Representations

RAMiCS 2026

Paul Brunet

Université Paris-Est Créteil val de Marne

Outline

I - Motivation

At first were Kleene algebras

My background: extensions of Kleene algebras.

At first were Kleene algebras

My background: extensions of Kleene algebras.

☞ Kleene algebras with converse [B and Pous, RAMiCS 2014]

At first were Kleene algebras

My background: extensions of Kleene algebras.

☞ Kleene algebras with converse [B and Pous, RAMiCS 2014]

☞ Kleene allegories [B and Pous, LICS 2015]

At first were Kleene algebras

My background: extensions of Kleene algebras.

- ☞ Kleene algebras with converse [B and Pous, RAMiCS 2014]
- ☞ Kleene allegories [B and Pous, LICS 2015]
- ☞ Concurrent Kleene algebras
  [B, Pous and Struth, CONCUR 2017;
  Kappé, B, Silva, Zanasi, ESOP 2018]

At first were Kleene algebras

My background: extensions of Kleene algebras.

- ☞ Kleene algebras with converse [B and Pous, RAMiCS 2014]
- ☞ Kleene allegories [B and Pous, LICS 2015]
- ☞ Concurrent Kleene algebras
  [B, Pous and Struth, CONCUR 2017;
  Kappé, B, Silva, Zanasi, ESOP 2018]
- ☞ (Concurrent) Kleene algebra with observations
  [Kappé, B, Silva, Zanasi, Rot, Wagemaker,
  CONCUR 2019 & FoSSaCS 2020]

At first were Kleene algebras

My background: extensions of Kleene algebras.

- ☞ Kleene algebras with converse [B and Pous, RAMiCS 2014]
- ☞ Kleene allegories [B and Pous, LICS 2015]
- ☞ Concurrent Kleene algebras
  [B, Pous and Struth, CONCUR 2017;
  Kappé, B, Silva, Zanasi, ESOP 2018]
- ☞ (Concurrent) Kleene algebra with observations
  [Kappé, B, Silva, Zanasi, Rot, Wagemaker,
  CONCUR 2019 & FoSSaCS 2020]
- ☞ Partially observable Concurrent Kleene Algebra
  [Wagemaker, B, Docherty, Rot, Kappé, Silva, CONCUR 2020]

At first were Kleene algebras

My background: extensions of Kleene algebras.

☞ Kleene algebras with converse [B and Pous, RAMiCS 2014]

☞ Kleene allegories [B and Pous, LICS 2015]

☞ Concurrent Kleene algebras
[B, Pous and Struth, CONCUR 2017;
Kappé, B, Silva, Zanasi, ESOP 2018]

☞ (Concurrent) Kleene algebra with observations
[Kappé, B, Silva, Zanasi, Rot, Wagemaker,
CONCUR 2019 & FoSSaCS 2020]

☞ Partially observable Concurrent Kleene Algebra
[Wagemaker, B, Docherty, Rot, Kappé, Silva, CONCUR 2020]

☞ Language Kleene lattices [B, MFCS 2017, CSL 2020]

At first were Kleene algebras

- ☞ After some time, one realises

  we're always doing the same thing.

At first were Kleene algebras

☞ After some time, one realises

we're always doing the same thing.

☞ when setting up a new extension:
- many proof obligations
- often discharged by tedious structural inductions...

At first were Kleene algebras

- ☞ After some time, one realises

we're always doing the same thing.

- ☞ when setting up a new extension:
  - many proof obligations
  - often discharged by tedious structural inductions...
- ☞ worst example: CKA + hypotheses
  - essentially the same as KA + hypotheses
  - no single lemma could be imported

  → everything had to be redone

At first were Kleene algebras

☞ After some time, one realises

<u>we're always doing the same thing.</u>

☞ when setting up a new extension:
  - many proof obligations
  - often discharged by tedious structural inductions...

☞ worst example: CKA + hypotheses
  - essentially the same as KA + hypotheses
  - no single lemma could be imported

    → everything had to be redone

☞ very frustrating situation.

What can be done ?

Modularity

What can be done ?

- ☛ Build extensions modularly.

What can be done ?

- ☞ Build extensions modularly.
- ☞ Derive properties modularly.

