Concurrent Kleene Algebra: Free Model and Completeness

Tobias Kappé Paul Brunet Alexandra Silva Fabio Zanasi

University College London

Leiden, 28 May 2018

Introduction

Let's write a program that outputs $n > 0$ space-separated ☺'s.

Introduction

Let's write a program that outputs $n > 0$ space-separated ☺'s.

```
$i := 1$
while $i < n$ do
    print ☺
    print ␣
    $i := i + 1$
end
print ☺
```

Introduction

Let's write a program that outputs $n > 0$ space-separated ☺'s.

```
$i := 1$
while $i < n$ do
    print ☺
    print ␣
    $i := i + 1$
end
print ☺
```

```
$i := 1$
print ☺
while $i < n$ do
    print ␣
    print ☺
    $i := i + 1$
end
```

Introduction

Let's write a program that outputs $n > 0$ space-separated ☺'s.

```
$i := 1$
while $i < n$ do
    print ☺
    print ␣
    $i := i + 1$
end
print ☺
```

```
$i := 1$
print ☺
while $i < n$ do
    print ␣
    print ☺
    $i := i + 1$
end
```

Are these programs equivalent?

Introduction

Programs are *expressions*

Introduction

Programs are *expressions*, thus we should be able to reason *equationally*.

Introduction

Programs are *expressions*, thus we should be able to reason *equationally*.

Kleene Algebra (KA) provides an *algebraic framework* to do this.

Introduction

program	expression
atomic action	$a, b, \ldots \in \Sigma$

Introduction

program	expression
atomic action	$a, b, \ldots \in \Sigma$
abort execution	0

Introduction

program	expression
atomic action	$a, b, \ldots \in \Sigma$
abort execution	0
no-operation	1

Introduction

program	expression
atomic action	$a, b, \ldots \in \Sigma$
abort execution	0
no-operation	1
nondeterministic choice	$e + f$

Introduction

program	expression
atomic action	$a, b, \ldots \in \Sigma$
abort execution	0
no-operation	1
nondeterministic choice	$e + f$
sequential composition	$e \cdot f$

Introduction

program	expression
atomic action	$a, b, \ldots \in \Sigma$
abort execution	0
no-operation	1
nondeterministic choice	$e + f$
sequential composition	$e \cdot f$
repetition	e^*

Introduction

```
$i := 1$
while $i < n$ do
    print ☺
    print ␣
    $i := i + 1$
end
print ☺
```

```
$i := 1$
print ☺
while $i < n$ do
    print ␣
    print ☺
    $i := i + 1$
end
```

Introduction

```
$i := 1$
while $i < n$ do
    print ☺
    print ␣
    $i := i + 1$
end
print ☺
```

$$\left.\right\} (☺ \cdot ␣)^* \cdot ☺$$

```
$i := 1$
print ☺
while $i < n$ do
    print ␣
    print ☺
    $i := i + 1$
end
```

Introduction

```
$i := 1$
while $i < n$ do
    print ☺
    print ␣
    $i := i + 1$
end
print ☺
```

$$(☺ \cdot ␣)^* \cdot ☺$$

```
$i := 1$
print ☺
while $i < n$ do
    print ␣
    print ☺
    $i := i + 1$
end
```

$$☺ \cdot (␣ \cdot ☺)^*$$

Introduction

Axioms of KA:

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

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

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

$$1 + e \cdot e^* \equiv e^* \qquad e \cdot f + g \leqq f \implies e^* \cdot g \leqq f$$

$$1 + e^* \cdot e \equiv e^* \qquad e \cdot f + g \leqq e \implies g \cdot f^* \leqq e$$

Introduction

Axioms of KA:

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

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

$e \cdot (f + g) \equiv e \cdot f + e \cdot g$ $\quad$ $(e + f) \cdot g \equiv e \cdot g + f \cdot g$

$1 + e \cdot e^* \equiv e^*$ $\quad$ $e \cdot f + g \leq f \implies e^* \cdot g \leq f$

