Deriving Bisimulation Congruences: 2-categories vs precategories

FOSSACS 2003

Vladimiro Sassone

joint work with: Paweł Sobociński

University of Sussex, MYTHS

Introduction

- Process calculi are often presented as:
  1. Syntax
  2. Structural congruence
  3. Reactions
  4. Labelled transition system(s) (LTS)

≪ ⋘ ⋙ ≫

Introduction

- Process calculi are often presented as:
  1. Syntax
  2. Structural congruence
  3. Reactions
  4. Labelled transition system(s) (LTS)

- This talk is about categorical machinery which allows the derivation of a LTS from reactions.

- Bisimulation on such an LTS is a congruence, provided a general condition is met.

What is a Reduction System?

A reduction system over a signature Σ is a relation $\searrow \subseteq T_\Sigma \times T_\Sigma$, T_Σ is the set of terms Σ.

Reduction systems are often presented parametrically.

Contexts: terms with variables: $\mathscr{C}[x_1, \ldots, x_n]$

Reduction rules: set $\mathscr{R}$ of parametric rewriting rules:
$\mathscr{C}[x_1, \ldots, x_n] \searrow \mathscr{D}[x_1, \ldots, x_n]$.

Evaluation Contexts: chosen set $\mathscr{E}$ of single-variable contexts.

$$\frac{\mathscr{C}[x_1, \ldots, x_n] \searrow \mathscr{D}[x_1, \ldots, x_n] \in \mathscr{R} \qquad \mathscr{E} \text{ evaluation context}}{\mathscr{E}[\mathscr{C}[t_1, \ldots, t_n]] \searrow \mathscr{E}[\mathscr{D}[t_1, \ldots, t_n]]}$$

≪ ⋘ ⋙ ≫

A Reduction System for CCS

Syntax:

$$p ::= \sum_i \alpha_i.p_i \mid p \mid p \mid (\nu a)p \mid A\langle\vec{u}\rangle$$

with $\vec{A}(\vec{x}) = \vec{p}$ a set of parametric, mutually recursive definitions.

Structural Congruence:

summands in $\sum_i$ can be rearranged arbitrarily

$\mid$ is a monoid with $0 \triangleq \sum_\varnothing$ for unit

$(\nu a)p \equiv (\nu b)p\{a := b\}$ $\quad$ (b not in p)

$(\nu a)(p \mid q) \equiv (\nu a)p \mid q$ $\quad$ (a not in q)

$A\langle\vec{u}\rangle \equiv p\{\vec{x} := \vec{u}\}$ $\quad$ (if $A(\vec{x}) = p$ is a def)

Reduction Rules: $(a.p + \sum_i \alpha_i.p_i) \mid (\bar{a}.q + \sum_j \beta_j.q_j) \searrow p \mid q$

Evaluation Contexts: $\mathscr{E} ::= (\_) \mid \mathscr{E} \mid p \mid p \mid \mathscr{E} \mid (\nu a)\mathscr{E}$

Barbed Congruence

Observations:

$p \downarrow_a$ iff $p \equiv (\nu b)(\alpha.p + \sum_i \alpha_i.p_i)$ with $\alpha \in \{a, \bar{a}\},\ a \neq b$

$p \Downarrow_a$ iff $p \searrow^* p'$ and $p' \downarrow_a$

Barbed Congruence

Observations:

$$p \downarrow_a \quad \text{iff} \quad p \equiv (\nu b)(\alpha.p + \textstyle\sum_i \alpha_i.p_i) \quad \text{with } \alpha \in \{a, \bar{a}\},\; a \neq b$$

$$p \Downarrow_a \quad \text{iff} \quad p \searrow^* p' \text{ and } p' \downarrow_a$$

Barbed Bisimulation: is symmetric relation $\mathscr{R}$ which is

- reduction closed: $p \mathscr{R} q$ and $p \searrow p'$ implies $q \searrow^* q'$ with $p' \mathscr{R} q'$;
- barb preserving: $p \mathscr{R} q$ and $p \downarrow_a$ implies $q \Downarrow_a$.

$p \stackrel{\cdot}{\cong} q$ if there exists a barbed bisimulation $\mathscr{R}$ such that $p \mathscr{R} q$.

Barbed Congruence

Observations:

$$p \downarrow_a \quad \text{iff} \quad p \equiv (\nu b)(\alpha.p + \textstyle\sum_i \alpha_i.p_i) \quad \text{with } \alpha \in \{a, \bar{a}\},\ a \neq b$$

$$p \Downarrow_a \quad \text{iff} \quad p \searrow^* p' \text{ and } p' \downarrow_a$$

Barbed Bisimulation: is symmetric relation $\mathscr{R}$ which is

- **reduction closed:** $p \mathrel{\mathscr{R}} q$ and $p \searrow p'$ implies $q \searrow^* q'$ with $p' \mathrel{\mathscr{R}} q'$;
- **barb preserving:** $p \mathrel{\mathscr{R}} q$ and $p \downarrow_a$ implies $q \Downarrow_a$.

