RECENT DEVELOPMENTS IN CONCURRENT KLEENE ALGEBRA

JETBRAINS RESEARCH SEMINAR

October 2021

Paul Brunet
Université Paris-Est Créteil, EPISEN, & LACL

CONCURRENT KLEENE ALGEBRA

Concurrent Kleene Algebra

C.A.R. Tony Hoare[1], Bernhard Möller[2], Georg Struth[3], and Ian Wehrman[4]

[1] Microsoft Research, Cambridge, UK
[2] Universität Augsburg, Germany
[3] University of Sheffield, UK
[4] University of Texas at Austin, USA

CONCURRENT KLEENE ALGEBRA

On Locality and the Exchange Law for Concurrent Processes

C.A.R. Hoare[1], Akbar Hussain[2], Bernhard Möller[3], Peter W. O'Hearn[2], Rasmus Lerchedahl Petersen[2], and Georg Struth[4]

[1] Microsoft Research Cambridge
[2] Queen Mary University of London
[3] Universität Augsburg
[4] University of Sheffield

CONCURRENT KLEENE ALGEBRA

Completeness Theorems for Bi-Kleene Algebras and Series-Parallel Rational Pomset Languages

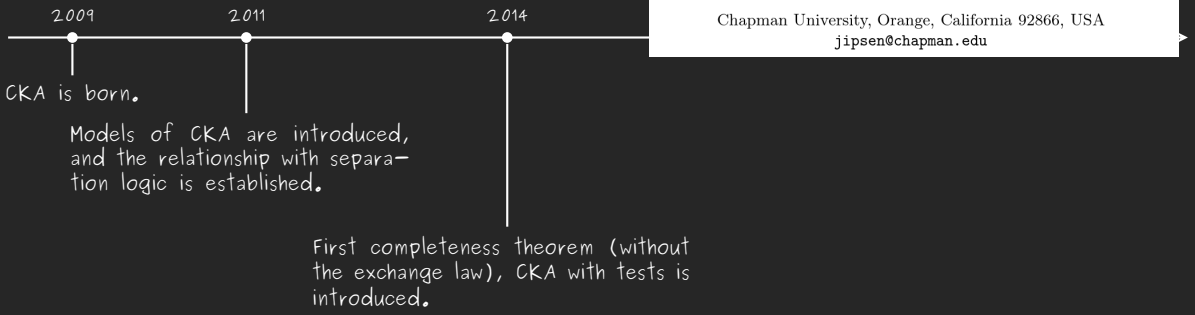
Michael R. Laurence and Georg Struth

Department of Computer Science, University of Sheffield, UK
{m.laurence,g.struth}@sheffield.ac.uk

Concurrent Kleene Algebra with Tests

Peter Jipsen

Chapman University, Orange, California 92866, USA
jipsen@chapman.edu

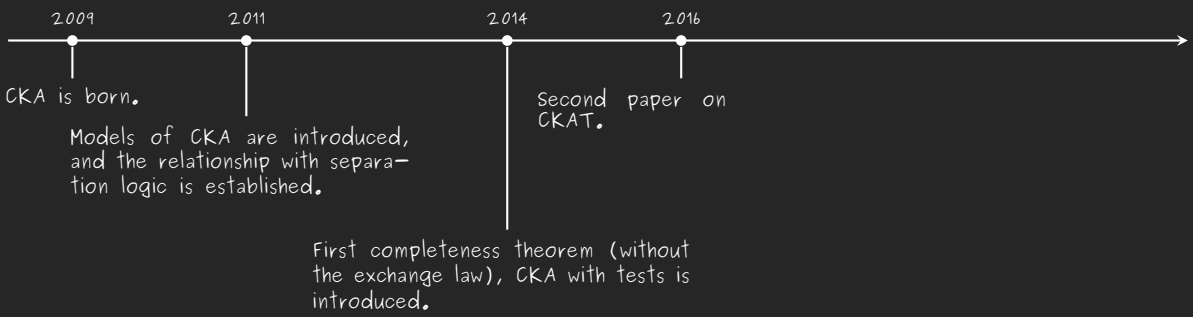


CONCURRENT KLEENE ALGEBRA

Concurrent Kleene algebra with tests and branching automata

Peter Jipsen*, M. Andrew Moshier

Chapman University, Orange, CA 92866, USA

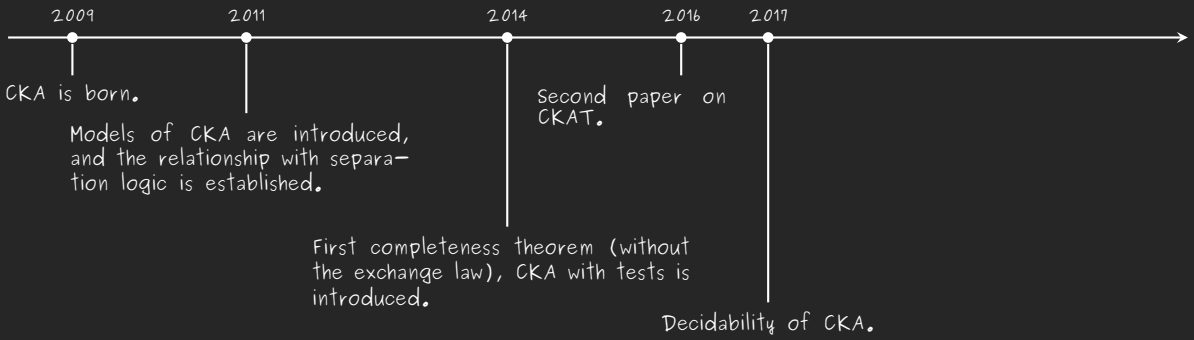


CONCURRENT KLEENE ALGEBRA

On Decidability of Concurrent Kleene Algebra[*†]

Paul Brunet[1], Damien Pous[2], and Georg Struth[3]

1 Univ. Lyon, CNRS, ENS de Lyon, UCB Lyon 1, LIP, France
2 Univ. Lyon, CNRS, ENS de Lyon, UCB Lyon 1, LIP, France
3 Department of Computer Science, The University of Sheffield, UK

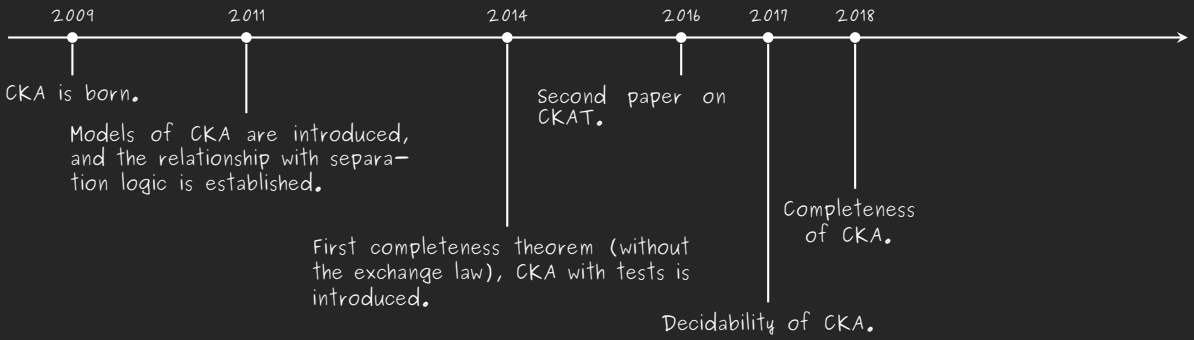


CONCURRENT KLEENE ALGEBRA

Concurrent Kleene Algebra: Free Model and Completeness

Tobias Kappé(✉), Paul Brunet, Alexandra Silva, and Fabio Zanasi

University College London, London, UK
tkappe@cs.ucl.ac.uk



CONCURRENT KLEENE ALGEBRA

Concurrent Kleene Algebra with Observations: from Hypotheses to Completeness

Tobias Kappé (✉), Paul Brunet, Alexandra Silva, Jana Wagemaker, and Fabio Zanasi

University College London, London, United Kingdom; tkappe@cs.ucl.ac.uk

Pomsets with Boxes: Protection, Separation, and Locality in Concurrent Kleene Algebra

Paul Brunet
University College London, UK
paul.brunet-zamansky.fr
paul@brunet-zamansky.fr

David Pym
University College London, UK
www.cantab.net/users/david.pym/
d.pym@ucl.ac.uk

Partially Observable Concurrent Kleene Algebra

Jana Wagemaker
Radboud University, Nijmegen
j.wagemaker@cs.ru.nl

Paul Brunet
University College London

Simon Docherty
University College London

Tobias Kappé
University College London

Jurriaan Rot
Radboud University, Nijmegen and University College London

Alexandra Silva
University College London

200 CKA is M a ti

2016 paper on

2017 without tests is Decidability of CKA.

