A cyclic proof system for Guarded Kleene Algebra with Tests

(j.w.w. Dexter Kozen & Alexandra Silva)

Jan Rooduijn

ILLC, University of Amsterdam

The Proof Society Workshop

11 November 2022

Syntax

Syntax

$$a, b ::= t \in T \mid a + b \mid a \cdot b \mid \overline{a} \mid 0 \mid 1$$

$$e, f ::= p \in \Sigma \mid e \cdot f \mid e +_b f \mid e^{(b)}$$

Syntax

Syntax

$$a, b ::= t \in T \mid a + b \mid a \cdot b \mid \overline{a} \mid 0 \mid 1$$
$$e, f ::= p \in \Sigma \mid e \cdot f \mid e +_b f \mid e^{(b)}$$

Intuition Imperative programs

Syntax

Syntax $a, b ::= t \in T \mid a + b \mid a \cdot b \mid \overline{a} \mid 0 \mid 1$

$e, f ::= p \in \Sigma \mid e \cdot f \mid e +_b f \mid e^{(b)}$

Intuition Imperative programs

Fragments KA $\subset$ KAT $\supset$ GKAT

Semantics

$$\mathsf{At} = 2^T \qquad \mathsf{GS} = \{\alpha_0 p_1 \alpha_1 \cdots p_n \alpha_n \mid n \geq 0\} \qquad [\![e]\!] \subseteq \mathcal{P}(\mathsf{GS})$$

Semantics

$\mathsf{At} = 2^T$ $\quad$ $\mathsf{GS} = \{\alpha_0 p_1 \alpha_1 \cdots p_n \alpha_n \mid n \geq 0\}$ $\quad$ $[\![e]\!] \subseteq \mathcal{P}(\mathsf{GS})$

Intuition Atoms are states of the machine, programs are executions.

Semantics

$\mathrm{At} = 2^T$ $\quad\quad$ $\mathrm{GS} = \{\alpha_0 p_1 \alpha_1 \cdots p_n \alpha_n \mid n \geq 0\}$ $\quad\quad$ $[\![e]\!] \subseteq \mathcal{P}(\mathrm{GS})$

Intuition Atoms are states of the machine, programs are executions.

Determinacy For every $\alpha_0 p_1 x$ and $\beta_0 q_1 y$ in $[\![e]\!]$: $\alpha_0 = \beta_0 \Rightarrow p_1 = q_1$.

Example

```
while a do
    if b then
        p;
    else
        q;
    end
end
```

Example

```
while a do
    if b then
        p;
    else
        q;
    end
end
```

$$(p +_b q)^{(a)}$$

Example

```
while a do
    if b then
        p;
    else
        q;
    end
end
```

$(p +_b q)^{(a)}$

$$\alpha p \beta q \gamma \in [\![(p +_b q)^{(a)}]\!]$$
$$\Updownarrow$$
$$\alpha \leq a, b$$
$$\beta \leq a, \overline{b}$$
$$\gamma \leq \overline{a}$$

Some earlier results[1]

- $[\![e]\!] = [\![f]\!] \Leftrightarrow \mathcal{L}(\mathbb{A}_e) = \mathcal{L}(\mathbb{A}_f)$.

[1] S. Smolka et al., *Guarded Kleene algebra with tests: verification of uninterpreted programs in nearly linear time.*

Some earlier results[1]

- $[\![e]\!] = [\![f]\!] \Leftrightarrow \mathcal{L}(\mathbb{A}_e) = \mathcal{L}(\mathbb{A}_f)$.
- Equivalence of expressions is decidable is nearly linear time.

[1] S. Smolka et al., *Guarded Kleene algebra with tests: verification of uninterpreted programs in nearly linear time*.

