Parallel Graph Rewriting with Overlapping Rules

Rachid Echahed and Aude Maignan

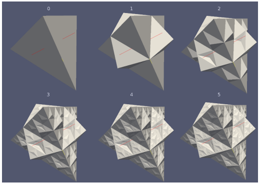
University Grenoble Alpes
Grenoble Informatics Laboratory
Applied Mathematics and Computer Science laboratory Jean Kuntzmann

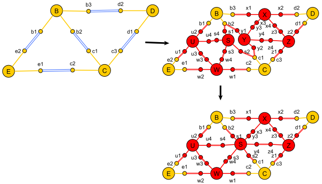
LPAR-21 2017, May

Table of Contents

Introduction

- Compute discrete bio-inspired systems (Cellular automata, L-systems)
- Speed up sequential rewriting





Algorithmic approach using

- pregraphs
- rewriting modulo

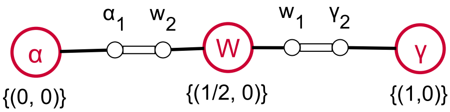
Table of Contents

Table of Contents

Pregraph definition

A pregraph H is a tuple $H = (\mathcal{N}_H, \mathcal{P}_H, \mathcal{PN}_H, \mathcal{PP}_H, \mathcal{A}_H, \lambda_H)$

- $\mathcal{N}_H$ Nodes
- $\mathcal{P}_H$ Ports
- $\mathcal{PN}_H$ Port-Node connections
- $\mathcal{PP}_H$ Port-Port connections
- Attributes : λ_H is a function $\lambda_H : \mathcal{P}_H \uplus \mathcal{N}_H \to 2^{\mathcal{A}_H}$

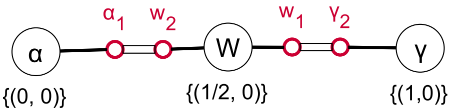


$\mathcal{N}_H = \{\alpha, \gamma, W\}$

Pregraph definition

A pregraph H is a tuple $H = (\mathcal{N}_H, \mathcal{P}_H, \mathcal{PN}_H, \mathcal{PP}_H, \mathcal{A}_H, \lambda_H)$

- $\mathcal{N}_H$ Nodes
- $\mathcal{P}_H$ Ports
- $\mathcal{PN}_H$ Port-Node connections
- $\mathcal{PP}_H$ Port-Port connections
- Attributes : λ_H is a function $\lambda_H : \mathcal{P}_H \uplus \mathcal{N}_H \rightarrow 2^{\mathcal{A}_H}$

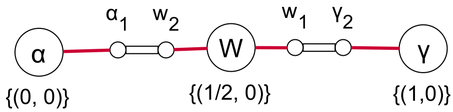


$\mathcal{P}_H = \{\alpha_1, w_2, w_1, \gamma_2\}$

Pregraph definition

A pregraph H is a tuple $H = (\mathcal{N}_H, \mathcal{P}_H, \mathcal{PN}_H, \mathcal{PP}_H, \mathcal{A}_H, \lambda_H)$

- $\mathcal{N}_H$ Nodes
- $\mathcal{P}_H$ Ports
- $\mathcal{PN}_H$ Port-Node connections
- $\mathcal{PP}_H$ Port-Port connections
- Attributes : λ_H is a function $\lambda_H : \mathcal{P}_H \uplus \mathcal{N}_H \rightarrow 2^{\mathcal{A}_H}$

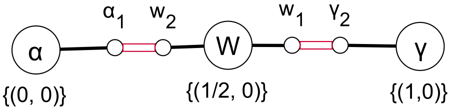


$\mathcal{PN}_H = \{(\alpha_1, \alpha), (w_1, W), (w_2, W), (\gamma_2, \gamma)\}$

Pregraph definition

A pregraph H is a tuple $H = (\mathcal{N}_H, \mathcal{P}_H, \mathcal{PN}_H, \mathcal{PP}_H, \mathcal{A}_H, \lambda_H)$

- $\mathcal{N}_H$ Nodes
- $\mathcal{P}_H$ Ports
- $\mathcal{PN}_H$ Port-Node connections
- $\mathcal{PP}_H$ Port-Port connections
- Attributes : λ_H is a function $\lambda_H : \mathcal{P}_H \uplus \mathcal{N}_H \rightarrow 2^{\mathcal{A}_H}$



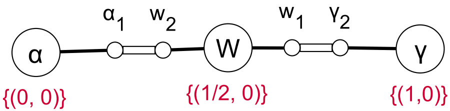
$\mathcal{PP}_H$ reduced to its non symmetric port-port connection is $\mathcal{PP}_H = \{(\alpha_1, w_2), (w_1, \gamma_2)\}$.

Pregraph definition

A pregraph H is a tuple $H = (\mathcal{N}_H, \mathcal{P}_H, \mathcal{PN}_H, \mathcal{PP}_H, \mathcal{A}_H, \lambda_H)$

- $\mathcal{N}_H$ Nodes
- $\mathcal{P}_H$ Ports
- $\mathcal{PN}_H$ Port-Node connections
- $\mathcal{PP}_H$ Port-Port connections

- Attributes : λ_H is a function $\lambda_H : \mathcal{P}_H \uplus \mathcal{N}_H \rightarrow 2^{\mathcal{A}_H}$