A first step

Kleene algebras with hypotheses

[Doumane, Kuperberg, Pous, and Pradic, FoSSaCS 2019]

☞ calculus of axioms

☞ automatic construction of sound interpretations

☞ first systematic techniques to get completeness for classes of extensions.

A first step

Kleene algebras with hypotheses

[Doumane, Kuperberg, Pous, and Pradic, FoSSaCS 2019]

- calculus of axioms
- automatic construction of sound interpretations
- first systematic techniques to get completeness for classes of extensions.

Limits:

- syntax of Kleene algebras
- fixed alphabet (very propositional)

This talk

☞ A more general framework for completeness proofs.

This talk

- ☞ A more general framework for completeness proofs.
- ☞ Based on the notion of representation.

This talk

- ☞ A more general framework for completeness proofs.
- ☞ Based on the notion of <u>representation</u>.
- ☞ In a nutshell:

This talk

- ☞ A more general framework for completeness proofs.
- ☞ Based on the notion of <u>representation</u>.
- ☞ In a nutshell:

a signature free theory
of
syntax/semantic correspondences.

II - Preliminaries

Relation algebra

In the paper: binary relations $\subseteq A \times B$

Functions and relations

- $f : A \to B$
- $x : A \circleonrightarrow B$

Operators

- identity 1
- composition $x ; y$
- converse x°
- residuals $x \setminus y$

Comparison

- $x \rightarrowtriangle y$
- $x \leftrightarrowtriangle y$

Misc

- $f_* : A \circleonrightarrow B$
- $f^* : B \circleonrightarrow A$

Relation algebra

what do I actually need ?

☞ In general I (boldly) claim that the following is enough:

- a category with arrows $x : A \mathrel{\ominus\!\!\rightarrow} B$
- identity 1, composition $x;y$, converse x°, residuals $x \setminus y$.
- order on relations $x \rightarrowtriangle y$
- A few simple axioms
- not the modularity law (Dedekind's rule)

Relation algebra

what do I actually need ?

☞ In general I (boldly) claim that the following is enough:

- a category with arrows $x : A \mathrel{\circ\!\!\!\!\rightarrow} B$
- identity 1, composition $x;y$, converse x°, residuals $x \setminus y$.
- order on relations $x \rightarrowtriangle y$
- A few simple axioms
- <u>not</u> the modularity law (Dedekind's rule)

<u>ToDo</u>: machine formalisation to check my claim

III - Representations

$$(M, S, \models, \leqslant)$$

Representations

$$\boxed{(M, S, \models, \leqslant)}$$

Representations

$$\boxed{(M, S, \vDash, \leqslant)}$$

☞ set of <u>models</u> M and a set of <u>specifications</u> S.

Representations

$$\boxed{(M, S, \vDash, \leqslant)}$$

- set of models M and a set of specifications S.
- a satisfaction relation $\vDash : M \circleonrightarrow S$ and an order $\leqslant : S \circleonrightarrow S$ s.t.:

Representations

$$\boxed{(M, S, \vDash, \leqslant)}$$

- set of models M and a set of specifications S.
- a satisfaction relation $\vDash$: $M \mathrel{⇴} S$ and an order $\leqslant$: $S \mathrel{⇴} S$ s.t.:
  - $\leqslant$ is reflexive and transitive;

Representations

$$\boxed{(M, S, \vDash, \leqslant)}$$

- set of models M and a set of specifications S.
- a satisfaction relation $\vDash$: $M \mathrel{\circ\!\!\!\!\rightarrow} S$ and an order $\leqslant$: $S \mathrel{\circ\!\!\!\!\rightarrow} S$ s.t.:
  - $\leqslant$ is reflexive and transitive;
  - $\leqslant$ is sound w.r.t. $\vDash$

    i.e. $m \vDash s$ and $s \leqslant t$ implies $m \vDash t$.