$p \stackrel{\cdot}{\cong} q$ if there exists a barbed bisimulation $\mathscr{R}$ such that $p \mathrel{\mathscr{R}} q$.

Barbed Congruence: $p \cong q$ if $\mathscr{C}(p) \stackrel{\cdot}{\cong} \mathscr{C}(q)$ for all contexts $\mathscr{C}$.

≪ ⋘ ⋙ ≫

What is a Labelled Transition System?

- Rather than describing the internal behaviour of a system (reductions) it describe the interactions this is willing to offer to the surrounding enviroment.
These are characterised and described using label transitions, where a transition indicates an activitiy and a label classifies it.
For instance client — *'insert coin'* → client′. Or perhaps,
machine — *'delivers candy'* → machine′.

- This yield a compositional semantics, as e.g.:

$$\frac{\text{client} - \textit{`insert coin'} \rightarrow \text{client}' \quad \text{machine} - \textit{`delivers candy'} \rightarrow \text{machine}'}{\text{client} \mid \text{machine} - \textit{`yum'} \rightarrow \text{client}' \mid \text{machine}'}$$

- Label transition systems admit proof techniques (LTS bisimulation), verification of logic formulas (model-checking), . . .

A Labelled Transition System for CCS

(Prefix)

$$\frac{}{\sum_i \alpha_i.p_i + \alpha.p + \sum_j \alpha_j.p_j \xrightarrow{\alpha} p}$$

(ParL)

$$\frac{p \xrightarrow{\alpha} p}{p \mid q \xrightarrow{\alpha} p' \mid q}$$

(ParR)

$$\frac{q \xrightarrow{\alpha} q'}{p \mid q \xrightarrow{\alpha} p \mid q'}$$

(Sync)

$$\frac{p \xrightarrow{a} p' \qquad q \xrightarrow{\bar{a}} q'}{p \mid q \xrightarrow{\tau} p' \mid q'}$$

(Restr)

$$\frac{p \xrightarrow{\alpha} p'}{(\nu a)p \xrightarrow{\alpha} (\nu a)p'}\ \alpha \notin \{a, \bar{a}\}$$

(Def)

$$\frac{p\{\vec{x} := \vec{u}\} \xrightarrow{\alpha} p'}{A\langle\vec{u}\rangle \xrightarrow{\alpha} p'}\ A\langle\vec{x}\rangle = p$$

LTS Bisimulation

LTS Bisimulation: is symmetric relation $\mathscr{R}$ which is **transition closed:**

- $p \mathrel{\mathscr{R}} q$ and $p \xrightarrow{\alpha} p'$ implies $q \xrightarrow{\tau^* \hat{\alpha} \tau^*} q'$ with $p' \mathrel{\mathscr{R}} q'$;

$p \approx q$ if there exists a LTS bisimulation $\mathscr{R}$ such that $p \mathrel{\mathscr{R}} q$.

Coinduction principle: To prove $p \approx q$ it suffices to present $\mathscr{R}$ with $p \mathrel{\mathscr{R}} q$.

LTS Bisimulation

LTS Bisimulation: is symmetric relation $\mathscr{R}$ which is **transition closed:**

- $p \mathrel{\mathscr{R}} q$ and $p \xrightarrow{\alpha} p'$ implies $q \xrightarrow{\tau^* \hat{\alpha} \tau^*} q'$ with $p' \mathrel{\mathscr{R}} q'$;

$p \approx q$ if there exists a LTS bisimulation $\mathscr{R}$ such that $p \mathrel{\mathscr{R}} q$.

Coinduction principle: To prove $p \approx q$ it suffices to present $\mathscr{R}$ with $p \mathrel{\mathscr{R}} q$.

LTSs Desiderata:

- **Congruence:** $\approx$ is a congruence.

Coinduction Principles for Reductions

The reactions in a process calculus often give the computational intuitions of the calculus (eg. CCS, pi-calculus, ambient-calculus, ...)

How can we give a congruent equivalence using the reaction rules?

Coinduction Principles for Reductions

The reactions in a process calculus often give the computational intuitions of the calculus (eg. CCS, pi-calculus, ambient-calculus, . . .)

How can we give a congruent equivalence using the reaction rules?

