Graph rewriting with cloning

Dominique Duval
based on work with Rachid Echahed and Frédéric Prost

LJK-LIG, University of Grenoble

February 28., 2012 – Grenoble – ANR CLIMT

Outline

Graph rewriting

L, R, G, H are graphs.

For each rewrite rule:

$$L \rightsquigarrow R$$

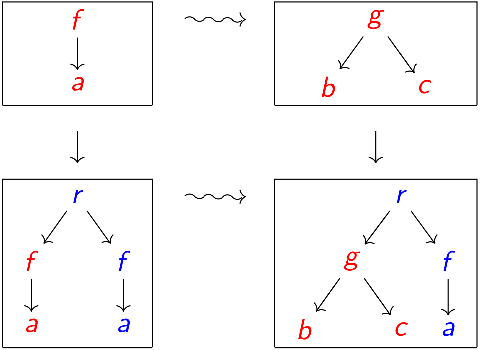
and each matching:

$$\begin{tikzcd} L \arrow[d, "\subseteq"] \\ G \end{tikzcd}$$

a rewrite step builds H by replacing
the occurrence of L in G
by some occurrence of R in H:

$$\begin{tikzcd} R \arrow[d, "\subseteq"] \\ H \end{tikzcd}$$

Example: term rewriting



Some questions:

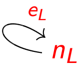
1. What is a graph?
2. What is a rule?
3. What does replacing mean?
4. Is there a rule $G \rightsquigarrow H$?

In this talk:

1. A graph is a directed multigraph.
2. A rule is a span $L \leftarrow K \rightarrow R$.
3. Several answers for replacing: DPO, SqPO, PSqPO.
4. $G \rightsquigarrow H$ is a rule $G \leftarrow D \rightarrow H$.

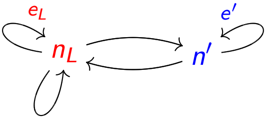
Subgraph classifier: What does replacing mean?

L, R are graphs.

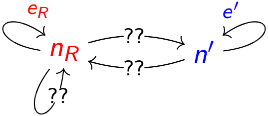
$L =$ 

$R =$ 

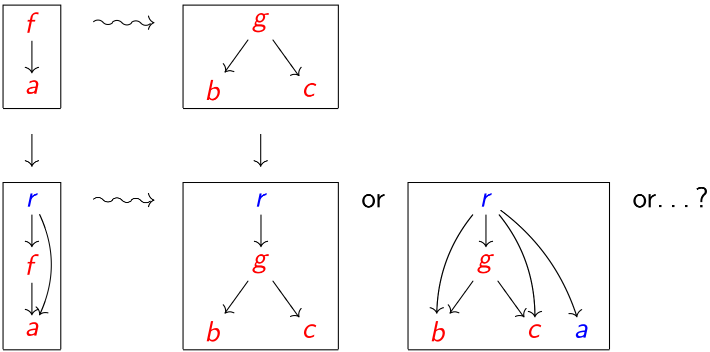
$L \subseteq G$.

$G =$ 

$R \subseteq H$, after rewriting.

$H =$ 

Example: What does replacing mean?



Outline

Algebraic graph transformation

Algebraic graph rewriting is based on category theory
especially on pushouts:

- Single-pushout: SPO
- Double-pushout: DPO
- Sesqui-pushout: SqPO

By: H. Ehrig, U. Montanari, M. Löwe,
A. Corradini, B. König, L. Ribeiro,
S. Lack, T. Heindel, P. Sobocinski, . . .

Pushouts

Union

$$
\begin{tikzcd}
X \cap Y \arrow[r, "\subseteq"] \arrow[d, "\subseteq"'] & Y \arrow[d, dashed, "\subseteq"] \\
X \arrow[r, dashed, "\subseteq"'] & X \cup Y
\end{tikzcd}
$$

Pushout: a kind of generalized union ("amalgamated sum")

$$
\begin{tikzcd}
W \arrow[r] \arrow[d] & Y \arrow[d, dashed] \\
X \arrow[r, dashed] & Z \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}
$$

- When a pushout exists, it is unique (up to iso).
- Categories **Set** and **Graph** have pushouts.

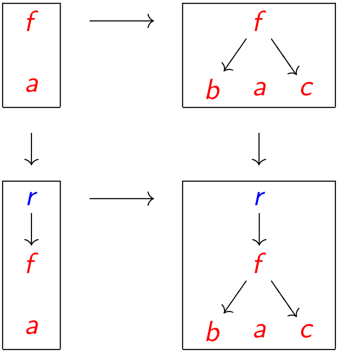
PO of graphs

There is a GRAPH OF GRAPHS:

$$\begin{tikzcd} V & E \arrow[l, bend right, "s"'] \arrow[l, bend left, "t"] \end{tikzcd}$$

- Pushouts of graphs exist
  and they can be computed pointwise.

