AVANCÉES RÉCENTES
SUR LES
ALGÈBRES DE KLEENE CONCURRENTES

SÉMINAIRE LIRICA - MARSEILLE

Janvier 2021

Paul Brunet
University College London

ALGÈBRE DE KLEENE CONCURRENTE

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

2009

CKA est proposée.

ALGÈBRE DE KLEENE CONCURRENTE

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

2009

CKA est proposée.

2011

Introduction de modèles de CKA, comparaison avec la logique de séparation.

ALGÈBRE DE KLEENE CONCURRENTE

2009
2011
2014

CKA est proposée.

Introduction de modèles de CKA, comparaison avec la logique de séparation.

Premier théorème de complétude (sans la loi d'échange), introduction de tests dans CKA.

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

ALGÈBRE DE KLEENE CONCURRENTE

Concurrent Kleene algebra with tests and branching automata

Peter Jipsen*, M. Andrew Moshier

Chapman University, Orange, CA 92866, USA

CrossMark

2009 — CKA est proposée.

2011 — Introduction de modèles de CKA, comparaison avec la logique de séparation.

2014 — Premier théorème de complétude (sans la loi d'échange), introduction de tests dans CKA.

2016 — Second article sur CKAT, qui corrige certaines erreurs du premier.

ALGÈBRE DE KLEENE CONCURRENTE

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

2009 — CKA est proposée.

2011 — Introduction de modèles de CKA, comparaison avec la logique de séparation.

2014 — Premier théorème de complétude (sans la loi d'échange), introduction de tests dans CKA.

2016 — Second article sur CKAT, qui corrige certaines erreurs du premier.

2017 — Décidabilité de CKA.

ALGÈBRE DE KLEENE CONCURRENTE

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

2009 — CKA est proposée.

2011 — Introduction de modèles de CKA, comparaison avec la logique de séparation.

2014 — Premier théorème de complétude (sans la loi d'échange), introduction de tests dans CKA.

2016 — Second article sur CKAT, qui corrige certaines erreurs du premier.

2017 — Décidabilité de CKA.

2018 — Complétude de CKA.

ALGÈBRE DE KLEENE C

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

2009 2011 2014 2016 2017 2018 2020

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

CKA es

de
a loi
ion de

Décidabilité de CKA.

Complétude de CKA.

CKA avec observations (totales/partielles), CKA avec boîtes.

Second article sur CKAT, qui corrige certaines erreurs du premier.

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

Interprétation : langages réguliers

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

example :

$$[\![a \cdot ((a+b) \cdot a)^\star]\!] = \left\{\begin{array}{l}\text{mots sur l'alphabet } \{a, b\} \text{ de lon-} \\ \text{gueur impaire et tels qu'une lettre} \\ \text{sur deux est un a}\end{array}\right\}$$

$$= \{a, aaa, aba, aaaaa, abaaa, aaaba, ababa, \ldots\}$$

Algèbre de Kleene – la théorie des langages réguliers

Les axiomes de KA

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

Théorème

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

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

KAT – la théorie des programmes impératifs

Syntaxe

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

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

Encode un langage `While` simple :

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

KAT – LA THÉORIE DES PROGRAMMES IMPÉRATIFS

skip

tuer l'execution

Syntaxe

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

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

Encode un langage `While` simple :

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

KAT – la théorie des programmes impératifs

skip

tuer l'execution

action atomique

Syntaxe

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

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

Encode un langage **While** simple :

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

KAT – LA THÉORIE DES PROGRAMMES IMPÉRATIFS

skip

test

tuer l'execution

action atomique

Syntaxe

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

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

test atomique

Encode un langage **While** simple :

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

KAT – La théorie des programmes impératifs

skip

test

choix non-déterministe

boucle non-déterministe

tuer l'execution

action atomique

composition séquentielle

Syntaxe

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

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

test atomique

Encode un langage `While` simple :

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

KAT – LA THÉORIE DES PROGRAMMES IMPÉRATIFS

skip
test
choix non-déterministe
tuer l'execution
action atomique
composition séquentielle
boucle non-déterministe

Syntaxe

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

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

test atomique

Encode un langage **While** simple :

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

Interprétation : langages de chaînes gardées

chaînes guardées : séquences alternant états $\in 2^B$ & actions $\in A$.

