Deriving Bisimulation Congruences with Borrowed Contexts

Barbara König

Universität Duisburg-Essen, Germany

Joint work with Hartmut Ehrig, Paolo Baldan, Filippo Bonchi, Fabio Gadducci, Tobias Heindel, Guilherme Rangel and Ugo Montanari

Motivation: Button-pushing Experiments

A metaphor for interacting with a reactive system and for defining behavioural equivalences:

> "Several testing scenarios, [...], are presented, phrased in terms of 'button pushing experiments' on generative and reactive machines."
>
> (Rob van Glabbeek in "The Linear Time – Branching Time Spectrum I")

Motivation: Button-pushing Experiments

A real life example of a 'button pushing experiment':

Motivation: Button-pushing Experiments

But . . .

What do we do if our system has no buttons?

More precisely:

- We can observe whether a system reacts, i.e., takes a step.
- We can "extend" the system by adding system components or by putting it into a new context or environment.

Motivation: Button-pushing Experiments

Therefore: replace 'button pushing experiments' by 'extension experiments'. Extend the system by putting it into a new environment and see what happens.

Here: we are especially interested in the "minimal" extensions that are needed in order to allow a reaction.

Overview

- Deriving bisimulation congruences
- Adhesive categories and adhesive rewriting systems
- Borrowed contexts and the congruence result
- Relation to reactive systems and relative pushouts (work by Leifer/Milner, Sassone/Sobociński)
- Further issues

Labels in process algebra: yes or no?

In process algebra: buttons usually correspond to transition labels.

Originally: operational semantics of CCS is given by a labelled transition system (LTS)

$$a.P \xrightarrow{a} P$$

But: For more complex process calculi it is usually easier to give (unlabelled) reaction rules

π-calculus: $\quad \overline{a}\langle b\rangle.P \mid a(x).Q \rightarrow P \mid Q\{b/x\}$

Problem: Bisimilarity $\sim$ defined on reaction rules is usually not a congruence (i.e., it is not preserved by contextualization/extension)

$$\overline{a}\langle b\rangle.P \sim \mathbf{0} \qquad \text{but:} \quad \overline{a}\langle b\rangle.P \mid a(x).Q \not\sim \mathbf{0} \mid a(x).Q$$

Deriving labelled transitions

Question: How to derive an LTS from unlabelled reaction rules such that the corresponding bisimilarity is a congruence?

Side condition: The LTS should be as small as possible.

The basic idea: Labels are the "minimal" contexts (borrowed contexts) which a process needs in order to evolve.

Example: $a.P \xrightarrow{-|\bar{a}.Q} P \mid Q$

Problem: How to define the notion of minimal context?

Work by: Milner/Leifer (relative pushouts, RPOs), Sassone/Sobociński (GRPOs), Sewell

Adhesive Rewriting Systems

Our setting: what we mainly need is a way of "gluing" a context to a system (for describing reaction steps as well as for contextualization/extension).
For the congruence result it will be necessary that gluing is "well-behaved".

Idea: describe "gluing" via pushouts in a category.

Extra conditions are necessary to have a category that is suitably well-behaved
$\rightsquigarrow$ adhesive categories

Our theory was originally developed for graph transformation, but thanks to the introduction of adhesive categories by Lack/Sobociński we are now able to state it more generally.

Adhesive Rewriting Systems

Why adhesive categories/adhesive rewriting systems?

- It works!
- It gives us an abstract theory of rewriting and behavioural equivalences.
- Compared to other abstract theories we have the possibility to inspect the "interior" of an object via arrows (very useful for locating (partial) left-hand sides!)

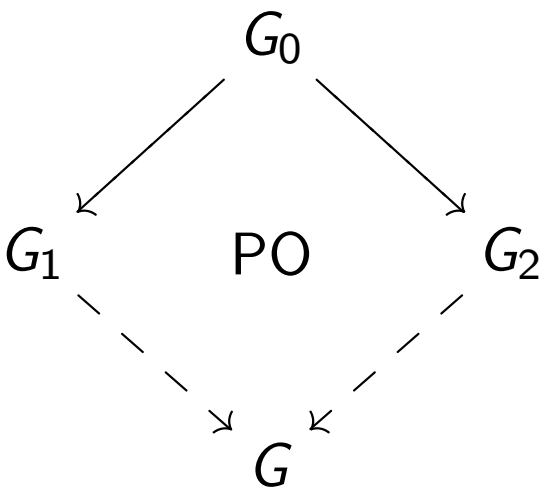
Adhesive Rewriting Systems

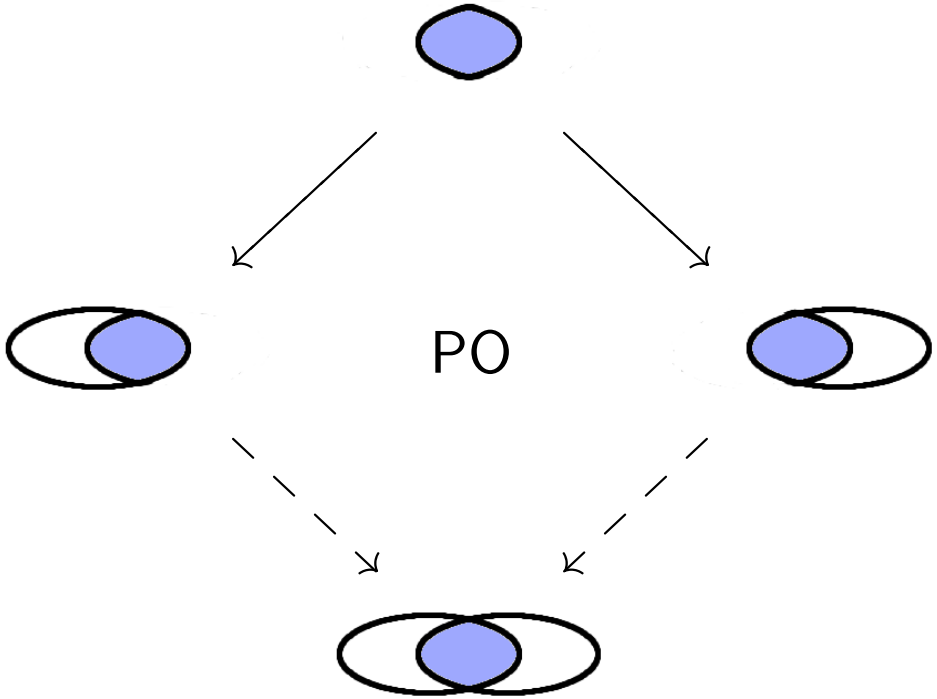
Why adhesive categories/adhesive rewriting systems? (continued)

- Graph transformation is an important example for adhesive rewriting systems.
  - Graph transformation is a natural and intuitive modelling language for concurrent and distributed systems.
  - Graph transformation can also be seen as a possible semantic foundation for process algebras. Label derivation seems to work smoother if processes are represented by graphs (compared to terms).
    $\leadsto$ Milner's bigraphs
- Transfer of concepts from process calculi to graph transformation and vice versa

