

Mathematical Theories of Abstraction,
Substitution and Naming in Computer Science
ICMS, Edinburgh, May 26-28 2007

Bigraphical models of calculi with names

Marino Miculan
(Joint work with Davide Grohmann)
University of Udine

Aim of this talk

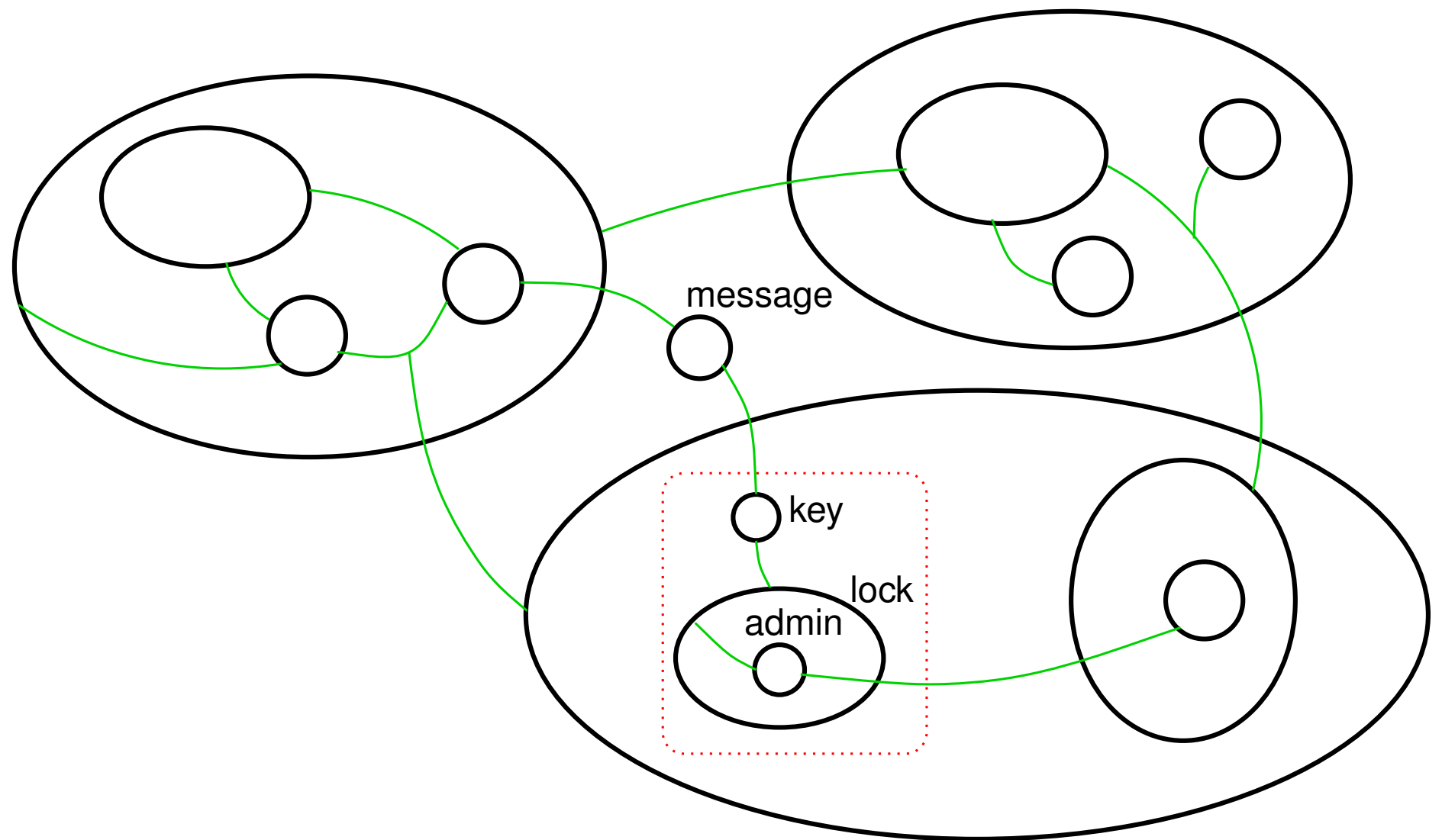
- Advocate the use of **directed bigraphs** as a general metamodel (*a framework*) for calculi with binders, names, resources, ...
- As for any framework, we will describe a formalism (*a metalanguage*) together with an encoding methodology; expected outcomes should include
 - mathematical techniques/constructions for reasoning about calculi
 - general tools/implementations
 - deeper insights and unifying view
- For the gory (categorical) details see papers in MFPS'07, TERMGRAPH'07...

The bigraphical paradigm

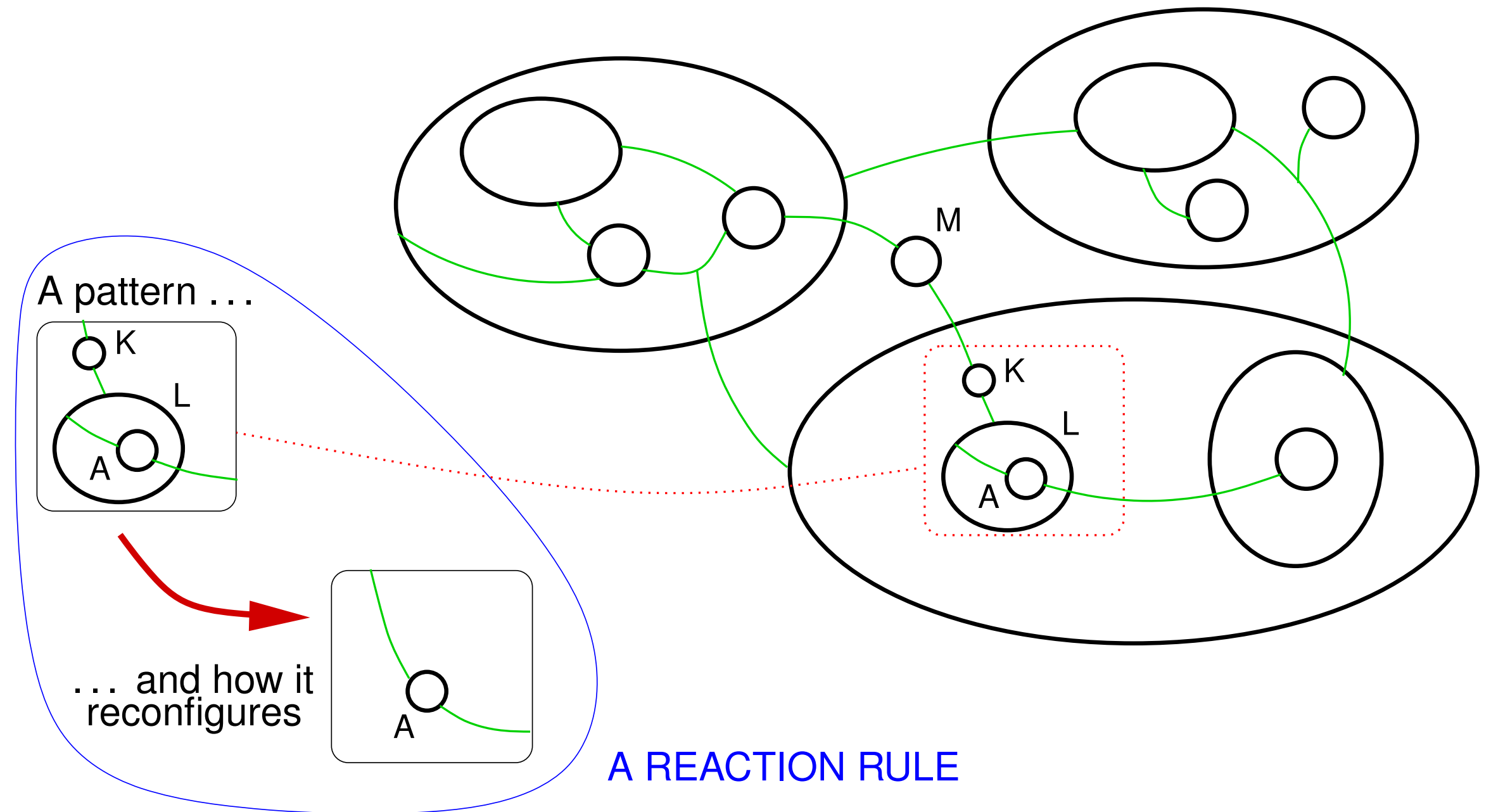
- Introduced by R. Milner (2001) “to express as much as possible of worldwide distributed computing in one mathematical model”
- Original (“pure”) bigraphs focus on **communications** and **locality**, represented by a double graphical structure:
 - a **hierarchical graph**, modeling the **topology** of the net
 - an **hypergraph** representing logical **connections** between nodes
- **Directed bigraphs** are a new version of bigraphs, subsuming pure bigraphs (and some other versions), adding a notion of **resource**, **resource access/request**, and **access control**.
- **Bigraphical Reactive Systems** are rewriting systems over bigraphs, where dynamics is represented by a set of local **graph rewriting rules**