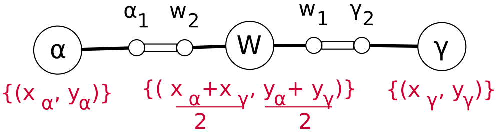


$\mathcal{A}_H = (\mathbb{Q}[x, y]; +, /)$, $\lambda_H(\alpha_1) = \lambda_H(w_2) = \lambda_H(w_1) = \lambda_H(\gamma_2) = \emptyset$
$\lambda_H(\alpha) = \{(0, 0)\}$, $\lambda_H(W) = \{(\frac{1}{2}, 0)\}$, $\lambda_H(\gamma) = \{(1, 0)\}$

Pregraph definition

A pregraph H is a tuple $H = (\mathcal{N}_H, \mathcal{P}_H, \mathcal{PN}_H, \mathcal{PP}_H, \mathcal{A}_H, \lambda_H)$

- $\mathcal{N}_H$ Nodes
- $\mathcal{P}_H$ Ports
- $\mathcal{PN}_H$ Port-Node connections
- $\mathcal{PP}_H$ Port-Port connections
- Attributes : λ_H is a function $\lambda_H : \mathcal{P}_H \uplus \mathcal{N}_H \to 2^{\mathcal{A}_H}$



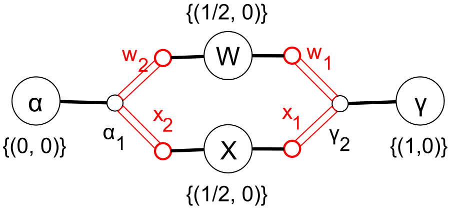
$\mathcal{A}_H = (\mathbb{Q}[x,y]; +, /)$, $\lambda_H(\alpha_1) = \lambda_H(w_2) = \lambda_H(w_1) = \lambda_H(\gamma_2) = \emptyset$
$\lambda_H(\alpha) = \{(x_\alpha, y_\alpha)\}$, $\lambda_H(\gamma) = \{(x_\gamma, y_\gamma)\}$, $\lambda_H(W) = \{(\frac{x_\alpha + x_\gamma}{2}, \frac{y_\alpha + y_\gamma}{2})\}$

Graph definition

A graph, G, is a pregraph $G = (\mathcal{N}, \mathcal{P}, \mathcal{PN}, \mathcal{PP}, \mathcal{A}, \lambda)$ such that :

(i) $((p_1, p_2) \in \mathcal{PP}$ and $(p_1, p_3) \in \mathcal{PP}) \implies p_2 = p_3$ and $\forall p \in \mathcal{P}, (p, p) \notin \mathcal{PP}$.

(ii) $((p, n_1) \in \mathcal{PN} \textit{ and } (p, n_2) \in \mathcal{PN}) \implies n_1 = n_2$.

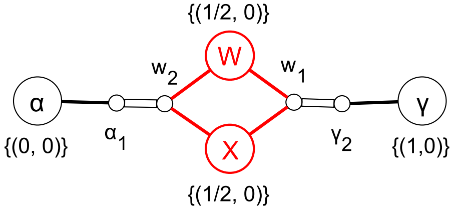


Graph definition

A graph, G, is a pregraph $G = (\mathcal{N}, \mathcal{P}, \mathcal{PN}, \mathcal{PP}, \mathcal{A}, \lambda)$ such that :

(i) $((p_1, p_2) \in \mathcal{PP}$ and $(p_1, p_3) \in \mathcal{PP}) \implies p_2 = p_3$ and $\forall p \in \mathcal{P}, (p, p) \notin \mathcal{PP}$.

(ii) $((p, n_1) \in \mathcal{PN}$ *and* $(p, n_2) \in \mathcal{PN}) \implies n_1 = n_2$.

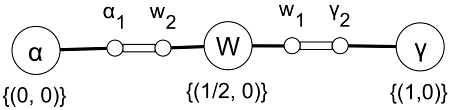


Graph definition

A graph, G, is a pregraph $G = (\mathcal{N}, \mathcal{P}, \mathcal{PN}, \mathcal{PP}, \mathcal{A}, \lambda)$ such that :

(i) $((p_1, p_2) \in \mathcal{PP}$ and $(p_1, p_3) \in \mathcal{PP}) \implies p_2 = p_3$ and $\forall p \in \mathcal{P}, (p, p) \notin \mathcal{PP}$.

(ii) $((p, n_1) \in \mathcal{PN} \text{ and } (p, n_2) \in \mathcal{PN}) \implies n_1 = n_2$.



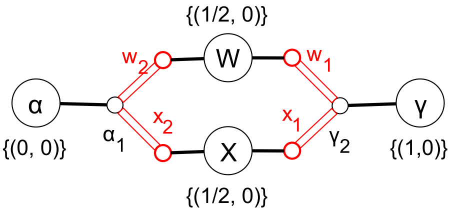
Equivalence relations over nodes and ports

Definition ($\equiv^P$, $\equiv^N$)

Let $H = (\mathcal{N}_H, \mathcal{P}_H, \mathcal{PN}_H, \mathcal{PP}_H, \mathcal{A}_H, \lambda_H)$ be a pregraph.

- $\equiv^P$ is defined as $(\mathcal{PP}_H \bullet \mathcal{PP}_H)^*$
- $\equiv^N$ is defined as $(\mathcal{PN}_H^{-} \bullet \equiv^P \bullet \mathcal{PN})^*$

where $\bullet$ denotes relation composition, $^-$ the converse of a relation and * the reflexive-transitive closure of a relation.



H

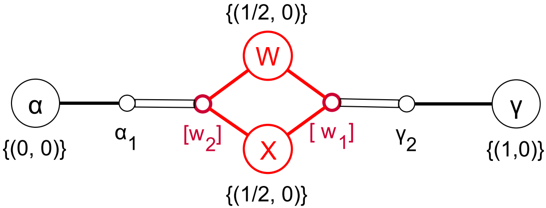
Equivalence relations over nodes and ports

Definition ($\equiv^P$, $\equiv^N$)

Let $H = (\mathcal{N}_H, \mathcal{P}_H, \mathcal{PN}_H, \mathcal{PP}_H, \mathcal{A}_H, \lambda_H)$ be a pregraph.

