(Abstract) GSOS for Trace Equivalence – Early Ideas

CALCO 2025 – Glasgow

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

16 June 2025

[*]ENS de Lyon, Université Savoie Mont Blanc
[†]University of Birmingham
[‡]**Friedrich-Alexander-Universität Erlangen-Nürnberg**
[§]Syddansk Universitet

1 (Abstract) GSOS

1.1 GSOS

- a **framework** for specifying reduction rules and semantics of systems $\to$ rule format

1.1 GSOS

- a **framework** for specifying reduction rules and semantics of systems $\to$ rule format
- **syntax**: operations with arities

1.1 GSOS

- a **framework** for specifying reduction rules and semantics of systems $\to$ rule format
- **syntax**: operations with arities
- **behaviour** (LTS): $x \xrightarrow{a} x'$ ($a \in L$ for some fixed set L)

1.1 GSOS

- a **framework** for specifying reduction rules and semantics of systems $\to$ rule format
- **syntax**: operations with arities
- **behaviour** (LTS): $x \xrightarrow{a} x'$ ($a \in L$ for some fixed set L)
- **GSOS rules**

$$\frac{x_1 \xrightarrow{a_1} y_1 \quad \ldots \quad x_1 \xrightarrow{a_1} y_k \quad x_1 \xrightarrow{a_2} y_{k+1} \quad \ldots \qquad x_2 \xrightarrow{a_1} y_l \quad \ldots \qquad\qquad x_1 \stackrel{b_1}{\not\to} \quad \ldots}{\sigma(x_1 \ldots x_n) \xrightarrow{c} u}$$

  - x_i, y_j distinct **variables**
  - u term with variables x_i, y_j

- a **framework** for specifying reduction rules and semantics of systems $\rightarrow$ rule format
- **syntax**: operations with arities
- **behaviour** (LTS): $x \xrightarrow{a} x'$ ($a \in L$ for some fixed set L)
- **GSOS rules**

$$\frac{x_1 \xrightarrow{a_1} y_1 \quad \dots \quad x_1 \xrightarrow{a_1} y_k \quad x_1 \xrightarrow{a_2} y_{k+1} \quad \dots \qquad x_2 \xrightarrow{a_1} y_l \quad \dots \qquad\quad x_1 \overset{b_1}{\nrightarrow} \quad \dots}{\sigma(x_1 \dots x_n) \xrightarrow{c} u}$$

  - x_i, y_j distinct **variables**
  - u term with variables x_i, y_j
- set of GSOS rules $\Rightarrow$ behaviour of terms

Example.

$$t \Coloneqq 0 \mid a.t \quad \forall a \in L \mid t + t$$

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

1.1 GSOS

Example.

$$t ::= 0 \mid a.t \quad \forall a \in L \mid t + t$$

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

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

Example.

$$t ::= 0 \mid a.t \quad \forall a \in L \mid t + t$$

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

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

1.2 Abstract GSOS

- syntax functor Σ and behaviour functor H (on a category $\mathbb{C}$)

1.2 Abstract GSOS

- syntax functor Σ and behaviour functor H (on a category $\mathbb{C}$)

$t ::= 0 \mid a.t \quad \forall a \in L \mid t + t \quad \rightsquigarrow \quad \Sigma X = 1 + A \times X + X^2$

$x \xrightarrow{a} x' \quad \rightsquigarrow \quad x' \in k(x)(a)$ with $k : X \to HX = \mathcal{P}(X)^L$

- rules $\rightsquigarrow$ a **natural transformation** $\rho_X : \Sigma(X \times HX) \to H\Sigma^\star X$

1.2 Abstract GSOS

- syntax functor Σ and behaviour functor H (on a category $\mathbb{C}$)

$$t ::= 0 \mid a.t \quad \forall a \in L \mid t + t \quad \rightsquigarrow \quad \Sigma X = 1 + A \times X + X^2$$

$$x \xrightarrow{a} x' \quad \rightsquigarrow \quad x' \in k(x)(a) \text{ with } k : X \to HX = \mathcal{P}(X)^L$$