2018 Completeness of CKA.

Observations boxes.

2020 (total/partial),

KLEENE ALGEBRA – THE ALGEBRA OF REGULAR EXPRESSIONS

$$e, f \in \mathbf{E}_A^{ka} ::= 0 \mid 1 \mid a \mid e \cdot f \mid e + f \mid e^*$$

Interpretation: regular languages

$$[\![\cdot]\!] : \mathbf{E}_A^{ka} \to \mathcal{P}(A^*)$$

example:

$$[\![a \cdot ((a + b) \cdot a)^*]\!] = \left\{ \begin{array}{l} \text{words of odd length over the} \\ \text{alphabet } \{a, b\} \text{ such that every} \\ \text{other letter is an a} \end{array} \right\}$$
$$= \{a, aaa, aba, aaaaa, abaaa, aaaba, ababa, \ldots\}$$

KLEENE ALGEBRA – THE ALGEBRA OF REGULAR EXPRESSIONS

The axioms of KA

$$e + e = e \qquad e + f = f + e \qquad e + (f + g) = (e + f) + g$$
$$e + 0 = 0 \qquad e \cdot 1 = e = 1 \cdot e \qquad e \cdot (f \cdot g) = (e \cdot f) \cdot g$$
$$e \cdot 0 = 0 = 0 \cdot e \qquad e \cdot (f + g) = e \cdot f + e \cdot g \qquad (e + f) \cdot g = e \cdot g + f \cdot g$$
$$e^* = 1 + e \cdot e^* \qquad e \cdot f \leq f \Rightarrow e^* \cdot f \leq f$$

Theorem

$$\mathsf{KA} \vdash e = f \Leftrightarrow [\![e]\!] = [\![f]\!].$$

Kozen, "A completeness theorem for Kleene algebras and the algebra of regular events", LiCS '90

KAT - THE ALGEBRA OF IMPERATIVE PROGRAMS

Syntax

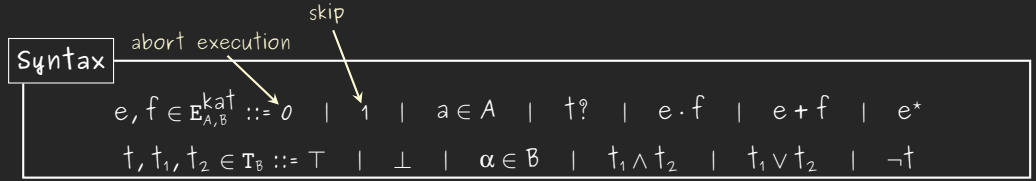
$$e, f \in \mathbf{E}^{\mathsf{kat}}_{A,B} ::= 0 \mid 1 \mid a \in A \mid t? \mid e \cdot f \mid e + f \mid e^*$$

$$t, t_1, t_2 \in \mathbf{T}_B ::= \top \mid \bot \mid \alpha \in B \mid t_1 \wedge t_2 \mid t_1 \vee t_2 \mid \neg t$$

Encodes a simple `While` language:

$$\texttt{if } b \texttt{ then } p \texttt{ else } q \mapsto b? \cdot p + (\neg b)? \cdot q \qquad \texttt{while } b \texttt{ do } p \mapsto (b? \cdot p)^* \cdot (\neg b)?$$

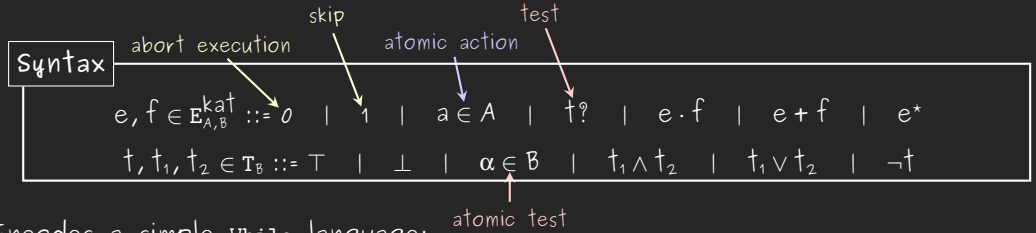
KAT - THE ALGEBRA OF IMPERATIVE PROGRAMS



Encodes a simple `While` language:

if b **then** p **else** $q \mapsto b? \cdot p + (\neg b)? \cdot q$ **while** b **do** $p \mapsto (b? \cdot p)^* \cdot (\neg b)?$

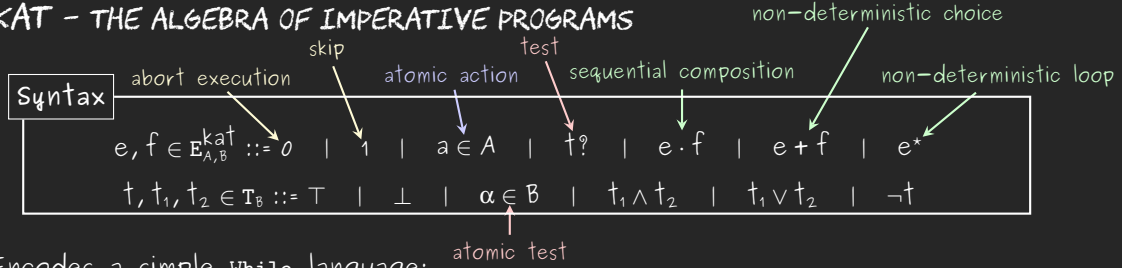
KAT - THE ALGEBRA OF IMPERATIVE PROGRAMS



Encodes a simple `While` language:

`if` b `then` p `else` $q \mapsto b? \cdot p + (\neg b)? \cdot q$ `while` b `do` $p \mapsto (b? \cdot p)^* \cdot (\neg b)?$

KAT - THE ALGEBRA OF IMPERATIVE PROGRAMS



Encodes a simple `While` language:

`if` b `then` p `else` $q \mapsto b? \cdot p + (\neg b)? \cdot q$ `while` b `do` $p \mapsto (b? \cdot p)^* \cdot (\neg b)?$

KAT - THE ALGEBRA OF IMPERATIVE PROGRAMS

Syntax

abort execution · skip · atomic action · test · sequential composition · non-deterministic choice · non-deterministic loop

$$e, f \in \mathrm{E}^{\mathrm{kat}}_{A,B} ::= 0 \mid 1 \mid a \in A \mid t? \mid e \cdot f \mid e + f \mid e^*$$

$$t, t_1, t_2 \in \mathrm{T}_B ::= \top \mid \bot \mid \alpha \in B \mid t_1 \wedge t_2 \mid t_1 \vee t_2 \mid \neg t$$

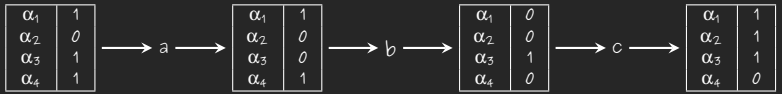
atomic test

Encodes a simple `While` language:

`if` b `then` p `else` $q \mapsto b? \cdot p + (\neg b)? \cdot q$ `while` b `do` $p \mapsto (b? \cdot p)^* \cdot (\neg b)?$

Interpretation: languages of guarded strings

guarded strings: alternating sequences of states $\in 2^B$ & actions $\in A$.



