On decidability of bigraphical sortings

Giorgio Bacci **Davide Grohmann**

Department of Mathematics and Computer Science
University of Udine, Italy

CALCO-jnr, September, 6 2009

Outline

- Bigraphical framework
- Sortings and Predicate Sortings
- Undecidability of Sortings
- Subclass of decidable Sortings

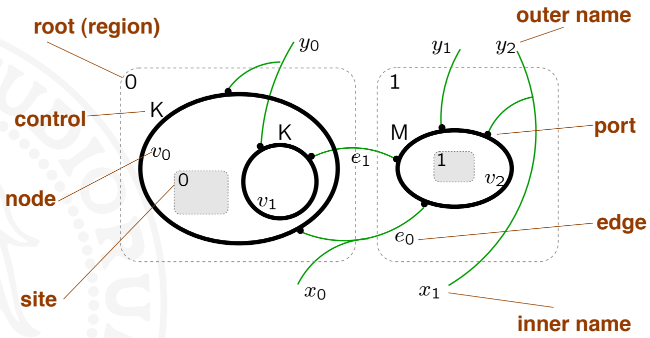
Outline

- Bigraphical framework
- Sortings and Predicate Sortings
- Undecidability of Sortings
- Subclass of decidable Sortings

Bigraphical Framework

- Bigraphical models are an emerging framework for concurrency and mobility.
- Long term aim: "to express as much as possible of worldwide distributed computing in one mathematical model" (Milner 2001).
- Bigraphs aim to be such a **framework**, i.e., unifying model for computations based on communications and locality.
- Many calculi have been represented in bigraphs: CCS, π-calculus, Petri nets, ...
- ...but they can be applied also for dealing with Systems Biology! (as shown yesterday at MeCBIC 2009)

An example of a bigraph



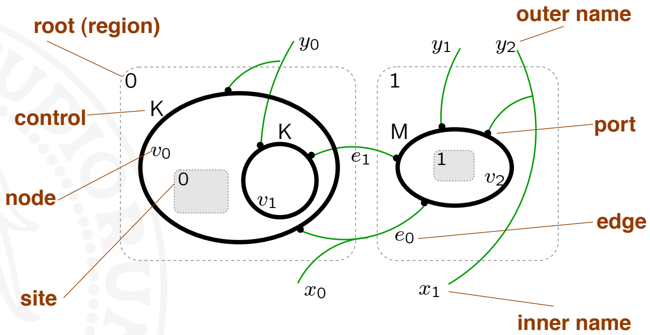
place = **root** or **node** or **site**

link = **edge** or **outer name**
point = **port** or **inner name**

Nodes

Each node $v_0, v_1, \ldots$ has an associate control which specifies its arity (i.e., a set of ports).

An example of a bigraph



place = **root** or **node** or **site**

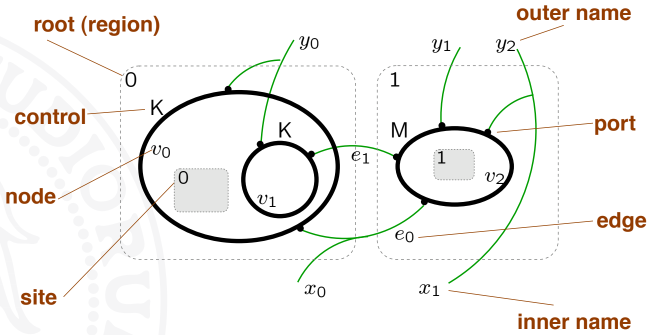
link = **edge** or **outer name**

point = **port** or **inner name**

Edges

Edges $e_0, e_1, \ldots$ represent global closed links.