$$\begin{array}{|c|c|}\hline \alpha_1 & 1 \\ \alpha_2 & 0 \\ \alpha_3 & 1 \\ \alpha_4 & 1 \\ \hline\end{array} \longrightarrow a \longrightarrow \begin{array}{|c|c|}\hline \alpha_1 & 1 \\ \alpha_2 & 0 \\ \alpha_3 & 0 \\ \alpha_4 & 1 \\ \hline\end{array} \longrightarrow b \longrightarrow \begin{array}{|c|c|}\hline \alpha_1 & 0 \\ \alpha_2 & 0 \\ \alpha_3 & 1 \\ \alpha_4 & 0 \\ \hline\end{array} \longrightarrow c \longrightarrow \begin{array}{|c|c|}\hline \alpha_1 & 1 \\ \alpha_2 & 1 \\ \alpha_3 & 1 \\ \alpha_4 & 0 \\ \hline\end{array}$$

KAT – La théorie des programmes impératifs

Les axiomes de KAT :

- ☞ Le axiomes de KA.
- ☞ Pour les tests, les axiomes des algèbres booléennes.
- ☞ Les axiomes "glue" suivants :

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

Théorème

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

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

KAT – La théorie des programmes impératifs

Les axiomes de KAT :

- Le axiomes de KA.
- Pour les tests, les axiomes des algèbres booléennes.
- Les axiomes "glue" suivants :

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

Théorème

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

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

Encode la logique de Hoare propositionnelle :

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

KAT – La théorie des programmes impératifs

Les axiomes de KAT :

- Le axiomes de KA.
- Pour les tests, les axiomes des algèbres booléennes.
- Les axiomes "glue" suivants :

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

Théorème

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

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

Encode la logique de Hoare propositionnelle :

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

Peut-on faire pareil pour les programmes concurrents ?

Plan

AVANCÉES RÉCENTES SUR CKA
Plan

I. Algèbre de Kleene concurrente

II. Observations booléennes

III. Observations partielles

IV. Boîtes

V. Travaux en cours et à venir

BI-ALGÈBRE DE KLEENE

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

Définition

Une bi-algèbre de Kleene est une structure $\langle \mathscr{A}, 0, 1, \cdot, \parallel, +, \star, ! \rangle$ telle que :

- ☞ $\langle \mathscr{A}, 0, 1, \cdot, +, \star \rangle$ est une KA
- ☞ $\langle \mathscr{A}, 0, 1, \parallel, +, ! \rangle$ est une KA commutative.

BI-ALGÈBRE DE KLEENE

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

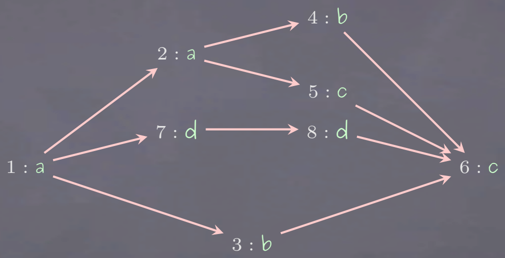
Définition

Une bi-algèbre de Kleene est une structure $\langle \mathscr{A}, 0, 1, \cdot, \parallel, +, \star, ! \rangle$ telle que :

- ☞ $\langle \mathscr{A}, 0, 1, \cdot, +, \star \rangle$ est une KA
- ☞ $\langle \mathscr{A}, 0, 1, \parallel, +, ! \rangle$ est une KA commutative.

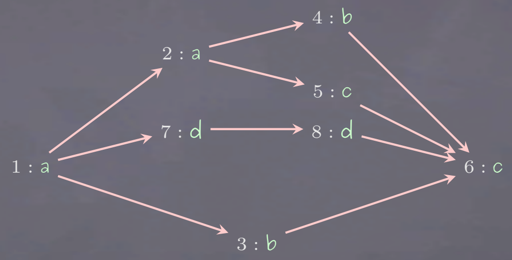
À quoi ressemble la bi-KA libre ?

