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

- Notes by Plotkin (1981) : method rather than theory, by example.
- Describe dynamics of programming languages, syntactically.
  - Terms from algebraic signature.
  - Dynamics as a (labelled) transition system.
  - Basic idea : describe behaviour of each operation.

$$\frac{\ldots \qquad x_i \xrightarrow{a_i} y_i \qquad \ldots}{f(x_1, \ldots, x_n) \xrightarrow{a} M(y_1, \ldots, y_n)}$$

Structural : behaviour of system determined by its components.

- Disturbing operation : bisimilarity in π not a congruence !

Structural $\not\Longrightarrow$ compositional.

Formats

De Simone (1985) : rule format.

- Algebraic signature + transition system specification

  $\rightsquigarrow$ transition system.
- Specification complies with format $\Longrightarrow$ 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 $TL \to LT$.
- 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).

Plan

Abstract over the following.

- Bisimulation : by lifting (cf. presheaf models), in a "category of transition systems", $\mathscr{C}$.
- SOS specifications : monad $\mathscr{T}$ on $\mathscr{C}$.
  Morally : saturation by the given rules.
- Model of a SOS specification : $\mathscr{T}$-algebra.
- Congruence proof $\backsim$ familiality of $\mathscr{T}$,...

My first transition category

Categories that look like transition systems and simulations.

Baby example

(Directed, multi-)graphs, **Gph**.

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

Definition (Functional bisimulation)

$$\begin{tikzcd} {[0]} \arrow[r, "v"] \arrow[d, "s"'] & X \arrow[d, "f"] \\ {[1]} \arrow[r, "e"'] \arrow[ur, dashed, "k"] & Y \end{tikzcd} \qquad \text{i.e.} \qquad \begin{tikzcd} v \arrow[r, mapsto, "f"] \arrow[d, dashed, "k"'] & f(v) \arrow[d, "e"] \\ k \cdot t \arrow[r, dashed, mapsto, "f"'] & e \cdot t \end{tikzcd}$$

A transition category with basic labels

- Let A be the considered set of labels.
- Presheaves over Ω_A :

$$
\begin{tikzcd}
\dots & {[a]} & \dots & (a \in A) \\
& {[0]} \arrow[u, bend left, "s"] \arrow[u, bend right, "t"'] & &
\end{tikzcd}
$$

- 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)

$$\begin{tikzcd} {[0]} \arrow[r, "v"] \arrow[d, "s"'] & X \arrow[d, "f"] \\ {[a]} \arrow[r, "e"'] \arrow[ur, dashed, "k"] & Y \end{tikzcd} \quad \text{i.e.} \quad \begin{tikzcd} v \arrow[r, mapsto, "f"] \arrow[d, dashed, "k:a"'] & f(v) \arrow[d, "e:a"] \\ k \cdot t \arrow[r, mapsto, dashed, "f"'] & e \cdot t \end{tikzcd}$$

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 $s \colon P \to L$.

Definition (Functional bisimulation)

$$\begin{tikzcd} P \arrow[r, "v"] \arrow[d, "s"'] & X \arrow[d, "f"] \\ L \arrow[r, "e"'] \arrow[ur, dashed, "k"] & Y \end{tikzcd}$$

i.e. $f \in \mathbf{T}_s^{\boxslash}$.

SOS specifications as monads

Idea (Staton) : view SOS rules as endofunctors.

Example, on $\widehat{\Omega_A}$

SOS specification $S \rightsquigarrow$ monad $\mathscr{T}_S$:

- $\mathscr{T}_S(X)[0]$: terms with constants in X,
- $\mathscr{T}_S(X)[a]$: derivations with transition axioms in X,
- multiplication $\mathscr{T}_S^2(X)[a] \to \mathscr{T}_S(X)[a]$: plugging derivations.

Example CCS.

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

Congruence of bisimilarity

Will follow from :

Theorem

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

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

up to hypotheses.

Hypothesis 1 : compositionality

Generally a vague concept (thanks for asking !).

Definition

An algebra $a \colon \mathscr{T}(X) \to X$ is *compositional* iff it is a functional bisimulation.

$$\begin{tikzcd} P \arrow[r, "v"] \arrow[d, "s"'] & \mathscr{T}(X) \arrow[d, "a"] \\ L \arrow[r, "e"'] \arrow[ur, dashed, "k"] & X \end{tikzcd}$$

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

$$\frac{\ldots \qquad x_i \xrightarrow{\alpha_i} y_i \qquad \ldots}{C[x_1, \ldots, x_n] \xrightarrow{\alpha} E[y_1, \ldots, y_n]}.$$

Congruence of bisimilarity

Will follow from :

Theorem

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

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

up to hypotheses.

It now suffices to prove that $\mathscr{T}(f)$ is a functional bisimulation.