An example of a bigraph



place = **root** or **node** or **site**

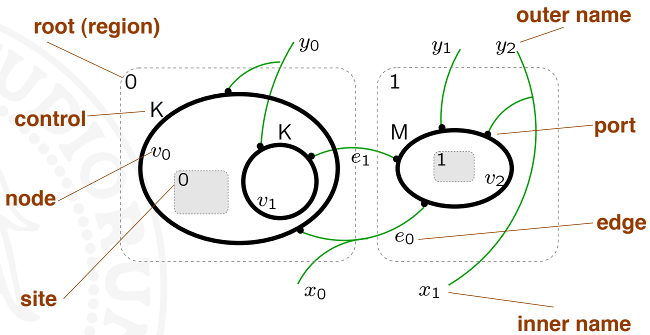
link = **edge** or **outer name**

point = **port** or **inner name**

Names

Outer names represents global open links.
Inner names represents connections coming from "sub-bigraphs".

An example of a bigraph



place = **root** or **node** or **site**

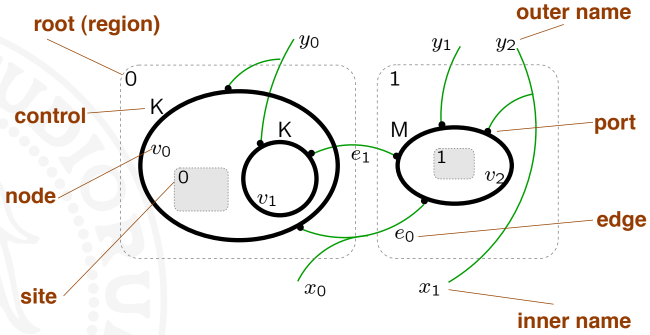
link = **edge** or **outer name**

point = **port** or **inner name**

Ports

Ports ask for connections to open or closed links.

