(Abstract) GSOS for Trace Equivalence

Séminaire LIMD

Robin Jourde[*], Stelios Tsampas[†], Sergey Goncharov[‡], Henning Urbat[§], Pouya Partow[‡], Jonas Forster[§]

27 Mars 2025

[*]ENS de Lyon – robin.jourde@ens-lyon.fr
[†]Syddansk Universitet
[‡]University of Birmingham
[§]**Friedrich-Alexander-Universität Erlangen-Nürnberg**

Outline

1 Trace equivalence for concrete systems

1.1 Processes and LTSs

- study the **behaviour of processes**

1.1 Processes and LTSs

- study the **behaviour of processes**
- running example: **labelled transition systems** (LTS) with explicit termination

1.1 Processes and LTSs

- study the **behaviour of processes**
- running example: **labelled transition systems** (LTS) with explicit termination
  - set of actions/labels A

1.1 Processes and LTSs

- study the **behaviour of processes**
- running example: **labelled transition systems** (LTS) with explicit termination
  - set of actions/labels A
  - set of processes/states X with some operations
    - a special state $0 \in X$
    - for each action $a \in A$ and process $x \in X$, a process $a.x \in X$

1.1 Processes and LTSs

- study the **behaviour of processes**
- running example: **labelled transition systems** (LTS) with explicit termination
  - set of actions/labels A
  - set of processes/states X with some operations
    - a special state $0 \in X$
    - for each action $a \in A$ and process $x \in X$, a process $a.x \in X$
    - and at will: a binary operation $+$, unary operations !, ? …

1.1 Processes and LTSs

- study the **behaviour of processes**
- running example: **labelled transition systems** (LTS) with explicit termination
  - set of actions/labels A
  - set of processes/states X with some operations
    - a special state $0 \in X$
    - for each action $a \in A$ and process $x \in X$, a process $a.x \in X$
    - and at will: a binary operation $+$, unary operations !, ? ...

Example. For $a, b, c, d \in A$

- 0 $\quad$ $a.0$ $\quad$ $d.a.0$

1.1 Processes and LTSs

- study the **behaviour of processes**
- running example: **labelled transition systems** (LTS) with explicit termination
  - set of actions/labels A
  - set of processes/states X with some operations
    - a special state $0 \in X$
    - for each action $a \in A$ and process $x \in X$, a process $a.x \in X$
    - and at will: a binary operation $+$, unary operations $!$, $?$ …

Example. For $a, b, c, d \in A$

- 0 $\quad a.0 \quad d.a.0$
- $a.0 + b.c.0$ $\quad ?\,(c.d.0)$

1.1 Processes and LTSs

- study the **behaviour of processes**
- running example: **labelled transition systems** (LTS) with explicit termination
  - set of actions/labels A
  - set of processes/states X with some operations
    - a special state $0 \in X$
    - for each action $a \in A$ and process $x \in X$, a process $a.x \in X$
    - and at will: a binary operation $+$, unary operations $!$, $?$ …

Example. For $a, b, c, d \in A$

- 0 $\quad$ $a.0$ $\quad$ $d.a.0$
- $a.0 + b.c.0$ $\quad$ $?\,(c.d.0)$
- $a.(b.a.0 + ?\,(d.a.c.0))$

1.1 Processes and LTSs

- study the **behaviour of processes**
- running example: **labelled transition systems** (LTS) with explicit termination
  - set of actions/labels A
  - set of processes/states X with some operations
    - a special state $0 \in X$
    - for each action $a \in A$ and process $x \in X$, a process $a.x \in X$
    - and at will: a binary operation $+$, unary operations !, ? ...

Example. For $a, b, c, d \in A$

- 0 — $a.0 = a$ — $d.a.0 = da$
- $a.0 + b.c.0 = a + bc$ — $?\,(c.d.0) = ?\,cd$
- $a.(b.a.0 + ?\,(d.a.c.0)) = a(ba + ?\,dac)$

1.1 Processes and LTSs

- **behaviour** of a process x:

1.1 Processes and LTSs

- **behaviour** of a process x:
  - can progress emitting a label $a \in A$ and continuing with process y: "$x \xrightarrow{a} y$"
  - or terminate: "$x \downarrow$"

- **behaviour** of a process x:
  - can progress emitting a label $a \in A$ and continuing with process y: "$x \xrightarrow{a} y$" $\rightsquigarrow (a, y) \in k(x)$
  - or terminate: "$x \downarrow$" $\rightsquigarrow \star \in k(x)$
  - collect everything in a map $k : X \to \mathcal{P}(A \times X + \{\star\})$

1.1 Processes and LTSs

- **behaviour** of a process x:
  - can progress emitting a label $a \in A$ and continuing with process y: "$x \xrightarrow{a} y$" $\rightsquigarrow (a, y) \in k(x)$
  - or terminate: "$x \downarrow$" $\rightsquigarrow \star \in k(x)$
  - collect everything in a map $k : X \to \mathcal{P}(A \times X + \{\star\})$
- given by the **rules**:

$$\frac{}{0 \downarrow}$$

1.1 Processes and LTSs

- **behaviour** of a process x:
  - can progress emitting a label $a \in A$ and continuing with process y: "$x \xrightarrow{a} y$" $\rightsquigarrow (a, y) \in k(x)$
  - or terminate: "$x \downarrow$" $\rightsquigarrow \star \in k(x)$
  - collect everything in a map $k : X \to \mathcal{P}(A \times X + \{\star\})$
- given by the **rules**:

$$\frac{}{0 \downarrow} \qquad \frac{}{a.t \xrightarrow{a} t}\,\forall a$$

- **behaviour** of a process x:
  - can progress emitting a label $a \in A$ and continuing with process y: "$x \xrightarrow{a} y$" $\rightsquigarrow (a, y) \in k(x)$
  - or terminate: "$x \downarrow$" $\rightsquigarrow \star \in k(x)$
  - collect everything in a map $k : X \to \mathcal{P}(A \times X + \{\star\})$
- given by the **rules**:

$$\frac{}{0 \downarrow} \qquad \frac{}{a.t \xrightarrow{a} t}\forall a \qquad \frac{t \xrightarrow{a} t'}{t + u \xrightarrow{a} t'}\forall a \qquad \frac{u \xrightarrow{a} u'}{t + u \xrightarrow{a} u'}\forall a \qquad \frac{t \downarrow}{t + u \downarrow} \qquad \frac{u \downarrow}{t + u \downarrow}$$

1.1 Processes and LTSs

$$\frac{}{0\downarrow} \qquad \frac{}{a.t \xrightarrow{a} t}\forall a \qquad \frac{t \xrightarrow{a} t'}{t+u \xrightarrow{a} t'}\forall a \qquad \frac{u \xrightarrow{a} u'}{t+u \xrightarrow{a} u'}\forall a \qquad \frac{t\downarrow}{t+u\downarrow} \qquad \frac{u\downarrow}{t+u\downarrow}$$

Example.

1.1 Processes and LTSs

$$\frac{}{0\downarrow} \qquad \frac{}{a.t \xrightarrow{a} t}\forall a \qquad \frac{t \xrightarrow{a} t'}{t+u \xrightarrow{a} t'}\forall a \qquad \frac{u \xrightarrow{a} u'}{t+u \xrightarrow{a} u'}\forall a \qquad \frac{t\downarrow}{t+u\downarrow} \qquad \frac{u\downarrow}{t+u\downarrow}$$

Example.

$$\begin{tikzcd}
 & a(b+c) \arrow[d, "a"'] & \\
 & b+c \arrow[dl, "b"'] \arrow[dr, "c"] & \\
0 & & 0
\end{tikzcd}$$

$$\frac{}{0\downarrow} \qquad \frac{}{a.t \xrightarrow{a} t}\,\forall a \qquad \frac{t \xrightarrow{a} t'}{t+u \xrightarrow{a} t'}\,\forall a \qquad \frac{u \xrightarrow{a} u'}{t+u \xrightarrow{a} u'}\,\forall a \qquad \frac{t\downarrow}{t+u\downarrow} \qquad \frac{u\downarrow}{t+u\downarrow}$$

Example.

$$\begin{tikzcd} & a(b+c) \arrow[d, "a"'] & \\ & b+c \arrow[dl, "b"'] \arrow[dr, "c"] & \\ 0 & & 0 \end{tikzcd} \qquad \begin{tikzcd} & ab+ac \arrow[dl, "a"'] \arrow[dr, "a"] & \\ b \arrow[d, "b"'] & & c \arrow[d, "c"'] \\ 0 & & 0 \end{tikzcd}$$

$$\frac{}{0\downarrow} \qquad \frac{}{a.t \xrightarrow{a} t}\,\forall a \qquad \frac{t \xrightarrow{a} t'}{t+u \xrightarrow{a} t'}\,\forall a \qquad \frac{u \xrightarrow{a} u'}{t+u \xrightarrow{a} u'}\,\forall a \qquad \frac{t\downarrow}{t+u\downarrow} \qquad \frac{u\downarrow}{t+u\downarrow}$$

Example.

$$\begin{tikzcd}
 & a(b+c) \arrow[d, "a"'] & \\
 & b+c \arrow[dl, "b"'] \arrow[dr, "c"] & \\
0 & & 0
\end{tikzcd}
\qquad
\begin{tikzcd}
 & ab+ac \arrow[dl, "a"'] \arrow[dr, "a"] & \\
b \arrow[d, "b"'] & & c \arrow[d, "c"'] \\
0 & & 0
\end{tikzcd}
\qquad
\begin{tikzcd}
 & \boxed{0+(a+aa)} \arrow[dl, "a"'] \arrow[dr, "a"] & \\
0 & & a \arrow[d, "a"'] \\
 & & 0
\end{tikzcd}$$

1.2 Program equivalences

- how to compare programs? what does it mean to be “the same” program?

1.2 Program equivalences

- how to compare programs? what does it mean to be “the same” program?

Example. $f(x) := x + x$ and $g(x) := 2 \times x$

1.2 Program equivalences

- how to compare programs? what does it mean to be “the same” program?

Example. $f(x) := x + x$ and $g(x) := 2 \times x$

- many different notions (linear time/branching time spectrum): bisimilarity, trace equivalence

1.2 Program equivalences

- how to compare programs? what does it mean to be "the same" program?

Example. $f(x) := x + x$ and $g(x) := 2 \times x$

- many different notions (linear time/branching time spectrum): bisimilarity, trace equivalence
- important property: contextual equivalence and **congruence**

$$(\forall i, x_i \sim y_i) \Rightarrow \sigma(x_1..x_n) \sim \sigma(y_1...y_n)$$

1.2 Program equivalences

- how to compare programs? what does it mean to be “the same” program?

Example. $f(x) := x + x$ and $g(x) := 2 \times x$

- many different notions (linear time/branching time spectrum): bisimilarity, trace equivalence
- important property: contextual equivalence and **congruence**

$$(\forall i, x_i \sim y_i) \Rightarrow \sigma(x_1..x_n) \sim \sigma(y_1...y_n)$$

Example. $x \sim y \Rightarrow a.x \sim a.y$, or $x_1 \sim y_1 \wedge x_2 \sim y_2 \Rightarrow x_1 + x_2 \sim y_1 + y_2$

1.3 Trace and trace equivalence

- different flavors: partial vs. complete, finite vs. infinite

1.3 Trace and trace equivalence

- different flavors: partial vs. complete, finite vs. infinite
- complete infinitary traces:

$$\mathrm{tr}(x) = \bigcup_{n\in\mathbb{N}} \left\{ w \in A^n \,\middle|\, x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \dots \xrightarrow{w_n} x_n \downarrow \right\} \cup \left\{ w \in A^\omega \,\middle|\, x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \dots \right\} \subseteq A^\infty$$

  Set of finite and infinite words with letters in A corresponding to all possible (terminating or infinite) executions.