- rules $\rightsquigarrow$ a **natural transformation** $\rho_X : \Sigma(X \times HX) \to H\Sigma^\star X$

Example.

$$\rho : 1 + A \times \left(X \times \mathcal{P}(X)^L\right) + \left(X \times \mathcal{P}(X)^L\right)^2 \to \mathcal{P}(\Sigma^\star X)^L$$

1.2 Abstract GSOS

- syntax functor Σ and behaviour functor H (on a category $\mathbb{C}$)

  $t ::= 0 \mid a.t \quad \forall a \in L \mid t + t \quad \rightsquigarrow \quad \Sigma X = 1 + A \times X + X^2$

  $x \xrightarrow{a} x' \quad \rightsquigarrow \quad x' \in k(x)(a)$ with $k : X \to HX = \mathcal{P}(X)^L$

- rules $\rightsquigarrow$ a **natural transformation** $\rho_X : \Sigma(X \times HX) \to H\Sigma^\star X$

Example.

$$\rho : 1 + A \times \left(X \times \mathcal{P}(X)^L\right) + \left(X \times \mathcal{P}(X)^L\right)^2 \to \mathcal{P}(\Sigma^\star X)^L$$

- $\rho(*)(a) = \emptyset$

- syntax functor Σ and behaviour functor H (on a category $\mathbb{C}$)
- rules $\rightsquigarrow$ a **natural transformation** $\rho_X : \Sigma(X \times HX) \to H\Sigma^\star X$

Example.

$$\rho : 1 + A \times \left(X \times \mathcal{P}(X)^L\right) + \left(X \times \mathcal{P}(X)^L\right)^2 \to \mathcal{P}(\Sigma^\star X)^L$$

- $\rho(*)(a) = \emptyset$
- $\rho((a', t, T))(a) = \{t\}$ if $a = a'$ and $\emptyset$ otherwise

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

1.2 Abstract GSOS

- syntax functor Σ and behaviour functor H (on a category $\mathbb{C}$)
- rules $\rightsquigarrow$ a **natural transformation** $\rho_X : \Sigma(X \times HX) \to H\Sigma^\star X$

Example.

$$\rho : 1 + A \times \left(X \times \mathcal{P}(X)^L\right) + \left(X \times \mathcal{P}(X)^L\right)^2 \to \mathcal{P}(\Sigma^\star X)^L$$

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

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

2 Trace equivalence

2.1 Program equivalence

- many different notions (linear-time branching-time spectrum), eg. bisimilarity, trace equivalence

2.1 Program equivalence

- many different notions (linear-time branching-time spectrum), eg. bisimilarity, trace equivalence
- important property: **congruence**

$$\forall \sigma \in \mathcal{O}, (\forall i, x_i \sim y_i) \Rightarrow \sigma(x_1..x_n) \sim \sigma(y_1...y_n)$$

2.1 Program equivalence

- many different notions (linear-time branching-time spectrum), eg. bisimilarity, trace equivalence
- important property: **congruence**

$$\forall \sigma \in \mathcal{O}, (\forall i, x_i \sim y_i) \Rightarrow \sigma(x_1..x_n) \sim \sigma(y_1...y_n)$$

Theorem. (D. Turi and G. Plotkin, 1997)

$$\text{abstract GSOS} \Rightarrow \text{bisimilarity is a congruence}$$

2.1 Program equivalence

- many different notions (linear-time branching-time spectrum), eg. bisimilarity, trace equivalence
- important property: **congruence**

$$\forall \sigma \in \mathcal{O}, (\forall i, x_i \sim y_i) \Rightarrow \sigma(x_1..x_n) \sim \sigma(y_1...y_n)$$

Theorem. (D. Turi and G. Plotkin, 1997)

abstract GSOS $\Rightarrow$ bisimilarity is a congruence

- what about trace equivalence?

