Global Transformations: A new formalism for rewriting system

L3 Internship Bibliography

MOUSSAOUI REMIL Naïm

Laboratoire d'Algorithmique, Complexité et Logique
Université Paris Est-Créteil
Supervisor: SPICHER Antoine

December 15, 2021

Table of Contents

Table of Contents

Rewriting System

Rewriting system (Informal)

System than can change in accord to some local rules.

Example:

- Biological System with L-System
- Graph rewriting
- Code Generation
- Particle System

Example: Lidenmayer Systems

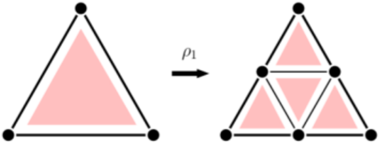
Definition

A $D0L$ system consists of :

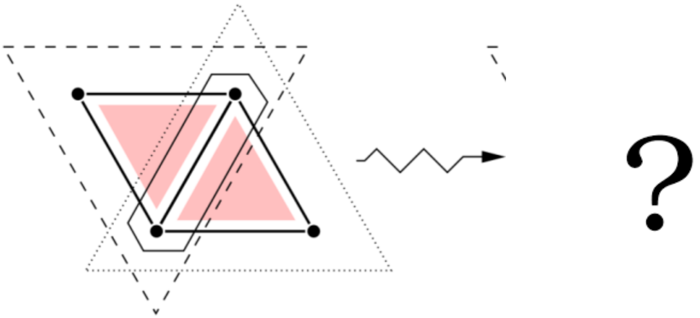
- an alphabet Σ_L
- a mapping P_L: $\Sigma_L \to \Sigma_L^*$

Example : "Filamentous organism"
$\Sigma_L = \{a, b\}$, P_L : $a \mapsto b$ $b \mapsto ab$
$abbab \Rightarrow bababbab \Rightarrow abbabbababbab \Rightarrow \dots$

Triangle of Sierpinski Rewriting System

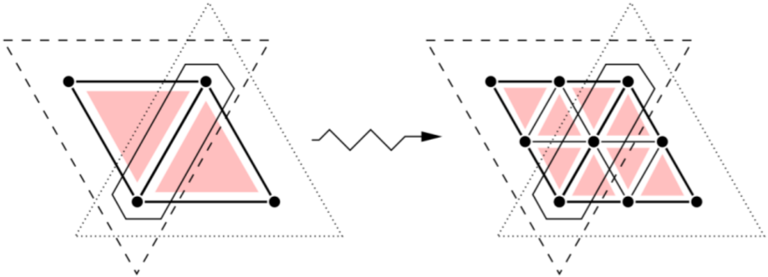


The Overlapping Problem 2



The Overlapping Problem 3

Ideally:



Objective

Objective

We want a global model which is :

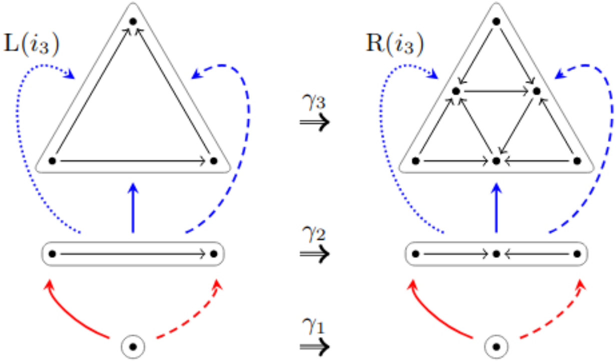
- **Local**
- **Synchronous**
- **Deterministic**

Table of Contents

The idea

Idea

- Define the rules systems with the inclusions of the lhs
- Composition of rules



Category

Goal:

- formalise the idea of a thing being a part an other
- capture transitivity of this notion

Definition (Category)

A category C consist of:
- a collection C_{ob} of elements called **object**
- for any $a, b \in C_{ob}$ a set $Arr_C(a, b)$ whose elements are called **arrows**
- for every a, b, c a **composition law** $Arr_C(b, c) \; x \; Arr_C(a, b) \mapsto Arr_C(a, c)$ which is associative
- for every x, an **identity** arrow id_x

Examples

Basic examples of categories:

- Set: category of set
- Grp: category of groupe
- Top: category of topology
- Graph: category of graph

Designing and Respecting Locality 1

Question: For a category C describing a locality structure , what does it mean for a transformation F on C to respect locality ?