1.3 Trace and trace equivalence

- different flavors: partial vs. complete, finite vs. infinite
- complete infinitary traces:

$$\operatorname{tr}(x) = \bigcup_{n\in\mathbb{N}} \left\{ w \in A^n \,\middle|\, x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \dots \xrightarrow{w_n} x_n \downarrow \right\} \cup \left\{ w \in A^\omega \,\middle|\, x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \dots \right\} \subseteq A^\infty$$

  Set of finite and infinite words with letters in A corresponding to all possible (terminating or infinite) executions.

Remark. tr is the greatest map such that $\varepsilon \in \operatorname{tr}(x) \Leftrightarrow x \downarrow$ and $a.w \in \operatorname{tr}(x) \Leftrightarrow x \xrightarrow{a} y \land w \in \operatorname{tr}(y)$ ("*coalgebra morphism*")

1.3 Trace and trace equivalence

- different flavors: partial vs. complete, finite vs. infinite
- complete infinitary traces:

$$\operatorname{tr}(x) = \bigcup_{n \in \mathbb{N}} \left\{ w \in A^n \;\middle|\; x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \dots \xrightarrow{w_n} x_n \downarrow \right\} \cup \left\{ w \in A^\omega \;\middle|\; x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \dots \right\} \subseteq A^\infty$$

  Set of finite and infinite words with letters in A corresponding to all possible (terminating or infinite) executions.

Remark. tr is the greatest map such that $\varepsilon \in \operatorname{tr}(x) \Leftrightarrow x \downarrow$ and $a.w \in \operatorname{tr}(x) \Leftrightarrow x \xrightarrow{a} y \wedge w \in \operatorname{tr}(y)$ ("*coalgebra morphism*")

- trace equivalence: $x \underset{\text{tr}}{\sim} y \Leftrightarrow \operatorname{tr}(x) = \operatorname{tr}(y)$

1.3 Trace and trace equivalence

$$\mathrm{tr}(x) = \bigcup_{n \in \mathbb{N}} \left\{ w \in A^n \;\middle|\; x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \dots \xrightarrow{w_n} x_n \downarrow \right\} \cup \left\{ w \in A^\omega \;\middle|\; x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \dots \right\}$$

Example. tr $a(b + c) =$

$$\begin{tikzcd}
& a(b+c) \arrow[d, "a"'] & \\
& b+c \arrow[dl, "b"'] \arrow[dr, "c"] & \\
0 & & 0
\end{tikzcd}$$

1.3 Trace and trace equivalence

$$\mathrm{tr}(x) = \bigcup_{n\in\mathbb{N}} \left\{ w \in A^n \;\middle|\; x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \dots \xrightarrow{w_n} x_n \downarrow \right\} \cup \left\{ w \in A^\omega \;\middle|\; x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \dots \right\}$$

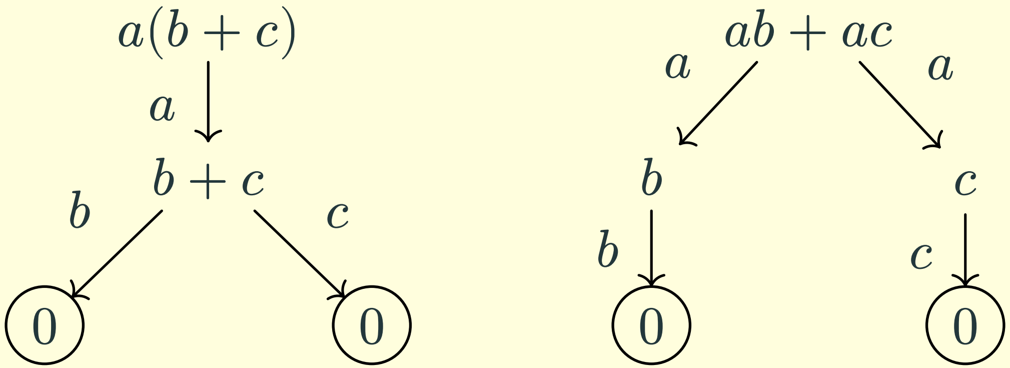
Example. $\mathrm{tr}\; a(b+c) = \{ab, ac\}$,

$$\begin{tikzcd}
& a(b+c) \arrow[d, "a"'] & \\
& b+c \arrow[dl, "b"'] \arrow[dr, "c"] & \\
0 & & 0
\end{tikzcd}$$

1.3 Trace and trace equivalence

$$\mathrm{tr}(x) = \bigcup_{n\in\mathbb{N}} \left\{ w \in A^n \,\middle|\, x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \dots \xrightarrow{w_n} x_n \downarrow \right\} \cup \left\{ w \in A^\omega \,\middle|\, x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \dots \right\}$$

Example. $\mathrm{tr}\ a(b+c) = \{ab, ac\}$, $\mathrm{tr}\ (ab+ac) =$



1.3 Trace and trace equivalence

$$\mathrm{tr}(x) = \bigcup_{n \in \mathbb{N}} \left\{ w \in A^n \,\middle|\, x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \dots \xrightarrow{w_n} x_n \downarrow \right\} \cup \left\{ w \in A^\omega \,\middle|\, x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \dots \right\}$$

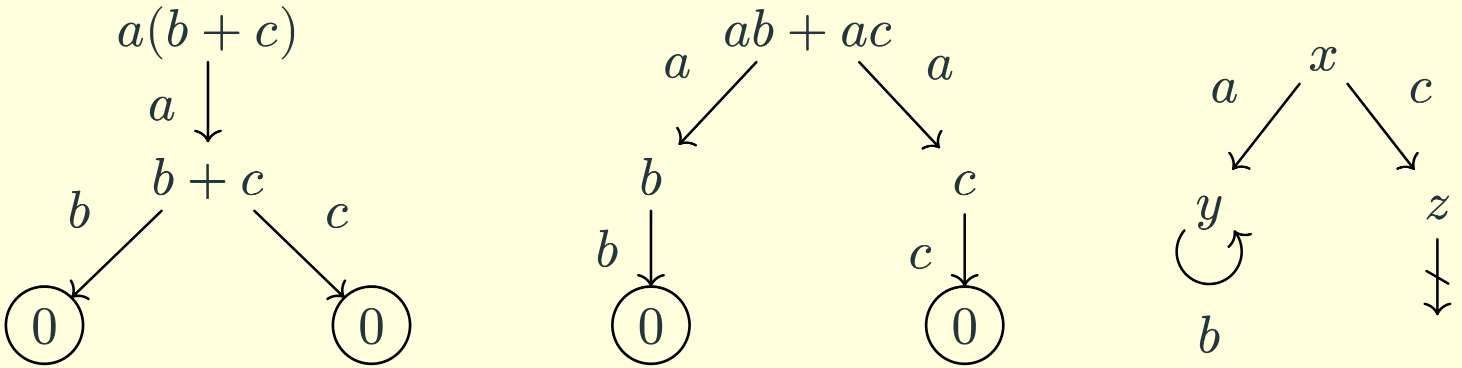
Example. tr $a(b+c) = \{ab, ac\}$, tr $(ab + ac) = \{ab, ac\}$,

$$\begin{tikzcd} & a(b+c) \arrow[d, "a"'] & \\ & b+c \arrow[dl, "b"'] \arrow[dr, "c"] & \\ 0 & & 0 \end{tikzcd} \quad \underset{\mathrm{tr}}{\sim} \quad \begin{tikzcd} & ab+ac \arrow[dl, "a"'] \arrow[dr, "a"] & \\ b \arrow[d, "b"'] & & c \arrow[d, "c"'] \\ 0 & & 0 \end{tikzcd}$$

1.3 Trace and trace equivalence

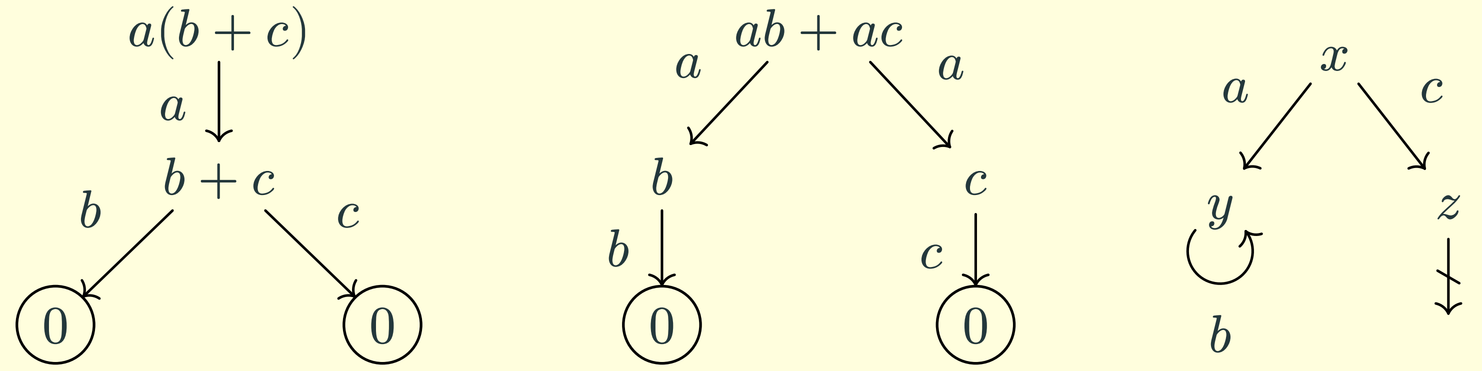
$$\mathrm{tr}(x) = \bigcup_{n \in \mathbb{N}} \left\{ w \in A^n \,\middle|\, x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \ldots \xrightarrow{w_n} x_n \downarrow \right\} \cup \left\{ w \in A^\omega \,\middle|\, x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \ldots \right\}$$

Example. tr $a(b+c) = \{ab, ac\}$, tr $(ab+ac) = \{ab, ac\}$, tr $x =$



$$\mathrm{tr}(x) = \bigcup_{n \in \mathbb{N}} \left\{ w \in A^n \,\middle|\, x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \dots \xrightarrow{w_n} x_n \downarrow \right\} \cup \left\{ w \in A^\omega \,\middle|\, x \xrightarrow{w_1} x_1 \xrightarrow{w_2} \dots \right\}$$

Example. $\mathrm{tr}\ a(b+c) = \{ab, ac\}$, $\mathrm{tr}\ (ab+ac) = \{ab, ac\}$, $\mathrm{tr}\ x = \{a.b^\omega\}$



1.4 Rule formats: GSOS

- a **framework** for reduction rules

[*] D. Turi et G. Plotkin, « Towards a mathematical operational semantics », 1997

1.4 Rule formats: GSOS

- a **framework** for reduction rules $\rightarrow$ rule format

*D. Turi et G. Plotkin, « Towards a mathematical operational semantics », 1997

1.4 Rule formats: GSOS

- a **framework** for reduction rules $\to$ rule format
- given a **syntax** : set of operations $\mathcal{O} = \{0, a., b., ..., +, !, ...\}$ with arity map $\mathrm{ar} : \mathcal{O} \to \mathbb{N}$

*D. Turi et G. Plotkin, « Towards a mathematical operational semantics », 1997

- a **framework** for reduction rules → rule format
- given a **syntax** : set of operations $\mathcal{O} = \{0, a., b., ..., +, !, ...\}$ with arity map $\mathrm{ar} : \mathcal{O} \to \mathbb{N}$
- **GSOS rules**

$$\frac{\left\{x_i \overset{a_{i,k}}{\to} y_{i,k}\right\}_{i \in I, k \in K_i} \quad \{x_j \downarrow\}_{j \in J}}{\sigma(x_1 ... x_n) \overset{b}{\to} u} \quad \text{or} \quad \frac{\left\{x_i \overset{a_{i,k}}{\to} y_{i,k}\right\}_{i \in I, k \in K_i} \quad \{x_j \downarrow\}_{j \in J}}{\sigma(x_1 ... x_n) \downarrow}$$

with $\sigma \in \mathcal{O}, n = \mathrm{ar}\ \sigma, a_{i,k}, b \in A, I, J, K_i \subset [\![1, n]\!]$ and u complex term with variables in $\{x_1 ... x_n, y_{i,k} ...\}$

