HAL open science

Parallel rewriting of attributed graphs

Thierry Boy de La Tour, Rachid Echahed

▶ To cite this version:

Thierry Boy de La Tour, Rachid Echahed. Parallel rewriting of attributed graphs. Theoretical Computer Science, 2020, 848, pp.106 - 132. 10.1016/j.tcs.2020.09.025 . hal-03430149

HAL Id: hal-03430149

https://hal.univ-grenoble-alpes.fr/hal-03430149

Submitted on 19 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Parallel Rewriting of Attributed Graphs

Thierry Boy de la Tour Rachid Echahed

Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG
38000 Grenoble, France
thierry.boy-de-la-tour@imag.fr rachid.echahed@imag.fr

Abstract

Some computations can be elegantly presented as the parallel or simultaneous application of rules. This is the case of cellular automata and of simultaneous assignments in Python. In both cases the expected result cannot be obtained by a sequential application of rules. A general framework of attributed graph transformations is proposed where such computations can be expressed and analyzed. Determinism is achieved by an exhaustive parallel application of rules, as in cellular automata that are shown to have a straightforward representation in this framework. A more concise parallel transformation is also proposed, where some applications of rules can be ignored thanks to their symmetries, while preserving determinism. Parallel transformations are then used to characterize the property of parallel independence.

1 Introduction

Graph structures are widely used in many areas of computer science and well beyond (e.g., Biology, Chemistry, Physics). Their visual appearance as well as their expressiveness give them an important place in the modeling of complex systems. Computing with graphs as first-class citizens requires the use of advanced graph-based computation models. Several approaches to graph transformations have been proposed in the literature, in particular those based on a variety of graph transformation rules. There are two main streams of research, on one hand the *algebraic* approaches where transformations are defined using notions borrowed from category theory and that apply to any suitable category, including the category of graphs (see, e.g., [1, 2]), on the other hand the *algorithmic* approaches where graph transformations are defined by means of the involved algorithms in the rewriting process (e.g., [3, 4]). In the present paper, an algorithmic framework is introduced that is geared toward a notion of true parallelism of attributed graph transformations. By true parallelism we mean simultaneous transformations acting over possibly overlapping parts of a subject graph as well as the simultaneous changes of the attributes or values, attached to graphs items, i.e., nodes or edges.

The study of rule-based graph transformations turns out to be more difficult than other structures such as strings [5] or terms [6]. A rule is usually expressed by means of a left-hand and a right-hand sides, L and R respectively, meaning that any occurence of L should be replaced by R. When L and R are graphs, applying the rule to a graph G amounts to (non-deterministically) finding a homomorphic image $\mu(L)$ in G, thus expressing G as a context $C[\mu(L)]$, and *replacing* this image by $\mu(R)$. The elementary way of performing this replacement consists in first removing $\mu(L)$ from G and then adding a subgraph $\mu(R)$ to obtain the transformed graph $H = C[\mu(R)]$. But this would sever all links between $\mu(L)$ and $C[]$, thus leaving an isolated $\mu(R)$ in H. In order to preserve at least some of these links, it is customary to add an *interface graph* K between L and R, specifying which part of L is not to be removed (imagine K as a subgraph of both L and R). Once the semantics of rules is defined, there are different ways to compute with graph rewrite systems. As confluence property is seldom true even for orthogonal graph rewrite systems [7, 4, 8], the use of strategies is often advocated to ensure unique results (see, e.g., GP system [9] or PORGY [10]).

Another way of specifying a deterministic graph transformation by means of rules is to apply all possible transformations simultaneously, thus averting a non-deterministic choice. This is the case for example in cellular automata, where the next generation is determined by applying a local rule simultaneously on all cells. It is well known that determinism cannot be achieved by applying the rules sequentially, since the resulting states of cells may depend on the undetermined order in which the cells would be considered.

In graph transformation theory, parallel transformations have so far been considered mostly under the interleaving semantics, that led to the notion of *parallel independence* [11] (see Section 9). This condition restricts the amount of overlap between images of left-hand sides in order to guarantee that a parallel rewrite step yields the same result as any sequence of rewrite steps using the same rules. It is therefore unable to account for the kind of parallelism that is used in cellular automata. For this, we need to allow the simultaneous applications of rules that are not parallel independent, i.e., that overlap in a *parallel dependent* way.

Yet not all overlaps can be allowed since the corresponding rules may then conflict with each other. For instance, a rule may remove a part of the left-hand side of another rule, or they may give different values to a same variable (e.g., different states to one cell). In order to preserve the semantics of all the rules involved in a parallel transformation, it is necessary to avoid such conflicts.

But conflict-free overlaps may not yield a powerful notion of parallelism by themselves. The notion of conflict between rules clearly depends on the semantics that is ascribed to the rules, i.e., to the graph rewrite relations that they are assumed to define. A narrow definition may yield unnecessary conflicts. A wider definition may ease conflict-free parallelism, but may also make determinism elusive.