Representations

$$\boxed{(M, S, \models, \leqslant)}$$

- set of <u>models</u> M and a set of <u>specifications</u> S.
- a satisfaction relation $\models$: $M ⇴ S$ and an order $\leqslant$: $S ⇴ S$ s.t.:
  - $\leqslant$ is reflexive and transitive;
  - $\leqslant$ is sound w.r.t. $\models$

    i.e. $m \models s$ and $s \leqslant t$ implies $m \models t$.
- completeness of $\leqslant$ w.r.t. $\models$:

  If for every $m \in M$, $m \models s$ implies $m \models t$, then $s \leqslant t$.

reflexivity

$$1 \to \leqslant$$

reflexivity

$1 \to \leqslant$

transitivity

$\leqslant ; \leqslant \to \leqslant$

reflexivity

$1 \to \leqslant$

transitivity

$\leqslant ; \leqslant \to \leqslant$

$\models ; \leqslant \to \models$

soundness

reflexivity

transitivity

$1 \rightarrow \leqslant$

$\leqslant ; \leqslant \rightarrow \leqslant$

$\models ; \leqslant \rightarrow \models$

$\models \backslash \models \rightarrow \leqslant$

soundness

completeness

Representations

$$\boxed{(M, S, \vDash, \leqslant)}$$

(reflexivity)

$$1 \rightarrowtriangle \leqslant$$

(transitivity)

$$\leqslant ; \leqslant \rightarrowtriangle \leqslant$$

$$\vDash ; \leqslant \rightarrowtriangle \vDash$$

(soundness)

$$\vDash \setminus \vDash \rightarrowtriangle \leqslant$$

(completeness)

- Preorder + soundness → representation
- … + completeness → <u>exact</u> representation

Examples

$$\boxed{\mathsf{KA}(A) = (A^\star, \mathsf{Reg}(A), \in; \mathsf{L}^*, \leqslant_{\mathsf{KA}})}$$

$$\boxed{\mathsf{CKA}(A) = \left(\mathsf{Pom}(A), \mathsf{biReg}(A), \in; \mathsf{L}^*_{\mathsf{CKA}}, \leqslant_{\mathsf{CKA}}\right)}$$

Examples

$$\boxed{\mathsf{KA}(A) = (A^\star, \mathrm{Reg}(A), \in; \mathrm{L}^*, \leqslant_{\mathsf{KA}})}$$

$$\boxed{\mathsf{CKA}(A) = \left(\mathrm{Pom}(A), \mathrm{biReg}(A), \in; \mathrm{L}^*_{\mathsf{CKA}}, \leqslant_{\mathsf{CKA}}\right)}$$

☞ In general, representation $\longleftrightarrow$ $(M, S, I, \leqslant)$ with $I : S \to P(M)$ s.t.

$$\forall s_1, s_2, \quad s_1 \leqslant s_2 \Rightarrow I(s_1) \subseteq I(s_2).$$

Examples

$$\boxed{(\mathcal{M}, \mathcal{F}, \vDash, \{(\varphi, \psi) \quad \text{s.t.} \;\; \vdash \varphi \to \psi\})}$$

$$\boxed{\left(\mathcal{V}, \mathrm{Prgm}, \Downarrow, \to_{\beta}^{\star}\right)}$$

$$\boxed{(\mathcal{O}, \mathrm{Prgm}, \text{👁}, \{(\mathrm{p}, \mathrm{q}) \quad \text{s.t.} \quad \mathrm{test}(\mathrm{p}, \mathrm{q}) = \top)}$$

IV - Reductions

A powerful proof technique

$$(M, S, \models, \leqslant) \rightsquigarrow \text{reduction} \rightsquigarrow\longrightarrow (M', S', \models', \leqslant')$$

A powerful proof technique

$$(M, S, \vDash, \leqslant) \rightsquigarrow \text{reduction} \rightsquigarrow (M', S', \vDash', \leqslant')$$

$I(s_1) \subseteq I(s_2)$

A powerful proof technique

$$\begin{tikzcd}
(M, S, \vDash, \leqslant) \arrow[rr, rightsquigarrow, "\text{reduction}"] \arrow[rr, bend left, "\varphi"] & & (M', S', \vDash', \leqslant') \\
I(s_1) \subseteq I(s_2) \arrow[rr, "\text{semantic preservation}"] & & I'(\varphi(s_1)) \subseteq I'(\varphi(s_2))
\end{tikzcd}$$