[*] D. Turi et G. Plotkin, « Towards a mathematical operational semantics », 1997

1.4 Rule formats: GSOS

- a **framework** for reduction rules → rule format
- given a **syntax** : set of operations $\mathcal{O} = \{0, a., b., ..., +, !, ...\}$ with arity map ar : $\mathcal{O} \to \mathbb{N}$
- **GSOS rules**

$$\frac{\left\{x_i \xrightarrow{a_{i,k}} y_{i,k}\right\}_{i\in I, k\in K_i} \quad \left\{x_j \downarrow\right\}_{j\in J}}{\sigma(x_1 ... x_n) \xrightarrow{b} u} \quad \text{or} \quad \frac{\left\{x_i \xrightarrow{a_{i,k}} y_{i,k}\right\}_{i\in I, k\in K_i} \quad \left\{x_j \downarrow\right\}_{j\in J}}{\sigma(x_1 ... x_n) \downarrow}$$

with $\sigma \in \mathcal{O}, n = \text{ar } \sigma, a_{i,k}, b \in A, I, J, K_i \subset [\![1, n]\!]$ and u complex term with variables in $\{x_1 ... x_n, y_{i,k} \cdots\}$

- GSOS ⇒ bisimilarity is a congruence[*]

[*] D. Turi et G. Plotkin, « Towards a mathematical operational semantics », 1997

1.4 Rule formats: GSOS

- a **framework** for reduction rules → rule format
- given a **syntax** : set of operations $\mathcal{O} = \{0, a., b., ..., +, !, ...\}$ with arity map $\mathrm{ar} : \mathcal{O} \to \mathbb{N}$
- **GSOS rules**

$$\frac{\left\{x_i \overset{a_{i,k}}{\to} y_{i,k}\right\}_{i \in I, k \in K_i} \quad \{x_j \downarrow\}_{j \in J}}{\sigma(x_1 ... x_n) \overset{b}{\to} u} \quad \text{or} \quad \frac{\left\{x_i \overset{a_{i,k}}{\to} y_{i,k}\right\}_{i \in I, k \in K_i} \quad \{x_j \downarrow\}_{j \in J}}{\sigma(x_1 ... x_n) \downarrow}$$

with $\sigma \in \mathcal{O}, n = \mathrm{ar}\ \sigma, a_{i,k}, b \in A, I, J, K_i \subset [\![1, n]\!]$ and u complex term with variables in $\{x_1 ... x_n, y_{i,k} \cdots\}$

- GSOS ⇒ bisimilarity is a congruence*, what about trace equivalence ?

*D. Turi et G. Plotkin, « Towards a mathematical operational semantics », 1997

1.5 Trace-GSOS

- recall behaviour $k : X \to \mathcal{P}(A \times X + \{\star\})$

1.5 Trace-GSOS

- recall behaviour $k : X \to \mathcal{P}(A \times X + \{\star\})$
  - $\mathcal{P}$: **effectful** behaviour (non-determinism)
  - $A \times \_ + \{\star\}$: **pure** behaviour (emit labels or terminate)

- recall behaviour $k : X \to \mathcal{P}(A \times X + \{\star\})$
  - $\mathcal{P}$: **effectful** behaviour (non-determinism)
  - $A \times \_ + \{\star\}$: **pure** behaviour (emit labels or terminate)
- **Trace**-GSOS rules

$$\frac{\left\{x_i \xrightarrow{a_i} y_i\right\}_{i \in I} \quad \left\{x_j \downarrow\right\}_{j \notin I}}{\sigma(x_1 ... x_n) \xrightarrow{b} u} \quad \text{or} \quad \frac{\left\{x_i \xrightarrow{a_i} y_i\right\}_{i \in I} \quad \left\{x_j \downarrow\right\}_{j \notin I}}{\sigma(x_1 ... x_n) \downarrow}$$

with u complex term with variables in $\{x_1 ... x_n, y_i ...\}$ with **at most one of x_i / y_i for each i** (*sublinearity*)

1.5 Trace-GSOS

- recall behaviour $k : X \to \mathcal{P}(A \times X + \{\star\})$
  - $\mathcal{P}$: **effectful** behaviour (non-determinism)
  - $A \times \_ + \{\star\}$: **pure** behaviour (emit labels or terminate)
- **Trace**-GSOS rules

$$\frac{\left\{x_i \xrightarrow{a_i} y_i\right\}_{i \in I} \quad \{x_j \downarrow\}_{j \notin I}}{\sigma(x_1 \ldots x_n) \xrightarrow{b} u} \quad \text{or} \quad \frac{\left\{x_i \xrightarrow{a_i} y_i\right\}_{i \in I} \quad \{x_j \downarrow\}_{j \notin I}}{\sigma(x_1 \ldots x_n) \downarrow}$$

with u complex term with variables in $\{x_1 \ldots x_n, y_i \ldots\}$ with **at most one of x_i / y_i for each i** (*sublinearity*)

Remark. only **pure** observations/premises

1.5 Trace-GSOS

- recall behaviour $k : X \to \mathcal{P}(A \times X + \{\star\})$
  - $\mathcal{P}$: **effectful** behaviour (non-determinism)
  - $A \times \_ + \{\star\}$: **pure** behaviour (emit labels or terminate)
- **Trace**-GSOS rules

$$\frac{\left\{x_i \xrightarrow{a_i} y_i\right\}_{i \in I} \quad \{x_j \downarrow\}_{j \notin I}}{\sigma(x_1 ... x_n) \xrightarrow{b} u} \quad \text{or} \quad \frac{\left\{x_i \xrightarrow{a_i} y_i\right\}_{i \in I} \quad \{x_j \downarrow\}_{j \notin I}}{\sigma(x_1 ... x_n) \downarrow}$$

with u complex term with variables in $\{x_1 ... x_n, y_i ...\}$ with **at most one of x_i / y_i for each i** (*sublinearity*)

Remark. only **pure** observations/premises: observe each variable **once and only once**

Example.

$$\frac{t \xrightarrow{a} t' \quad t \xrightarrow{a} t''}{?\,t \xrightarrow{a} t + t'} \qquad \frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} t + t} \qquad \frac{}{a.t \xrightarrow{a} t}\,\forall a \qquad \frac{t \xrightarrow{b} t'}{a.t \xrightarrow{a} t}\,\forall a, b \qquad \frac{t \downarrow}{a.t \xrightarrow{a} t}\,\forall a$$

😕 😕 😕 😎 😎

Example.

$$\frac{t \xrightarrow{a} t' \quad t \xrightarrow{a} t''}{?\,t \xrightarrow{a} t + t'} \qquad \frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} t + t} \qquad \frac{}{a.t \xrightarrow{a} t}\,\forall a \qquad \frac{t \xrightarrow{b} t'}{a.t \xrightarrow{a} t}\,\forall a, b \qquad \frac{t \downarrow}{a.t \xrightarrow{a} t}\,\forall a$$

😕 😕 😕 😎 😎

- require 2 extra conditions on the set of rules:

Example.

$$\frac{t \xrightarrow{a} t' \quad t \xrightarrow{a} t''}{?\,t \xrightarrow{a} t + t'} \qquad \frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} t + t} \qquad \frac{}{a.t \xrightarrow{a} t}\,\forall a \qquad \frac{t \xrightarrow{b} t'}{a.t \xrightarrow{a} t}\,\forall a, b \qquad \frac{t \downarrow}{a.t \xrightarrow{a} t}\,\forall a$$

😕 😕 😕 😎 😎

- require 2 extra conditions on the set of rules:
  - **affineness**: for each term, there is a least one rule that can apply

Example.

$$\frac{t \xrightarrow{a} t' \quad t \xrightarrow{a} t''}{?\,t \xrightarrow{a} t + t'} \qquad \frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} t + t} \qquad \frac{}{a.t \xrightarrow{a} t}\,\forall a \qquad \frac{t \xrightarrow{b} t'}{a.t \xrightarrow{a} t}\,\forall a, b \qquad \frac{t \downarrow}{a.t \xrightarrow{a} t}\,\forall a$$

😕 😕 😕 😎 😎

- require 2 extra conditions on the set of rules:
  - **affineness**: for each term, there is a least one rule that can apply $\rightsquigarrow k : X \to \mathcal{P}(A \times X + \{\star\})$

Example.

$$\frac{t \xrightarrow{a} t' \quad t \xrightarrow{a} t''}{?\,t \xrightarrow{a} t + t'} \qquad \frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} t + t} \qquad \frac{}{a.t \xrightarrow{a} t}\,\forall a \qquad \frac{t \xrightarrow{b} t'}{a.t \xrightarrow{a} t}\,\forall a, b \qquad \frac{t \downarrow}{a.t \xrightarrow{a} t}\,\forall a$$

😕 😕 😕 😎 😎

- require 2 extra conditions on the set of rules:
  - **affineness**: for each term, there is a least one rule that can apply $\rightsquigarrow k : X \to \mathcal{P}_{\mathrm{ne}}(A \times X + \{\star\})$

Example.

$$\frac{t \xrightarrow{a} t' \quad t \xrightarrow{a} t''}{?t \xrightarrow{a} t + t'} \qquad \frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} t + t} \qquad \frac{}{a.t \xrightarrow{a} t}\forall a \qquad \frac{t \xrightarrow{b} t'}{a.t \xrightarrow{a} t}\forall a, b \qquad \frac{t \downarrow}{a.t \xrightarrow{a} t}\forall a$$

😕 😕 😕 😎 😎

- require 2 extra conditions on the set of rules:
  - **affineness**: for each term, there is a least one rule that can apply $\rightsquigarrow k : X \to \mathcal{P}_{\mathrm{ne}}(A \times X + \{\star\})$
  - **smoothness**: x_i in the target, the observation on x_i is **irrelevant** ie. any other observation could have been done (the same rule for each other possible observation exists)

1.6 Theorem

Theorem 1. Let $\mathcal{R}$ be a smooth and affine set of Trace-GSOS rules. Let X be a set of terms equipped with behaviour $k : X \to \mathcal{P}_{\text{ne}}(A \times X + \{\star\})$ induced by $\mathcal{R}$. Then trace equivalence $\underset{\text{tr}}{\sim}$ is a congruence.

1.6 Theorem

Theorem 2. Let $\mathcal{R}$ be a smooth and affine set of Trace-GSOS rules. Let X be a set of terms equipped with behaviour $k : X \to \mathcal{P}_{\mathrm{ne}}(A \times X + \{\star\})$ induced by $\mathcal{R}$. Then trace equivalence $\underset{\mathrm{tr}}{\sim}$ is a congruence.

Proof. Show that the trace of a complex term can be obtained from the traces of its subterms.

Theorem 3. Let $\mathcal{R}$ be a smooth and affine set of Trace-GSOS rules. Let X be a set of terms equipped with behaviour $k : X \to \mathcal{P}_{\text{ne}}(A \times X + \{\star\})$ induced by $\mathcal{R}$. Then trace equivalence $\underset{\text{tr}}{\sim}$ is a congruence.

Proof. Show that the trace of a complex term can be obtained from the traces of its subterms.

- consider complex terms with words of A^∞ as leaves, and behaviour induced by $\mathcal{R}$ with $a.w \xrightarrow{a} w$ and $\varepsilon \downarrow$

1.6 Theorem

Theorem 4. Let $\mathcal{R}$ be a smooth and affine set of Trace-GSOS rules. Let X be a set of terms equipped with behaviour $k : X \to \mathcal{P}_{\text{ne}}(A \times X + \{\star\})$ induced by $\mathcal{R}$. Then trace equivalence $\underset{\text{tr}}{\sim}$ is a congruence.