Categorical operations

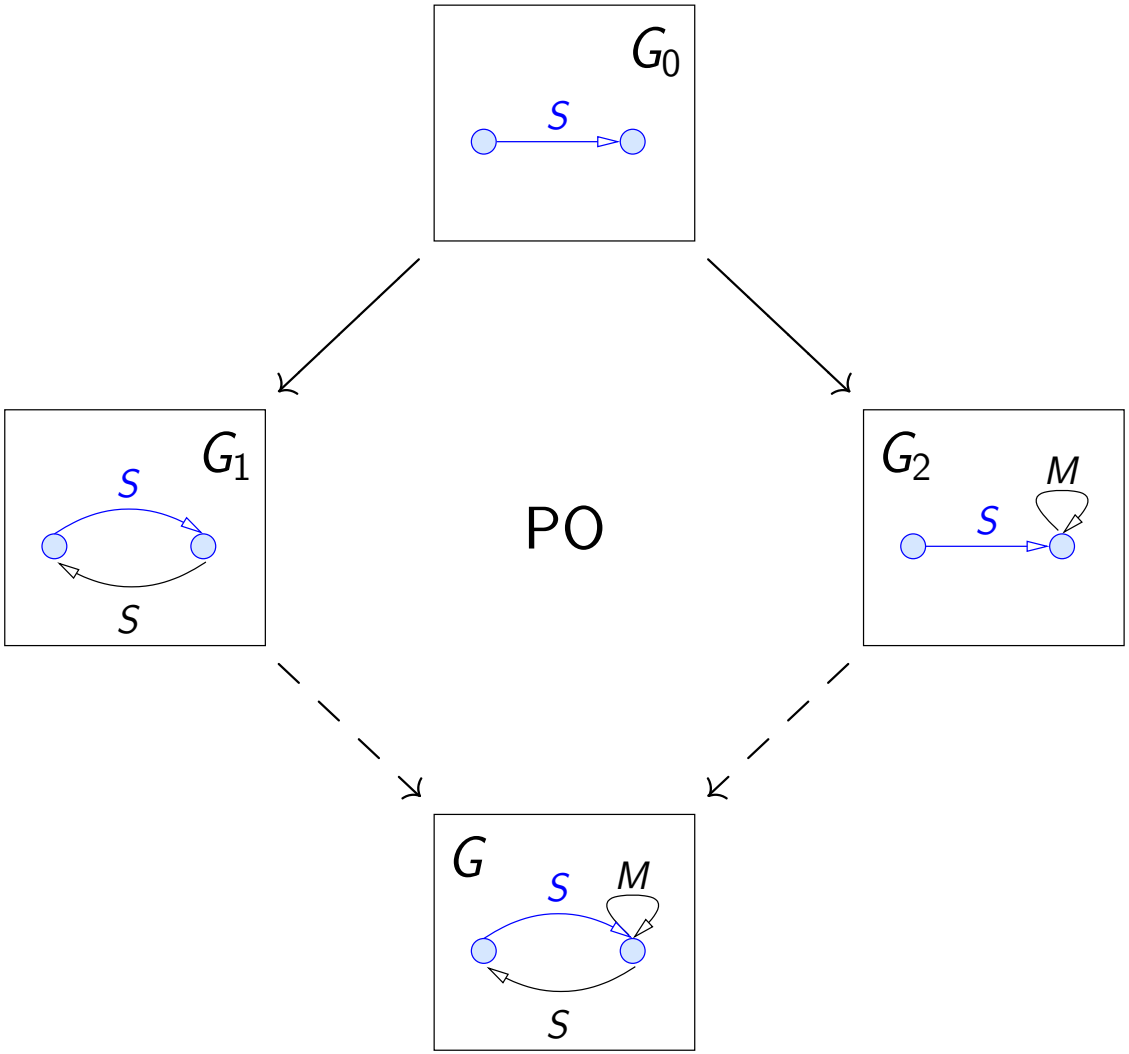
Pushout $\triangleq$ Union/Gluing





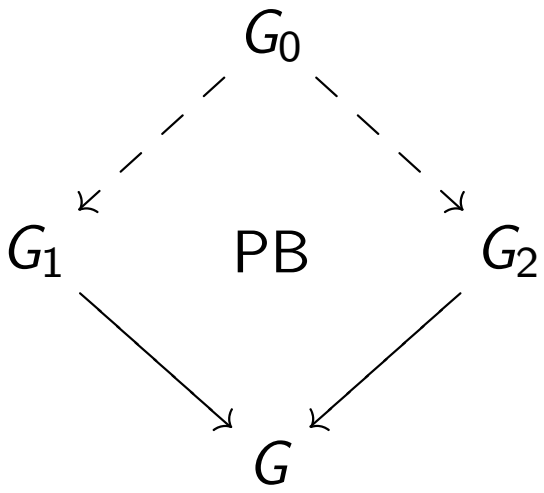
all (graph) morphisms injective/mono

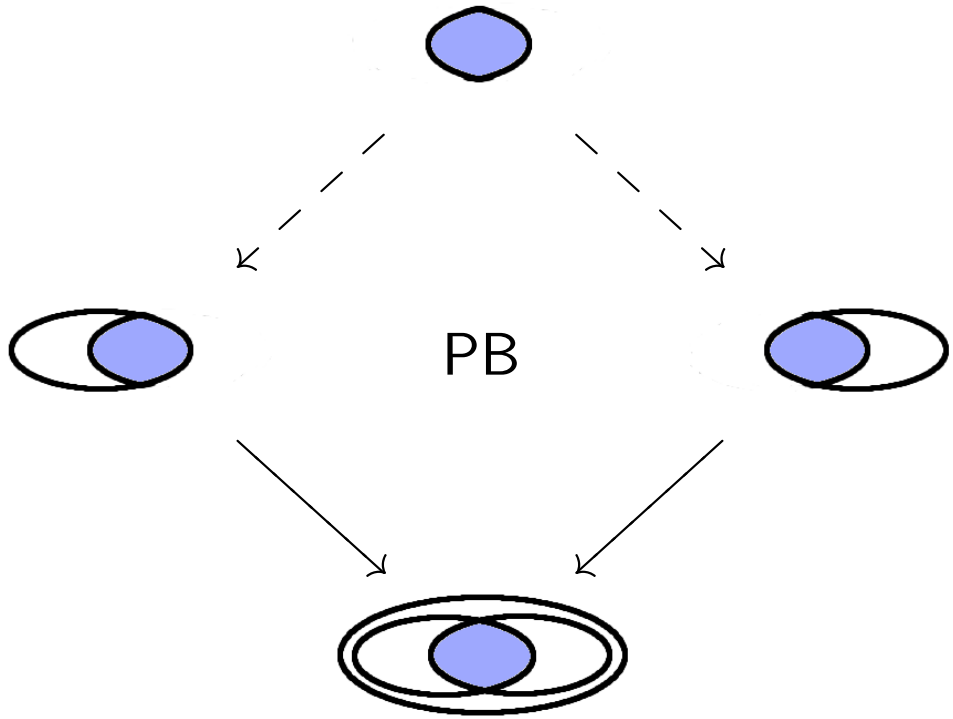
Example (pushout)



