

Compositionality in Coalgebraic Trace Semantics

Robin Jourde^{*}, Henning Urbat[†], Sergey Goncharov[‡], Stelios Tsampas[§], Jonas Forster[†]

September 3, 2026

CONCUR 2026 – Liverpool

^{*} Université Savoie Mont Blanc, France

[†] Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

[‡] University of Birmingham, UK

[§] Syddansk Universitet, Denmark

Program Equivalence

What does it mean for two programs to be the same?

- **behave** the same
- many different notions: e.g. bisimilarity, trace equivalence
- important property: **compositionality** (a.k.a. congruence)

$$t \sim u \Rightarrow \forall C, C[t] \sim C[u]$$

Theorem (Turi and Plotkin, 1997)

abstract GSOS $\Rightarrow$ bisimilarity is a congruence

This paper abstract (categorical) format for compositional **trace equivalence**

- ▷ **Coalgebraic Trace Semantics**
- ▷ **Abstract GSOS**
- ▷ **De Simone Laws**

Coalgebraic Trace Semantics

Labelled Transition Systems with Explicit Termination

Syntax Set of operations $\mathcal{O}$, with arities

Example

$\mathcal{O} = \{0^0, +^2, f^1\} \cup \{a.^1 \mid a \in L\} \rightsquigarrow$ Terms:

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

e.g. for $a, b, c \in L$

$$f(0) \quad a.b.0 = ab \quad a.(b.a.0 + f(a.c.0)) = a(ba + f(ac)) \quad \dots$$

Labelled Transition Systems with Explicit Termination

Syntax Set of operations $\mathcal{O}$, with arities

Example

$\mathcal{O} = \{0^0, +^2, f^1\} \cup \{a.^1 \mid a \in L\} \rightsquigarrow$ Terms:

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

e.g. for $a, b, c \in L$

$$f(0) \quad a.b.0 = ab \quad a.(b.a.0 + f(a.c.0)) = a(ba + f(ac)) \quad \dots$$

- Polynomial functor $\Sigma : \mathbf{Set} \rightarrow \mathbf{Set}$
- Terms $\mu\Sigma = \text{initial } \Sigma\text{-algebra}$

Example

$$\Sigma X = 1 + X^2 + X + L \times X$$

Labelled Transition Systems with Explicit Termination

Behaviour

- transition $t \xrightarrow{a} t'$ ($a \in L$ for some fixed set of labels L)
- termination $t \downarrow$

Labelled Transition Systems with Explicit Termination

Behaviour

- transition $t \xrightarrow{a} t'$ ($a \in L$ for some fixed set of labels L)
- termination $t \downarrow$
- Functor $BX = L \times X + 1$ modelling **interaction**
- Monad $T = \mathcal{P}$ (powerset) modelling **computational effects** (non-determinism)
- Coalgebra $c: X \rightarrow TBX = \mathcal{P}(L \times X + 1)$

$$t \xrightarrow{a} t' \Leftrightarrow (a, t') \in c(t) \qquad t \downarrow \Leftrightarrow * \in c(t)$$

Labelled Transition Systems with Explicit Termination

Behaviour

- transition $t \xrightarrow{a} t'$ ($a \in L$ for some fixed set of labels L)
- termination $t \downarrow$
- Functor $BX = L \times X + 1$ modelling **interaction**
- Monad $T = \mathcal{P}$ (powerset) modelling **computational effects** (non-determinism)
- Coalgebra $c: X \rightarrow TBX = \mathcal{P}(L \times X + 1)$

$$t \xrightarrow{a} t' \Leftrightarrow (a, t') \in c(t)$$

$$t \downarrow \Leftrightarrow * \in c(t)$$

Definition (Trace Semantics for LTSs)

