Familial monads and structural operational semantics

Tom Hirschowitz

Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LAMA
Chambéry, France

Journées inaugurales Logique, Homotopie, Catégories
October 18, 2018
Structural operational semantics

- Describe dynamics of programming languages, syntactically.
  - Terms from algebraic signature.
  - Dynamics as a (labelled) transition system.
  - Basic idea: describe behaviour of each operation.

\[
\begin{array}{c}
\vdots \\
x_i \xrightarrow{a_i} y_i \\
\vdots \\
f(x_1, \ldots, x_n) \xrightarrow{a} M(y_1, \ldots, y_n)
\end{array}
\]

Structural: behaviour of system determined by its components.

- Disturbing operation: bisimilarity in \(\pi\) not a congruence!

\[
\text{Structural } \nleftrightarrow \text{ compositional.}
\]
De Simone (1985) : rule **format**.

- Algebraic signature + transition system specification 
  \( \leadsto \) transition system.

- Specification complies with format \( \implies \) transition system behaves well.

- E.g.,
  - (weak) bisimilarity is a congruence,
  - conservative extension,
  - bisimulation up to \( X \) is sound.
A wealth of formats

Since then, lots of different formats, combining:

- negative premises,
- predicates,
- look ahead,
- terms as labels,
- variable binding,...
Functorial operational semantics

- Attempt to tame the diversity of formats.
- Appealing simplicity:
  - terms = monad $T$,
  - labels = comonad $L$,
  - rules = distributive law $TL \rightarrow LT$.

- But not so widely adopted.
- Possible reasons:
  - too abstract,
  - not expressive enough (e.g., no negative premises afaik),
  - does not scale well to variable binding.

- Simplifying attempt by Staton (2008):
  - SOS = monad on labeled relations.
  - Better treatment of variable binding.
  - But not better adopted.
Proposal

Two distinct goals:

1. Find the right language for describing
   - what goes on in proofs of congruence of bisimilarity, etc,...,
   - under which hypotheses.

2. Generate instances satisfying the hypotheses.

Here: focus on (1).
Abstract over the following.

- **Bisimulation**: by lifting (cf. presheaf models), in a “category of transition systems”, \( \mathcal{C} \).
- **SOS specifications**: monad \( \mathcal{T} \) on \( \mathcal{C} \). Morally: saturation by the given rules.
- **Model of a SOS specification**: \( \mathcal{T} \)-algebra.
- **Congruence proof** \( \leftrightarrow \) **familiality** of \( \mathcal{T} \),...
My first transition category

Categories that look like transition systems and simulations.

Baby example

(Directed, multi-)graphs, $\mathbf{Gph}$.

- Untyped, one label.
- Presheaves over $s, t: [0] \Rightarrow [1]$.

Definition (Functional bisimulation)

$$
\begin{align*}
[0] & \xrightarrow{v} X \\
[1] & \xrightarrow{e} Y
\end{align*}
$$

\text{i.e.}

$$
\begin{align*}
v & \xrightarrow{f} f(v) \\
k \cdot t & \xrightarrow{f} e \cdot t
\end{align*}
$$
A transition category with basic labels

- Let $A$ be the considered set of labels.
- Presheaves over $\Omega_A$:
  \[
  \begin{array}{ccc}
  \vdots & [a] & \vdots \\
  s & \overset{t}{\leftrightarrow} & \\
  [0] & \end{array}
  \quad (a \in A)
  \]
  
- Any $X \in \widehat{\Omega}_A$ has
  - a set of vertices $X[0]$,
  - a set of $a$-transitions $X[a]$ for all $a \in A$, each with its source and target.
A transition category with basic labels

Definition (Functional bisimulation)

\[
\begin{align*}
[0] & \xrightarrow{v} X \\
\downarrow s & \quad & \quad \quad \quad \downarrow f \\
[a] & \xrightarrow{e} Y \\
\end{align*}
\quad \text{i.e.}
\quad
\begin{align*}
\Downarrow k : a \\
\Downarrow e : t
\end{align*}
\quad
\begin{align*}
\Downarrow f \\
\Downarrow e : a \\
k \cdot t \quad \rightarrow \, \rightarrow \quad e \cdot t
\end{align*}
\]
**Transition categories**

**Definition**

Category with distinguished cospans

\[
P \xrightarrow{s} L \xleftarrow{t} Q
\]

+ finite completeness, cocompleteness, well-poweredness, images, and tininess of all \( P \in \mathbf{P} \).

