Sesqui-pushout rewriting

Andrea Corradini

Dipartimento di Informatica, Pisa, Italy

IFIP WG 1.3 - La Roche en Ardennes, June 6, 2006.

Joint work with

Tobias Heindel Universität Stuttgart, Germany

Frank Hermann Technisce Universität Berlin, Germany

Barbara König Universität Duisburg-Essen, Germany

Outline

- What and why? Some shallow motivations...
- Algebraic graph rewriting: DPO and SPO
- **Sesqui-pushout (SqPO) rewriting**
- An example of SqPO rewriting at work
- On the existence of pushback complements
- Relating SqPO with DPO and SPO
- Some deeper motivations and future perspectives

Some shallow motivations

Sesqui-pushout rewriting is a new categorical definition of rewriting in an arbitrary category, similar to double-pushout or single-pushout rewriting

- The name (sesqui means one and a half in Latin) indicates that conceptually it lies between the SPO and the DPO
- Technically, it is defined as DPO rewriting, where the left pushout is replaced by a suitable pullback
- It looks more adequate than DPO/SPO in some cases, and it enjoys several nice properties that will be detailed later

Double-pushout rewriting in $\mathbb{C}$

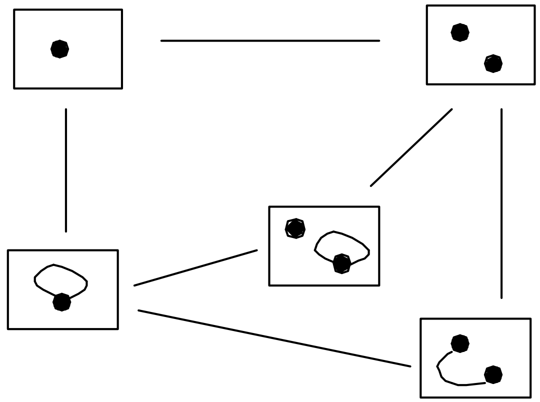
- A rule is a span $q = L \stackrel{\alpha}{\leftarrow} K \stackrel{\beta}{\rightarrow} R$
- A match is an arrow $m : L \to G$
- Direct derivation $A \stackrel{\langle m,q \rangle}{\Longrightarrow} B$ if the following double-pushout diagram can be constructed:

$$\begin{array}{ccccc} L & \stackrel{\alpha}{\longleftarrow} & K & \stackrel{\beta}{\longrightarrow} & R \\ {\scriptstyle m}\downarrow & \ulcorner & \downarrow{\scriptstyle i} & \urcorner & \downarrow{\scriptstyle c} \\ A & \underset{\gamma}{\longleftarrow} & D & \underset{\delta}{\longrightarrow} & B \end{array}$$

Note: The left square is built as a pushout complement, not characterized as a universal construction: "general DPO is ambiguous"

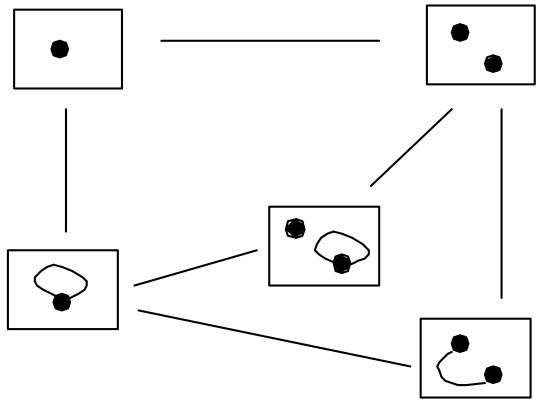
Pushout complements in Graph

Example of ambiguity:

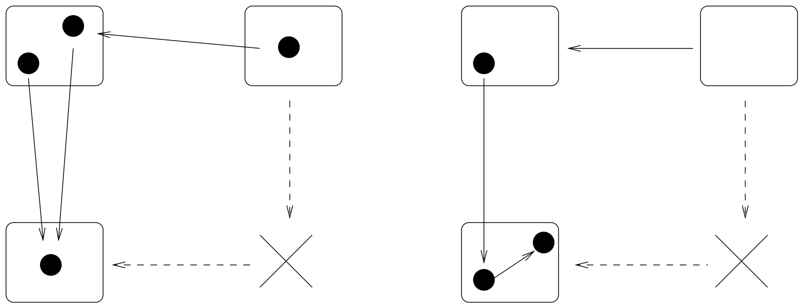


Pushout complements in Graph

Example of ambiguity:



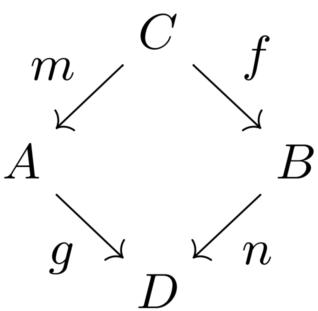
- for injective top-arrows, if the POC exists, it is unique;
- in this case, it exists iff the identification and dangling conditions are satisfied.