Example of pure bigraphs

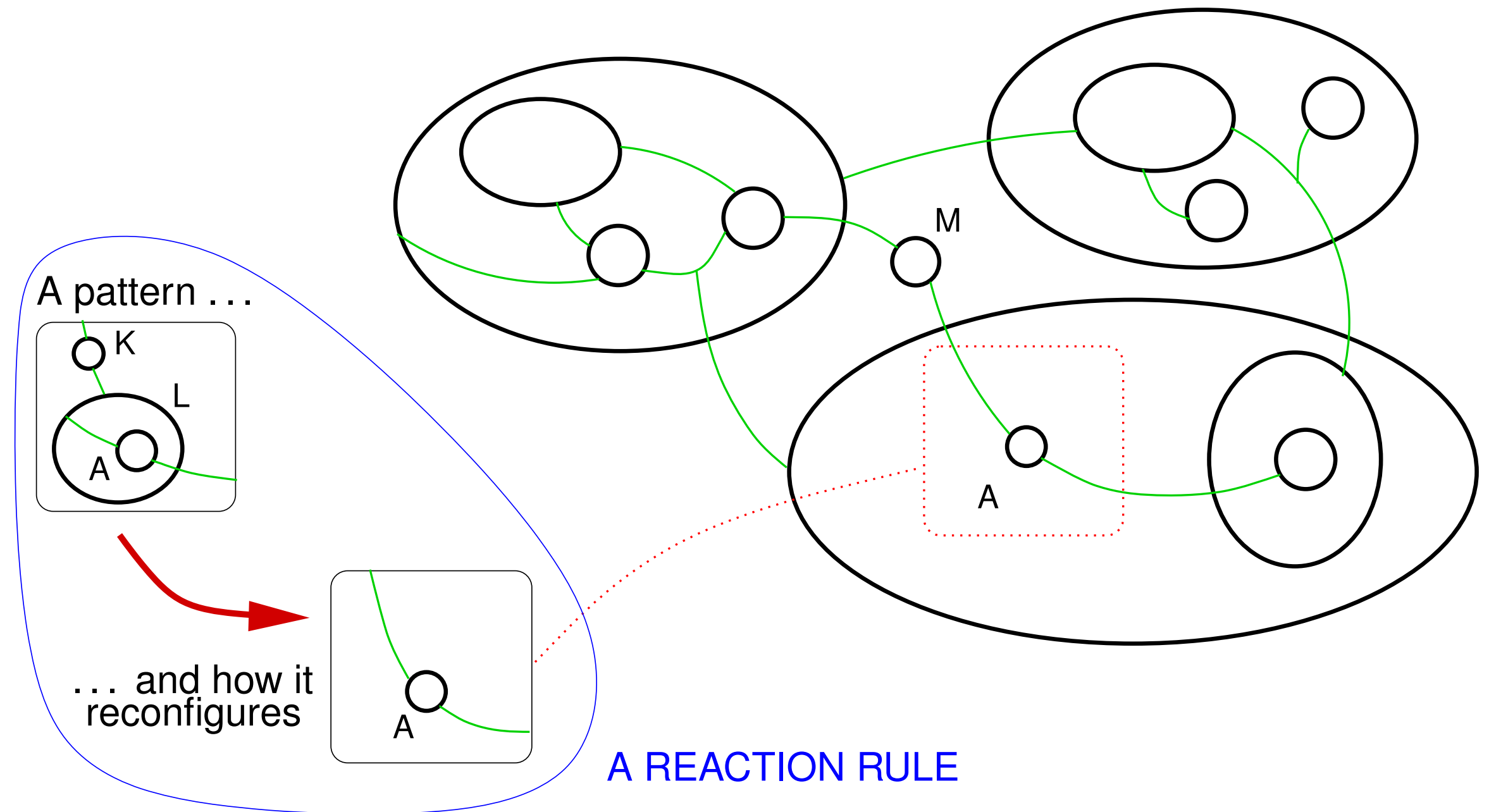
Nodes (here ovals and circles) are the **controls**, which can have **sites** (holes where other bigraphs can be fitted) and **ports**, which can be connected by **links** (the thin lines)



Example of pure bigraphs: dynamics

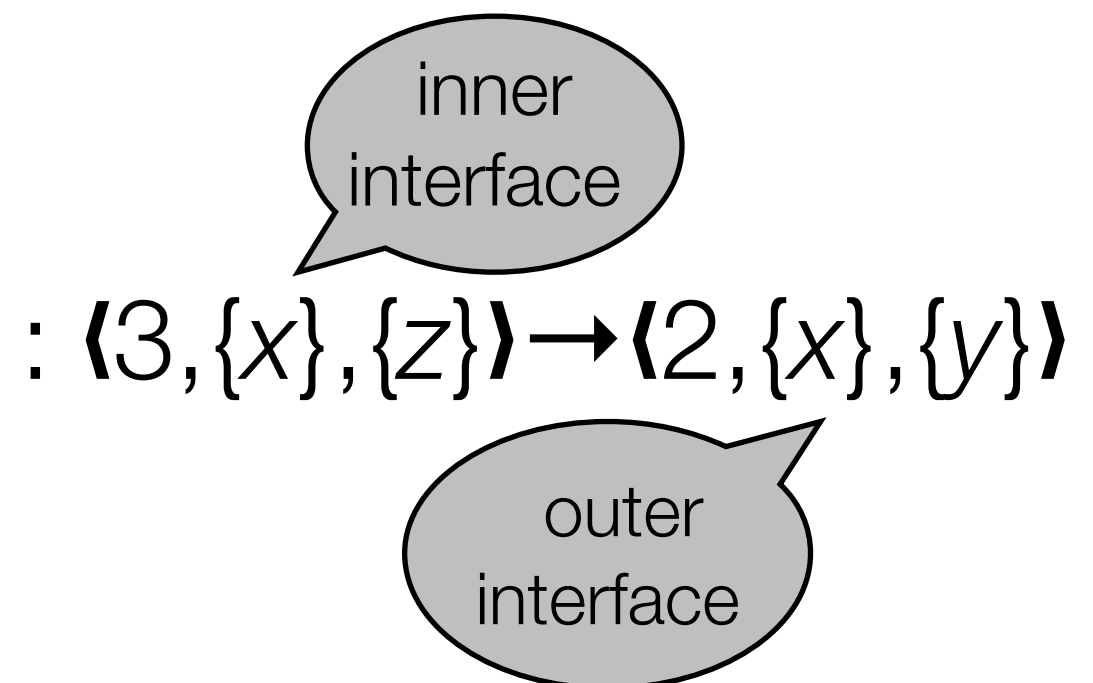
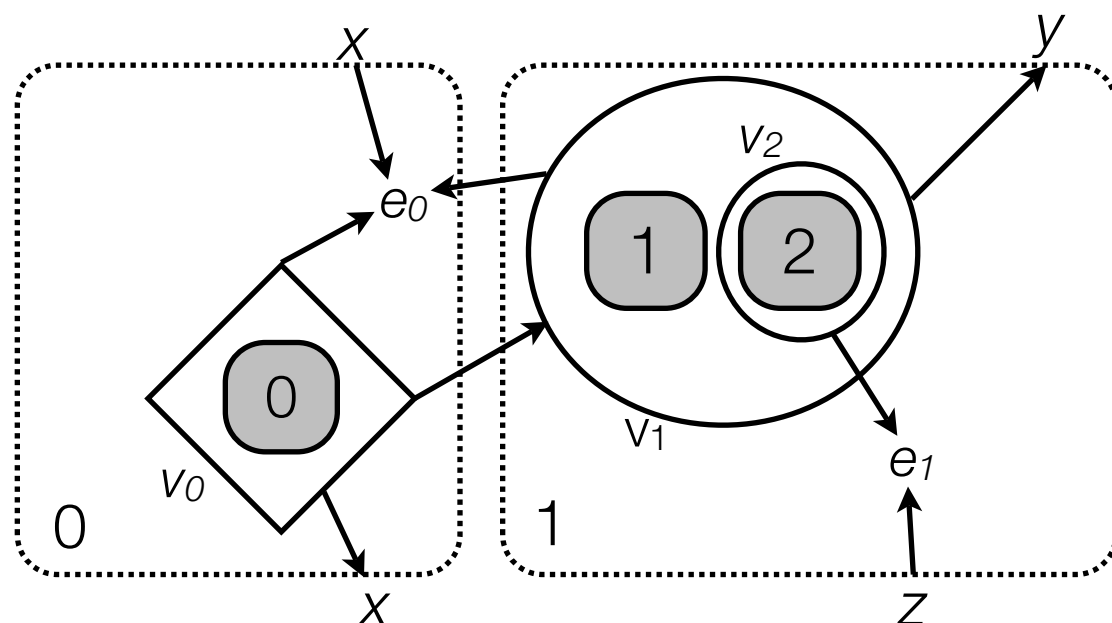