- Barbed bisimulation
  - (Milner, Sangiorgi 92)

≪ ⋘ ⋙ ≫

Coinduction Principles for Reductions

The reactions in a process calculus often give the computational intuitions of the calculus (eg. CCS, pi-calculus, ambient-calculus, . . .)

How can we give a congruent equivalence using the reaction rules?

- Barbed bisimulation
  - (Milner, Sangiorgi 92)
- Equating insensitive terms
  - (Honda, Yoshida 95)

Coinduction Principles for Reductions

The reactions in a process calculus often give the computational intuitions of the calculus (eg. CCS, pi-calculus, ambient-calculus, . . .)

How can we give a congruent equivalence using the reaction rules?

- Barbed bisimulation
  - (Milner, Sangiorgi 92)
- Equating insensitive terms
  - (Honda, Yoshida 95)
- Deriving an LTS
  - (Sewell 98, Leifer and Milner 00, this talk)

≪ ⋘ ⋙ ≫

Coinduction Principles for Reductions

The reactions in a process calculus often give the computational intuitions of the calculus (eg. CCS, pi-calculus, ambient-calculus, ...)

How can we give a congruent equivalence using the reaction rules?

- Barbed bisimulation
  - (Milner, Sangiorgi 92)
- Equating insensitive terms
  - (Honda, Yoshida 95)
- Deriving an LTS
  - (Sewell 98, Leifer and Milner 00, this talk)

LTSs Desiderata:

- Operational Correspondence: $p \searrow q$ iff $p \xrightarrow{\tau} q$ (up to $\equiv$)
- Correctness: $p \approx q$ implies $p \cong q$
- Completeness: $p \cong q$ implies $p \approx q$

≪ ⋘ ⋙ ≫

Contexts as Labels

The intuition:

$$a \xrightarrow{\mathscr{C}} b \text{ iff } \mathscr{C}[a] \searrow b$$

≪ ⋘ ⋙ ≫

Contexts as Labels

The intuition:

$$a \xrightarrow{\mathscr{C}} b \text{ iff } \mathscr{C}[a] \searrow b$$

For instance:

$$a \xrightarrow{-|\bar{a}} \mathbf{0} \qquad M \xrightarrow{(\lambda x.-)N} M\{N/x\} \qquad \mathsf{K}M \xrightarrow{-N} M$$

Contexts as Labels

The intuition:

$$a \xrightarrow{\mathscr{C}} b \text{ iff } \mathscr{C}[a] \searrow b$$

For instance:

$$a \xrightarrow{-|\bar{a}} \mathbf{0} \qquad M \xrightarrow{(\lambda x.-)N} M\{N/x\} \qquad \mathbf{K}M \xrightarrow{-N} M$$

Yep, but not quite:

- Too many labels not desirable:
  - Useless combinatorial explosion: $\lambda x.xx \xrightarrow{-MN} MMN$
  - Messes up the bisimulation (too coarse): $l \xrightarrow{\mathscr{D}} \mathscr{D}[r]$ for all rules $l \searrow r$.

≪ ⋘ ⋙ ≫

Contexts as Labels

The intuition:

$$a \xrightarrow{\mathscr{C}} b \text{ iff } \mathscr{C}[a] \searrow b$$

For instance:

$$a \xrightarrow{-|\bar{a}} \mathbf{0} \qquad M \xrightarrow{(\lambda x.-)N} M\{N/x\} \qquad \mathsf{K}M \xrightarrow{-N} M$$

Yep, but not quite:

- Too many labels not desirable:
  - Useless combinatorial explosion: $\lambda x.xx \xrightarrow{-MN} MMN$
  - Messes up the bisimulation (too coarse): $l \xrightarrow{\mathscr{D}} \mathscr{D}[r]$ for all rules $l \searrow r$.

Choose only 'minimal' redex-enabling contexts

- Case analysis of basic situations: Sewell. Abstract approach: Leifer-Milner

$$\boxed{\boxed{\boxed{a} \mid \bar{a}} \mid a}$$

≪ ⋘ ⋙ ≫

Reactive Systems

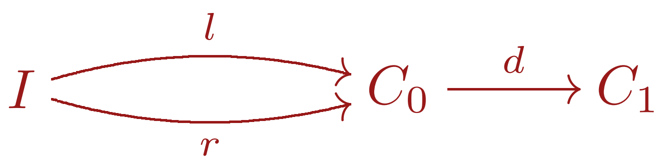
A generalisation of ground term rewriting systems.

- A category $\mathbb{C}$ with distinguished object I.
- A set of reaction rules $\mathcal{R} \subseteq \bigcup_{C \in \mathbb{C}} \mathbb{C}(I, C) \times \mathbb{C}(I, C)$.
- A set $\mathbb{D}$ of arrows of $\mathbb{C}$ called the reactive contexts.