KAT: THE ALGEBRA OF IMPERATIVE PROGRAMS

The axioms of KAT:

- ☞ The axioms of KA.
- ☞ For tests, the axioms of boolean algebra.
- ☞ The following "glue" axioms:

$$(t_1 \vee t_2)? = t_1? + t_2? \qquad (t_1 \wedge t_2)? = t_1? \cdot t_2? \qquad \top? = 1 \qquad \bot? = 0$$

Theorem

$$\mathsf{KAT} \vdash e = f \Leftrightarrow [\![e]\!] = [\![f]\!].$$

Kozen & Smith, "Kleene algebra with tests: Completeness and decidability", CSL '96

KAT: THE ALGEBRA OF IMPERATIVE PROGRAMS

The axioms of KAT:

- ☞ The axioms of KA.
- ☞ For tests, the axioms of boolean algebra.
- ☞ The following "glue" axioms:

$$(t_1 \vee t_2)? = t_1? + t_2? \qquad (t_1 \wedge t_2)? = t_1? \cdot t_2? \qquad \top? = 1 \qquad \bot? = 0$$

Theorem

$$\mathsf{KAT} \vdash e = f \Leftrightarrow [\![e]\!] = [\![f]\!].$$

Kozen & Smith, "Kleene algebra with tests: Completeness and decidability", CSL '96

Subsumes Hoare logic:

$$\begin{aligned} \{b\}\, p\, \{c\} &\Leftrightarrow b? \cdot p \leq p \cdot c? \\ &\Leftrightarrow b? \cdot p = b? \cdot p \cdot c? \\ &\Leftrightarrow b? \cdot p \cdot (\neg c)? = 0 \end{aligned}$$

KAT: THE ALGEBRA OF IMPERATIVE PROGRAMS

The axioms of KAT:

- ☞ The axioms of KA.
- ☞ For tests, the axioms of boolean algebra.
- ☞ The following "glue" axioms:

$$(t_1 \vee t_2)? = t_1? + t_2? \qquad (t_1 \wedge t_2)? = t_1? \cdot t_2? \qquad \top? = 1 \qquad \bot? = 0$$

Theorem

$$\mathsf{KAT} \vdash e = f \Leftrightarrow [\![e]\!] = [\![f]\!].$$

Kozen & Smith, "Kleene algebra with tests: Completeness and decidability", CSL '96

Subsumes Hoare logic:

$$\begin{aligned} \{b\}\, p\, \{c\} &\Leftrightarrow b? \cdot p \leq p \cdot c? \\ &\Leftrightarrow b? \cdot p = b? \cdot p \cdot c? \\ &\Leftrightarrow b? \cdot p \cdot (\neg c)? = 0 \end{aligned}$$

Can we do the same for concurrent programs?

RECENT DEVELOPMENTS IN CKA
Outline

I. Concurrent Kleene Algebra

II. CKA with observations

III. Partially observable CKA

IV. CKA with boxes

V. Ongoing and future work

RECENT DEVELOPMENTS IN CKA
Outline

I. Concurrent Kleene Algebra

II. CKA with observations

III. Partially observable CKA

IV. CKA with boxes

V. Ongoing and future work

BI-KLEENE ALGEBRA

$$e, f \in \mathbf{E}_A^{\mathsf{bika}} ::= 1 \mid 0 \mid x \mid e \cdot f \mid e \parallel f \mid e + f \mid e^\star \mid e^!$$

Definition

A bi-Kleene algebra is a structure $\langle \mathscr{A}, 0, 1, \cdot, \parallel, +, \star, ! \rangle$ such that:

☞ $\langle \mathscr{A}, 0, 1, \cdot, +, \star \rangle$ is a KA

☞ $\langle \mathscr{A}, 0, 1, \parallel, +, ! \rangle$ is a commutative KA.

BI-KLEENE ALGEBRA

$$e, f \in \mathbf{E}_A^{\mathsf{bika}} ::= 1 \mid 0 \mid x \mid e \cdot f \mid e \parallel f \mid e + f \mid e^\star \mid e^!$$

Definition

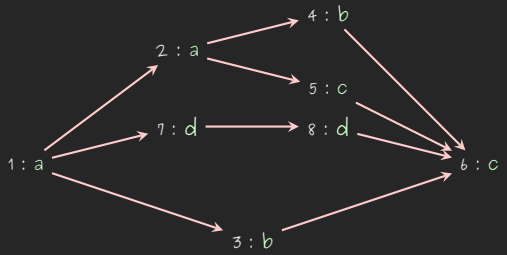
A bi-Kleene algebra is a structure $\langle \mathscr{A}, 0, 1, \cdot, \parallel, +, \star, ! \rangle$ such that:

☞ $\langle \mathscr{A}, 0, 1, \cdot, +, \star \rangle$ is a KA

☞ $\langle \mathscr{A}, 0, 1, \parallel, +, ! \rangle$ is a commutative KA.

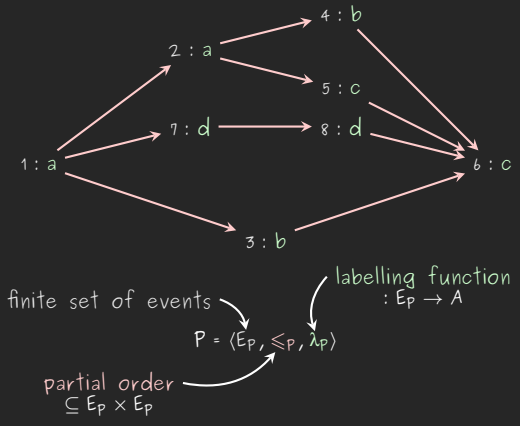
What is the free bi-KA?

POMSETS - CONCURRENT TRACES



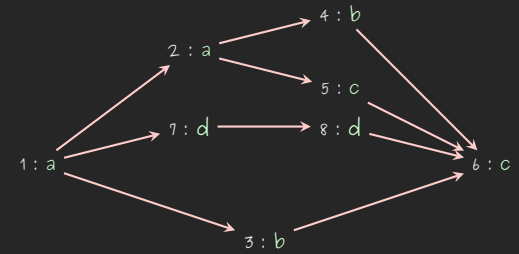
POMSETS – CONCURRENT TRACES

A is some alphabet of actions.



POMSETS – CONCURRENT TRACES

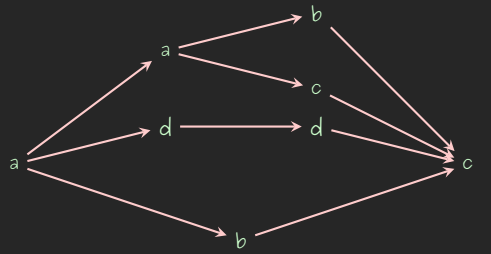
A is some alphabet of actions.



Up-to isomorphism $\equiv$.

POMSETS – CONCURRENT TRACES

A is some alphabet of actions.



Up-to isomorphism $\equiv$.

COMBINING POMSETS

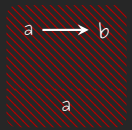
a = 

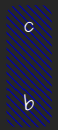
1 = 

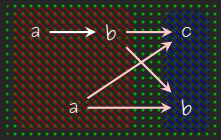
COMBINING POMSETS

$a =$ 

$1 =$ 

$P_1 =$ 

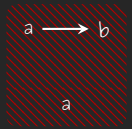
$P_2 =$ 

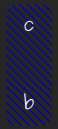
$P_1 ; P_2 =$ 

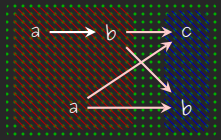
COMBINING POMSETS

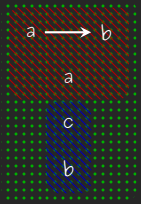
$a =$ 

$1 =$ 

$P_1 =$ 

$P_2 =$ 

$P_1 ; P_2 =$ 

$P_1 \parallel P_2 =$ 

COMPLETENESS OF BIKA

$$[\![1]\!] := \{1\} \qquad [\![0]\!] := \emptyset$$

$$[\![a]\!] := \{a\} \qquad [\![e + f]\!] := [\![e]\!] \cup [\![f]\!]$$

$$[\![e \cdot f]\!] := \left\{ P ; Q \;\middle|\; P \in [\![e]\!], Q \in [\![f]\!] \right\} \qquad [\![e \parallel f]\!] := \left\{ P \parallel Q \;\middle|\; P \in [\![e]\!], Q \in [\![f]\!] \right\}$$

$$[\![e^*]\!] := \{P_1 ; \cdots ; P_n \mid n \in \mathbb{N}, P_i \in [\![e]\!]\} \qquad [\![e^\prime]\!] := \{P_1 \parallel \cdots \parallel P_n \mid n \in \mathbb{N}, P_i \in [\![e]\!]\}$$

Theorem

$$\mathsf{biKA} \vdash e = f \Leftrightarrow [\![e]\!] \equiv [\![f]\!].$$

Laurence & Struth, "Completeness Theorems for Bi-Kleene Algebras and Series-Parallel Rational Pomset Languages", RAMiCS '14

INTERLEAVINGS AND SUBSUMPTION

Interchange law

$$(a \parallel b) \cdot (c \parallel d) \leq (a \cdot c) \parallel (b \cdot d).$$

$P \sqsubseteq Q$ when there is a homomorphism from Q to P, i.e. a bijective map $\varphi : E_Q \to E_P$ such that $\lambda_P \circ \varphi = \lambda_Q$ and $\varphi(\leq_Q) \subseteq \leq_P$.