POMSETS - TRACES CONCURRENTES



POMSETS – TRACES CONCURRENTES

A est un alphabet d'actions.



ensemble fini d'évènements

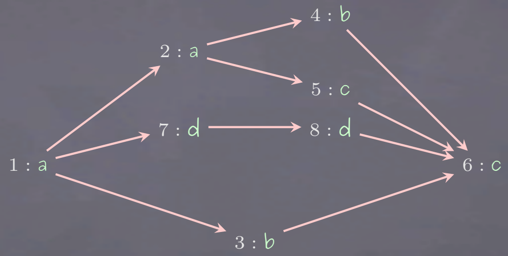
fonction d'étiquetage $: E_P \to A$

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

ordre partiel $\subseteq E_P \times E_P$

POMSETS – TRACES CONCURRENTES

A est un alphabet d'actions.



ensemble fini d'évènements

fonction d'étiquetage $: E_P \to A$

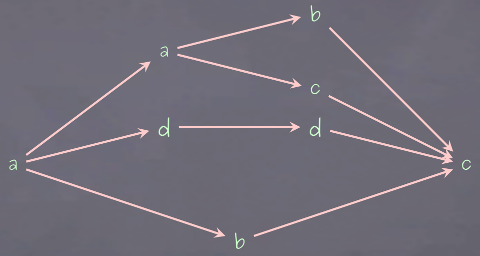
$$P = \langle E_P, \leqslant_P, \lambda_P \rangle$$

ordre partiel $\subseteq E_P \times E_P$

$\boxed{\text{À isomorphisme près } \equiv.}$

POMSETS - TRACES CONCURRENTES

A est un alphabet d'actions.



ensemble fini d'évènements

fonction d'étiquetage $: E_P \to A$

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

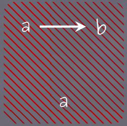
ordre partiel $\subseteq E_P \times E_P$

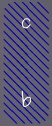
$\boxed{\text{À isomorphisme près } \equiv.}$

COMBINER LES POMSETS

$a =$ 

$1 =$ 

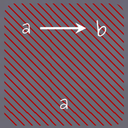
$P_1 =$ 

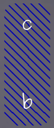
$P_2 =$ 

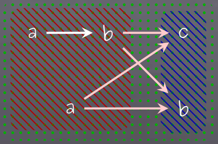
COMBINER LES POMSETS

$a =$ 

$1 =$ 

$P_1 =$ 

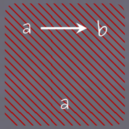
$P_2 =$ 

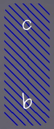
$P_1 ; P_2 =$ 

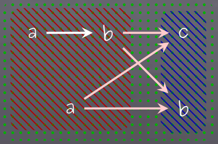
COMBINER LES POMSETS

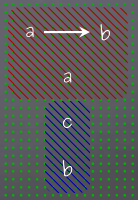
$a =$ 

$1 =$ 

$P_1 =$ 

$P_2 =$ 

$P_1 ; P_2 =$ 

$P_1 \parallel P_2 =$ 

COMPLÉTUDE DE BIKA

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

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

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

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

Théorème

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

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

ENTRELACEMENTS ET SUBSOMPTION

Loi d'échange faible

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

$$\begin{tikzcd} a \arrow[r] \arrow[dr] & c \\ b \arrow[r] \arrow[ur] & d \end{tikzcd} \quad \sqsubseteq \quad \begin{tikzcd} a \arrow[r] & c \\ b \arrow[r] & d \end{tikzcd}$$

$P \sqsubseteq Q$ quand il y a un homomorphisme de Q à P, c.à.d. une application bijective $\varphi : E_Q \to E_P$ telle que $\lambda_P \circ \varphi = \lambda_Q$ et $\varphi(\leq_Q) \subseteq \leq_P$.

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

ALGÈBRE DE KLEENE CONCURRENTE

Loi d'échange faible

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

Pas d'itération parallèle

CKA

Une algèbre de Kleene concurrente est une bi-algèbre de Kleene faible $\langle \mathscr{A}, 0, 1, \cdot, \parallel, +, \star \rangle$ qui satisfait la loi d'échange faible.

COMPLÉTUDE ET DÉCIDABILITÉ DE CKA

Théorème

Tester que deux expressions dénotent le même langage clos est un problème ExpSpace-complet.

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

Théorème

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

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

AVANCÉES RÉCENTES SUR CKA
Plan

I. Algèbre de Kleene concurrente

II. Observations booléennes

III. Observations partielles

IV. Boîtes

V. Travaux en cours et à venir

CKAT

Slogan

Une KAT est une KA avec une sous-algèbre booléenne.
Une CKAT est une CKA avec une sous-algèbre booléenne.