Categorical operations

Pullback $\triangleq$ Intersection

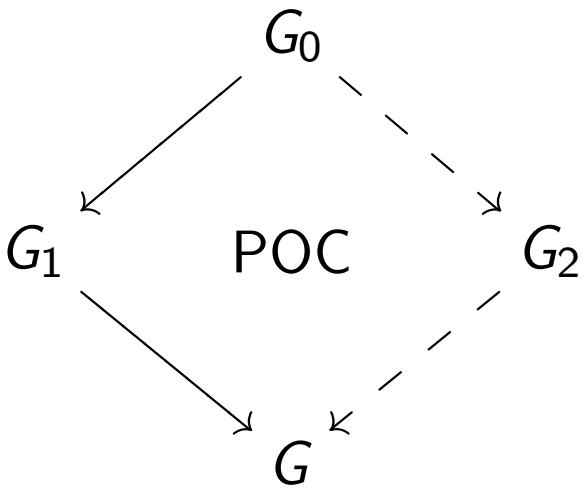


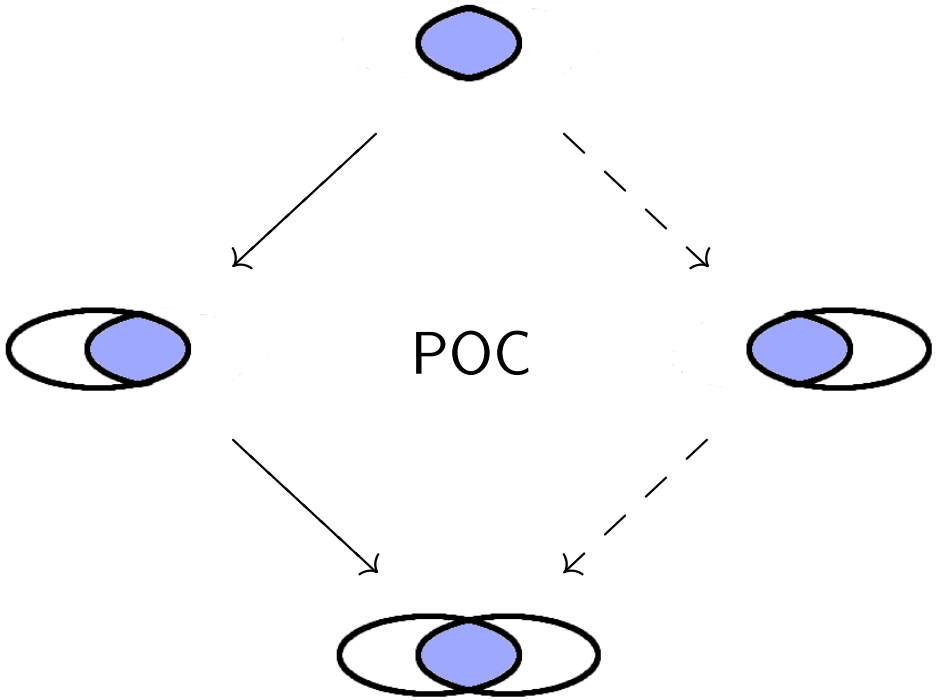


all (graph) morphisms injective/mono

Categorical operations

Pushout complement $\triangleq$ Difference

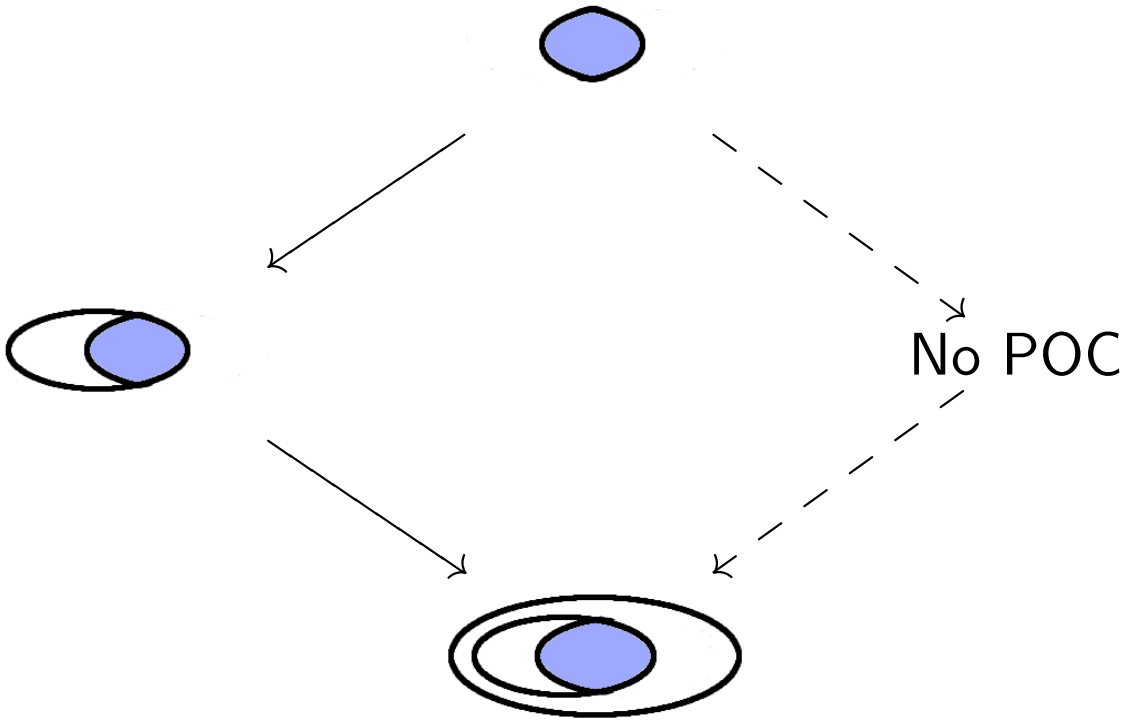




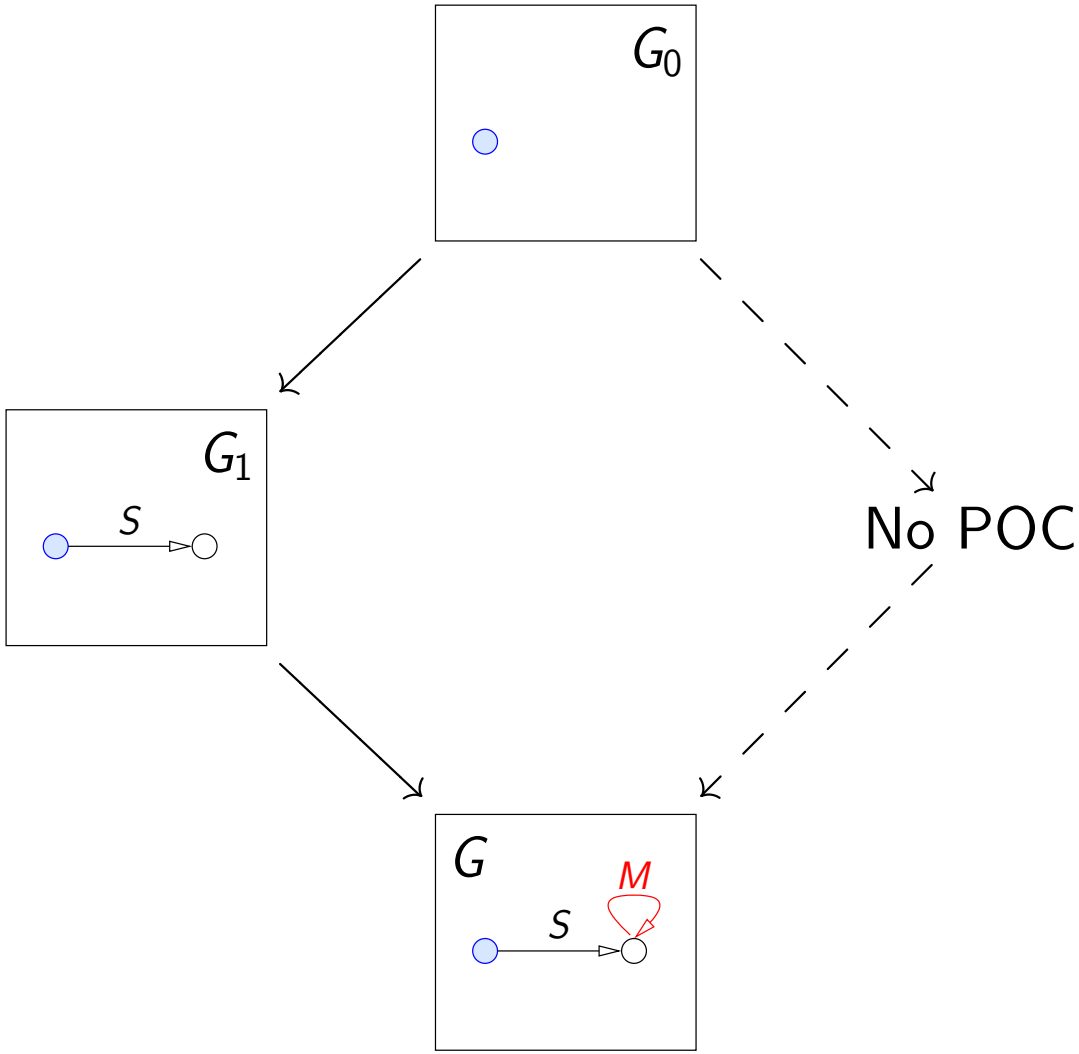
all (graph) morphisms injective/mono

Categorical operations

Difference $\triangleq$ Pushout complement



Example (failed pushout complement)

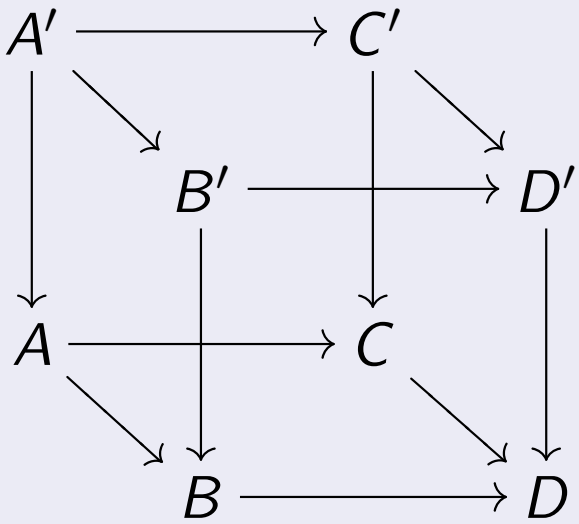


Adhesive Categories (Lack/Sobociński)

An adhesive category has ...

