From Reactions to Observations: the Directed Bigraphical Model

Davide Grohmann, Marino Miculan

University of Udine

Institute of Cybernetics - Tallinn
February 4, 2010

Reaction Systems

- Semantics is specified by *reaction* (or "reduction") rules, which are pairs "(redex, reactum)". For instance:

$$\begin{array}{rcl} (5+3, 8) & \text{written as} & 5+3 \longrightarrow 8 \\ ((\lambda x.M)N, M\{N/x\}) & \text{written as} & (\lambda x.M)N \longrightarrow M\{N/x\} \end{array}$$

- A reaction system (RS) is specified by a set $\mathcal{R}$ of such rules, and possibly a family of *active contexts* where redexes have to be found in order to fire the rule.

$$\frac{(l, r) \in \mathcal{R}}{C[l] \longrightarrow C[r]}$$

- Only a silent, "internal" state changes.
- No interaction with the surrounding environment, thus no observation is specified.

Labelled Transition Systems

- Labelled transition systems are relations of the form

$$(P, Q, a) \quad \text{written as} \quad P \xrightarrow{a} Q$$

  where P, Q are systems (processes, programs with state, etc. . .) and a is a *label*, that is an *observation.*
- LTSs are used for defining the behaviour of calculi/systems because they endorse most important techniques for verifying properties (e.g., *model checking*) and observational equivalence (e.g., *bisimulations*).
- The labels should be enough to describe faithfully the aspects we are observing, still not too many to be impractible to use.
- In general good LTSs are difficult to describe, and often many ad hoc choices can be done (compare e.g. CCS, π-calculus and Ambient calculus).
- RSs are much easier to state than LTSs, but are not as useful!

Labelled Transition Systems from Reaction Systems?

Principle

What can be observed about a process P are its interactions with the surrounding environment.

Since a reaction system defines completely the behaviour of a system, it contains also the informations about interactions, although hidden.

Problem

Given a reaction system, is it possible to derive a “good” LTS?

By “good” we intend that

- the induced bisimulation must be a congruence
- labels should be not too many (otherwise it is difficult to use in practice)

Ad hoc solutions

Sometimes it can be done ad hoc, e.g, CCS: from reaction rule

$$a.P \mid \bar{a}.Q \longrightarrow P \mid Q$$

we *guess* the transitions

$$\alpha.P \xrightarrow{\alpha} P \qquad \frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\bar{a}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

because we *recognize* labels as the (minimal) interaction with the surrounding contexts.
Ad hoc solutions are difficult, error prone and require lot of work and experience. (Cf. the plethora of LTSs and bisimulations for π-calculus)

Aim

We look for a general, uniform way for deriving LTSs from RSs.

The “sledgehammer” approach

Define the observations (the *labels*) of an agent as the *contexts* which trigger a reaction rule

$$M[P] \longrightarrow Q \Longrightarrow P \xrightarrow{M} Q$$

More formally: a transition $a \xrightarrow{M} b$ is defined when there exist a reaction rule (l, r) and an active context D, such that $M \circ a = D \circ l$ and $b = D \circ r$.

$$\begin{tikzcd}
{} \arrow[r, "M"] & {} & \\
{} \arrow[u, "a"] \arrow[r, "l"'] & {} \arrow[u, "D"'] & {} \arrow[l, "r"] \arrow[ul, bend right, "b"']
\end{tikzcd}$$

Theorem

The bisimilarity induced by the contextual LTS is a congruence.

The "sledgehammer" approach

Define the observations (the *labels*) of an agent as the *contexts* which trigger a reaction rule

$$M[P] \longrightarrow Q \Longrightarrow P \xrightarrow{M} Q$$

More formally: a transition $a \xrightarrow{M} b$ is defined when there exist a reaction rule (l, r) and an active context D, such that $M \circ a = D \circ l$ and $b = D \circ r$.

$$\begin{tikzcd}
{} \arrow[r, "M"] & {} & \\
{} \arrow[u, "a"] \arrow[r, "l"'] & {} \arrow[u, "D"'] & {} \arrow[l, "r"] \arrow[ul, bend right, "b"']
\end{tikzcd}$$

Theorem

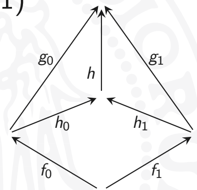
The bisimilarity induced by the contextual LTS is a congruence.

- But there are infinite labels for each process.
- Many labels are subsumed by simpler ones: let L be a label for every context $C[\cdot]$, $C[L]$ is a new label.
- How to restrict the set of labels to only "minimal" contexts?

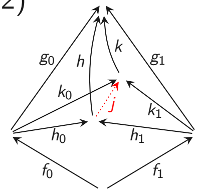
Relative and Idem Pushouts [Leifer, Milner 2000]

The “minimality” can be elegantly expressed as a universal categorical property.

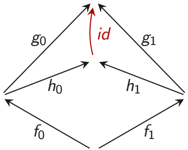
(1)



(2)



(3)



Call g_0, g_1 a *bound* for f_0, f_1 if $g_0 \circ f_0 = g_1 \circ f_1$.

1. A *relative bound* (h_0, h_1, h) for f_0, f_1 to g_0, g_1.
2. A *relative pushout (RPO)* (h_0, h_1, h) for f_0, f_1 to g_0, g_1: for any other relative bound (k_0, k_1, k), there is a unique mediator j.
3. A *idem pushout (IPO)* g_0, g_1 for f_0, g_1: (g_0, g_1, id) is an RPO for f_0, f_1 to g_0, g_1.