Example of pure bigraphs: dynamics



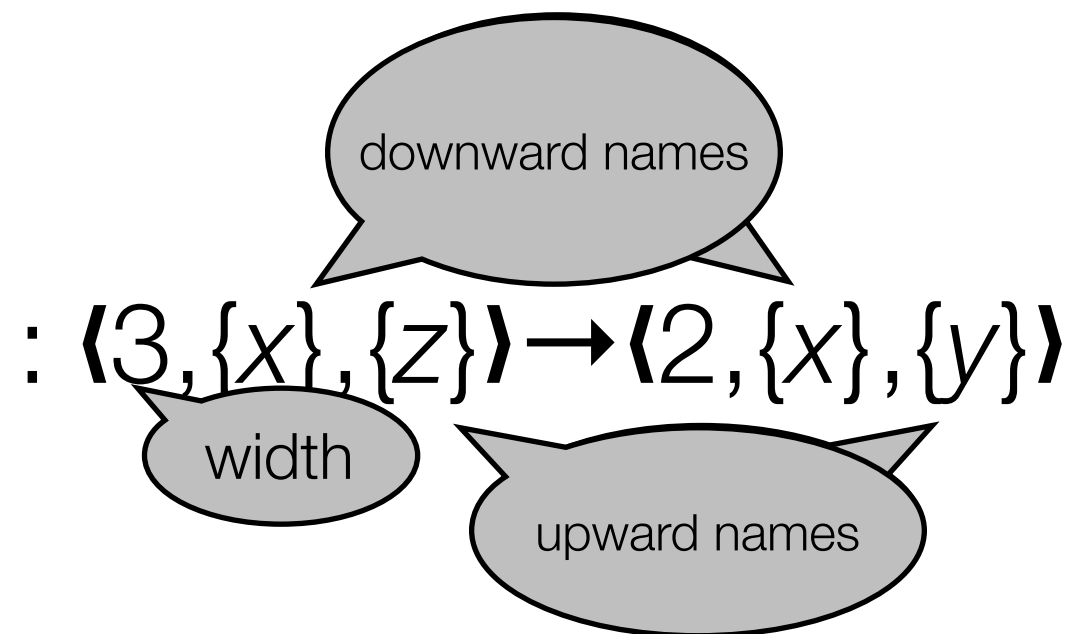
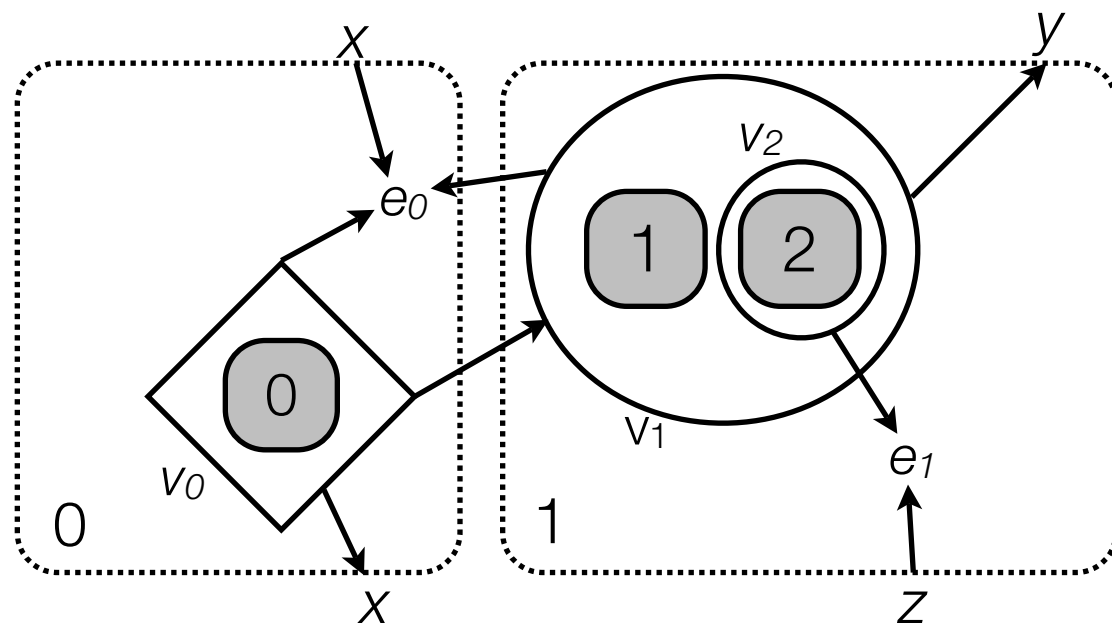
Directed bigraphs: controls, edges, links, names

- **Edges** e_0, e_1, \dots are atoms, and represent abstract **resources**, not subject to the hierarchical topology (they have no “position”)
- **Links** are oriented arcs (no hyper), and represent **resource requests** or **access**
- **Controls** v_0, v_1, \dots (taken from a **signature**) have sites (holes) and **polarized** ports. Controls access to (or request) resources through their outbound ports; inbound port “stop” (or “limit”, “control”, “manage”) other control’s requests
- **Names** x, y, z, \dots are “channels” through which bigraphs can give or request access to resources in the surrounding context (above) or inner bigraphs (below)



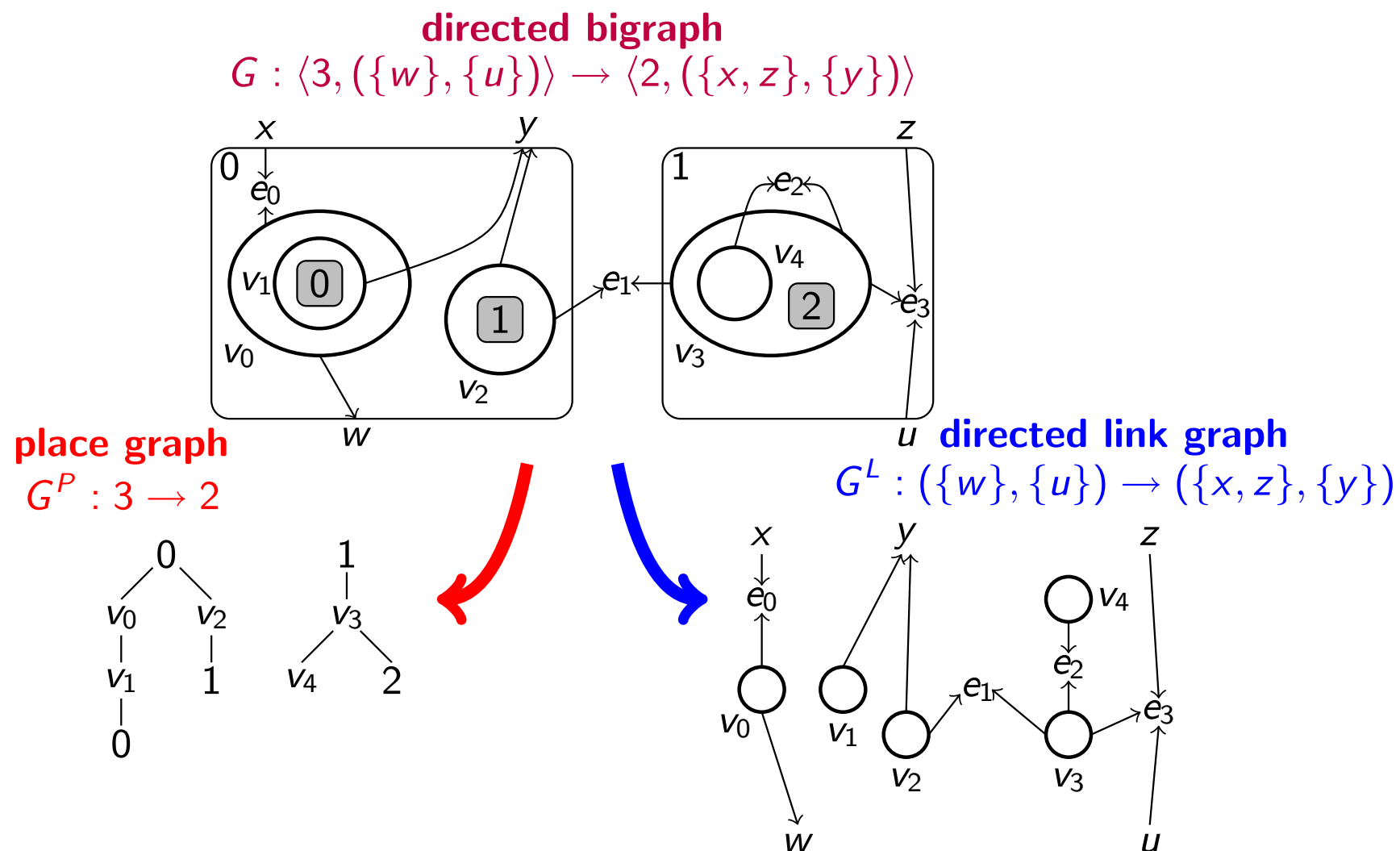
Directed bigraphs: controls, edges, links, names

- **Edges** e_0, e_1, \dots are atoms, and represent abstract **resources**, not subject to the hierarchical topology (they have no “position”)
- **Links** are oriented arcs (no hyper), and represent **resource requests** or **access**
- **Controls** v_0, v_1, \dots (taken from a **signature**) have sites (holes) and **polarized** ports. Controls access to (or request) resources through their outbound ports; inbound port “stop” (or “limit”, “control”, “manage”) other control’s requests
- **Names** x, y, z, \dots are “channels” through which bigraphs can give or request access to resources in the surrounding context (above) or inner bigraphs (below)



A directed bigraph = a place graph + a directed link graph

- Like for pure bigraphs, each directed bigraph can be decomposed into two orthogonal structures, sharing only the nodes (the controls):
 - the **place graph**, a forest of nodes (the same of pure bigraphs)
 - the **directed link graph**, where the resources and resource accesses live