  Assume that $d_0 \,.\, d_1 \in \mathbb{D}$ implies d_0 and $d_1 \in \mathbb{D}$.

The reaction relation is defined as

$$a \longrightarrow b \quad \text{iff} \quad a = d \,.\, l,\ b = d \,.\, r,\ d \in \mathbb{D} \text{ and } \langle l, r \rangle \in \mathcal{R}.$$

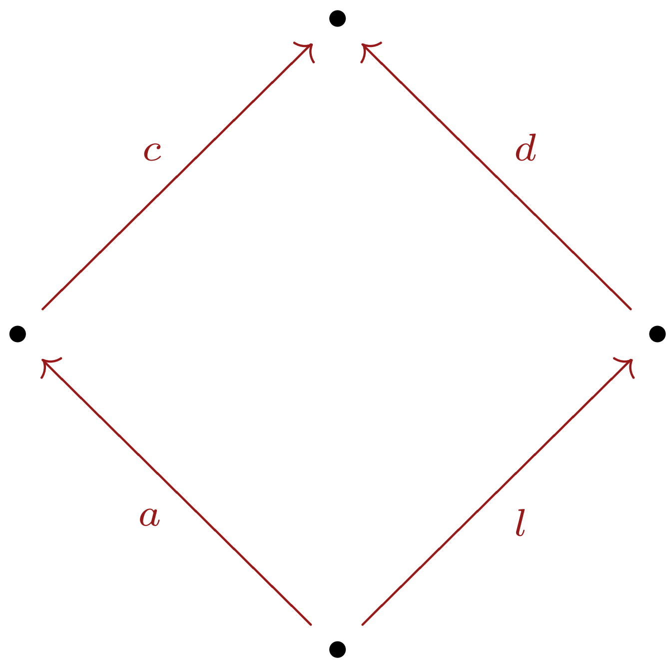


Well-known Reactive Systems

- Term rewriting systems
- Graph rewriting systems
  - via cospans
- Simple process calculi
  - with terms up to structural congruence

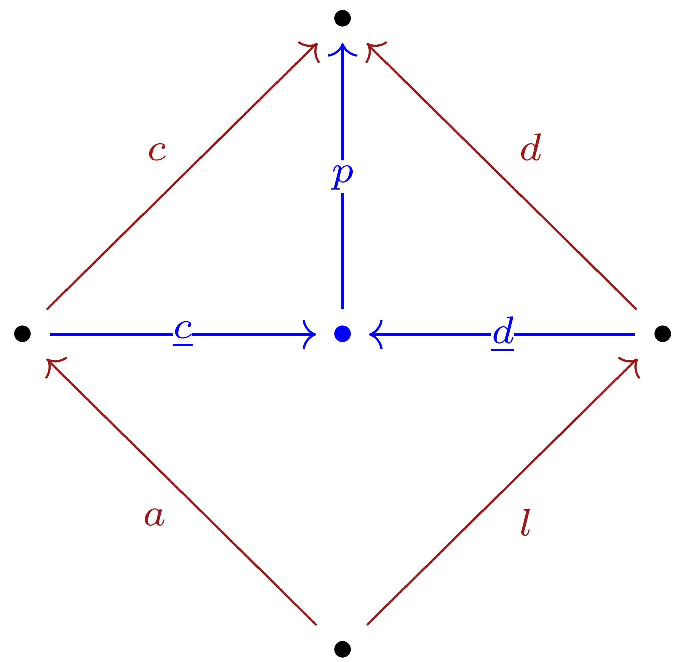
RPOs

Suppose that $\mathbb{C}$ is a category and consider a redex square



RPOs

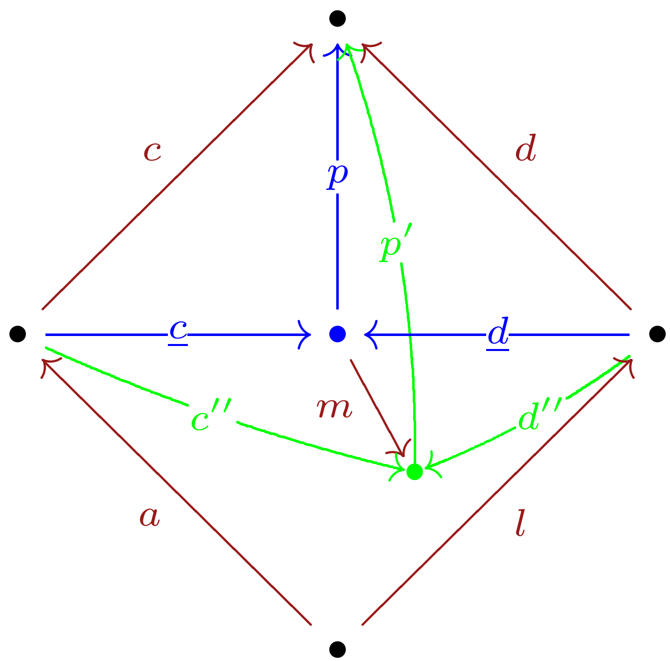
Suppose that $\mathbb{C}$ is a category and consider a redex square



