Kleene Algebra with Observations

Tobias Kappé Paul Brunet Jurriaan Rot

Alexandra Silva Jana Wagemaker Fabio Zanasi

University College London

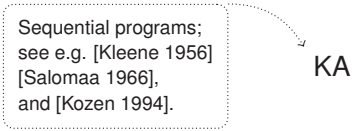
CONCUR 2019

Introduction

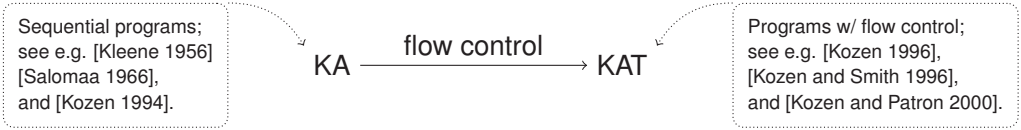
Introduction

Introduction

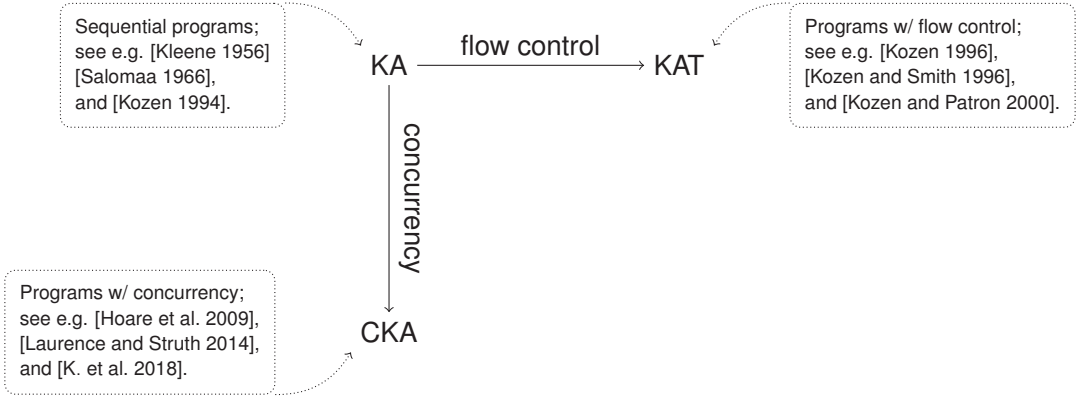
Introduction



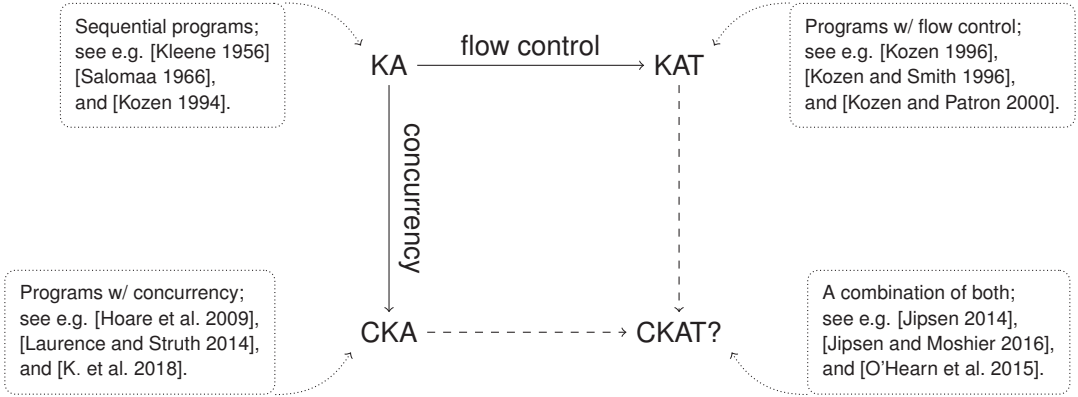
Introduction



Introduction



Introduction



Introduction

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

$$p, q ::= \bot \mid \top \mid t \in T \mid p \vee q \mid p \wedge q \mid \overline{p}$$

$$e, f ::= \underbrace{p \mid 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e^*}_{\text{KAT terms}} \mid e \parallel f$$

$$p, q ::= \bot \mid \top \mid t \in T \mid p \vee q \mid p \wedge q \mid \overline{p}$$