Relative and Idem Pushouts: a simple example

Processes

$P ::= a \mid P_1|P_2$

Contexts

$C ::= - \mid C|P \mid P|C$

Relative and Idem Pushouts: a simple example

Processes

$P ::= a \mid P_1 | P_2$

Contexts

$C ::= - \mid C|P \mid P|C$

$$
\begin{tikzcd}
 & {} & \\
{} \arrow[ur, "-|c|d|e"] & & {} \arrow[ul, "a|-|d|e"'] \\
 & {} \arrow[ul, "a|b"] \arrow[ur, "b|c"'] &
\end{tikzcd}
$$

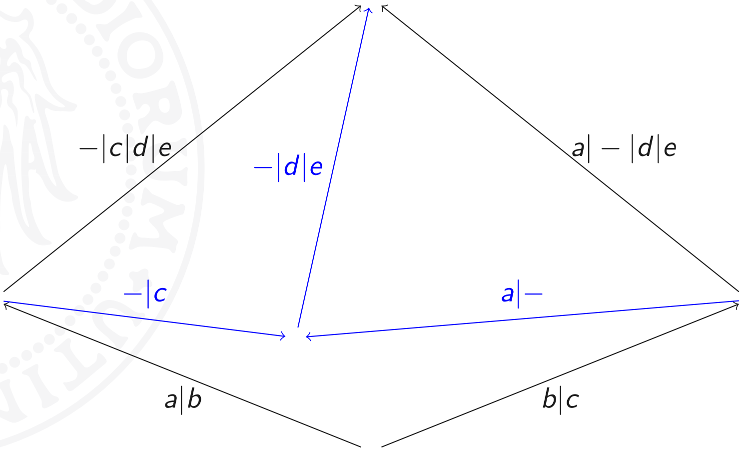
Relative and Idem Pushouts: a simple example

Processes

$P ::= a \mid P_1|P_2$

Contexts

$C ::= - \mid C|P \mid P|C$



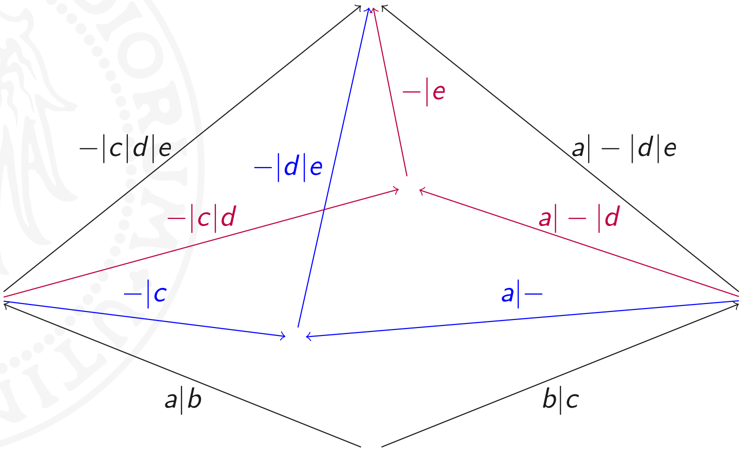
Relative and Idem Pushouts: a simple example

Processes

$P ::= a \mid P_1|P_2$

Contexts

$C ::= - \mid C|P \mid P|C$



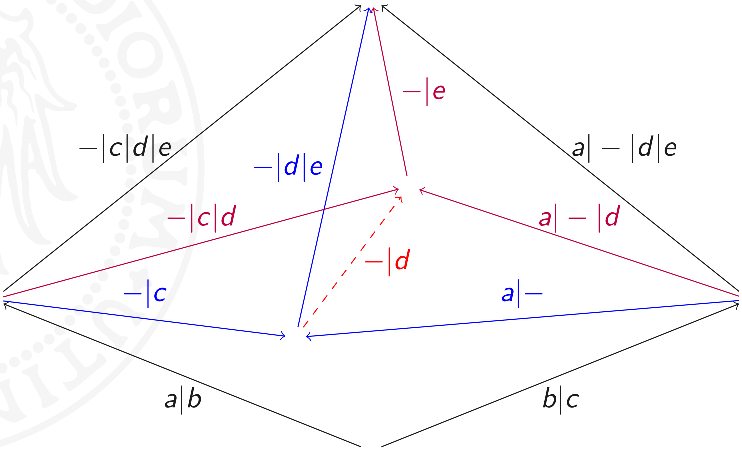
Relative and Idem Pushouts: a simple example

Processes

$P ::= a \mid P_1|P_2$

Contexts

$C ::= - \mid C|P \mid P|C$



LTSs from RSs and IPOs

Given a reactive system (by means of a set of reaction rules)

an *IPO transition* $a \xrightarrow{M} b$ is defined when there exist a reaction rule (l, r) and an active context D, such that (M, D) is an IPO for (a, l) and $b = D \circ r$.

$$\begin{tikzcd}
\cdot \arrow[r, "M"] & \cdot & \\
\cdot \arrow[u, "a"] \arrow[r, "l"'] & \cdot \arrow[u, "D"'] & \cdot \arrow[l, "r"] \arrow[ul, bend right, "b"']
\end{tikzcd}$$

Labels are only the contexts which form an IPO, that is, the *minimal completion context* for a that allows a reaction to take place, that is $M \circ a = D \circ l$ (and then $M \circ a$ rewrites into $D \circ r$).

Theorem (Leifer, Milner 2000)

The bisimilarity induced by an IPO LTS is a congruence.

The Plan: Metamodels with RPOs

For reaching our Aim ("general frameworks for turning RSs into LTSs"), we need to find:

- a category where RPOs exist and can be calculated;
- conditions for establishing when a span (f_0, f_1) has a bound (and hence an IPO), and how to calculate these IPOs;
- encoding methodologies, that is, how to represent calculi and systems (with reaction semantics) in these categories.

Those frameworks allow to obtain "automatically" a reduced LTS, whose bisimulation is a congruence and it is sound with respect to observational equivalence.

The Plan: Metamodels with RPOs

For reaching our Aim ("general frameworks for turning RSs into LTSs"), we need to find:

- a category where RPOs exist and can be calculated;
- conditions for establishing when a span (f_0, f_1) has a bound (and hence an IPO), and how to calculate these IPOs;
- encoding methodologies, that is, how to represent calculi and systems (with reaction semantics) in these categories.