- a relative pushout (RPO) is a tuple $\langle \underline{c}, \underline{d}, p \rangle$ which satisfies the universal property that:

RPOs

Suppose that $\mathbb{C}$ is a category and consider a redex square

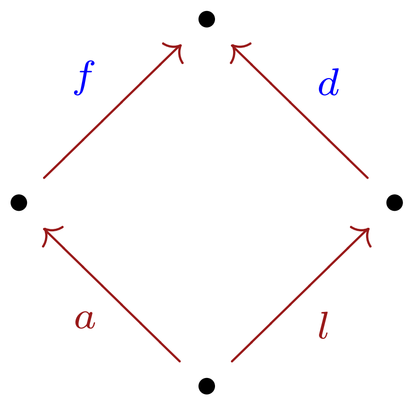


- a relative pushout (RPO) is a tuple $\langle \underline{c}, \underline{d}, p \rangle$ which satisfies the universal property that:
- for any other such $\langle c', d', p' \rangle$ there exists a unique mediating morphism m.

Deriving LTS

The LTS derived from the reactive system has:

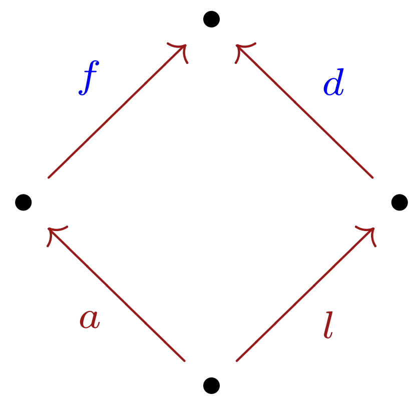
- Nodes: $a : I \to N$
- Transitions: $a \xrightarrow{f} dr$ iff for $\langle l, r\rangle \in \mathcal{R}$ and $d \in \mathbb{D}$, $\langle f, d, \mathrm{id}\rangle$ is a relative pushout (idem pushout or IPO) of the square



Deriving LTS

The LTS derived from the reactive system has:

- Nodes: $a : I \to N$
- Transitions: $a \xrightarrow{f} dr$ iff for $\langle l, r\rangle \in \mathcal{R}$ and $d \in \mathbb{D}$, $\langle f, d, \mathrm{id}\rangle$ is a relative pushout (idem pushout or IPO) of the square



- Thm. If all redex squares like the above have IPOs then the bisimulation on the derived LTS is a congruence [Leifer-Milner 00]

Applying RPOs

- When applied to term rewriting, RPOs yield the same LTS as Sewell's.
- Leifer (2000) found RPOs in a restricted class of action graph contexts.
- Milner (2001) worked out RPOs for a graphical formalism called bigraphs.
- Jensen and Milner (2002) derived (essentially) the usual bisimulation for asynchronous π using RPOs.

Applying RPOs

- When applied to term rewriting, RPOs yield the same LTS as Sewell's.
- Leifer (2000) found RPOs in a restricted class of action graph contexts.
- Milner (2001) worked out RPOs for a graphical formalism called bigraphs.
- Jensen and Milner (2002) derived (essentially) the usual bisimulation for asynchronous π using RPOs.

What about even very simple process calculi?

The technique doesn't actually scale up!

A Simple Calculus

Syntax:

$$p \ ::= \ \mathbf{0} \ \mid \ a \ \mid \ \overline{a} \ \mid \ p \mid p \qquad \text{where } a \in N.$$

Structural Congruence:

'$|$' associative, commutative with identity $\mathbf{0}$

Reactions:

$$a \mid \overline{a} \longrightarrow \mathbf{0}$$

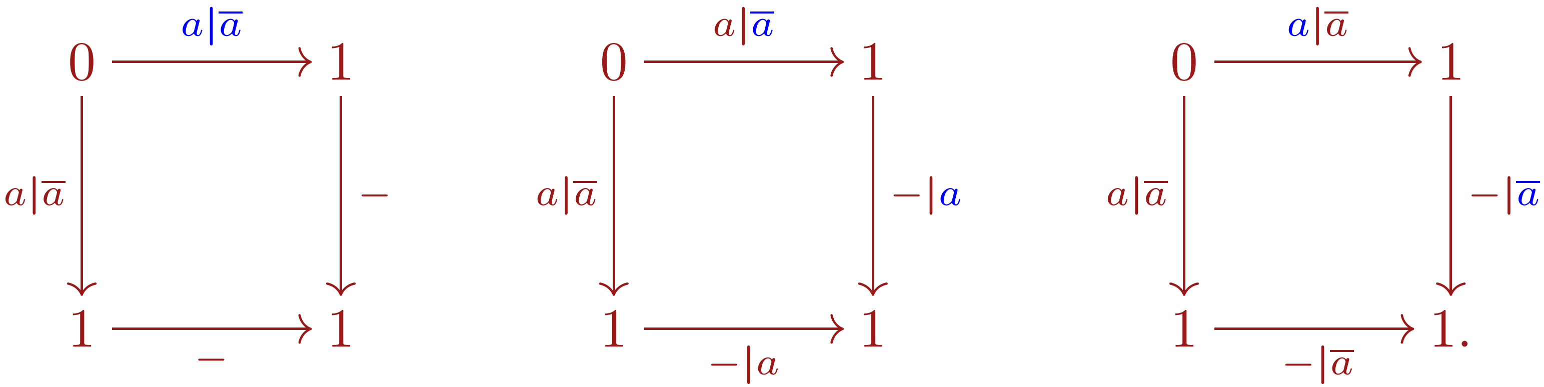
The Standard Labelled Transition System:

$$a \xrightarrow{a} \mathbf{0} \qquad \overline{a} \xrightarrow{\overline{a}} \mathbf{0} \qquad a \mid \overline{a} \xrightarrow{\tau} \mathbf{0}$$

$$\frac{p \xrightarrow{x} p'}{q \mid p \xrightarrow{x} q \mid p'} \qquad \frac{p \equiv p' \quad p' \xrightarrow{x} q' \quad q' \equiv q}{p \xrightarrow{x} q}$$

≪ ⋘ ⋙ ≫

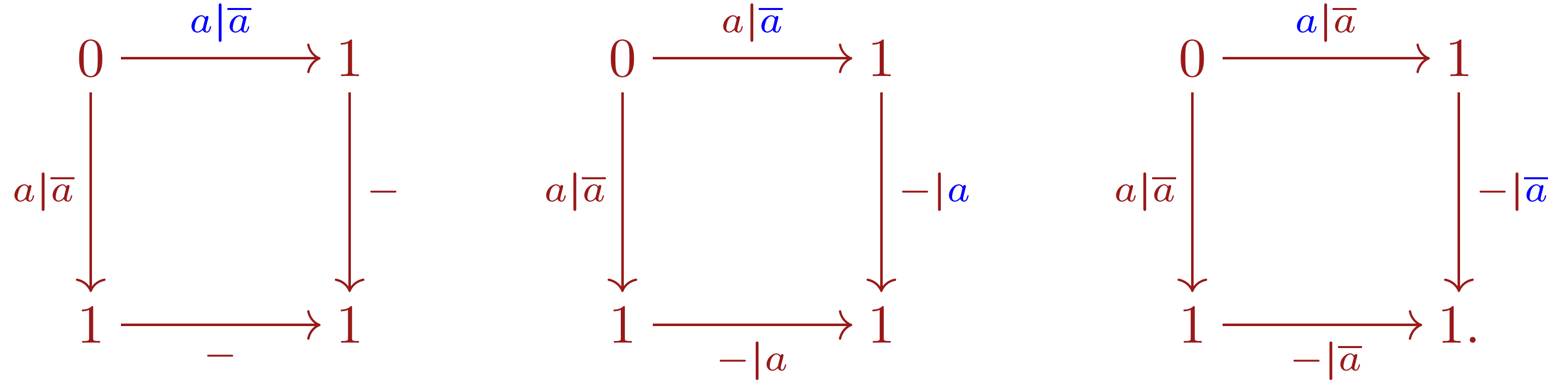
Simple Calculus ctd.



Only the left one could possibly be an IPO!

Yet, because of the structural congruence, the redex could partially come from the context. The derived LTS cannot account for this: only a $a \mid \overline{a} \xrightarrow{-} \mathbf{0}$ transition. And that is bad!

Simple Calculus ctd.



Only the left one could possibly be an IPO!

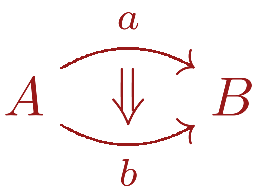
Yet, because of the structural congruence, the redex could partially come from the context. The derived LTS cannot account for this: only a $a \mid \bar{a} \xrightarrow{-} \mathbf{0}$ transition.
And that is bad!