Introduction

$$e, f ::= \overbrace{p \mid 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e^* \mid e \parallel f}^{\text{CKA terms}}$$

$$p, q ::= \bot \mid \top \mid t \in T \mid p \vee q \mid p \wedge q \mid \overline{p}$$

Introduction

$$e + 0 \equiv e \qquad e + e \equiv e \qquad e + f \equiv f + e \qquad e + (f + g) \equiv (e + f) + g$$

$$e \cdot 1 \equiv e \qquad e \equiv 1 \cdot e \qquad e \cdot 0 \equiv 0 \qquad 0 \equiv 0 \cdot e \qquad e \cdot (f \cdot g) \equiv (e \cdot f) \cdot g$$

$$e \cdot (f + g) \equiv e \cdot f + e \cdot g \qquad 1 + e \cdot e^\star \equiv e^\star \qquad e + f \cdot g \leqq g \implies f^\star \cdot e \leqq g$$

$$(e + f) \cdot g \equiv e \cdot g + f \cdot g \qquad 1 + e^\star \cdot e \equiv e^\star \qquad e + f \cdot g \leqq f \implies e \cdot g^\star \leqq f$$

$$e \leqq f \iff e + f \equiv f$$

Introduction

$p \vee \bot \equiv p$ $\quad p \vee q \equiv q \vee p$ $\quad p \vee \overline{p} \equiv \top$ $\quad p \vee (q \vee r) \equiv (p \vee q) \vee r$

$p \wedge \top \equiv p$ $\quad p \wedge q \equiv q \wedge p$ $\quad p \wedge \overline{p} \equiv \bot$ $\quad p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$

$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $\quad p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$

$p \vee q \equiv p + q$ $\quad p \wedge q \equiv p \cdot q$ $\quad \top \equiv 1$ $\quad \bot \equiv 0$

Introduction

$$e \parallel f \equiv f \parallel e \qquad e \parallel 1 \equiv e \qquad e \parallel 0 \equiv 0 \qquad (e + f) \parallel g \equiv e \parallel g + f \parallel g$$

$$e \parallel (f \parallel g) \equiv (e \parallel f) \parallel g \qquad (e \parallel f) \cdot (g \parallel h) \leqq (e \cdot g) \parallel (f \cdot h)$$

Introduction

$$e \parallel f \equiv f \parallel e \qquad e \parallel 1 \equiv e \qquad e \parallel 0 \equiv 0 \qquad (e + f) \parallel g \equiv e \parallel g + f \parallel g$$

$$e \parallel (f \parallel g) \equiv (e \parallel f) \parallel g \qquad (e \parallel f) \cdot (g \parallel h) \leqq (e \cdot g) \parallel (f \cdot h)$$

In particular: $e \cdot f \leqq e \parallel f$.

The problem with CKAT

$$p \cdot e \cdot \overline{p} \leqq (p \parallel e) \cdot \overline{p}$$

The problem with CKAT

$$
\begin{aligned}
p \cdot e \cdot \overline{p} &\leqq (p \parallel e) \cdot \overline{p} \\
&\equiv (p \parallel e) \cdot (\overline{p} \parallel 1)
\end{aligned}
$$

The problem with CKAT

$$
\begin{aligned}
p \cdot e \cdot \overline{p} &\leqq (p \parallel e) \cdot \overline{p} \\
&\equiv (p \parallel e) \cdot (\overline{p} \parallel 1) \\
&\leqq (p \cdot \overline{p}) \parallel (e \cdot 1)
\end{aligned}
$$

The problem with CKAT

$$
\begin{aligned}
p \cdot e \cdot \overline{p} &\leqq (p \parallel e) \cdot \overline{p} \\
&\equiv (p \parallel e) \cdot (\overline{p} \parallel 1) \\
&\leqq (p \cdot \overline{p}) \parallel (e \cdot 1) \\
&\equiv (p \cdot \overline{p}) \parallel e
\end{aligned}
$$