$1 + e^* \cdot e \equiv e^*$ $\quad$ $e \cdot f + g \leq e \implies g \cdot f^* \leq e$

Introduction

$$☺ \cdot (␣ \cdot ☺)^* \equiv (☺ \cdot ␣)^* \cdot ☺$$

Introduction

Theorem (Kozen 1990)

The axioms for KA *are sound & **complete** for equivalence:*

$$e \equiv f \iff \mathcal{L}(e) = \mathcal{L}(f)$$

$\mathcal{L}(e)$ *is the regular language interpretation of* e*.*

Introduction

Theorem (Kozen 1990)

The axioms for KA *are sound &* ***complete*** *for equivalence:*

$$e \equiv f \iff \mathcal{L}(e) = \mathcal{L}(f)$$

$\mathcal{L}(e)$ *is the regular language interpretation of e.*

Upshot:

- to check KA equivalence is to check regular language equivalence
- through Kleene's theorem, this means checking DFA equivalence
- sophisticated (near-linear) algorithms exist to do this

Adding concurrency

Which *new* axioms do we need for parallel composition?

Adding concurrency

Which *new* axioms do we need for parallel composition?

$$e \parallel f \equiv f \parallel e$$

Adding concurrency

Which *new* axioms do we need for parallel composition?

$$e \parallel f \equiv f \parallel e \qquad\qquad e \parallel (f \parallel g) \equiv (e \parallel f) \parallel g$$

Adding concurrency

Which *new* axioms do we need for parallel composition?

$$e \parallel f \equiv f \parallel e \qquad e \parallel (f \parallel g) \equiv (e \parallel f) \parallel g$$

$$e \parallel 1 \equiv e$$

Adding concurrency

Which *new* axioms do we need for parallel composition?

$$e \parallel f \equiv f \parallel e \qquad e \parallel (f \parallel g) \equiv (e \parallel f) \parallel g$$

$$e \parallel 1 \equiv e \qquad e \parallel 0 \equiv 0$$

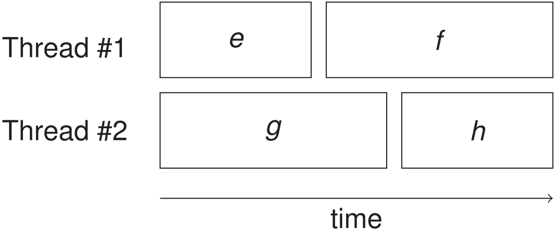
Adding concurrency

Which *new* axioms do we need for parallel composition?

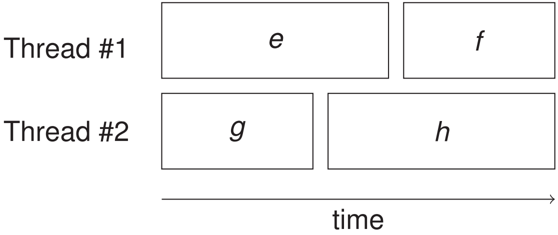
$$e \parallel f \equiv f \parallel e \qquad\qquad e \parallel (f \parallel g) \equiv (e \parallel f) \parallel g$$