- $\equiv^P$ is defined as $(\mathcal{PP}_H \bullet \mathcal{PP}_H)^*$
- $\equiv^N$ is defined as $(\mathcal{PN}_H^- \bullet \equiv^P \bullet \mathcal{PN})^*$

where $\bullet$ denotes relation composition, $^-$ the converse of a relation and * the reflexive-transitive closure of a relation.



$[w_2] = \{w_2, x_2\}$ $[w_1] = \{w_1, x_1\}$

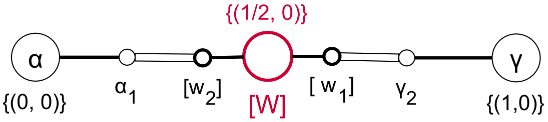
Equivalence relations over nodes and ports

Definition ($\equiv^P$, $\equiv^N$)

Let $H = (\mathcal{N}_H, \mathcal{P}_H, \mathcal{PN}_H, \mathcal{PP}_H, \mathcal{A}_H, \lambda_H)$ be a pregraph.

- $\equiv^P$ is defined as $(\mathcal{PP}_H \bullet \mathcal{PP}_H)^*$
- $\equiv^N$ is defined as $(\mathcal{PN}_H^{-} \bullet \equiv^P \bullet \mathcal{PN})^*$

where $\bullet$ denotes relation composition, $^-$ the converse of a relation and * the reflexive-transitive closure of a relation.



$[w_2] = \{w_2, x_2\}$ $\quad [W] = \{W, X\}$ $\quad [w_1] = \{w_1, x_1\}$

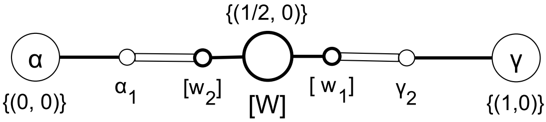
Equivalence relations over nodes and ports

Definition ($\equiv^P$, $\equiv^N$)

Let $H = (\mathcal{N}_H, \mathcal{P}_H, \mathcal{PN}_H, \mathcal{PP}_H, \mathcal{A}_H, \lambda_H)$ be a pregraph.

- $\equiv^P$ is defined as $(\mathcal{PP}_H \bullet \mathcal{PP}_H)^*$
- $\equiv^N$ is defined as $(\mathcal{PN}_H^- \bullet \equiv^P \bullet \mathcal{PN})^*$

where $\bullet$ denotes relation composition, $^-$ the converse of a relation and * the reflexive-transitive closure of a relation.



Quotient pregraph $\bar{H}$

Quotient pregraph and Graph isomorphism

Quotient pregraph : $\overline{H} = (\mathcal{N}_{\overline{H}}, \mathcal{P}_{\overline{H}}, \mathcal{PN}_{\overline{H}}, \mathcal{PP}_{\overline{H}}, \mathcal{A}_{\overline{H}}, \lambda_{\overline{H}})$

- $\mathcal{N}_{\overline{H}} = \{[n] \mid n \in \mathcal{N}_H\}$,
- $\mathcal{P}_{\overline{H}} = \{[p] \mid p \in \mathcal{P}_H\}$,
- $\mathcal{PN}_{\overline{H}} = \{([p],[n]) \mid (p,n) \in \mathcal{PN}_H\}$,
- $\mathcal{PP}_{\overline{H}} = \{([p],[q]) \mid (p,q) \in \mathcal{PP}_H\}$,
- $\mathcal{A}_{\overline{H}} = \mathcal{A}_H$ and $\lambda_{\overline{H}}([x]) = \cup_{x' \in [x]} \lambda_H(x')$ where $[x] \in \mathcal{N}_{\overline{H}} \uplus \mathcal{P}_{\overline{H}}$

Theorem : Let H and H' be two isomorphic pregraphs. Then $\bar{H}$ and $\bar{H}'$ are isomorphic.

Table of Contents

Environment Sensitive Rewrite Rule

$$l \longrightarrow r$$

$l = (\mathcal{N}_l, \mathcal{P}_l, \mathcal{PN}_l, \mathcal{PP}_l, \mathcal{A}, \lambda_l)$

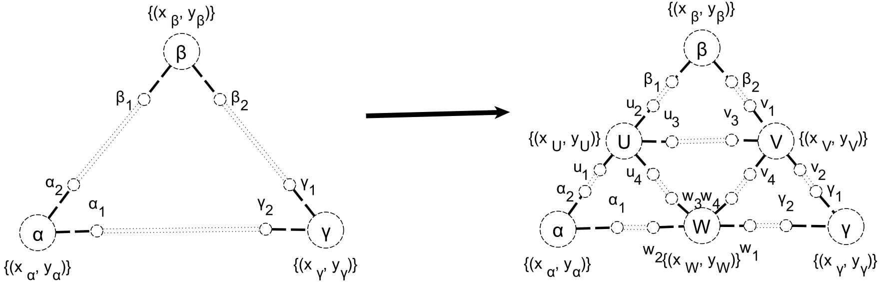
env part l^{env}

cut part l^{cut}

$r = (\mathcal{N}_r, \mathcal{P}_r, \mathcal{PN}_r, \mathcal{PP}_r, \mathcal{A}, \lambda_r)$

env part : r^{env}

new part r^{new}



Environment Sensitive Rewrite Rule

$$l \longrightarrow r$$

$l = (\mathcal{N}_l, \mathcal{P}_l, \mathcal{PN}_l, \mathcal{PP}_l, \mathcal{A}, \lambda_l)$