An example of a bigraph



place = **root** or **node** or **site**

link = **edge** or **outer name**

point = **port** or **inner name**

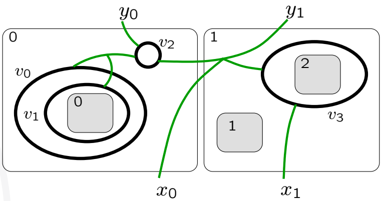
Placing

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

a bigraph = a place graph + a link graph

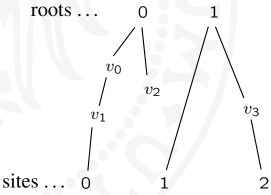
bigraph

$G\colon \langle m, X\rangle \to \langle n, Y\rangle$



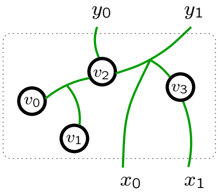
place graph

$G^{\mathrm{P}}\colon m \to n$



link graph

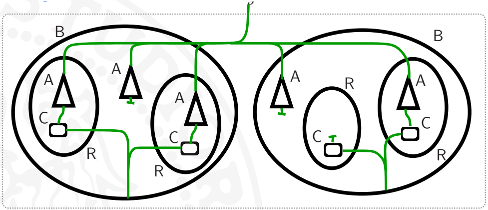
$G^{\mathrm{L}}\colon X \to Y$

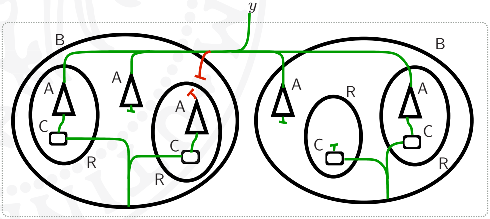


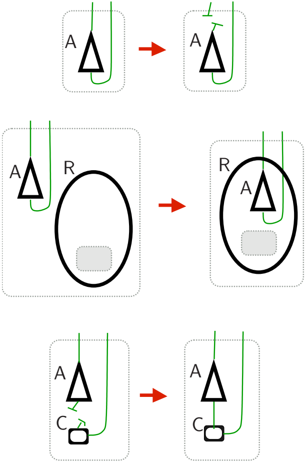
… outer names

… inner names

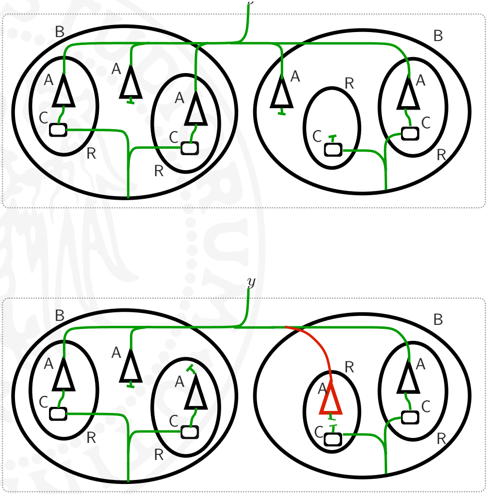
Dynamics: reaction rules

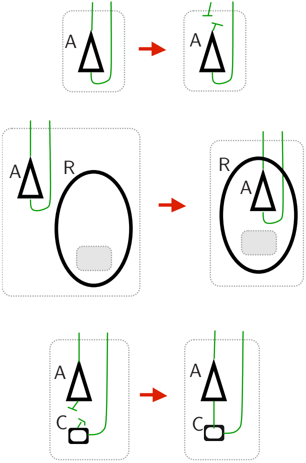




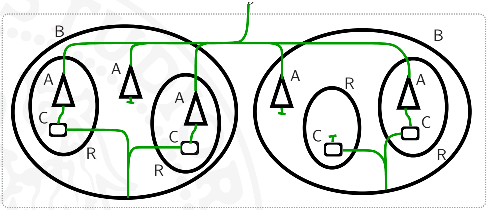


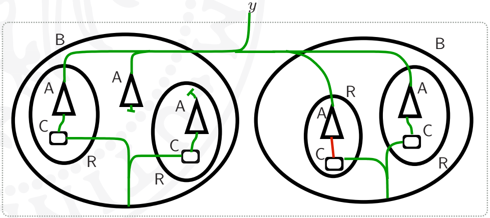
Dynamics: reaction rules

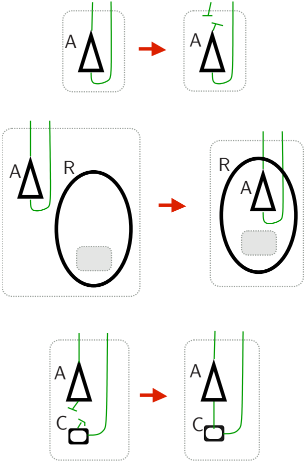




Dynamics: reaction rules







Outline

Sorting Motivations

What a sorting gives you

- Techniques to specific a sort/typing over bigraph's elements, that is nodes and edges.
- Techniques to impose a formation rule that limits the admissible bigraphs, that is it rules out unwanted bigraphs.

Why sortings?

Bigraphs is a very general framework, maybe even too general!
Leifer and Milner claimed that:
Sortings are likely to be needed in any significant application.

Remarkable property

Sortings preserve the behavioral theory of bigraph.

Sortings and Predicate Sortings

Definition (Sorting)

A *sorting* for a category C is a functor $F : X \to C$ that is *faithful* and *surjective on objects*.

Definition (Decomposible Predicate)

A predicate P on morphisms is *decomposible* iff it reflects identities and $P(f \circ g)$ implies $P(f)$ and $P(g)$.

Theorem (Factorization)

A predicate P is decomposible iff there exists a set Φ of morphisms such that $P(f)$ iff for any g, ψ, h if $f = g \circ \psi \circ h$ then $\psi \notin \Phi$.

Intuitively

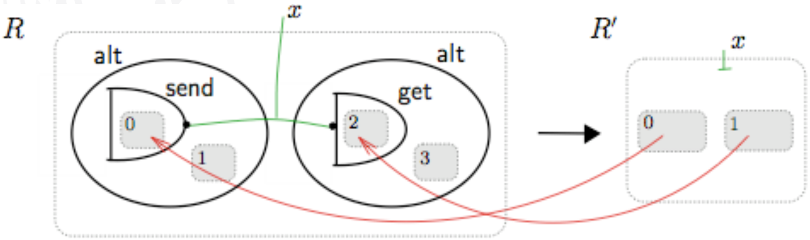
The set Φ describes all the unwanted processes/systems, so a *Predicate Sorting* rules out all the unwanted systems.