- pushouts along monos
- pullbacks
- pushouts along monos are Van-Kampen squares, i.e., given a cube diagram as shown on the right with:
  - $A \to C$ mono
  - the bottom square a pushout *and*
  - the left and back squares pullbacks,

  we have that the top square is a pushout if and only if the front and right squares are pullbacks.



Adhesive Categories

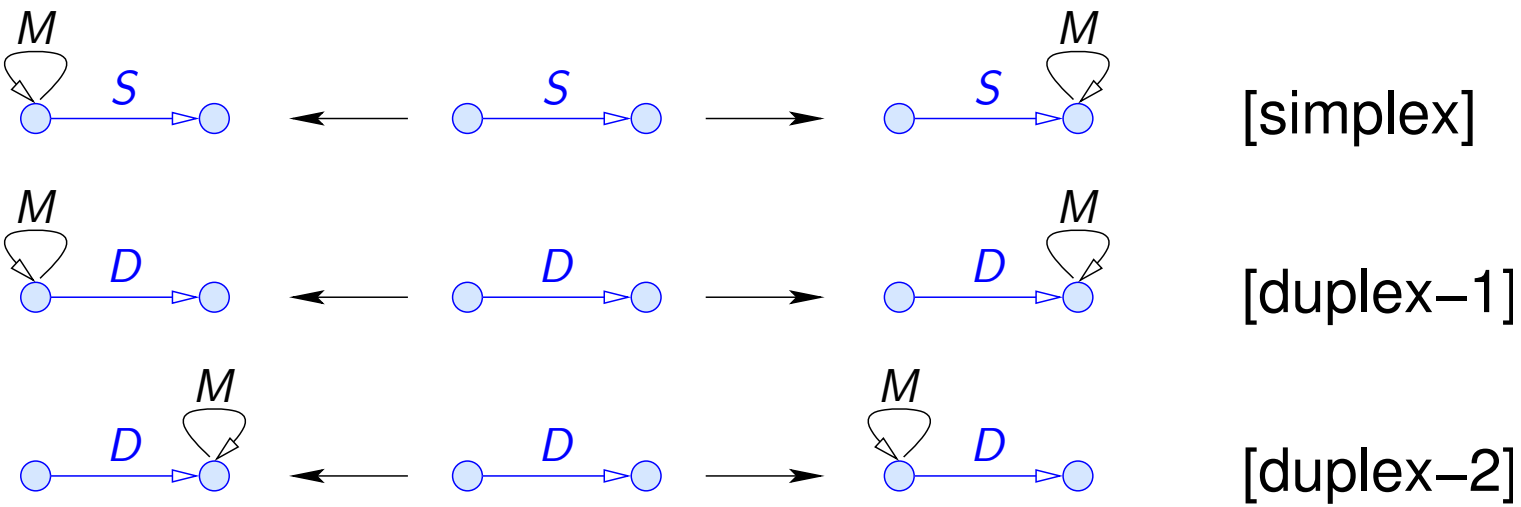
An important property of adhesive categories:

Given a fixed object T, its subobjects form a distributive lattice.

We will exploit this fact for this talk and reason informally via Venn diagrams (= colored blobs), at least in the case of subobjects and monos.

Adhesive Rewriting

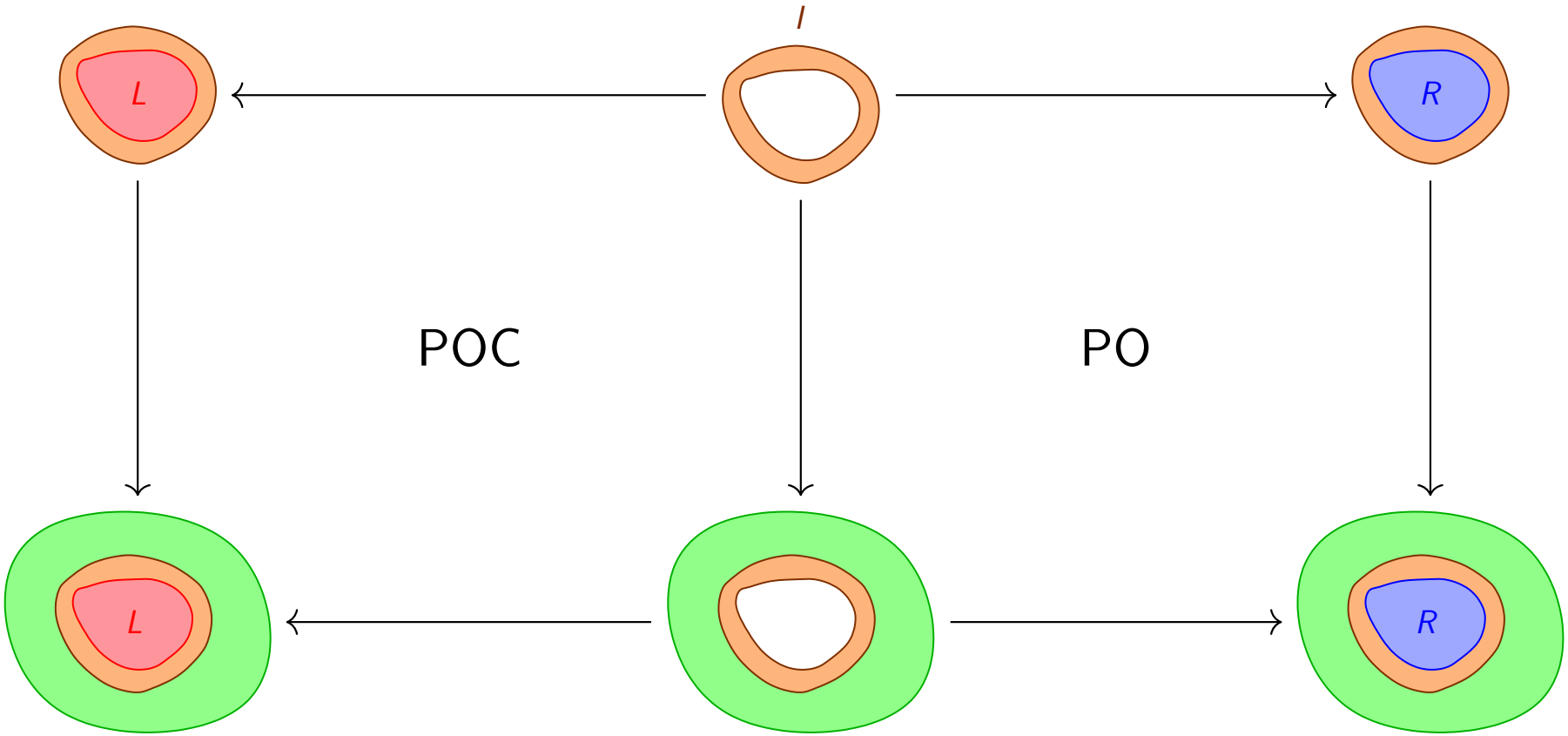
(DPO) Rewriting rules: $L \leftarrow I \rightarrow R$



