

Operational Models for Resource-aware Calculi

Matteo Sammartino

Dipartimento di Informatica, Università di Pisa

CINA kick-off meeting

Models for global computing systems

Explicit representation of resources is essential



Nominal calculi

- Resources represented as **names**
- Operations on resources as primitives of the language

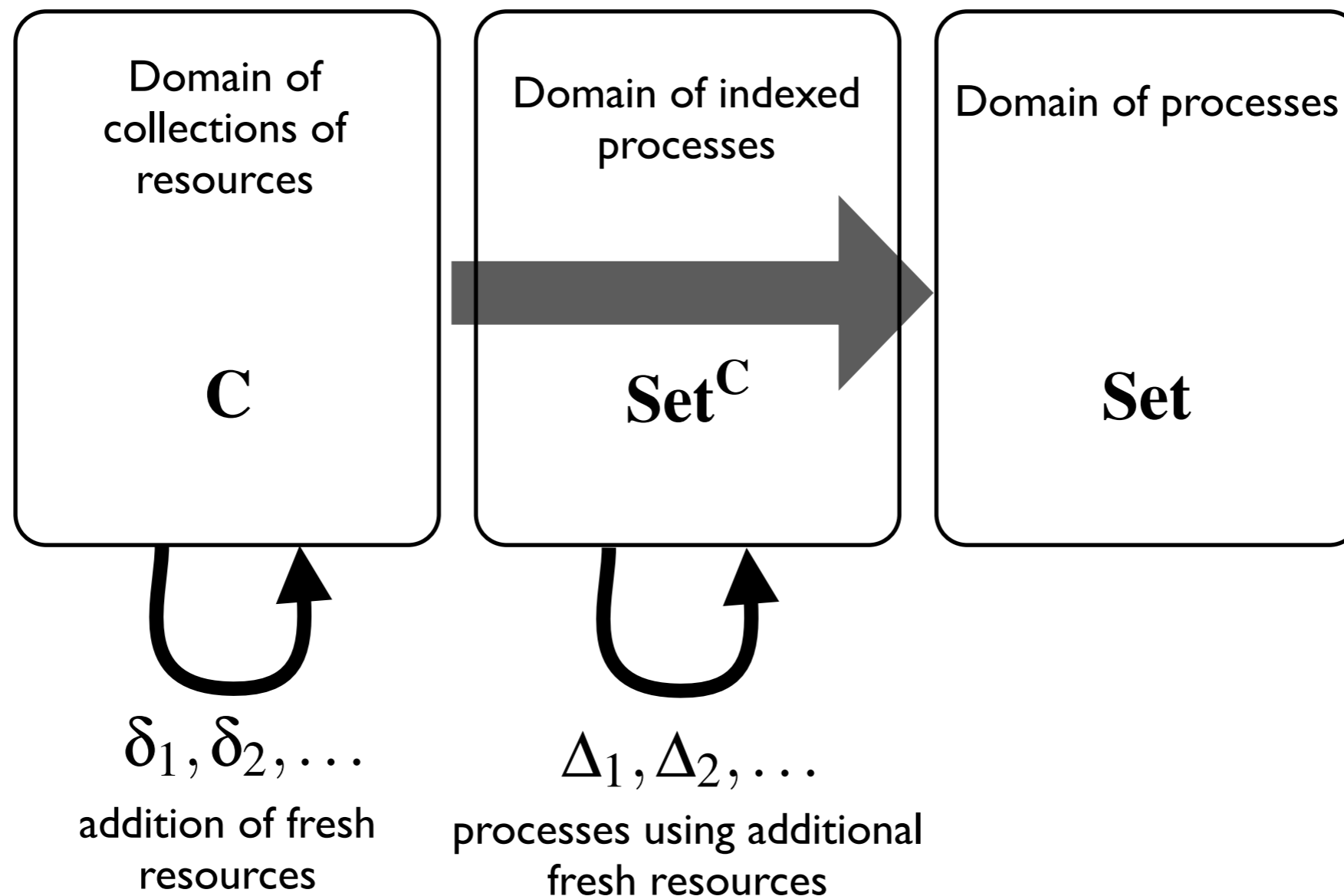
$$(b)\bar{a}b.p \xrightarrow{a(b)} p \quad (b \text{ has been allocated})$$

