Accretive Computation of Global Transformations

A. Fernandez  L. Maignan  A. Spicher

Univ Paris Est Creteil, LACL, 94000, Creteil, France
firstname.lastname@u-pec.fr

RAMICS 2021
Global Transformations (GT): general frame of work

- Describe local, deterministic and synchronous systems
Global Transformations (GT): general frame of work
- Describe local, deterministic and synchronous systems
- Cellular automata

\[
\begin{align*}
\text{abbab} & \\
\downarrow & \\
\text{babababbab} & 
\end{align*}
\]
Global Transformations (GT): general frame of work

- Describe local, deterministic and synchronous systems
- Cellular automata, L-systems

\[
\begin{align*}
\text{abbab} & \quad \implies \quad \text{babababbab}
\end{align*}
\]
Global Transformations (GT): general frame of work
- Describe local, deterministic and synchronous systems
- Cellular automata, L-systems, causal graph dynamics
Global Transformations (GT): general frame of work
- Describe local, deterministic and synchronous systems
- Cellular automata, L-systems, causal graph dynamics
- Different structure / space
Global Transformations (GT): general frame of work

- Describe local, deterministic and synchronous systems
- Cellular automata, L-systems, causal graph dynamics

Different structure / space

Same rewriting procedure

\{ \text{pattern matching} \}

Local application

Result reconstruction
Context

- Global Transformations (GT): general frame of work
  - Describe local, deterministic and synchronous systems
  - Cellular automata, L-systems, causal graph dynamics
- Different structure / space
- Same rewriting procedure
  - pattern matching

\{ \text{\textbullet} \rightarrow \text{\textbullet}, \ldots \}
Context

- Global Transformations (GT): general frame of work
  - Describe local, deterministic and synchronous systems
  - Cellular automata, L-systems, causal graph dynamics
- Different structure / space
- Same rewriting procedure
  - pattern matching
  - Local application

\[
\{ \text{element} \rightarrow \text{element}, \ldots \}
\]
Global Transformations (GT): general frame of work

- Describe local, deterministic and synchronous systems
- Cellular automata, L-systems, causal graph dynamics

Different structure / space

Same rewriting procedure

- pattern matching
- Local application
- Result reconstruction

\[
\begin{array}{c}
\{ \text{cell} \rightarrow \text{cell}, \ldots \} \\
\end{array}
\]
Scope of this work

Define global transformations of presheaves

- Presheaves: generalized graphs
- This presentation: focus only on graphs

Give an **online algorithm** for computing GT:
Scope of this work

Define global transformations of presheaves

► Presheaves: generalized graphs
► This presentation: focus only on graphs

Give an online algorithm for computing GT:
Scope of this work

Define global transformations of presheaves

▶ Presheaves: generalized graphs
▶ This presentation: focus only on graphs

Give an online algorithm for computing GT:
Scope of this work

Define global transformations of presheaves

- Presheaves: generalized graphs
- This presentation: focus only on graphs

Give an online algorithm for computing GT:
Scope of this work

Define global transformations of presheaves

- Presheaves: generalized graphs
- This presentation: focus only on graphs

Give an online algorithm for computing GT:

Diagram showing a sequence of triangle graphs.
Scope of this work

Define global transformations of presheaves

- Presheaves: generalized graphs
- This presentation: focus only on graphs

Give an online algorithm for computing GT:
Scope of this work

Define global transformations of presheaves

- Presheaves: generalized graphs
- This presentation: focus only on graphs

Give an online algorithm for computing GT:
Scope of this work

Define global transformations of presheaves
  ▶ Presheaves: generalized graphs
  ▶ This presentation: focus only on graphs

Give an online algorithm for computing GT:
Scope of this work

Define global transformations of presheaves

- Presheaves: generalized graphs
- This presentation: focus only on graphs

Give an online algorithm for computing GT:
Scope of this work

Define global transformations of presheaves

▶ Presheaves: generalized graphs
▶ This presentation: focus only on graphs

Give an online algorithm for computing GT:
Scope of this work

Define global transformations of presheaves

- Presheaves: generalized graphs
- This presentation: focus only on graphs

Give an online algorithm for computing GT:

![Diagram of two triangle fractals]
Scope of this work

Define global transformations of presheaves

- Presheaves: generalized graphs
- This presentation: focus only on graphs

Give an online algorithm for computing GT:
Scope of this work

Define global transformations of presheaves
- Presheaves: generalized graphs
- This presentation: focus only on graphs

Give an online algorithm for computing GT:
History:

- L. M., A. S. **Global Graph Transformations**. ICGT 2015
- *Triangular mesh refinement Rule-based Specification*

SIGGRAPH 98 Course Notes
History:

- L. M., A. S. **Global Graph Transformations.** ICGT 2015
- Triangular mesh refinement **Rule-based Specification**

- Simple, but what happens on *overlaps*?
History:

- L. M., A. S. **Global Graph Transformations**. ICGT 2015
- Triangular mesh refinement **Rule-based Specification**

- Simple, but what happens on overlaps?