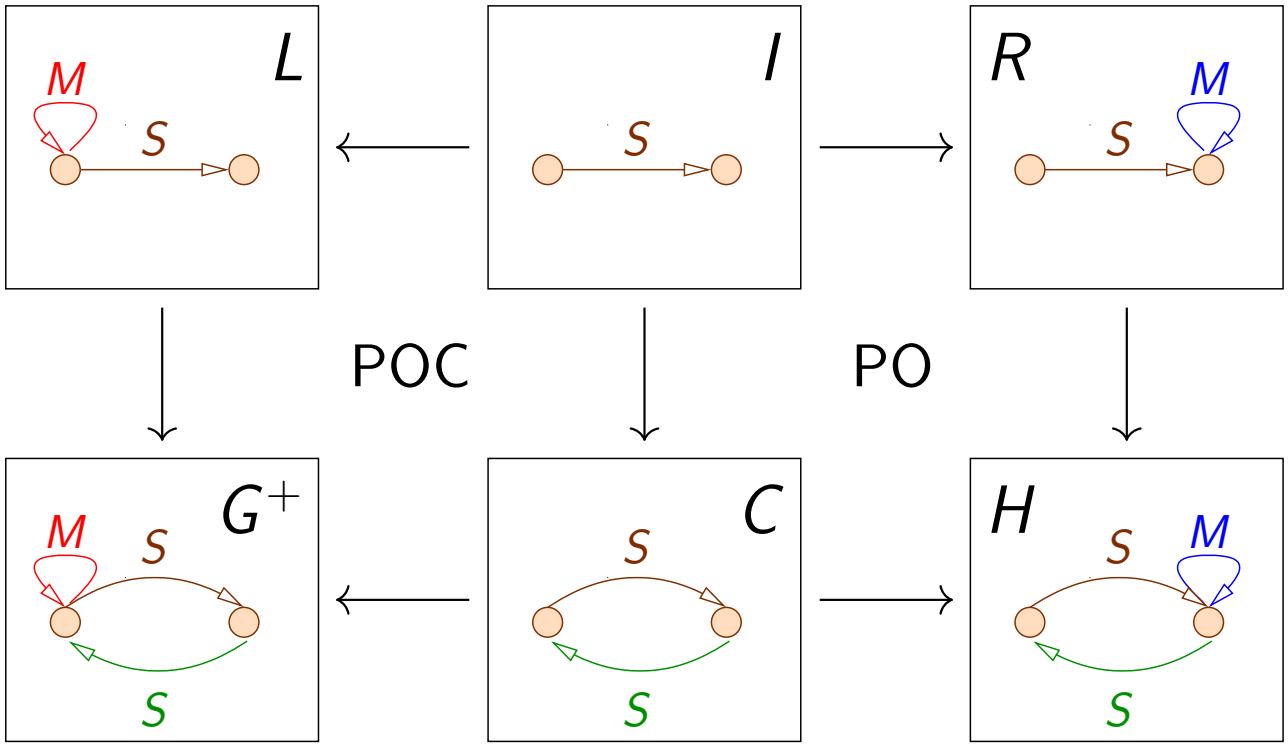
In our case: rule spans and matches consist of monos (extensions possible)

Rewriting in the DPO approach

DPO = Double-pushout (classical rewriting approach for the algebraic theory of graph transformation)



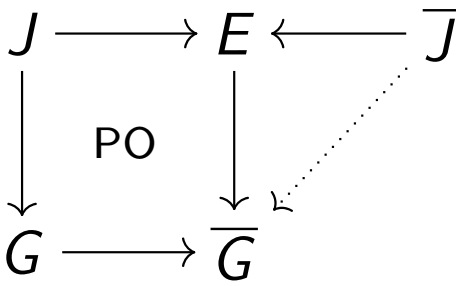
Example (DPO rewriting)

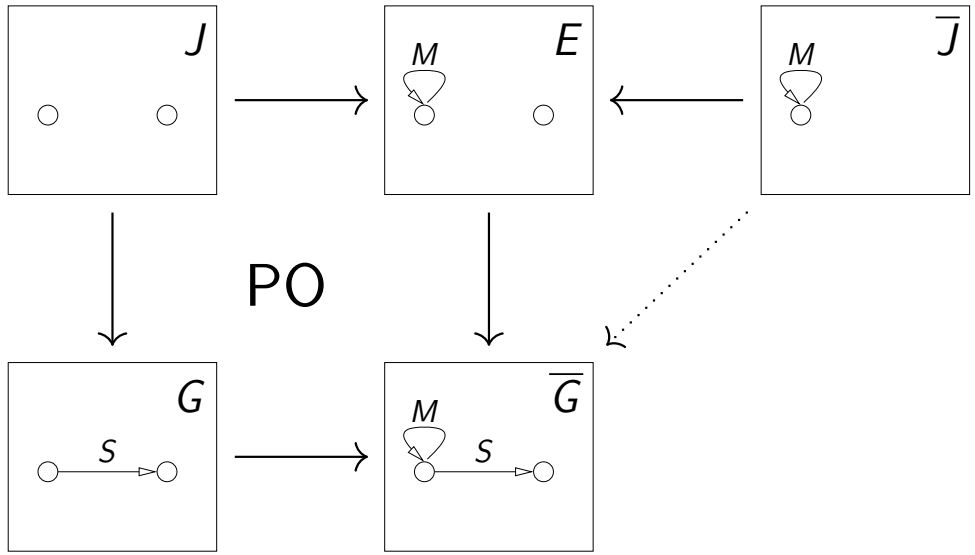


Graphs with interfaces and graph contexts

We consider a graph G with interface J: $J \to G$

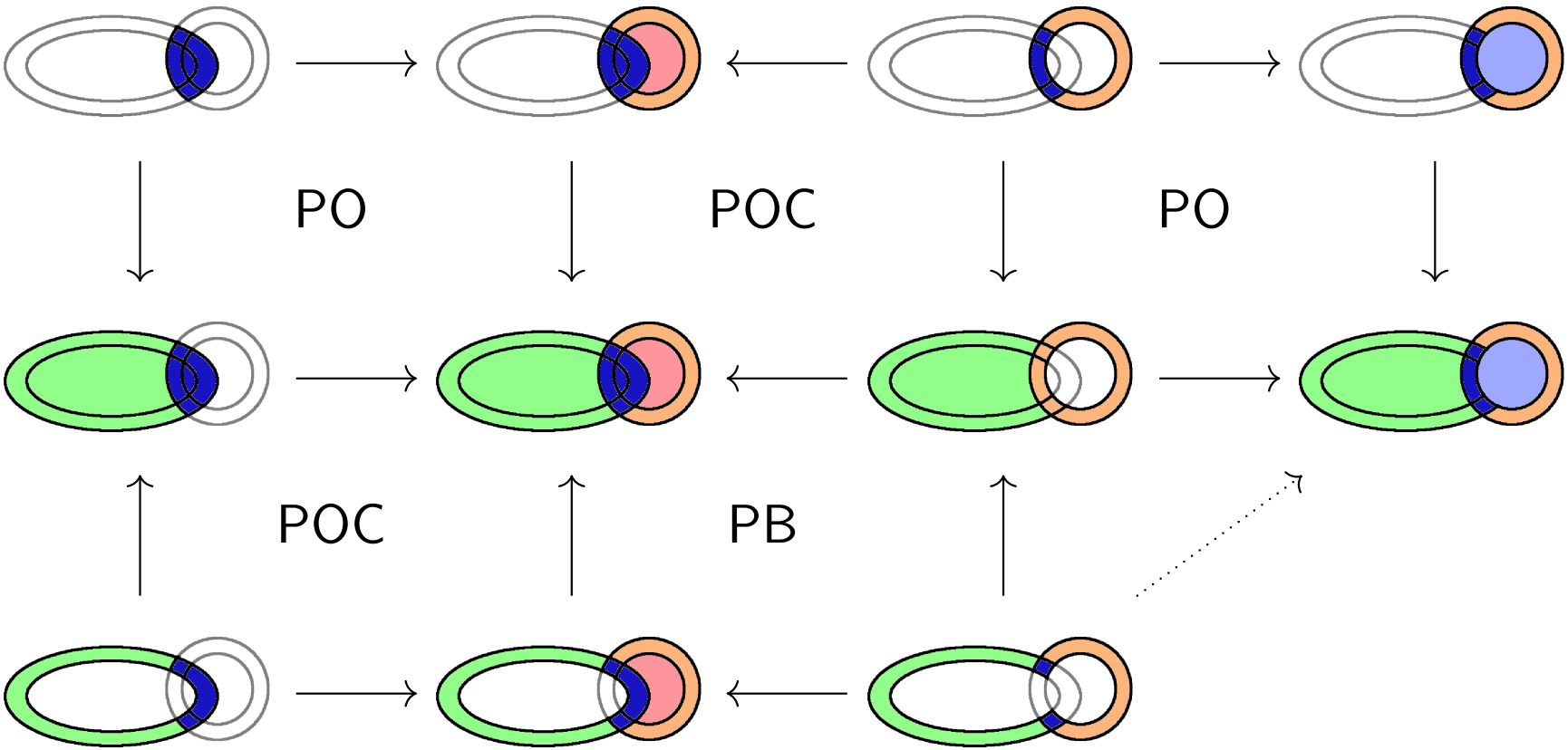
Composition with a context $J \to E \leftarrow \overline{J}$ is defined as follows:



