GS}(\Phi)$ as expected: $L \diamond K = \{w \diamond u \mid w \in L \ \ \& \ \ u \in K\}$.

KAT and the algebra of guarded languages – 1

Definition 5

The algebra of Φ-guarded languages is

$$\mathbf{GL}(\Phi) = \langle 2^{\mathbb{GS}(\Phi)}, 2^{\mathbb{A}(\Phi)}, \cup, \diamond, {}^*, {}^\perp, {}^\top, \emptyset, \mathbb{A}(\Phi) \rangle$$

where $K^* = \bigcup_{n \geq 0} K^n$ *(*$K^0 = \mathbb{A}(\Phi)$ *and* $K^{n+1} = K^n \diamond K$*) and*

$$L^\perp = \{G \in \mathbb{A}(\Phi) \mid \{G\} \diamond L = \emptyset\} \qquad L^\top = \{G \in \mathbb{A}(\Phi) \mid \{G\} \diamond L \neq \emptyset\}.$$

Definition 6

If Φ is a finite set of parameters, then the standard Φ-interpretation of $\mathbb{RE}(\Phi^\pm)$ is the unique homomorphism $[-]_\Phi : \mathbb{RE}(\Phi^\pm) \to \mathbf{GL}(\Phi)$ such that

$$[\mathtt{a}]_\Phi = \{G\mathtt{a}H \mid G, H \in \mathbb{A}(\Phi)\} \qquad [\phi]_\Phi = \{G \mid G \in \mathbb{A}(\Phi) \;\;\&\;\; G \lhd \phi\}$$

for $\mathtt{a} \in \Sigma$ *and* $\phi \in \Phi^\pm$*.*

KAT and the algebra of guarded languages – 2

Theorem 2 (Essentially (Kozen and Smith 1997))

Let Φ be a finite set of parameters. For all $e, f \in \mathbb{E}_{\mathsf{KAT}(\Phi)}$,

$$e \overset{\mathsf{KAT}(\Phi)}{\equiv} f \iff [e]_\Phi = [f]_\Phi$$

Consequently, we have

$$[e]_\Gamma = [f]_\Gamma \implies e \equiv f$$

for all $e, f \in \mathbb{RE}(\Gamma^\pm) \subseteq \mathbb{E}$.

The converse fails! $\mathtt{a}^\top \mathtt{a}^{\perp\top} \in \mathbb{A}(\Gamma)$ for $\Gamma = \{\mathtt{a}^\top, \mathtt{a}^{\perp\top}\}$. Hence, $[\mathtt{a}^\top \cdot \mathtt{a}^{\perp\top}]_\Gamma \neq \emptyset$. $\to$ Pay attention to consistency of atoms! Moreover, $[\mathtt{a}^\perp \cdot \mathtt{a}]_\Gamma = \{G\mathtt{a}H \mid G \triangleleft \mathtt{a}^\perp\} \neq \emptyset$. $\to$ Consider only consistent one-step gstrings!

aKAT and the algebra of consistently guarded languages – 1

We don't distinguish between a non-empty sequence of expressions $e_1 \ldots e_n$ and the expression $e_1 \cdot \ldots \cdot e_n$ (assuming some fixed bracketing). An atom G is consistent iff $G \not\equiv 0$. $\mathbb{C}(\Phi)$ is the set of all consistent atoms over Φ.

A consistently guarded string over Φ is any guarded string $G_1 \mathsf{a}_1 \ldots \mathsf{a}_{n-1} G_n$ where all $G_i \in \mathbb{C}(\Phi)$. $\mathbb{CS}(\Phi)$ is the set of consistently guarded strings over Φ.