Those frameworks allow to obtain "automatically" a reduced LTS, whose bisimulation is a congruence and it is sound with respect to observational equivalence.

Proposal

Bigraphical frameworks.

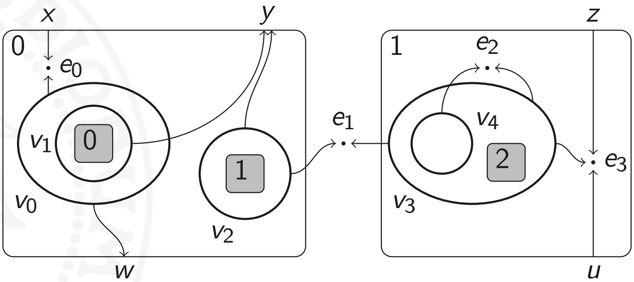
Directed Bigraphs

Directed bigraphs are a bi-graphical framework, in the style of Milner's Bigraphs.

- They have RPOs, thus allows the definition of LTS using the IPO construction.
- They unify previous, incompatible versions of bigraphs:
  - output-linear link graphs (i.e. Milner's);
  - input-linear link graphs (i.e. Sassone-Sobociński's).
- They allow to represent systems and calculi not yet covered by previous proposals.

Example of a directed bigraph

A directed bigraph has **nodes**, **edges** and **links**.

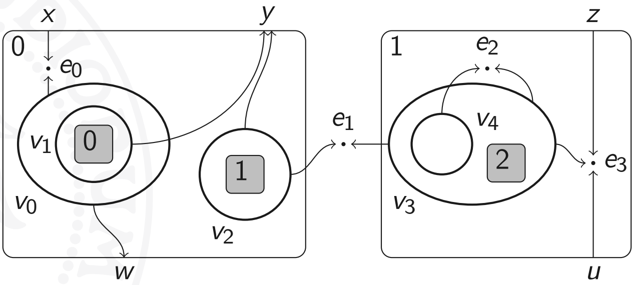


Nodes

Each node $v_0, v_1, \ldots,$ has an arity (i.e. a set of ports);

Example of a directed bigraph

A directed bigraph has **nodes**, **edges** and **links**.

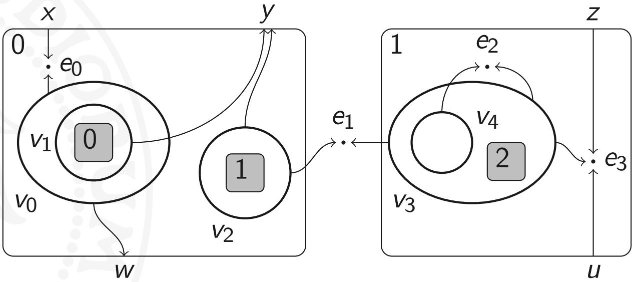


Edges

Edges $e_0, e_1, \ldots$ represent global resources.

Example of a directed bigraph

A directed bigraph has **nodes**, **edges** and **links**.

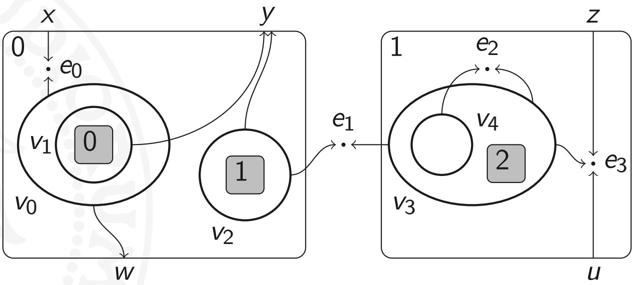


Ports

Ports "ask for connections" to resources.

Example of a directed bigraph

A directed bigraph has **nodes**, **edges** and **links**.

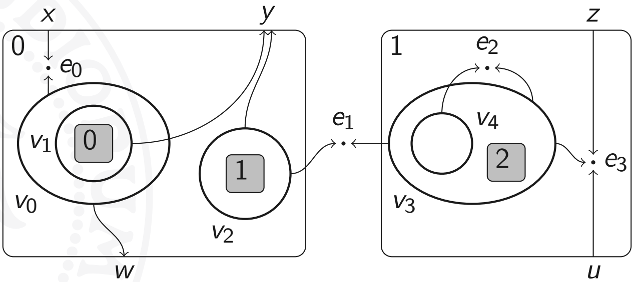


Placing

Nodes can be nested, instead edges are not subject to positions.
Sites are holes which can be fitted by roots of another bigraph.

Example of a directed bigraph

A directed bigraph has **nodes**, **edges** and **links**.

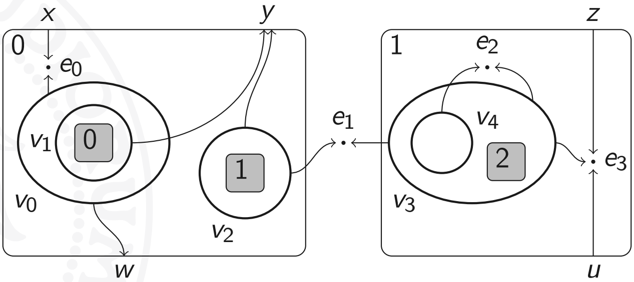


Links

Links describes the access/request of resources by nodes.

Example of a directed bigraph

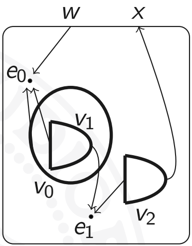
A directed bigraph has **nodes**, **edges** and **links**.



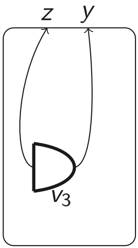
Names

Names $x, y, z, \ldots$ are "channels" through which controls can give or request access to resources.