Let \( \mathbf{T}_s \) denote the set of all such \( s: P \to L \).

**Definition (Functional bisimulation)**

\[
\begin{array}{ccc}
P & \xrightarrow{v} & X \\
\downarrow{s} & & \downarrow{f} \\
L & \xrightarrow{e} & Y \\
\end{array}
\]

i.e. \( f \in \mathbf{T}_s^\varnothing \).
# SOS specifications as monads

Idea (Staton) : view SOS rules as endofunctors.

**Example, on $\Omega^A$**

<table>
<thead>
<tr>
<th>SOS specification $S \rightsquigarrow$ monad $\mathcal{T}_S$ :</th>
</tr>
</thead>
<tbody>
<tr>
<td>- $\mathcal{T}_S(X)[0]$ : terms with constants in $X$,</td>
</tr>
<tr>
<td>- $\mathcal{T}_S(X)[a]$ : derivations with transition axioms in $X$,</td>
</tr>
<tr>
<td>- multiplication $\mathcal{T}_S^2(X)[a] \rightarrow \mathcal{T}_S(X)[a]$ : plugging derivations.</td>
</tr>
</tbody>
</table>

Example CCS.

$\rightsquigarrow$ basic abstract framework : transition category with a monad on it.
Congruence of bisimilarity

Will follow from:

**Theorem**

*If* $f : R \rightarrow X$ *is a functional bisimulation and* $X$ *is a* $\mathcal{T}$-*algebra, then so is*

$$\mathcal{T}(R) \xrightarrow{\mathcal{T}(f)} \mathcal{T}(X) \xrightarrow{a} X,$$

*up to hypotheses.*
Hypothesis 1: Compositionality

Generally a vague concept (thanks for asking!).

**Definition**

An algebra \( a: \mathcal{T}(X) \to X \) is *compositional* iff it is a functional bisimulation.

\[
\begin{array}{cccc}
P & \overset{v}{\longrightarrow} & \mathcal{T}(X) \\
\downarrow s & & \downarrow a \\
L & \overset{e}{\longrightarrow} & X \\
\end{array}
\]

Morally: any transition \( C[x_1, \ldots, x_n] \xrightarrow{\alpha} x' \) decomposes as

\[
\begin{array}{cccc}
\ldots & x_i & \overset{\alpha_i}{\longrightarrow} & y_i & \ldots \\
\hline
\end{array}
\]

\[
C[x_1, \ldots, x_n] \xrightarrow{\alpha} E[y_1, \ldots, y_n]
\]
## Congruence of bisimilarity

Will follow from:

### Theorem

If $f : R \to X$ is a functional bisimulation and $X$ is a compositional $\mathcal{T}$-algebra, then so is

$$\mathcal{T}(R) \xrightarrow{\mathcal{T}(f)} \mathcal{T}(X) \xrightarrow{a} X,$$

up to hypotheses.

It now suffices to prove that $\mathcal{T}(f)$ is a functional bisimulation.
Standard proof method

- Consider any \( C[r_1, ..., r_n] \in \mathcal{T}(R) \) and let \( x_i = f(r_i) \).

- Assume \( C[x_1, ..., x_n] \xrightarrow{L} E[x'_1, ..., x'_m] \) (say \( m = n \) to simplify!).

- But \( f \) is a bisimulation, so find

\[
C[r_1, ..., r_n] \xrightarrow{\mathcal{T}(f)} C[x_1, ..., x_n] \xrightarrow{E[e_1, ..., e_n]:L} D[x'_1, ..., x'_m].
\]

- That’s the intuition. In practice:
  - transition contexts \( E \) are not first-class citizens,
  - \( \sim \) induction on \( C \).
Standard proof method