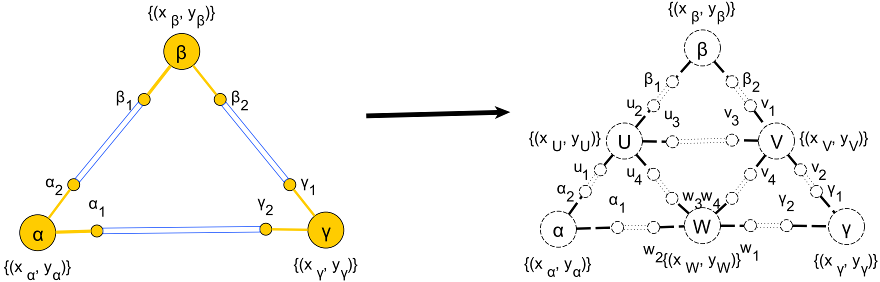
env part l^{env}

cut part l^{cut}

$r = (\mathcal{N}_r, \mathcal{P}_r, \mathcal{PN}_r, \mathcal{PP}_r, \mathcal{A}, \lambda_r)$

env part : r^{env}

new part r^{new}



Environment Sensitive Rewrite Rule

$$l \longrightarrow r$$

$l = (\mathcal{N}_l, \mathcal{P}_l, \mathcal{PN}_l, \mathcal{PP}_l, \mathcal{A}, \lambda_l)$

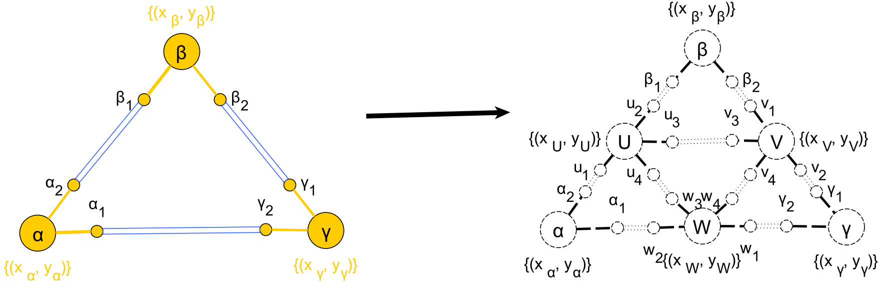
env part l^{env}

cut part l^{cut}

$r = (\mathcal{N}_r, \mathcal{P}_r, \mathcal{PN}_r, \mathcal{PP}_r, \mathcal{A}, \lambda_r)$

env part : r^{env}

new part r^{new}



Environment Sensitive Rewrite Rule

$$l \longrightarrow r$$

$l = (\mathcal{N}_l, \mathcal{P}_l, \mathcal{PN}_l, \mathcal{PP}_l, \mathcal{A}, \lambda_l)$

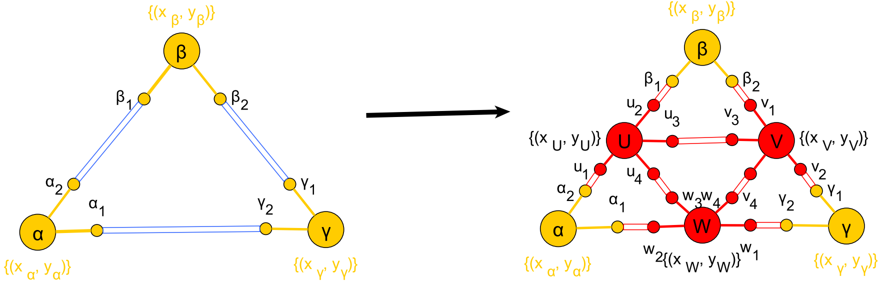
env part l^{env}

cut part l^{cut}

$r = (\mathcal{N}_r, \mathcal{P}_r, \mathcal{PN}_r, \mathcal{PP}_r, \mathcal{A}, \lambda_r)$

env part : r^{env}

new part r^{new}

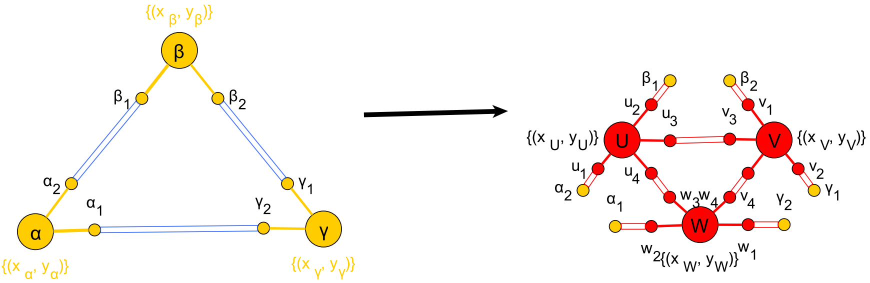


Environment Sensitive Rewrite Rule

$$l \longrightarrow r$$

$l = (\mathcal{N}_l, \mathcal{P}_l, \mathcal{PN}_l, \mathcal{PP}_l, \mathcal{A}, \lambda_l)$
env part l^{env}
cut part l^{cut}

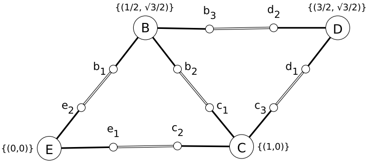
$r = (\mathcal{N}_r, \mathcal{P}_r, \mathcal{PN}_r, \mathcal{PP}_r, \mathcal{A}, \lambda_r)$
env part : r^{env}
new part r^{new}