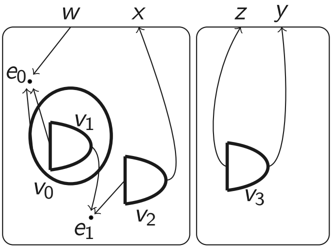
Interfaces and tensor product



$G_0 : \langle 0, (\emptyset, \emptyset)\rangle \to \langle 1, (\{w\}, \{x\})\rangle$



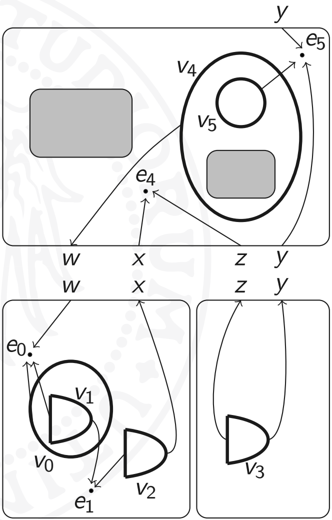
$G_1 : \langle 0, (\emptyset, \emptyset)\rangle \to \langle 1, (\emptyset, \{z, y\})\rangle$



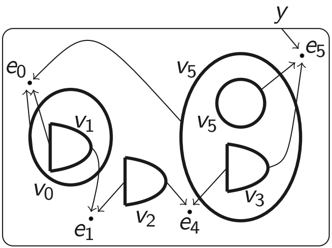
$G_0 \otimes G_1 : \langle 0, (\emptyset, \emptyset)\rangle \to \langle 2, (\{w\}, \{x, z, y\})\rangle$

Interfaces and composition

$H : \langle 2, (\{w\}, \{x, z, y\}) \rangle \to \langle 1, (\{y\}, \emptyset) \rangle$



$G : \langle 0, (\emptyset, \emptyset) \rangle \to \langle 2, (\{w\}, \{x, z, y\}) \rangle$

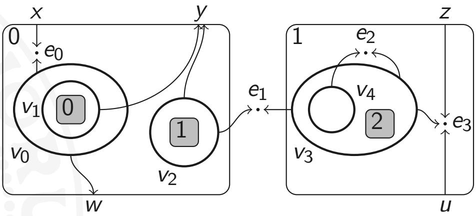


$H \circ G : \langle 0, (\emptyset, \emptyset) \rangle \to \langle 1, (\{y\}, \emptyset) \rangle$

directed bigraphs = place graphs + dir. link graphs

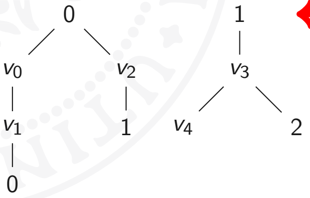
directed bigraph

$G : \langle 3, (\{w\}, \{u\}) \rangle \to \langle 2, (\{x, z\}, \{y\}) \rangle$



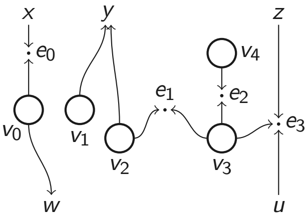
place graph

$G^P : 3 \to 2$



directed link graph

$G^L : (\{w\}, \{u\}) \to (\{x, z\}, \{y\})$



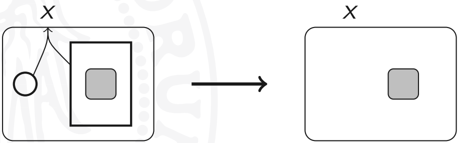
Reaction rules

A signature is a typing over nodes.

Reaction rules

A signature is a typing over nodes.

A reaction rule is a pair of (parametric) bigraphs.



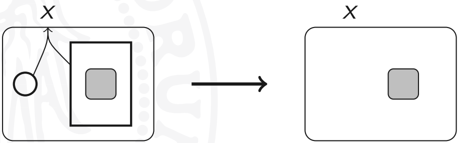
key is atomic

door is not atomic

Reaction rules

A signature is a typing over nodes.

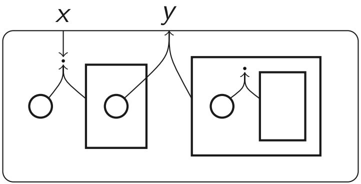
A reaction rule is a pair of (parametric) bigraphs.



key is atomic

door is not atomic

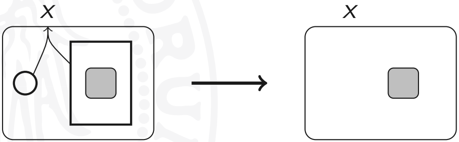
Reaction application: matching a redex inside an agent.



Reaction rules

A signature is a typing over nodes.

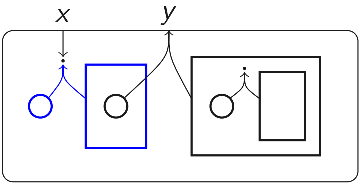
A reaction rule is a pair of (parametric) bigraphs.



key is atomic

door is not atomic

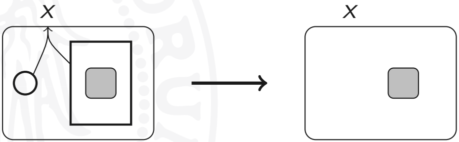
Reaction application: matching a redex inside an agent.



Reaction rules

A signature is a typing over nodes.

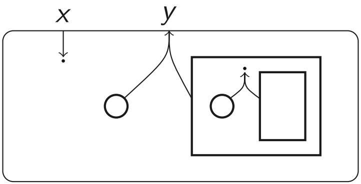
A reaction rule is a pair of (parametric) bigraphs.



key is atomic

door is not atomic

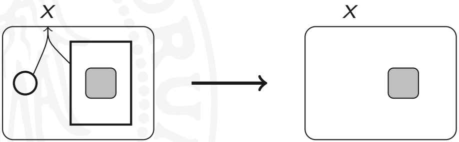
Reaction application: matching a redex inside an agent.