Proof. Show that the trace of a complex term can be obtained from the traces of its subterms.

- consider complex terms with words of A^∞ as leaves, and behaviour induced by $\mathcal{R}$ with $a.w \xrightarrow{a} w$ and $\varepsilon \downarrow$
- extend the trace function of this system to maps $[\![u]\!] : (\mathcal{P}_{\text{ne}} A^\infty)^n \to \mathcal{P}_{\text{ne}} A^\infty$ for each complex term u with n free variables

Theorem 5. Let $\mathcal{R}$ be a smooth and affine set of Trace-GSOS rules. Let X be a set of terms equipped with behaviour $k : X \to \mathcal{P}_{\mathrm{ne}}(A \times X + \{\star\})$ induced by $\mathcal{R}$. Then trace equivalence $\underset{\mathrm{tr}}{\sim}$ is a congruence.

Proof. Show that the trace of a complex term can be obtained from the traces of its subterms.

- consider complex terms with words of A^∞ as leaves, and behaviour induced by $\mathcal{R}$ with $a.w \xrightarrow{a} w$ and $\varepsilon \downarrow$
- extend the trace function of this system to maps $[\![u]\!] : (\mathcal{P}_{\mathrm{ne}} A^\infty)^n \to \mathcal{P}_{\mathrm{ne}} A^\infty$ for each complex term u with n free variables
- prove $\mathrm{tr}\ u[t_1 ... t_n] = [\![u]\!](\mathrm{tr}\ t_1 ...\ \mathrm{tr}\ t_n)$ by showing that both sides are maximal coalgebra morphisms

$\square$

1.7 Counter-examples

Affineness, smoothness and sublinearity are **necessary**

Affineness, smoothness and sublinearity are **necessary**

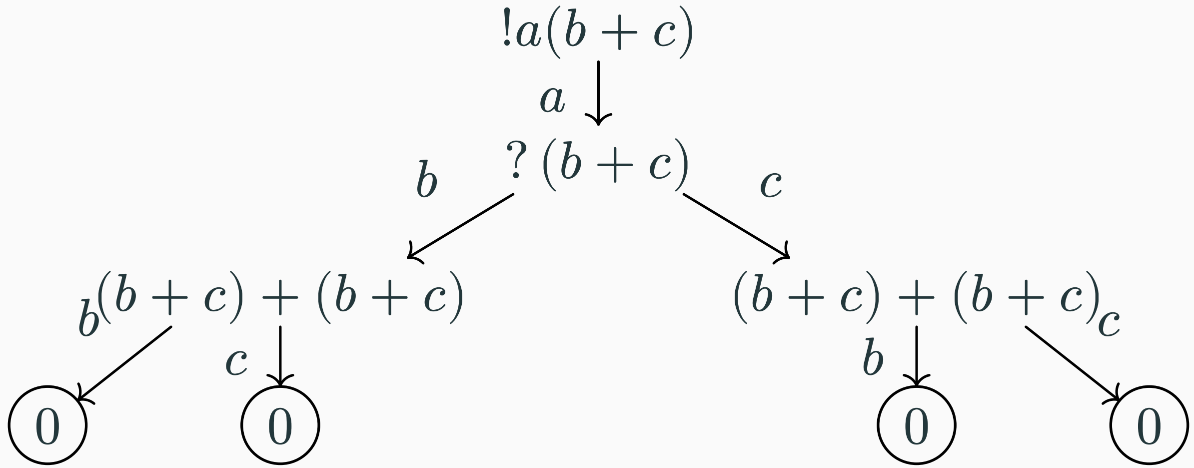
1.7.1 Sublinearity

$$\frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} ?\,t'}\forall a \qquad \frac{t \xrightarrow{a} t'}{?\,t \xrightarrow{a} t + t}\forall a$$

Affineness, smoothness and sublinearity are **necessary**

1.7.1 Sublinearity

$$\frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} ?\,t'}\forall a \qquad \frac{t \xrightarrow{a} t'}{?\,t \xrightarrow{a} t + t}\forall a$$



Affineness, smoothness and sublinearity are **necessary**

1.7.1 Sublinearity

$$\frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} ?\,t'}\,\forall a \qquad \frac{t \xrightarrow{a} t'}{?\,t \xrightarrow{a} t + t}\,\forall a$$

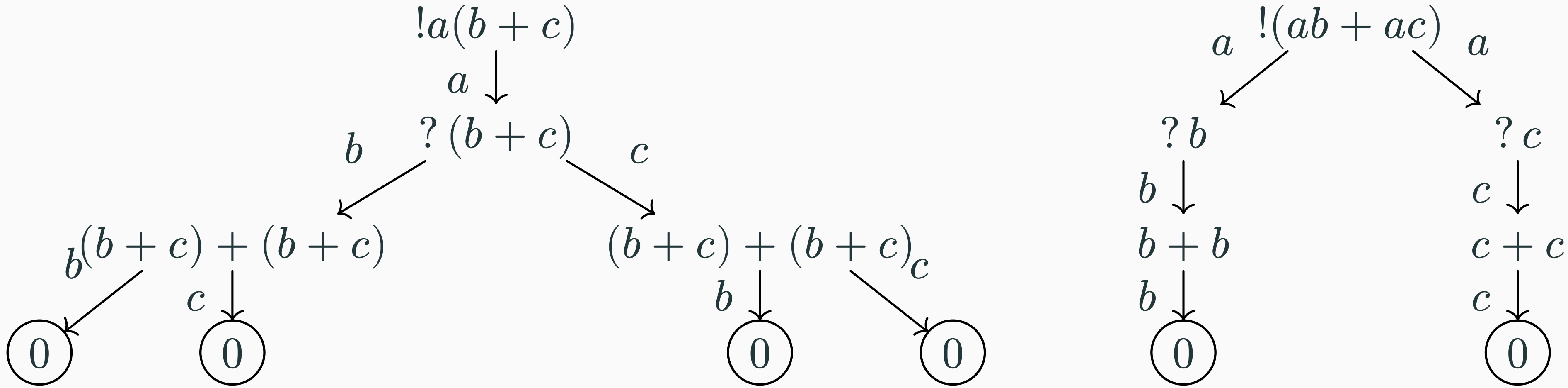
$$\begin{tikzcd}
 & & !a(b+c) \arrow[d, "a"'] & & \\
 & & ?\,(b+c) \arrow[dl, "b"'] \arrow[dr, "c"] & & \\
 & (b+c)+(b+c) \arrow[dl, "b"'] \arrow[d, "c"'] & & (b+c)+(b+c) \arrow[d, "b"'] \arrow[dr, "c"] & \\
0 & 0 & & 0 & 0
\end{tikzcd}$$

$\mathrm{tr}\,!a(b+c) = \{abb, \underline{abc}, \underline{acb}, acc\}$

Affineness, smoothness and sublinearity are **necessary**

1.7.1 Sublinearity

$$\frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} ?\,t'}\,\forall a \qquad \frac{t \xrightarrow{a} t'}{?\,t \xrightarrow{a} t + t}\,\forall a$$

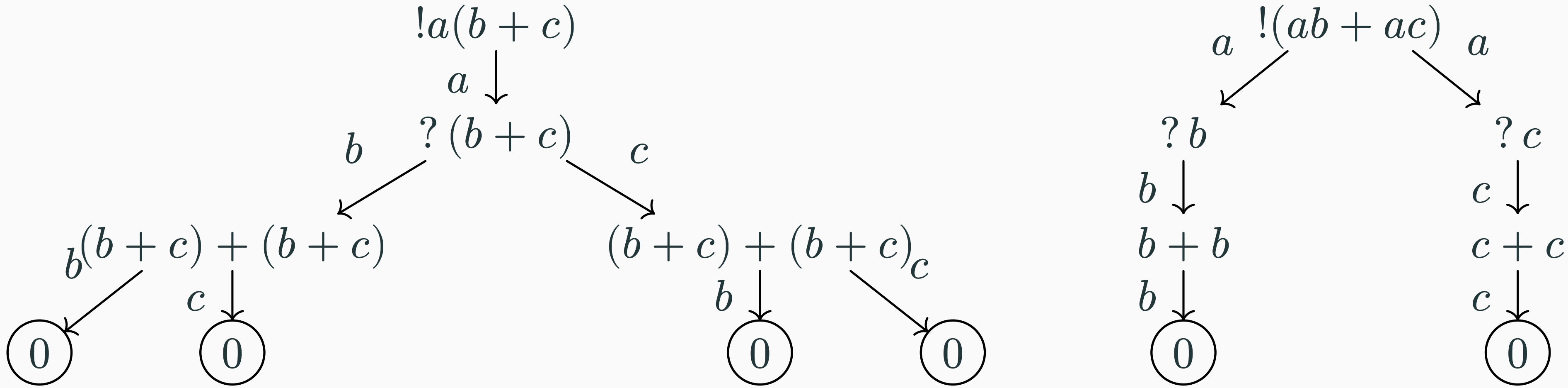


$\operatorname{tr} !a(b+c) = \{abb, \underline{abc}, \underline{acb}, acc\}$

Affineness, smoothness and sublinearity are **necessary**

1.7.1 Sublinearity

$$\frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} ?\,t'}\,\forall a \qquad \frac{t \xrightarrow{a} t'}{?\,t \xrightarrow{a} t + t}\,\forall a$$



$$\mathrm{tr}\, !a(b+c) = \{abb, \underline{abc}, \underline{acb}, acc\} \neq \{abb, acc\} = \mathrm{tr}\, !(ab+ac)$$

1.7.2 Smoothness

$$\frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} ?\,t'}\forall a \qquad \frac{t \xrightarrow{c} t'}{?\,t \xrightarrow{c} t} \qquad \frac{t \xrightarrow{a} t'}{?\,t\downarrow}\forall a \neq c$$

1.7 Counter-examples

1.7.2 Smoothness

$$\frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} ?\,t'}\,\forall a \qquad \frac{t \xrightarrow{c} t'}{?\,t \xrightarrow{c} t} \qquad \frac{t \xrightarrow{a} t'}{?\,t \downarrow}\,\forall a \neq c$$

$$\begin{tikzcd}
& !a(b+c) \arrow[d, "a"'] & \\
& ?\,(b+c) \arrow[d, "c"'] & \\
& b+c \arrow[dl, "b"'] \arrow[dr, "c"] & \\
0 & & 0
\end{tikzcd}$$

1.7 Counter-examples

1.7.2 Smoothness

$$\frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} ?\,t'}\forall a \qquad \frac{t \xrightarrow{c} t'}{?\,t \xrightarrow{c} t} \qquad \frac{t \xrightarrow{a} t'}{?\,t \downarrow}\forall a \neq c$$

$$\begin{tikzcd}
 & !a(b+c) \arrow[d, "a"'] & \\
 & ?\,(b+c) \arrow[d, "c"'] & \\
 & b+c \arrow[dl, "b"'] \arrow[dr, "c"] & \\
0 & & 0
\end{tikzcd}$$

$\mathrm{tr}\,!a(b+c) = \{\underline{acb}, acc\}$

1.7.2 Smoothness

$$\frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} ?\,t'}\,\forall a \qquad \frac{t \xrightarrow{c} t'}{?\,t \xrightarrow{c} t} \qquad \frac{t \xrightarrow{a} t'}{?\,t \downarrow}\,\forall a \neq c$$

$$\begin{tikzcd}[column sep=small]
 & !a(b+c) \arrow[d, "a"'] & \\
 & ?\,(b+c) \arrow[d, "c"'] & \\
 & b+c \arrow[dl, "b"'] \arrow[dr, "c"] & \\
0 & & 0
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small]
 & !(ab+ac) \arrow[dl, "a"'] \arrow[dr, "a"] & \\
\boxed{?\,b} & & ?\,c \arrow[d, "c"] \\
 & & c \arrow[d, "c"] \\
 & & 0
\end{tikzcd}$$