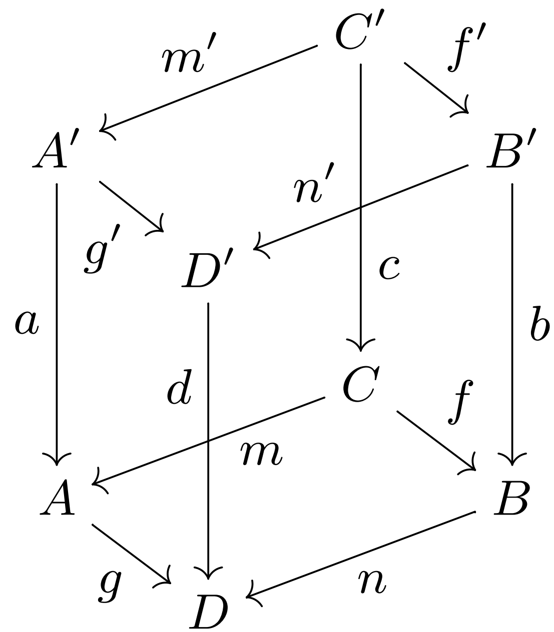


Quasi-adhesive categories

A quasi-adhesive category:

- has pullbacks, has pushouts along regular monos
- pushouts along regular monos are Van Kampen squares





DPO theory in quasi-adhesive cats

- Parallel and Sequential Independence
- Parallel Productions and Derivations
- Local Church-Rosser and Parallelism Theorem
- Shift Equivalence and Canonical Derivations
- Concurrency Theorem
- Embedding and extensions
- Critical pair lemma
- ...

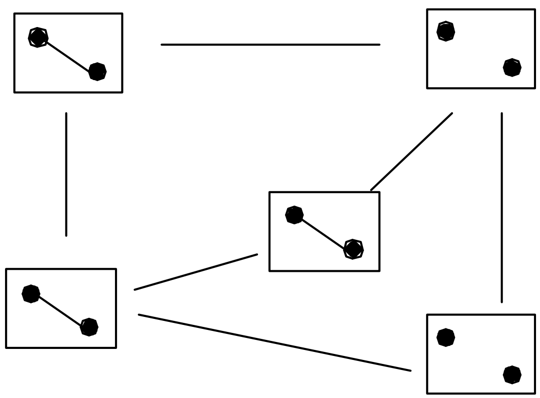
POC in quasi-adhesive categories

- Pushout complements along regular mono are unique.

  Examples of quasi-adhesive categories:

- Category of term graphs
  - regular monos are monos reflecting variables
- Category of simple graphs
  - regular monos are monos reflecting edges

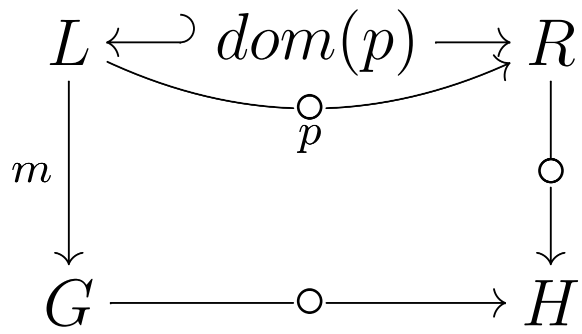
If the top arrow is mono but not regular, the POC might not be unique.



Single-Pushout rewriting on graphs

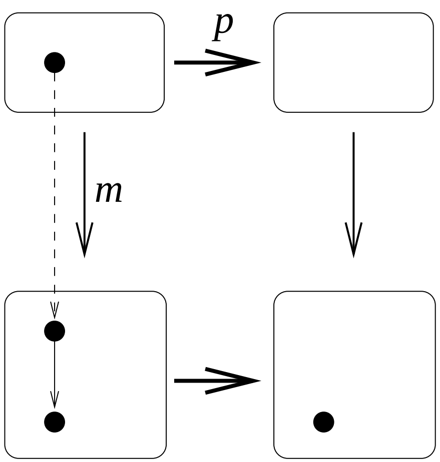
Productions are partial morphisms

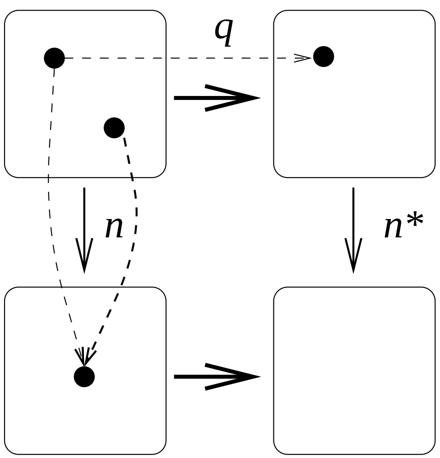
Match: total morphism



No dangling and identification conditions

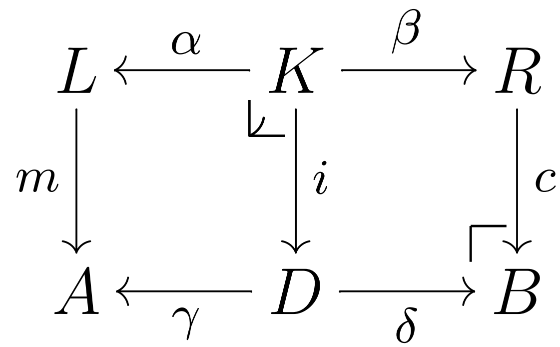
- deletion in unknown context
- deletion stronger than preservation