A powerful proof technique

$$\begin{tikzcd}
(M, S, \vDash, \leqslant) \arrow[rr, squiggly, "\text{reduction}"] \arrow[rr, bend left, "\varphi"] & & (M', S', \vDash', \leqslant') \\
I(s_1) \subseteq I(s_2) \arrow[r, "\text{semantic preservation}"] & I'(\varphi(s_1)) \subseteq I'(\varphi(s_2)) \arrow[d, "\text{exactness}"] & \\
& \varphi(s_1) \leqslant' \varphi(s_2) &
\end{tikzcd}$$

A powerful proof technique

$$\begin{tikzcd}
(M, S, \vDash, \leqslant) \arrow[rrr, squiggly, "\text{reduction}"] \arrow[rrr, bend left, "\varphi"] & & & (M', S', \vDash', \leqslant') \arrow[lll, bend left, "\psi"] \\
I(s_1) \subseteq I(s_2) \arrow[rr, "\text{semantic preservation}"] & & I'(\varphi(s_1)) \subseteq I'(\varphi(s_2)) \arrow[dr, "\text{exactness}"] & \\
 & \psi(\varphi(s_1)) \leqslant \psi(\varphi(s_2)) & & \varphi(s_1) \leqslant' \varphi(s_2) \arrow[ll, "\text{order preservation}"]
\end{tikzcd}$$

A powerful proof technique

$$\begin{tikzcd}
(M, S, \models, \leqslant) \arrow[r, bend left, "\varphi"] \arrow[r, squiggly, "\text{reduction}"] & (M', S', \models', \leqslant') \arrow[l, bend left, "\psi"]
\end{tikzcd}$$

$$\begin{tikzcd}
I(s_1) \subseteq I(s_2) \arrow[r, "\text{semantic preservation}"] & I'(\varphi(s_1)) \subseteq I'(\varphi(s_2)) \arrow[d, "\text{exactness}"] \\
\psi(\varphi(s_1)) \leqslant \psi(\varphi(s_2)) \arrow[d, "\text{loop ppties}"] & \varphi(s_1) \leqslant' \varphi(s_2) \arrow[l, "\text{order preservation}"] \\
s_1 \leqslant \psi(\varphi(s_1)) \leqslant \psi(\varphi(s_2)) \leqslant s_2 &
\end{tikzcd}$$

Definition $r \rightsquigarrow r'$

Let r, r' be representations. A reduction is a triple φ, ψ, τ where:

$$\varphi : S \to S' \qquad \psi : S' \to S \qquad \tau : M' \mathrel{\circ\!\!\!\!\rightarrow} M$$

and subject to the conditions:

$$\vDash'; \varphi^* \leftrightarrowtriangle \tau; \vDash$$

$$\psi^*; \leqslant' \rightarrowtriangle \leqslant; \psi^*$$

$$\psi^*; \varphi^* \rightarrowtriangle \leqslant \qquad \psi^*; \varphi^* \rightarrowtriangle \geqslant$$

A complete proof technique

Lemma

The reduction relation $\rightsquigarrow$ is a preorder.

Lemma

r is exact iff $\exists$ an exact r' such that $r \rightsquigarrow r'$.

A complete proof technique

Lemma

The reduction relation $\rightsquigarrow$ is a preorder.