Standard proof method

- Consider any $C[r_1, ..., r_n] \in \mathscr{T}(R)$ 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

$$\begin{tikzcd} C[r_1, ..., r_n] \arrow[r, mapsto, "\mathscr{T}(f)"] & C[x_1, ..., x_n] \arrow[d, "{E[e_1, ..., e_n]:L}"] \\ & D[x'_1, ..., x'_m]. \end{tikzcd}$$

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

Standard proof method

- Consider any $C[r_1, ..., r_n] \in \mathscr{T}(R)$ 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

$$\begin{tikzcd} C[r_1, ..., r_n] \arrow[r, mapsto, "\mathscr{T}(f)"] \arrow[d, dashed, "{E[k_1,...,k_n]:L}"'] & C[x_1, ..., x_n] \arrow[d, "{E[e_1,...,e_n]:L}"] \\ D[r'_1, ..., r'_n] \arrow[r, dashed, mapsto, "\mathscr{T}(f)"'] & D[x'_1, ..., x'_m]. \end{tikzcd}$$

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

In the abstract framework

$$
\begin{tikzcd}
P \arrow[r, "r"] \arrow[d, "s"'] & \mathscr{T}(R) \arrow[d, "\mathscr{T}(f)"] \\
L \arrow[r, "e"'] & \mathscr{T}(X).
\end{tikzcd}
$$

In the abstract framework

$$\begin{tikzcd}
P \arrow[rr, "r"] \arrow[dr, "C"] \arrow[ddd, "s"'] & & \mathscr{T}(R) \arrow[ddd, "\mathscr{T}(f)"] \\
& \mathscr{T}(\sum_i P_i) \arrow[ur, "\mathscr{T}([r_i]_i)"'] \arrow[d, dashed, "\mathscr{T}(\sum_i s_i)"'] & \\
& \mathscr{T}(\sum_i L_i) \arrow[dr, "\mathscr{T}([e_i]_i)"] & \\
L \arrow[ur, "E"] \arrow[rr, "e"'] & & \mathscr{T}(X).
\end{tikzcd}$$

Familiality !

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

$$\begin{tikzcd}
U \arrow[r, "\chi"] \arrow[d, "\xi"'] & \mathscr{T}(Z) \arrow[d, "\mathscr{T}(g)"] \\
\mathscr{T}(Y) \arrow[ur, dashed, "\mathscr{T}(l)"] \arrow[r, "\mathscr{T}(f)"'] & \mathscr{T}(X)
\end{tikzcd}$$

(meaning $g \circ l = f$)

In the abstract framework

$$\begin{tikzcd}
P \arrow[rr, "r"] \arrow[dr, "C"] \arrow[ddd, "s"'] & & \mathscr{T}(R) \arrow[ddd, "\mathscr{T}(f)"] \\
& \mathscr{T}(\sum_i P_i) \arrow[ur, "\mathscr{T}([r_i]_i)"'] \arrow[d, dashed, "\mathscr{T}(\sum_i s_i)"'] & \\
& \mathscr{T}(\sum_i L_i) \arrow[uur, dashed, "\mathscr{T}([k_i]_i)"'] \arrow[dr, "\mathscr{T}([e_i]_i)"] & \\
L \arrow[ur, "E"] \arrow[rr, "e"'] & & \mathscr{T}(X).
\end{tikzcd}$$

$\mathbf{T}_s$-familiality

$$\begin{tikzcd}
P \arrow[r, "s"] \arrow[d, "\xi"'] & L \arrow[d, "\zeta"] \\
\mathscr{T}(Y) \arrow[r, "\mathscr{T}(f)"'] & \mathscr{T}(Z)
\end{tikzcd}$$

If ξ and ζ are generic, then $f \in {}^{\boxslash}(\mathbf{T}_s^{\boxslash})$.

Progression

Standard definition of bisimulation up to context

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

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

$$\begin{tikzcd} x \arrow[d] & R & y \arrow[d, dashed] \\ x' & \mathscr{C}(R) & \exists y' \end{tikzcd}$$

Progression

Standard definition of bisimulation up to context

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

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

$$\begin{tikzcd}
x \arrow[d] & R & y \arrow[d, dashed] \\
x' & R' & \exists y'
\end{tikzcd}$$

Generalises to R progresses to R'.

Progression in the abstract framework

Definition

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

$$
\begin{tikzcd}
P \arrow[dd, "s"'] \arrow[dr, "c"] \arrow[drr, bend left, "r\circ s"] & & \\
 & R \arrow[r, "\pi\circ i"'] \arrow[dd, "i"] & X \arrow[dd, equal] \\
L \arrow[dr, dashed, "u"] \arrow[drr, bend left, "r"] & & \\
 & X \times Y \arrow[r, "\pi"] & X \arrow[dd, equal] \\
Q \arrow[uu, "t"] \arrow[dr, dashed, "d"'] \arrow[drr, bend left, "r\circ t"] & & \\
 & R' \arrow[uu, "i"] \arrow[r, "\pi\circ i"] & X
\end{tikzcd}
$$

and symmetrically for Y.

Example : bisimulation up to context

$R \rightsquigarrow \mathscr{T}(R)$.

But wait...

Question

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

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

Proposition

Under mild hypotheses, factors as

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

with $\overline{R}$ a bisimulation.

Soundness of bisimulation up to context

Theorem

Under hypotheses, $R \rightsquigarrow \mathscr{T}(R)$ *entails* $\mathscr{T}(R) \rightsquigarrow \mathscr{T}(R)$.

Corollary

Any bisimulation up to context embeds into some bisimulation.

Conclusion

Summary :

- SOS specification = monad on a transition category.
- Hypotheses $\Longrightarrow$
  - congruence of bisimilarity,
  - soundness of bisimulation up to context.

Perspectives :

- Existing formats $\rightsquigarrow$ 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 accomodate structural congruence.
- Go quantitative ?