- Additional expressivity, but
 - freshness must be enforced everywhere
 - infinite-branching, infinite-states LTS

Operational models for nominal calculi

- Well-studied models: **coalgebras**
- How to make coalgebras and coalgebraic bisimulations ‘resource-aware’?
- The idea: **indexing processes with the resources they use**
- Univocal characterization of freshness (processes up to α -conversion)

Indexed processes as presheaves



Generalizing Fiore-Turi's approach

Syntax

- presheaf of processes and renamings freely generated from $\Sigma: \mathbf{Set}^{\mathbf{C}} \rightarrow \mathbf{Set}^{\mathbf{C}}$
- Δ_i model binders

Operational semantics + bisimulations

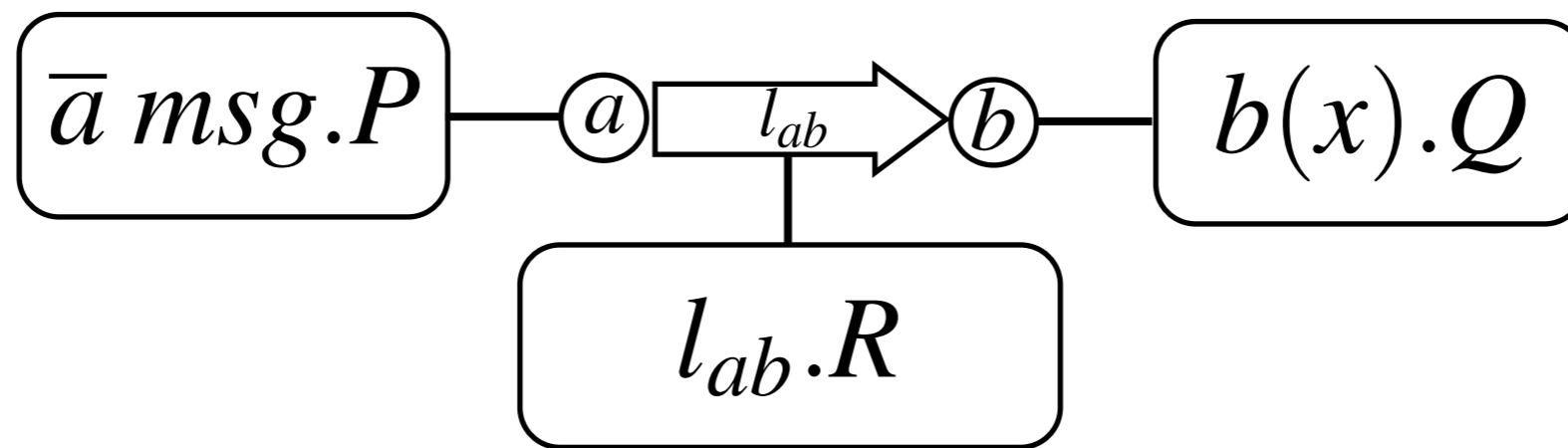
- coalgebra and coalgebraic bisimulations for
$$B: \mathbf{Set}^{\mathbf{C}} \rightarrow \mathbf{Set}^{\mathbf{C}}$$
- Δ_i model allocation along transitions

+ HD-automaton

NCPi: networks as resources

(with Ugo Montanari : MFPS 2012, submitted to TCS)