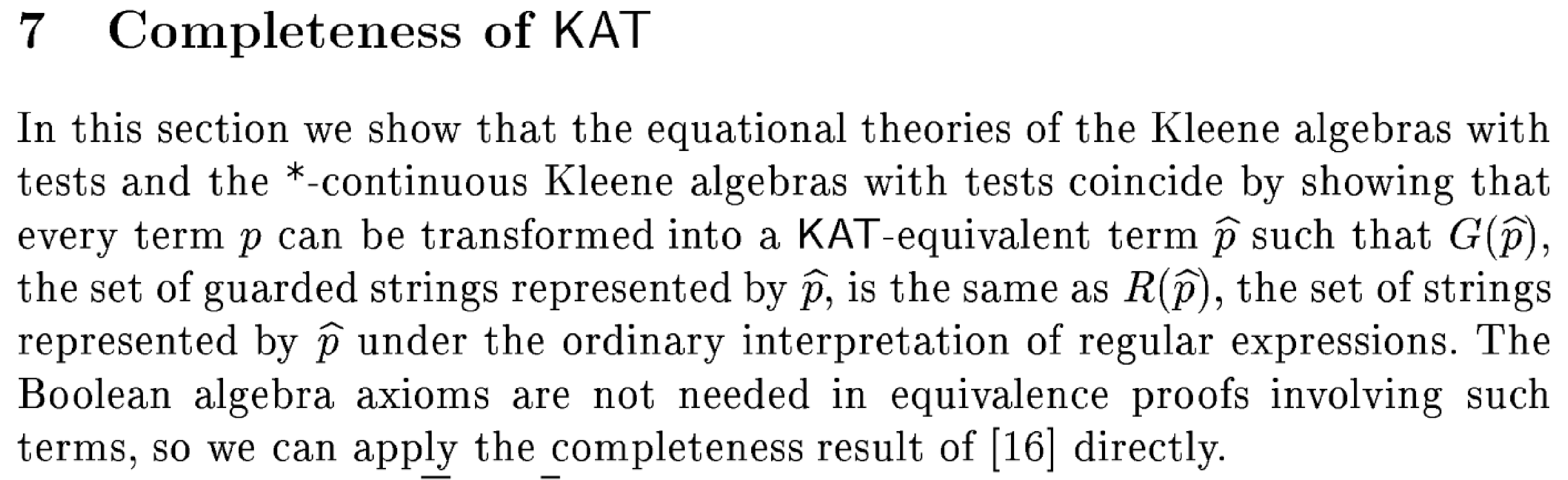
Lemma

r is exact iff $\exists$ an exact r' such that $r \rightsquigarrow r'$.

$$r' = (M, S, \vDash, (\vDash \setminus \vDash))$$

Reductions in the wild

Proof of completeness of KAT



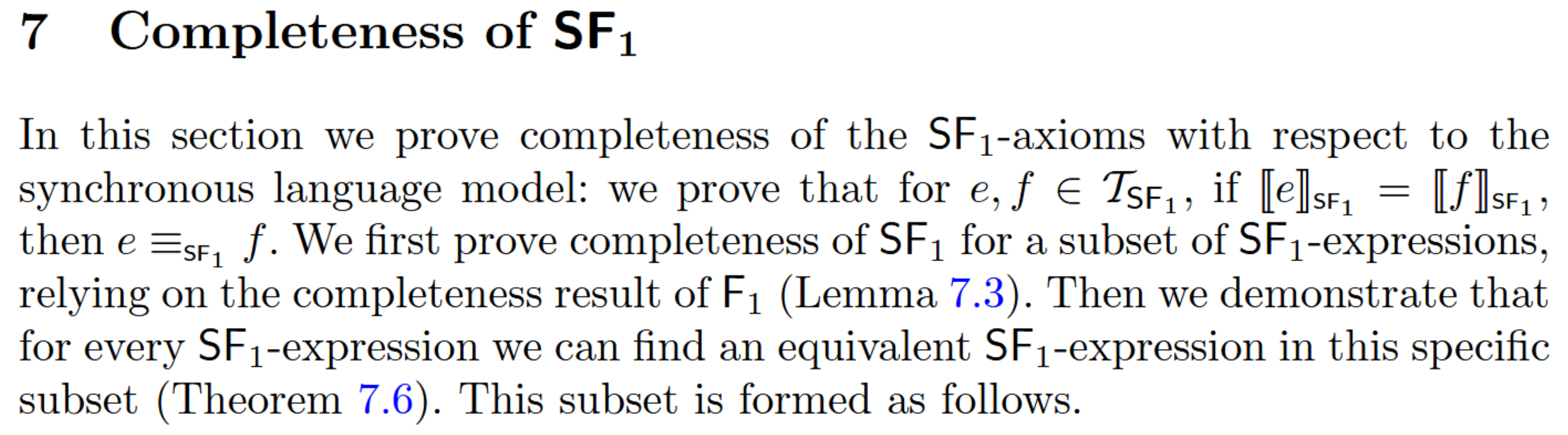
7 Completeness of KAT

In this section we show that the equational theories of the Kleene algebras with tests and the *-continuous Kleene algebras with tests coincide by showing that every term p can be transformed into a KAT-equivalent term $\widehat{p}$ such that $G(\widehat{p})$, the set of guarded strings represented by $\widehat{p}$, is the same as $R(\widehat{p})$, the set of strings represented by $\widehat{p}$ under the ordinary interpretation of regular expressions. The Boolean algebra axioms are not needed in equivalence proofs involving such terms, so we can apply the completeness result of [16] directly.