$\mathrm{tr}\,!a(b+c) = \{\underline{acb}, acc\}$

1.7.2 Smoothness

$$\frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} ?\,t'}\,\forall a \qquad \frac{t \xrightarrow{c} t'}{?\,t \xrightarrow{c} t} \qquad \frac{t \xrightarrow{a} t'}{?\,t \downarrow}\,\forall a \neq c$$

$$\begin{tikzcd}
& !a(b+c) \arrow[d, "a"'] & \\
& ?\,(b+c) \arrow[d, "c"'] & \\
& b+c \arrow[dl, "b"'] \arrow[dr, "c"] & \\
0 & & 0
\end{tikzcd}
\qquad
\begin{tikzcd}
& !(ab+ac) \arrow[dl, "a"'] \arrow[dr, "a"] & \\
\boxed{?\,b} & & ?\,c \arrow[d, "c"] \\
& & c \arrow[d, "c"] \\
& & 0
\end{tikzcd}$$

$$\mathrm{tr}\,!a(b+c) = \{\underline{acb}, acc\} \neq \{\underline{a}, acc\} = \mathrm{tr}\,!(ab+ac)$$

1.7.3 Affineness

- the proof need *deep smoothness*: rules on complex terms obtained by stacking rules of $\mathcal{R}$ need to be smooth

1.7.3 Affineness

- the proof need *deep smoothness*: rules on complex terms obtained by stacking rules of $\mathcal{R}$ need to be smooth
- without affineness, no deep smoothness:

$$\frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} b.?\,t'}\,\forall a \qquad \frac{t \xrightarrow{c} t'}{?\,t \xrightarrow{c} t'}$$

1.7 Counter-examples

1.7.3 Affineness

- the proof need *deep smoothness*: rules on complex terms obtained by stacking rules of $\mathcal{R}$ need to be smooth
- without affineness, no deep smoothness:

$$\frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} b.?\,t'}\forall a \qquad \frac{t \xrightarrow{c} t'}{?\,t \xrightarrow{c} t'}$$

$$\cfrac{\cfrac{t \xrightarrow{c} t'}{?\,t \xrightarrow{c} t'}}{a.?\,t \xrightarrow{a} ?\,t}$$

1.7.3 Affineness

- the proof need *deep smoothness*: rules on complex terms obtained by stacking rules of $\mathcal{R}$ need to be smooth
- without affineness, no deep smoothness:

$$\frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} b.?\,t'}\,\forall a \qquad \frac{t \xrightarrow{c} t'}{?\,t \xrightarrow{c} t'}$$

$$\dfrac{\dfrac{t \xrightarrow{c} t'}{?\,t \xrightarrow{c} t'}}{a.?\,t \xrightarrow{a} ?\,t} \quad = \quad \frac{t \xrightarrow{c} t'}{a.?\,t \xrightarrow{a} ?\,t}$$

1.7.3 Affineness

- the proof need *deep smoothness*: rules on complex terms obtained by stacking rules of $\mathcal{R}$ need to be smooth
- without affineness, no deep smoothness:

$$\frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} b.?\,t'}\forall a \qquad \frac{t \xrightarrow{c} t'}{?\,t \xrightarrow{c} t'}$$

$$\dfrac{\dfrac{t \xrightarrow{c} t'}{?\,t \xrightarrow{c} t'}}{a.?\,t \xrightarrow{a} ?\,t} = \frac{t \xrightarrow{c} t'}{a.?\,t \xrightarrow{a} ?\,t} \qquad \text{for smoothness, need} \qquad \frac{t \xrightarrow{b} t'}{a.?\,t \xrightarrow{a} ?\,t}$$

1.7.3 Affineness

- the proof need *deep smoothness*: rules on complex terms obtained by stacking rules of $\mathcal{R}$ need to be smooth
- without affineness, no deep smoothness:

$$\frac{t \xrightarrow{a} t'}{!t \xrightarrow{a} b.?\,t'}\forall a \qquad \frac{t \xrightarrow{c} t'}{?\,t \xrightarrow{c} t'}$$

$$\cfrac{\cfrac{t \xrightarrow{c} t'}{?\,t \xrightarrow{c} t'}}{a.?\,t \xrightarrow{a} ?\,t} \;=\; \frac{t \xrightarrow{c} t'}{a.?\,t \xrightarrow{a} ?\,t} \quad \text{for smoothness, need} \quad \frac{t \xrightarrow{b} t'}{a.?\,t \xrightarrow{a} ?\,t} \quad \text{but} \quad \cfrac{\cfrac{t \xrightarrow{b} t'}{?\,t \not\rightarrow}}{a.?\,t \not\rightarrow}$$

☢️ 🔞 WARNING 🔞 ☢️

YOU ARE ABOUT TO ENTER THE MONAD ZONE

COME IN AT YOUR PERIL

(please note that the following part of the talk is heavily populated by monads, (co)algebras, functors, natural transformations and akin, it is highly recommended to not be allergic to those if you wish to pursue your journey with us)

2 Abstraction

2.1 Algebras and coalgebras

- $\mathbb{C}$ a category with products (eg. Set)

2.1 Algebras and coalgebras

- $\mathbb{C}$ a category with products (eg. Set)
- **syntax**: endofunctor $\Sigma : \mathbb{C} \to \mathbb{C}$

2.1 Algebras and coalgebras

- $\mathbb{C}$ a category with products (eg. Set)
- **syntax**: endofunctor $\Sigma : \mathbb{C} \to \mathbb{C}$
  - Σ-algebra $i : \Sigma X \to X$

2.1 Algebras and coalgebras

- $\mathbb{C}$ a category with products (eg. Set)
- **syntax**: endofunctor $\Sigma : \mathbb{C} \to \mathbb{C}$
  - Σ-algebra $i : \Sigma X \to X$

Example. $\Sigma X = \coprod_{\sigma \in \mathcal{O}} X^{\mathrm{ar}\ \sigma}$

For $t ::= 0 \mid a.t \mid t + t \mid\ !t$, $\Sigma X = 1 + A \times X + X^2 + X$

2.1 Algebras and coalgebras

- $\mathbb{C}$ a category with products (eg. Set)
- **syntax**: endofunctor $\Sigma : \mathbb{C} \to \mathbb{C}$
  - Σ-algebra $i : \Sigma X \to X$

Example. $\Sigma X = \coprod_{\sigma \in \mathcal{O}} X^{\mathrm{ar}\ \sigma}$

For $t ::= 0 \mid a.t \mid t + t \mid !t$, $\Sigma X = 1 + A \times X + X^2 + X$

- **behaviour**: endofunctor $H : \mathbb{C} \to \mathbb{C}$

2.1 Algebras and coalgebras

- $\mathbb{C}$ a category with products (eg. Set)
- **syntax**: endofunctor $\Sigma : \mathbb{C} \to \mathbb{C}$
  - Σ-algebra $i : \Sigma X \to X$

Example. $\Sigma X = \coprod_{\sigma \in \mathcal{O}} X^{\mathrm{ar}\ \sigma}$

For $t ::= 0 \mid a.t \mid t + t \mid !t$, $\Sigma X = 1 + A \times X + X^2 + X$

- **behaviour**: endofunctor $H : \mathbb{C} \to \mathbb{C}$
  - H-coalgebra $k : X \to HX$

- $\mathbb{C}$ a category with products (eg. Set)
- **syntax**: endofunctor $\Sigma : \mathbb{C} \to \mathbb{C}$
  - Σ-algebra $i : \Sigma X \to X$

Example. $\Sigma X = \coprod_{\sigma \in \mathcal{O}} X^{\mathrm{ar}\ \sigma}$

For $t ::= 0 \mid a.t \mid t + t \mid !t$, $\Sigma X = 1 + A \times X + X^2 + X$

- **behaviour**: endofunctor $H : \mathbb{C} \to \mathbb{C}$
  - H-coalgebra $k : X \to HX$

Example. $HX = \mathcal{P}_{\mathrm{ne}}(A \times X + 1)$

2.2 Abstract GSOS

- set of rules $\mathcal{R} \rightsquigarrow$ a **natural transformation** $\rho_X : \Sigma(X \times HX) \to H\Sigma^* X$

2.2 Abstract GSOS

- set of rules $\mathcal{R} \rightsquigarrow$ a **natural transformation** $\rho_X : \Sigma(X \times HX) \to H\Sigma^* X$

Example. For the previous example without $! : \Sigma X = 1 + A \times X + X^2$,

$$\rho : 1 + A \times (X \times \mathcal{P}(1 + A \times X)) + (X \times \mathcal{P}(1 + A \times X))^2 \to \mathcal{P}(1 + A \times \Sigma^* X)$$

- set of rules $\mathcal{R} \rightsquigarrow$ a **natural transformation** $\rho_X : \Sigma(X \times HX) \to H\Sigma^* X$

Example. For the previous example without ! : $\Sigma X = 1 + A \times X + X^2$,

$$\rho : 1 + A \times (X \times \mathcal{P}(1 + A \times X)) + (X \times \mathcal{P}(1 + A \times X))^2 \to \mathcal{P}(1 + A \times \Sigma^* X)$$

- $\rho(*) = \{*\}$

$$\frac{}{0 \downarrow}$$

2.2 Abstract GSOS

- set of rules $\mathcal{R} \rightsquigarrow$ a **natural transformation** $\rho_X : \Sigma(X \times HX) \to H\Sigma^* X$

Example. For the previous example without ! : $\Sigma X = 1 + A \times X + X^2$,

$$\rho : 1 + A \times (X \times \mathcal{P}(1 + A \times X)) + (X \times \mathcal{P}(1 + A \times X))^2 \to \mathcal{P}(1 + A \times \Sigma^* X)$$

- $\rho(*) = \{*\}$
- $\rho((a, t, T)) = \{(a, t)\}$

$$\frac{}{a.t \xrightarrow{a} t}\ \forall a$$

- set of rules $\mathcal{R} \rightsquigarrow$ a **natural transformation** $\rho_X : \Sigma(X \times HX) \to H\Sigma^* X$

Example. For the previous example without ! : $\Sigma X = 1 + A \times X + X^2$,

$$\rho : 1 + A \times (X \times \mathcal{P}(1 + A \times X)) + (X \times \mathcal{P}(1 + A \times X))^2 \to \mathcal{P}(1 + A \times \Sigma^* X)$$

- $\rho(*) = \{*\}$
- $\rho((a, t, T)) = \{(a, t)\}$
- $\rho((t, T), (u, U)) = \{(a, t') \mid \forall (a, t') \in T\} \cup$

$$\frac{t \xrightarrow{a} t'}{t + u \xrightarrow{a} t'} \forall a$$

2.2 Abstract GSOS

- set of rules $\mathcal{R} \rightsquigarrow$ a **natural transformation** $\rho_X : \Sigma(X \times HX) \to H\Sigma^* X$

Example. For the previous example without $! : \Sigma X = 1 + A \times X + X^2$,

$$\rho : 1 + A \times (X \times \mathcal{P}(1 + A \times X)) + (X \times \mathcal{P}(1 + A \times X))^2 \to \mathcal{P}(1 + A \times \Sigma^* X)$$

- $\rho(*) = \{*\}$
- $\rho((a, t, T)) = \{(a, t)\}$
- $\rho((t, T), (u, U)) = \{(a, t') \mid \forall (a, t') \in T\} \cup \{(a, u') \mid \forall (a, u') \in U\} \cup$

$$\frac{t \xrightarrow{a} t'}{t + u \xrightarrow{a} t'}\forall a \qquad \frac{u \xrightarrow{a} u'}{t + u \xrightarrow{a} u'}\forall a$$

- set of rules $\mathcal{R} \rightsquigarrow$ a **natural transformation** $\rho_X : \Sigma(X \times HX) \to H\Sigma^* X$