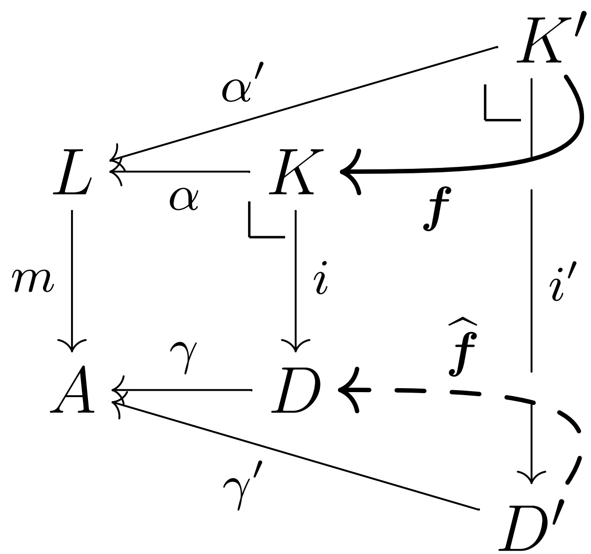
Defining sesqui-pushout rewriting

$A \xrightarrow{\langle m,q \rangle} B$ if



- the right square is a pushout
- $\langle D, i, \gamma \rangle$ is a pushback complement of $A \xleftarrow{m} L \xleftarrow{\alpha} K$

that is, a final pullback complement of m:

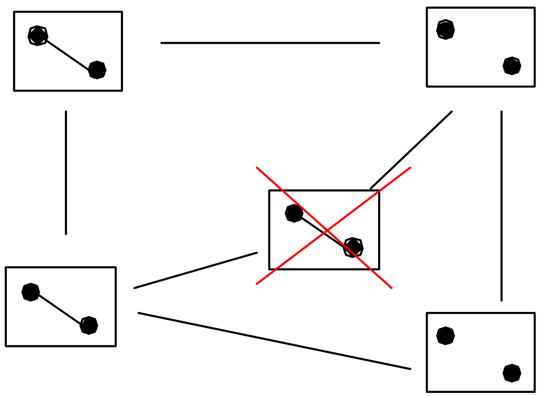


- the square is a pullback
- for each other pullback (over m) and for each $\boldsymbol{f} \colon K' \to K$ such that $\alpha \circ \boldsymbol{f} = \alpha'$, there exists a unique $\widehat{\boldsymbol{f}} \colon D' \to D$ making everything commute

A few properties of SqPO rewriting

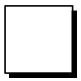
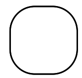
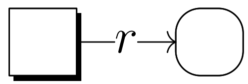
- In any category $\mathbb{C}$, pushback complements are unique: SqPO rewriting is not ambiguous!
  - even if $L \xleftarrow{\alpha} K$ is not mono $\Rightarrow$ cloning
  - even if $\mathbb{C}$ is quasi-adhesive, $L \xleftarrow{\alpha} K$ is mono but not regular.
- If $L \xleftarrow{\alpha} K$ is mono, then $A \xleftarrow{\gamma} D$ is mono, and D is the largest subobject of A making the square a pullback

Pushback complement along monic, non-regular morphism in category of simple graphs.



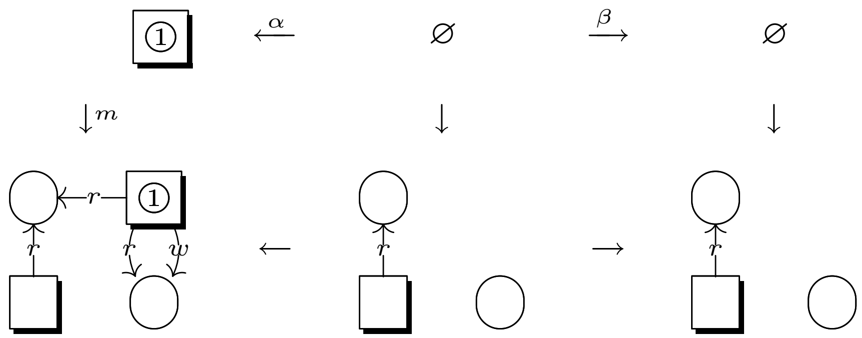
An example: Access Control

Modeling basic operations of simple Access Control system from [Harrison, Ruzzo, Ullman, CACM 1976]

- Simple graphs including nodes representing subjects () and objects (), and labeled edges representing rights ().
- Already modeled by [Koch, Mancini, Parisi-Presicce, ESORICS'00] using DPO on (multi-)graphs, with Negative Application Conditions.
- Basic Operations:

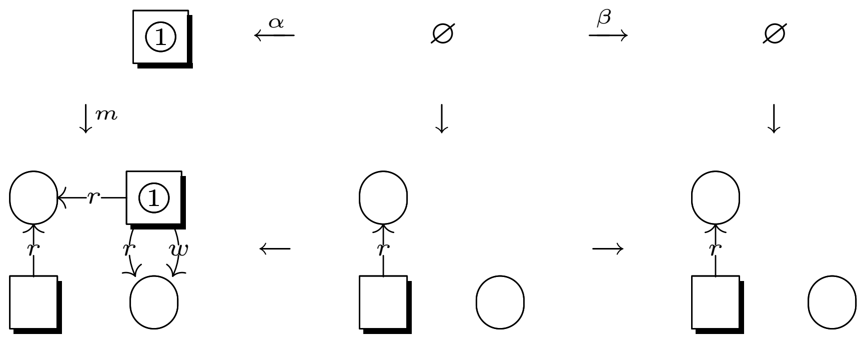
create subject X_s	create object X_o
destroy subject X_s	destroy object X_o
enter i into(X_s, X_o)	delete i into(X_s, X_o)

Some rules and their effect

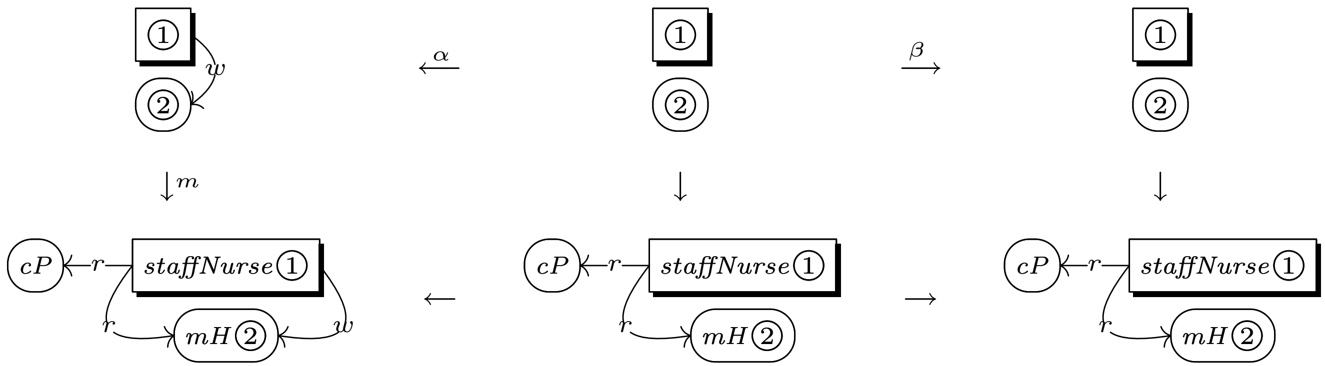


Application of `destroy subject` X_s: deletion in unknown context as for SPO

Some rules and their effect



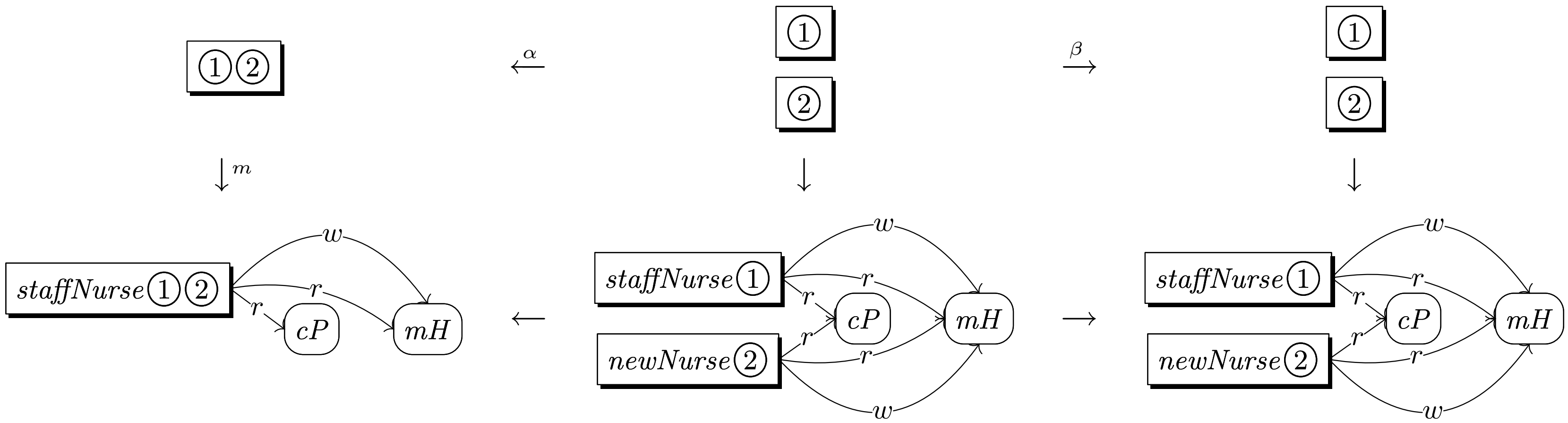
Application of destroy subject X_s: deletion in unknown context as for SPO



Application of delete w into(X_s, X_o): not ambiguous.