CCS

Syntax: $\alpha ::= a \mid \bar{a} \qquad P ::= \mathbf{0} \mid \sum_i \alpha_i.P_i \mid P|P$

Semantics: $(a.P + \sum_i \alpha_i.P_i)|(\bar{a}.Q + \sum_j \alpha_j.Q_j) \to P|Q$

In Bigraphs:



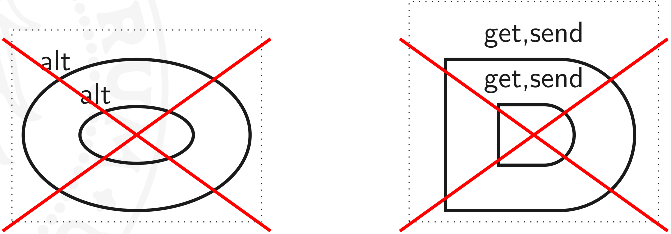
$\mathsf{alt}.(\mathsf{send}_x.d_0 \mid d_1) \mid \mathsf{alt}.(\mathsf{get}_x.d_2 \mid d_3)$ $\qquad$ $x \mid d_0 \mid d_2$

CCS

Syntax: $\alpha ::= a \mid \bar{a} \qquad P ::= \mathbf{0} \mid \sum_i \alpha_i.P_i \mid P|P$

Semantics: $(a.P + \sum_i \alpha_i.P_i)|(\bar{a}.Q + \sum_j \alpha_j.Q_j) \rightarrow P|Q$

"Bad-formed" bigraphs:



Solution

Define a predicate sorting by defining the set Φ (of unwanted bigraphs) as the set containing the above bigraphs.

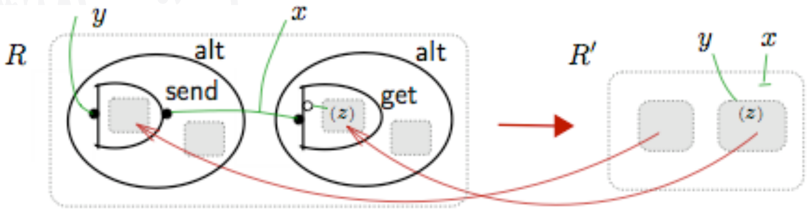
Sortings by examples: π-calculus - I

π-calculus

Syntax: $\alpha ::= a(x) \mid \bar{a}b \qquad P ::= \mathbf{0} \mid \sum_i \alpha_i.P_i \mid P|P$

Semantics: $(a(x).P + \sum_i \alpha_i.P_i)|(\bar{a}b.Q + \sum_j \alpha_j.Q_j) \rightarrow P\{b/x\}|Q$

In Bigraphs:



$$\mathsf{alt}.(\mathsf{send}_{xy}.d_0 \mid d_1) \mid \mathsf{alt}.(\mathsf{get}_{x(z)}.d_2 \mid d_3) \longrightarrow x \mid d_0 \mid {}^{y}/_{(z)}.d_2$$

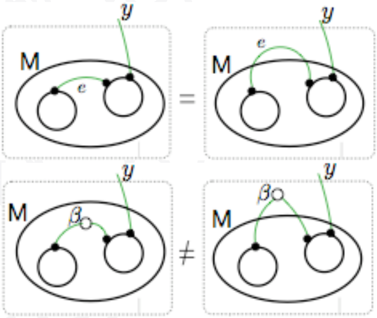
Sortings by examples: π-calculus - II

π-calculus

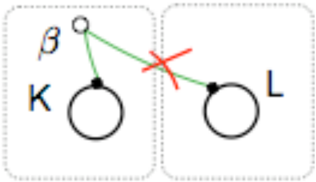
Syntax: $\alpha ::= a(x) \mid \bar{a}b \qquad P ::= \mathbf{0} \mid \sum_i \alpha_i.P_i \mid P|P$

Semantics: $(a(x).P + \sum_i \alpha_i.P_i)|(\bar{a}b.Q + \sum_j \alpha_j.Q_j) \to P\{b/x\}|Q$

Bigraphs with "sorted edges":



"Bad-formed" bigraphs:



Solution

Again use a Predicate Sorting.