Some earlier results[1]

- $[\![e]\!] = [\![f]\!] \Leftrightarrow \mathcal{L}(\mathbb{A}_e) = \mathcal{L}(\mathbb{A}_f)$.
- Equivalence of expressions is decidable is nearly linear time.
- 'Algebraic' equational axiomatisation eGKAT:

$$\frac{g \equiv eg +_b f \qquad \text{e is productive}}{g \equiv e^{(b)} f}$$

[1]S. Smolka et al., *Guarded Kleene algebra with tests: verification of uninterpreted programs in nearly linear time.*

Some earlier results[1]

- $[\![e]\!] = [\![f]\!] \Leftrightarrow \mathcal{L}(\mathbb{A}_e) = \mathcal{L}(\mathbb{A}_f)$.
- Equivalence of expressions is decidable is nearly linear time.
- 'Algebraic' equational axiomatisation eGKAT:

$$\frac{g \equiv eg +_b f \qquad \text{e is productive}}{g \equiv e^{(b)} f}$$

- Soundness: $e\text{GKAT} \vdash e \equiv f$ implies $[\![e]\!] = [\![f]\!]$.

[1] S. Smolka et al., *Guarded Kleene algebra with tests: verification of uninterpreted programs in nearly linear time.*

Some earlier results[1]

- $[\![e]\!] = [\![f]\!] \Leftrightarrow \mathcal{L}(\mathbb{A}_e) = \mathcal{L}(\mathbb{A}_f)$.
- Equivalence of expressions is decidable is nearly linear time.
- 'Algebraic' equational axiomatisation eGKAT:

$$\frac{g \equiv eg +_b f \qquad \text{e is productive}}{g \equiv e^{(b)} f}$$

- Soundness: $e\text{GKAT} \vdash e \equiv f$ implies $[\![e]\!] = [\![f]\!]$.
- Completeness of eGKAT is unknown.

[1] S. Smolka et al., *Guarded Kleene algebra with tests: verification of uninterpreted programs in nearly linear time.*

Proof Theory

	Sound	Complete
*e*GKAT		

Proof Theory

	Sound	Complete
eGKAT	✓	

Proof Theory

	Sound	Complete
eGKAT	✓	?

Proof Theory

	Sound	Complete
eGKAT	✓	?
eKA		

Proof Theory

	Sound	Complete
*e*GKAT	✓	?
*e*KA	✓	

Proof Theory

	Sound	Complete
eGKAT	✓	?
eKA	✓	✓

Proof Theory

	Sound	Complete	Structural
eGKAT	✓	?	
eKA	✓	✓	

Proof Theory

	Sound	Complete	Structural
eGKAT	✓	?	
eKA	✓	✓	✗

Proof Theory

	Sound	Complete	Structural
eGKAT	✓	?	✗
eKA	✓	✓	✗

Proof Theory

	Sound	Complete	Structural
*e*GKAT	✓	?	✗
*e*KA	✓	✓	✗
HKA[2]			

[2] A. Das and D. Pous, *A cut-free cyclic proof system for Kleene algebra*.

Proof Theory

	Sound	Complete	Structural
*e*GKAT	✓	?	✗
*e*KA	✓	✓	✗
HKA[2]			✓

[2] A. Das and D. Pous, *A cut-free cyclic proof system for Kleene algebra*.

Proof Theory

	Sound	Complete	Structural
*e*GKAT	✓	?	✗
*e*KA	✓	✓	✗
HKA[2]	✓		✓

[2] A. Das and D. Pous, *A cut-free cyclic proof system for Kleene algebra*.

Proof Theory

	Sound	Complete	Structural
*e*GKAT	✓	?	✗
*e*KA	✓	✓	✗
HKA[2]	✓	✓	✓

[2] A. Das and D. Pous, *A cut-free cyclic proof system for Kleene algebra*.

Proof Theory

	Sound	Complete	Structural
*e*GKAT	✓	?	✗
*e*KA	✓	✓	✗
HKA[2]	✓	✓	✓
SGKAT			

[2] A. Das and D. Pous, *A cut-free cyclic proof system for Kleene algebra*.

Proof Theory

	Sound	Complete	Structural
*e*GKAT	✓	?	✗
*e*KA	✓	✓	✗
HKA[2]	✓	✓	✓
SGKAT			✓

[2] A. Das and D. Pous, *A cut-free cyclic proof system for Kleene algebra*.

Proof Theory

	Sound	Complete	Structural
*e*GKAT	✓	?	✗
*e*KA	✓	✓	✗
HKA[2]	✓	✓	✓
SGKAT	✓		✓

[2] A. Das and D. Pous, *A cut-free cyclic proof system for Kleene algebra*.