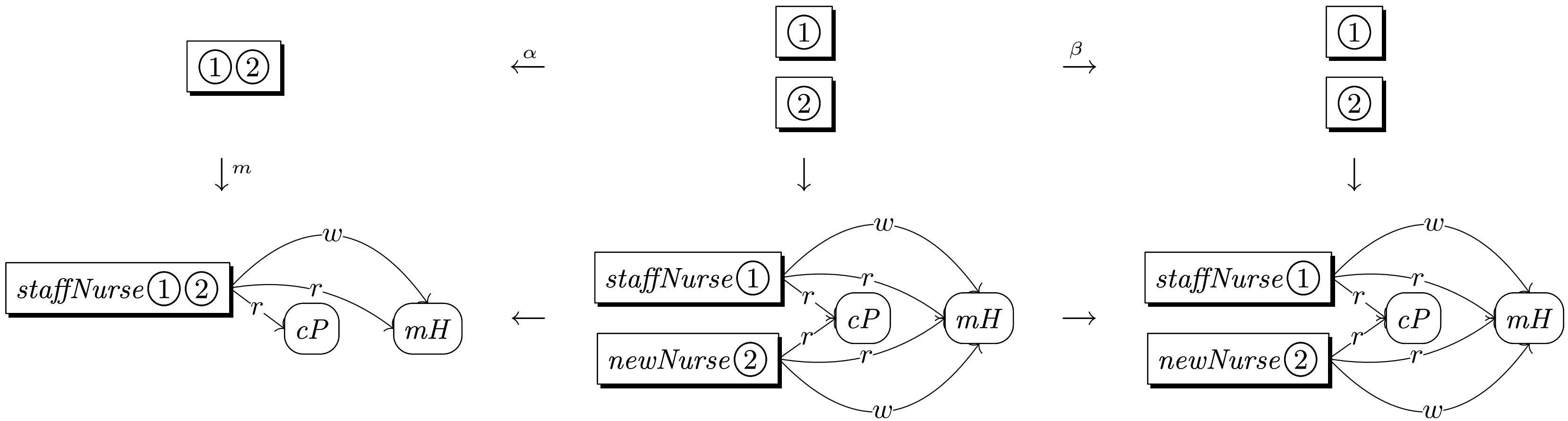
A new rule: `clone subject`

Non-left-injective rules model cloning

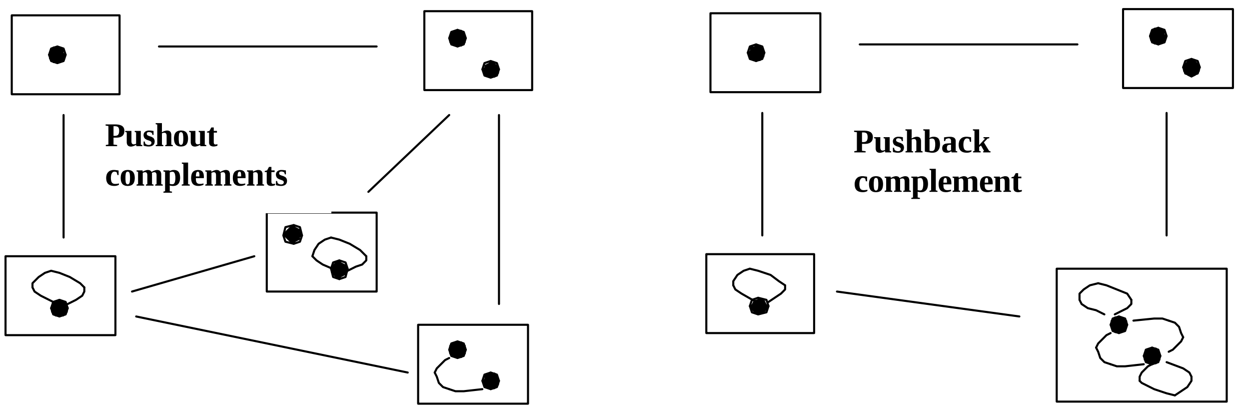


A new rule: `clone subject`

Non-left-injective rules model cloning



In **Graph**, the pushback complement might not be a POC.



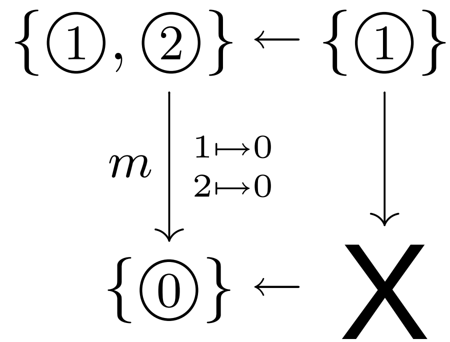
On the existence of pushback complement

In categories like **Set**, **Graph**, graph structures in general:

On the existence of pushback complement

In categories like **Set**, **Graph**, graph structures in general:

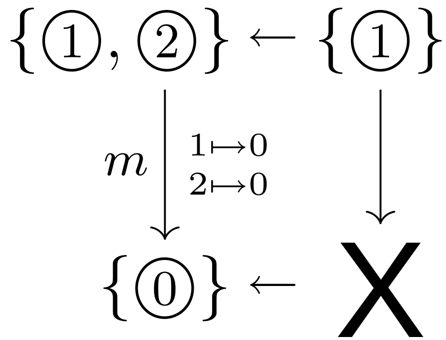
- for injective $L \overset{\alpha}{\hookleftarrow} K$ the pushback complement exists iff the match is conflict-free, i.e., $m(L \setminus K) \cap m(K) = \varnothing$.