- Consider any $C[r_1, \ldots, r_n] \in \mathcal{T}(R)$ and let $x_i = f(r_i)$.
- Assume $C[x_1, \ldots, x_n] \xrightarrow{L} E[x'_1, \ldots, x'_m]$ (say $m = n$ to simplify!).
- But $f$ is a bisimulation, so find

$$C[r_1, \ldots, r_n] \xrightarrow{\mathcal{T}(f)} C[x_1, \ldots, x_n]$$

$$E[k_1, \ldots, k_n]:L \downarrow\quad \downarrow E[e_1, \ldots, e_n]:L$$

$$D[r'_1, \ldots, r'_n] \xrightarrow{\mathcal{T}(f)} D[x'_1, \ldots, x'_m].$$

- That’s the intuition. In practice:
  - transition contexts $E$ are not first-class citizens,
  - $\rightsquigarrow$ induction on $C$. 
In the abstract framework

\[
P \xrightarrow{r} \mathcal{T}(R) \\
\downarrow s \quad \mathcal{T}(f) \\
\mathcal{T}(X) \xrightarrow{e} \mathcal{T}(X).
\]
In the abstract framework

Familiality!

Any $U \to \mathcal{T}(X)$ factors as $U \xrightarrow{\xi} \mathcal{T}(Y) \xrightarrow{\mathcal{T}(f)} \mathcal{T}(X)$ with $\xi$ generic:

$$
U \xrightarrow{\chi} \mathcal{T}(Z) \xrightarrow{\mathcal{T}(l)} \mathcal{T}(Y) \xrightarrow{\mathcal{T}(f)} \mathcal{T}(X)
$$

(meaning $g \circ l = f$)
In the abstract framework

\[ P \xrightarrow{s} L \]

\[ \mathcal{T}(\sum_i P_i) \xrightarrow{r} \mathcal{T}(R) \]

\[ \mathcal{T}(\sum_i L_i) \xrightarrow{e} \mathcal{T}(X). \]

\[ \text{\textbf{\text{T}_s-familiality}} \]

\[ P \xrightarrow{s} L \]

\[ \xi \downarrow \]

\[ \mathcal{T}(Y) \xrightarrow{\mathcal{T}(f)} \mathcal{T}(Z) \]

If \( \xi \) and \( \zeta \) are generic, then \( f \in \Box(\text{T}_s^\Box). \)
Standard definition of bisimulation up to context

$R$ progresses to $\mathcal{C}(R)$, where

$$\mathcal{C}(R) := \{(C[P_1, \ldots, P_n], C[Q_1, \ldots, Q_n]) \mid P_i R Q_i\}.$$
Progression

Standard definition of bisimulation up to context

$R$ progresses to $\mathcal{C}(R)$, where

$$\mathcal{C}(R) := \{ (C[P_1, \ldots, P_n], C[Q_1, \ldots, Q_n]) \mid P_i \mathrel{R} Q_i \}.$$ 

Generalises to $R$ progresses to $R'$. 

$$
\begin{array}{ccc}
x & \overset{R}{\longrightarrow} & y \\
\downarrow & & \downarrow \\
x' & \overset{R'}{\longrightarrow} & \exists y'
\end{array}
$$
Progression in the abstract framework

Definition

- Relations $R, R' \in X \times Y$ in transition category.
- $R \sim R'$ iff

$$
P \xrightarrow{c} R \xrightarrow{r \circ s} X \quad \text{and symmetrically for } Y.
$$

Example: bisimulation up to context

$R \sim \mathcal{T}(R)$. 

Tom Hirschowitz
But wait...

**Question**

Does $R \sim R$ iff $R$ is a bisimulation?

- Not quite, but artefact of formalism.
- Reason: in $R \sim R$, $R \leftrightarrow X \times Y$ may have no transition.
- Good news, we can add them:

**Proposition**

Under mild hypotheses, factors as

$$R \rightarrow \overline{R} \rightarrow X \times Y$$

with $\overline{R}$ a bisimulation.
# Soundness of bisimulation up to context

## Theorem

*Under hypotheses,* $R \leadsto \mathcal{T}(R)$ *entails* $\mathcal{T}(R) \leadsto \mathcal{T}(R)$.

## Corollary

*Any bisimulation up to context embeds into some bisimulation.*
Conclusion

Summary:

- **SOS specification** = monad on a transition category.
- Hypotheses \( \Rightarrow \)
  - congruence of bisimilarity,
  - soundness of bisimulation up to context.

Perspectives:

- Existing formats \( \sim \) instances?
- More general format along the lines of free monads.
- Other up to techniques.
- Related questions, e.g., process equations, environmental bisimulation.
- Broader scope: analytic monads, to accommodate structural congruence.
- Go quantitative?