CKAT

Slogan

Une KAT est une KA avec une sous-algèbre booléenne.
Une CKAT est une CKA avec une sous-algèbre booléenne.

$$
\begin{aligned}
t? \cdot p \cdot \left(\neg t\right)? &\leq p \parallel \left(t? \cdot \left(\neg t\right)?\right) && \text{(axiomes de CKA)} \\
&= p \parallel \left(t \wedge \neg t\right)? && (a \wedge b)? \;=\; a? \cdot b? \\
&= p \parallel \bot? && \text{(axiomes booléens)} \\
&= p \parallel 0 && (\bot? = 0) \\
&= 0 && \text{(axiomes de CKA)}
\end{aligned}
$$

CKAT

DOOMED !

Slogan

Une KAT est une KA avec une sous-algèbre booléenne.
Une CKAT est une CKA avec une sous-algèbre booléenne.

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

$\leftrightarrow$ Pour tout programme et toute assertion, le triplet $\{t\}p\{t\}$ est valide.

$\leftrightarrow$ Chaque test est un invariant de tous les programmes.

À QUI LA FAUTE?

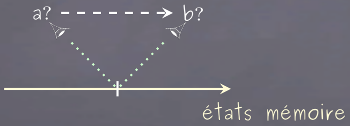
$$
\begin{aligned}
t? \cdot p \cdot (\neg t)? &\leq p \parallel \left(t? \cdot (\neg t)?\right) && \text{(axiomes de CKA)}\\
&= p \parallel (t \wedge \neg t)? && (a \wedge b)? \;=\; a? \cdot b?\\
&= p \parallel \bot? && \text{(axiomes booléens)}\\
&= p \parallel 0 = 0 && (\bot? = 0 \;+\; \text{axiomes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

À QUI LA FAUTE?

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

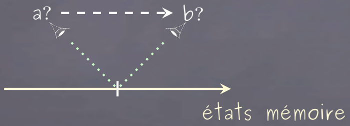
À QUI LA FAUTE ?

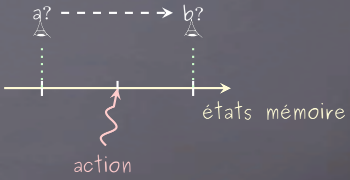
$$
\begin{aligned}
t? \cdot p \cdot (\neg t)? &\leq p \parallel \left(t? \cdot (\neg t)?\right) && \text{(axiomes de CKA)} \\
&= p \parallel (t \wedge \neg t)? && \boxed{(a \wedge b)? = a? \cdot b?} \\
&= p \parallel \bot? && \text{(axiomes booléens)} \\
&= p \parallel 0 = 0 && (\bot? = 0 \text{ + axiomes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



À QUI LA FAUTE ?

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





À QUI LA FAUTE ?

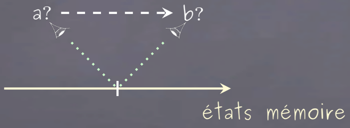
$$
\begin{aligned}
t? \cdot p \cdot (\neg t)? &\leq p \parallel \big(t? \cdot (\neg t)?\big) \\
&= p \parallel (t \wedge \neg t)? \\
&= p \parallel \bot? \\
&= p \parallel 0 = 0
\end{aligned}
$$

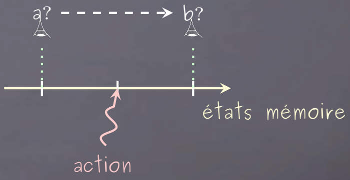
(axiomes de CKA)

$\boxed{\cancel{(a \wedge b)? = a? \cdot b?}}$

(axiomes booléens)