On the existence of pushback complement

In categories like **Set**, **Graph**, graph structures in general:

- for injective $L \overset{\alpha}{\leftarrowtail} K$ the pushback complement exists iff the match is conflict-free, i.e., $m(L \setminus K) \cap m(K) = \varnothing$.

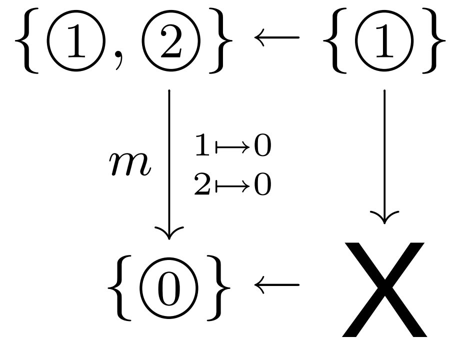


- for arbitrary $L \overset{\alpha}{\leftarrow} K$ and injective matches: see Construction 6 in the paper...

On the existence of pushback complement

In categories like **Set**, **Graph**, graph structures in general:

- for injective $L \overset{\alpha}{\longleftarrow\!\!\!<} K$ the pushback complement exists iff the match is conflict-free, i.e., $m(L \setminus K) \cap m(K) = \varnothing$.

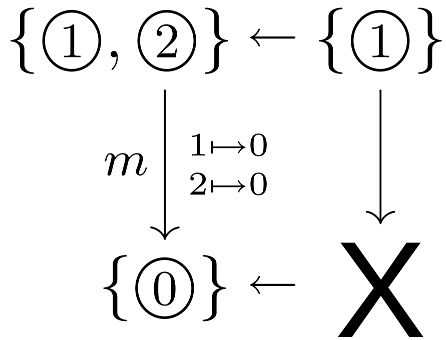


- for arbitrary $L \overset{\alpha}{\leftarrow} K$ and injective matches: see Construction 6 in the paper...
- for arbitrary $L \overset{\alpha}{\leftarrow} K$ and arbitrary matches:

On the existence of pushback complement

In categories like **Set**, **Graph**, graph structures in general:

- for injective $L \overset{\alpha}{\leftarrowtail} K$ the pushback complement exists iff the match is conflict-free, i.e., $m(L \setminus K) \cap m(K) = \varnothing$.



- for arbitrary $L \overset{\alpha}{\leftarrow} K$ and injective matches: see Construction 6 in the paper...

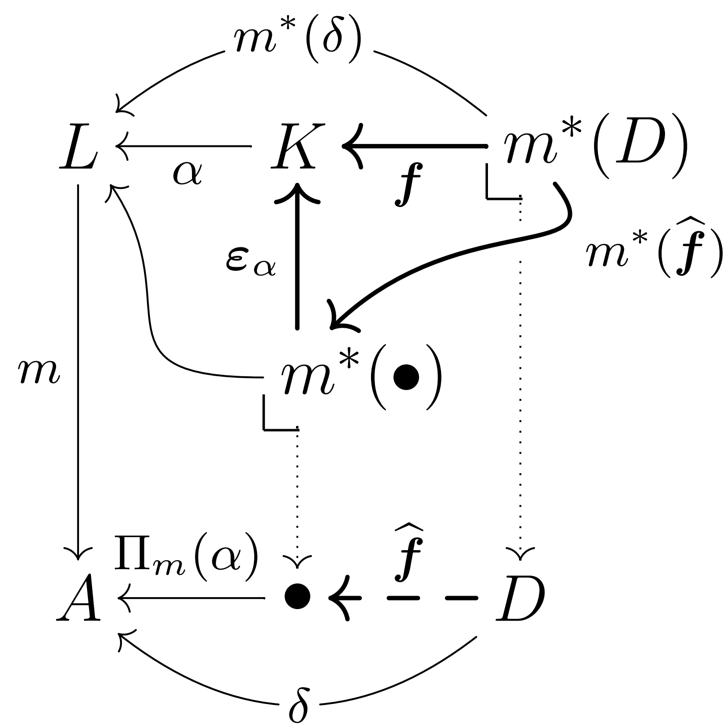
- for arbitrary $L \overset{\alpha}{\leftarrow} K$ and arbitrary matches: *“conditions... rather involved...; ... beyond the scope of the paper...; the interested reader is encouraged to specialize the concepts that are availabe for every topos [Goldblatt] and the results in next section...; ... the pushback construction cannot be performed componentwise...”*

Existence in an arbitrary category $\mathbb{C}$

Given $L \xrightarrow{m} A$ consider the pullback functor $m^*\colon \mathbb{C}\downarrow\mathbb{A} \to \mathbb{C}\downarrow\mathbb{L}$ along m.
If its right adjoint

$$\Pi_m\colon \mathbb{C}\downarrow\mathbb{L} \to \mathbb{C}\downarrow\mathbb{A}$$

exists partially at α, it provides a pullback complement iff the co-unit ε_α is an iso.