- a **completed trace** at t = a string $a_1 \cdots a_n \in L^\star$ s.t. $t \xrightarrow{a_1} t_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} t_n \downarrow$
- $\text{tr } t \subseteq L^\star$ the set of all completed traces at t

$$\text{tr}: X \rightarrow \mathcal{P}(L^\star)$$

Trace Equivalence

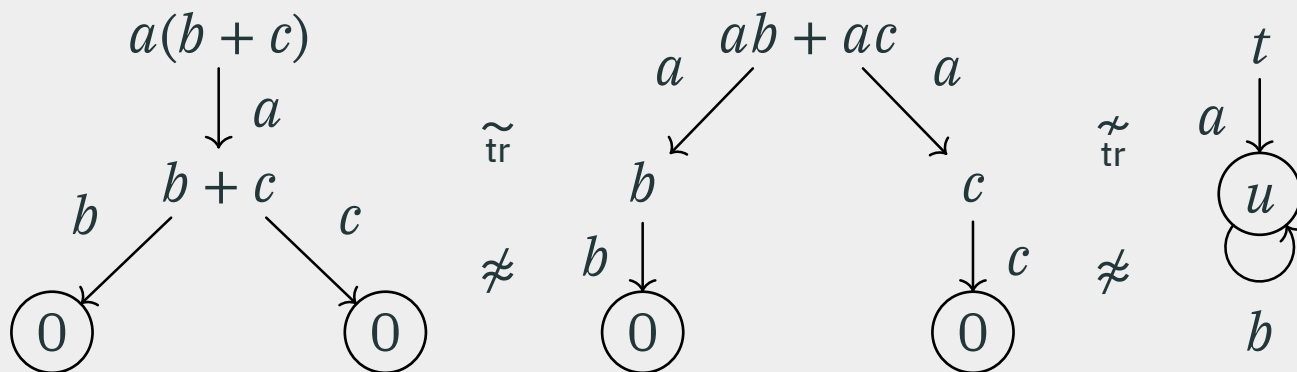
$$\text{tr } t = \left\{ a_1 \cdots a_n \in L^\star \mid t \xrightarrow{a_1} t_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} t_n \downarrow \right\}$$

Definition (Trace equivalence)

$$t \sim_{\text{tr}} u \Leftrightarrow \text{tr } t = \text{tr } u$$

Example

$\text{tr } a(b + c) = \{ab, ac\}$, $\text{tr } (ab + ac) = \{ab, ac\}$, $\text{tr } t = \{a, ab, abb, abbb, \dots\}$



Coalgebraic Trace Semantics

Trace semantics at a higher level of abstraction?

Definition (Kleisli category $\mathbf{Kl}(T)$)

For T a monad on a category $\mathbb{C}$.

objects those of $\mathbb{C}$

morphisms $m : X \rightarrowtail Y \in \mathbf{Kl}(T) \Leftrightarrow m : X \rightarrow TY \in \mathbb{C}$

composition, identity using multiplication and unit of the monad

Idea 💡

1. see the coalgebra $X \rightarrow TBX$ as a coalgebra $X \rightarrowtail BX$ in $\mathbf{Kl}(T)$
2. find the final B -coalgebra in $\mathbf{Kl}(T)$ (easy)
3. compute tr by universal property

Coalgebraic Trace Semantics

Definition (Setting for Coalgebraic Trace Semantics)

A category $\mathbb{C}$, a “behaviour” functor $B: \mathbb{C} \rightarrow \mathbb{C}$ and an “effect” monad $T: \mathbb{C} \rightarrow \mathbb{C}$, such that

1. B has a **Kleisli extension** $\bar{B}: \mathbf{Kl}(T) \rightarrow \mathbf{Kl}(T)$ (on objects $\bar{B}X = BX$),
2. B has an **initial algebra** $(\mu B, \beta)$,
3. $\mathbf{Kl}(T)$ is **locally ordered**, with nice properties:
left strictly $\mathbf{DCPO}_\perp$ -enriched and $\bar{B}$ is locally monotone.