The problem with CKAT

$$
\begin{aligned}
p \cdot e \cdot \overline{p} &\leqq (p \parallel e) \cdot \overline{p} \\
&\equiv (p \parallel e) \cdot (\overline{p} \parallel 1) \\
&\leqq (p \cdot \overline{p}) \parallel (e \cdot 1) \\
&\equiv (p \cdot \overline{p}) \parallel e \\
&\equiv (p \wedge \overline{p}) \parallel e
\end{aligned}
$$

The problem with CKAT

$$
\begin{aligned}
p \cdot e \cdot \overline{p} &\leqq (p \parallel e) \cdot \overline{p} \\
&\equiv (p \parallel e) \cdot (\overline{p} \parallel 1) \\
&\leqq (p \cdot \overline{p}) \parallel (e \cdot 1) \\
&\equiv (p \cdot \overline{p}) \parallel e \\
&\equiv (p \wedge \overline{p}) \parallel e \\
&\equiv \bot \parallel e
\end{aligned}
$$

The problem with CKAT

$$
\begin{aligned}
p \cdot e \cdot \overline{p} &\leqq (p \parallel e) \cdot \overline{p} \\
&\equiv (p \parallel e) \cdot (\overline{p} \parallel 1) \\
&\leqq (p \cdot \overline{p}) \parallel (e \cdot 1) \\
&\equiv (p \cdot \overline{p}) \parallel e \\
&\equiv (p \wedge \overline{p}) \parallel e \\
&\equiv \bot \parallel e \\
&\equiv 0 \parallel e
\end{aligned}
$$

The problem with CKAT

$$
\begin{aligned}
p \cdot e \cdot \overline{p} &\leqq (p \parallel e) \cdot \overline{p} \\
&\equiv (p \parallel e) \cdot (\overline{p} \parallel 1) \\
&\leqq (p \cdot \overline{p}) \parallel (e \cdot 1) \\
&\equiv (p \cdot \overline{p}) \parallel e \\
&\equiv (p \wedge \overline{p}) \parallel e \\
&\equiv \bot \parallel e \\
&\equiv 0 \parallel e \\
&\equiv 0
\end{aligned}
$$

The problem with CKAT

$$
\begin{aligned}
p \cdot e \cdot \overline{p} &\leqq (p \parallel e) \cdot \overline{p} \\
&\equiv (p \parallel e) \cdot (\overline{p} \parallel 1) \\
&\leqq (p \cdot \overline{p}) \parallel (e \cdot 1) \\
&\equiv (p \cdot \overline{p}) \parallel e \\
&\equiv (p \wedge \overline{p}) \parallel e \\
&\equiv \bot \parallel e \\
&\equiv 0 \parallel e \\
&\equiv 0
\end{aligned}
$$

The problem with CKAT

The problem with CKAT

Drop $e \parallel 0 \equiv 0$?

The problem with CKAT

Drop $e \parallel 0 \equiv 0$?

Drop exchange law?

The problem with CKAT

Non-congruence?

Drop $e \parallel 0 \equiv 0$?

Drop exchange law?

The problem with CKAT

See also [Bergstra and Ponse 2011].

The problem with CKAT

See also [Bergstra and Ponse 2011].

The problem with CKAT

See also [Bergstra and Ponse 2011].

The problem with CKAT

See also [Bergstra and Ponse 2011].

A new hope

$$p \wedge q \equiv p \cdot q \rightsquigarrow p \wedge q \leqq p \cdot q$$

$$1 \not\equiv \top$$

A new hope

$$p \wedge q \equiv p \cdot q \rightsquigarrow p \wedge q \leqq p \cdot q$$

$1 \not\equiv \top$

Our contribution: a study of the *sequential fragment* (i.e., without $\|$).

A new hope

$$p \wedge q \equiv p \cdot q \rightsquigarrow p \wedge q \leqq p \cdot q \qquad \not{1} \not{\geq} \top$$

Our contribution: a study of the *sequential fragment* (i.e., without $\|$).