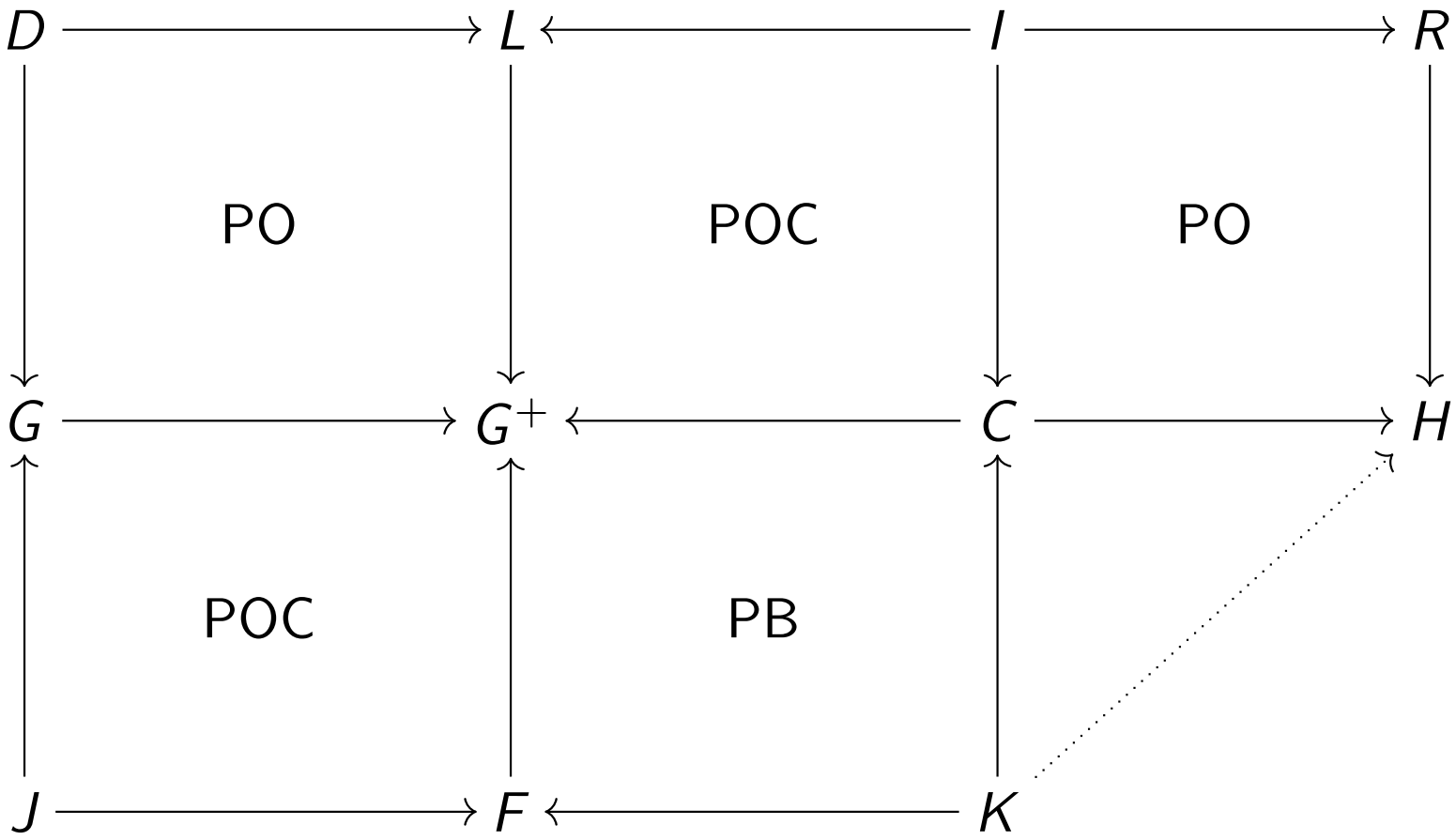


Motivation: contexts will be used as labels of a transition system; interfaces are needed in order to know where to attach contexts

Rewriting with borrowed contexts

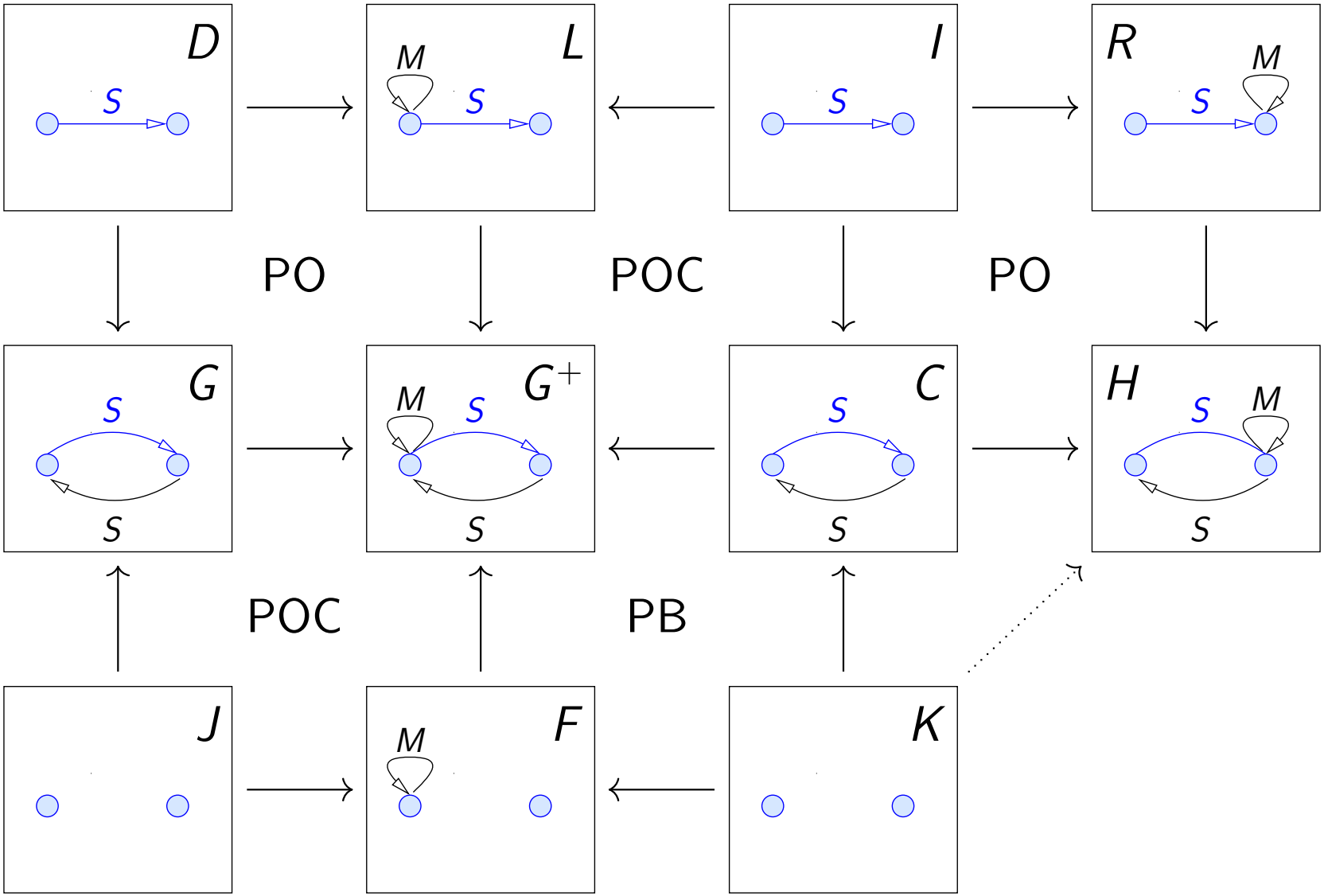


Rewriting with borrowed contexts



In this setting we write: $(J \to G) \overset{J \to F \leftarrow K}{\longrightarrow} (K \to H)$