Kozen & Smith, CSL 1997

Reductions in the wild

Proof of completeness of SKA



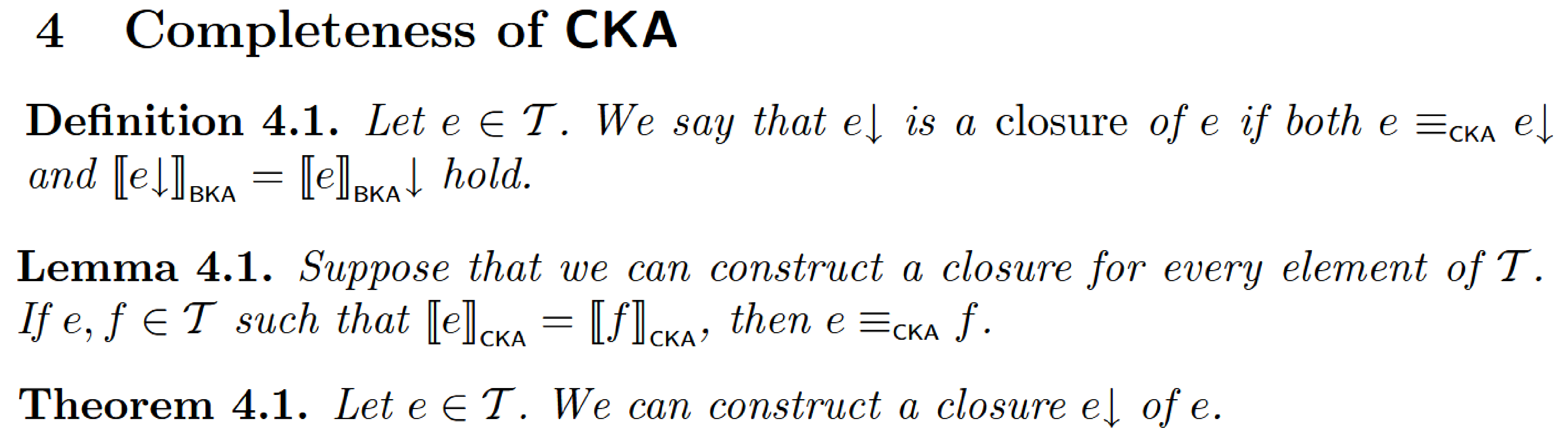
7 Completeness of SF_1

In this section we prove completeness of the SF_1-axioms with respect to the synchronous language model: we prove that for $e, f \in \mathcal{T}_{\mathsf{SF}_1}$, if $[\![e]\!]_{\mathsf{SF}_1} = [\![f]\!]_{\mathsf{SF}_1}$, then $e \equiv_{\mathsf{SF}_1} f$. We first prove completeness of SF_1 for a subset of SF_1-expressions, relying on the completeness result of F_1 (Lemma 7.3). Then we demonstrate that for every SF_1-expression we can find an equivalent SF_1-expression in this specific subset (Theorem 7.6). This subset is formed as follows.

Wagemaker, Bonsangue, Kappé, Rot & Silva, MPC 2019

Reductions in the wild

Proof of completeness of CKA



4 Completeness of CKA

Definition 4.1. *Let* $e \in \mathcal{T}$. *We say that* $e{\downarrow}$ *is a* closure *of* e *if both* $e \equiv_{\mathsf{CKA}} e{\downarrow}$ *and* $[\![e{\downarrow}]\!]_{\mathsf{BKA}} = [\![e]\!]_{\mathsf{BKA}}{\downarrow}$ *hold.*