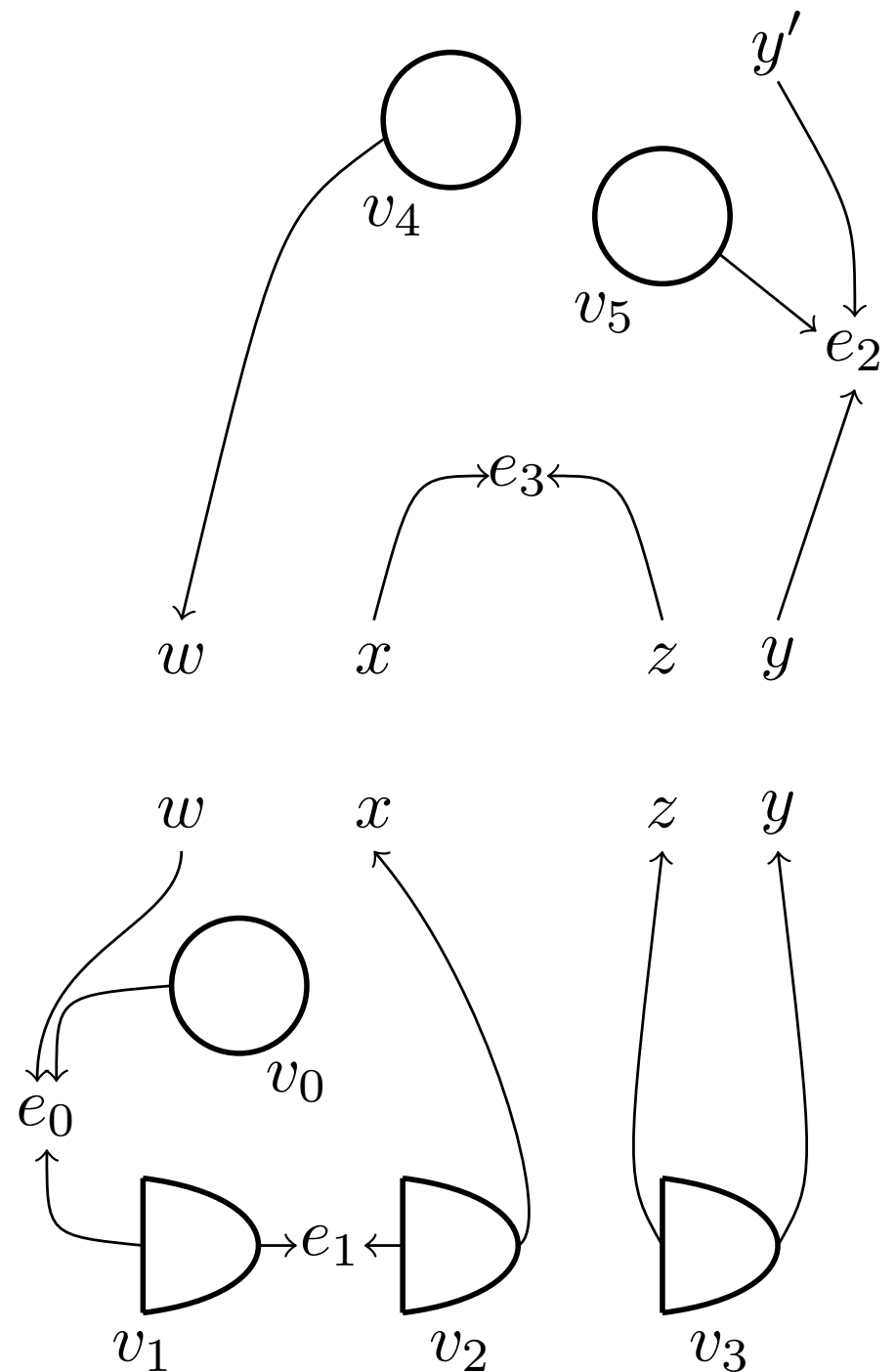
Composition of directed link graphs

- Directed link graphs can be composed when direction of links are respected, and nodes and edges (their **supports**) are disjoint.

In the composition, names in the common interface disappear

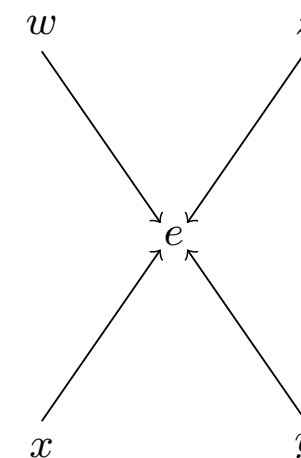
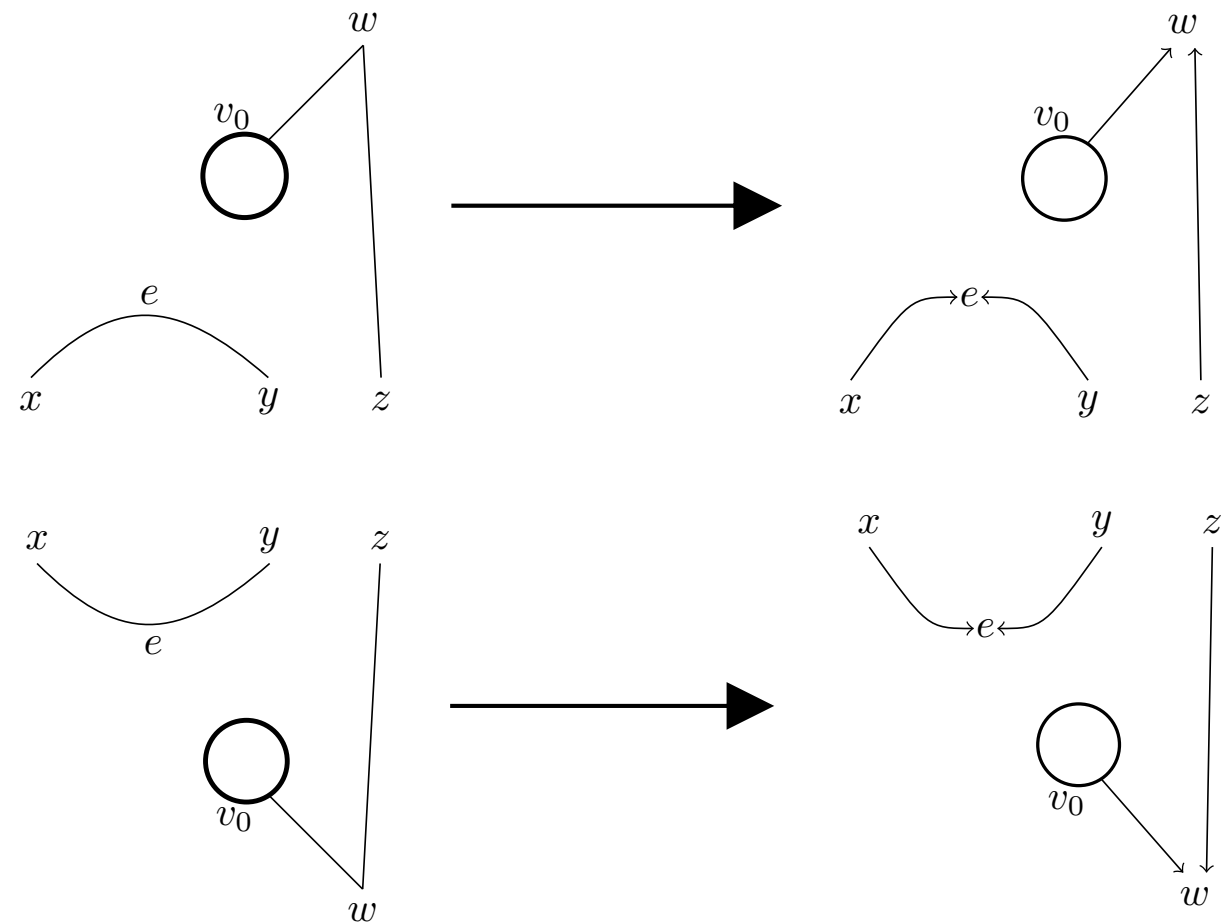
- Polarized interfaces and directed link graphs are the objects and morphisms of the (pre) category **DLG**.

(For you higher category buffs: it's a 2-category where 2-cells are groupoids, or alternatively a $\text{Set}^{\mathbf{B}}$ -enriched category (Fiore))



Embedding previous versions of link graphs

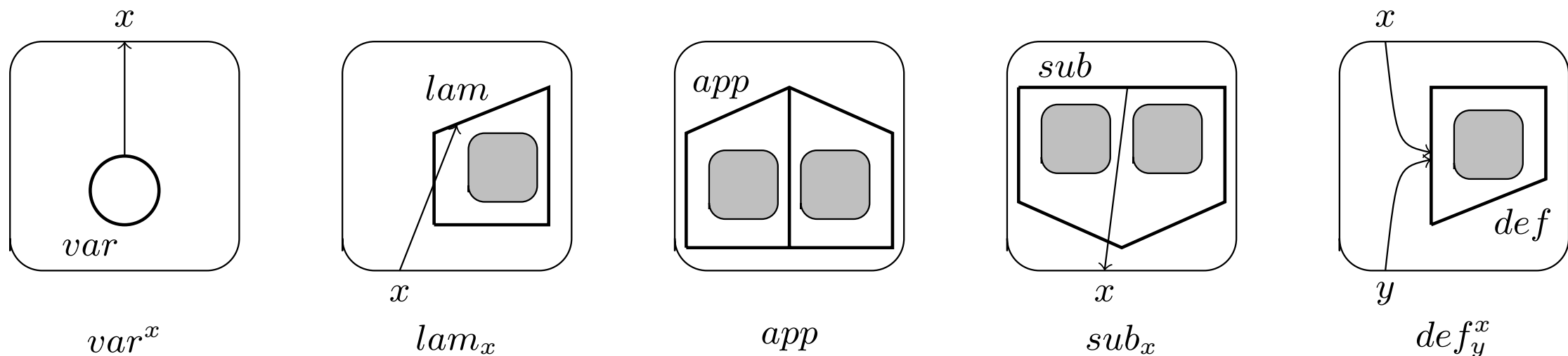
- Easy to define two embedding functors from pure (aka output-linear) link graphs and Sassone-Sobocinski (aka input-linear) link graphs into directed link graphs
- This extends to output-linear / input-linear bigraphs (into directed bigraphs)
- But there are also new link graphs neither input-linear nor output-linear (e.g., the “cross”, which will be useful later)



Enough! Let's play with it.