This is what we want!
History:

- L. M., A. S. **Global Graph Transformations**. ICGT 2015
- *Triangular mesh refinement Rule-based Specification*

- Simple, but what happens on overlaps?

Also compliant with the rules...
Solution

How to solve this ambiguity?
Solution

How to solve this ambiguity?

- Pattern-matching
Solution

How to solve this ambiguity?

▶ Pattern-matching ⇒ Local results
Solution

How to solve this ambiguity?

- Pattern-matching $\Rightarrow$ Local results
- Add a rule
Solution

How to solve this ambiguity?

- Pattern-matching ⇒ Local results
- Add a rule + inclusions for overlap
Solution

How to solve this ambiguity?

- Pattern-matching $\Rightarrow$ Local results $\Rightarrow$ Reconstruction
- Add a rule + inclusions for overlap

![Diagram showing the process of solving ambiguity with pattern-matching and local results leading to reconstruction, along with rule and inclusion adjustments.](image-url)
The category $G_M$

Structure for “thing being part of other thing”?

- Preorder
The category $G_M$

Structure for “thing being part of other thing at some place”?

- Preorder $\Rightarrow$ Category
The category $G_M$

Structure for “thing being part of other thing at some place”?

- Preorder $\Rightarrow$ Category

Work in $G_M$: category of graphs with inclusions

- **Objects**: Graphs (directed multigraphs)
- **Arrows**: Inclusions of graphs (monomorphisms)

$\subset G$: category of graphs with graph homomorphism
Functor

- Transformation respects arrow ($\simeq$ monotony): 
The way the **local output** occur in the **global output** depends on the way the **local input** occur as a part of the **global input**
Functor

- Transformation respects arrow (≃ monotony):
  The way the **local output** occur in the **global output** depends on
  the way the **local input** occur as a part of the **global input**

\[
T(g) \circ T(f) = T(g \circ f)
\]
A rule system $T$ is tuple $\langle \Gamma, L, R \rangle$ where $\Gamma$ is a category, $L, R : \Gamma \rightarrow G_M$ are functors, and $L$ is full and faithfull ($\simeq$ injective).

Equivalently: Partial functor $P : G_M \rightarrow G_M := R \circ L^{-1}$
Computation

Definition: Global Rewrite Step

The rewrite step is the functor $\overline{T} : G_M \rightarrow G$ given by:

$$\overline{T}(-) = \text{Colim}(U \circ P \circ L \circ \Pi_{L/-})$$

with $U : G_M \rightarrow G$ forgetful functor.
Global Transformation

No colimits in $G_M$ . . .

- Taking colimits in $G$ gives $\bar{T} : G_M \to G$
- Rule systems do not preserve injectivity !

**Definition: Global Transformation**

A global transformation $T$ is a rule system such that $\bar{T} : G_M \to G$ factors through $U : G_M \to G$. In this case $T : G_M \to G_M$ is the functor such that $U \circ T = \bar{T}$.

Global transformation: $T : G_M \to G_M$ is iterable !

- Triangular mesh refinement, sierpinsky ...
Not global transformations

Consider this rule system:

\[
\begin{align*}
\text{\longrightarrow} & \text{\longrightarrow} \\
\text{\longrightarrow} & \text{\longrightarrow}
\end{align*}
\]

And the inclusion of graphs \( m \):
Consider this rule system:

\[
\begin{align*}
\text{Graph 1} & \quad \Rightarrow \quad \text{Graph 2} \\
& \quad \Rightarrow \quad \text{Graph 3}
\end{align*}
\]

And the inclusion of graphs \( m \):

\[
\begin{align*}
m & \quad \uparrow \\
\text{Graph 1} & \quad T \quad \Rightarrow \quad \text{Graph 2}
\end{align*}
\]
Not global transformations

Consider this rule system:

And the inclusion of graphs $m$:
Not global transformations

Consider this rule system:

And the inclusion of graphs $m$:

No inclusion $T(m)$: $T$ not a global transformation!

Problem: Find a local way to check if rule system is GT?