Example (borrowed contexts)



Bisimilarity is a congruence

Main theorem

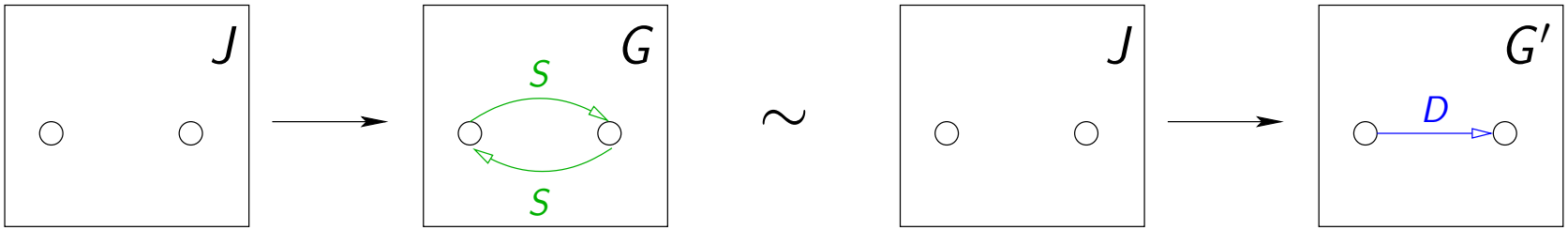
Bisimilarity defined on this labelled transition system is a congruence

$\rightsquigarrow$ it is preserved under composition with graph contexts

$\rightsquigarrow$ a subgraph can be replaced by a bisimilar one without changing the behaviour of the entire system

A simple example

Using some proof techniques (up-to techniques, etc.) one can easily show that:



$\rightsquigarrow$ Two simplex connections are bisimilar to a duplex connection.

Reactive systems

How does this relate to the work of Leifer/Milner and Sassone/Sobociński?

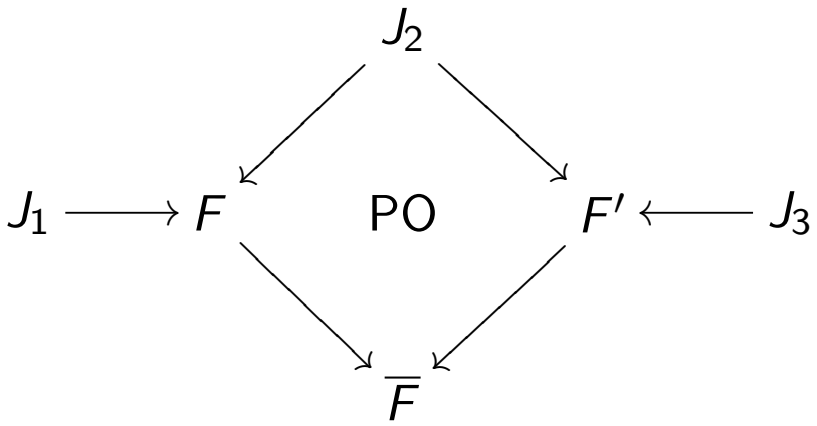
Reactive system over a category $\mathbf{C}$ (with a distinguished—usually initial—object 0):

- Rewriting rules: pairs of arrows (l, r) where $l, r : 0 \longrightarrow m$.
- Rewriting: $a \longrightarrow a'$ whenever there exists a rewriting rule (l, r) and an arrow D such that $a = Dl$ and $a' = Dr$.

Reactive systems and DPO

Let $\mathbf{C}$ be a category where

- objects are (interface) graphs
- arrows are cospans of graph morphisms $J \to F \leftarrow K$, where the middle graph is fixed only up to isomorphism (= contexts)
- composition of arrows is done by taking pushouts, i.e., by gluing together the interfaces (composition = contextualization)

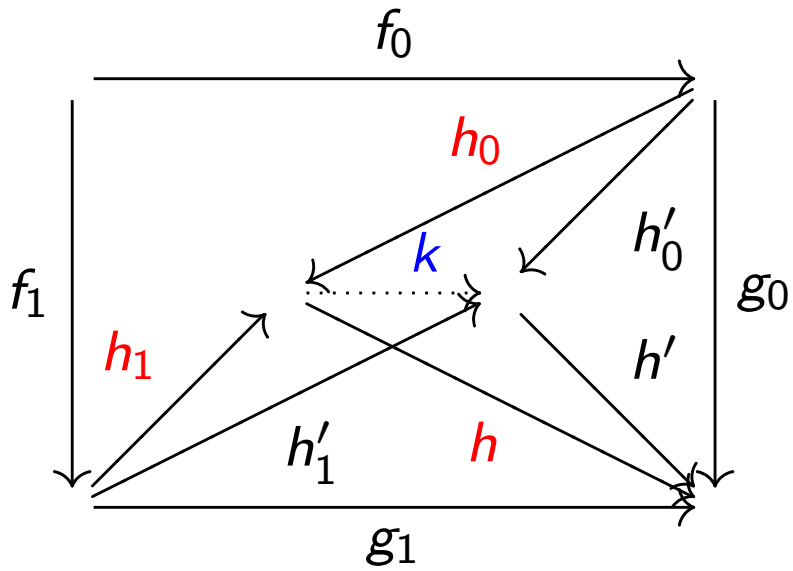


In this way we obtain exactly DPO rewriting.

Relative pushouts

How to characterize minimal contexts in this setting?
$\rightsquigarrow$ Minimality with respect to a given square.

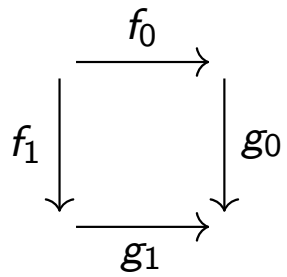
Relative pushout (concept due to Leifer/Milner)



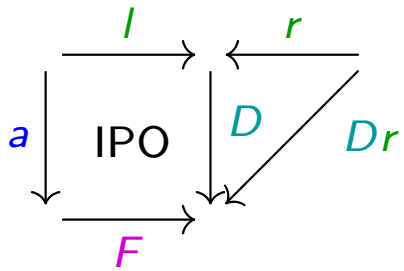
commuting diagram, i.e., $g_0 f_0 = g_1 f_1$.
there exists a unique arrow k such that everything commutes.
The relative pushout (RPO) of the outer square consists of the three arrows h_0, h_1, h.

Idem pushouts and labelled transitions

The square below is called an idem pushout (IPO) if the triple g_0, g_1, id is a relative pushout.



Deriving labels: Let $a\colon 0 \to m$ be an arrow. If there exists a rewriting rule (l, r) such that the following diagram exists and the square is an IPO, then we write $a \xrightarrow{F} Dr$.