$L^{\sqsubseteq} := \{P \mid \exists Q \in L : P \sqsubseteq Q\}$.

CONCURRENT KLEENE ALGEBRA

Interchange law

$$(a \parallel b) \cdot (c \parallel d) \leq (a \cdot c) \parallel (b \cdot d).$$

No parallel iteration

CKA

A concurrent Kleene algebra is a weak bi-Kleene algebra $\langle \mathcal{A}, 0, 1, \cdot, \parallel, +, \star \rangle$ satisfying the interchange law.

COMPLETENESS AND DECIDABILITY OF CKA

Theorem

The problem of testing whether two given expressions e, f denote the same closed language is ExpSpace-complete.

B., Pous, & Struth, "On Decidability of Concurrent Kleene Algebra", CONCUR '17

Theorem

$$\mathsf{CKA} \vdash e = f \Leftrightarrow [\![e]\!]^{\sqsubseteq} = [\![f]\!]^{\sqsubseteq}.$$

Kappé, B., Silva, & Zanasi, "Concurrent Kleene Algebra: Free Model and Completeness", ESOP '18

RECENT DEVELOPMENTS IN CKA
Outline

I. Concurrent Kleene Algebra


II. CKA with observations

III. Partially observable CKA

IV. CKA with boxes

V. Ongoing and future work

CONCURRENT KLEENE ALGEBRA WITH TESTS

Slogan

☞ KAT: KA with a boolean sub-algebra.

☞ CKAT: CKA with a boolean sub-algebra.

CONCURRENT KLEENE ALGEBRA WITH TESTS

Slogan

☞ KAT: KA with a boolean sub-algebra.

☞ CKAT: CKA with a boolean sub-algebra.

$$
\begin{aligned}
t? \cdot p \cdot (\neg t)? &\leq p \parallel (t? \cdot (\neg t)?) && \text{CKA axioms} \\
&= p \parallel (t \wedge \neg t)? && (a \wedge b)? = a? \cdot b? \\
&= p \parallel \bot? && \text{boolean axioms} \\
&= p \parallel 0 && \bot? = 0 \\
&= 0 && \text{CKA axioms}
\end{aligned}
$$

CONCURRENT KLEENE ALGEBRA WITH TESTS

Slogan

☞ KAT: KA with a boolean sub-algebra.

☞ CKAT: CKA with a boolean sub-algebra.

$$
\begin{aligned}
t? \cdot p \cdot (\neg t)? &\leq p \parallel (t? \cdot (\neg t)?) && \text{CKA axioms} \\
&= p \parallel (t \wedge \neg t)? && (a \wedge b)? = a? \cdot b? \\
&= p \parallel \bot? && \text{boolean axioms} \\
&= p \parallel 0 && \bot? = 0 \\
&= 0 && \text{CKA axioms}
\end{aligned}
$$

↔ For every program and every assertion, the triple $\{t\}\, p\, \{t\}$ holds.

↔ Every test is invariant under every program.

~~CONCURRENT KLEENE ALGEBRA WITH TESTS~~

slogan

☞ KAT: KA with a boolean sub-algebra.

☞ CKAT: CKA with a boolean sub-algebra.

$$
\begin{aligned}
t? \cdot p \cdot (\neg t)? &\leq p \parallel (t? \cdot (\neg t)?) && \text{CKA axioms} \\
&= p \parallel (t \wedge \neg t)? && (a \wedge b)? = a? \cdot b? \\
&= p \parallel \bot? && \text{boolean axioms} \\
&= p \parallel 0 && \bot? = 0 \\
&= 0 && \text{CKA axioms}
\end{aligned}
$$

↔ For every program and every assertion, the triple $\{t\}\,p\,\{t\}$ holds.

↔ Every test is invariant under every program.

WHO'S TO BLAME?

$$
\begin{aligned}
t \cdot p \cdot \neg t &\leq p \parallel \left(t \cdot \neg t\right) && \text{CKA axioms} \\
&= p \parallel \left(t \wedge \neg t\right) && (a \wedge b)? = a? \cdot b? \\
&= p \parallel \bot && \text{Boolean axioms} \\
&= p \parallel 0 = 0 && \bot? = 0 + \text{CKA axioms}
\end{aligned}
$$

WHO'S TO BLAME?

$$
\begin{aligned}
t \cdot p \cdot \neg t &\leq p \parallel \left(t \cdot \neg t\right) \\
&= p \parallel \left(t \wedge \neg t\right) \\
&= p \parallel \bot \\
&= p \parallel 0 = 0
\end{aligned}
$$

CKA axioms

$$(a \wedge b)? = a? \cdot b?$$

Boolean axioms

$\bot? = 0$ + CKA axioms

WHO'S TO BLAME?

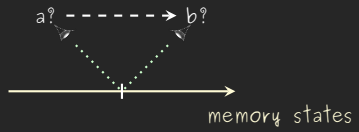
$$
\begin{aligned}
t \cdot p \cdot \neg t &\leq p \parallel \left(t \cdot \neg t\right) \\
&= p \parallel \left(t \wedge \neg t\right) \\
&= p \parallel \bot \\
&= p \parallel 0 = 0
\end{aligned}
$$

CKA axioms

$$(a \wedge b)? = a? \cdot b?$$

Boolean axioms

$\bot? = 0$ + CKA axioms



WHO'S TO BLAME?

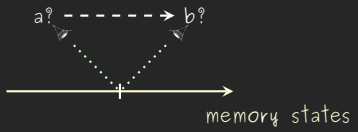
$$
\begin{aligned}
t \cdot p \cdot \neg t &\leq p \parallel \left(t \cdot \neg t\right) \\
&= p \parallel \left(t \wedge \neg t\right) \\
&= p \parallel \bot \\
&= p \parallel 0 = 0
\end{aligned}
$$

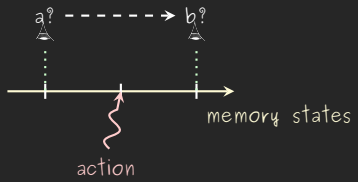
CKA axioms

$$(a \wedge b)? = a? \cdot b?$$

Boolean axioms

$\bot? = 0$ + CKA axioms





WHO'S TO BLAME?

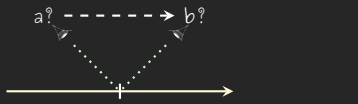
$$
\begin{aligned}
t \cdot p \cdot \neg t &\leq p \parallel \left(t \cdot \neg t\right) \\
&= p \parallel \left(t \wedge \neg t\right) \\
&= p \parallel \bot \\
&= p \parallel 0 = 0
\end{aligned}
$$

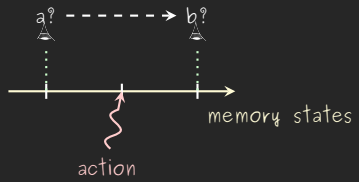
CKA axioms

~~$(a \wedge b)? = a? \cdot b?$~~

Boolean axioms

$\bot? = 0$ + CKA axioms





$$(a \wedge b)? \leq a? \cdot b?$$

CKA WITH OBSERVATIONS – SYNTAX

$$e, f \in \mathrm{E}^{\mathrm{ckao}}_{A,B} ::= 0 \mid 1 \mid a \in A \mid t? \mid e \cdot f \mid e \parallel f \mid e + f \mid e^*$$