(i.e. by just looking at the rules)
Computing with GT

Define *generic* algorithm:
- Big operations $\Rightarrow$ small generic operations
- Breaks down global step $\Rightarrow$ sequence of local steps

Preserving injectivity for sequence of local steps?
- Link with global transformation?

We restrict to connected finite graphs and finite rule systems.
Example run

Online algorithm:
Example run

Online algorithm:
Example run

Online algorithm:
Example run

Online algorithm:

![Diagram of an online algorithm](image-url)
Example run

Online algorithm:

\[
\begin{align*}
\text{Diagram 1} & \quad \rightarrow \\
\text{Diagram 2} & \quad \rightarrow \\
\text{Diagram 3} & \quad \rightarrow \\
\end{align*}
\]
Example run

Online algorithm:

![Diagram of online algorithm](image-url)
Example run

Online algorithm:
Example run

Online algorithm:
Example run

Online algorithm:
Example run

Online algorithm:

[Diagram of an online algorithm process]

A, Fernandez (UPEC)  Accretive Computation of GT  RAMICS 2021 13 / 25
Example run

Online algorithm:
Example run

Online algorithm:
Example run

Online algorithm:
Example run

Online algorithm:
Example run

Online algorithm:
Example run

Online algorithm:
Example run

Online algorithm:
Proposition

The comma category $L/g$ is isomorphic to a **preorder** (if no automorphism of rule $\Rightarrow$ **poset**).

- Maximal occurrences gives the **maximal local results**
- Other occurrences used to glue these results

Maximal occurrences **constructed** with breadth-first search.
Proposition

The comma category $L/g$ is isomorphic to a preorder (if no automorphism of rule $\Rightarrow$ poset).

- Maximal occurrences gives the maximal local results
- Other occurrences used to glue these results

Maximal occurrences constructed with breadth-first search.
Pattern matching

**Proposition**

The comma category $L/g$ is isomorphic to a **preorder** (if no automorphism of rule $\Rightarrow$ **poset**).

- Maximal occurrences gives the **maximal local results**
- Other occurrences used to **glue** theses results

Maximal occurrences **constructed** with breadth-first search.
Proposition

The comma category $L/g$ is isomorphic to a preorder (if no automorphism of rule $\Rightarrow$ poset).

- Maximal occurrences gives the maximal local results
- Other occurrences used to glue these results

Maximal occurrences constructed with breadth-first search.
Pattern matching

Proposition

The comma category $L/g$ is isomorphic to a **preorder** (if no automorphism of rule $\Rightarrow$ **poset**).

- Maximal occurrences give the **maximal local results**
- Other occurrences used to **glue** these results

Maximal occurrences **constructed with breadth-first search**.
Gluing partial results

Compute *gluing along suture*:
Gluing partial results

Compute gluing along suture:
**Partial computations**

**Definition: Partial decomposition**
Given a graph $g$, a partial decomposition is a subset $M$ of maximal occurrences of $L/g$ such that the restriction $\tilde{M}$ of $L/g$ to $M$ and morphisms into $M$ is connected.

$\overline{L/g}$: category of partial decompositions with set inclusions.

**Definition: Partial computation functor**
Let $\tilde{T}_g : \overline{L/g} \to G$ given by $T(\tilde{M}) = \text{Colim}(U \circ P \circ L \circ \Pi_{L/g} \mid \tilde{M})$.

**Definition: Accretive rule system**
$T$ is accretive iff. for any $g$, $\tilde{T}_g$ factors through $U : G_M \to G$.

- Accretive $\nRightarrow$ Global transformation
**Generalized pushout**

**Definition: suture diagram**

A suture diagram \( D : I \to G_M \) is a diagram (functor) with the following shape:

\[
\begin{array}{c}
D(m_1) & D(i_k) & D(j_1) & D(m_2) \\
D(i_1) & \downarrow & \uparrow & D(j_k) \\
D(n_1) & \cdots & D(n_k)
\end{array}
\]

The gluing is called a *generalized pushout*:

**Definition: generalized pushout**

Given a suture diagram \( D : I \to G_M \) its generalized pushout is

\[ \text{Colim}(U \circ D) \]

- **Colimit in** \( G \): does not preserve injectivity
Accretive global transformations

Pattern matching $\implies$ sequence $\langle r_1, \ldots, r_k \rangle$ of maximal local results.
Sequence of gluings $\implies$ partial results

$$r_1 = P_1 \xrightarrow{i_1} P_2 \xrightarrow{i_2} P_3 \xrightarrow{i_3} P_4 \xrightarrow{i_4} \cdots \xrightarrow{\cdots} P_k$$

\[ r_2 \overset{j_1}{\longrightarrow} r_3 \overset{j_2}{\longrightarrow} r_4 \overset{j_3}{\longrightarrow} r_5 \overset{j_4}{\longrightarrow} \cdots \]

**Proposition**

Given path $\tilde{M}_1 \subseteq \cdots \subseteq \tilde{M}_k$ in $\tilde{L}/\tilde{g}$ where:

$\tilde{M}_1 = \{ r_1 \}, \tilde{M}_{i+1} = \tilde{M}_i \cup \{ r_{i+1} \}$.