2.2 Trace and trace equivalence

- partial finite traces

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

2.2 Trace and trace equivalence

- partial finite traces

$$\mathrm{tr}(x) = \bigcup_{n \in \mathbb{N}} \left\{ w \in L^n \;\middle|\; x \overset{w_1}{\to} x_1 \overset{w_2}{\to} \ldots \overset{w_n}{\to} x_n \right\}$$

- $\mathrm{tr}(x) \in \mathfrak{T} := \{A \subseteq L^\star \mid \varepsilon \in A \land A \text{ prefix-closed}\}$

2.2 Trace and trace equivalence

- partial finite traces

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

- $\mathrm{tr}(x) \in \mathfrak{T} := \{A \subseteq L^\star \mid \varepsilon \in A \wedge A \text{ prefix-closed}\}$

Remark. tr is the unique map $X \to \mathfrak{T}$ such that $a.w \in \mathrm{tr}(x) \Leftrightarrow x \xrightarrow{a} y \wedge w \in \mathrm{tr}(y)$ ("*coalgebra morphism*")

- partial finite traces

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

- $\mathrm{tr}(x) \in \mathfrak{T} := \{A \subseteq L^\star \mid \varepsilon \in A \wedge A \text{ prefix-closed}\}$

Remark. tr is the unique map $X \to \mathfrak{T}$ such that $a.w \in \mathrm{tr}(x) \Leftrightarrow x \xrightarrow{a} y \wedge w \in \mathrm{tr}(y)$ ("*coalgebra morphism*")

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

$$\mathrm{tr}(x) = \bigcup_{n \in \mathbb{N}} \left\{ w \in L^n \,\middle|\, x \overset{w_1}{\to} x_1 \overset{w_2}{\to} \ldots \overset{w_n}{\to} x_n \right\}$$

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

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

2.2 Trace and trace equivalence

$$\mathrm{tr}(x) = \bigcup_{n \in \mathbb{N}} \left\{ w \in L^n \;\middle|\; x \overset{w_1}{\to} x_1 \overset{w_2}{\to} \ldots \overset{w_n}{\to} x_n \right\}$$

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

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

2.2 Trace and trace equivalence

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

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

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

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

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

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

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

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

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

2.2 Trace and trace equivalence

$$\mathrm{tr}(x) = \bigcup_{n \in \mathbb{N}} \left\{ w \in L^n \,\middle|\, x \overset{w_1}{\to} x_1 \overset{w_2}{\to} \dots \overset{w_n}{\to} x_n \right\}$$

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

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

- De Simone rule format (R. de Simone, 1985):

2.3 De Simone Format

- De Simone rule format (R. de Simone, 1985):
  - no negative premise
  - at most one premise per variable
  - linearity: each variable at most once in u + no x_i such that $x_i \xrightarrow{a_i} y_i$ in u

$$\frac{x_i \xrightarrow{a_i} y_i \quad \dots \quad x_j \xrightarrow{a_j} y_j}{\sigma(x_1 \dots x_n) \xrightarrow{c} u}$$

- De Simone rule format (R. de Simone, 1985):
  - no negative premise
  - at most one premise per variable
  - linearity: each variable at most once in u + no x_i such that $x_i \xrightarrow{a_i} y_i$ in u

$$\frac{x_i \xrightarrow{a_i} y_i \quad \dots \quad x_j \xrightarrow{a_j} y_j}{\sigma(x_1 \dots x_n) \xrightarrow{c} u}$$

Theorem. (B. Bloom, 1994, restricted to GSOS)

De Simone $\Rightarrow$ trace equivalence is a congruence

Example. (With negative premises)

$$\ldots + \quad \frac{x \xrightarrow{a} y}{f(x) \xrightarrow{a} g(y)} \forall a \qquad \frac{x \stackrel{b}{\nrightarrow}}{g(x) \xrightarrow{b} x}$$