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

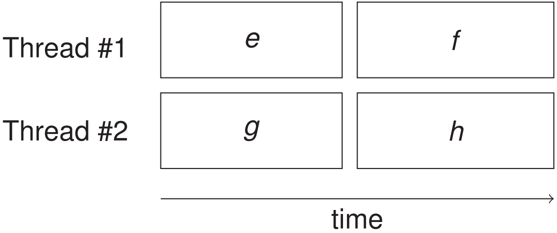
Adding concurrency



Adding concurrency



Adding concurrency

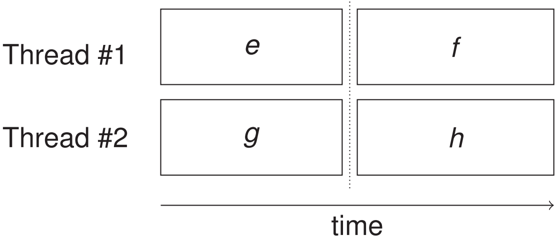


Thread #1 *e* *f*

Thread #2 *g* *h*

time

Adding concurrency



Thread #1

e

f

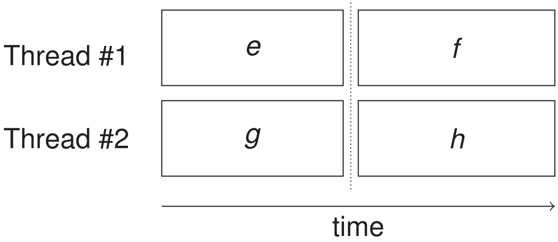
Thread #2

g

h

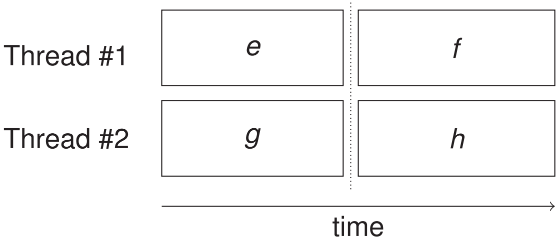
time

Adding concurrency



Equationally: $(e \parallel g) \cdot (f \parallel h) \leqq (e \cdot f) \parallel (g \cdot h)$.

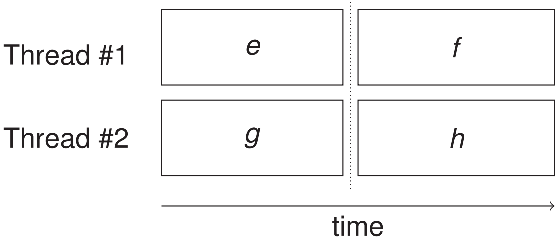
Adding concurrency



Equationally: $(e \parallel g) \cdot (f \parallel h) \leqq (e \cdot f) \parallel (g \cdot h)$.

$$p \leqq q \iff p + q \equiv q$$

Adding concurrency



Equationally: $(e \parallel g) \cdot (f \parallel h) \leqq (e \cdot f) \parallel (g \cdot h)$.

Nondeterministic interleaving as special case: $e \cdot f + f \cdot e \leqq e \parallel f$.

Adding concurrency

Question

Can we have a regular interpretation $[\![-]\!]$ *such that* $e \equiv f \iff [\![e]\!] = [\![f]\!]$*?*

Adding concurrency

Question

Can we have a regular interpretation $\llbracket - \rrbracket$ *such that* $e \equiv f \iff \llbracket e \rrbracket = \llbracket f \rrbracket$*?*

NB: $\llbracket - \rrbracket$ should generalize $\mathcal{L}(-)$: for $\parallel$-less terms, $\mathcal{L}(e)$ should resemble $\llbracket e \rrbracket$.

Regular interpretation: first attempt

Partially ordered multiset (pomset):

$$a \cdot b \cong a \longrightarrow b$$

Regular interpretation: first attempt

Partially ordered multiset (pomset):

$$a \cdot b \cong a \longrightarrow b \qquad\qquad a \parallel b \cong \begin{matrix} b \\ \\ a \end{matrix}$$

Regular interpretation: first attempt

Partially ordered multiset (pomset):

$$a \cdot b \cong a \longrightarrow b$$

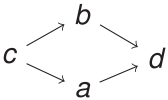
$$c \cdot (a \parallel b) \cong \begin{tikzcd} & b \\ c \arrow[ur] \arrow[dr] & \\ & a \end{tikzcd}$$

Regular interpretation: first attempt

Partially ordered multiset (pomset):

$$a \cdot b \cong a \longrightarrow b$$

$$c \cdot (a \parallel b) \cdot d \cong$$



Regular interpretation: first attempt

Partially ordered multiset (pomset):

$$a \cdot b \cong a \longrightarrow b$$

$$\begin{tikzcd} & & b \arrow[dr] & \\ c \cdot (a \parallel b) \cdot d \cong & c \arrow[ur] \arrow[dr] & & d \\ & & a \arrow[ur] & \end{tikzcd}$$

Composition lifts to *sets of pomsets* in the obvious way.