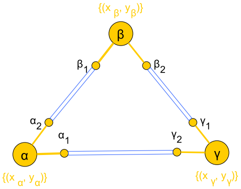
Definition (Matches)

Let l and g be two graphs. A *match* $m^a : l \to g$ is defined as an injective graph homomorphism. $a : \mathcal{A}_l \to \mathcal{A}_g$ being an injective homomorphism over attributes.

g



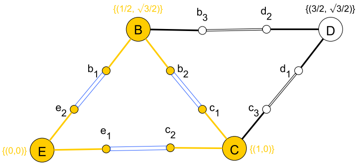
l



Definition (Matches)

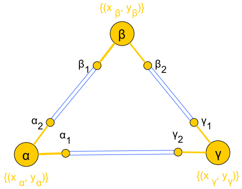
Let l and g be two graphs. A *match* $m^a : l \to g$ is defined as an injective graph homomorphism. $a : \mathcal{A}_l \to \mathcal{A}_g$ being an injective homomorphism over attributes.

g



$$m_1^{a_1} : \alpha \to E; \beta \to B; \gamma \to C; \alpha_1 \to e_1; \alpha_2 \to e_2; \beta_1 \to b_1; \beta_2 \to b_2; \gamma_1 \to c_1; \gamma_2 \to c_2.$$

l

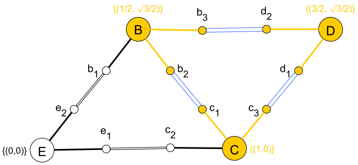


$$a_1 : x_\alpha \to 0; y_\alpha \to 0; x_\beta \to \frac{1}{2}; y_\beta \to \frac{\sqrt{3}}{2}; x_\gamma \to 1; y_\gamma \to 0.$$

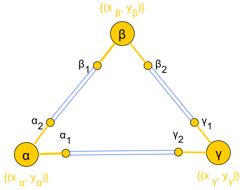
Definition (Matches)

Let l and g be two graphs. A *match* $m^a : l \to g$ is defined as an injective graph homomorphism. $a : \mathcal{A}_l \to \mathcal{A}_g$ being an injective homomorphism over attributes.

g



l



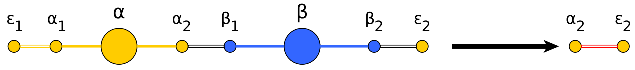
$$m_2^{a_2} : \alpha \to B; \beta \to D; \gamma \to C; \\ \alpha_1 \to b_2;\ \alpha_2 \to b_3;\ \beta_1 \to d_2; \\ \beta_2 \to d_1;\ \gamma_1 \to c_3;\ \gamma_2 \to c_1.$$

$$a_2 : x_\alpha \to \frac{1}{2}; y_\alpha \to \frac{\sqrt{3}}{2}; \\ x_\beta \to \frac{3}{2}; y_\beta \to \frac{\sqrt{3}}{2}; \\ x_\gamma \to 1; y_\gamma \to 0.$$

Compatible rules

$l_1 \to r_1$ and $l_2 \to r_2$ are said to be compatible iff for all graphs G and matches $m_1^{a_1} : l_1 \to G$ and $m_2^{a_2} : l_2 \to G$,

- no element of $m_1^{a_1}(r_1^{env})$ is in $m_2^{a_2}(l_2^{cut})$ and
- no element of $m_2^{a_2}(r_2^{env})$ is in $m_1^{a_1}(l_1^{cut})$.



not compatible

Theorem : The problem of the verification of compatibility of two rules is decidable.

Considered rewrite systems are consisting of pairwise compatible rules

Parallel rewrite step $G \Rightarrow_{I,M} G'$

- $\mathcal{R} = \{L_i \to R_i \mid i = 1 \ldots n\}$
- I set of variants $I = \{l_i \to r_i \mid i = 1 \ldots k\}$
- M a set of matches $M = \{m_i^{a_i} : l_i \to G \mid i = 1 \ldots k\}$

First step: A pregraph $H = (\mathcal{N}_H, \mathcal{P}_H, \mathcal{PN}_H, \mathcal{PP}_H, \mathcal{A}_H, \lambda_H)$ is computed using the different matches and rules

- $H = (G - \cup_{i=1}^{k} m_i^{a_i}(l_i^{cut})) \uplus \cup_{i=1}^{k} r_i^{new}$

Second step: $G' = \overline{H}$

Parallel rewrite step $G \Rightarrow_{I,M} G'$

- $\mathcal{R} = \{L_i \rightarrow R_i \mid i = 1 \dots n\}$
- I set of variants $I = \{l_i \rightarrow r_i \mid i = 1 \dots k\}$
- M a set of matches $M = \{m_i^{a_i} : l_i \rightarrow G \mid i = 1 \dots k\}$

First step: A pregraph $H = (\mathcal{N}_H, \mathcal{P}_H, \mathcal{PN}_H, \mathcal{PP}_H, \mathcal{A}_H, \lambda_H)$ is computed using the different matches and rules