We need to keep track of structural congruence to locate the reaction.

In other words, we need to keep track of how regions in the diagrams commute!

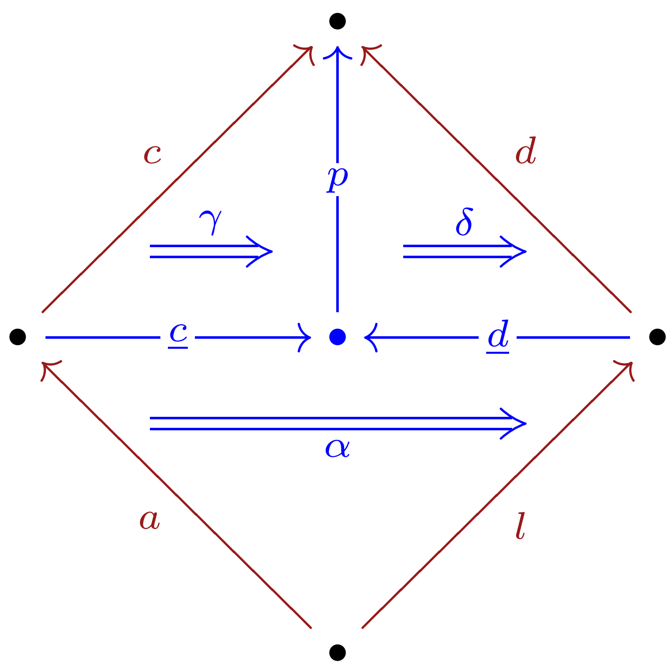
2-categories:

- Objects and arrows like in categories
- 2-cells: morphisms between arrows



GRPOs

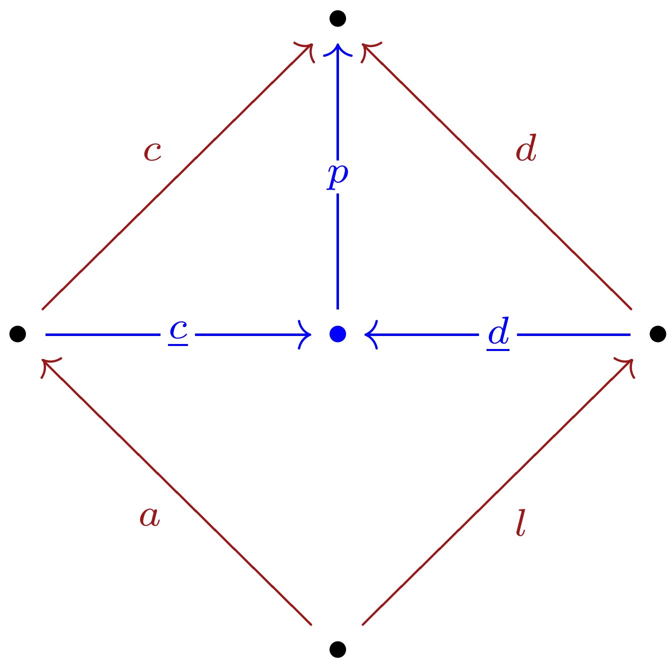
Suppose that $\mathbb{C}$ is a 2-category with all 2-cells isomorphisms



- a G-relative pushout (GRPO) is a tuple $\langle \underline{c}, \underline{d}, p, \alpha, \gamma, \delta \rangle$ which satisfies the universal property that:
- for any other such $\langle c', d', p', \alpha', \gamma', \delta' \rangle$ there exists an essentially unique mediating morphism m.

GRPOs

Suppose that $\mathbb{C}$ is a category



- a relative pushout (RPO) is a tuple $\langle \underline{c}, \underline{d}, p \rangle$ which satisfies the universal property that:
- for any other such $\langle c', d', p' \rangle$ there exists a unique mediating morphism m.

Simple Process Calculus ctd.