Example. (With negative premises)

$$\dots + \qquad \frac{x \xrightarrow{a} y}{f(x) \xrightarrow{a} g(y)} \forall a \qquad \qquad \frac{x \overset{b}{\nrightarrow}}{g(x) \xrightarrow{b} x}$$

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

2.3 De Simone Format

Example. (With more than one premise per variable)

$$\ldots + \qquad \frac{x \xrightarrow{a} y}{f'(x) \xrightarrow{a} g'(y)}\forall a \qquad \frac{x \xrightarrow{b} x' \quad x \xrightarrow{c} x''}{g'(x) \xrightarrow{a} x' + x''}$$

Example. (With more than one premise per variable)

$$\dots + \qquad \frac{x \overset{a}{\rightarrow} y}{f'(x) \overset{a}{\rightarrow} g'(y)}\,\forall a \qquad\qquad \frac{x \overset{b}{\rightarrow} x' \quad x \overset{c}{\rightarrow} x''}{g'(x) \overset{a}{\rightarrow} x' + x''}$$

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

2.4 Abstract Trace Semantics

- abstract?

2.4 Abstract Trace Semantics

- abstract?

Remark. (B. Klin, 2005 and 2009) test suites and logical distributive laws

- abstract?

Remark. (B. Klin, 2005 and 2009) test suites and logical distributive laws

- recall behaviour $k : X \to HX = \mathcal{P}(X)^L$

- abstract?

Remark. (B. Klin, 2005 and 2009) test suites and logical distributive laws

- recall behaviour $k : X \to HX = \mathcal{P}(X)^L \cong (\mathcal{P}_{\mathrm{ne}}(X) + 1)^L = BTX$
  - $T = \mathcal{P}_{\mathrm{ne}}$: **effectful/branching** behaviour (non-determinism)
  - $B = (-+1)^L$: **pure** behaviour (labels and termination)

2.4 Abstract Trace Semantics

- abstract?

Remark. (B. Klin, 2005 and 2009) test suites and logical distributive laws

- recall behaviour $k : X \to HX = \mathcal{P}(X)^L \cong (\mathcal{P}_{\text{ne}}(X) + 1)^L = BTX$
  - $T = \mathcal{P}_{\text{ne}}$: **effectful/branching** behaviour (non-determinism)
  - $B = (-+1)^L$: **pure** behaviour (labels and termination)
- $\nu B = \mathfrak{T}$ set of (partial finite) traces: final B-coalgebra

2.4 Abstract Trace Semantics

- abstract?

Remark. (B. Klin, 2005 and 2009) test suites and logical distributive laws

- recall behaviour $k : X \to HX = \mathcal{P}(X)^L \cong (\mathcal{P}_{\mathrm{ne}}(X) + 1)^L = BTX$
  - $T = \mathcal{P}_{\mathrm{ne}}$: **effectful/branching** behaviour (non-determinism)
  - $B = (-+1)^L$: **pure** behaviour (labels and termination)
- $\nu B = \mathfrak{T}$ set of (partial finite) traces: final B-coalgebra
- categories of algebras (Eilenberg-Moore) $\mathbf{Set}^T = \mathbf{CSLat}$ category of unbounded complete semi-lattices

2.4 Abstract Trace Semantics

- abstract?

Remark. (B. Klin, 2005 and 2009) test suites and logical distributive laws

- recall behaviour $k : X \to HX = \mathcal{P}(X)^L \cong (\mathcal{P}_{\mathrm{ne}}(X) + 1)^L = BTX$
  - $T = \mathcal{P}_{\mathrm{ne}}$: **effectful/branching** behaviour (non-determinism)
  - $B = (-+1)^L$: **pure** behaviour (labels and termination)
- $\nu B = \mathfrak{T}$ set of (partial finite) traces: final B-coalgebra
- categories of algebras (Eilenberg-Moore) $\mathbf{Set}^T = \mathbf{CSLat}$ category of unbounded complete semi-lattices
- B lifts (with $\delta^B : TB \to BT$) and $\nu\overline{B} = \nu B$