($\bot? = 0$ + axiomes de CKA)





$$\boxed{(a \wedge b)? \leq a? \cdot b?}$$

CKAO - SYNTAXE

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

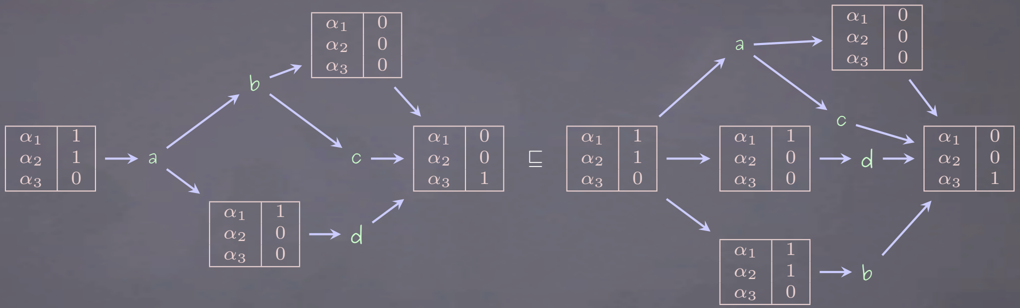
$$t, t_1, t_2 \in \mathbf{O}_A^{bool} ::= \top \mid \bot \mid \alpha \in B \mid t_1 \wedge t_2 \mid t_1 \vee t_2 \mid \neg t$$

Axiomes de CKAO

- Tous les axiomes de CKA.
- Pour les obsertions, les axiomes booléens.
- Les axiomes "glue" suivants :

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

CKAO – MODÈLE



Théorème

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

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

INTERLUDE – (C)KA MODULO HYPOTHÈSES

H : ensemble d'hypothèses $e \leq f$ sur un alphabet A donné.

- ☞ structure algébrique de l'alphabet (e.g. $\alpha \wedge \beta = \beta \wedge \alpha$) ;
- ☞ propriétés dynamiques (e.g. $\alpha \leq \alpha \cdot \alpha$)
- ☞ autres propriétés du système modélisé.