- Defining a **directed bigraphical reactive system (DBRS)** means giving
 - a signature of controls (yielding a precategory of directed bigraphs)
 - a set of rewriting (“reaction”) rules
- Let's describe the encoding methodology by means of some examples:
 - λ -calculus (for a change!)
 - ν -calculus
 - Fusion calculus
 - (for the π -calculus, translate Milner's encoding with the previous functors)

The λ -calculus: signature



- lam , def , sub are *passive* (rewriting cannot occur inside their holes)
- Extra constructors sub and def needed for implementing “single step substitutions” (will be used in reaction rules)
- a λ -term M is represented by a ground bigraph $\llbracket M \rrbracket : \epsilon \rightarrow \langle 1, \emptyset, FV(M) \rangle$
 - place hierarchy reflects the syntactic tree of M .
 - variables (leaves of the tree) are controls which request something to the environment, crossing the tree structure; these requests can be “stopped” (“bounded”) by lam (hence defining the scopes).
- Notice that no edges are needed here: no need of de-localized resources (pure λ -calculus is just a game of asking and answering to the environment)

The λ -calculus: signature

- Put formally (using some operators from the algebraic characterization of directed bigraphs):

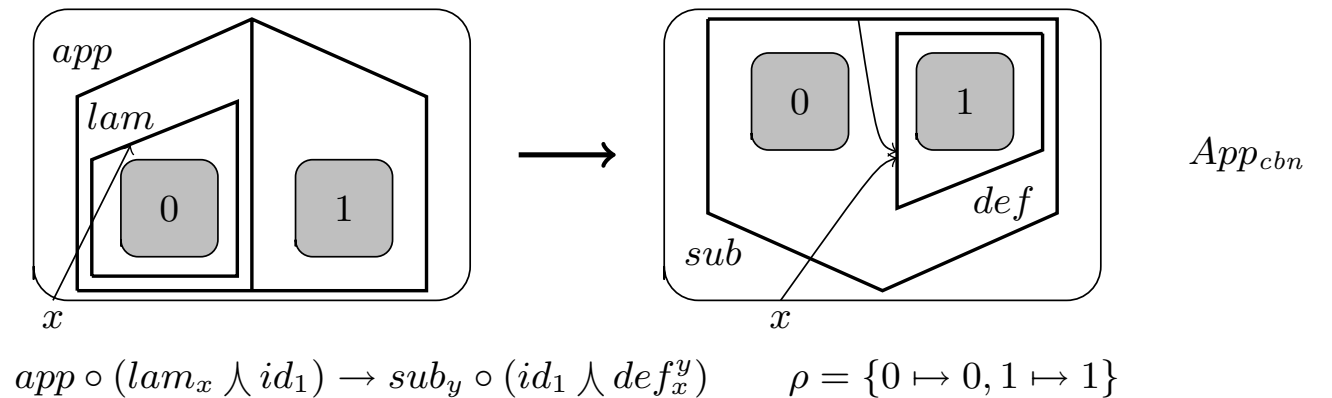
$$\llbracket x \rrbracket = \mathit{var}^x \quad \llbracket \lambda x.M \rrbracket = \mathit{lam}_x \circ (\llbracket M \rrbracket \wedge \Delta^x) \quad \llbracket MN \rrbracket = \mathit{app} \circ (\llbracket M \rrbracket \wedge \llbracket N \rrbracket)$$

- **Proposition:** Let M, N be two λ -terms; then, $M \equiv_\alpha N$ iff $\llbracket M \rrbracket = \llbracket N \rrbracket$
- Let's see the semantics. Recall that

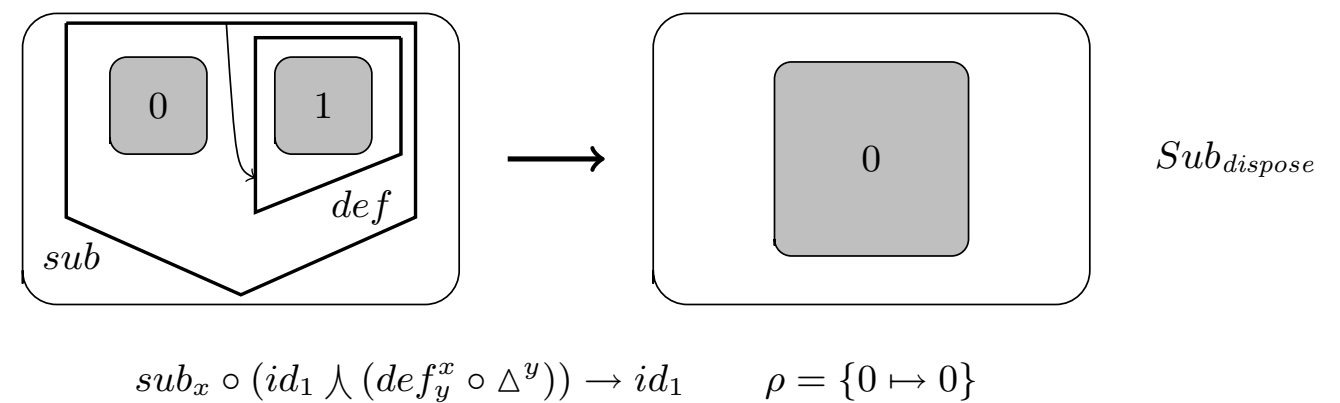
$$(\lambda x.M)N \rightarrow M[N/x] \quad (\beta) \quad \frac{M \rightarrow M'}{MN \rightarrow M'N} \quad \frac{N \rightarrow N'}{MN \rightarrow MN'}$$

The λ -calculus: call-by-name semantics (other variants also possible)

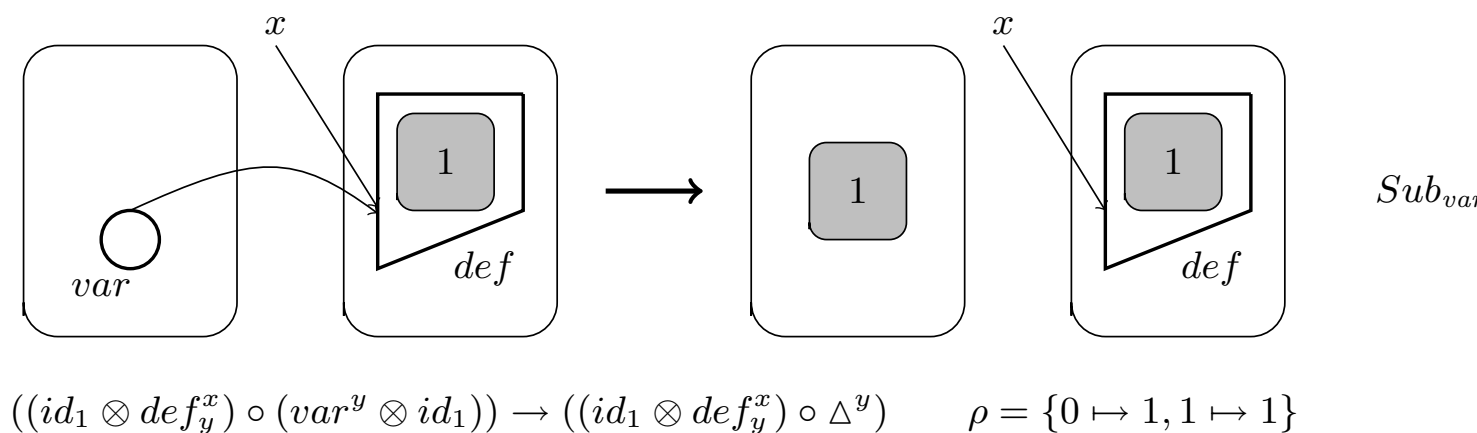
- the *app-lam* redex is rewritten into a new term, where x is bound by *def*



- useless definitions (i.e., the x above is idle) are disposed



- leaves linked to a *def* are replaced with the corresponding term



- Proposition:**

If $M \rightarrow M'$ then $\llbracket M \rrbracket \rightarrow^* \llbracket M' \rrbracket$;
If $\llbracket M \rrbracket \rightarrow^* \llbracket M' \rrbracket$ then $M \rightarrow^* M'$.

The ν -calculus

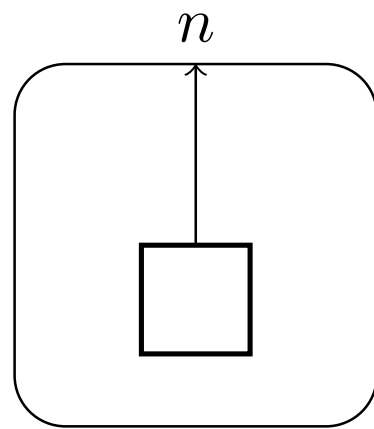
- Pitts-Stark's ν -calculus is a call-by-value lazy λ -calculus extended with primitives for creating and comparing atomic names

$M ::=$	x	variable
	$ $	
	n	name
	$ $	
	$true \quad \quad false$	truth values
	$ $	
	$if\ M\ then\ M\ else\ M$	conditional
	$ $	
	$M = M$	compare names
	$ $	
	$\nu n.M$	create new name
	$ $	
	$\lambda x:\sigma.M$	function abstraction
	$ $	
	MM	function application.