- A semantics $[\![-]\!]$ that sends a term to its *observation language*.

A new hope

$$p \wedge q \equiv p \cdot q \rightsquigarrow p \wedge q \leqq p \cdot q$$

$1 \not\geq \top$

Our contribution: a study of the *sequential fragment* (i.e., without $\|$).

- A semantics $[\![-]\!]$ that sends a term to its *observation language*.
- Practically feasible decision procedure w.r.t. semantics.

A new hope

$$p \wedge q \equiv p \cdot q \rightsquigarrow p \wedge q \leqq p \cdot q$$

$1 \geq \top$

Our contribution: a study of the *sequential fragment* (i.e., without $\|$).

- A semantics $[\![-]\!]$ that sends a term to its *observation language*.
- Practically feasible decision procedure w.r.t. semantics.
- Soundness & completeness: $e \equiv f$ if and only if $[\![e]\!] = [\![f]\!]$.

A new hope

$$p \wedge q \equiv p \cdot q \rightsquigarrow p \wedge q \leqq p \cdot q$$

$1 \geq \top$

Our contribution: a study of the *sequential fragment* (i.e., without $\|$).

- A semantics $[\![-]\!]$ that sends a term to its *observation language*.
- Practically feasible decision procedure w.r.t. semantics.
- Soundness & completeness: $e \equiv f$ if and only if $[\![e]\!] = [\![f]\!]$.

These form the foundation for extending to the full syntax.

Semantics

$$\llbracket 0 \rrbracket = \emptyset \qquad \llbracket a \rrbracket = \{a\} \qquad \llbracket e + f \rrbracket = \llbracket e \rrbracket \cup \llbracket f \rrbracket \qquad \llbracket e^\star \rrbracket = \llbracket e \rrbracket^\star$$
$$\llbracket 1 \rrbracket = \{\epsilon\} \qquad \llbracket p \rrbracket = \{\alpha \in \mathsf{At} : \alpha \leqslant p\} \qquad \llbracket e \cdot f \rrbracket = \llbracket e \rrbracket \cdot \llbracket f \rrbracket \,,$$

Semantics

$$\llbracket 0 \rrbracket = \emptyset \qquad \llbracket a \rrbracket = \{a\} \qquad \llbracket e + f \rrbracket = \llbracket e \rrbracket \cup \llbracket f \rrbracket \qquad \llbracket e^\star \rrbracket = \llbracket e \rrbracket^\star$$
$$\llbracket 1 \rrbracket = \{\epsilon\} \qquad \llbracket p \rrbracket = \{\alpha \in \mathsf{At} : \alpha \leqslant p\} \qquad \llbracket e \cdot f \rrbracket = \llbracket e \rrbracket \cdot \llbracket f \rrbracket \,,$$

$$\alpha \equiv \alpha \wedge \alpha \leqq \alpha \cdot \alpha \quad \text{but} \quad \llbracket \alpha \rrbracket \not\subseteq \llbracket \alpha \cdot \alpha \rrbracket$$

Semantics

$$\frac{w, x \in (\Sigma \cup \mathsf{At})^* \qquad \alpha \in \mathsf{At}}{w\alpha x \preceq w\alpha\alpha x}$$

Semantics

$$\frac{w, x \in (\Sigma \cup \mathsf{At})^* \qquad \alpha \in \mathsf{At}}{w\alpha x \preceq w\alpha\alpha x}$$

Let $[\![-]\!]^{\preceq}$ be the $\preceq$-downclosure of $[\![-]\!]$; then

Lemma (Soundness)

If $e \equiv f$, then $[\![e]\!]^{\preceq} = [\![f]\!]^{\preceq}$.

Decidability

Can we decide whether $[\![e]\!]^{\preceq} = [\![f]\!]^{\preceq}$?

Decidability

Can we decide whether $[\![e]\!]^{\preceq} = [\![f]\!]^{\preceq}$?

1. Translate e to automaton A_e such that $L(A_e) = [\![e]\!]^{\preceq}$.

Decidability

Can we decide whether $[\![e]\!]^{\preceq} = [\![f]\!]^{\preceq}$?