Definition 7

The algebra of consistently Φ-guarded languages is

$$\mathbf{CL}(\Phi) = \langle 2^{\mathbb{CS}(\Phi)}, 2^{\mathbb{C}(\Phi)}, \cup, \diamond, {}^*, {}^\perp, {}^\top, \emptyset, \mathbb{C}(\Phi) \rangle$$

where $K^* = \bigcup_{n \geq 0} K^n$ *(*$K^0 = \mathbb{C}(\Phi)$ *and* $K^{n+1} = K^n \diamond K$*) and*

$$L^\perp = \{G \in \mathbb{C}(\Phi) \mid \{G\} \diamond L = \emptyset\} \qquad L^\top = \{G \in \mathbb{C}(\Phi) \mid \{G\} \diamond L \neq \emptyset\} .$$

Not a subalgbera ob $\mathbf{GL}(\Phi)$ although of course $\mathbb{CS}(\Phi) \subseteq \mathbb{GS}(\Phi)$.

aKAT and the algebra of consistently guarded languages – 2

Definition 8

Let Γ *be a set of parameters. The* *canonical* Γ*-interpretation* *of* $\mathbb{E}$ *is the unique homomorphism* $[\![-]\!] : \mathbb{E} \to \mathbf{CL}(\Gamma)$ *such that*

$$[\![\mathtt{a}]\!]_\Gamma = \{G\mathtt{a}H \in \mathbb{CS}(\Gamma) \mid G\mathtt{a}H \not\equiv \mathtt{0}\} \quad [\![\mathtt{p}]\!]_\Gamma = \{G \mid G \in \mathbb{C}(\Gamma) \ \ \& \ \ G \leqq \mathtt{p}\}$$

for all $\mathtt{a} \in \Sigma$ *and* $\mathtt{p} \in \Pi$*.*

Lemma 1 (Language soundness)

Let Γ *be any finite set of parameters. For all* $e, f \in \mathbb{E}$*:*

$$e \equiv f \implies [\![e]\!]_\Gamma = [\![f]\!]_\Gamma \,.$$

Language completeness – 1

Definition 9

We define $\widehat{\ }$ *as the smallest function* $\mathbb{RE}(\Gamma) \to \mathbb{RE}(\Gamma^{\pm})$ *such that (for* $\phi \in \Gamma$ *and* $\mathtt{a} \in \Sigma$*)*

$$\widehat{\phi} = \sum\{G \in \mathbb{C}(\Gamma) \mid G \leqq \phi\} \qquad \widehat{\mathtt{a}} = \sum[\![\mathtt{a}]\!]_\Gamma \qquad \widehat{\mathtt{1}} = \sum \mathbb{C}(\Gamma)$$

and that commutes with $\mathtt{0}$, $\cdot, +$ *and* *.

Lemma 2

For all $e \in \mathbb{RE}(\Gamma)$, $e \equiv \widehat{e}$.

Lemma 3

If Γ *is "FL-closed", then* $[\![e]\!]_\Gamma = [\widehat{e}]_\Gamma$ *for all* $e \in \mathbb{RE}(\Gamma)$.

Language completeness – 2

Theorem 3 (Language completeness)

Let $E \subseteq \mathbb{E}$ be finite and let Γ be the FL-closure of the sets of tests of subformulas of elements of E. Then, for all $e, f \in E$:

$$e \equiv f \iff [\![e]\!]_\Gamma = [\![f]\!]_\Gamma$$

Proof.

The implication from left to right follows from Lemma 1. The converse implication is established as follows:

$$[\![e]\!]_\Gamma = [\![f]\!]_\Gamma \xrightarrow{\text{Lemma 3}} [\widehat{e}]_\Gamma = [\widehat{f}]_\Gamma \xrightarrow{\text{Theorem 2}} \widehat{e} \overset{\mathsf{KAT}(\Gamma)}{\equiv} \widehat{f}$$

$$\widehat{e} \overset{\mathsf{KAT}(\Gamma)}{\equiv} \widehat{f} \xrightarrow{\text{by def.}} \widehat{e} \equiv \widehat{f} \xrightarrow{\text{Lemma 2}} e \equiv f$$

□

5. Relational completeness

Relational completeness – 1

Definition 10

We define the function $\mathsf{cay} : 2^{\mathbb{CS}(\Gamma)} \to 2^{\mathbb{CS}(\Gamma) \times \mathbb{CS}(\Gamma)}$ *as follows:*

$$\mathsf{cay}(L) = \{\langle w, w \diamond u \rangle \mid w \in \mathbb{CS}(\Gamma) \ \ \& \ \ u \in L\}$$