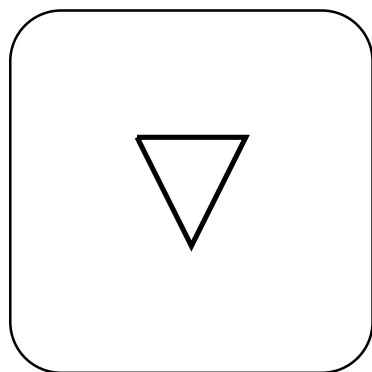
- We will consider an untyped version, and also with *new* instead of ν . (Recall however that $\nu n.M = (\lambda n.M)\ new$)

The ν -calculus: signature

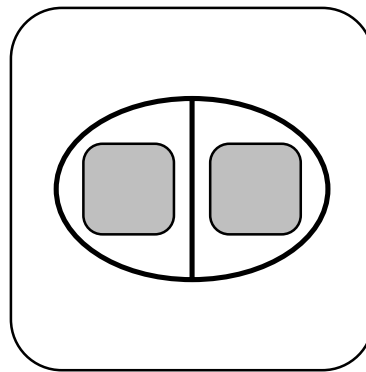
- Extend that of λ -calculus with the following



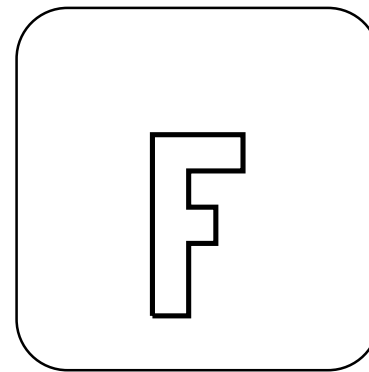
$name^n$



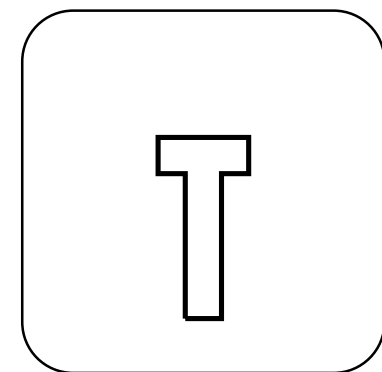
new



$equal$



$false$

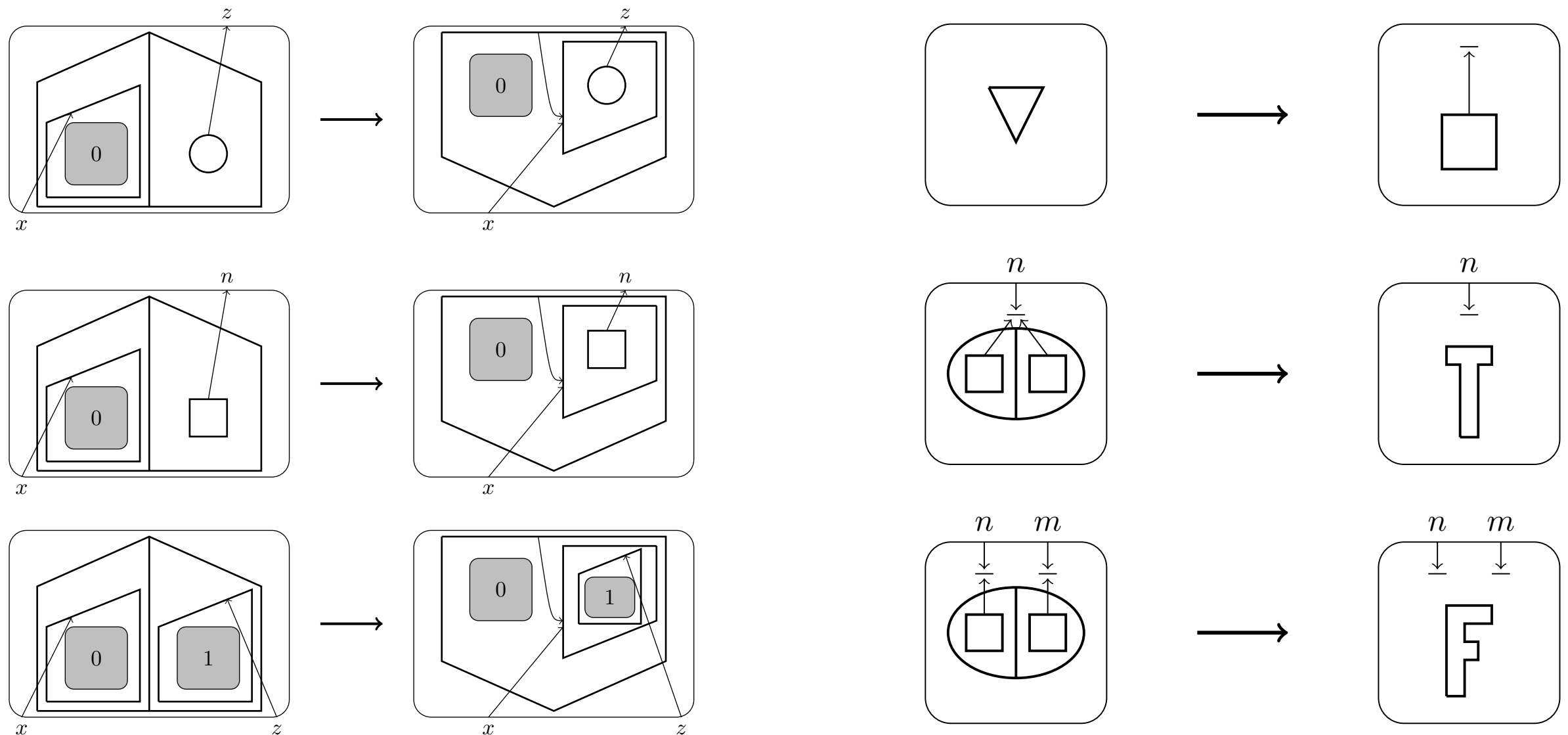


$true$

- and also the translation is extended in the obvious way
- Notice that a ν -calculus name is much like a variable: it's just a different kind of leaf asking for the *real name* it is associated with, to the environment (the *store*).
- *Real* names (those in store) are the resources, and will be represented by edges.
- But a term without free names is translated to a edge-free bigraph, much like λ -calculus. Where do the edges come from?

The ν -calculus: semantics

- The *app-lam* redex is reduced only when the argument is a value
- The edges (the *real* names, the resources) are generated by evaluating *new*



- Faithfulness can be proved, but w.r.t. a small step semantics of untyped ν -calculus (the original one was typed, big-step)

The Fusion calculus: syntax and semantics

- In the syntax, there is only one binder (which defines the scope of variables)

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

$$(x)\mathbf{0} \equiv \mathbf{0} \quad (x)(y)P \equiv (y)(x)P \quad P|(x)Q \equiv (x)(P|Q) \text{ where } x \notin fn(P)$$

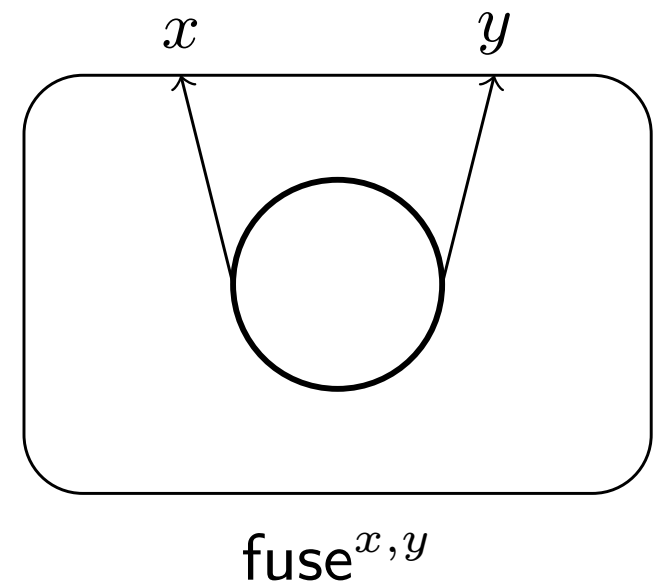
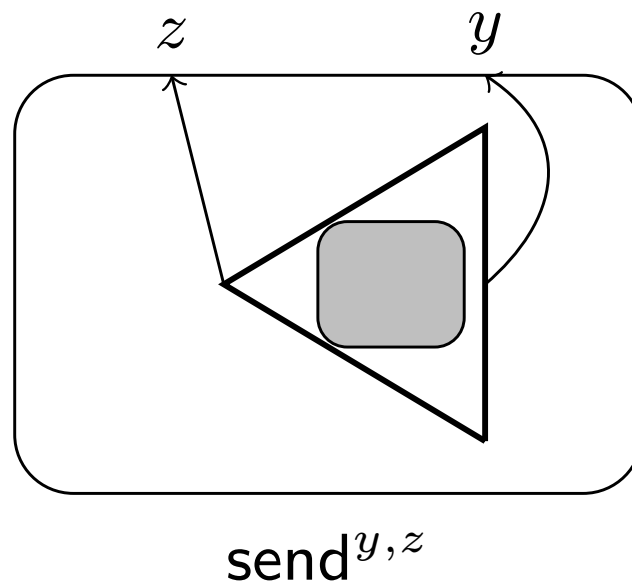
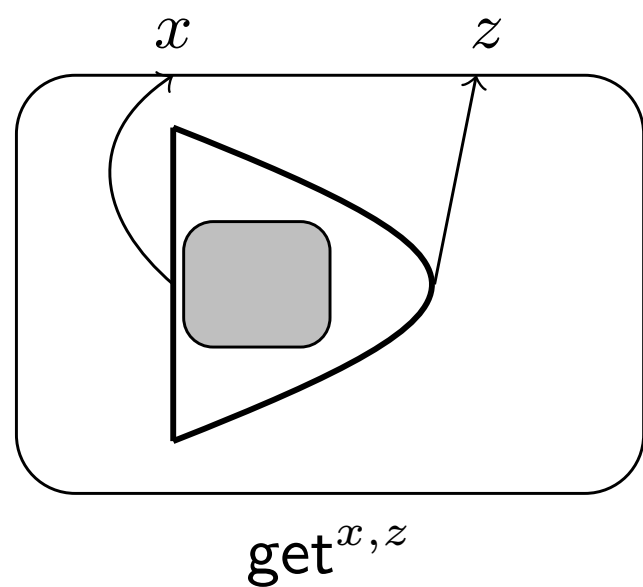
- In the semantics, at communication we observe *fusions* of names, delaying actual substitutions until the *Scope* rule

$$\begin{array}{c}
 Pref \frac{-}{\alpha.P \xrightarrow{\alpha} P} \qquad
 Par \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \qquad
 Open \frac{P \xrightarrow{uz} P', u \notin \{z, \bar{z}\}}{(z)P \xrightarrow{(z)uz} P'} \\
 \\
 Scope \frac{P \xrightarrow{\{x=y\}} P'}{(y)P \xrightarrow{1} P'\{x/y\}} \qquad
 Pass \frac{P \xrightarrow{\alpha} P', x \notin n(\alpha)}{(x)P \xrightarrow{\alpha} (x)P'} \qquad
 Com \frac{P \xrightarrow{ux} P', Q \xrightarrow{\bar{u}y} Q'}{P|Q \xrightarrow{\{x=y\}} P'|Q'}
 \end{array}$$

- For the calculus-zoologist, Fusion appears to be a different animal from both π -calculus and λ -calculus, still it shares something with both.