Definition (Functor)

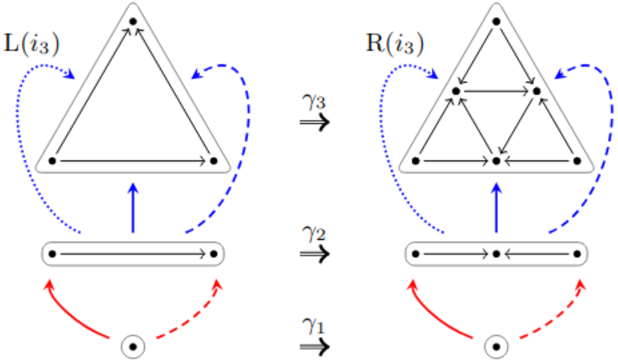
A functor $F : C \mapsto D$:
- a map $F_{ob} : C_{ob} \mapsto D_{ob}$
- a map $F_{arr} : Arr_C(a, b) \mapsto Arr_D(F_{ob}(a), F_{ob}(b))$ for every a, b

These maps have to preserve identity and composition

Designing and Respecting Locality 2

Role of arrows

- Rule system is a category
- We need to maps rules in the category C



The Global Transformation

Definition (Global Transformation)

A global transformation T consists of :

- a category C (output,input, hs)
- a category Γ whose object are rules $\phi =< l, r >$ and arrows are inclusions
- a fully faithful injective functor $L : \Gamma \mapsto C$ "lhs"
- a functor $R : \Gamma \mapsto C$ "rhs"

The construction of the output is done by categorical operation on T

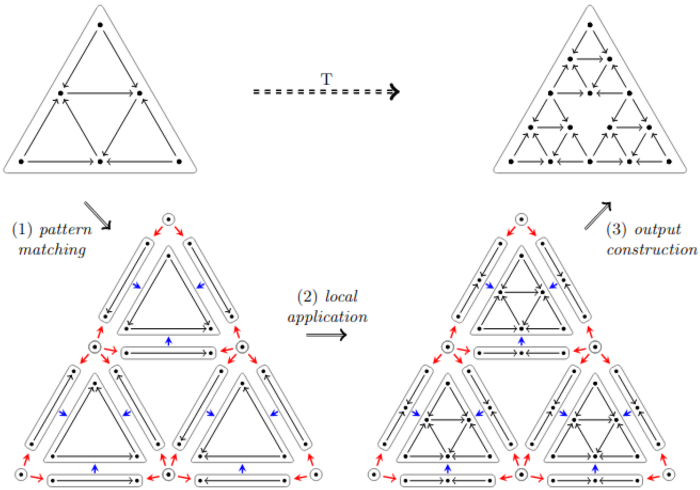
Construction of the output 1

How do we process?
In three steps :

- **Pattern Matching**: where are the lhs in the input
- **Application of local rules**
- **Construct the output**

How can we formalise that ?

Construction of the output 2



Pattern Matching

Goal: Distinguish each lhs in the input

Rule Instance

Input X and a rule γ

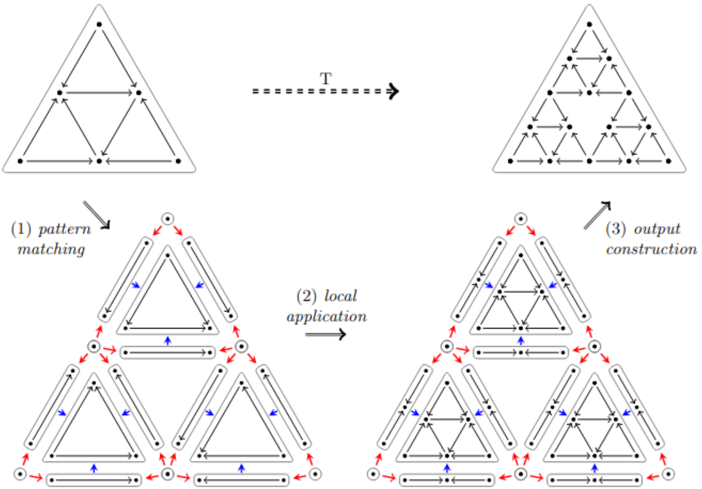
A Rule Instance of γ is a pair: $< \gamma$, p: $L(\gamma) \mapsto X >$

Comma Category L/X

Let L/X be the category that has:

- for object : every rules instances of γ (for $\gamma \in \Gamma$)
- for arrow from $< \gamma_1, p_1 >$ to $< \gamma_2, p_2 >$ every $g : \gamma_1 \mapsto \gamma_2$ such that $p_1 = p_2 \circ L(g)$

Visualisation of the Comma Category



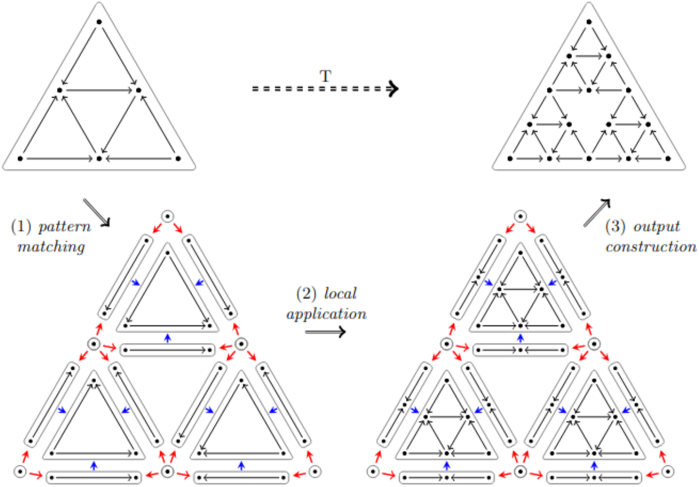
Local Application

Goal: Rewrite each lhs into rhs

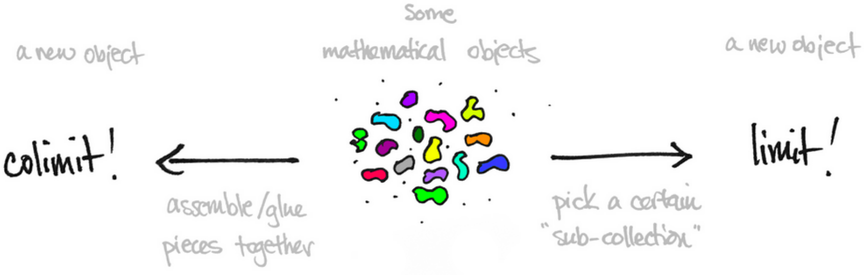
- Local application is achieved by : $R_{ob} \circ Proj_{L/X}$
- (P): inclusions between rhs is justified by an inclusion between corresponding lhs
- (P) $=>$ Structure of inclusion is preserved after the local application

Local Application 2

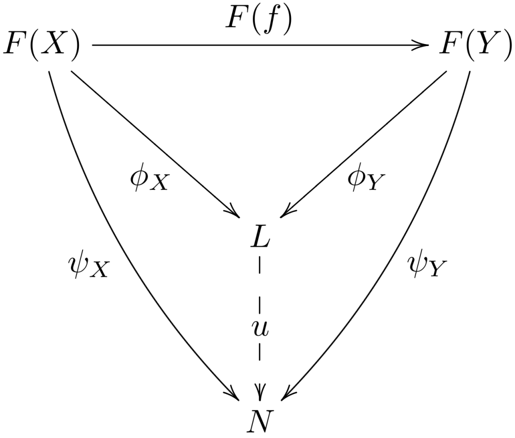
Visualisation of the Local Application



Construction of the output 1



Intuition:



Construction of the output 3

Definition

Given a global transformation T, the result $T(X)$ is an object $T(X) = Colim(R_{ob} \circ Proj_{L/X})$

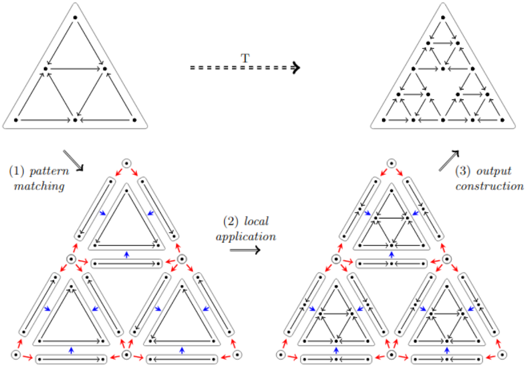


Table of Contents

Conclusion

Conclusion

Done:

- Formalism respect the pre-condition
- Capture different data-structure

Future works:

- Benchmark vs other framework
- Extends the design-tool web app