Regular interpretation: first attempt

Straightforward semantics: $(\!| - |\!) : \mathcal{T} \to 2^{\text{Pomsets}}$ given by

$$
\begin{array}{lll}
(\!| 0 |\!) = \emptyset & (\!| e + f |\!) = (\!| e |\!) \cup (\!| f |\!) & (\!| e^* |\!) = (\!| e |\!)^* \\
(\!| 1 |\!) = \{1\} & (\!| e \cdot f |\!) = (\!| e |\!) \cdot (\!| f |\!) & \\
(\!| a |\!) = \{a\} & (\!| e \parallel f |\!) = (\!| e |\!) \parallel (\!| f |\!) &
\end{array}
$$

Regular interpretation: first attempt

Straightforward semantics: $(\!| - |\!) : \mathcal{T} \to 2^{\text{Pomsets}}$ given by

$$
\begin{aligned}
(\!| 0 |\!) &= \emptyset & (\!| e + f |\!) &= (\!| e |\!) \cup (\!| f |\!) & (\!| e^* |\!) &= (\!| e |\!)^* \\
(\!| 1 |\!) &= \{1\} & (\!| e \cdot f |\!) &= (\!| e |\!) \cdot (\!| f |\!) & & \\
(\!| a |\!) &= \{a\} & (\!| e \parallel f |\!) &= (\!| e |\!) \parallel (\!| f |\!) & &
\end{aligned}
$$

Problem: $(\!| - |\!)$ is not sound for the exchange law.

Regular interpretation: first attempt

Straightforward semantics: $\llparenthesis - \rrparenthesis : \mathcal{T} \to 2^{\text{Pomsets}}$ given by

$$\begin{aligned} \llparenthesis 0 \rrparenthesis &= \emptyset & \llparenthesis e + f \rrparenthesis &= \llparenthesis e \rrparenthesis \cup \llparenthesis f \rrparenthesis & \llparenthesis e^* \rrparenthesis &= \llparenthesis e \rrparenthesis^* \\ \llparenthesis 1 \rrparenthesis &= \{1\} & \llparenthesis e \cdot f \rrparenthesis &= \llparenthesis e \rrparenthesis \cdot \llparenthesis f \rrparenthesis & & \\ \llparenthesis a \rrparenthesis &= \{a\} & \llparenthesis e \parallel f \rrparenthesis &= \llparenthesis e \rrparenthesis \parallel \llparenthesis f \rrparenthesis & & \end{aligned}$$

Problem: $\llparenthesis - \rrparenthesis$ is not sound for the exchange law.

For instance: $a \cdot b \leqq a \parallel b$ should imply that $\llparenthesis a \cdot b \rrparenthesis \subseteq \llparenthesis a \parallel b \rrparenthesis$, but

$$\llparenthesis a \cdot b \rrparenthesis = \left\{ \; a \longrightarrow b \; \right\} \qquad\qquad \llparenthesis a \parallel b \rrparenthesis = \left\{ \; a \qquad b \; \right\}$$

Regular interpretation: first attempt

Axioms to build $\approx$ are axioms for $\equiv$, minus exchange law.

Regular interpretation: first attempt

Axioms to build $\approx$ are axioms for $\equiv$, minus exchange law.

Theorem (Laurence and Struth 2014)

The axioms for $\approx$ are sound & complete w.r.t. $(\!| - |\!)$:

$$e \approx f \iff (\!| e |\!) = (\!| f |\!)$$

Regular interpretation: second attempt

We define the *subsumption order* $\sqsubseteq$ on pomsets.

Intuition: $U \sqsubseteq V$ if

- i U and V have the same events, and
- ii U has all order in V (and possibly more)

Regular interpretation: second attempt

We define the *subsumption order* $\sqsubseteq$ on pomsets.

Intuition: $U \sqsubseteq V$ if

- i U and V have the same events, and
- ii U has all order in V (and possibly more)

For example:

$$a \longrightarrow b \sqsubseteq a \qquad b$$

Regular interpretation: second attempt

We define the *subsumption order* $\sqsubseteq$ on pomsets.