Definition 11

Define the relational model $CS(\Gamma) = \langle \mathbb{CS}(\Gamma), \mathsf{rel}_{CS(\Gamma)}, \mathsf{sat}_{CS(\Gamma)} \rangle$ *where*

$$\mathsf{rel}_{CS(\Gamma)}(\mathsf{a}) = \mathsf{cay}([\![\mathsf{a}]\!]_\Gamma) \qquad \mathsf{sat}_{CS(\Gamma)}(\mathtt{p}) = \{w \mid \mathsf{last}(w) \leqq \mathtt{p}\}$$

for $\mathtt{a} \in \Sigma$ *and* $\mathtt{p} \in \Pi$*.*

Relational completeness – 2

Lemma 4

If Γ is FL-closed, then for all $e \in \mathbb{RE}(\Gamma)$,

$$\text{cay}\left(\llbracket e \rrbracket_\Gamma\right) = \llbracket e \rrbracket_{CS(\Gamma)} .$$

Theorem 4 (Relational completeness)

For all $e, f \in \mathbb{E}$:

$$e \equiv f \iff (\forall M)(\llbracket e \rrbracket_M = \llbracket f \rrbracket_M)$$

Theorem 5

The problem of deciding equivalence of arbitrary expressions is $\mathrm{EXPTIME}$-complete.

6. Completeness and complexity of fragments

Completeness and complexity of fragments – 1

Theorem 6

Take a finite $E \subseteq \mathbb{E}$ and $e, f \in E$. The following are equivalent:

1. $e \overset{\mathsf{K}}{\equiv} f$
2. $[\![e]\!]_\Gamma = [\![f]\!]_\Gamma$ *where Γ is the FL-closure of $\mathrm{St}(E)$*
3. $[\![e]\!]_M = [\![f]\!]_M$ *for all relational models M*

Completeness and complexity of fragments – 2

Lemma 5

Let e' be the result of replacing every occurrence of $\mathtt{a}_n$ in e by an occurrence of $\mathtt{a}_{2n}$ and replacing every occurrence of $\mathtt{p}_n$ by an occurrence of $(\mathtt{a}_{2n+1})^\top$. Then

$$e \equiv f \iff e' \equiv f' .$$

Completeness and complexity of fragments – 2

Lemma 5

Let e' be the result of replacing every occurrence of $\mathtt{a}_n$ in e by an occurrence of $\mathtt{a}_{2n}$ and replacing every occurrence of $\mathtt{p}_n$ by an occurrence of $(\mathtt{a}_{2n+1})^\top$. Then

$$e \equiv f \iff e' \equiv f' .$$

Proof. Left to right: Equivalence is preserved under substitution. Moreover, clearly $\mathtt{p}' \equiv (\mathtt{p}')^\top$.
Right to left: If $e \not\equiv f$, then there is a relational model M where $[\![e]\!]_M \neq [\![f]\!]_M$ (Theorem 4).
We define M' by taking the universe X of M and stipulating that

$$\mathsf{rel}_{M'}(\mathtt{a}_m) = \begin{cases} \mathsf{rel}_M(\mathtt{a}_n) & m = 2n \\ 1_{\mathsf{sat}_M(\mathtt{p}_n)} & m = 2n+1 \end{cases} \qquad \mathsf{sat}_{M'}(\mathtt{p}) = \emptyset$$

It can be shown by induction on g that $[\![g]\!]_M = [\![g']\!]_{M'}$. Only the base case is interesting. The base case for $\mathtt{a}_n$: $[\![(\mathtt{a}_n)']\!]_{M'} = [\![\mathtt{a}_{2n}]\!]_{M'} = [\![\mathtt{a}_n]\!]_M$. The base case for $\mathtt{p}_n$:
$[\![(\mathtt{p}_n)']\!]_{M'} = [\![(\mathtt{a}_{2m+1})^\top]\!]_{M'} = \{\langle x, x\rangle \mid \exists y.\langle x, y\rangle \in [\![\mathtt{a}_{2n+1}]\!]_{M'}\} = [\![\mathtt{p}_n]\!]_M$. Now clearly $[\![e']\!]_{M'} \neq [\![f']\!]_{M'}$ and so $e' \not\equiv f'$ by relational soundness. □