- $\mathcal{N}_H = (\mathcal{N}_G - \cup_{i=1}^{k} \mathcal{N}^{cut}_{m_i^{a_i}(l_i)}) \uplus \cup_{i=1}^{k} \mathcal{N}^{new}_{r_i}$
- $\mathcal{P}_H = (\mathcal{P}_G - \cup_{i=1}^{k} \mathcal{P}^{cut}_{m_i^{a_i}(l_i)}) \uplus \cup_{i=1}^{k} \mathcal{P}^{new}_{r_i}$
- $\mathcal{PN}_H = ((\mathcal{PN}_G \cap (\mathcal{P}_H \times \mathcal{N}_H)) - \cup_{i=1}^{k} \mathcal{PN}^{cut}_{m_i^{a_i}(l_i)}) \uplus \cup_{i=1}^{k} \mathcal{PN}^{new}_{m_i^{a_i}(r_i)}$
- $\mathcal{PP}_H = ((\mathcal{PP}_G \cap (\mathcal{P}_H \times \mathcal{P}_H)) - \cup_{i=1}^{k} \mathcal{PP}^{cut}_{m_i^{a_i}(l_i)}) \uplus \cup_{i=1}^{k} \mathcal{PP}^{new}_{m_i^{a_i}(r_i)}$
- $\mathcal{A}_H = \mathcal{A}_G$ and $\lambda_H = (\lambda_G - \cup_{i=1}^{k} \lambda^{cut}_{m_i^{a_i}(l_i)}) \cup \cup_{i=1}^{n} \lambda^{new}_{m_i^{a_i}(r_i)}$

Second step: $G' = \overline{H}$

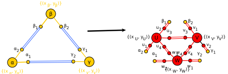
Parallel rewrite step $G \Rightarrow_{I,M} G'$

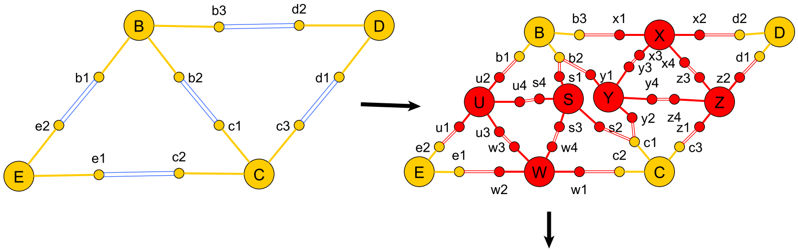
- $\mathcal{R} = \{L_i \to R_i \mid i = 1 \dots n\}$
- I set of variants $I = \{l_i \to r_i \mid i = 1 \dots k\}$
- M a set of matches $M = \{m_i^{a_i} : l_i \to G \mid i = 1 \dots k\}$

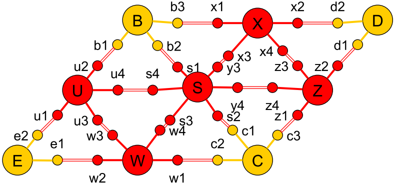
First step: A pregraph $H = (\mathcal{N}_H, \mathcal{P}_H, \mathcal{PN}_H, \mathcal{PP}_H, \mathcal{A}_H, \lambda_H)$ is computed using the different matches and rules

- $\mathcal{N}_H = (\mathcal{N}_G - \cup_{i=1}^k \mathcal{N}^{cut}_{m_i^{a_i}(l_i)}) \uplus \cup_{i=1}^k \mathcal{N}^{new}_{r_i}$
- $\mathcal{P}_H = (\mathcal{P}_G - \cup_{i=1}^k \mathcal{P}^{cut}_{m_i^{a_i}(l_i)}) \uplus \cup_{i=1}^k \mathcal{P}^{new}_{r_i}$
- $\mathcal{PN}_H = ((\mathcal{PN}_G \cap (\mathcal{P}_H \times \mathcal{N}_H)) - \cup_{i=1}^k \mathcal{PN}^{cut}_{m_i^{a_i}(l_i)}) \uplus \cup_{i=1}^k \mathcal{PN}^{new}_{m_i^{a_i}(r_i)}$
- $\mathcal{PP}_H = ((\mathcal{PP}_G \cap (\mathcal{P}_H \times \mathcal{P}_H)) - \cup_{i=1}^k \mathcal{PP}^{cut}_{m_i^{a_i}(l_i)}) \uplus \cup_{i=1}^k \mathcal{PP}^{new}_{m_i^{a_i}(r_i)}$
- $\mathcal{A}_H = \mathcal{A}_G$ and $\lambda_H = (\lambda_G - \cup_{i=1}^k \lambda^{cut}_{m_i^{a_i}(l_i)}) \cup \cup_{i=1}^n \lambda^{new}_{m_i^{a_i}(r_i)}$

Second step: $G' = \overline{H}$ **Theorem : $\overline{H}$ is a graph**

$\mathcal{R} = \{$  $\}, I = \{l_1 \to r_1, l_2 \to r_2\}, M = \{m_1^{a_1}, m_2^{a_2}\}.$





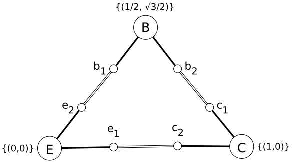
$[S] = \{S, Y\},$
$[s_1] = \{s_1, y_1\}, [s_2] = \{s_2, y_2\}.$