Reaction rules

A signature is a typing over nodes.

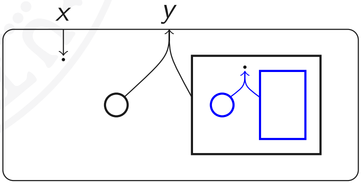
A reaction rule is a pair of (parametric) bigraphs.



key is atomic

door is not atomic

Reaction application: matching a redex inside an agent.

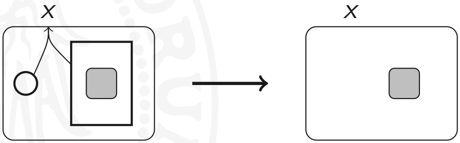


door is active

Reaction rules

A signature is a typing over nodes.

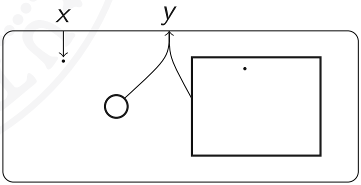
A reaction rule is a pair of (parametric) bigraphs.



key is atomic

door is not atomic

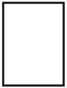
Reaction application: matching a redex inside an agent.



door is active

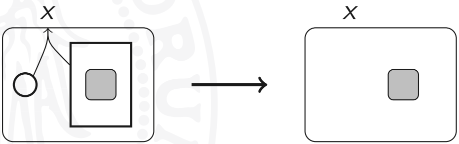
Reaction rules

A signature is a typing over nodes.


door


key

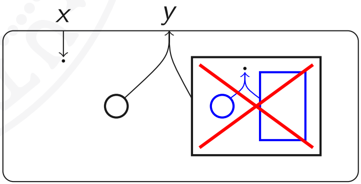
A reaction rule is a pair of (parametric) bigraphs.



key is atomic

door is not atomic

Reaction application: matching a redex inside an agent.

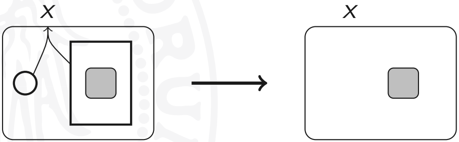


door is passive

Reaction rules

A signature is a typing over nodes.

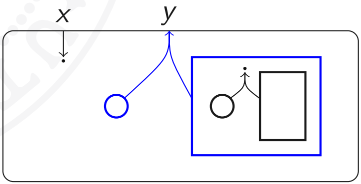
A reaction rule is a pair of (parametric) bigraphs.



key is atomic

door is not atomic

Reaction application: matching a redex inside an agent.

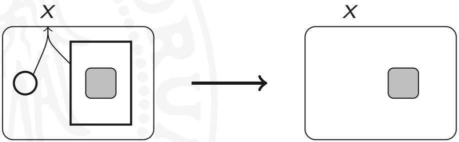


door is passive

Reaction rules

A signature is a typing over nodes.

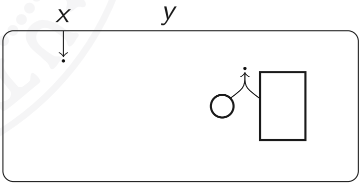
A reaction rule is a pair of (parametric) bigraphs.



key is atomic

door is not atomic

Reaction application: matching a redex inside an agent.

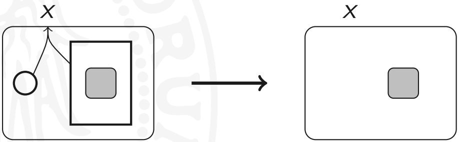


door is passive

Reaction rules

A signature is a typing over nodes.

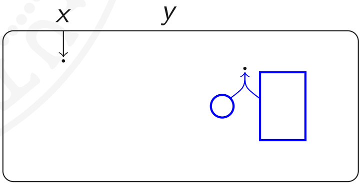
A reaction rule is a pair of (parametric) bigraphs.



key is atomic

door is not atomic

Reaction application: matching a redex inside an agent.

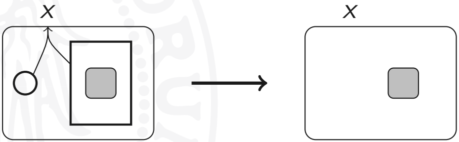


door is passive

Reaction rules

A signature is a typing over nodes.

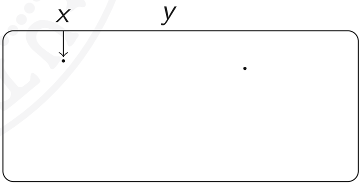
A reaction rule is a pair of (parametric) bigraphs.



key is atomic

door is not atomic

Reaction application: matching a redex inside an agent.



door is passive

Directed Bigraphical Reactive Systems

Directed Bigraphical Reactive System (DBRS)

A DBRS $\mathcal{D}(\mathcal{K}, \mathcal{R})$ (or simply $\mathcal{D}$) over a signature $\mathcal{K}$ is

1. the category $\mathrm{DB{\scriptsize IG}}(\mathcal{K})$
2. equipped with a set of (parametric) reaction rules $\mathcal{R}$.
3. A compositional reflective subcategory $\mathcal{A}$ of *active contexts*.

- A signature $\mathcal{K}$ is a set of types which can be assigned to nodes.
- A parametric rule is a pair (*redex*, *reactum*) of bigraphs that can have holes, this describes a set of ground rules.
- A rule can be fired only when its redex appears in an active context (i.e. a bigraphs with holes).

Example: The Fusion Calculus

Process syntax

$$P, Q ::= \mathbf{0} \mid zx.P \mid \bar{z}x.P \mid P|Q \mid (x)P$$

(up to a structural congruence $\equiv$)

System configuration

A system configuration is denoted by a pair (P, φ) to mean that P has associated fusion φ, which is an equivalence relation of the form $\{x_1 = y_1, \ldots, x_n = y_n\}$.