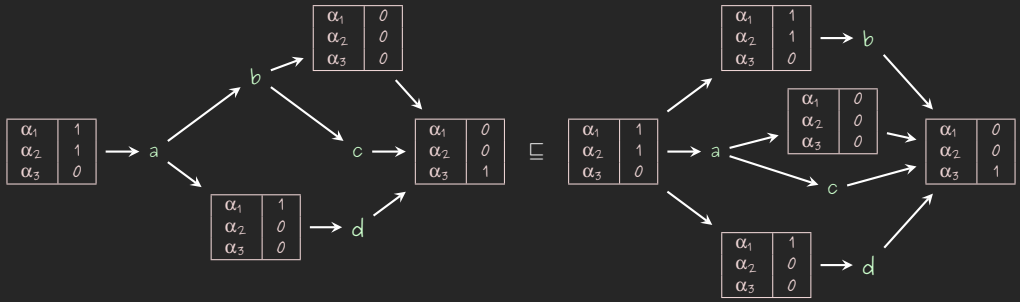
$$t, t_1, t_2 \in \mathrm{O}^{\mathrm{bool}}_{B} ::= \top \mid \bot \mid \alpha \in B \mid t_1 \wedge t_2 \mid t_1 \vee t_2 \mid \neg t$$

The axioms of CKAO

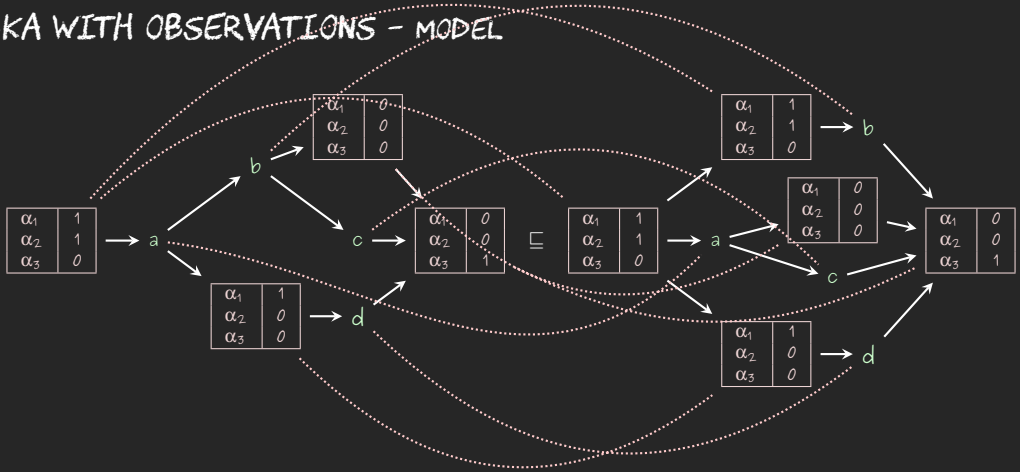
- The axioms of CKA.
- For tests, the axioms of boolean algebra.
- The following "glue" axioms:

$$(t_1 \vee t_2)? = t_1? + t_2? \qquad (t_1 \wedge t_2)? \leq t_1? \cdot t_2? \qquad \bot? = 0$$

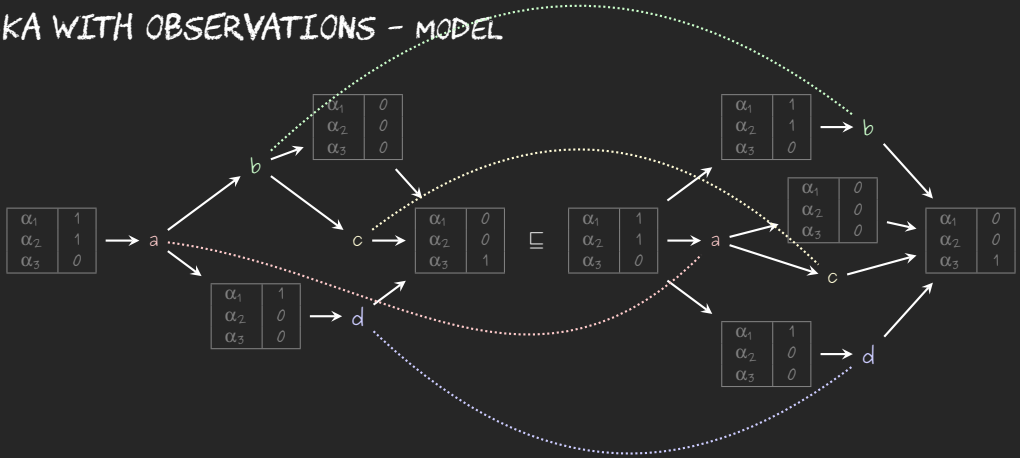
CKA WITH OBSERVATIONS – MODEL



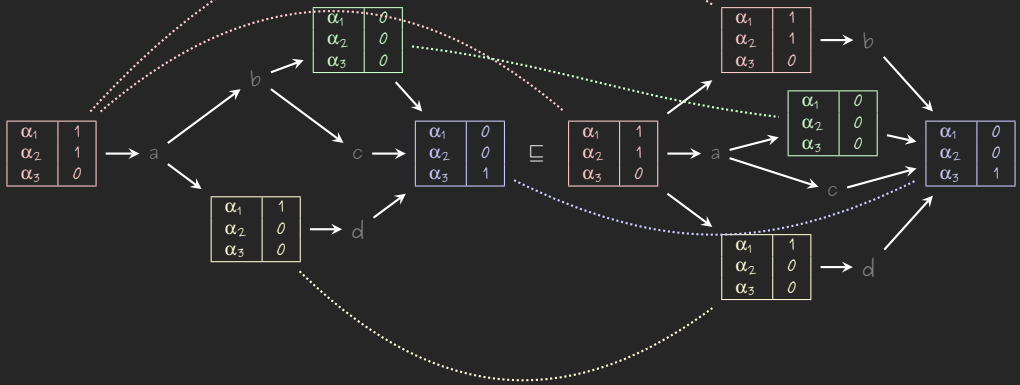
CKA WITH OBSERVATIONS – MODEL



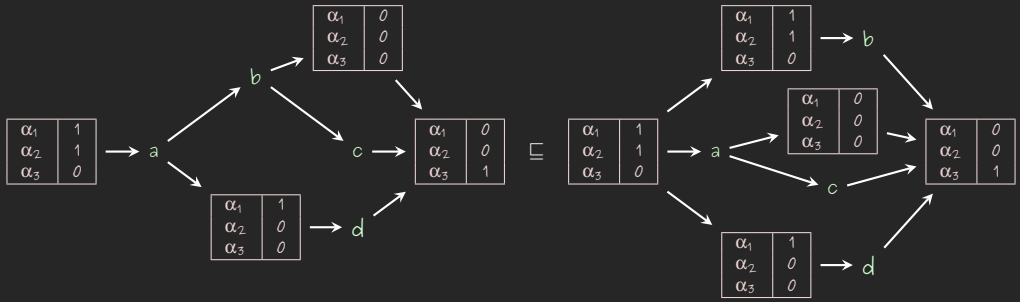
CKA WITH OBSERVATIONS – MODEL



CKA WITH OBSERVATIONS – MODEL



CKA WITH OBSERVATIONS – MODEL



Theorem

$$\mathsf{CKAO} \vdash e = f \Leftrightarrow [\![e]\!]\downarrow = [\![f]\!]\downarrow.$$

Kappé, B., Silva, Wagemaker, & Zanasi, "Concurrent Kleene Algebra with Observations: from Hypotheses to Completeness", FoSSaCS '20

INTERLUDE – (C)KA WITH HYPOTHESES

H: set of hypotheses $e \leq f$ over some fixed alphabet A.

- ☞ extra structure on the alphabet (e.g. $\alpha \wedge \beta = \beta \wedge \alpha$);
- ☞ extra structure on traces (e.g. $\alpha \leq \alpha \cdot \alpha$)
- ☞ other domain-specific assumptions.