Table of Contents

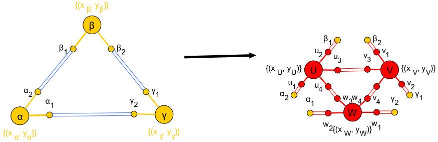
Full Parallel Rewrite Relation

Full parallel matches : M the maximal subset of
$\mathcal{M}_{\mathcal{R}}(g) = \{m_i^{a_i} : l_i \to g \mid m_i^{a_i}$ is a match and $l_i \to r_i$ is a variant$\}$.
s.t. $\forall m_1^{a_1}, m_2^{a_2} \in M, m_1^{a_1} \not\approx m_2^{a_2}$.
Full parallel rewriting : $g \rightrightarrows_M g'$

$$g$$



$$l \to r$$

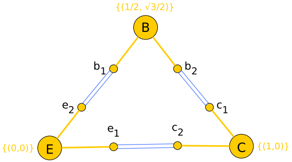


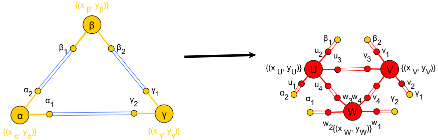
Full Parallel Rewrite Relation

Full parallel matches : M the maximal subset of
$\mathcal{M}_{\mathcal{R}}(g) = \{m_i^{a_i} : l_i \to g \mid m_i^{a_i}$ is a match and $l_i \to r_i$ is a variant$\}$.
s.t. $\forall m_1^{a_1}, m_2^{a_2} \in M, m_1^{a_1} \not\approx m_2^{a_2}$.
Full parallel rewriting : $g \rightrightarrows_M g'$

g

$l \to r$



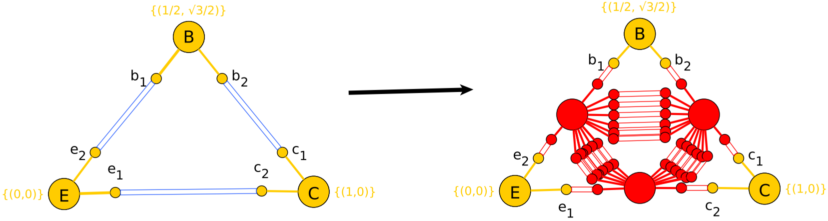


6 matches!

Full Parallel Rewrite Relation

Full parallel matches : M the maximal subset of
$\mathcal{M}_{\mathcal{R}}(g) = \{m_i^{a_i} : l_i \to g \mid m_i^{a_i}$ is a match and $l_i \to r_i$ is a variant$\}$.
s.t. $\forall m_1^{a_1}, m_2^{a_2} \in M, m_1^{a_1} \not\approx m_2^{a_2}$.
Full parallel rewriting : $g \rightrightarrows_M g'$

$$g \rightrightarrows_M g'$$

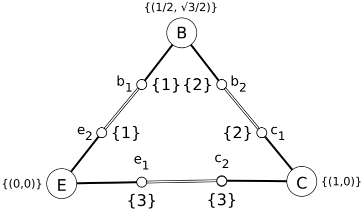


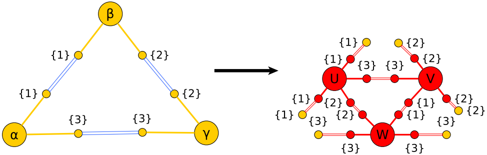
Full Parallel Rewrite Relation and distinguishing attributes

Full parallel rewriting : $g \rightrightarrows_M g'$

$$g$$

$$l \to r$$

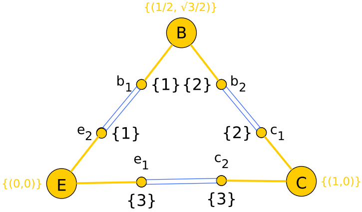




Full Parallel Rewrite Relation and distinguishing attributes

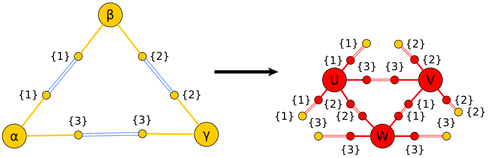
Full parallel rewriting : $g \rightrightarrows_M g'$

$$g$$



1 match!

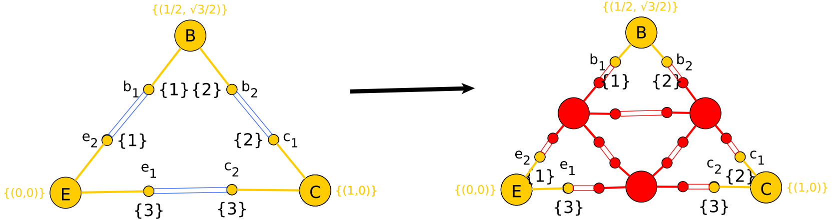
$$l \rightarrow r$$



Full Parallel Rewrite Relation and distinguishing attributes

Full parallel rewriting : $g \rightrightarrows_M g'$

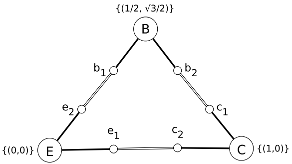
$$g \rightrightarrows_M g'$$



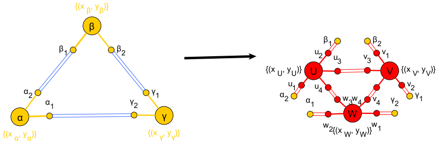
Parallel Rewrite Relation up to Automorphisms

Symmetry condition :
$\forall h^a \in H(l),\ \exists\ h'^a \in H(r),$ such that $\forall x \in r^{env}, h^a(x) = h'^a(x)$
Matches up to automorphism
there exists an auto. $h^a : l \to l$ such that $m_1^{b_1} = m_2^{b_2} \circ h^a$
Rewriting up to automorphisms
$M^{auto}_{\mathcal{R},G} = \{m_i^{a_i} : l_i \to G\ |m_i^{a_i}$ is a match up to auto$\}$.
Rewriting up to automorphisms : $g \rightrightarrows_{auto} g'$

g



$l \to r$



Parallel Rewrite Relation up to Automorphisms

Symmetry condition :

$\forall h^a \in H(l),\ \exists\, h'^a \in H(r),$ such that $\forall x \in r^{env}, h^a(x) = h'^a(x)$

Matches up to automorphism

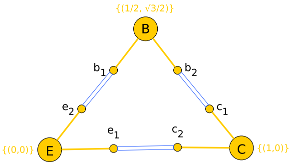
there exists an auto. $h^a : l \to l$ such that $m_1^{b_1} = m_2^{b_2} \circ h^a$

Rewriting up to automorphisms

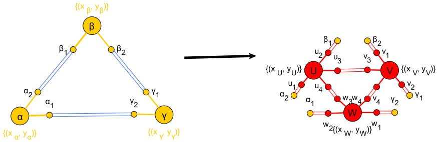
$M^{auto}_{\mathcal{R},G} = \{m_i^{a_i} : l_i \to G \mid m_i^{a_i}$ is a match up to auto$\}$.