Outline

Undecidability of (Predicate) Sortings

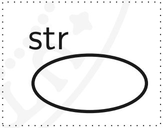
The problem

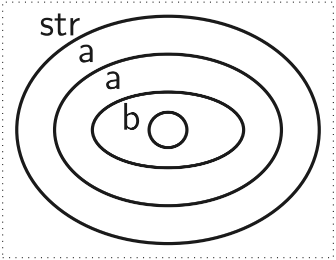
To decide if a bigraph is in a predicate sorted category involves to decide if a bigraph belongs to the set Φ.

Idea

Reduce the problem to a undecidability problem: to decide if a word $w \in \{a, b\}^*$ belongs to a co-RE language $\mathcal{L} \subset \{a, b\}^*$.

Encoding of words ($[\![-]\!]$):

$[\![\epsilon]\!] =$ 

$[\![aab]\!] =$ 

The reduction

- Let $\mathcal{L}$ be co-RE language.
- Take $\Phi = [\![\mathcal{L}]\!]$.
- Let a bigraph $f = g \circ \psi \circ h$.
- Does the bigraph ψ belong to Φ?

To decide if a bigraph ψ belongs to Φ is undecidable

even if there are **finite** possible decompositions $f = g \circ \psi \circ h$

The reduction

- Let $\mathcal{L}$ be co-RE language.
- Take $\Phi = [\![\mathcal{L}]\!]$.
- Let a bigraph $f = g \circ \psi \circ h$.
- Does the bigraph ψ belong to Φ?

To decide if a bigraph ψ belongs to Φ is undecidable

even if there are **finite** possible decompositions
$f = g \circ \psi \circ h$

Outline

- Bigraphical framework
- Sortings and Predicate Sortings
- Undecidability of Sortings
- Subclass of decidable Sortings

Decidable Sortings

Are there decidable Sortings? Yes!

Observation

To find an "unwanted bigraph" resembles the matching problem, i.e., to find a (sub)bigraph inside another one.

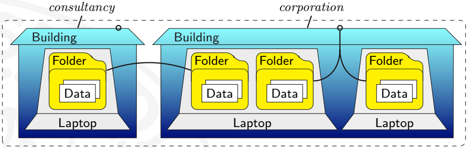
Idea of how to construct decidable Sortings

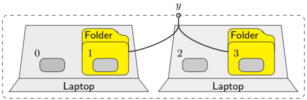
1. Take a recursive set M of unwanted bigraphs.
2. Define $\Phi = \{m \otimes id_X \mid m \in M \wedge X \text{ is a name set}\}$.
3. Specialize the Factorization Theorem.

Theorem

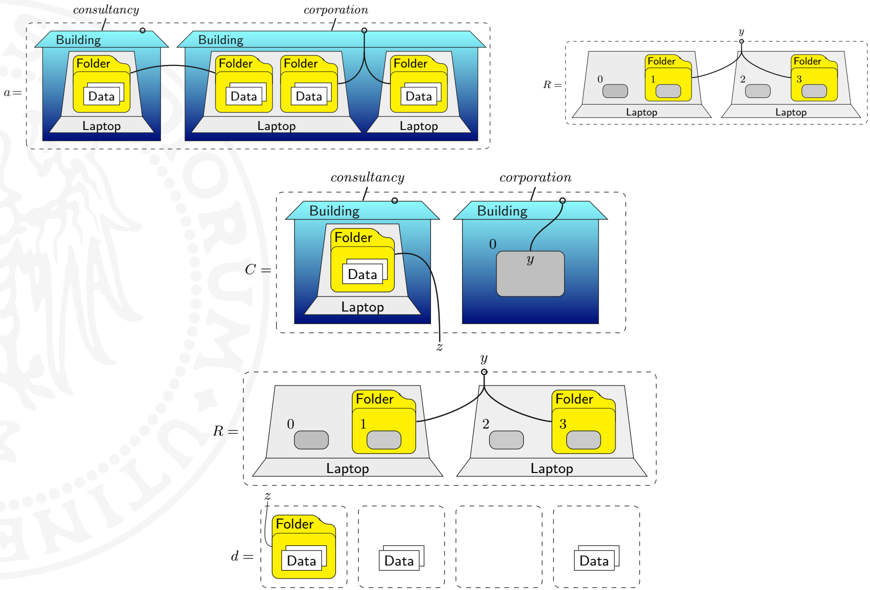
P is match-decomposable iff there exists Φ finite such that $P(f)$ iff for any g, ψ, h, X if $f = g \circ (\psi \otimes id_X) \circ h$ then $(\psi \otimes id_X) \notin \Phi$.