Theorem

$$\mathrm{CKA} + \mathrm{H} \vdash e = f \Rightarrow [\![e]\!]\downarrow^{\mathrm{H}} = [\![f]\!]\downarrow^{\mathrm{H}}$$

- ☞ Doumane, Kuperberg, Pous, & Pradic, "Kleene Algebra with Hypotheses", FoSSaCS '19
- ☞ Kappé, B., Silva, Wagemaker, & Zanasi, "Concurrent Kleene Algebra with Observations: from Hypotheses to Completeness", FoSSaCS '20

RECENT DEVELOPMENTS IN CKA
Outline

I. Concurrent Kleene Algebra

II. CKA with observations

III. Partially observable CKA

IV. CKA with boxes

V. Ongoing and future work

LITMUS TEST: SEQUENTIAL CONSISTENCY

```
{ r0 == 0 && r1 == 0 }

 x := 1   ||   y := 1
 r0 := y  ||   r1 := x

{ !( r0 == 1 || r1 == 1 ) }
```

Ingredients:

- ☞ Assignments $x \leftarrow 1$
- ☞ Observations $r_0 \mapsto 0$

WHAT KIND OF OBSERVATIONS DO WE NEED?
First attempt: boolean algebra

☞ Atomic observations: $V_{AR} \mapsto V_{AL}$ e.g. $r_0 \mapsto 1$

WHAT KIND OF OBSERVATIONS DO WE NEED?
First attempt: boolean algebra

☞ Atomic observations: $V_{AR} \mapsto V_{AL}$ e.g. $r_0 \mapsto 1$

☞ Boolean formula: set of memory states $V_{AR} \to V_{AL}$ e.g.

r_0	1
r_1	0

WHAT KIND OF OBSERVATIONS DO WE NEED?
First attempt: boolean algebra

- Atomic observations: $\mathrm{VAR} \mapsto \mathrm{VAL}$ e.g. $r_0 \mapsto 1$
- Boolean formula: set of memory states $\mathrm{VAR} \to \mathrm{VAL}$ e.g. $\begin{array}{|c|c|}\hline r_0 & 1 \\ \hline r_1 & 0 \\ \hline \end{array}$
- Assignments: $\sum_{s \in \mathrm{State}} s \cdot (v \leftarrow n) \cdot s[v \mapsto n]$, i.e.

$$[\![x \leftarrow 1]\!] := \left\{ \begin{array}{|c|c|}\hline x & 0 \\ \hline y & 0 \\ \hline \end{array} \to [x \leftarrow 1] \to \begin{array}{|c|c|}\hline x & 1 \\ \hline y & 0 \\ \hline \end{array}, \begin{array}{|c|c|}\hline x & 0 \\ \hline y & 1 \\ \hline \end{array} \to [x \leftarrow 1] \to \begin{array}{|c|c|}\hline x & 1 \\ \hline y & 1 \\ \hline \end{array} \right\}$$

WHAT KIND OF OBSERVATIONS DO WE NEED?
First attempt: boolean algebra

- Atomic observations: $\mathrm{VAR} \mapsto \mathrm{VAL}$ — e.g. $r_0 \mapsto 1$
- Boolean formula: set of memory states $\mathrm{VAR} \to \mathrm{VAL}$ — e.g. $\begin{array}{|c|c|}\hline r_0 & 1 \\ \hline r_1 & 0 \\ \hline \end{array}$
- Assignments: $\sum_{s \in \mathrm{State}} s \cdot (v \leftarrow n) \cdot s[v \mapsto n]$, i.e.

$$[\![x \leftarrow 1]\!] := \left\{ \begin{array}{|c|c|}\hline x & 0 \\ \hline y & 0 \\ \hline \end{array} \to [x \leftarrow 1] \to \begin{array}{|c|c|}\hline x & 1 \\ \hline y & 0 \\ \hline \end{array} \;,\; \begin{array}{|c|c|}\hline x & 0 \\ \hline y & 1 \\ \hline \end{array} \to [x \leftarrow 1] \to \begin{array}{|c|c|}\hline x & 1 \\ \hline y & 1 \\ \hline \end{array} \right\}$$

Problem: parallel composition?

$$\begin{array}{|c|c|}\hline x & 0 \\ \hline y & 0 \\ \hline \end{array} \to [x \leftarrow 1] \to \begin{array}{|c|c|}\hline x & 1 \\ \hline y & 0 \\ \hline \end{array} \qquad \begin{array}{|c|c|}\hline x & 0 \\ \hline y & 0 \\ \hline \end{array} \to [y \leftarrow 1] \to \begin{array}{|c|c|}\hline x & 0 \\ \hline y & 1 \\ \hline \end{array}$$

ALGEBRA OF PARTIAL OBSERVATIONS

Idea: Instead of memory states $\mathrm{VAR} \to \mathrm{VAL}$, consider partial functions $\mathrm{VAR} \rightharpoonup \mathrm{VAL}$.

PCDL of observations

$$t, t_1, t_2 \in \mathbb{O}_B^{\text{pocka}} ::= \top \mid \bot \mid \alpha \in B \mid t_1 \wedge t_2 \mid t_1 \vee t_2 \mid \overline{t}$$

Same axioms as BA regarding $\vee, \wedge, \top, \bot$, plus:

- ☞ $p \leq \overline{q} \Leftrightarrow p \wedge q = \bot$
- ☞ $\overline{v \mapsto n} = \bigvee_{m \neq n} v \mapsto m$

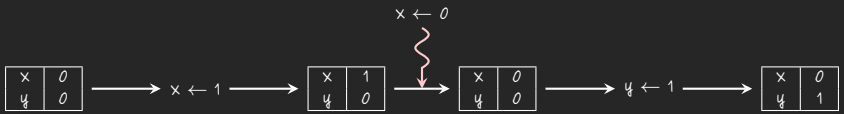
PCDL : pseudo-complemented distributive lattice

Theoreme

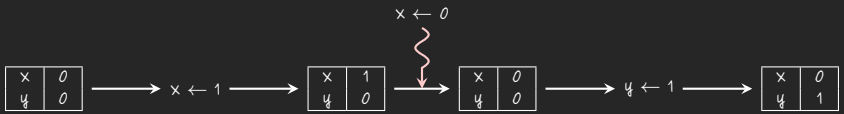
$$\text{POCKA} \vdash e = f \Leftrightarrow [\![e]\!]\downarrow^{\text{pocka}} = [\![f]\!]\downarrow^{\text{pocka}}.$$

Wagemaker, B., Docherty, Kappé, Rot, & Silva, "Partially Observable Concurrent Kleene Algebra", CONCUR '20

CAUSALITY VS COMPOSITIONALITY



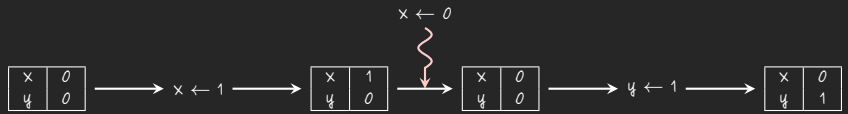
CAUSALITY VS COMPOSITIONALITY



Solution: we need to explicitly close the system.

$$[\![e]\!] \quad \rightarrow \quad [\![e]\!] \cap \text{CausalPomsets}.$$

CAUSALITY VS COMPOSITIONALITY



Solution: we need to explicitly close the system.

$$[\![e]\!] \quad \rightarrow \quad [\![e]\!] \cap \mathsf{CausalPomsets}.$$

Litmus test:

$$t := (r_0 = 0 \wedge r_1 = 0)? \cdot ((x \leftarrow 1 \cdot r_0 \leftarrow y) \parallel (y \leftarrow 1 \cdot r_1 \leftarrow x)) \cdot \overline{(r_0 = 1 \vee r_1 \vee 1)}?$$

$$[\![t]\!] \cap \mathsf{CausalPomsets} = \emptyset$$

RECENT DEVELOPMENTS IN CKA
Outline

I. Concurrent Kleene Algebra

II. CKA with observations

III. Partially observable CKA


IV. CKA with boxes

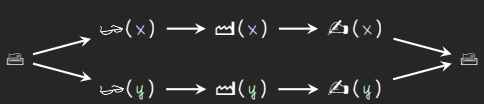
V. Ongoing and future work

MUTUAL EXCLUSION

```
      print(counter);

x:=counter;  ||  y:=counter;
x:=x+1;      ||  y:=y+1;
counter:=x;  ||  counter:=y;

      print(counter);
```

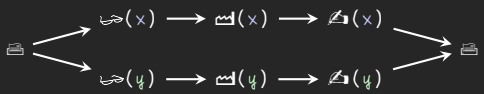


MUTUAL EXCLUSION

```
      print(counter);

x:=counter;  ||  y:=counter;
x:=x+1;      ||  y:=y+1;
counter:=x;  ||  counter:=y;

      print(counter);
```

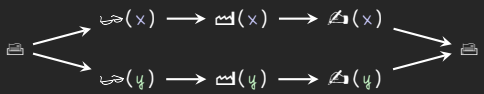


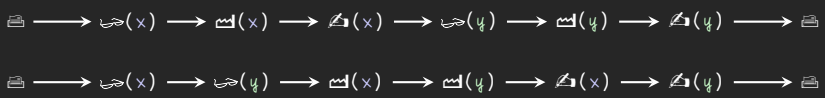
MUTUAL EXCLUSION

```
      print(counter);

x:=counter;  ||  y:=counter;
x:=x+1;      ||  y:=y+1;
counter:=x;  ||  counter:=y;

      print(counter);
```