This provides a construction of pushback complements in categories where the pullback functors have right adjoints (like locally cartesian closed cats).

esqui-pushout *vs* double-pushout rewritin

- In quasi-adhesive categories, for a left-regular rule q and a match m, if the POC exists, then it is a PshBC. Thus
  1. If $A \overset{\langle m,q\rangle}{\underset{\text{DPO}}{\Longrightarrow}} B$ then also $A \overset{\langle m,q\rangle}{\Longrightarrow} B$.
  2. If $A \overset{\langle m,q\rangle}{\Longrightarrow} B$ and a pushout complement exists, then also $A \overset{\langle m,q\rangle}{\underset{\text{DPO}}{\Longrightarrow}} B$.

esqui-pushout *vs* double-pushout rewritin

- In quasi-adhesive categories, for a left-regular rule q and a match m, if the POC exists, then it is a PshBC. Thus
  1. If $A \overset{\langle m,q \rangle}{\underset{\text{DPO}}{\Longrightarrow}} B$ then also $A \overset{\langle m,q \rangle}{\Longrightarrow} B$.
  2. If $A \overset{\langle m,q \rangle}{\Longrightarrow} B$ and a pushout complement exists, then also $A \overset{\langle m,q \rangle}{\underset{\text{DPO}}{\Longrightarrow}} B$.
- For left-monic but non-left-regular rules, in some examples the PshBC is a POC. It is open if this is a general property.

esqui-pushout *vs* double-pushout rewritin

- In quasi-adhesive categories, for a left-regular rule q and a match m, if the POC exists, then it is a PshBC. Thus
  1. If $A \overset{\langle m,q \rangle}{\underset{\text{DPO}}{\Longrightarrow}} B$ then also $A \overset{\langle m,q \rangle}{\Longrightarrow} B$.
  2. If $A \overset{\langle m,q \rangle}{\Longrightarrow} B$ and a pushout complement exists, then also $A \overset{\langle m,q \rangle}{\underset{\text{DPO}}{\Longrightarrow}} B$.
- For left-monic but non-left-regular rules, in some examples the PshBC is a POC. It is open if this is a general property.
- For non-left-monic rules, the PshBC is not a POC, in general.

Sesqui-pushout *vs* single-pushout rewriting

In categories of graph structures, given a partial morphism q (seen also as left-injective span) and a match m,

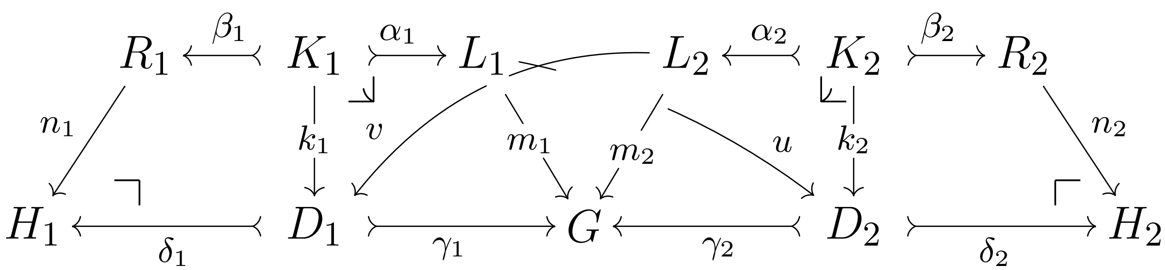
1. If $A \stackrel{\langle m,q\rangle}{\Longrightarrow} B$ then $A \underset{\text{SPO}}{\stackrel{\langle m,q\rangle}{\Longrightarrow}} B$.
2. If $A \underset{\text{SPO}}{\stackrel{\langle m,q\rangle}{\Longrightarrow}} B$ and m is conflict-free then $A \stackrel{\langle m,q\rangle}{\Longrightarrow} B$.

Note that usually non-conflict-free matches are ruled out in practical uses or theoretical developments of the SPO theory, restricting to d-injective or even to injective matches.

Theory of parallelism

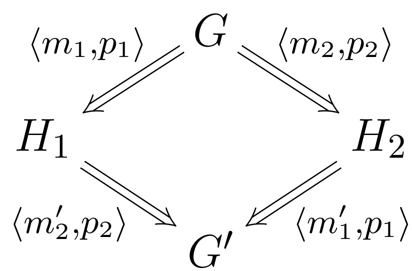
Some first results of the DPO/SPO theory have been recast for SqPO rewriting

Parallel Independence



Local Church-Rosser Theorem

Given parallel independent $G \xRightarrow{\langle m_1, p_1 \rangle} H_1$ and $G \xRightarrow{\langle m_2, p_2 \rangle} H_2$, there are an object G' and direct derivations $H_1 \xRightarrow{\langle m_2', p_2 \rangle} G'$ and $H_2 \xRightarrow{\langle m_1', p_1 \rangle} G'$.