Example. For the previous example without $! : \Sigma X = 1 + A \times X + X^2$,

$$\rho : 1 + A \times (X \times \mathcal{P}(1 + A \times X)) + (X \times \mathcal{P}(1 + A \times X))^2 \to \mathcal{P}(1 + A \times \Sigma^* X)$$

- $\rho(*) = \{*\}$
- $\rho((a, t, T)) = \{(a, t)\}$
- $\rho((t, T), (u, U)) = \{(a, t') \mid \forall (a, t') \in T\} \cup \{(a, u') \mid \forall (a, u') \in U\} \cup \{* \mid * \in T\} \cup \{* \mid * \in U\}$

$$\frac{t \xrightarrow{a} t'}{t + u \xrightarrow{a} t'}\forall a \qquad \frac{u \xrightarrow{a} u'}{t + u \xrightarrow{a} u'}\forall a \qquad \frac{t\downarrow}{t + u\downarrow} \qquad \frac{u\downarrow}{t + u\downarrow}$$

2.3 Kleisli trace semantics

- recall $HX = \mathcal{P}_{\text{ne}}(1 + A \times X)$

2.3 Kleisli trace semantics

- recall $HX = \mathcal{P}_{\mathrm{ne}}(1 + A \times X) = TBX$
  - $T = \mathcal{P}_{\mathrm{ne}}$ **effectful** behaviour
  - $B = 1 + A \times X$ **pure** behaviour

2.3 Kleisli trace semantics

- recall $HX = \mathcal{P}_{\mathrm{ne}}(1 + A \times X) = TBX$
  - $T = \mathcal{P}_{\mathrm{ne}}$ **effectful** behaviour $\rightsquigarrow$ non empty powerset : non-determinism
  - $B = 1 + A \times X$ **pure** behaviour $\rightsquigarrow$ words : $Z = A^\infty$ (final B-algebra)

2.3 Kleisli trace semantics

- recall $HX = \mathcal{P}_{\mathrm{ne}}(1 + A \times X) = TBX$
  - $T = \mathcal{P}_{\mathrm{ne}}$ **effectful** behaviour $\rightsquigarrow$ non empty powerset : non-determinism
  - $B = 1 + A \times X$ **pure** behaviour $\rightsquigarrow$ words : $Z = A^{\infty}$ (final B-algebra)
- $\mathrm{tr}\, x \in \mathcal{P}(A^{\infty}) = TZ$

2.3 Kleisli trace semantics

- recall $HX = \mathcal{P}_{\text{ne}}(1 + A \times X) = TBX$
  - $T = \mathcal{P}_{\text{ne}}$ **effectful** behaviour $\rightsquigarrow$ non empty powerset : non-determinism
  - $B = 1 + A \times X$ **pure** behaviour $\rightsquigarrow$ words : $Z = A^{\infty}$ (final B-algebra)
- $\operatorname{tr} x \in \mathcal{P}(A^{\infty}) = TZ$
- T is a **monad**

2.3 Kleisli trace semantics

- recall $HX = \mathcal{P}_{\text{ne}}(1 + A \times X) = TBX$
  - $T = \mathcal{P}_{\text{ne}}$ **effectful** behaviour $\rightsquigarrow$ non empty powerset : non-determinism
  - $B = 1 + A \times X$ **pure** behaviour $\rightsquigarrow$ words : $Z = A^\infty$ (final B-algebra)
- $\operatorname{tr} x \in \mathcal{P}(A^\infty) = TZ$
- T is a **monad**
- **Kleisli category** of T

2.3 Kleisli trace semantics

- recall $HX = \mathcal{P}_{\text{ne}}(1 + A \times X) = TBX$
  - $T = \mathcal{P}_{\text{ne}}$ **effectful** behaviour $\rightsquigarrow$ non empty powerset : non-determinism
  - $B = 1 + A \times X$ **pure** behaviour $\rightsquigarrow$ words : $Z = A^\infty$ (final B-algebra)
- $\operatorname{tr} x \in \mathcal{P}(A^\infty) = TZ$
- T is a **monad**
- **Kleisli category** of T

$$A \in \mathrm{Kl}(T) \Leftrightarrow A \in \mathbb{C} \qquad\qquad A \nrightarrow B \in \mathrm{Kl}(T) \Leftrightarrow A \to TB \in \mathbb{C}$$

- recall $HX = \mathcal{P}_{\text{ne}}(1 + A \times X) = TBX$
  - $T = \mathcal{P}_{\text{ne}}$ **effectful** behaviour $\rightsquigarrow$ non empty powerset : non-determinism
  - $B = 1 + A \times X$ **pure** behaviour $\rightsquigarrow$ words : $Z = A^\infty$ (final B-algebra)
- $\operatorname{tr} x \in \mathcal{P}(A^\infty) = TZ$
- T is a **monad**
- **Kleisli category** of T

$$A \in \mathrm{Kl}(T) \Leftrightarrow A \in \mathbb{C} \qquad\qquad A \nrightarrow B \in \mathrm{Kl}(T) \Leftrightarrow A \to TB \in \mathbb{C}$$

- $B : \mathbb{C} \to \mathbb{C}$ **extends** to $\overline{B} : \mathrm{Kl}(T) \to \mathrm{Kl}(T)$ (distributive law $\lambda^B : BT \Rightarrow TB$)

2.3 Kleisli trace semantics

- $Z = A^\infty$ is a B-coalgebra in $\mathrm{Kl}(T)$

2.3 Kleisli trace semantics

- $Z = A^\infty$ is a B-coalgebra in $\mathrm{Kl}(T)$

$$\zeta : Z \nrightarrow BZ \text{ or } Z \to TBZ \qquad\qquad \zeta(\varepsilon) = \{*\}, \quad \zeta(a.w) = \{(a, w)\}$$

- $Z = A^\infty$ is a B-coalgebra in $\mathrm{Kl}(T)$

$$\zeta : Z \nrightarrow BZ \text{ or } Z \to TBZ \qquad\qquad \zeta(\varepsilon) = \{*\}, \quad \zeta(a.w) = \{(a, w)\}$$

- for any $k : X \nrightarrow BX$, $h : X \nrightarrow Z$ is a $\overline{B}$-coalgebra morphism if

$$\begin{tikzcd} X \arrow[r, "k", "\vert"'] \arrow[d, dashed, "h"', "\vert"] & BX \arrow[d, dashed, "\overline{B}h", "\vert"'] \\ Z \arrow[r, "\zeta", "\vert"'] & BZ \end{tikzcd}$$

2.3 Kleisli trace semantics

- $Z = A^\infty$ is a B-coalgebra in $\mathrm{Kl}(T)$

$$\zeta : Z \nrightarrow BZ \text{ or } Z \to TBZ \qquad\qquad \zeta(\varepsilon) = \{*\}, \quad \zeta(a.w) = \{(a, w)\}$$

- for any $k : X \nrightarrow BX$, $h : X \nrightarrow Z$ is a $\overline{B}$-coalgebra morphism if

$$\begin{tikzcd} X \arrow[r, "k", "|"] \arrow[d, dashed, "h"', "-"] & BX \arrow[d, dashed, "\overline{B}h", "-"'] \\ Z \arrow[r, "\zeta", "|"] & BZ \end{tikzcd}$$

- suppose $\mathrm{Kl}(T)$ enriched with an **order on maps with maximums**, define tr_k the **greatest $\overline{B}$-coalgebra morphism**

2.4 Trace-GSOS

- GSOS rule

$$\rho : \Sigma(X \times HX) \to H\Sigma^* X$$

2.4 Trace-GSOS

- GSOS rule

$$\rho : \Sigma(X \times TBX) \to TB\Sigma^* X$$

2.4 Trace-GSOS

- GSOS rule

$$\rho : \Sigma(X \times TBX) \to TB\Sigma^* X$$

- **Trace**-GSOS rule

$$\rho : \Sigma(X \times BX) \to TB\Sigma^* X$$

2.4 Trace-GSOS

- GSOS rule

$$\rho : \Sigma(X \times TBX) \to TB\Sigma^* X$$

- **Trace**-GSOS rule

$$\rho : \Sigma(X \times BX) \to TB\Sigma^* X$$

$\rightsquigarrow$ only pure observations

2.4 Trace-GSOS

- GSOS rule

$$\rho : \Sigma(X \times TBX) \to TB\Sigma^* X$$

- **Trace**-GSOS rule

$$\rho : \Sigma(X \times BX) \nrightarrow B\Sigma^* X$$

$\rightsquigarrow$ only pure observations

2.4 Trace-GSOS

- GSOS rule

$$\rho : \Sigma(X \times TBX) \to TB\Sigma^* X$$

- **Trace**-GSOS rule

$$\rho : \Sigma(X \times BX) \nrightarrow B\Sigma^* X$$

$\rightsquigarrow$ only pure observations

- B and Σ extend to $\mathrm{Kl}(T)$ (and Σ^*) but not $+$ 😕

- GSOS rule

$$\rho : \Sigma(X \times TBX) \to TB\Sigma^* X$$

- **Trace**-GSOS rule

$$\rho : \Sigma(X \times BX) \nrightarrow B\Sigma^* X$$

$\rightsquigarrow$ only pure observations

- B and Σ extend to $\mathrm{Kl}(T)$ (and Σ^*) but not $+$ 😕
- **affineness**: ask T to be an **affine monad**

2.5 Strong and affine monads

- **strong monad**: $\mathrm{st}_{X,Y} : X \times TY \to T(X \times Y)$

2.5 Strong and affine monads

- **strong monad**: $\mathrm{st}_{X,Y} : X \times TY \to T(X \times Y) \rightsquigarrow \mathrm{st}' : TX \times Y \to T(X \times Y)$

2.5 Strong and affine monads

- **strong monad**: $\mathrm{st}_{X,Y} : X \times TY \to T(X \times Y) \rightsquigarrow \mathrm{st}' : TX \times Y \to T(X \times Y)$
- double strength $\mathrm{dst} : TX \times TY \xrightarrow{\mathrm{st}} T(TX \times Y) \xrightarrow{T\mathrm{st}'} T^2(X \times Y) \xrightarrow{\mu} T(X \times Y)$

2.5 Strong and affine monads

- **strong monad**: $\mathrm{st}_{X,Y} : X \times TY \to T(X \times Y) \rightsquigarrow \mathrm{st}' : TX \times Y \to T(X \times Y)$
- double strength dst : $TX \times TY \xrightarrow{\mathrm{st}} T(TX \times Y) \xrightarrow{T\mathrm{st}'} T^2(X \times Y) \xrightarrow{\mu} T(X \times Y)$ (and dst′)

2.5 Strong and affine monads

- **strong monad**: $\mathrm{st}_{X,Y} : X \times TY \to T(X \times Y) \rightsquigarrow \mathrm{st}' : TX \times Y \to T(X \times Y)$
- double strength $\mathrm{dst} : TX \times TY \xrightarrow{\mathrm{st}} T(TX \times Y) \xrightarrow{T\mathrm{st}'} T^2(X \times Y) \xrightarrow{\mu} T(X \times Y)$ (and dst')
- **affine monad**: $TX \times TY \xrightarrow{\mathrm{dst}} T(X \times Y) \xrightarrow{\langle T\pi_1, T\pi_2 \rangle} TX \times TY = \mathrm{id}$ or $\eta_1 : 1 \xrightarrow{\simeq} T1$

2.5 Strong and affine monads

- **strong monad**: $\mathrm{st}_{X,Y} : X \times TY \to T(X \times Y) \rightsquigarrow \mathrm{st}' : TX \times Y \to T(X \times Y)$
- double strength $\mathrm{dst} : TX \times TY \xrightarrow{\mathrm{st}} T(TX \times Y) \xrightarrow{T\mathrm{st}'} T^2(X \times Y) \xrightarrow{\mu} T(X \times Y)$ (and dst')
- **affine monad**: $TX \times TY \xrightarrow{\mathrm{dst}} T(X \times Y) \xrightarrow{\langle T\pi_1, T\pi_2 \rangle} TX \times TY = \mathrm{id}$ or $\eta_1 : 1 \xrightarrow{\simeq} T1$
- **affine part**: greatest affine submonad

2.5 Strong and affine monads

- **strong monad**: $\mathrm{st}_{X,Y} : X \times TY \to T(X \times Y) \rightsquigarrow \mathrm{st}' : TX \times Y \to T(X \times Y)$
- double strength $\mathrm{dst} : TX \times TY \xrightarrow{\mathrm{st}} T(TX \times Y) \xrightarrow{T\mathrm{st}'} T^2(X \times Y) \xrightarrow{\mu} T(X \times Y)$ (and dst')
- **affine monad**: $TX \times TY \xrightarrow{\mathrm{dst}} T(X \times Y) \xrightarrow{\langle T\pi_1, T\pi_2 \rangle} TX \times TY = \mathrm{id}$ or $\eta_1 : 1 \xrightarrow{\simeq} T1$
- **affine part**: greatest affine submonad

Example.