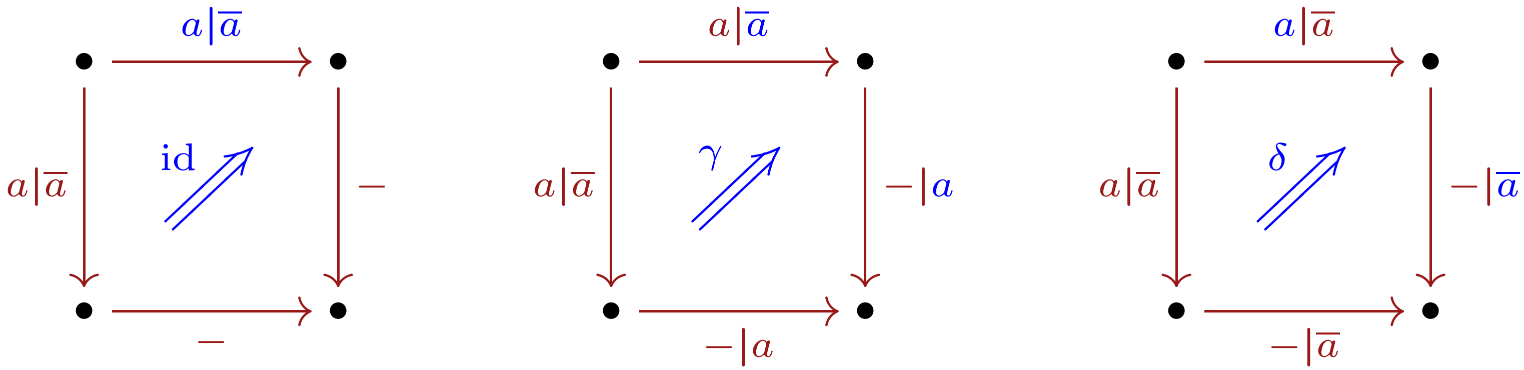
Let $\mathbb{C}$ be the 2-category with

- A single Object
- Arrows: strings $a_1 \mid a_2 \mid \ldots \mid a_n$
  - composition by concatenation
- 2-cells: permutations $a_1 \mid a_2 \mid \ldots \mid a_n \Rightarrow a_{\sigma(1)} \mid a_{\sigma(2)} \mid \ldots \mid a_{\sigma(n)}$

Then GRPOs exist and give the expected LTS.

In particular, $a \mid \overline{a}$ has transitions $a \mid \overline{a} \xrightarrow{-\mid a} a$ and $a \mid \overline{a} \xrightarrow{-\mid \overline{a}} \overline{a}$, witnesses of its potential interations with the environment.

Simple Calculus ctd.



Each of these squares is a GRPO!

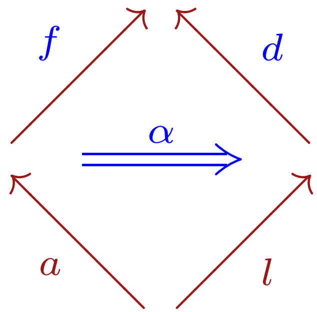
The 2-cells trace the structural congruence and place the reaction.

Note: γ and δ swap the 2nd and 3rd element, so as to put in evidence the intended redex.

Deriving LTS

The LTS derived from the reactive system with structural congruence has:

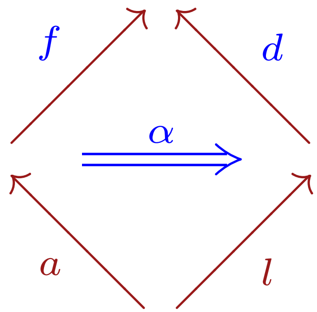
- Nodes: $[a] : I \to N$
- Transitions: $a \xrightarrow{[f]} dr$ iff there exists a 2-cell α such that for $\langle l, r \rangle \in \mathcal{R}$ and $d \in \mathbb{D}$, $\langle f, d, \mathrm{id}, \alpha, \mathbf{1}, \mathbf{1} \rangle$ is a G-relative pushout (G-idem pushout or GIPO) of the square



Deriving LTS

The LTS derived from the reactive system with structural congruence has:

- Nodes: $[a] : I \to N$
- Transitions: $a \xrightarrow{[f]} dr$ iff there exists a 2-cell α such that for $\langle l, r \rangle \in \mathcal{R}$ and $d \in \mathbb{D}$, $\langle f, d, \mathrm{id}, \alpha, \mathbf{1}, \mathbf{1} \rangle$ is a G-relative pushout (G-idem pushout or GIPO) of the square



Thm. If every square in $\mathbb{C}$ such as the one above has a GRPO, then the LTS bisimulation on the synthesised LTS is a congruence.

the theory is a generalisation of the theory of RPOs.

Conclusion

- GRPOs have been applied successfully to simple, yet significative examples such as Leifer-Milner's category of 'bunches and wires', to the theory of Milner's 'precategories' and Leifer's category 'above' construction.

  In all these cases, they uniformly yield labelled transition systems and bisimulation congruences better than those derived by the previous theories, while dispensing with complex, ad-hoc notions (such as 'trails', 'support sets and translations' and partially defined composition, and 'functorial reactive systems') in favour of streamline 2-category theory.

  So far, the price of the initial 2-categorical investment seems worth paying...

- Extend to more complicated process calculi (e.g. ambients), with complex structural congruences (e.g. replication).

- Apply the theory to graph rewriting to obtain interesting new semantics.