Théorème

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

- ☞ 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

Plan

I. Algèbre de Kleene concurrente

II. Observations booléennes

III. Observations partielles

IV. Boîtes

V. Travaux en cours et à venir

LITMUS TEST – COHÉRENCE SÉQUENTIELLE

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

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

{ r0 == 1 || r1 == 1 }
```

Ingrédients :

☞ Affectations $x \leftarrow 1$

☞ Observations $r_0 \mapsto 0$

DE QUELLES OBSERVATIONS AVONS NOUS BESOIN ?
Première tentative : algèbre booléenne

☞ <u>Observations atomiques</u> : $\mathrm{VAR} \mapsto \mathrm{VAL}$ e.g. $r_0 \mapsto 1$

DE QUELLES OBSERVATIONS AVONS NOUS BESOIN ?

Première tentative : algèbre booléenne

- Observations atomiques : $\mathrm{VAR} \mapsto \mathrm{VAL}$ e.g. $r_0 \mapsto 1$
- Formules booléennes : ensembles d'états mémoire $\mathrm{VAR} \to \mathrm{VAL}$ e.g.

r_0	1
r_1	0

DE QUELLES OBSERVATIONS AVONS NOUS BESOIN ?
Première tentative : algèbre booléenne

- Observations atomiques : $\mathrm{Var} \mapsto \mathrm{Val}$ e.g. $r_0 \mapsto 1$
- Formules booléennes : ensembles d'états mémoire $\mathrm{Var} \to \mathrm{Val}$ e.g. $\begin{array}{|c|c|}\hline r_0 & 1 \\ r_1 & 0 \\ \hline\end{array}$
- Affectations : $\sum_{s \in \mathrm{State}} s \cdot (v \leftarrow n) \cdot s[v \mapsto n]$, c.à.d.

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

DE QUELLES OBSERVATIONS AVONS NOUS BESOIN ?
Première tentative : algèbre booléenne

- <u>Observations atomiques</u> : $\mathrm{VAR} \mapsto \mathrm{VAL}$ e.g. $r_0 \mapsto 1$
- <u>Formules booléennes</u> : ensembles d'états mémoire $\mathrm{VAR} \to \mathrm{VAL}$ e.g. $\begin{array}{|c|c|}\hline r_0 & 1 \\ r_1 & 0 \\ \hline\end{array}$
- <u>Affectations</u> : $\sum_{s \in \mathrm{state}} s \cdot (v \leftarrow n) \cdot s[v \mapsto n]$, c.à.d.

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

$\boxed{\text{Problème : composition parallèle ?}}$

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

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

ALGÈBRE D'OBSERVATIONS PARTIELLES

Idée : au lieu d'états mémoire $\mathrm{VAR} \to \mathrm{VAL}$, on considère des fonctions partielles $\mathrm{VAR} \rightharpoonup \mathrm{VAL}$.

PCDL d'observations

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

Mêmes axiomes que BA pour $\vee, \wedge, \top, \bot$, plus :

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

PCDL : trellis distributif pseudo-complémenté

Théorème

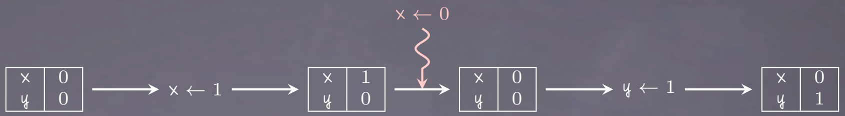
$$\mathrm{POCKA} \vdash e = f \Leftrightarrow [\![e]\!]\downarrow^{\mathrm{pocka}} = [\![f]\!]\downarrow^{\mathrm{pocka}}.$$

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

CAUSALITÉ VS COMPOSITIONNALITÉ

$$\begin{array}{|c|c|}\hline x & 0 \\ y & 0 \\ \hline \end{array} \longrightarrow x \leftarrow 1 \longrightarrow \begin{array}{|c|c|}\hline x & 1 \\ y & 0 \\ \hline \end{array} \xrightarrow{x \leftarrow 0} \begin{array}{|c|c|}\hline x & 0 \\ y & 0 \\ \hline \end{array} \longrightarrow y \leftarrow 1 \longrightarrow \begin{array}{|c|c|}\hline x & 0 \\ y & 1 \\ \hline \end{array}$$

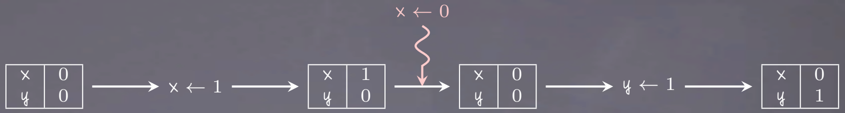
CAUSALITÉ VS COMPOSITIONNALITÉ



Solution : il faut « clore » le système explicitement.

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