Proposition

In this setting, $\bar{B}: \mathbf{Kl}(T) \rightarrow \mathbf{Kl}(T)$ has a final coalgebra $\mu B \xrightarrow{\beta^{-1}} \bar{B}(\mu B)$:

$$\mu \bar{B} = \mu B = v \bar{B}$$

If $c: X \rightarrow \bar{B}X$, by universal property $\text{tr}_c: X \rightarrow \mu B$.

Example (LTS)

$\mathbb{C} = \mathbf{Set}$, $B = L \times (-) + 1$, $T = \mathcal{P}$,

1. Kleisli extension $\overline{B}: \mathbf{Kl}(\mathcal{P}) \rightarrow \mathbf{Kl}(\mathcal{P})$
2. initial algebra of B given by $\mu B = L^\star = \{a_1 \cdots a_n \mid a_i \in L\}$
3. $\mathbf{Kl}(\mathcal{P})$ is ordered by the subset order: for $f, g: X \rightarrow \mathcal{P}Y$,

$$f \leq g \Leftrightarrow \forall x \in X, f(x) \subseteq g(x)$$

Then $L^\star$ is the final $\overline{B}$ -coalgebra, we get

$$\text{tr}: X \rightarrow L^\star \quad \text{i.e. } \text{tr}: X \rightarrow \mathcal{P}(L^\star)$$

$$\text{tr } t = \left\{ a_1 \cdots a_n \in L^\star \mid t \xrightarrow{a_1} t_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} t_n \downarrow \right\}$$

Abstract GSOS

GSOS – syntactically

- a **framework** for specifying semantics of systems by reduction rules $\rightsquigarrow$ rule format

*with explicit termination

GSOS – syntactically

- a **framework** for specifying semantics of systems by reduction rules $\rightsquigarrow$ rule format
- **syntax** and **behaviour**: LTSs
- **GSOS*** rules (inference rules)

$$\frac{x_{i_j} \xrightarrow{a_j} y_j \quad (j \in J) \quad x_k \downarrow \quad (k \in K)}{f(x_1, \dots, x_n) \xrightarrow{c} u}$$

$$\frac{x_{i_j} \xrightarrow{a_j} y_j \quad (j \in J) \quad x_k \downarrow \quad (k \in K)}{f(x_1, \dots, x_n) \downarrow}$$

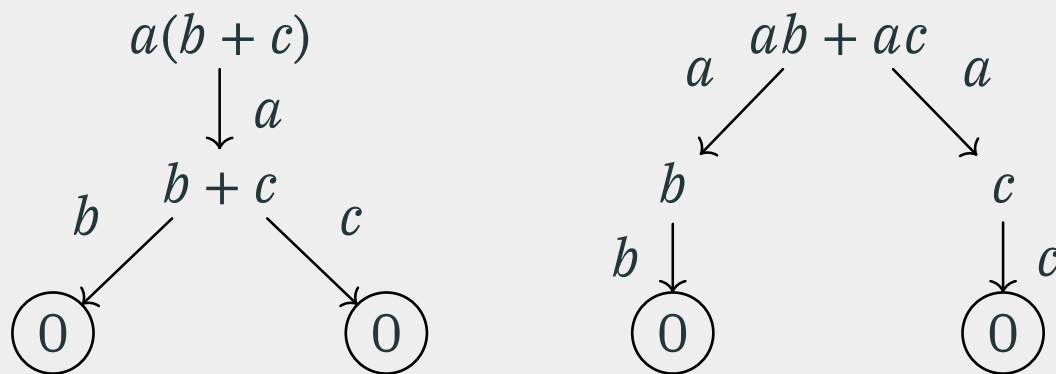
- ▶ x_i, y_j distinct **variables**
 - ▶ u term with variables x_i, y_j
 - ▶ inspect **one layer of syntax** and **“shallow” behaviour of variables**
- set of GSOS rules $\Rightarrow$ inductively derive the behaviour $\cdot \xrightarrow{\cdot}$ and $\cdot \downarrow$ of terms