Intuition: $U \sqsubseteq V$ if

i. U and V have the same events, and

ii. U has all order in V (and possibly more)

For example:

$$\begin{tikzcd}
a \arrow[r] \arrow[ddr] & c & & a \arrow[r] & c \\
& & \sqsubseteq & & \\
b \arrow[r] \arrow[uur] & d & & b \arrow[r] & d
\end{tikzcd}$$

Regular interpretation: second attempt

“Fixed” semantics: $[\![e]\!] = (\!| e |\!)\downarrow$.

downward closure w.r.t. $\sqsubseteq$

Regular interpretation: second attempt

“Fixed” semantics: $[\![e]\!] = (\!| e |\!)\!\downarrow$.

Previous problem no longer occurs:

$$[\![a \cdot b]\!] = \left\{ \; a \longrightarrow b \; \right\} \subseteq \left\{ \; a \longrightarrow b \; , \; a \longleftarrow b \; , \; a \qquad b \; \right\} = [\![a \parallel b]\!]$$

Regular interpretation: second attempt

“Fixed” semantics: $[\![e]\!] = (\!|\, e \,|\!)\!\downarrow$.

Previous problem no longer occurs:

$$[\![a \cdot b]\!] = \left\{\; a \longrightarrow b \;\right\} \subseteq \left\{\; a \longrightarrow b \;,\; a \longleftarrow b \;,\; a \qquad b \;\right\} = [\![a \parallel b]\!]$$

Lemma (Hoare et al. 2009)

The axioms for $\equiv$ *are sound w.r.t.* $[\![-]\!]$, *i.e.,* $e \equiv f$ *implies* $[\![e]\!] = [\![f]\!]$.

Closure

Definition

Let $e \in \mathcal{T}$; a *closure* of e is a term $e{\downarrow}$ such that

1. $e{\downarrow} \equiv e$
2. $[\![e]\!] = (\!| e{\downarrow} |\!)$

Closure

Definition

Let $e \in \mathcal{T}$; a *closure* of e is a term $e{\downarrow}$ such that

1. $e{\downarrow} \equiv e$
2. $[\![e]\!] = (\!| e{\downarrow} |\!)$

Lemma (Laurence and Struth 2017)

If closures exist for all terms, then $\equiv$ *is complete w.r.t.* $[\![-]\!]$*, i.e.,* $[\![e]\!] = [\![f]\!]$ *implies* $e \equiv f$.

Closure

Definition

Let $e \in \mathcal{T}$; a *closure* of e is a term $e{\downarrow}$ such that

1. $e{\downarrow} \equiv e$
2. $[\![e]\!] = (\!| e{\downarrow} |\!)$

Lemma (Laurence and Struth 2017)

If closures exist for all terms, then $\equiv$ *is complete w.r.t.* $[\![-]\!]$, *i.e.,* $[\![e]\!] = [\![f]\!]$ *implies* $e \equiv f$.

Proof.

If $[\![e]\!] = [\![f]\!]$, ☐

Closure

Definition

Let $e \in \mathcal{T}$; a *closure* of e is a term $e{\downarrow}$ such that

1. $e{\downarrow} \equiv e$
2. $[\![e]\!] = (\!|\, e{\downarrow} \,|\!)$

Lemma (Laurence and Struth 2017)

If closures exist for all terms, then $\equiv$ *is complete w.r.t.* $[\![-]\!]$*, i.e.,* $[\![e]\!] = [\![f]\!]$ *implies* $e \equiv f$*.*

Proof.

If $[\![e]\!] = [\![f]\!]$, then $(\!|\, e{\downarrow} \,|\!) = (\!|\, f{\downarrow} \,|\!)$, $\square$

Closure

Definition

Let $e \in \mathcal{T}$; a *closure* of e is a term $e{\downarrow}$ such that

1. $e{\downarrow} \equiv e$
2. $[\![e]\!] = (\!| e{\downarrow} |\!)$

Lemma (Laurence and Struth 2017)

If closures exist for all terms, then $\equiv$ is complete w.r.t. $[\![-]\!]$, i.e., $[\![e]\!] = [\![f]\!]$ implies $e \equiv f$.