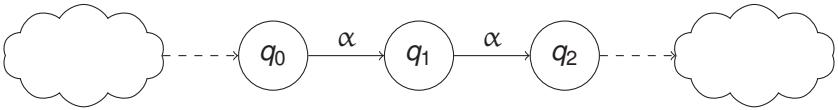
1. Translate e to automaton A_e such that $L(A_e) = [\![e]\!]^{\preceq}$.
2. Translate f to automaton A_f such that $L(A_f) = [\![f]\!]^{\preceq}$.

Decidability

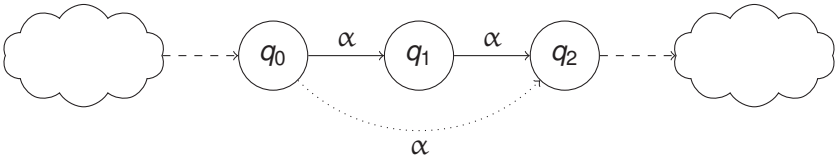
Can we decide whether $[\![e]\!]^{\preceq} = [\![f]\!]^{\preceq}$?

1. Translate e to automaton A_e such that $L(A_e) = [\![e]\!]^{\preceq}$.
2. Translate f to automaton A_f such that $L(A_f) = [\![f]\!]^{\preceq}$.
3. Decide whether $L(A_e) = L(A_f)$.

Decidability



Decidability



Decidability

$$\epsilon(0) = 0 \qquad \epsilon(e + f) = \epsilon(e) \vee \epsilon(f)$$
$$\epsilon(1) = 1 \qquad \epsilon(e \cdot f) = \epsilon(e) \wedge \epsilon(f)$$
$$\epsilon(a) = 0 \qquad \epsilon(e^*) = 1$$
$$\epsilon(p) = 0$$

See also [Brzozowski 1964].

Decidability

$$\delta(0, a) = \delta(1, a) = \emptyset \qquad \delta(e + f, a) = \delta(e, a) \cup \delta(f, a)$$
$$\delta(a, a') = \{1 : a = a'\} \qquad \delta(e \cdot f, a) = \{e' \cdot f : e' \in \delta(e, a)\} \cup \Delta(e, f, a)$$
$$\delta(p, a) = \emptyset \qquad \delta(e^\star, a) = \{e' \cdot e^\star : e' \in \delta(e, a)\},$$

$$\Delta(e, f, a) = \begin{cases} \delta(f, a) & \epsilon(e) = 1 \\ \emptyset & \epsilon(e) = 0 \end{cases}$$

See also [Antimirov 1996].

Decidability

$$\zeta(0,\alpha) = \delta(1,\alpha) = \emptyset \qquad \zeta(e+f,\alpha) = \zeta(e,\alpha) \cup \zeta(f,\alpha)$$
$$\zeta(a,\alpha) = \emptyset \qquad \zeta(e \cdot f,\alpha) = \{e' \cdot f : e' \in \zeta(e,\alpha)\} \cup Z(e,f,\alpha)$$
$$\zeta(p,\alpha) = \{1 : \alpha \leqslant p\} \qquad \zeta(e^\star,\alpha) = \{e' \cdot e^\star : e' \in \zeta(e,\alpha)\}\,,$$

$$Z(e,f,\alpha) = \begin{cases} \zeta(f,\alpha) & \epsilon(e') = 1 \text{ for some } e' \in \zeta(e,\alpha) \cup \{e\} \\ \emptyset & \text{otherwise} \end{cases}$$

Decidability

Let $\rho(e)$ be the expressions reached through δ or ζ from e.

Lemma

For any e, we have that $A_e = \langle \rho(e), \delta \cup \zeta, \epsilon \rangle$ is a finite non-deterministic automaton.

Decidability

Let $\rho(e)$ be the expressions reached through δ or ζ from e.

Lemma

For any e, we have that $A_e = \langle \rho(e), \delta \cup \zeta, \epsilon \rangle$ is a finite non-deterministic automaton.