Resources are **nodes** and **links**



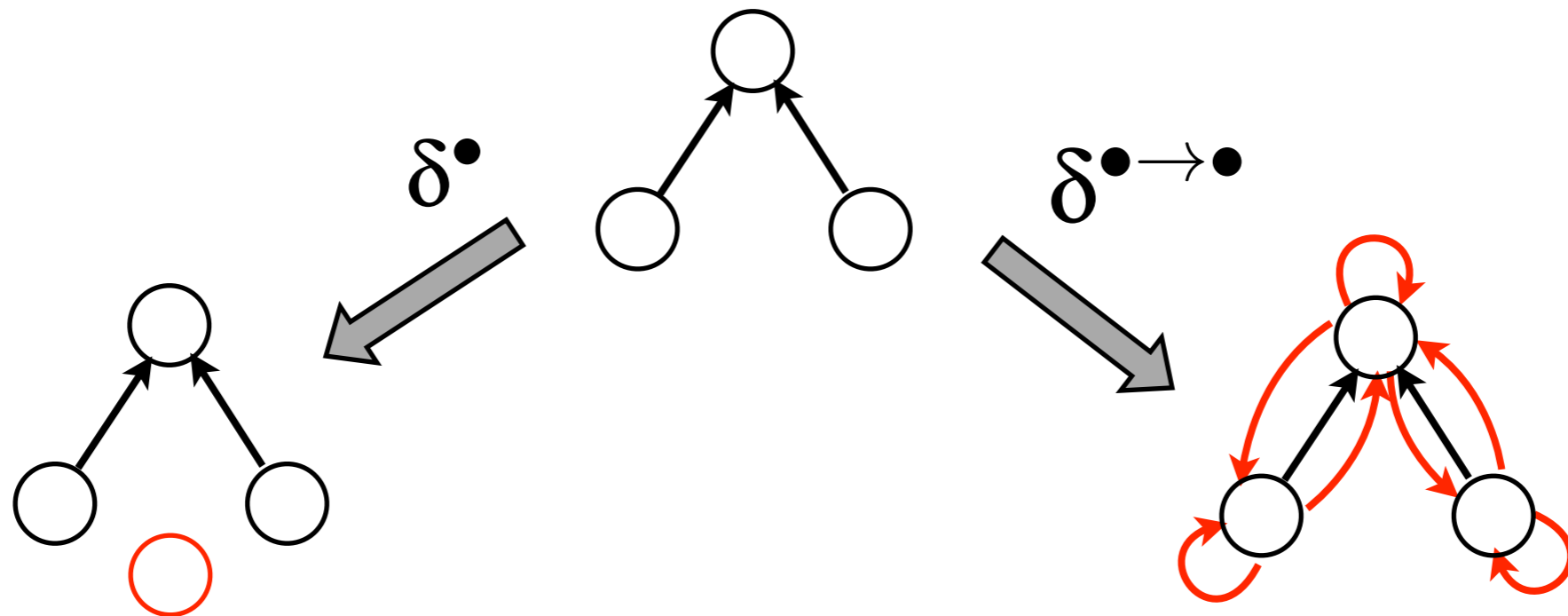
Usual mechanism for resource creation

$$(l_{bc})\bar{a}l_{bc}.P \mid l_{ab}.R \xrightarrow{\bullet; l_{ab}; \bar{b}l_{bc}} P \mid R$$

Non-trivial resources: **names (links)** have other **names (nodes)** as parameters

Category of resources \mathbf{G}

- Objects = finite, directed multigraphs
- Morphisms = graph homomorphisms
- small, with pullbacks and finite colimits



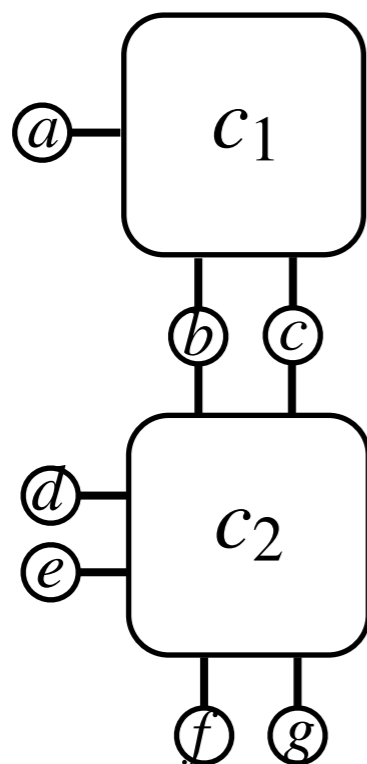
Results

- Coalgebras for operational semantics + observational equivalence
- Coalgebras for saturated semantics + observational congruence
- Hd-automaton for observational equivalence (observational congruence?)