Example: a PO of graphs



DPO, SqPO, PSqPO

In this talk, every ??PO of graphs looks like:

$$
\begin{tikzcd}
L \arrow[d] \arrow[dr, phantom, "??"] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, dashed] & R \arrow[d, dotted] \\
G & D \arrow[l, dashed, "l_1"] \arrow[r, dotted, "r_1"'] & H \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}
$$

Outline

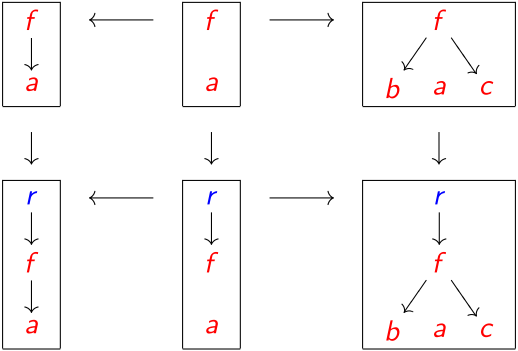
Double-pushout (DPO)

The LHS square is a pushout complement (POC)

$$
\begin{tikzcd}
L \arrow[d] \arrow[dr, phantom, "\lrcorner", very near end] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, dashed] \arrow[dr, phantom, "\llcorner", very near end] & R \arrow[d, dotted] \\
G & D \arrow[l, dashed, "l_1"] \arrow[r, dotted, "r_1"'] & H
\end{tikzcd}
$$

- \+ Easy to understand: symmetric
- \+ Easy to define
- \+ Sound categorical base: adhesive categories

Example: DPO



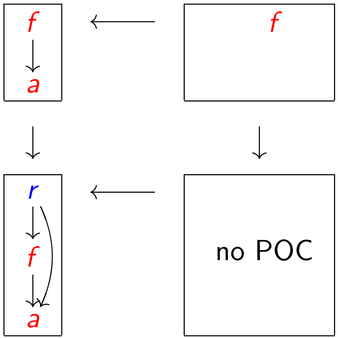
Adhesive categories

- Definition of adhesive categories
  involves Van Kampen squares...
- Categories **Set** and **Graph** are adhesive.

In an adhesive category:

- pushouts of monos are monos
- pushouts along monos are pullbacks
- pushout complements of monos are unique (if they exist)

Example: no POC



Outline

Sesqui-pushout (SqPO)

The LHS square is a final pullback complement (FPBC)

$$\begin{tikzcd}
L \arrow[d] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, dashed] & R \arrow[d, dotted] \\
G & D \arrow[l, dashed, "l_1"] \arrow[r, dotted, "r_1"'] & H
\end{tikzcd}$$

+ FPBC of graphs exist and are unique (up to iso)
+ PBCs are more general than POCs

Pullbacks

Intersection

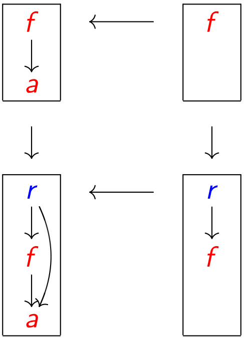
$$
\begin{tikzcd}
X \cap Y \arrow[r, dashed, "\subseteq"] \arrow[d, dashed, "\subseteq"'] & Y \arrow[d, "\subseteq"] \\
X \arrow[r, "\subseteq"'] & Z
\end{tikzcd}
$$

Pullback: a kind of generalized intersection ("fibered product")

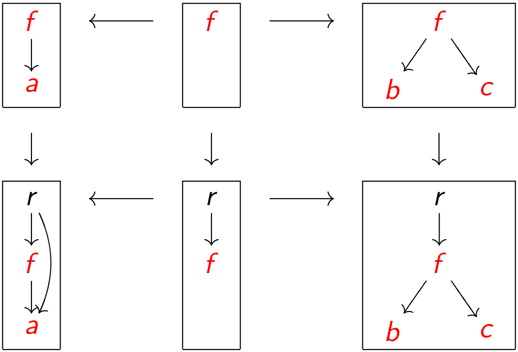
$$
\begin{tikzcd}
W \arrow[r, dashed] \arrow[d, dashed] \arrow[dr, phantom, "\lrcorner", very near start] & Y \arrow[d] \\
X \arrow[r] & Z
\end{tikzcd}
$$

- When a pullback exists, it is unique (up to iso).
- Categories **Set** and **Graph** have pullbacks.

Example: a FPBC

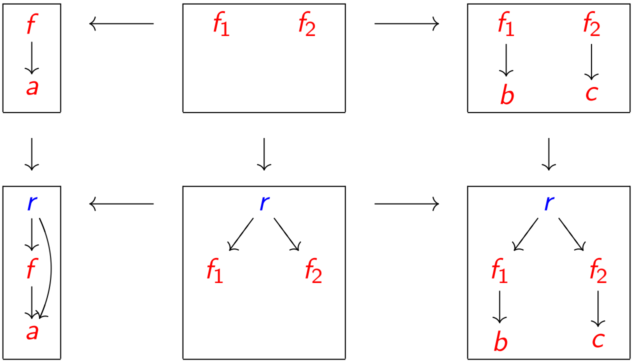


Example: a SqPO



Example: cloning and deleting nodes with a SqPO

Goal: cloning and deleting some nodes and their incident edges.
Node f is clone twice. Node a is deleted.