The Fusion calculus: signature

- Idea of representation: names of Fusion correspond to names of bigraphs, and when two names are fused, they point to the same edges
- Thus, in Fusion the resources are *equivalence classes of names*
- Signature (again, edge-free)



The Fusion calculus: translation

- Translation of a Fusion process P under a given fusion φ is done in two steps. First, we define a ground bigraph $\llbracket P \rrbracket_{FV(P)} : \epsilon \rightarrow \langle 1, \emptyset, FV(P) \rangle$

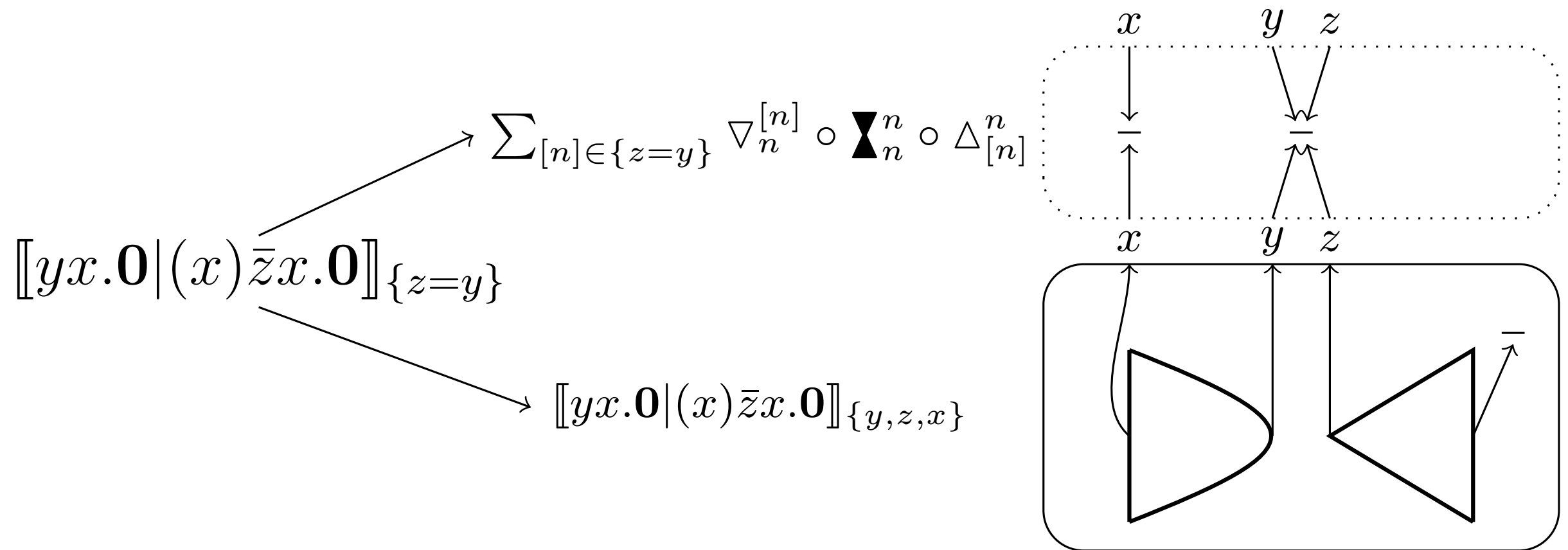
$$\begin{aligned} \llbracket \mathbf{0} \rrbracket_X &= 1 \hat{\wedge} X & \llbracket P|Q \rrbracket_X &= \llbracket P \rrbracket_X \hat{\wedge} \llbracket Q \rrbracket_X & \llbracket (x)P \rrbracket_X &= \blacktriangle^x \circ \llbracket P \rrbracket_{X \uplus \{x\}} \\ \llbracket zx.P \rrbracket_X &= \text{get}^{x,z} \circ \llbracket P \rrbracket_X & \llbracket \bar{z}x.P \rrbracket_X &= \text{send}^{x,z} \circ \llbracket P \rrbracket_X & \text{where } x, z \in X \end{aligned}$$

- Then for each equivalence classes in φ , we add an edge and map to it the corresponding free names of P . Formally

$$\llbracket P \rrbracket_\varphi = \left(\sum_{[n]_\varphi \in \varphi} \nabla_n^{[n]_\varphi} \circ \blacktriangleright_n^n \circ \Delta_{[n]_\varphi}^n \right) \circ \left(\llbracket P \rrbracket_{fn(P)} \otimes \sum_{m \in Y \setminus fn(P)} \Delta^m \right)$$

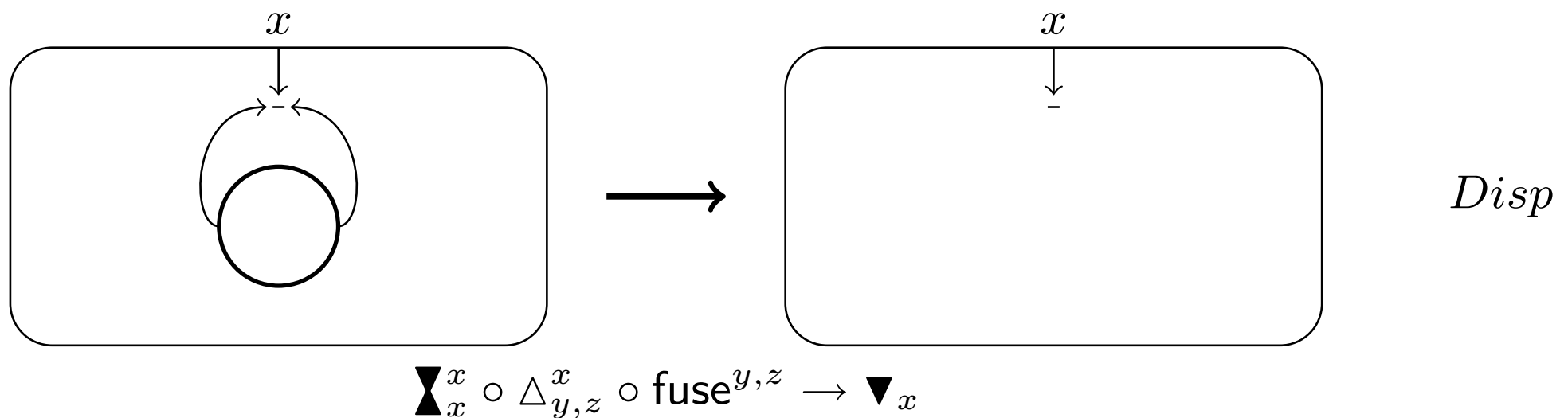
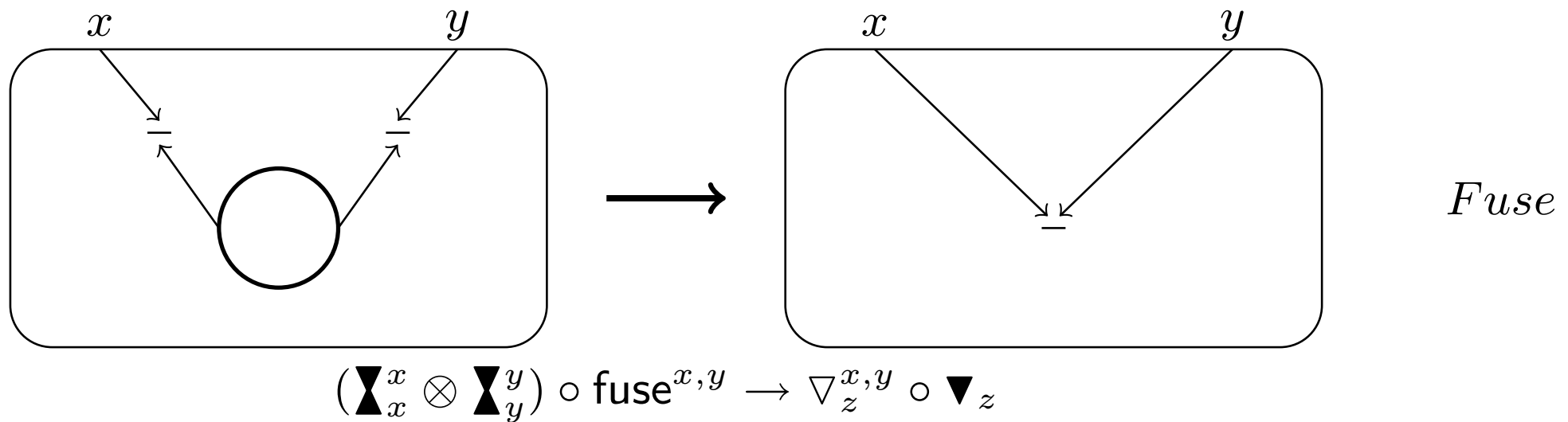
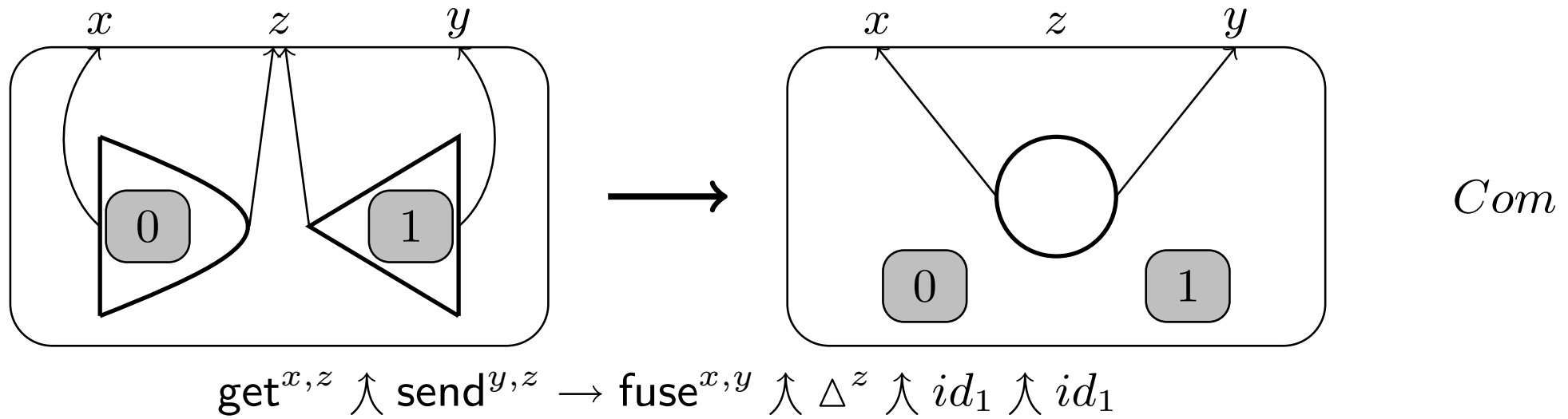
- Good structural properties about the translation still hold