What's next: architectures as resources

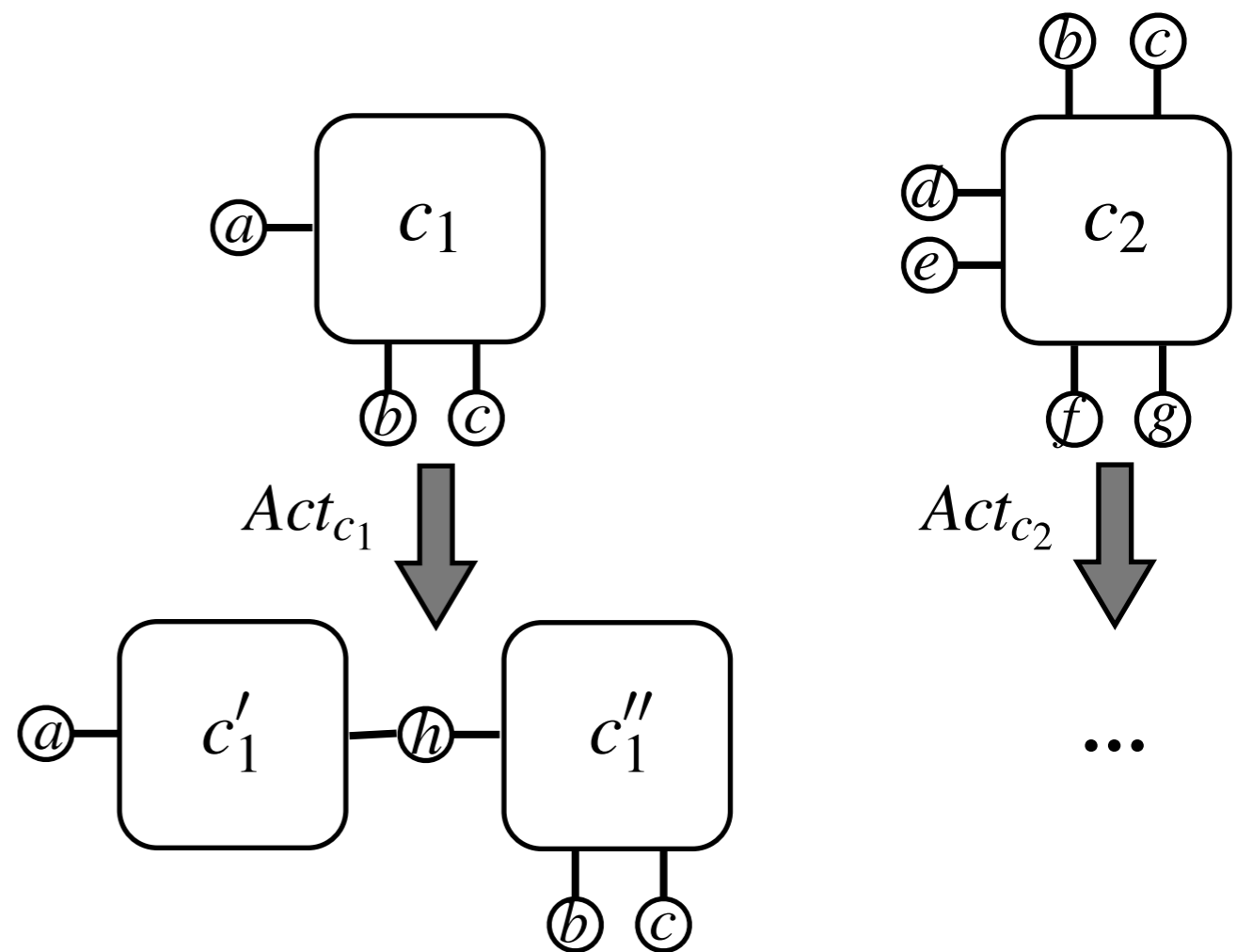
Synchronized Hyperedge Replacement to describe:

Software architecture



rules of synchronization
among components

Detailed design



What's next: architectures as resources

- architectures as a category of hypergraphs
- δ s add new components (hyperedges) and connections (nodes)
- presheaves index systems by their architecture
- Two levels of behavior