Completeness and complexity of fragments – 3

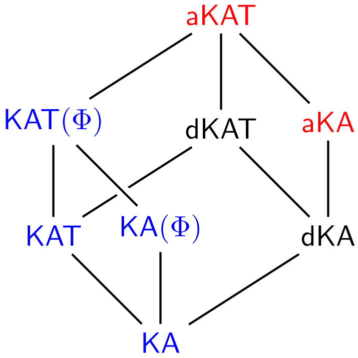
Theorem 7

The problem of deciding equivalence between aKA*-expressions is* $\mathrm{EXPTIME}$*-complete.*

Proof. The problem is in $\mathrm{EXPTIME}$ since so is deciding equivalence between arbitrary expressions. The problem is $\mathrm{EXPTIME}$-complete thanks to Lemma 5: deciding equivalence between arbitrary expressions can be polynomially reduced to deciding equivalence between aKA-expressions. The former is $\mathrm{EXPTIME}$-complete by Theorem 1. □

aKA is the "Kleene algebra with domain" of Desharnais and Struth (2011).

Completeness and complexity of fragments – 4



PSPACE-compete   EXPTIME-complete

7. Conclusion

Conclusion

We discussed various extensions of KA(T) with $^{\top}$ (domain) and $^{\bot}$ (antidomain). This family contains KA, KAT, PDL (aKAT), and various versions of “Kleene algebra with domain” that appeared in the literature.

Results:

- aKAT and all fragments considered are sound and complete with respect to relational and (parametrized) guarded-language models
- aKAT and aKAT are $\mathrm{EXPTIME}$-complete

Conclusion

Problems:

1. Are dKA and dKAT $\mathrm{EXPTIME}$-hard?
2. Do fragments of quasi-eq. theories of aKAT and its fragments with assumptions $e \equiv 0$ reduce to their eq. theories?
3. Are there natural fragments of aKAT that are stronger than KAT but still have a PSPACE-complete eq. theory?
4. What are the natural automata-theoretic formulation of the various versions of Kleene algebra with dynamic tests considered here?

Thank you!

References I

E. Cohen, D. Kozen, and F. Smith. The complexity of Kleene algebra with tests. Technical Report TR96-1598, Computer Science Department, Cornell University, July 1996.

J. Desharnais, B. Möller, and G. Struth. Kleene algebra with domain. *ACM Trans. Comput. Logic*, 7(4):798–833, oct 2006. doi:10.1145/1183278.1183285.

J. Desharnais and G. Struth. Internal axioms for domain semirings. *Science of Computer Programming*, 76(3):181–203, 2011. Special issue on the Mathematics of Program Construction (MPC 2008). doi:10.1016/j.scico.2010.05.007.

J. Groenendijk and M. Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14(1):39–100, 1991. doi:10.1007/BF00628304.

M. Hollenberg. Equational axioms of test algebra. In M. Nielsen and W. Thomas, editors, *International Workshop on Computer Science Logic. CSL 1997*, pages 295–310, Berlin, Heidelberg, 1998. Springer. doi:10.1007/BFb0028021.

T. Kappé. Kleene algebra. Course notes, University of Amsterdam, 2022. URL: https://staff.fnwi.uva.nl/t.w.j.kappe/teaching/ka/.

T. Kappé. Elements of Kleene algebra. Course notes, ESSLLI 2023, 2023. URL: https://tobias.kap.pe/esslli/.

D. Kozen. Kleene algebra with tests and commutativity conditions. In T. Margaria and B. Steffen, editors, *Proc. Second Int. Workshop Tools and Algorithms for the Construction and Analysis of Systems (TACAS'96)*, volume 1055 of *Lecture Notes in Computer Science*, pages 14–33, Passau, Germany, March 1996. Springer-Verlag. doi:10.1007/3-540-61042-1_35.

References II

D. Kozen. Kleene algebra with tests. *ACM Trans. Program. Lang. Syst.*, 19(3):427–443, May 1997. doi:10.1145/256167.256195.

D. Kozen and F. Smith. Kleene algebra with tests: Completeness and decidability. In D. van Dalen and M. Bezem, editors, *Computer Science Logic*, pages 244–259, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg. doi:10.1145/256167.256195.