Rewriting semantics

$$(Com)\ (xy.P|\bar{z}w.Q|R, \varphi) \rightarrow (P|Q|R, (\varphi \cup \{y = w\})^*)$$

if $x\,\varphi\,z$

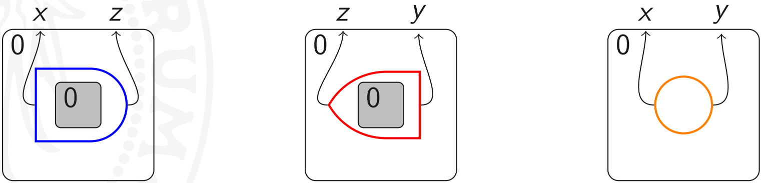
$(\ )^*$ is the reflexive and transitive closure.

The Fusion Calculus - Signature

The corresponding bigraphical signature is

$$\mathcal{K}_F = \{\mathsf{get} : 2, \mathsf{send} : 2, \mathsf{fuse} : 2\},$$

where get and send are passive and fuse is atomic as follows:



$\mathsf{get}^{x,z}$ $\mathsf{send}^{y,z}$ $\mathsf{fuse}^{x,y}$

The place graph will represent the syntactic tree of the fusion processes.

Encoding of fusion processes in bigraphs

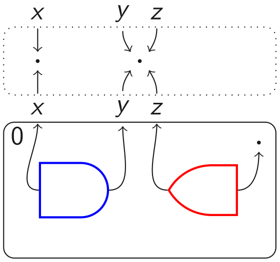
A fusion process is encoded into a bigraph in two steps:

1. first, we translate the process into a bigraph without resources (i.e. edges, representing equivalence class of names), except for the bindings;
2. next, we add the resources according to a fusion relation, and eventually fusing names together, that is the names are linked to the same equivalence class.

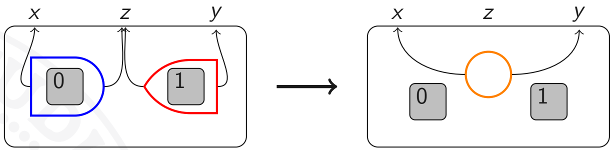
$$\sum_{[n]\in\{z=y\}} \nabla^{[n]}_{n} \circ \text{⧗}^{n}_{n} \circ \triangle^{n}_{[n]}$$

$$[\![(yx.\mathbf{0} \mid (x)\bar{z}x.\mathbf{0}, \{z = y\})]\!]$$

$$[\![yx.\mathbf{0} \mid (x)\bar{z}x.\mathbf{0}]\!]_{\{y,z,x\}}$$

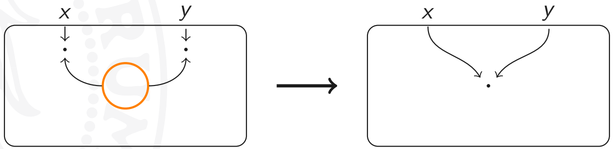


The Fusion Calculus - Rules



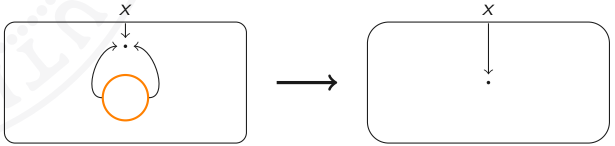
Com

$$\mathrm{get}^{x,z} \curlywedge \mathrm{send}^{y,z} \longrightarrow \mathrm{fuse}^{x,y} \curlywedge \triangle^{z} \curlywedge id_1 \curlywedge id_1 \qquad \rho(0) \mapsto 0 \quad \rho(1) \mapsto 1$$



Fuse

$$(⧗^{x}_{x} \otimes ⧗^{y}_{y}) \circ \mathrm{fuse}^{x,y} \longrightarrow \triangledown^{x,y}_{z} \circ \blacktriangledown^{z}$$



Disp

$$⧗^{x}_{x} \circ \triangle^{x}_{y,z} \circ \mathrm{fuse}^{y,z} \longrightarrow \blacktriangledown^{x}$$

Adequacy of the encoding

Proposition - Syntax

Let P and Q be two processes and φ any fusion; then $P \equiv Q$ if and only if $[\![(P,\varphi)]\!] = [\![(Q,\varphi)]\!]$.

Proposition - Semantics

1. if $(P,\varphi) \rightarrow (P',\varphi')$ then $[\![(P,\varphi)]\!] \longrightarrow^2 [\![(P',\varphi')]\!]$;
2. if $[\![(P,\varphi)]\!] \longrightarrow G$ then $\exists P',\varphi'.\ (P,\varphi) \rightarrow (P',\varphi')$ and $G \longrightarrow [\![(P',\varphi')]\!]$.

Comparing labelled transition systems

We can compare the systematically derived LTS with the original (ad hoc) one given by Parrow and Victor.

Labelled transition system

$$\frac{-}{\alpha.P \xrightarrow{\alpha} P} \qquad \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \qquad \frac{P \xrightarrow{uz} P',\ u \notin \{z, \bar{z}\}}{(z)P \xrightarrow{(z)uz} P'}$$