2.4 Abstract Trace Semantics

- abstract GSOS theory in **CSLat**:

$$\overline{\rho} : \overline{\Sigma}\Big(X \times \overline{B}X\Big) \to \overline{B}\Big(\overline{\Sigma}^{\star} X\Big)$$

- abstract GSOS theory in **CSLat**:

$$\overline{\rho} : \overline{\Sigma}\left(X \times \overline{B}X\right) \to \overline{B}\left(\overline{\Sigma}^{\star}X\right)$$

- universal semantics $\tau : \mu\overline{\Sigma} \to \nu\overline{B}$

$$\begin{tikzcd}
\overline{\Sigma}\mu\overline{\Sigma} \arrow[r] \arrow[d, dashed, "\overline{\Sigma}\tau"'] & \mu\overline{\Sigma} \arrow[r] \arrow[d, dashed, "\tau"'] & \overline{B}\mu\overline{\Sigma} \arrow[d, dashed, "\overline{B}\tau"'] \\
\overline{\Sigma}\nu\overline{B} \arrow[r] & \nu\overline{B} \arrow[r] & \overline{B}\nu\overline{B}
\end{tikzcd}$$

2.4 Abstract Trace Semantics

- abstract GSOS theory in **CSLat**:

$$\overline{\rho} : \overline{\Sigma}\left(X \times \overline{B}X\right) \to \overline{B}\left(\overline{\Sigma}^\star X\right)$$

- universal semantics $\tau : \mu\overline{\Sigma} \to \nu\overline{B}$

$$\begin{tikzcd}
\overline{\Sigma}\mu\overline{\Sigma} \arrow[r] \arrow[d, dashed, "\overline{\Sigma}\tau"'] & \mu\overline{\Sigma} \arrow[r] \arrow[d, dashed, "\tau"'] & \overline{B}\mu\overline{\Sigma} \arrow[d, dashed, "\overline{B}\tau"'] \\
\overline{\Sigma}\nu\overline{B} \arrow[r] & \nu\overline{B} \arrow[r] & \overline{B}\nu\overline{B}
\end{tikzcd}$$

- $\mathrm{tr} : \mu\Sigma \xrightarrow{\xi} \mu\overline{\Sigma} \xrightarrow{\tau} \nu B$

2.4 Abstract Trace Semantics

- abstract GSOS theory in **CSLat**:

$$\overline{\rho} : \overline{\Sigma}\left(X \times \overline{B}X\right) \to \overline{B}\left(\overline{\Sigma}^{\star} X\right)$$

- universal semantics $\tau : \mu\overline{\Sigma} \to \nu\overline{B}$

$$\begin{tikzcd}
\overline{\Sigma}\mu\overline{\Sigma} \arrow[r] \arrow[d, dashed, "\overline{\Sigma}\tau"'] & \mu\overline{\Sigma} \arrow[r] \arrow[d, dashed, "\tau"'] & \overline{B}\mu\overline{\Sigma} \arrow[d, dashed, "\overline{B}\tau"'] \\
\overline{\Sigma}\nu\overline{B} \arrow[r] & \nu\overline{B} \arrow[r] & \overline{B}\nu\overline{B}
\end{tikzcd}$$

- $\mathrm{tr} : \mu\Sigma \xrightarrow{\xi} \mu\overline{\Sigma} \xrightarrow{\tau} \nu B$

- left square $\Rightarrow$ tr is a congruence 😀

- abstract GSOS theory in **CSLat**:

$$\overline{\rho} : \overline{\Sigma}\big(X \times \overline{B}X\big) \to \overline{B}\big(\overline{\Sigma}^{\star}X\big)$$