Lemma

For any e, we have that $L(A_e) = [\![e]\!]^{\preceq}$.

Decidability

Let $\rho(e)$ be the expressions reached through δ or ζ from e.

Lemma

For any e, we have that $A_e = \langle \rho(e), \delta \cup \zeta, \epsilon \rangle$ is a finite non-deterministic automaton.

Lemma

For any e, we have that $L(A_e) = [\![e]\!]^{\preceq}$.

Theorem

Given e, f, we can decide whether $[\![e]\!]^{\preceq} = [\![f]\!]^{\preceq}$.

Axiomatisation

UCL

An expression e is ...

Axiomatisation

An expression e is ...

- ... *atomic* if its Boolean subterms are atoms.

Axiomatisation

An expression e is ...

- ... *atomic* if its Boolean subterms are atoms.
- ... *closed* if $[\![e]\!] = [\![e]\!]^{\preceq}$.

Axiomatisation

An expression e is ...

- ... *atomic* if its Boolean subterms are atoms.
- ... *closed* if $[\![e]\!] = [\![e]\!]^{\preceq}$.

Lemma

Let e and f be atomic and closed. If $[\![e]\!]^{\preceq} = [\![f]\!]^{\preceq}$, then $e \equiv f$.

Axiomatisation

Lemma

For every e, we can construct an atomic and closed $\hat{e}$ such that $e \equiv \hat{e}$.

Axiomatisation

UCL

Lemma

For every e, we can construct an atomic and closed $\hat{e}$ such that $e \equiv \hat{e}$.

Proof sketch.

Idea: convert the automaton for e back into an expression.

Axiomatisation

UCL

Lemma

For every e, we can construct an atomic and closed $\hat{e}$ such that $e \equiv \hat{e}$.

Proof sketch.

Idea: convert the automaton for e back into an expression.

Formally, this is the least x_e that solves the system generated by:

$$\epsilon(e) + \sum_{e' \in \delta(e,a)} a \cdot x_{e'} + \sum_{e' \in \zeta(e,\alpha)} \alpha \cdot x_{e'} \leqq x_e$$

Axiomatisation

UCL

Theorem

If $[\![e]\!]^{\preceq} = [\![f]\!]^{\preceq}$, *then* $e \equiv f$.

Axiomatisation

UCL

Theorem

If $[\![e]\!]^{\preceq} = [\![f]\!]^{\preceq}$*, then* $e \equiv f$*.*

Proof.

First, note that

$$[\![\hat{e}]\!] = [\![\hat{e}]\!]^{\preceq} = [\![e]\!]^{\preceq} = [\![f]\!]^{\preceq} = [\![\hat{f}]\!]^{\preceq} = [\![\hat{f}]\!]$$

Axiomatisation

UCL

Theorem

If $[\![e]\!]^{\preceq} = [\![f]\!]^{\preceq}$*, then* $e \equiv f$*.*

Proof.

First, note that

$$[\![\hat{e}]\!] = [\![\hat{e}]\!]^{\preceq} = [\![e]\!]^{\preceq} = [\![f]\!]^{\preceq} = [\![\hat{f}]\!]^{\preceq} = [\![\hat{f}]\!]$$

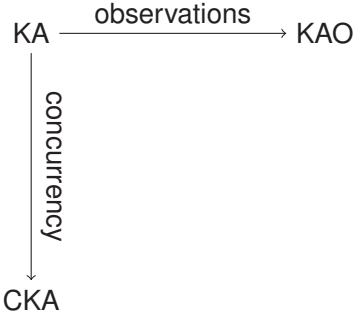
Since $\hat{e}$ and $\hat{f}$ are atomic and closed, we conclude

$$e \equiv \hat{e} \equiv \hat{f} \equiv f$$

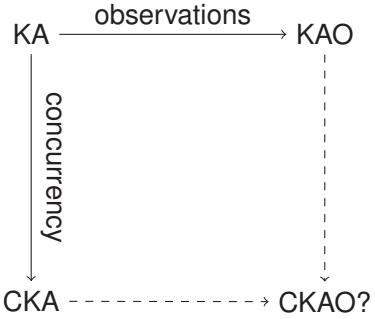
□

Future work

Future work



Future work



Future work

UCL

1. Build programming language (e.g. NetKAT) on top of this.

Future work

1. Build programming language (e.g. NetKAT) on top of this.
2. Isolation: some form of persistence of observations?

Future work

1. Build programming language (e.g. NetKAT) on top of this.
2. Isolation: some form of persistence of observations?
3. Optimise decision procedure; exploit transitive property?