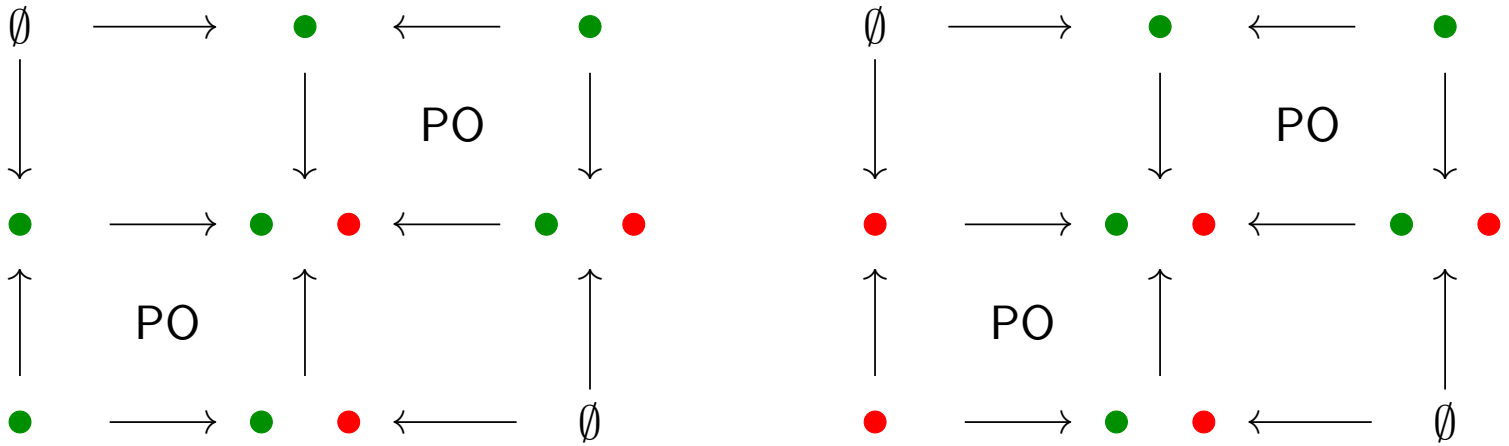


(The diagram shows one "minimal" way to extend a such that it contains a left-hand side l.)

Relative pushouts and cospan categories

Idea: Take relative pushouts in the cospan category.

But: relative pushouts do not exist. The problem is related to the following phenomenon: Due to non-trivial automorphisms the diagrams underlying commuting squares in the cospan category are not unique.



S-categories

Solution by Leifer/Milner

- Do not consider graphs up to isomorphism.
- Give unique unchangeable names to graph components (= add support).
- Composition of graphs is undefined whenever the supports are not disjoint.

$\rightsquigarrow$ composition is a partial operation
$\rightsquigarrow$ work in a precategory

This theory has also been developed for bigraphs (a variant of hierarchical graphs).

Groupoidal relative pushouts

Solution by Sassone/Sobociński:

Do not take the middle graph in a cospan up to isomorphism (choice of pushout)

$\rightsquigarrow$ associativity only up to isomorphism, one needs a bicategory

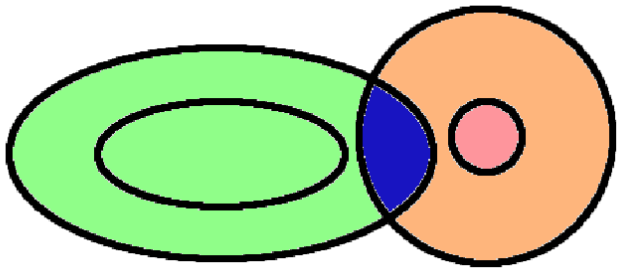
$\rightsquigarrow$ commuting squares come with a proof of isomorphism

- once (groupoidal) relative pushouts are known to exist (this can be fairly complicated), the congruence proofs (also for other equivalences such as trace or failures equivalence) are easier than in our case
- the derived labels are the same than in the case of borrowed contexts

Further Issues

Proof techniques:

- Whenever a graph and a left-hand side overlap only in their interfaces, the corresponding rewriting step does not have to be considered.



- Bisimulation up-to context

  If $(J \to G)\,\mathcal{R}\,(J \to G')$ and $(J \to G) \overset{J \to F \leftarrow K}{\longrightarrow} (K \to H)$,
  we only have to show that

  $$(J \to G') \overset{J \to F \leftarrow K}{\longrightarrow} (K \to H') \text{ and } (J \to H)\,\hat{\mathcal{R}}\,(J \to H').$$

  where $\hat{\mathcal{R}}$ is the closure of $\mathcal{R}$ under contextualization.

Further Issues

Mechanization of bisimulation proofs:

Automatic bisimulation proofs require:

- efficient label derivation techniques (based on the borrowed context method)
- powerful up-to techniques (up-to-context, etc.)
- on-the-fly bisimulation checkers (Hirschkoff, Fernandez/Mounier) in order to handle the infinite state space

Some initial results by Rangel/Ehrig/König

Further Issues

Structured Operational Semantics:

$\rightsquigarrow$ Recover the inductive view of process algebra!

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

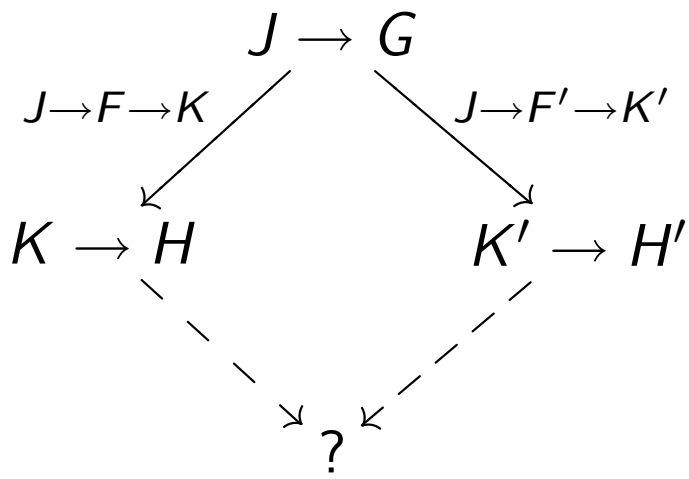
Idea: define morphisms between rewriting steps and compose rewriting steps by taking (relative?) pushouts in a suitable category.

Some initial results for composition/decomposition of borrowed context transformations can be found in: Baldan, Ehrig, König: "Composition and Decomposition of DPO Transformations with Borrowed Context"

Further Issues

Church-Rosser, Sequential/Parallel Independence:

Given two labelled transitions of a graph/an object. Under which conditions are they independent, i.e., when can the diamond be closed?



$\rightsquigarrow$ Filippo Bonchi, Tobias Heindel: "Church-Rosser for Borrowed Context Rewriting" (talk at CALCO-jnr on Monday)

Further Issues

Automatically deriving behavioural equivalences:

- Petri nets (Sassone/Sobociński, Leifer/Milner)
- CCS (Milner, Bonchi/Gadducci/König)
- π-calculus (Bundgaard/Sassone, Jensen/Milner)
- Logic programming (Bonchi/König/Montanari)

Lessons learned:

- The equivalence you get is dependent on the encoding you choose.
- Deriving labels and comparing the transition systems (and the induced equivalences) can be a lot of work.

Further Issues

Fine-tuning behavioural equivalences: which equivalence is the right one?

Bisimilarity based on borrowed contexts/RPOs allows us to observe "minimal contexts".

But: this is not the coarsest bisimulation (on unlabelled transitions) which is a congruence
$\rightsquigarrow$ saturated bisimilarity (Bonchi/König/Montanari).

Idea: try to classify behavioural equivalences according to the power that is given to the environment. The environment may observe so-called "barbs" (in asynchronous π-calculus: output barbs) $\rightsquigarrow$ study barbed equivalences.

Rathke/Sassone/Sobociński: "Semantic barbs and biorthogonality"

Other equivalences have not been studied to a great extent: weak bisimilarity, failures equivalence, trace equivalence

Further Issues

Possible Applications:

- Automatically derive behavioural equivalences for process calculi
- Show behaviour preservation of model transformations (deal with negative application conditions!)
- Show correctness of (cryptographic) protocols (secrecy, authentication), similar to the work on Spi-calculus (Abadi/Gordon)