- universal semantics $\tau : \mu\overline{\Sigma} \to \nu\overline{B}$

$$\begin{tikzcd}
\overline{\Sigma}\mu\overline{\Sigma} \arrow[r] \arrow[d, dashed, "\overline{\Sigma}\tau"'] & \mu\overline{\Sigma} \arrow[r] \arrow[d, dashed, "\tau"'] & \overline{B}\mu\overline{\Sigma} \arrow[d, dashed, "\overline{B}\tau"'] \\
\overline{\Sigma}\nu\overline{B} \arrow[r] & \nu\overline{B} \arrow[r] & \overline{B}\nu\overline{B}
\end{tikzcd}$$

- $\mathrm{tr} : \mu\Sigma \xrightarrow{\xi} \mu\overline{\Sigma} \xrightarrow{\tau} \nu B$
- left square $\Rightarrow$ tr is a congruence 😀
- how to get $\overline{\rho}$ natural transformation in **CSLat**?

2.5 Trace-GSOS

- rule format
  - for each x_i and for each a, either $x_i \xrightarrow{a} y$ or $x_i \overset{a}{\nrightarrow}$

$$\frac{x_1 \xrightarrow{a_1} y_1 \quad \dots \quad x_1 \xrightarrow{a_k} y_k \quad x_1 \overset{b_1}{\nrightarrow} \quad \dots \qquad\qquad x_2 \dots \qquad\qquad \dots}{\sigma(x_1 \dots x_n) \xrightarrow{c} u}$$

2.5 Trace-GSOS

- rule format
  - for each x_i and for each a, either $x_i \xrightarrow{a} y$ or $x_i \overset{a}{\nrightarrow}$

$$\frac{x_1 \xrightarrow{a_1} y_1 \quad \ldots \quad x_1 \xrightarrow{a_k} y_k \quad x_1 \overset{b_1}{\nrightarrow} \quad \ldots \qquad\qquad x_2 \ldots \qquad\qquad \ldots}{\sigma(x_1 \ldots x_n) \xrightarrow{c} u}$$

Remark. Only **pure** premises: observe each variable at each label **once and only once**

2.5 Trace-GSOS

- rule format
  - for each x_i and for each a, either $x_i \xrightarrow{a} y$ or $x_i \overset{a}{\nrightarrow}$

$$\frac{x_1 \xrightarrow{a_1} y_1 \quad \ldots \quad x_1 \xrightarrow{a_k} y_k \quad x_1 \overset{b_1}{\nrightarrow} \quad \ldots \qquad x_2 \ldots \qquad \ldots}{\sigma(x_1 \ldots x_n) \xrightarrow{c} u}$$

Remark. Only **pure** premises: observe each variable at each label **once and only once**

- natural transformation

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

2.5 Trace-GSOS

- rule format
  - for each x_i and for each a, either $x_i \xrightarrow{a} y$ or $x_i \stackrel{a}{\nrightarrow}$

$$\frac{x_1 \xrightarrow{a_1} y_1 \quad \dots \quad x_1 \xrightarrow{a_k} y_k \quad x_1 \stackrel{b_1}{\nrightarrow} \quad \dots \qquad x_2 \dots \qquad \dots}{\sigma(x_1 \dots x_n) \xrightarrow{c} u}$$

Remark. Only **pure** premises: observe each variable at each label **once and only once**

- natural transformation

$$\rho : \Sigma(X \times BTX) \to BT\Sigma^\star X$$

2.5 Trace-GSOS

- rule format
  - for each x_i and for each a, either $x_i \xrightarrow{a} y$ or $x_i \overset{a}{\not\to}$

$$\frac{x_1 \xrightarrow{a_1} y_1 \quad \dots \quad x_1 \xrightarrow{a_k} y_k \quad x_1 \overset{b_1}{\not\to} \quad \dots \qquad x_2 \dots \qquad \dots}{\sigma(x_1 \dots x_n) \xrightarrow{c} u}$$

Remark. Only **pure** premises: observe each variable at each label **once and only once**