Proof Theory

	Sound	Complete	Structural
*e*GKAT	✓	?	✗
*e*KA	✓	✓	✗
HKA[2]	✓	✓	✓
SGKAT	✓	✓	✓

[2] A. Das and D. Pous, *A cut-free cyclic proof system for Kleene algebra*.

First result: no multisets needed

SKA

$$\frac{e \Rightarrow f}{e \Rightarrow f + g}\ +\text{-}r_1 \qquad\qquad \frac{e \Rightarrow g}{e \Rightarrow f + g}\ +\text{-}r_2$$

First result: no multisets needed

SKA

$$\frac{e \Rightarrow f}{e \Rightarrow f + g} \; +\text{-}r_1 \qquad\qquad \frac{e \Rightarrow g}{e \Rightarrow f + g} \; +\text{-}r_2$$

HKA

$$\frac{e \Rightarrow [f, g]}{e \Rightarrow f + g} \; +\text{-}r$$

First result: no multisets needed

SKA

$$\frac{e \Rightarrow f}{e \Rightarrow f + g}\ +\text{-}r_1 \qquad\qquad \frac{e \Rightarrow g}{e \Rightarrow f + g}\ +\text{-}r_2$$

HKA

$$\frac{e \Rightarrow [f, g]}{e \Rightarrow f + g}\ +\text{-}r$$

SGKAT

$$\frac{b \cdot e \Rightarrow f \qquad \overline{b} \cdot e \Rightarrow f}{e \Rightarrow f +_b g}\ +\text{-}r$$

Future work

- What is the complexity of proof search?

Future work

- What is the complexity of proof search?
- Use SGKAT to prove completeness of *e*GKAT or some other algebraic axiomatisation.

Thank you!

Left logical rules

$$\frac{\Gamma \Rightarrow_{A\restriction b} \Delta}{b, \Gamma \Rightarrow_A \Delta}\ b\text{-}l \qquad \frac{e, g, \Gamma \Rightarrow_A \Delta}{e \cdot g, \Gamma \Rightarrow_A \Delta}\ \cdot\text{-}l$$

$$\frac{e, \Gamma \Rightarrow_{A\restriction b} \Delta \qquad e, \Gamma \Rightarrow_{A\restriction \overline{b}} \Delta}{e +_b f, \Gamma \Rightarrow_A \Delta}\ +_b\text{-}l \qquad \frac{e, e^{(b)}, \Gamma \Rightarrow_{A\restriction b} \Delta \qquad \Gamma \Rightarrow_{A\restriction \overline{b}} \Delta}{e^{(b)}, \Gamma \Rightarrow_A \Delta}\ (b)\text{-}l$$

Right logical rules

$$(\dagger)\ \frac{\Gamma \Rightarrow_A \Delta}{\Gamma \Rightarrow_A b, \Delta}\ b\text{-}r \qquad \frac{\Gamma \Rightarrow_A e, f, \Delta}{\Gamma \Rightarrow_A e \cdot f, \Delta}\ \cdot\text{-}r$$

$$\frac{\Gamma \Rightarrow_{A\restriction b} e, \Delta \qquad \Gamma \Rightarrow_{A\restriction \overline{b}} f, \Delta}{\Gamma \Rightarrow_A e +_b f, \Delta}\ +_b\text{-}r \qquad \frac{\Gamma \Rightarrow_{A\restriction b} e, e^{(b)}, \Delta \qquad \Gamma \Rightarrow_{A\restriction \overline{b}} \Delta}{\Gamma \Rightarrow_A e^{(b)}, \Delta}\ (b)\text{-}r$$

Axioms and modal rule

$$\frac{}{\varepsilon \Rightarrow_A \varepsilon}\ \mathsf{id} \qquad \frac{}{\Gamma \Rightarrow_\emptyset \Delta}\ \bot \qquad \frac{\Gamma \Rightarrow_{\mathsf{At}} \Delta}{p, \Gamma \Rightarrow_A p, \Delta}\ \mathsf{k}$$

$A \restriction b = \{\alpha \in A : \alpha \leq b\}$ $\qquad$ $(\dagger)$ $A \restriction b = A$.