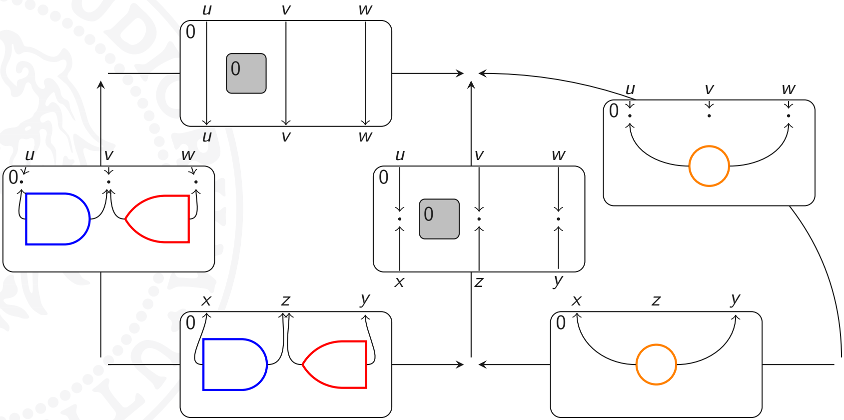
$$\frac{P \xrightarrow{\{x=y\}} P'}{(y)P \xrightarrow{\mathbf{1}} P'\{x/y\}} \qquad \frac{P \xrightarrow{\alpha} P',\ x \notin n(\alpha)}{(x)P \xrightarrow{\alpha} (x)P'} \qquad \frac{P \xrightarrow{ux} P',\ Q \xrightarrow{\bar{u}y} Q'}{P|Q \xrightarrow{\{x=y\}} P'|Q'}$$

Hyperequivalence

The bisimulation (*hyperbisimulation*) needs to be closed under all substitutions to be a congruence (*hyperequivalence*, $\sim_F$).

Examples of transitions - Communication

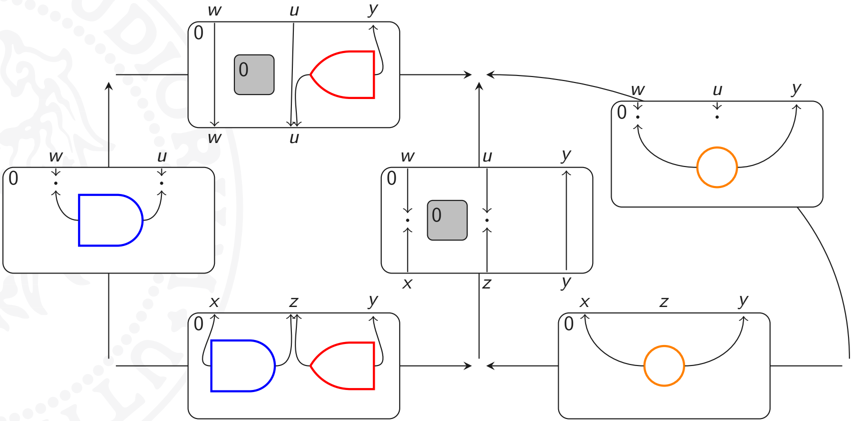
Half step of the *Com* rule. The "identity label" corresponds to τ.



$$[\![(vu \mid \bar{v}w, \emptyset)]\!] \xrightarrow{id_{\langle 1,(\{u,v,w\},\emptyset)\rangle}} ((\blacktriangledown^u_x \otimes \blacktriangledown^w_y) \circ \mathsf{fuse}^{x,y}) \otimes \blacktriangledown^v$$

Examples of transitions - Prefix

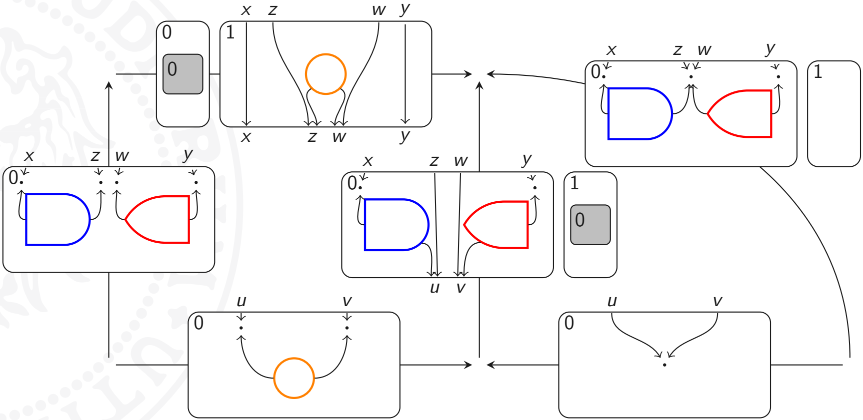
Completing an input with an output. This label corresponds to *uw*.



$$[\![(uw, \emptyset)]\!] \xrightarrow{id_{\langle 1,(\{u,w\},\emptyset)\rangle} | \mathsf{send}^y_u} ((\blacktriangledown\!\!\blacktriangle^w_x \otimes id_y) \circ \mathsf{fuse}^{x,y}) \otimes \blacktriangledown^u$$

Examples of transitions - Communication enabling

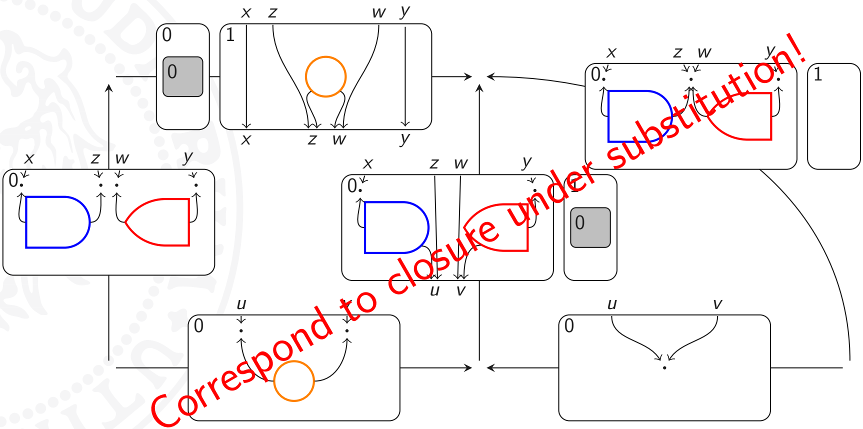
Enabling a communication by merging two different names.



$$[\![(xz \mid \bar{w}y, \emptyset)]\!] \xrightarrow{id_1 \otimes (id_{(\{x,y,z,w\},\emptyset)} | \mathsf{fuse}_{z,w})} [\![(xz \mid wy, \{z = w\})]\!] \otimes 1$$