- natural transformation

$$\rho : \Sigma(X \times BX) \to BT\Sigma^\star X$$

- rule format
  - for each x_i and for each a, either $x_i \xrightarrow{a} y$ or $x_i \overset{a}{\nrightarrow}$

$$\frac{x_1 \xrightarrow{a_1} y_1 \quad \ldots \quad x_1 \xrightarrow{a_k} y_k \quad x_1 \overset{b_1}{\nrightarrow} \quad \ldots \qquad x_2 \ldots \qquad \ldots}{\sigma(x_1 \ldots x_n) \xrightarrow{c} u}$$

Remark. Only **pure** premises: observe each variable at each label **once and only once**

- natural transformation

$$\rho : \Sigma(X \times BX) \to BT\Sigma^{\star}X$$

- De Simone rules $\to$ Trace-GSOS rules:

$$\mathcal{S}(r) = \{r' \text{ Trace-GSOS rule} \mid \text{each premise of } r \text{ is a premise of } r'\}$$

2.5 Trace-GSOS

- characterize "good" Trace-GSOS specifications: smooth and minimally linear

2.5 Trace-GSOS

- characterize "good" Trace-GSOS specifications: smooth and minimally linear

Theorem. "good" Trace-GSOS specifications $\Longleftrightarrow$ De Simone specifications

- characterize "good" Trace-GSOS specifications: smooth and minimally linear

Theorem. "good" Trace-GSOS specifications $\iff$ De Simone specifications

Theorem. "good" Trace-GSOS specifications $\implies$ ρ yields a **natural** transformation $\overline{\rho}$ in **CSLat**

$$\overline{\rho} : \Sigma(X \times BX) \xrightarrow{\rho} BT\Sigma^{\star}X \xrightarrow{BT\xi_X} BT\overline{\Sigma}^{\star}X \xrightarrow{B\sigma_X} B\overline{\Sigma}^{\star}X$$

2.5 Trace-GSOS

- characterize "good" Trace-GSOS specifications: smooth and minimally linear

Theorem. "good" Trace-GSOS specifications $\iff$ De Simone specifications

Theorem. "good" Trace-GSOS specifications $\implies$ ρ yields a **natural** transformation $\overline{\rho}$ in **CSLat**

$$\overline{\rho} : \Sigma(X \times BX) \xrightarrow{\rho} BT\Sigma^\star X \xrightarrow{BT\xi_X} BT\overline{\Sigma}^\star X \xrightarrow{B\sigma_X} B\overline{\Sigma}^\star X$$

- 🚧 find an easy/better abstract characterization of "good" Trace-GSOS

3 Conclusion...

- Eilenberg-Moore trace semantics

3 Conclusion…

- Eilenberg-Moore trace semantics
- abstract GSOS in Eilenberg-Moore category

3 Conclusion...

- Eilenberg-Moore trace semantics
- abstract GSOS in Eilenberg-Moore category
- works for LTSs

3 Conclusion…

- Eilenberg-Moore trace semantics
- abstract GSOS in Eilenberg-Moore category
- works for LTSs
- 🚧 easier to check abstract condition

3 Conclusion...

- Eilenberg-Moore trace semantics
- abstract GSOS in Eilenberg-Moore category
- works for LTSs
- 🚧 easier to check abstract condition
- 🚧 other monads: probabilistic distributions

3 Conclusion...

- Eilenberg-Moore trace semantics
- abstract GSOS in Eilenberg-Moore category
- works for LTSs
- 🚧 easier to check abstract condition
- 🚧 other monads: probabilistic distributions
- 🚧 adequacy of operational models

3 Conclusion…

- Eilenberg-Moore trace semantics
- abstract GSOS in Eilenberg-Moore category
- works for LTSs
- 🚧 easier to check abstract condition
- 🚧 other monads: probabilistic distributions
- 🚧 adequacy of operational models

~ Thank you for your attention ~