CAUSALITÉ VS COMPOSITIONNALITÉ



Solution : il faut « clore » le système explicitement.

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

Litmus test :

$$t \coloneqq (r_0 \mapsto 0 \wedge r_1 \mapsto 0) \cdot ((x \leftarrow 1 \cdot r_0 \leftarrow y) \parallel (y \leftarrow 1 \cdot r_1 \leftarrow x)) \cdot \overline{(r_0 \mapsto 1 \vee r_1 \mapsto 1)}$$

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

Plan

I. Algèbre de Kleene concurrente

II. Observations booléennes

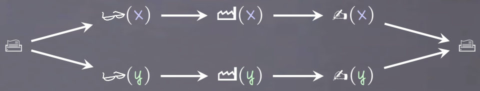
III. Observations partielles

IV. Boîtes

V. Travaux en cours et à venir

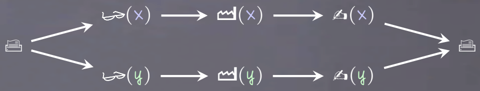
EXCLUSION MUTUELLE

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



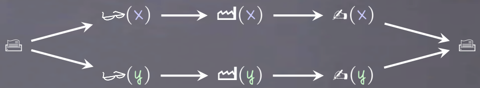
EXCLUSION MUTUELLE

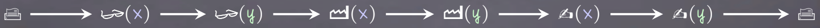
```
      print(counter);
              ||
x:=counter;   ||   y:=counter;
x:=x+1;       ||   y:=y+1;
counter:=x;   ||   counter:=y;
              ||
      print(counter);
```



EXCLUSION MUTUELLE

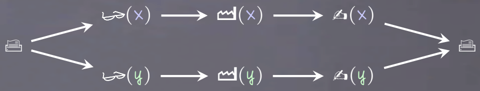
```
     print(counter);
            ||
x:=counter; ||   y:=counter;
x:=x+1;     ||   y:=y+1;
counter:=x; ||   counter:=y;
            ||
     print(counter);
```

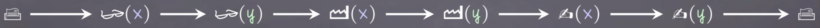




EXCLUSION MUTUELLE

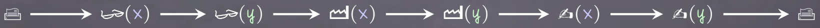
```
         print(counter);
atomic{                 || atomic{
    x:=counter;         ||     y:=counter;
    x:=x+1;             ||     y:=y+1;
    counter:=x;         ||     counter:=y;
}                       || }
         print(counter);
```





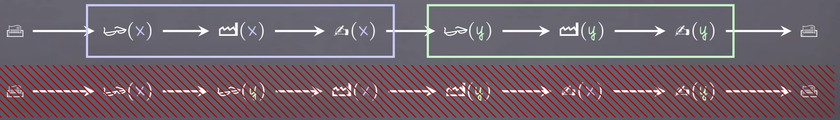
EXCLUSION MUTUELLE

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

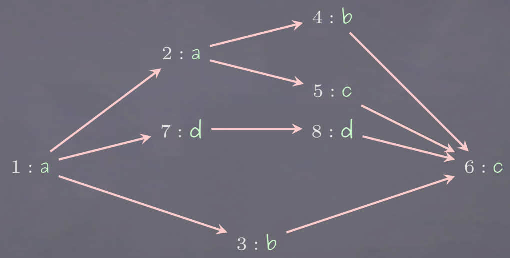


EXCLUSION MUTUELLE

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



POMSETS EMBOÎTÉS



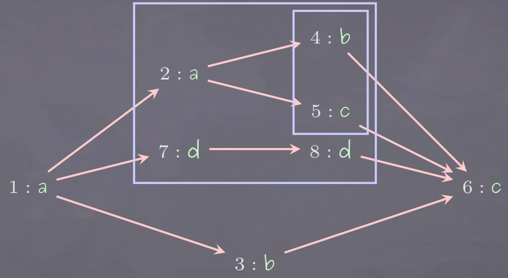
ensemble fini d'évènements

fonction d'étiquetage $: E_P \to A$

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

ordre partiel $\subseteq E_P \times E_P$

POMSETS EMBOÎTÉS



fonction d'étiquetage
$: E_P \to A$

ensemble fini d'évènements

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

ordre partiel
$\subseteq E_P \times E_P$

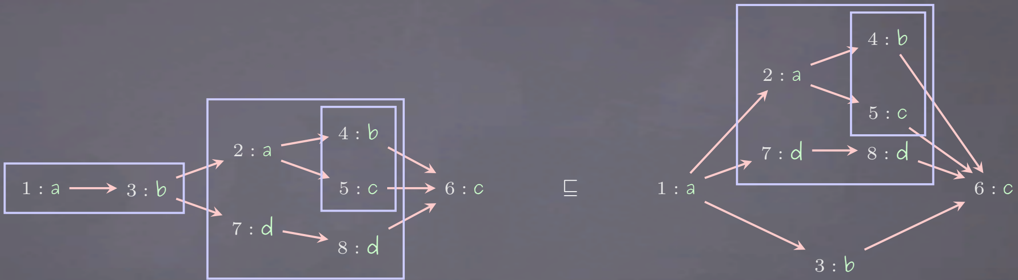
ensemble de boîtes
$\subseteq \mathcal{P}^{\neq\emptyset}(E_P)$

SUBSOMPTION AVEC BOÎTES

$P \sqsubseteq Q$ lorsequ'il existe un homomorphisme de Q vers P, c.à.d. une bijection $\varphi : E_Q \to E_P$ telle que :

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

SUBSOMPTION AVEC BOÎTES



$P \sqsubseteq Q$ lorsequ'il existe un homomorphisme de Q vers P, c.à.d. une bijection $\varphi : E_Q \to E_P$ telle que :

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

AXIOMATISATION

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

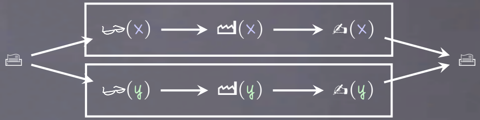
Thèse

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

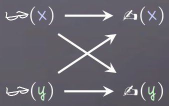
B. & Pym, "Pomsets with Boxes : Protection, Separation, and Locality in Concurrent Kleene Algebra.", FSCD '20

EXCLUSION MUTUELLE – II

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



Enfreindre l'exclusion mutuelle ↔ admettre une exécution contenant le « motif » suivant :



POMSET LOGIC *

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