*with explicit termination

Example

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

$$\frac{}{0 \downarrow} \quad \frac{}{a.x \xrightarrow{a} x} \forall a \quad \frac{x \xrightarrow{a} y}{x + x' \xrightarrow{a} y} \forall a \quad \frac{x' \xrightarrow{a} y'}{x + x' \xrightarrow{a} y'} \forall a \quad \frac{x \downarrow}{x + x' \downarrow} \quad \frac{x' \downarrow}{x + x' \downarrow}$$



Abstract GSOS – categorically

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

Example (LTS)

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

$$(a, t') \in c(t) \Leftrightarrow t \xrightarrow{a} t' \text{ and } * \in c(t) \Leftrightarrow t \downarrow \text{ with } c : X \rightarrow HX = \mathcal{P}(L \times X + 1)$$

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

Abstract GSOS – categorically

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

Example (LTS)

$$\rho_X : 1 + L \times (X \times \mathcal{P}(L \times X + 1)) + (X \times \mathcal{P}(L \times X + 1))^2 \rightarrow \mathcal{P}(L \times \Sigma^* X + 1)$$

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

$$\frac{x_{i_j} \xrightarrow{a_j} y_j \quad (j \in J) \quad x_k \downarrow \quad (k \in K)}{f(x_1, \dots, x_n) \xrightarrow{c} u / \downarrow}$$

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

Abstract GSOS – categorically

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

Example (LTS)

$$\rho_X : 1 + L \times (X \times \mathcal{P}(L \times X + 1)) + (X \times \mathcal{P}(L \times X + 1))^2 \rightarrow \mathcal{P}(L \times \Sigma^* X + 1)$$

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

$$\rho(a, (x, \xi)) = \{(a, x)\}$$

$$\frac{x_{i_j} \xrightarrow{a_j} y_j \quad (j \in J) \quad x_k \downarrow \quad (k \in K)}{f(x_1, \dots, x_n) \xrightarrow{c} u / \downarrow}$$

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

Abstract GSOS – categorically

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

Example (LTS)

$$\rho_X : 1 + L \times (X \times \mathcal{P}(L \times X + 1)) + (X \times \mathcal{P}(L \times X + 1))^2 \rightarrow \mathcal{P}(L \times \Sigma^* X + 1)$$

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

$$\rho(a, (x, \xi)) = \{(a, x)\}$$

$$\rho(((x, \xi), (x', \xi')) = \{(a, y) \in \xi\} \cup$$

$$\frac{x \xrightarrow{a} y}{x + x' \xrightarrow{a} y} \forall a$$

Abstract GSOS – categorically

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

Example (LTS)

$$\rho_X : 1 + L \times (X \times \mathcal{P}(L \times X + 1)) + (X \times \mathcal{P}(L \times X + 1))^2 \rightarrow \mathcal{P}(L \times \Sigma^* X + 1)$$

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

$$\rho(a, (x, \xi)) = \{(a, x)\}$$

$$\rho(((x, \xi), (x', \xi')) = \{(a, y) \in \xi\} \cup \{(a, y') \in \xi'\} \cup$$

$$\frac{x \xrightarrow{a} y}{x + x' \xrightarrow{a} y} \forall a$$

$$\frac{x' \xrightarrow{a} y'}{x + x' \xrightarrow{a} y'} \forall a$$

Abstract GSOS – categorically

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

Example (LTS)

$$\rho_X : 1 + L \times (X \times \mathcal{P}(L \times X + 1)) + (X \times \mathcal{P}(L \times X + 1))^2 \rightarrow \mathcal{P}(L \times \Sigma^* X + 1)$$