Future work

1. Build programming language (e.g. NetKAT) on top of this.
2. Isolation: some form of persistence of observations?
3. Optimise decision procedure; exploit transitive property?
4. Possible applications in weak memory.

Thank you for your attention

https://coneco-project.org

Extended paper: https://arxiv.org/abs/1811.10401.

Icons (CC-BY) from The Noun Project (https://thenounproject.com)

- "Puzzle" by BlueTip design.
- "Scales" by jai.
- "The Scream" by Jonah Bethlehem.
- "Crossroads Decision" by Anna Sophie.
- "Biking" by tulpahn.
- "Chicken" by BomSymbols.

References I

- V. M. Antimirov [1996]. “Partial Derivatives of Regular Expressions and Finite Automaton Constructions”. In: *Theor. Comput. Sci.* 155.2, pp. 291–319. DOI: 10.1016/0304-3975(95)00182-4.
- J. A. Bergstra and A. Ponse [2011]. “Proposition algebra”. In: *ACM Trans. Comput. Log.* 12.3, 21:1–21:36. DOI: 10.1145/1929954.1929958.
- J. A. Brzozowski [1964]. “Derivatives of Regular Expressions”. In: *J. ACM* 11.4, pp. 481–494. DOI: 10.1145/321239.321249.
- T. Hoare et al. [2009]. “Concurrent Kleene Algebra”. In: *CONCUR*, pp. 399–414. DOI: 10.1007/978-3-642-04081-8_27.
- P. Jipsen [2014]. “Concurrent Kleene Algebra with Tests”. In: *RAMiCS*, pp. 37–48. DOI: 10.1007/978-3-319-06251-8_3.

References II

- P. Jipsen and M. A. Moshier [2016]. “Concurrent Kleene algebra with tests and branching automata”. In: *J. Log. Algebr. Meth. Program.* 85.4, pp. 637–652. DOI: 10.1016/j.jlamp.2015.12.005.
- T. Kappé et al. [2018]. “Concurrent Kleene Algebra: Free Model and Completeness”. In: *ESOP*, pp. 856–882. DOI: 10.1007/978-3-319-89884-1_30.
- 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”. In: *Inf. Comput.* 110.2, pp. 366–390. DOI: 10.1006/inco.1994.1037.
- — [1996]. “Kleene algebra with tests and commutativity conditions”. In: *TACAS*, pp. 14–33. DOI: 10.1007/3-540-61042-1_35.

References III

- D. Kozen and M. Patron [2000]. “Certification of Compiler Optimizations Using Kleene Algebra with Tests”. In: *CL*, pp. 568–582. DOI: 10.1007/3-540-44957-4_38.
- D. Kozen and F. Smith [1996]. “Kleene Algebra with Tests: Completeness and Decidability”. In: *CSL*, pp. 244–259. DOI: 10.1007/3-540-63172-0_43.
- M. R. Laurence and G. Struth [2014]. “Completeness Theorems for Bi-Kleene Algebras and Series-Parallel Rational Pomset Languages”. In: *RAMiCS*, pp. 65–82. DOI: 10.1007/978-3-319-06251-8_5.
- P. W. O’Hearn et al. [2015]. “On the relation between Concurrent Separation Logic and Concurrent Kleene Algebra”. In: *J. Log. Algebr. Meth. Program.* 84.3, pp. 285–302. DOI: 10.1016/j.jlamp.2014.08.002.
- A. Salomaa [1966]. “Two Complete Axiom Systems for the Algebra of Regular Events”. In: *J. ACM* 13.1, pp. 158–169. DOI: 10.1145/321312.321326.