Proof.

If $[\![e]\!] = [\![f]\!]$, then $(\!| e{\downarrow} |\!) = (\!| f{\downarrow} |\!)$, thus $e{\downarrow} \approx f{\downarrow}$. ☐

Closure

Definition

Let $e \in \mathcal{T}$; a *closure* of e is a term $e{\downarrow}$ such that

1. $e{\downarrow} \equiv e$
2. $\llbracket e \rrbracket = \llparenthesis\, e{\downarrow} \,\rrparenthesis$

Lemma (Laurence and Struth 2017)

If closures exist for all terms, then $\equiv$ *is complete w.r.t.* $\llbracket - \rrbracket$*, i.e.,* $\llbracket e \rrbracket = \llbracket f \rrbracket$ *implies* $e \equiv f$*.*

Proof.

If $\llbracket e \rrbracket = \llbracket f \rrbracket$, then $\llparenthesis\, e{\downarrow} \,\rrparenthesis = \llparenthesis\, f{\downarrow} \,\rrparenthesis$, thus $e{\downarrow} \approx f{\downarrow}$. Therefore, $e{\downarrow} \equiv f{\downarrow}$. □

Closure

Definition

Let $e \in \mathcal{T}$; a *closure* of e is a term $e{\downarrow}$ such that

1. $e{\downarrow} \equiv e$
2. $\llbracket e \rrbracket = \llparenthesis\, e{\downarrow} \,\rrparenthesis$

Lemma (Laurence and Struth 2017)

If closures exist for all terms, then $\equiv$ *is complete w.r.t.* $\llbracket - \rrbracket$*, i.e.,* $\llbracket e \rrbracket = \llbracket f \rrbracket$ *implies* $e \equiv f$*.*

Proof.

If $\llbracket e \rrbracket = \llbracket f \rrbracket$, then $\llparenthesis\, e{\downarrow} \,\rrparenthesis = \llparenthesis\, f{\downarrow} \,\rrparenthesis$, thus $e{\downarrow} \approx f{\downarrow}$. Therefore, $e \equiv e{\downarrow} \equiv f{\downarrow} \equiv f$. □

Main contribution

Theorem

If $e \in \mathcal{T}$, then we can ***compute*** *a term $e{\downarrow}$ that is a closure of e.*

Main contribution

Theorem

If $e \in \mathcal{T}$, then we can ***compute*** *a term $e{\downarrow}$ that is a closure of e.*

Corollary

The axioms for CKA *are sound & complete w.r.t.* $[\![-]\!]$*:*

$$e \equiv f \iff [\![e]\!] = [\![f]\!]$$

Main contribution

Theorem

If $e \in \mathcal{T}$, then we can ***compute*** *a term $e{\downarrow}$ that is a closure of e.*

Corollary

The axioms for CKA *are sound & complete w.r.t.* $[\![-]\!]$*:*

$$e \equiv f \iff [\![e]\!] = [\![f]\!]$$

The latter can be decided; c.f. [Brunet, Pous, and Struth 2017].

Further work

- Explore coalgebraic perspective:
  - Efficient equivalence checking through bisimulation?
  - Can completeness be shown coalgebraically?
- Add “parallel star” operator — closure method does not apply.
- Extend *Kleene Algebra with Tests* (KAT) to add concurrency.
- Extend extend NetKAT with concurrency.

Thank you for your attention

GoNeCo

Implementation: `https://doi.org/10.5281/zenodo.926651`.

Extended paper: `https://arxiv.org/abs/1710.02787`.

Bonus: computing the closure

So, how does one compute a closure?

Bonus: computing the closure

So, how does one compute a closure?