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

$$\rho(a, (x, \xi)) = \{(a, x)\}$$

$$\rho(((x, \xi), (x', \xi')) = \{(a, y) \in \xi\} \cup \{(a, y') \in \xi'\} \cup \{*\} \mid \text{if } * \in \xi \cup \xi'\}$$

$$\frac{x \xrightarrow{a} y}{x + x' \xrightarrow{a} y} \forall a$$

$$\frac{x' \xrightarrow{a} y'}{x + x' \xrightarrow{a} y'} \forall a$$

$$\frac{x \downarrow}{x + x' \downarrow}$$

Abstract GSOS – categorically

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

Example (LTS)

$$\rho_X : 1 + L \times (X \times \mathcal{P}(L \times X + 1)) + (X \times \mathcal{P}(L \times X + 1))^2 \rightarrow \mathcal{P}(L \times \Sigma^* X + 1)$$

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

$$\rho(a, (x, \xi)) = \{(a, x)\}$$

$$\begin{aligned} \rho(((x, \xi), (x', \xi')))) &= \{(a, y) \in \xi\} \cup \{(a, y') \in \xi'\} \cup \{*\mid \text{if } * \in \xi\} \cup \{*\mid \text{if } * \in \xi'\} \\ &= \xi \cup \xi' \end{aligned}$$

$$\frac{x \xrightarrow{a} y}{x + x' \xrightarrow{a} y} \forall a$$

$$\frac{x' \xrightarrow{a} y'}{x + x' \xrightarrow{a} y'} \forall a$$

$$\frac{x \downarrow}{x + x' \downarrow}$$

$$\frac{x' \downarrow}{x + x' \downarrow}$$

Theorem (Turi and Plotkin, 1997)

abstract GSOS $\Rightarrow$ bisimilarity is a congruence

Contribution New proof with weaker hypotheses: do not require the GSOS law ρ to be a full natural transformation, only **natural w.r.t. some specific morphisms!**

Abstract GSOS – categorically

Theorem (Turi and Plotkin, 1997)

abstract GSOS $\Rightarrow$ bisimilarity is a congruence

Contribution New proof with weaker hypotheses: do not require the GSOS law ρ to be a full natural transformation, only **natural w.r.t. some specific morphisms!**

Idea 💡 Notice, if $\mathbb{C} = \mathbf{K1}(T)$ and $H = \overline{B}$, bisimilarity is trace equivalence

$\rightsquigarrow$ Just run the theorem in $\mathbf{K1}(T)$!

$\rightsquigarrow$ Looking for a format ensuring that we get a GSOS law natural w.r.t. the appropriate morphisms in $\mathbf{K1}(T)$

De Simone Laws

De Simone Rule Format – syntactically

Definition (De Simone Rule Format^{*} – de Simone, 1985)

Subset of GSOS^{*} such that

1. each x_k appears in at most one of:

- the premises, or
- the conclusion term u

2. each variable appears at most once in u

$$\frac{x_i \xrightarrow{a_i} y_i \quad (i \in I) \quad x_j \downarrow \quad (j \in J)}{f(x_1, \dots, x_n) \xrightarrow{c} u}$$

& idem for $f(x_1, \dots, x_n) \downarrow$

Theorem (Bloom, 1994)

De Simone $\Rightarrow$ trace equivalence is a congruence

$\rightsquigarrow$ abstract the De Simone format

^{*}with explicit termination

De Simone Law – categorically

Key hypotheses setting for coalgebraic trace semantics & $T(X + Y) \cong TX \times TY$

Definition (De Simone Law)

Natural transformation

$$\rho_X : \Sigma(X + BX) \rightarrow TB\Sigma^\star X$$

Remark

It encodes De Simone rules!

$$\frac{x_i \xrightarrow{a_i} y_i \quad (i \in I) \quad x_j \downarrow \quad (j \in J)}{f(x_1, \dots, x_n) \xrightarrow{c} u}$$

Proposition

Every De Simone law induces a pre-GSOS law (**not necessary natural**) in $\mathbf{Kl}(T)$:

$$\bar{\rho}_X : \bar{\Sigma}(X \times \bar{B}X) \rightarrow \bar{B}\bar{\Sigma}^\star X$$

Definition (Affine part, affine morphism)