Back to motivations...

Semantics of concurrency for GTS

- Rewriting of graphs is intrinsically concurrent
- Petri nets are a reference model for concurrency
- (Place/Transition) Petri nets can be seen as a degenerate case of Graph Transformation
- A *robust* semantics of concurrency for GT should specialize to known semantics for nets

Some contributions to the field

We defined generalizations to other classes of nets and to DPO or SPO rewriting in Graph of

- deterministic and non-deterministic processes
- unfolding and event structure semantics
- functorial (coreflective) semantics

Winskel's style semantics for GTS

$$\textbf{Safe Nets} \;\; \begin{array}{c} \xleftarrow{\qquad}\!\!\supset \\ \perp \\ \xrightarrow[\mathcal{U}]{} \end{array} \;\; \textbf{Occurrence Nets} \;\; \begin{array}{c} \xleftarrow{\mathcal{N}} \\ \perp \\ \xrightarrow[\mathcal{E}]{} \end{array} \;\; \textbf{Prime Event Structures} \;\; \begin{array}{c} \xleftarrow{\mathcal{P}} \\ \sim \\ \xrightarrow[\mathcal{L}]{} \end{array} \;\; \textbf{Domains}$$

Winskel's style semantics for GTS

$$\textbf{Safe Nets} \underset{\mathcal{U}}{\overset{\hookleftarrow}{\underset{\bot}{\longrightarrow}}} \textbf{Occurrence Nets} \underset{\mathcal{E}}{\overset{\mathcal{N}}{\underset{\bot}{\rightleftarrows}}} \textbf{Prime Event Structures} \underset{\mathcal{L}}{\overset{\mathcal{P}}{\underset{\sim}{\rightleftarrows}}} \textbf{Domains}$$

[Works with Paolo Baldan, Ugo Montanari, Leila Ribeiro]

$$\textbf{DPO Graph Grammars} \overset{\hookleftarrow}{\underset{\bot}{\longrightarrow}} \textbf{DPO Occ. Grammars} \longrightarrow \textbf{Inhibitor Event Structures} \overset{\longleftarrow}{\underset{\bot}{\longrightarrow}} \textbf{Domains}$$

Winskel's style semantics for GTS

$$\text{Safe Nets} \; \underset{\mathcal{U}}{\overset{\perp}{\rightleftarrows}} \; \text{Occurrence Nets} \; \underset{\mathcal{E}}{\overset{\mathcal{N}}{\underset{\perp}{\rightleftarrows}}} \; \text{Prime Event Structures} \; \underset{\mathcal{L}}{\overset{\mathcal{P}}{\underset{\sim}{\rightleftarrows}}} \; \text{Domains}$$

[Works with Paolo Baldan, Ugo Montanari, Leila Ribeiro]

$$\text{DPO Graph Grammars} \; \overset{\perp}{\rightleftarrows} \; \text{DPO Occ. Grammars} \longrightarrow \text{Inhibitor Event Structures} \; \overset{\perp}{\rightleftarrows} \; \text{Domains}$$

$$\text{SPO Graph Grammars} \; \overset{\perp}{\rightleftarrows} \; \text{SPO Occ. Grammars} \; \overset{\perp}{\rightleftarrows} \; \text{Asymmetric Event Structures} \; \overset{\perp}{\rightleftarrows} \; \text{Domains}$$

The next steps, quite obviously...

...generalizing the semantics developed for concrete models to rewriting systems in adhesive categories...

The next steps, quite obviously...

...generalizing the semantics developed for concrete models to rewriting systems in adhesive categories...

- First results: generalization of processes [with Paolo Baldan, Tobias Heindel, Barbara König, FoSSaCS’06]

The next steps, quite obviously...

...generalizing the semantics developed for concrete models to rewriting systems in adhesive categories...

- First results: generalization of processes [with Paolo Baldan, Tobias Heindel, Barbara König, FoSSaCS'06]

Before moving to unfolding semantics, we noted that:

- for DPO rewriting, a coreflective semantics is impossible;
- SPO rewriting more appealing, but generalization to arbitrary categories is quite involved; no consensus on the way conflicts are resolved;
- thus, need for a notion of rewriting similar to DPO, but without application conditions

Conclusions...

- I presented the definition and a few properties of Sesqui-pushout rewriting, relating it to DPO and SPO rewriting
- It is not-ambiguous, allows to model cloning, and coincides with DPO and SPO under suitable assumptions
- Some basic results about parallelism have been lifted to SqPO rewriting: a lot has to be done, still. Several results of DPO/SPO theory should lift easily.

... and Future Work

- The expressiveness of the approach should be compared with that of DPO/SPO rewriting on practical case studies
- Generalizing the coreflective semantics of SPO rewriting to that if SqPO rewriting in quasi-adhesive theories