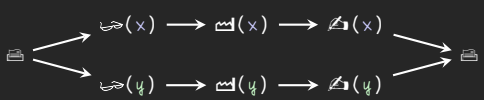


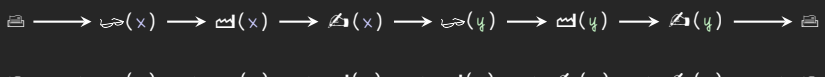


Paul Brunet

MUTUAL EXCLUSION

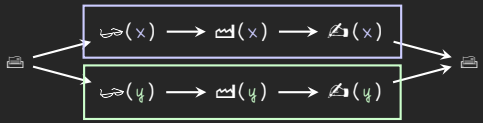
```
          print(counter);
atomic{                  ||  atomic{
    x:=counter;          ||      y:=counter;
    x:=x+1;              ||      y:=y+1;
    counter:=x;          ||      counter:=y;
}                        ||  }
          print(counter);
```





MUTUAL EXCLUSION

```
          print(counter);
atomic{           ||  atomic{
    x:=counter;   ||      y:=counter;
    x:=x+1;       ||      y:=y+1;
    counter:=x;   ||      counter:=y;
}                 ||  }
          print(counter);
```

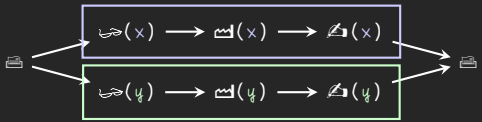


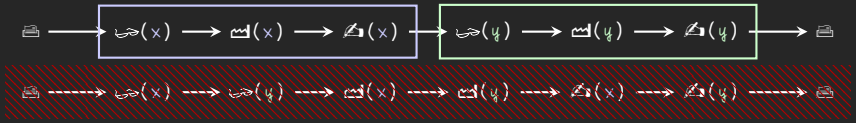
🖴 ⟶ 🖮(x) ⟶ 🖬(x) ⟶ 🖉(x) ⟶ 🖮(y) ⟶ 🖬(y) ⟶ 🖉(y) ⟶ 🖴

🖴 ⟶ 🖮(x) ⟶ 🖮(y) ⟶ 🖬(x) ⟶ 🖬(y) ⟶ 🖉(x) ⟶ 🖉(y) ⟶ 🖴

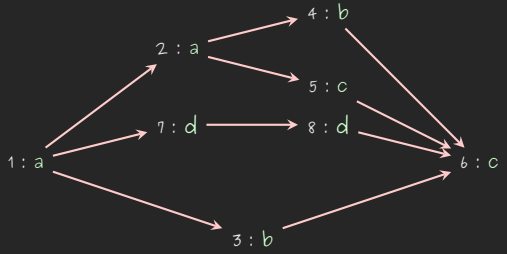
MUTUAL EXCLUSION

```
          print(counter);
atomic{                  ||  atomic{
    x:=counter;          ||      y:=counter;
    x:=x+1;              ||      y:=y+1;
    counter:=x;          ||      counter:=y;
}                        ||  }
          print(counter);
```





POMSETS WITH BOXES



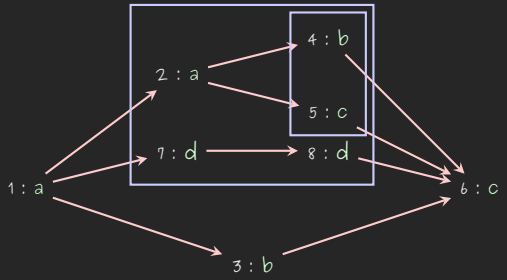
finite set of events

labeling function
$: E_P \to A$

$P = \langle E_P, \leqslant_P, \lambda_P \rangle$

partial order
$\subseteq E_P \times E_P$

POMSETS WITH BOXES



finite set of events

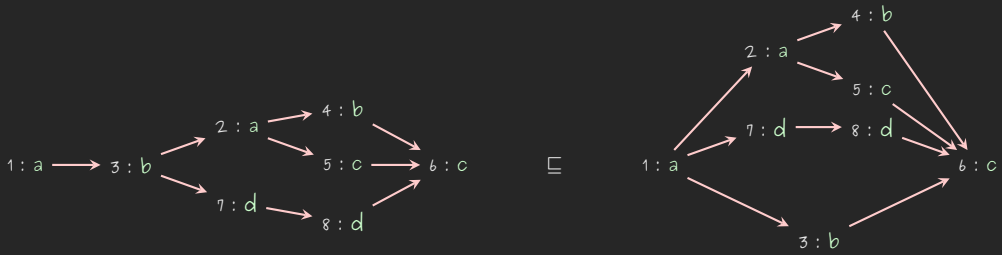
labeling function
$: E_P \to A$

$P = \langle E_P, \leqslant_P, \lambda_P, B_P \rangle$

partial order
$\subseteq E_P \times E_P$

set of boxes
$\subseteq \mathcal{P}^{\geq 0}(E_P)$

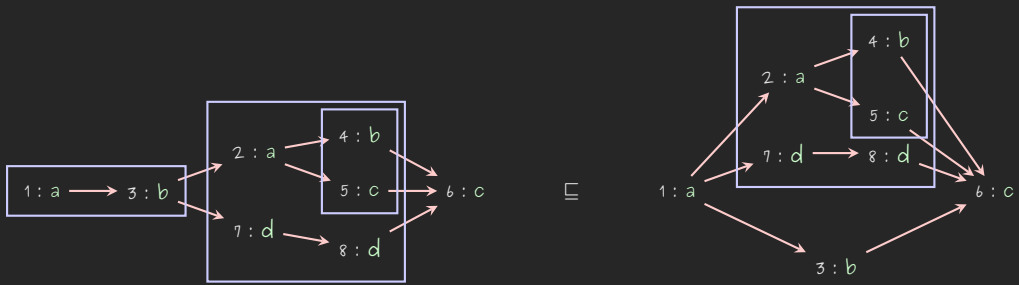
SUBSUMPTION WITH BOXES



$P \sqsubseteq Q$ when there is a homomorphism from Q to P, i.e. a bijective map $\varphi : E_Q \to E_P$ such that

1) $\lambda_P \circ \varphi = \lambda_Q$
2) $\varphi(\leq_Q) \subseteq \leq_P$

SUBSUMPTION WITH BOXES



$P \sqsubseteq Q$ when there is a homomorphism from Q to P, i.e. a bijective map $\varphi : E_Q \to E_P$ such that

1) $\lambda_P \circ \varphi = \lambda_Q$
2) $\varphi(\leq_Q) \subseteq \leq_P$
3) $\varphi(\mathcal{B}_P) \subseteq \mathcal{B}_Q$

AXIOMATISATION

(We remove the Kleene star from the signature)

$$[[e]] = [e]$$
$$[1] = 1$$
$$[0] = 0$$
$$[e + f] = [e] + [f]$$
$$[e] \leq e$$

Concurrent semi-ring, i.e. CKA w/o star

Theorem

$$\mathrm{CSR} + B \vdash e = f \quad \Leftrightarrow \quad [\![e]\!] = [\![f]\!].$$

B. & Pym, "Pomsets with Boxes: Protection, Separation, and Locality in Concurrent Kleene Algebra.", FSCD '20

AXIOMATISATION

(We remove the Kleene star from the signature)

$$[[e]] = [e]$$
$$[1] = 1$$
$$[0] = 0$$
$$[e + f] = [e] + [f]$$
$$[e] \leq e$$

Concurrent semi-ring, i.e. CKA w/o star

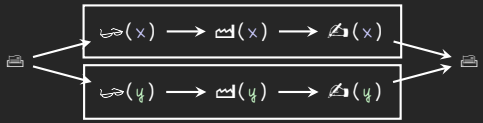
Theorem

$$\mathsf{CSR} + B \vdash e = f \quad \Leftrightarrow \quad [\![e]\!] = [\![f]\!].$$

B. & Pym, "Pomsets with Boxes: Protection, Separation, and Locality in Concurrent Kleene Algebra.", FSCD '20

MUTUAL EXCLUSION (II)