Then for any $i \in \{ 1, \ldots, k \}$, $\tilde{T}_g(\tilde{M}_i) = P_i$.

▶ If some gluing is non-injective then $T$ is not acccretive.
Non-accrative global transformations

Consider the following rule system:
Non-accretive global transformations

Consider the following rule system:

\[ \ldots \rightarrow \begin{array}{c}
\bullet & \bullet & \bullet \\
\uparrow & \uparrow & \\
\bullet & \bullet & \bullet
\end{array} \rightarrow \begin{array}{c}
\bullet & \bullet & \bullet \\
\uparrow & \uparrow & \\
\bullet & \bullet & \bullet
\end{array} \rightarrow \ldots \]

Removes isolated vertices $\Rightarrow$ Global Transformation
Consider the following rule system:

Removes isolated vertices $\Rightarrow$ Global Transformation

Not accretive:

A, Fernandez (UPEC)
Non-accretive global transformations

Consider the following rule system:

... ...

Removes isolated vertices \( \Rightarrow \) Global Transformation

Not accretive:

\[ T \]
Non-accretive global transformations

Consider the following rule system:

... → ...

Removes isolated vertices ⇒ Global Transformation
Not accretive:

\[ T \]

A. Fernandez (UPEC)
Issue with last example

Find a criterion on rule systems which implies GT?

- Criterion on rule systems which implies accretive GT
Issue with last example

Find a criterion on rule systems which implies GT?

- Criterion on rule systems which implies accretive GT
Issue with last example

Find a criterion on rule systems which implies GT?

- Criterion on rule systems which implies accretive GT

The three local results must be **glued** at the same time!

- This **gluing** is not induced by a **suture** of two local result!
- Define a criterion which forbids that!
Definition: Incrementality

T is incremental if for any $\gamma_1 \xrightarrow{i_1} \gamma \xleftarrow{i_2} \gamma_2$ in $\Gamma$, any graph $e \in \{\cdot, \cdot \to \cdot\}$, and any $R(\gamma_1) \xleftarrow{x_1} e \xrightarrow{x_2} R(\gamma_2)$ such that $R(i_1) \circ x_1 = R(i_2) \circ x_2$, there are $\gamma_1 \xrightarrow{\pi_1} \gamma' \xleftarrow{\pi_2} \gamma_2$ and $x : e \to R(\gamma')$ such that:

Theorem

If a rule-system is incremental, then it is an accretive GT.
Conclusion & Perspectives

Global transformations: Synchronous rewriting
  ▶ Works on wide variety of structures

Rewriting algorithm: Sequentialization of GT
  ▶ Designed to be generic

Preserving injections:

<table>
<thead>
<tr>
<th></th>
<th>non-incr.</th>
<th>incr.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>non-G.T.</td>
<td>G.T.</td>
</tr>
<tr>
<td>non-accretive</td>
<td>ex. Fig. 3a</td>
<td>ex. Fig. 3c</td>
</tr>
<tr>
<td>accretive</td>
<td>ex. Fig. 3b</td>
<td>ex. Fig. 3d</td>
</tr>
</tbody>
</table>
Perspectives

Non deterministic computations
  ▶ Bicategory of open functors (arxiv)

Algorithm in other categories?
  ▶ Cellular automata, L-Systems, CGD
  ▶ (\mathcal{M})-adhesive categories, topos?

Pattern matching like Knuth-Morris-Pratt
  ▶ Break down pattern matching

---

1Y. V. Srinivas, A Sheaf-Theoretic approach to pattern matching, 1993
# Table of Contents

- Introduction
- Global transformations
- Accretive computation
- Incremental criterion
- Conclusion
- Appendix
Rule system

**Definition: Rule system**

A *rule system* $T$ is tuple $\langle \Gamma, L, R \rangle$ where $\Gamma$ is a category, $L, R : \Gamma \to G_M$ are functors, and $L$ is full and faithfull ($\simeq$ injective).