Matching by an example

$a =$ 

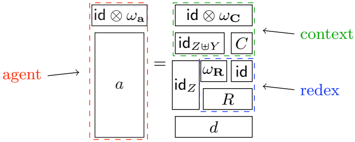
$R =$ 

Matching by an example

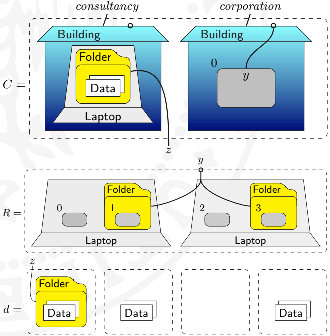


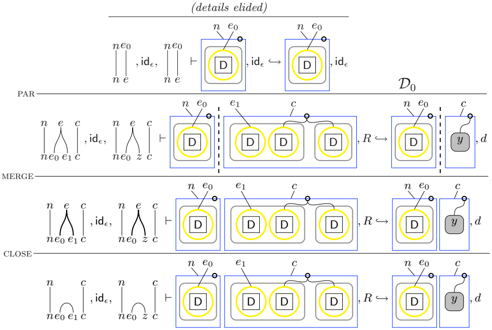
How to decide the matching problem

1. Transform the agent and the redex into their normal forms.
2. Use the inference systems from [Damgaard et al., 2007] to derive the context and the parameters.

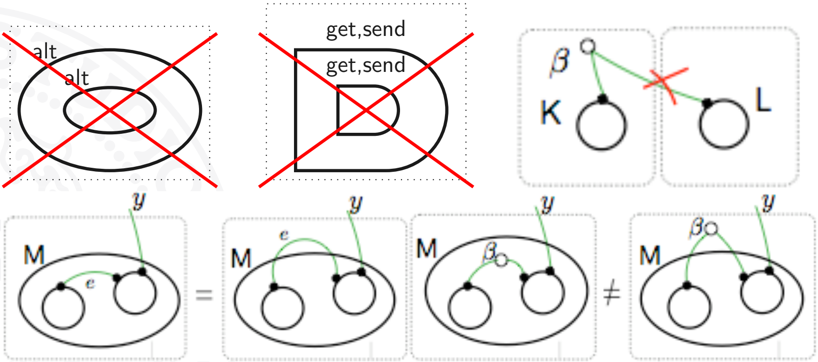


$$\overbrace{(\mathsf{id}\otimes\omega_{\mathbf{a}})a}^{\text{agent}} = \overbrace{(\mathsf{id}\otimes\omega_{\mathbf{C}})(\mathsf{id}_{Z\uplus Y}\otimes C)}^{\text{context}}(\mathsf{id}_Z\otimes\overbrace{(\mathsf{id}\otimes\omega_{\mathbf{R}})R}^{\text{redex}})d.$$





What about the Sortings for CCS and π-calculus?



Independence from identities

- Sorted elements do not depend on identities.
- Sorting on nestings and linkings are also independent from ids.
- Our Sortings are **"Homset Independent"**.

Conclusion and Future Work

Done

- We have proved that (Predicate) Sortings are undecidable.
- We have shown a way to construct decidable Sortings, based on the decidability of the matching problem.
- Those Sortings are powerful enough to capture some of the Sortings introduced in literature.

To do

- Study if other known Sortings can be expressed with our construction.
- Analyze the complexity of our approach.
- Investigate if more optimized algorithms exist.
- Integration into Tools?