Affine part of T = a submonad $i : T_+ \rightarrow T$, “the positive part” of T

A morphism $p : X \rightarrow TY$ is **affine** if it factors through i :

$$p : X \xrightarrow{p_+} T_+ Y \xrightarrow{i_Y} TY.$$

Example (LTS)

For $T = \mathcal{P}$, $T_+ = \mathcal{P}_+$ non empty powerset, $i_X : \mathcal{P}_+ X \hookrightarrow \mathcal{P}X$

$$p : X \rightarrow \mathcal{P}Y \text{ is affine} \Leftrightarrow \forall x \in X, p(x) \neq \emptyset$$

Compositionality for Coalgebraic Trace Semantics

Theorem

For every De Simone law ρ , if

- $\bar{\rho}$ is natural w.r.t. $i_X : T_+X \rightarrow X$
 - $\text{tr} : \mu\Sigma \rightarrow T(\mu B)$ is affine
- then, trace equivalence is a congruence.

Proof sketch: Apply the weakened Turi and Plotkin's theorem to $\bar{\rho}$ (in $\mathbf{K1}(T)$), where the two conditions above amount to the required naturality. $\square$

- | | | |
|---------------------------------|--------------------|--|
| De Simone ρ | $\rightsquigarrow$ | 1. each variable appears in at most one of the premises or u |
| $\bar{\rho}$ natural w.r.t. i | $\rightsquigarrow$ | 2. each variable appears at most once in u |
| tr affine | $\rightsquigarrow$ | global property: every state has a terminating path |

$$\frac{x_i \xrightarrow{a_i} y_i \quad (i \in I) \quad x_j \downarrow \quad (j \in J)}{f(x_1, \dots, x_n) \xrightarrow{c} u}$$

Probabilistic De Simone

- replace non-determinism $T = \mathcal{P}$ with positive real **weights** $T = \mathcal{W} = [0, \infty]^{(-)}$
- we have $T_+X = \mathcal{D}X = \{\varphi \in \mathcal{W}X \mid \sum_x \varphi(x) = 1\}$ the **probability distribution monad**

$p: X \rightarrow \mathcal{W}Y$ is affine $\Leftrightarrow \forall x, p(x)$ is a probability distribution

By unfolding our abstract setting, we get

Definition (Probabilistic De Simone Rule Format)

$$\frac{x_i \xrightarrow{a_i} y_i \quad (i \in I) \quad x_j \downarrow \quad (j \in J)}{f(x_1, \dots, x_n) \xrightarrow{c/r} u} \qquad \frac{x_i \xrightarrow{a_i} y_i \quad (i \in I) \quad x_j \downarrow \quad (j \in J)}{f(x_1, \dots, x_n) \downarrow^r}$$

with $r \in [0, \infty]$, such that

- the operational model $\gamma: \mu\Sigma \rightarrow \mathcal{W}(\mu\Sigma)$ is probabilistic (i.e. an affine morphism)
- tr is affine, or equivalently the system is **almost surely terminating**

A **compositional** theory of **coalgebraic trace semantics** at the generality of **abstract GSOS**

⇒ “vanilla” abstract GSOS is more powerful than it seems!

Accounts for the **De Simone rule format** for LTSs and can be used to derive a **novel De Simone-type format** for **probabilistic LTSs**.

Future work

- overcome the “almost surely terminating”: e.g. with infinite trace semantics?
- computational effects beyond probabilities, e.g. states or higher-order behaviour
- trace distance in lieu of trace equivalence

A **compositional** theory of **coalgebraic trace semantics** at the generality of **abstract GSOS**

⇒ “vanilla” abstract GSOS is more powerful than it seems!

Accounts for the **De Simone rule format** for LTSs and can be used to derive a **novel De Simone-type format** for **probabilistic LTSs**.

Future work

- overcome the “almost surely terminating”: e.g. with infinite trace semantics?
- computational effects beyond probabilities, e.g. states or higher-order behaviour
- trace distance in lieu of trace equivalence