Equivalently: Partial functor $P : G_M \to G_M := R \circ L^{-1}$
Perspectives

Non deterministic computations

Figure: Left: ND on labels, Right: ND on structure
Not global transformations

Consider this rule system:

And the inclusion of graphs $m$:

Let’s compute:
Not global transformations

Consider this rule system:

And the inclusion of graphs $m$:

Let’s compute:
Not global transformations

Consider this rule system:

And the inclusion of graphs $m$:

Let’s compute:
Not global transformations

Consider this rule system:

And the inclusion of graphs $m$:

Let’s compute:
Not global transformations

Consider this rule system:

And the inclusion of graphs $m$:

Let's compute:
Not global transformations

Consider this rule system:

And the inclusion of graphs $m$:

Let's compute:
Not global transformations

Consider this rule system:

And the inclusion of graphs $m$:

Let’s compute:
Not global transformations

Consider this rule system:

And the inclusion of graphs $m$:

Let’s compute:
Not global transformations

Consider this rule system:

And the inclusion of graphs $m$:

Let’s compute:
Not global transformations

Consider this rule system:

And the inclusion of graphs $m$:

No inclusion $T(m)$: $T$ not a global transformation!

Problem: Find a local way to check if rule system is GT? (ie. by just looking at the rules)
Amalgamation

\[
\begin{array}{c}
\text{Accretive Computation of GT} \\
\text{RAMICS 2021} \\
25 / 25
\end{array}
\]
Global transformations

Pattern matching

Local applications

Result construction
Rules
Not global transformations
Not global transformation 2
Example: L-systems

**Definition**

A (deterministic context-free) L-system $L : \Sigma^* \to \Sigma^*$ is a function given by:

- an alphabet $\Sigma$ and
- a function $P : \Sigma \to \Sigma^*$ such that
- $L(a_0a_1 \ldots a_n) = P(a_0)P(a_1) \ldots P(a_n)$, for any $a_0a_1 \ldots a_n \in \Sigma^*$

Example:

$\Sigma = \{a, b\}$

$P = \{a \mapsto \rightarrow b, b \mapsto \rightarrow ab\}$
Example: L-systems

**Definition**

A (deterministic context-free) L-system $L : \Sigma^* \rightarrow \Sigma^*$ is a function given by:

- an alphabet $\Sigma$ and
- a function $P : \Sigma \rightarrow \Sigma^*$ such that

$L(a_0a_1 \ldots a_n) = P(a_0)P(a_1) \ldots P(a_n)$, for any $a_0a_1 \ldots a_n \in \Sigma^*$

**Example:**

$$\Sigma = \{a, b\} \quad \text{abbab}$$

$$P = \begin{cases} a \mapsto b \\ b \mapsto ab \end{cases}$$
Example: L-systems

Definition

A (deterministic context-free) L-system \( L : \Sigma^* \rightarrow \Sigma^* \) is a function given by:

- an alphabet \( \Sigma \) and
- a function \( P : \Sigma \rightarrow \Sigma^* \) such that

\[
L(a_0a_1 \ldots a_n) = P(a_0)P(a_1) \ldots P(a_n), \text{ for any } a_0a_1 \ldots a_n \in \Sigma^*
\]

Example:

\[
\Sigma = \{a, b\}
\]

\[
P = \begin{cases} 
  a &\mapsto b \\
  b &\mapsto ab
\end{cases}
\]

\[
abbbab \\
\downarrow \\
bababbabbab
\]
Example: L-systems

Definition

A (deterministic context-free) L-system $L : \Sigma^* \rightarrow \Sigma^*$ is a function given by:

- an alphabet $\Sigma$ and
- a function $P : \Sigma \rightarrow \Sigma^*$ such that
- $L(a_0a_1 \ldots a_n) = P(a_0)P(a_1) \ldots P(a_n)$, for any $a_0a_1 \ldots a_n \in \Sigma^*$

Example:

$$\Sigma = \{ a, b \}$$

$$P = \begin{cases} 
    a &\mapsto b \\
    b &\mapsto ab 
\end{cases}$$

$\scriptstyle \begin{array}{c}
\text{abbab} \\
\downarrow \\
\text{babababbab} \\
\downarrow \\
\text{abbabbbabababbab}
\end{array}$
Rewriting: Decomposition and recomposition

- Notion of “thing” is part of other “thing”
- Preorder structure
  - Reflexivity
  - Transitivity
Rewriting: Decomposition and recomposition

- Notion of “thing” is part of other “thing”
- Preorder structure
  - Reflexivity
  - Transitivity
Category

Rewriting: Decomposition and recomposition

- Notion of "thing" is part of other "thing" at some "place"
- Category
  - Identity
  - Composition