Lemma

If e, f have closures $e{\downarrow}$ and $f{\downarrow}$ respectively, then

1. $e{\downarrow} + f{\downarrow}$ *is a closure of* $e + f$
2. $e{\downarrow} \cdot f{\downarrow}$ *is a closure of* $e \cdot f$
3. $e{\downarrow}^*$ *is a closure of* e^*

Bonus: computing the closure

So, how does one compute a closure?

Lemma

If e, f have closures $e{\downarrow}$ and $f{\downarrow}$ respectively, then

1. $e{\downarrow} + f{\downarrow}$ *is a closure of* $e + f$
2. $e{\downarrow} \cdot f{\downarrow}$ *is a closure of* $e \cdot f$
3. $e{\downarrow}^*$ *is a closure of* e^*

One case remains: parallel composition.

Bonus: computing the closure

Sketch: given $e \parallel f$, apply exchange law syntactically, “in the limit”.

Bonus: computing the closure

Sketch: given $e \parallel f$, apply exchange law syntactically, “in the limit”.

For instance: if $e = a \cdot b$ and $f = c \cdot d$:

- $(a \parallel c) \cdot (b \parallel d) \leqq e \parallel f$ $(e = a \bullet b, f = c \bullet d)$

Bonus: computing the closure

Sketch: given $e \parallel f$, apply exchange law syntactically, “in the limit”.

For instance: if $e = a \cdot b$ and $f = c \cdot d$:

- $(a \parallel c) \cdot (b \parallel d) \leqq e \parallel f$ $\qquad (e = a \bullet b, f = c \bullet d)$
- $(a \parallel 1) \cdot (b \parallel (c \cdot d)) \leqq e \parallel f$ $\qquad (e = a \bullet b, f = 1 \bullet c \cdot d)$

Bonus: computing the closure

Sketch: given $e \parallel f$, apply exchange law syntactically, “in the limit”.

For instance: if $e = a \cdot b$ and $f = c \cdot d$:

- $(a \parallel c) \cdot (b \parallel d) \leqq e \parallel f$ $\qquad (e = a \bullet b, f = c \bullet d)$
- $(a \parallel 1) \cdot (b \parallel (c \cdot d)) \leqq e \parallel f$ $\qquad (e = a \bullet b, f = 1 \bullet c \cdot d)$
- $(1 \parallel c) \cdot ((a \cdot b) \parallel d) \leqq e \parallel f$ $\qquad (e = 1 \bullet a \cdot b, f = c \bullet d)$
- . . .

Bonus: computing the closure

Sketch: given $e \parallel f$, apply exchange law syntactically, “in the limit”.

For instance: if $e = a \cdot b$ and $f = c \cdot d$:

- $(a \parallel c) \cdot (b \parallel d) \leqq e \parallel f$ $\quad (e = a \bullet b, f = c \bullet d)$
- $(a \parallel 1) \cdot (b \parallel (c \cdot d)) \leqq e \parallel f$ $\quad (e = a \bullet b, f = 1 \bullet c \cdot d)$
- $(1 \parallel c) \cdot ((a \cdot b) \parallel d) \leqq e \parallel f$ $\quad (e = 1 \bullet a \cdot b, f = c \bullet d)$
- . . .

Goal: find enough of these terms to cover all pomsets in $[\![e \parallel f]\!]$.

Bonus: computing the closure

Obstacles to overcome:

- How to split terms e and f into heads and tails?

Bonus: computing the closure

Obstacles to overcome:

- How to split terms *e* and *f* into heads and tails?
- What to do about recursion? For instance,

$$(e \parallel f) \cdot (e^* \parallel f^*) \leqq e^* \parallel f^*$$

Bonus: computing the closure

Obstacles to overcome:

- How to split terms *e* and *f* into heads and tails? ☞ splicing relations
- What to do about recursion? For instance,

$$(e \parallel f) \cdot (e^* \parallel f^*) \leqq e^* \parallel f^*$$

Bonus: computing the closure

Obstacles to overcome:

- How to split terms e and f into heads and tails? ☞ splicing relations
- What to do about recursion? For instance, ☞ fixpoints of inequations

$$(e \parallel f) \cdot (e^* \parallel f^*) \leqq e^* \parallel f^*$$

Bonus: computing the closure

Definition

Let $e \in \mathcal{T}$. We define $\nabla_e \subseteq \mathcal{T} \times \mathcal{T}$ as the smallest relation such that

$$\frac{}{1 \nabla_1 1} \qquad \frac{}{a \nabla_a 1} \qquad \frac{}{1 \nabla_a a} \qquad \frac{}{1 \nabla_{e^*} 1} \qquad \frac{\ell \nabla_e r}{\ell \nabla_{e+f} r}$$

$$\frac{\ell \nabla_e r}{\ell \nabla_{e \cdot f} r \cdot f} \qquad \frac{\ell \nabla_e r}{\ell \nabla_{e \cdot f} r \cdot f} \qquad \frac{\ell \nabla_f r}{e \cdot \ell \nabla_{e \cdot f} r} \qquad \frac{\ell_0 \nabla_e r_0 \qquad \ell_1 \nabla_f r_1}{\ell_0 \parallel \ell_1 \nabla_{e \parallel f} r_0 \parallel r_1} \qquad \frac{\ell \nabla_e r}{e^* \cdot \ell \nabla_{e^*} r \cdot e^*}$$

Lemma

Let $e \in \mathcal{T}$ and $U \cdot V \in [\![e]\!]_{\mathrm{WCKA}}$; there exist $\ell \nabla_e r$ such that $U \in [\![\ell]\!]$ and $V \in [\![r]\!]$.

Bonus: computing the closure

Suppose that for all $g, h \in \mathcal{T}$, we have that $X_{g \parallel h}$ is a closure of $g \parallel h$.

Then we find

$$e \parallel f + \sum_{\substack{\ell_e \, \nabla_e \, r_e \\ \ell_f \, \nabla_f \, r_f}} (\ell_e \parallel \ell_f) \cdot (r_e \parallel r_f) \leqq X_{e \parallel f}$$

Bonus: computing the closure

Suppose that for all $g, h \in \mathcal{T}$, we have that $X_{g \parallel h}$ is a closure of $g \parallel h$.

Then we find

$$e \parallel f + \sum_{\substack{\ell_e \;\nabla_e\; r_e \\ \ell_f \;\nabla_f\; r_f}} (\ell_e \parallel \ell_f) \cdot X_{r_e \parallel r_f} \leqq X_{e \parallel f}$$

Bonus: computing the closure

Suppose that for all $g, h \in \mathcal{T}$, we have that $X_{g \parallel h}$ is a closure of $g \parallel h$.

Then we find

$$e \parallel f + \sum_{\substack{\ell_e \; \nabla_e \; r_e \\ \ell_f \; \nabla_f \; r_f}} (\ell_e \parallel \ell_f) \cdot X_{r_e \parallel r_f} \leqq X_{e \parallel f}$$

For $X_{r_e \parallel r_f}$, we find another inequation, et cetera…

Bonus: computing the closure

Suppose that for all $g, h \in \mathcal{T}$, we have that $X_{g \parallel h}$ is a closure of $g \parallel h$.

Then we find

$$e \parallel f + \sum_{\substack{\ell_e \triangledown_e r_e \\ \ell_f \triangledown_f r_f}} (\ell_e \parallel \ell_f) \cdot X_{r_e \parallel r_f} \leqq X_{e \parallel f}$$

For $X_{r_e \parallel r_f}$, we find another inequation, et cetera. . .

Lemma

Continuing this, we get a finite system of inequations $\langle M, \vec{b} \rangle_{e \parallel f}$.

Bonus: computing the closure

Theorem

Let $e \otimes f$ *be the least solution to* $X_{e \parallel f}$ *in* $\langle M, \vec{b} \rangle_{e \parallel f}$*. Then the following hold:*

1. $e \otimes f \equiv e \parallel f$
2. $(\!| e \otimes f |\!) = [\![e \parallel f]\!]$

In other words, $e \otimes f$ *is a closure of* $e \parallel f$*.*