Outline

Graph transformation

Algebraic graph transformation

Double-pushout (DPO)

Sesqui-pushout (SqPO)

Polarized sesqui-pushout (PSqPO)

Polarized graphs

There is a "GRAPH" OF POLARIZED GRAPHS:

$$\begin{tikzcd} V^+ \arrow[d, tail] & \\ V & E \arrow[ul, bend right, "s^+"'] \arrow[dl, bend left, "t^-"] \\ V^- \arrow[u, tail] & \end{tikzcd}$$

- Pushouts of polarized graphs exist.

$$\textbf{Graph} \to \textbf{PolGraph} \quad n \mapsto n^{\pm}$$

$$\textbf{PolGraph} \to \textbf{Graph} \quad n^{\pm}, n^{+}, n^{-}, n \mapsto n$$

Graph is a reflective subcategory of **PolGraph**.

L, K are polarized graphs.

$$L \;=\; \begin{tikzcd} {n_L}^{\pm} \arrow[loop above, "e_L"] \end{tikzcd} \qquad\qquad K \;=\; \begin{tikzcd} {n_{K,1}}^{+} \arrow[d, "e_K"'] \\ {n_{K,2}}^{-} \end{tikzcd}$$

$L \subseteq G$.

$$G \;=\; \begin{tikzcd} {n_L}^{\pm} \arrow[loop above, "e_L"] \arrow[loop below] \arrow[r, bend left] & {n'}^{\pm} \arrow[l, bend left] \arrow[loop above, "e'"] \end{tikzcd}$$

$K \subseteq D$.

$$D \;=\; \begin{tikzcd} {n_{K,1}}^{+} \arrow[d, shift right=2, "e_K"'] \arrow[d] \arrow[d, shift left=2] \arrow[dr, bend left] & \\ & {n'}^{\pm} \arrow[loop above, "e'"] \arrow[dl, bend left] \\ {n_{K,2}}^{-} & \end{tikzcd}$$

Polarized sesqui-pushout (PSqPO)

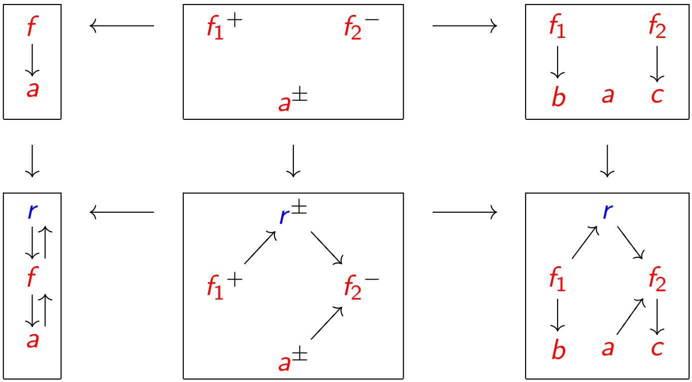
The LHS square is a final pullback complement of polarized graphs

$$
\begin{tikzcd}
L \arrow[d] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, dashed] & R \arrow[d, dotted] \\
G & D \arrow[l, dashed, "l_1"] \arrow[r, dotted, "r_1"'] & H
\end{tikzcd}
$$

+ FPBC of polarized graphs exist and are unique (up to iso)
+ polarization allows more flexible cloning

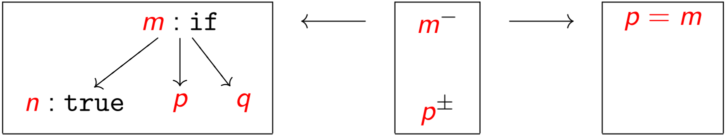
!!! In fact, only the interface is polarized!

Example: PSqPO

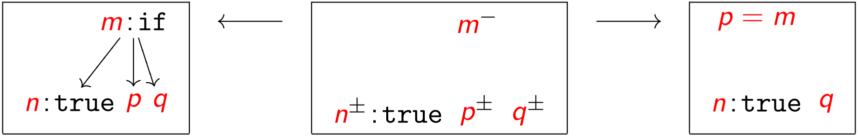


“if ...then...else...”

“Destructive” rules:



“Non-destructive” rules:



Conclusion

+ SqPO and PSqPO exist and are unique (up to iso)
+ SqPO and PSqPO are more general than DPO
- SqPO is not easy to define
- PSqPO is still less easy to define

CLIMT:

- A better understanding of PSqPO
- ...via a better understanding of SqPO?
- for various applications of “polarized” cloning
- ...involving some “complexified” graphs in the interface?