Examples of transitions - Communication enabling

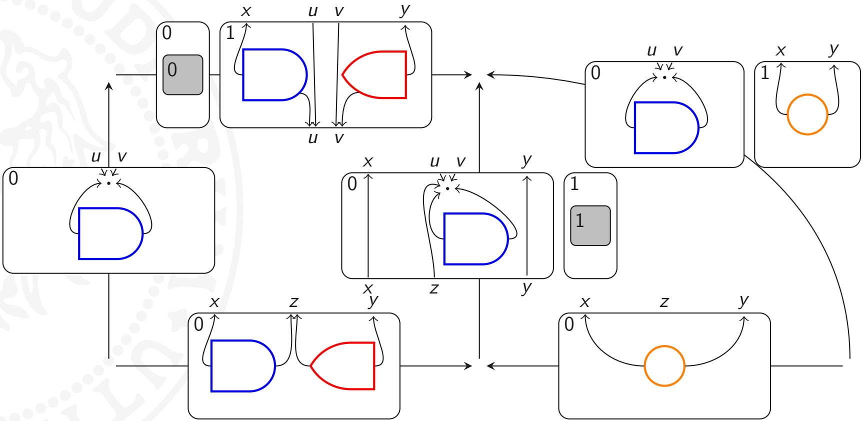
Enabling a communication by merging two different names.



$$[\![(xz \mid \bar{w}y, \emptyset)]\!] \xrightarrow{id_1 \otimes (id_{(\{x,y,z,w\},\emptyset)} | \mathsf{fuse}_{z,w})} [\![(xz \mid wy, \{z = w\})]\!] \otimes 1$$

Examples of transitions - Equality test

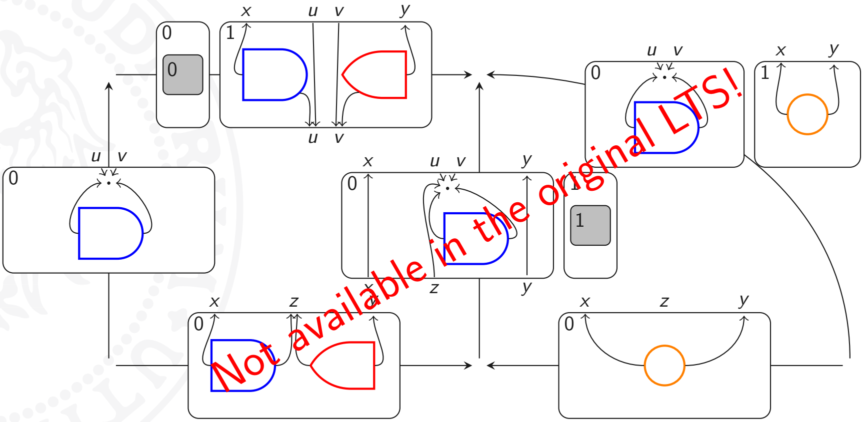
A test transition. An agent can perform this transition if $u \varphi v$.



$$[\![(uv, \{u = v\})]\!] \xrightarrow{id_1 \otimes (id_{(\{u,v\},\emptyset)} | \mathrm{get}^x_u | \mathrm{send}^y_v)} (\nabla^{u,v}_s \circ \bowtie^s_s \circ get^{s,s}) \otimes \mathrm{fuse}^{x,y}$$

Examples of transitions - Equality test

A test transition. An agent can perform this transition if $u \,\varphi\, v$.



$$[\![(uv, \{u = v\})]\!] \xrightarrow{id_1 \otimes (id_{(\{u,v\},\emptyset)} | \mathrm{get}^x_u | \mathrm{send}^y_v)} (\nabla^{u,v}_s \circ \bowtie^s_s \circ get^{s,s}) \otimes \mathrm{fuse}^{x,y}$$

IPO bisimilarity is sound

From IPO bisimilarity ($\sim_{\mathrm{IT}}$) we can define a congruence on processes of Fusion calculus:

$$P \sim_{\mathrm{IT}} Q \overset{\triangle}{\iff} \forall \varphi. [\![(P, \varphi)]\!] \sim_{\mathrm{IT}} [\![(Q, \varphi)]\!]$$

This congruence can be characterized more easily as follows:

Proposition

For all P, Q: $P \sim_{\mathrm{IT}} Q \iff [\![(P, \emptyset)]\!] \sim_{\mathrm{IT}} [\![(Q, \emptyset)]\!]$.

Theorem

$\sim_{\mathrm{IT}} \subseteq \sim_F$

Proof. Prove that $\sim_{\mathrm{IT}}$ is an hyperbisimulation.

What about completeness?

Claim

$\sim_F \subseteq \sim_{\mathrm{IT}}$

Difficult to prove because labels and descendants in bigraphical encoding may not represent any Fusion process (e.g. width > 1).

Two possible approaches:

1. prove that $\sim_F$ is an IPO bisimulation by means of some up-to technique (e.g. progressions);
2. try to restrict to correct agents (e.g. by sorting).

Conclusion

Summary

Directed bigraphs provide a general operational framework where reactive systems can be presented and studied.
LTS with compositional bisimulations can be systematically derived.

Other results not shown here

1. A complete algebraic axiomatization for directed bigraphs, based on a set of elementary bigraphs and a normal form.
2. How to reduce further the number of derived labels.
3. Encoding of Petri nets, web services and chemical reactions.
4. Apply the model to system biology, trying encoding (and possibly extending) some important formalisms as κ-calculus and brane calculus. (Marino's talk in Andu)

Future Work

1. Web service interaction, such as SCC or CC-Pi.
2. Adding quantitative aspects (i.e., reaction rates).
3. Apply sorting techniques a là Birkedal-Debois to directed bigraphs, to obtain bindings, locality of names,...
4. Implementation and tools. Work in progress
5. Generalize the framework to deal with n-graph-like structures. Work in progress (My talk in Andu)