- Powerset $\mathcal{P} \rightsquigarrow \mathcal{P}_{\mathrm{ne}}$

2.5 Strong and affine monads

- **strong monad**: $\mathrm{st}_{X,Y} : X \times TY \to T(X \times Y) \rightsquigarrow \mathrm{st}' : TX \times Y \to T(X \times Y)$
- double strength $\mathrm{dst} : TX \times TY \xrightarrow{\mathrm{st}} T(TX \times Y) \xrightarrow{T\mathrm{st}'} T^2(X \times Y) \xrightarrow{\mu} T(X \times Y)$ (and dst')
- **affine monad**: $TX \times TY \xrightarrow{\mathrm{dst}} T(X \times Y) \xrightarrow{\langle T\pi_1, T\pi_2 \rangle} TX \times TY = \mathrm{id}$ or $\eta_1 : 1 \xrightarrow{\simeq} T1$
- **affine part**: greatest affine submonad

Example.

- Powerset $\mathcal{P} \rightsquigarrow \mathcal{P}_{\mathrm{ne}}$
- (Sub)distribution $\mathcal{S} \rightsquigarrow \mathcal{D}$

2.5 Strong and affine monads

- **strong monad**: $\mathrm{st}_{X,Y} : X \times TY \to T(X \times Y) \rightsquigarrow \mathrm{st}' : TX \times Y \to T(X \times Y)$
- double strength $\mathrm{dst} : TX \times TY \xrightarrow{\mathrm{st}} T(TX \times Y) \xrightarrow{T\mathrm{st}'} T^2(X \times Y) \xrightarrow{\mu} T(X \times Y)$ (and dst')
- **affine monad**: $TX \times TY \xrightarrow{\mathrm{dst}} T(X \times Y) \xrightarrow{\langle T\pi_1, T\pi_2 \rangle} TX \times TY = \mathrm{id}$ or $\eta_1 : 1 \xrightarrow{\simeq} T1$
- **affine part**: greatest affine submonad

Example.

- Powerset $\mathcal{P} \rightsquigarrow \mathcal{P}_{\mathrm{ne}}$
- (Sub)distribution $\mathcal{S} \rightsquigarrow \mathcal{D}$ with $\mathcal{D}X = \left\{\sum_{i \in I} p_i x_i \mid \sum p_i = 1, x_i \in X, I \text{ finite}\right\}$ and $\mathcal{S}X = \left\{\sum_{i \in I} p_i x_i \mid \sum p_i \leq 1, x_i \in X, I \text{ finite}\right\}$

2.5 Strong and affine monads

- **strong monad**: $\mathrm{st}_{X,Y} : X \times TY \to T(X \times Y) \rightsquigarrow \mathrm{st}' : TX \times Y \to T(X \times Y)$
- double strength $\mathrm{dst} : TX \times TY \xrightarrow{\mathrm{st}} T(TX \times Y) \xrightarrow{T\mathrm{st}'} T^2(X \times Y) \xrightarrow{\mu} T(X \times Y)$ (and dst')
- **affine monad**: $TX \times TY \xrightarrow{\mathrm{dst}} T(X \times Y) \xrightarrow{\langle T\pi_1, T\pi_2 \rangle} TX \times TY = \mathrm{id}$ or $\eta_1 : 1 \xrightarrow{\simeq} T1$
- **affine part**: greatest affine submonad

Example.

- Powerset $\mathcal{P} \rightsquigarrow \mathcal{P}_{\mathrm{ne}}$
- (Sub)distribution $\mathcal{S} \rightsquigarrow \mathcal{D}$ with $\mathcal{D}X = \left\{\sum_{i \in I} p_i x_i \mid \sum p_i = 1, x_i \in X, I \text{ finite}\right\}$ and $\mathcal{S}X = \left\{\sum_{i \in I} p_i x_i \mid \sum p_i \leq 1, x_i \in X, I \text{ finite}\right\}$
- Maybe $- + 1 \rightsquigarrow \mathrm{Id}$

2.6 Abstract smoothness

- diagrammatical condition

- diagrammatical condition

$$
\begin{tikzcd}
\Sigma T(X \times BX) \arrow[r, "\langle T\pi_1, T\pi_2 \rangle"] \arrow[rr, dashed, bend left, "\mathrm{mix}"] \arrow[d, "\lambda"'] & \Sigma(TX \times TBX) \arrow[r, "\mathrm{dst}"] & \Sigma T(X \times BX) \arrow[d, "\lambda"'] \\
T\Sigma(X \times BX) \arrow[d, "T\rho"'] & & T\Sigma(X \times BX) \arrow[d, "T\rho"'] \\
T^2 B\Sigma^* X \arrow[r, "\mu"] & TB\Sigma^* X & T^2 B\Sigma^* X \arrow[l, "\mu"']
\end{tikzcd}
$$

- diagrammatical condition

$$
\begin{tikzcd}
\Sigma T(X \times BX) \arrow[r, "\langle T\pi_1, T\pi_2 \rangle"] \arrow[d, "\lambda"'] \arrow[rr, dashed, bend left, "\mathrm{mix}"] & \Sigma(TX \times TBX) \arrow[r, "\mathrm{dst}"] & \Sigma T(X \times BX) \arrow[d, "\lambda"'] \\
T\Sigma(X \times BX) \arrow[d, "T\rho"'] & & T\Sigma(X \times BX) \arrow[d, "T\rho"'] \\
T^2 B\Sigma^* X \arrow[r, "\mu"] & TB\Sigma^* X & T^2 B\Sigma^* X \arrow[l, "\mu"']
\end{tikzcd}
$$

$$
\bigcup\{\rho(\sigma(\xi_1 \ldots \xi_n)) \mid \xi_i \in \mathrm{mix}\ \Xi_i\} = \bigcup\{\rho(\sigma(\xi_1 \ldots \xi_n)) \mid \xi_i \in \Xi_i\}
$$

$$
\text{where } \Xi_i \subset X \times BX
$$

2.7 Sketch of the proof

- Recall congruence $\forall \sigma, (\forall i, t_i \sim u_i) \Rightarrow \sigma(t_1 \ldots t_n) \sim \sigma(u_1 \ldots u_n)$

2.7 Sketch of the proof

- Recall congruence $\forall \sigma, (\forall i, t_i \sim u_i) \Rightarrow \sigma(t_1 ... t_n) \sim \sigma(u_1 ... u_n)$
  Prove $\mathrm{tr}(\sigma(t_1 ... t_n)) = [\![\sigma]\!](\mathrm{tr}\ t_1 ... \ \mathrm{tr}\ t_n)$

2.7 Sketch of the proof

- Recall congruence $\forall \sigma, (\forall i, t_i \sim u_i) \Rightarrow \sigma(t_1 ... t_n) \sim \sigma(u_1 ... u_n)$
  Prove $\mathrm{tr}(\sigma(t_1 ... t_n)) = [\![\sigma]\!](\mathrm{tr}\ t_1 ... \ \mathrm{tr}\ t_n)$

$$
\begin{tikzcd}
\Sigma^* X \arrow[r, "i^*", "\shortmid" marking] \arrow[d, "\Sigma^* \mathrm{tr}_k"', "\shortmid" marking] \arrow[dr, phantom, "?"] & X \arrow[r, "k", "\shortmid" marking] \arrow[d, "\mathrm{tr}_k"', "\shortmid" marking] \arrow[dr, phantom, "\checkmark"] & BX \arrow[d, "B\mathrm{tr}_k", "\shortmid" marking] \\
\Sigma^* Z \arrow[r, "g"', "\shortmid" marking] & Z \arrow[r, "\zeta"', "\shortmid" marking] & BZ
\end{tikzcd}
$$

2.7 Sketch of the proof

- Recall congruence $\forall\sigma, (\forall i, t_i \sim u_i) \Rightarrow \sigma(t_1 ... t_n) \sim \sigma(u_1 ... u_n)$
  Prove $\mathrm{tr}(\sigma(t_1 ... t_n)) = [\![\sigma]\!](\mathrm{tr}\ t_1 ...\ \mathrm{tr}\ t_n)$

$$\begin{tikzcd}
\Sigma^* X \arrow[r, "i^*"] \arrow[d, "\Sigma^*\mathrm{tr}_k"'] & X \arrow[r, "k"] \arrow[d, "\mathrm{tr}_k"'] \arrow[dl, phantom, "?"] & BX \arrow[d, "B\mathrm{tr}_k"] \arrow[dl, phantom, "\checkmark"] \\
\Sigma^* Z \arrow[r, "g"'] & Z \arrow[r, "\zeta"'] & BZ
\end{tikzcd}$$

- define $[\![-]\!]/g$: semantics of Z (B-coalgebra) + induction + trace

2.7 Sketch of the proof

- Recall congruence $\forall \sigma, (\forall i, t_i \sim u_i) \Rightarrow \sigma(t_1 ... t_n) \sim \sigma(u_1 ... u_n)$
  Prove $\mathrm{tr}(\sigma(t_1 ... t_n)) = [\![\sigma]\!](\mathrm{tr}\ t_1 ... \ \mathrm{tr}\ t_n)$

$$
\begin{tikzcd}
\Sigma^* X \arrow[r, "i^*", "\shortmid"{marking}] \arrow[d, "\Sigma^* \mathrm{tr}_k"', "\shortmid"{marking}] \arrow[dr, phantom, "?"] & X \arrow[r, "k", "\shortmid"{marking}] \arrow[d, "\mathrm{tr}_k"', "\shortmid"{marking}] \arrow[dr, phantom, "\checkmark"] & BX \arrow[d, "B\mathrm{tr}_k", "\shortmid"{marking}] \\
\Sigma^* Z \arrow[r, "g"', "\shortmid"{marking}] & Z \arrow[r, "\zeta"', "\shortmid"{marking}] & BZ
\end{tikzcd}
$$

- define $[\![-]\!]/g$: semantics of Z (B-coalgebra) + induction + trace
- $\Sigma^* X \nrightarrow B\Sigma^* X$ (with ρ^*)

2.7 Sketch of the proof

- Recall congruence $\forall \sigma, (\forall i, t_i \sim u_i) \Rightarrow \sigma(t_1 ... t_n) \sim \sigma(u_1 ... u_n)$
  Prove $\mathrm{tr}(\sigma(t_1 ... t_n)) = [\![\sigma]\!](\mathrm{tr}\ t_1 ... \ \mathrm{tr}\ t_n)$

$$\begin{tikzcd}
\Sigma^* X \arrow[r, "i^*", "\shortmid" marking] \arrow[d, "\Sigma^* \mathrm{tr}_k"', "\shortmid" marking] & X \arrow[r, "k", "\shortmid" marking] \arrow[d, "? \ \mathrm{tr}_k"', "\shortmid" marking] \arrow[dr, phantom, "\checkmark"] & BX \arrow[d, "B\mathrm{tr}_k", "\shortmid" marking] \\
\Sigma^* Z \arrow[r, "g"', "\shortmid" marking] & Z \arrow[r, "\zeta"', "\shortmid" marking] & BZ
\end{tikzcd}$$