Rewriting up to automorphisms : $g \rightrightarrows_{auto} g'$

g



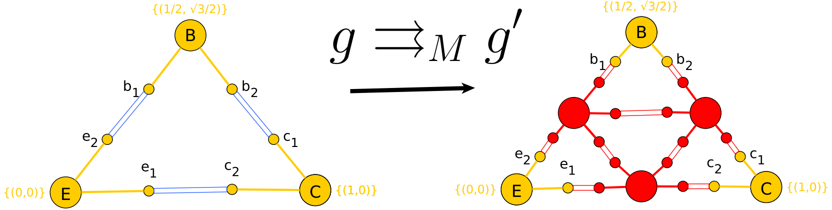
$l \to r$



1 match up to automorphism! (instead of 6)

Parallel Rewrite Relation up to Automorphisms

Symmetry condition :
$\forall h^a \in H(l), \exists\, h'^a \in H(r),$ such that $\forall x \in r^{env}, h^a(x) = h'^a(x)$
Matches up to automorphism
there exists an auto. $h^a : l \to l$ such that $m_1^{b_1} = m_2^{b_2} \circ h^a$
Rewriting up to automorphisms
$M^{auto}_{\mathcal{R},G} = \{m_i^{a_i} : l_i \to G \mid m_i^{a_i}$ is a match up to auto$\}$.
Rewriting up to automorphisms : $g \rightrightarrows_{auto} g'$

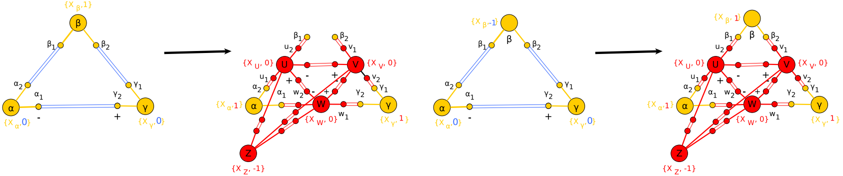


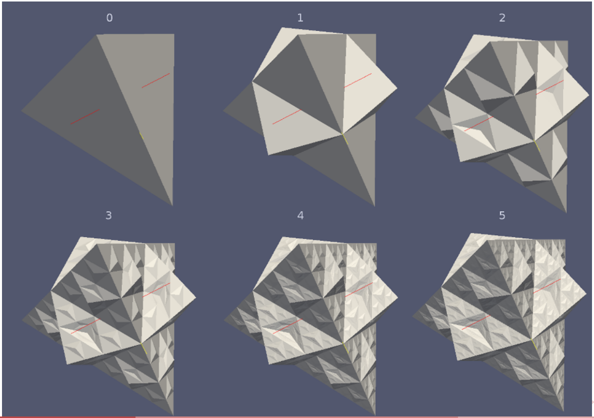
Parallel rewriting and determinism

Theorem : The rewrite relation $\rightrightarrows$ is deterministic :
For all graphs G, $(G \rightrightarrows G_1 \; and \; G \rightrightarrows G_2)$ implies that G_1 and G_2 are isomorphic.

Theorem : The rewrite relation $\rightrightarrows_{auto}$ is deterministic :
For all graphs G, for all $\mathcal{R}$ satisfying the symmetry condition $(G \rightrightarrows_{auto} G'_1$ and $G \rightrightarrows_{auto} G'_2)$ implies that G'_1 and G'_2 are isomorphic.

Example : Koch snow flake





Example : Koch snow flake

rule 1

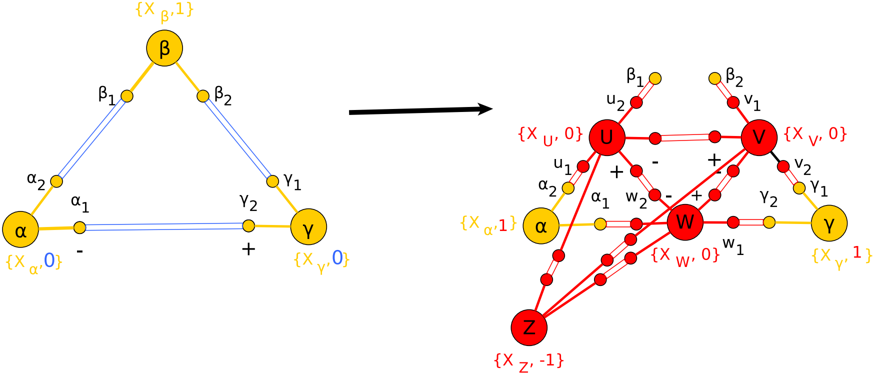


Table of Contents

Conclusion

- Algorithmic approach of parallel graph rewriting
- Rules can overlap
- Two deterministic parallel rewrite strategies

Future work

- Software
- Theoretical extensions
  - Stochastic parallel rewriting
  - Conditional parallel rewriting
  - Bloc parallel rewriting