- ☞ $P \models \varphi \blacktriangleright \psi$ ssi $\exists P_1, P_2$ tels que $P \sqsupseteq P_1 ; P_2$ et $P_1 \models \varphi$ et $P_2 \models \psi$
- ☞ $P \models \varphi \star \psi$ ssi $\exists P_1, P_2$ tels que $P \sqsupseteq P_1 \parallel P_2$ et $P_1 \models \varphi$ et $P_2 \models \psi$
- ☞ $P \models [\varphi]$ ssi $\exists Q$ tel que $P \sqsupseteq [Q]$ et $Q \models \varphi$
- ☞ $P \models (\!|\varphi|\!)$ ssi $\exists P', Q$ tels que $P \sqsupseteq P'$ et $P' \ni Q$ et $Q \models \varphi$.

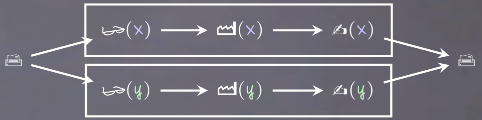
Théorème

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

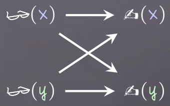
*. À ne pas confondre avec la logique du même nom proposée par Christian Rétoré & Alain Lecomte en 1995

Exclusion mutuelle – III

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



Enfreindre l'exclusion mutuelle ↔ admettre une exécution contenant le « motif » suivant :



↔ $P \models (\!| (☞(x) \star ☞(y)) \blacktriangleright (✍(x) \star ✍(y)) |\!)$

Plan

I. Algèbre de Kleene concurrente

II. Observations booléennes

III. Observations partielles

IV. Boîtes

V. Travaux en cours et à venir

Algèbres à hypothèses

☞ Doumane, Kuperberg, Pous, & Pradic, "Kleene Algebra with Hypotheses", FoSSaCS '19.

Algèbres à hypothèses

- 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.

Algèbres à hypothèses

- 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 avec boîtes et hypothèses ?

ALGÈBRES À HYPOTHÈSES

- 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 avec boîtes et hypothèses ?

Il faut tout refaire !

Algèbres à hypothèses

- 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 avec boîtes et hypothèses ?

Il faut tout refaire !

Peut-on faire mieux ?

LOGIQUES DE COMPORTEMENTS

☞ Les approches traditionnelles pour les logiques de programmes reposent sur une notion d'état
e.g. Hennessy-Milner Logic, (Propositional) Dynamic Logic…

LOGIQUES DE COMPORTEMENTS

- ☞ Les approches traditionnelles pour les logiques de programmes reposent sur une notion d'état
  e.g. Hennessy–Milner Logic, (Propositional) Dynamic Logic…
- ☞ La « Pomset logic » en revanche repose sur une notion abstraite de **comportement.**

LOGIQUES DE COMPORTEMENTS

- ☞ Les approches traditionnelles pour les logiques de programmes reposent sur une notion d'état
  e.g. Hennessy-Milner Logic, (Propositional) Dynamic Logic…
- ☞ La « Pomset logic » en revanche repose sur une notion abstraite de **comportement.**

Quelles propriétés des comportements peut-on/souhaite-t-on capturer ?

EXTENSIONS DU MODÈLE

- ☞ Fusionner les boîtes : $[e \cdot [f] \cdot g] = [e \cdot f \cdot g]$.

EXTENSIONS DU MODÈLE

- **Fusionner les boîtes :** $[e \cdot [f] \cdot g] = [e \cdot f \cdot g]$.

- **Au delà des états mémoire :** Relation de cohérence arbitraire entre les observations atomiques.

$$v \mapsto 1 \asymp v \mapsto 0$$

EXTENSIONS DU MODÈLE

☞ **Fusionner les boîtes :** $[e \cdot [f] \cdot g] = [e \cdot f \cdot g]$.

☞ **Au delà des états mémoire :** Relation de cohérence arbitraire entre les observations atomiques.

$$v \mapsto 1 \asymp v \mapsto 0$$

☞ **Ajouter des données :** Algèbres nominales.

C'EST TOUT !

Merci de votre attention !

https://paul.brunet-zamansky.fr

Plan

I. Algèbre de Kleene concurrente

II. Observations booléennes

III. Observations partielles

IV. Boîtes

V. Travaux en cours et à venir