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 transformation

Algebraic graph transformation

Double-pushout (DPO)

Sesqui-pushout (SqPO)

Polarized sesqui-pushout (PSqPO)
Graph rewriting

$L, R, G, H$ are graphs.

For each rewrite rule:

$$L \rightarrow R$$

and each matching:

$$L \subseteq G$$

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

$$R \subseteq H$$
Example: term rewriting

\[
\begin{align*}
&\quad f \\
&\quad \downarrow \\
&\quad a \\
&\quad \downarrow \\
&\quad r \\
&\quad \downarrow \\
&\quad f \\
&\quad \downarrow \\
&\quad a \\
&\quad \downarrow \\
&\quad f \\
&\quad \downarrow \\
&\quad a \\
&\quad \downarrow \\
&\quad g \\
&\quad \downarrow \\
&\quad b \\
&\quad \downarrow \\
&\quad c \\
&\quad \downarrow \\
&\quad r \\
&\quad \downarrow \\
&\quad g \\
&\quad \downarrow \\
&\quad f \\
&\quad \downarrow \\
&\quad b \\
&\quad \downarrow \\
&\quad c \\
&\quad \downarrow \\
&\quad a
\end{align*}
\]
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.

$L = \begin{array}{c} e_L \\ n_L \end{array}$ \quad $R = \begin{array}{c} e_R \\ n_R \end{array}$

$L \subseteq G$.

$G = \begin{array}{c} e_L \\ n_L \end{array} \quad e' \quad \begin{array}{c} e'_R \\ n'_R \end{array}$

$R \subseteq H$, after rewriting.

$H = \begin{array}{c} e_R \\ n_R \end{array} \quad ?? \quad \begin{array}{c} ?? \\ n'_R \\ ?? \end{array}$
Example: What does \textit{replacing} mean?

\begin{align*}
\text{f} & \rightarrow \text{a} \\
\text{r} & \rightarrow \text{f} \rightarrow \text{a}
\end{align*}

\begin{align*}
\text{g} & \rightarrow \text{b} \rightarrow \text{c} \\
\text{r} & \rightarrow \text{g} \\
\text{b} & \rightarrow \text{c} \rightarrow \text{a}
\end{align*}

or...?
Outline

Graph transformation

Algebraic graph transformation

Double-pushout (DPO)

Sesqui-pushout (SqPO)

Polarized sesqui-pushout (PSqPO)
Algebraic graph transformation

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

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

Pushouts

Union

\[
\begin{array}{ccc}
X \cap Y & \subseteq & Y \\
\subseteq & & \subseteq \\
\downarrow & & \downarrow \\
X & \subseteq & X \cup Y
\end{array}
\]

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

\[
\begin{array}{ccc}
W & \rightarrow & Y \\
\downarrow & & \downarrow \\
X & \rightarrow & Z
\end{array}
\]

- When a pushout exists, it is unique (up to iso).
- Categories \textbf{Set} and \textbf{Graph} have pushouts.
PO of graphs

There is a **GRAPH OF GRAPHS**:

![Graph Diagram]

- Pushouts of graphs exist
  and they can be computed pointwise.
Example: a PO of graphs

\[ \xymatrix{ f \ar[d] \ar[r] & b \\
 a \ar[d] & a \\
 r \ar[r] & r \\
f \ar[d] & f \\
a & b \\
} \]
In this talk, every ??PO of graphs looks like:

```
DPO, SqPO, PSqPO

\[ \begin{array}{c}
L & \xrightarrow{l} & K & \xrightarrow{r} & R \\
\downarrow & & \downarrow & & \downarrow \\
G & \rightleftharpoons & D & \rightleftharpoons & H \\
\ \ \ l_1 & & \ \ \ \ r_1 \\
\end{array} \]
```
Outline

Graph transformation

Algebraic graph transformation

Double-pushout (DPO)

Sesqui-pushout (SqPO)

Polarized sesqui-pushout (PSqPO)
Double-pushout (DPO)

The LHS square is a pushout complement (POC)

\[
\begin{array}{c}
L \leftarrow l \\ \downarrow \\
G \leftarrow l_1 \\ \downarrow \\
D \leftarrow r_1 \\ \downarrow \\
H \\
\end{array}
\]

\[
\begin{array}{c}
K \rightarrow r \\ \downarrow \\
R \\
\end{array}
\]