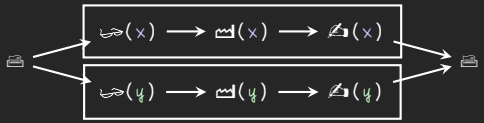
```
          print(counter);
atomic{           ||  atomic{
    x:=counter;   ||      y:=counter;
    x:=x+1;       ||      y:=y+1;
    counter:=x;   ||      counter:=y;
}                 ||  }
          print(counter);
```



$$P := 🖷 \cdot \left[🖷(x) \cdot 🖬(x) \cdot ✍(x)\right] \parallel \left[🖷(y) \cdot 🖬(y) \cdot ✍(y)\right] \cdot 🖷$$

MUTUAL EXCLUSION (II)

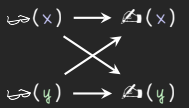
```
          print(counter);
atomic{           ||  atomic{
    x:=counter;   ||      y:=counter;
    x:=x+1;       ||      y:=y+1;
    counter:=x;   ||      counter:=y;
}                 ||  }
          print(counter);
```



$$P := 🖶 \cdot [🖳(x) \cdot 🖫(x) \cdot 🖉(x)] \parallel [🖳(y) \cdot 🖫(y) \cdot 🖉(y)] \cdot 🖶$$

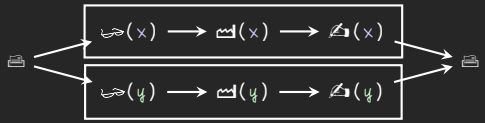
Breaking mutual exclusion

$\leftrightarrow$ admitting an execution with the "pattern":



MUTUAL EXCLUSION (II)

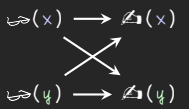
```
          print(counter);
atomic{           ||  atomic{
    x:=counter;   ||      y:=counter;
    x:=x+1;       ||      y:=y+1;
    counter:=x;   ||      counter:=y;
}                 ||  }
          print(counter);
```



$$P := 🖶 \cdot \left[🡒(x) \cdot 🖫(x) \cdot ✍(x)\right] \parallel \left[🡒(y) \cdot 🖫(y) \cdot ✍(y)\right] \cdot 🖶$$

Breaking mutual exclusion

$\leftrightarrow$ admitting an execution with the "pattern":



Problem: this property cannot be stated in (in)equational form, e.g.

program $\leq$ specification.

POMSET LOGIC*

$$\varphi, \psi ::= \bot \mid a \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \blacktriangleright \psi \mid \varphi \star \psi \mid [\varphi] \mid (\!|\varphi|\!)$$

- $P \models \varphi \blacktriangleright \psi$ iff $\exists P_1, P_2$ such that $P \sqsupseteq P_1 ; P_2$ and $P_1 \models \varphi$ and $P_2 \models \psi$
- $P \models \varphi \star \psi$ iff $\exists P_1, P_2$ such that $P \sqsupseteq P_1 \parallel P_2$ and $P_1 \models \varphi$ and $P_2 \models \psi$
- $P \models [\varphi]$ iff $\exists Q$ such that $P \sqsupseteq [Q]$ and $Q \models \varphi$
- $P \models (\!|\varphi|\!)$ iff $\exists P', Q$ such that $P \sqsupseteq P'$ and $P' \Subset Q$ and $Q \models \varphi$.

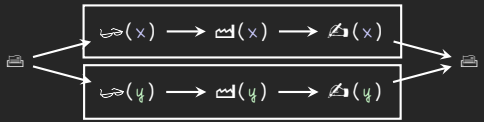
Theorem

$$P \sqsupseteq Q \Leftrightarrow \forall \varphi, (P \models \varphi \Rightarrow Q \models \varphi)\ .$$

*Not to be confused with the logic with the same name introduced by Christian Rétoré & Alain Lecomte in 1995

MUTUAL EXCLUSION (III)

```
          print(counter);
atomic{           ||  atomic{
    x:=counter;   ||      y:=counter;
    x:=x+1;       ||      y:=y+1;
    counter:=x;   ||      counter:=y;
}                 ||  }
          print(counter);
```



$$P := 🖶 \cdot \left[𝓇(x) \cdot 𝓌(x) \cdot 𝒶(x)\right] \parallel \left[𝓇(y) \cdot 𝓌(y) \cdot 𝒶(y)\right] \cdot 🖶$$

Breaking mutual exclusion

$\leftrightarrow$ admitting an execution with the "pattern":

$$P \models \left(\!\left|\left(𝓇(x) \star 𝓇(y)\right) \blacktriangleright \left(𝒶(x) \star 𝒶(y)\right)\right|\!\right)$$

RECENT DEVELOPMENTS IN CKA
Outline

I. Concurrent Kleene Algebra

II. CKA with observations

III. Partially observable CKA

IV. CKA with boxes

V. Ongoing and future work

ALGEBRAS WITH HYPOTHESES

- ☞ Doumane, Kuperberg, Pous, & Pradic, “Kleene Algebra with Hypotheses”, FoSSaCS ’19.

ALGEBRAS WITH HYPOTHESES

- Doumane, Kuperberg, Pous, & Pradic, "Kleene Algebra with Hypotheses", FoSSaCS '19.
- Kappé, B., Silva, Wagemaker, & Zanasi, "Concurrent Kleene Algebra with Observations: from Hypotheses to Completeness", FoSSaCS '20.

ALGEBRAS WITH HYPOTHESES

- Doumane, Kuperberg, Pous, & Pradic, "Kleene Algebra with Hypotheses", FoSSaCS '19.
- Kappé, B., Silva, Wagemaker, & Zanasi, "Concurrent Kleene Algebra with Observations: from Hypotheses to Completeness", FoSSaCS '20.
- CKA with boxes and hypotheses?

ALGEBRAS WITH HYPOTHESES

- ☞ Doumane, Kuperberg, Pous, & Pradic, "Kleene Algebra with Hypotheses", FoSSaCS '19.
- ☞ Kappé, B., Silva, Wagemaker, & Zanasi, "Concurrent Kleene Algebra with Observations: from Hypotheses to Completeness", FoSSaCS '20.
- ☞ CKA with boxes and hypotheses?

All proofs had to be re-done from scratch.

ALGEBRAS WITH HYPOTHESES

- ☞ Doumane, Kuperberg, Pous, & Pradic, "Kleene Algebra with Hypotheses", FoSSaCS '19.
- ☞ Kappé, B., Silva, Wagemaker, & Zanasi, "Concurrent Kleene Algebra with Observations: from Hypotheses to Completeness", FoSSaCS '20.
- ☞ CKA with boxes and hypotheses?

All proofs had to be re-done from scratch.

Can we do better?

LOGICS OF BEHAVIOUR

☞ Traditional approaches to program logic rely on states
e.g. Hennessy–Milner Logic, (Propositional) Dynamic Logic…

LOGICS OF BEHAVIOUR

- ☞ Traditional approaches to program logic rely on states
  e.g. Hennessy–Milner Logic, (Propositional) Dynamic Logic…
- ☞ Pomset logic relies on an abstract notion of "behaviour" instead.

LOGICS OF BEHAVIOUR

- ☞ Traditional approaches to program logic rely on states
  e.g. Hennessy–Milner Logic, (Propositional) Dynamic Logic…
- ☞ Pomset logic relies on an abstract notion of "behaviour" instead.

What kinds of properties of behaviours are interesting and/or tractable?

EXTENSIONS OF THE MODEL

☞ Merging boxes: $[e \cdot [f] \cdot g] = [e \cdot f \cdot g]$.

EXTENSIONS OF THE MODEL

☞ Merging boxes: $[e \cdot [f] \cdot g] = [e \cdot f \cdot g]$.

☞ Beyond partial memory states: Arbitrary coherence relation between atomic observations.

$$V = 1 \asymp V = 0$$

EXTENSIONS OF THE MODEL

- Merging boxes: $[e \cdot [f] \cdot g] = [e \cdot f \cdot g]$.

- Beyond partial memory states: Arbitrary coherence relation between atomic observations.

$$V = 1 \asymp V = 0$$

- Add data: Nominal algebras.

THAT'S ALL FOLKS!

Thank you!

See more at:
http://paul.brunet-zamansky.fr

RECENT DEVELOPMENTS IN CKA
Outline

I. Concurrent Kleene Algebra

II. CKA with observations

III. Partially observable CKA

IV. CKA with boxes

V. Ongoing and future work