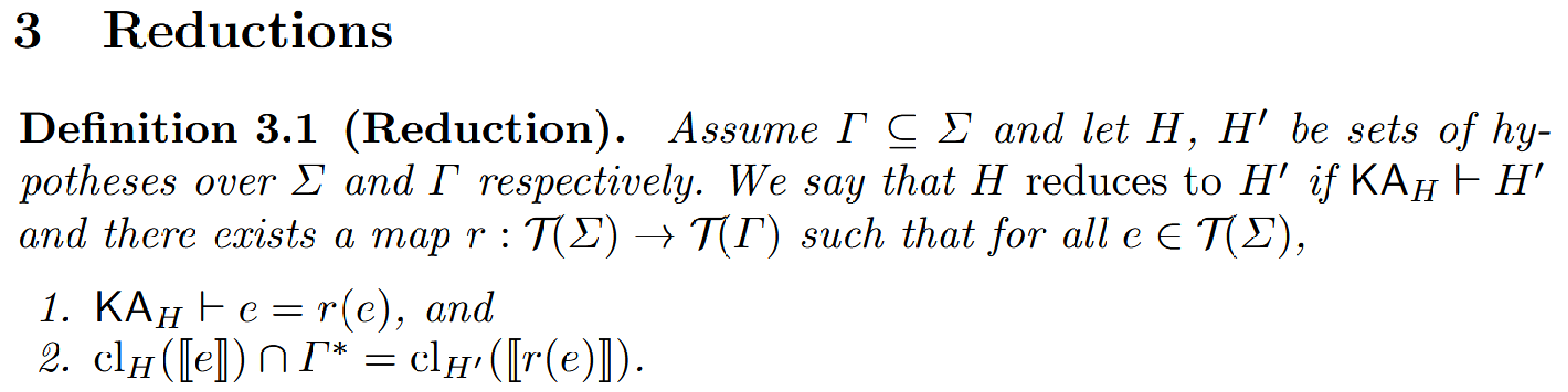
Lemma 4.1. *Suppose that we can construct a closure for every element of* $\mathcal{T}$. *If* $e, f \in \mathcal{T}$ *such that* $[\![e]\!]_{\mathsf{CKA}} = [\![f]\!]_{\mathsf{CKA}}$, *then* $e \equiv_{\mathsf{CKA}} f$.

Theorem 4.1. *Let* $e \in \mathcal{T}$. *We can construct a closure* $e{\downarrow}$ *of* e.

Kappé, Brunet, Silva & Zanasi, CONCUR 2018

→ Generic notion of <u>closure</u>

Reductions in the wild



3 Reductions

Definition 3.1 (Reduction). *Assume $\Gamma \subseteq \Sigma$ and let H, H' be sets of hypotheses over Σ and Γ respectively. We say that H* reduces to *H' if $\mathsf{KA}_H \vdash H'$ and there exists a map $r : \mathcal{T}(\Sigma) \to \mathcal{T}(\Gamma)$ such that for all $e \in \mathcal{T}(\Sigma)$,*

1. $\mathsf{KA}_H \vdash e = r(e)$, *and*
2. $\mathrm{cl}_H([\![e]\!]) \cap \Gamma^* = \mathrm{cl}_{H'}([\![r(e)]\!])$.

On Tools for Completeness of Kleene Algebra with Hypotheses. Pous, Rot & Wagemaker, RAMiCS 2021

V - Conclusions

What we've seen

- ☞ A generic notion of representations

What we've seen

- ☞ A generic notion of representations
- ☞ with a generic proof technique for completeness

What we've seen

- ☞ A generic notion of representations
- ☞ with a generic proof technique for completeness
- ☞ worth noting the instances of reduction we've seen only scratch the surface → more powerful in general

What we've seen

- A generic notion of representations
- with a generic proof technique for completeness
- <u>worth noting</u> the instances of reduction we've seen only scratch the surface → more powerful in general
- also true for the instances of representation :
  - specifications need not be terms
  - need not be closed by any operations
  - very little structure assumed (≠ institutions)

Also in the paper

- ☞ a Cartesian category of representations Repr
- ☞ parametric representations: functors Set → Repr
- ☞ lifting such functors to preorders, and then to representations

→ tiered representations

For the future

For the future

...many things, stay tuned!

Thanks

The most amazing audience in the world

Mikołaj Bojańczyk

Alexandra Silva

Lukas Mulder

Amazigh Amrane

Paul-André Melliès

Jana Wagemaker

Daniele Varacca

Georg Struth

Marguerite Zamansky

Guillaume Baudart

Uli Fahrenberg

Jurriaan Rot

Yusaku Nishimiya

Amina Doumane

Thomas Colcombet

Yoshiki Nakamura

Damien Pous

Tobias Kappé