+ Easy to understand: symmetric
+ Easy to define
+ Sound categorical base: adhesive categories
Example: DPO

\[
\begin{array}{ccc}
\begin{array}{c}
f \\
\downarrow \\
a
\end{array} & \leftrightarrow & \begin{array}{c}
f \\
a
\end{array} & \rightarrow & \begin{array}{c}
f \\
\ \ \\
b & a & c
\end{array}
\end{array}
\]

\[
\begin{array}{ccc}
\begin{array}{c}
r \\
\downarrow \\
f \\
\downarrow \\
a
\end{array} & \leftrightarrow & \begin{array}{c}
r \\
\downarrow \\
f \\
a
\end{array} & \rightarrow & \begin{array}{c}
r \\
\ \ \\
b & a & c
\end{array}
\end{array}
\]
Adhesive categories

- Definition of adhesive categories involves Van Kampen squares.

- Categories \textbf{Set} and \textbf{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

Graph transformation

Algebraic graph transformation

Double-pushout (DPO)

Sesqui-pushout (SqPO)

Polarized sesqui-pushout (PSqPO)
Sesqui-pushout (SqPO)

The LHS square is a final pullback complement (FPBC)

\[ L \leftarrow^l K \rightarrow^r R \]
\[ G \leftarrow_{l_1} D \rightarrow_{r_1} H \]

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

Intersection

\[
X \cap Y \subseteq \Downarrow \subseteq \rightarrow Y
\]

\[
\subseteq \\
\Downarrow \\
X \rightarrow Z
\]

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

\[
W \longrightarrow \longrightarrow \rightarrow Y
\]

\[
\Downarrow \\
\Downarrow \\
X \rightarrow Z
\]

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

```
\[f\]
\[\downarrow a\]
\[\downarrow\]
\[r\]
\[\downarrow f\]
\[\downarrow\]
\[a\]
```

```
\[f\]
\[\downarrow\]
\[r\]
\[\downarrow f\]
\[\downarrow\]
```
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:

\[ \begin{array}{c}
V^+ \\
\downarrow \\
V \\
\downarrow \\
V^-
\end{array} \quad \xrightarrow{s^+} \quad \xleftarrow{t^-} \quad E \]

- Pushouts of polarized graphs exist.

\[
\text{Graph} \rightarrow \text{PolGraph} \quad n \mapsto n^\pm
\]

\[
\text{PolGraph} \rightarrow \text{Graph} \quad n^\pm, n^+, n^-, n \mapsto n
\]

\text{Graph} is a reflective subcategory of \text{PolGraph}. 
$L$, $K$ are polarized graphs.

$L = e_L \ n_L^\pm$ \hspace{1cm} $K = n_K,1^+$

$L \subseteq G.$

$G = e_L \ n_L^\pm \hspace{1cm} n'_\pm$ \hspace{1cm} $e' \ n'_\pm$

$K \subseteq D.$

$D = n_K,1^+$ \hspace{1cm} $e_K \ n_K,2^-$ \hspace{1cm} $e' \ n'_\pm$
Polarized sesqui-pushout (PSqPO)

The LHS square is a final pullback complement of polarized graphs.

\[
\begin{align*}
L & \leftarrow I \rightarrow K \rightarrow R \\
\downarrow & \downarrow \downarrow \downarrow \\
G & \leftarrow D \rightarrow H
\end{align*}
\]

+ 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

\[
\begin{array}{c}
\begin{array}{c}
\left( f \downarrow a \right)
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\left( f_1^+ f_2^- \right)
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\left( f_1 \downarrow b \quad f_2 \downarrow c \right)
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\left( r \uparrow \uparrow f \downarrow a \right)
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\left( r^\pm \quad f_1^+ \quad f_2^- \quad a^\pm \right)
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\left( r \quad f_1 \downarrow b \quad f_2 \downarrow c \right)
\end{array}
\end{array}
\end{array}
\]
“if ...then...else...”

“Destructive” rules:

```
m: if
    n: true
    p
    q
```

```
m
    m
    p±
```

```
p = m
```

“Non-destructive” rules:

```
m: if
    n: true
    p
    q
```

```
m
    m
    p±
```

```
p = m
```

```
n±: true
    p±
    q±
```

```
n: true
    q
```
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?