- define $[\![-]\!]/g$: semantics of Z (B-coalgebra) + induction + trace
- $\Sigma^* X \nrightarrow B\Sigma^* X$ (with ρ^*) and $Z \nrightarrow BZ$

2.7 Sketch of the proof

- Recall congruence $\forall \sigma, (\forall i, t_i \sim u_i) \Rightarrow \sigma(t_1 ... t_n) \sim \sigma(u_1 ... u_n)$
  Prove $\mathrm{tr}(\sigma(t_1 ... t_n)) = [\![\sigma]\!](\mathrm{tr}\ t_1 ... \ \mathrm{tr}\ t_n)$

$$\begin{tikzcd}
\Sigma^* X \arrow[r, "i^*", "\shortmid" marking] \arrow[d, "\Sigma^*\mathrm{tr}_k"', "\shortmid" marking] & X \arrow[r, "k", "\shortmid" marking] \arrow[d, "?\ \mathrm{tr}_k"', "\shortmid" marking] \arrow[dr, phantom, "\checkmark"] & BX \arrow[d, "B\mathrm{tr}_k", "\shortmid" marking] \\
\Sigma^* Z \arrow[r, "g"', "\shortmid" marking] & Z \arrow[r, "\zeta"', "\shortmid" marking] & BZ
\end{tikzcd}$$

- define $[\![-]\!]/g$: semantics of Z (B-coalgebra) + induction + trace
- $\Sigma^* X \nrightarrow B\Sigma^* X$ (with ρ^*) and $Z \nrightarrow BZ$
- show $\overline{B}$-coalgebra morphisms:

2.7 Sketch of the proof

- Recall congruence $\forall \sigma, (\forall i, t_i \sim u_i) \Rightarrow \sigma(t_1 ... t_n) \sim \sigma(u_1 ... u_n)$
  Prove $\mathrm{tr}(\sigma(t_1 ... t_n)) = [\![\sigma]\!](\mathrm{tr}\ t_1 ... \ \mathrm{tr}\ t_n)$

$$
\begin{tikzcd}
\Sigma^* X \arrow[r, "i^*", "|" marking] \arrow[d, "\Sigma^* \mathrm{tr}_k"', "|" marking] \arrow[dr, phantom, "?"] & X \arrow[r, "k", "|" marking] \arrow[d, "\mathrm{tr}_k", "|" marking] \arrow[dr, phantom, "\checkmark"] & BX \arrow[d, "B\mathrm{tr}_k", "|" marking] \\
\Sigma^* Z \arrow[r, "g"', "|" marking] & Z \arrow[r, "\zeta"', "|" marking] & BZ
\end{tikzcd}
$$

- define $[\![-]\!]/g$: semantics of Z (B-coalgebra) + induction + trace
- $\Sigma^* X \nrightarrow B\Sigma^* X$ (with ρ^*) and $Z \nrightarrow BZ$
- show $\overline{B}$-coalgebra morphisms:
  - $\mathrm{tr} \circ i \quad \checkmark$

2.7 Sketch of the proof

- Recall congruence $\forall\sigma, (\forall i, t_i \sim u_i) \Rightarrow \sigma(t_1 ... t_n) \sim \sigma(u_1 ... u_n)$
  Prove $\mathrm{tr}(\sigma(t_1 ... t_n)) = [\![\sigma]\!](\mathrm{tr}\ t_1 ... \ \mathrm{tr}\ t_n)$

$$
\begin{tikzcd}
\Sigma^* X \arrow[r, "i^*", "\shortmid"{marking}] \arrow[d, "\Sigma^*\mathrm{tr}_k"', "\shortmid"{marking}] \arrow[dr, phantom, "?"] & X \arrow[r, "k", "\shortmid"{marking}] \arrow[d, "\mathrm{tr}_k"', "\shortmid"{marking}] \arrow[dr, phantom, "\checkmark"] & BX \arrow[d, "B\mathrm{tr}_k", "\shortmid"{marking}] \\
\Sigma^* Z \arrow[r, "g"', "\shortmid"{marking}] & Z \arrow[r, "\zeta"', "\shortmid"{marking}] & BZ
\end{tikzcd}
$$

- define $[\![-]\!]/g$: semantics of Z (B-coalgebra) + induction + trace
- $\Sigma^* X \nrightarrow B\Sigma^* X$ (with ρ^*) and $Z \nrightarrow BZ$
- show $\overline{B}$-coalgebra morphisms:
  - $\mathrm{tr} \circ i$ $\checkmark$
  - $g \circ \Sigma^*\mathrm{tr}$ more complicated : naturality + smoothness + map of distributive law of ρ^*

2.7 Sketch of the proof

- Recall congruence $\forall\sigma, (\forall i, t_i \sim u_i) \Rightarrow \sigma(t_1 ... t_n) \sim \sigma(u_1 ... u_n)$
  Prove $\mathrm{tr}(\sigma(t_1 ... t_n)) = [\![\sigma]\!](\mathrm{tr}\ t_1 ...\ \mathrm{tr}\ t_n)$

$$\begin{tikzcd}
\Sigma^* X \arrow[r, "i^*", "\shortmid"{marking}] \arrow[d, "\Sigma^*\mathrm{tr}_k"', "\shortmid"{marking}] \arrow[dr, phantom, "?"] & X \arrow[r, "k", "\shortmid"{marking}] \arrow[d, "\mathrm{tr}_k", "\shortmid"{marking}] \arrow[dr, phantom, "\checkmark"] & BX \arrow[d, "B\mathrm{tr}_k", "\shortmid"{marking}] \\
\Sigma^* Z \arrow[r, "g"', "\shortmid"{marking}] & Z \arrow[r, "\zeta"', "\shortmid"{marking}] & BZ
\end{tikzcd}$$

- define $[\![-]\!]/g$: semantics of Z (B-coalgebra) + induction + trace
- $\Sigma^* X \nrightarrow B\Sigma^* X$ (with ρ^*) and $Z \nrightarrow BZ$
- show $\overline{B}$-coalgebra morphisms:
  - $\mathrm{tr} \circ i$ ✓
  - $g \circ \Sigma^*\mathrm{tr}$ more complicated : naturality + smoothness + map of distributive law of ρ^*
- **maximality ?**

Theorem 6. Let $\mathbb{C}$ be a cartesian category,

2.8 To sum up: the theorem

Theorem 7. Let $\mathbb{C}$ be a cartesian category, T be a strong **affine** monad *for effects*,

2.8 To sum up: the theorem

Theorem 8. Let $\mathbb{C}$ be a cartesian category, T be a strong **affine** monad *for effects*, B an endofunctor *for behaviour* that extends to $\mathrm{Kl}(T)$,

2.8 To sum up: the theorem

Theorem 9. Let $\mathbb{C}$ be a cartesian category, T be a strong **affine** monad *for effects*, B an endofunctor *for behaviour* that extends to $\mathrm{Kl}(T)$, Σ a endofunctor *for syntax* that extends to $\mathrm{Kl}(T)$ with all free objects $(\Sigma^* X)$,

2.8 To sum up: the theorem

Theorem 10. Let $\mathbb{C}$ be a cartesian category, T be a strong **affine** monad *for effects*, B an endofunctor *for behaviour* that extends to $\mathrm{Kl}(T)$, Σ a endofunctor *for syntax* that extends to $\mathrm{Kl}(T)$ with all free objects $(\Sigma^* X)$, let Z be the final B-algebra, suppose there is an infinitary trace situation, and

Theorem 11. Let $\mathbb{C}$ be a cartesian category, T be a strong **affine** monad *for effects*, B an endofunctor *for behaviour* that extends to $\mathrm{Kl}(T)$, Σ a endofunctor *for syntax* that extends to $\mathrm{Kl}(T)$ with all free objects $(\Sigma^* X)$, let Z be the final B-algebra, suppose there is an infinitary trace situation, and let $\rho : \Sigma(X \times BX) \to TB\Sigma^* X$ be a natural transformation *representing Trace-GSOS rules*

Theorem 12. Let $\mathbb{C}$ be a cartesian category, T be a strong **affine** monad *for effects*, B an endofunctor *for behaviour* that extends to $\mathrm{Kl}(T)$, Σ a endofunctor *for syntax* that extends to $\mathrm{Kl}(T)$ with all free objects $(\Sigma^* X)$, let Z be the final B-algebra, suppose there is an infinitary trace situation, and let $\rho : \Sigma(X \times BX) \to TB\Sigma^* X$ be a natural transformation *representing Trace-GSOS rules* such that ρ is **smooth** and is a map of distributive laws,

Theorem 13. Let $\mathbb{C}$ be a cartesian category, T be a strong **affine** monad *for effects*, B an endofunctor *for behaviour* that extends to $\mathrm{Kl}(T)$, Σ a endofunctor *for syntax* that extends to $\mathrm{Kl}(T)$ with all free objects $(\Sigma^* X)$, let Z be the final B-algebra, suppose there is an infinitary trace situation, and let $\rho : \Sigma(X \times BX) \to TB\Sigma^* X$ be a natural transformation *representing Trace-GSOS rules* such that ρ is **smooth** and is a map of distributive laws, **and that is sublinear (?)**

2.8 To sum up: the theorem

Theorem 14. Let $\mathbb{C}$ be a cartesian category, T be a strong **affine** monad *for effects*, B an endofunctor *for behaviour* that extends to $\mathrm{Kl}(T)$, Σ a endofunctor *for syntax* that extends to $\mathrm{Kl}(T)$ with all free objects $(\Sigma^* X)$, let Z be the final B-algebra, suppose there is an infinitary trace situation, and let $\rho : \Sigma(X \times BX) \to TB\Sigma^* X$ be a natural transformation *representing Trace-GSOS rules* such that ρ is **smooth** and is a map of distributive laws, **and that is sublinear (?)** then trace equivalence is a congruence.

Theorem 15. Let $\mathbb{C}$ be a cartesian category, T be a strong **affine** monad *for effects*, B an endofunctor *for behaviour* that extends to $\mathrm{Kl}(T)$, Σ a endofunctor *for syntax* that extends to $\mathrm{Kl}(T)$ with all free objects $(\Sigma^* X)$, let Z be the final B-algebra, suppose there is an infinitary trace situation, and let $\rho : \Sigma(X \times BX) \to TB\Sigma^* X$ be a natural transformation *representing Trace-GSOS rules* such that ρ is **smooth** and is a map of distributive laws, **and that is sublinear (?)** then trace equivalence is a congruence. (Hopefully !)

3 Conclusion

3 Conclusion

- concrete proof ✓

*

3 Conclusion

- concrete proof ✓
- abstract:
  - missing the "sublinearity" $\rightsquigarrow$ using presheaves?

*

3 Conclusion

- concrete proof ✓
- abstract:
  - missing the "sublinearity" $\rightsquigarrow$ using presheaves?
  - moving to Eilenberg-Moore to have *finite*, *partial* and trace semantics as a **final coalgebra**

*

3 Conclusion

- concrete proof ✓
- abstract:
  - missing the "sublinearity" $\rightsquigarrow$ using presheaves?
  - moving to Eilenberg-Moore to have *finite*, *partial* and trace semantics as a **final coalgebra**
- thank you and bravo if you are still there !

*

3 Conclusion

- concrete proof ✓
- abstract:
  - missing the “sublinearity” $\rightsquigarrow$ using presheaves?
  - moving to Eilenberg-Moore to have *finite*, *partial* and trace semantics as a **final coalgebra**
- thank you and bravo if you are still there !

~ *The End*[*] ~

Powered by Typst

[*] for today…