The Fusion calculus: translation example



- A π -calculus beak over a λ -calculus body, but with eggs inside: a platypus!
- Notice that we need the “cross wirings” in the wiring box above; thus this encoding cannot be done in previous versions of Bigraphs.

The Fusion calculus: semantics



“Ok, nice. But why should I care?”

- A metamodel is useful insomuch as it provides general tools, techniques, ...
- For directed bigraphs we have a *general* and *uniform* way for obtaining **compositional strong** and **weak bisimulations** out of the reaction rules
- This boils down to define a **labelled transition system** (LTS) such that
 - its strong bisimilarity is a congruence
 - “silent” transitions can be easily identified, so that a weak bisimilarity can be canonically defined (and has to be a congruence)
- The construction of such labelled transition system is based on the theory of *relative* and *idem-potent pushouts*.

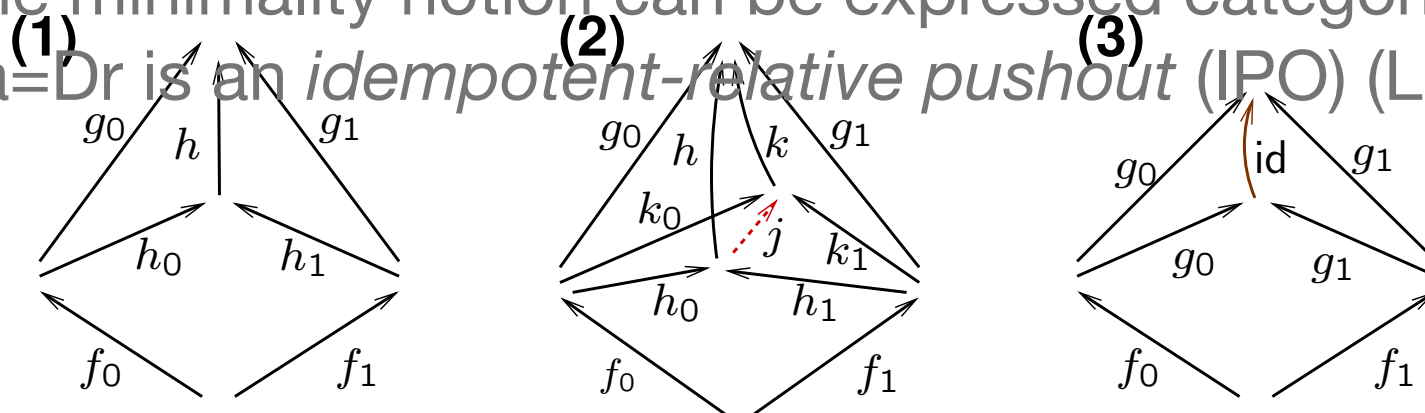
Labeled transition systems defined by *minimal* contexts

- Take as labels the “minimal contexts” which may trigger a reaction

there exists $D(\cdot)$ and a rule $(r, r') \in \mathcal{R}$ s.t. $La = Dr, a' = Dr'$ and L is *minimal*

$$a \xrightarrow{L} a'$$

- The minimality notion can be expressed categorically as the requirement that $La=Dr$ is an *idempotent-relative pushout* (IPO) (Leifer & Milner)



Write \vec{f} for f_0, f_1 .
 Call \vec{g} a *bound* for \vec{f} if $g_0 \circ f_0 = g_1 \circ f_1$

- 1 A *relative bound* (\vec{h}, h) for \vec{f} to \vec{g} .
- 2 A *relative pushout* (RPO) (\vec{h}, h) for \vec{f} to \vec{g} : for any other relative bound (\vec{k}, k) , there is a unique mediator j .
- 3 A *idem pushout* (IPO) \vec{g} for \vec{f} : (\vec{g}, id) is an RPO for \vec{f} to \vec{g} .

Labeled transition systems defined by *minimal* contexts

- Take as labels the “minimal contexts” which may trigger a reaction

there exists $D(\cdot)$ and a rule $(r, r') \in \mathcal{R}$ s.t. $La = Dr, a' = Dr'$ and L is *minimal*

$$a \xrightarrow{L} a'$$

- The minimality notion can be expressed categorically as the requirement that $La=Dr$ is an *idempotent-relative pushout* (IPO) (Leifer & Milner)
- Let us denote by \sim_{IPO} the bisimilarity induced by this LTS
- **Theorem (Leifer-Milner):** \sim_{IPO} is a congruence
- Also a canonical **weak IPO bisimilarity** \approx_{IPO} can be defined: the silent transitions (the “ τ ”) are naturally identified as the empty context (i.e., *id*)

RPOs and IPOs in directed bigraphs

- **Theorem:** Directed bigraphs have RPOs, and there is an algorithm for constructing all possible IPOs of a span of (compatible) bigraphs
- Thus, given a directed bigraphical reactive system, we can derive all transitions of a given agent a by calculating the IPOs of (a,r) for each rule (r,r')
- The resulting LTS gives us the **strong IPO bisimilarity** \sim_{IPO}
- We can also define a **weak bisimilarity** by “skipping” the silent transitions, which in this case are easily identified as the empty contexts (id)
- Hence for each DBRS, we are provided a **strong** and a **weak compositional bisimilarity** for comparing processes in the original calculus

Example: IPO bisimilarity for the λ -calculus

- The strong IPO bisimilarity for λ -calculus is too fine, because it distinguishes for the effective number of substitutions performed during an application
- But the labels of these transitions are *id*, the identity context, which is the silent transition. Thus we can turn attention to the weak bisimilarity:

Theorem: the weak IPO bisimilarity for the λ -calculus corresponds to Abramsky's applicative bisimilarity \sim^B

Proof: Prove that $\sim^B \subseteq \approx_{\text{IPO}}$ and that $\approx_{\text{FT}} \subseteq \sim^0$. Then the result follows from the general $\approx_{\text{IPO}} \subseteq \approx_{\text{FT}}$ and Abramsky's result $\sim^B = \sim^0$

Notice the IPO LTSs is larger than Abramsky's (has strictly more transitions).

What about the other calculi?

- **Fusion calculus:** not clear the relation with *hyperequivalence* (the DBRS uses non-prime agents, that is, directed bigraphs which cannot be mapped back to any process of Fusion)
- **v-calculus:** not clear yet the relation with observational equivalence (we need an observational equivalence theorem, like for applicative bisimulation)
- But even if it does not correspond to any previously known congruences, the **IPO bisimilarity is a useful tool** in any case!
 - In general behavioural congruence are hard to characterize coinductively (often, closures under some contexts/substitutions not easy to deal with; this is the case of hyperequivalence above)

To conclude

- Directed bigraphs are a general framework for calculi with binders, resources, names, variables, ...
- So far, they provide a general and uniform notion of compositional bisimilarity

Future work: a lot! In particular:

- General implementations, along the line of Bigraphical Programming Language (developed by Birkeedal's group at ITU, Copenhagen)
- Spatial/Temporal Logic (like BiLog, [Conforti, Macedonio, Sassone]). Can include the \forall quantifier, like Ambient Logic
- Better theory: some steps in the constructions are unnatural (e.g., quite ad hoc quotients) – it seems that in some cases the enrichment in the category of bigraphs should be something different (Set^I, Schanuel topos...)