We are therefore interested in defining a notion of rule-based graph transformation, with suitable rules, whose (conflict-free but possibly dependent) parallelization yields an expressive notion of deterministic parallel graph transformation. The expressiveness shall encompass cellular automata, but also possibly other examples of dependent parallelism.

The paper is organized as follows. We first introduce some basic definitions and notations in Section 2, including a notion of attributed graphs where attributes are sets of values. This eliminates a source of conflict between rules in the sense that such attributes always accomodate enough space for new data.

The algorithmic framework is introduced in Section 3, where unions and intersections of graphs are defined, and then used to define a natural way of removing objects in an attributed graph.

The shapes of the rules as well as the notion of parallel graph transformations investigated in this paper are the subject of Section 4. A simple example is introduced to motivate the necessity to generalize the standard notion of rules to triples of graphs (L, K, R) where K is not required to be a subgraph of R. This simple extension supports a general definition of parallel transformation that is not deterministic.

In Section 5, a specific graph transformation called *parallel rewriting* is defined under a condition called the *effective deletion property*. It is proved to be both a most general instance of the general transformation and deterministic modulo isomorphism when applied exhaustively. This transformation is further validated in Section 6 where it is shown to generalize cellular automata. This is illustrated on John Conway's Game of Life.

One then observes that the presence of symmetries in a rule induces matchings that yield similar transformations, if taken individually. This suggests that determinism can be preserved by selecting only one matching among the similar ones. A notion of automorphism group of a rule is thus introduced in Section 7, based on the symmetries of the graphs L, K and R. The framework established in Sections 2, 3 and 4 allows a simple definition of this group as a *permutation group*, and it is shown to be finite.

Based on these groups, a new parallel rewriting relation is proposed in Section 8 where the rules may not be applied in an exhaustive way. By using only one matching in every equivalence class modulo the automorphisms of the corresponding rule, it is shown that determinism modulo isomorphism is preserved.

Section 9 is devoted to *parallel independence*, i.e., to the condition on the overlaps between pairs of matchings of rules that not only guarantees, but characterizes the fact that parallel rewriting yields the same result as (a restricted form of) sequential rewriting, a condition known as *sequential independence*.

Concluding remarks and related work are given in Section 10, especially other notions of parallel graph transformations are compared with the one proposed in this paper.

2 Preliminaries

In this section, we recall or define some basic notions that are used throughout the paper, such as Σ-algebras and attributed graphs.

2.1 Signatures and Σ-algebras

A *(many-sorted) signature* is a triple $\Sigma = (S, \Omega, \tau)$ where S and Ω are sets and τ is a function from Ω to $S^\star \times S$. The elements of S are called *sorts*. For all $f \in \Omega$ such that $\tau(f) = (s_1 \cdots s_n, s)$, if $n = 0$ then f is a *constant of sort* s, and if $n > 0$ then f is a *function symbol of type* $s_1 \times \cdots \times s_n \to s$. We assume throughout this paper a fixed signature Σ; only in some examples and in Section 6 will Σ be narrowed down to some particular signatures.

A *Σ-algebra* is a pair $\mathcal{A} = ((\mathcal{A}_s)_{s\in S}, (f_\mathcal{A})_{f\in\Omega})$ where $(\mathcal{A}_s)_{s\in S}$ is a family of pairwise disjoint[1] sets, $c_\mathcal{A} \in \mathcal{A}_s$ for all constants c of sort s, and $f_\mathcal{A}$ is a function from $\mathcal{A}_{s_1} \times \cdots \times \mathcal{A}_{s_n}$ to $\mathcal{A}_s$ for all function symbols f of type $s_1 \times \cdots \times s_n \to s$. $\mathcal{A}_s$ (resp. $f_\mathcal{A}$) is the *interpretation of* $s \in S$ (resp. $f \in \Omega$) *in* $\mathcal{A}$. The *carrier set* of $\mathcal{A}$ is $\lfloor\mathcal{A}\rfloor \stackrel{\text{def}}{=} \bigcup_{s\in S} \mathcal{A}_s$. Note that every element of $\lfloor\mathcal{A}\rfloor$ belongs to a unique $\mathcal{A}_s$, hence is implicitly typed[2] (by s).

A *Σ-homomorphism* from Σ-algebra $\mathcal{A}$ to Σ-algebra $\mathcal{B}$ is a function α from $\lfloor\mathcal{A}\rfloor$ to $\lfloor\mathcal{B}\rfloor$ such that $\alpha(\mathcal{A}_s) \subseteq \mathcal{B}_s$ for all sorts s, $\alpha(c_\mathcal{A}) = c_\mathcal{B}$ for all constants c, and $\alpha \circ f_\mathcal{A}(a_1, \ldots, a_n) = f_\mathcal{B}(\alpha(a_1), \ldots, \alpha(a_n))$ for all function symbols f of type $s_1 \times \cdots \times s_n \to s$ and all $(a_1, \ldots, a_n) \in \mathcal{A}_{s_1} \times \cdots \times \mathcal{A}_{s_n}$. If α is bijective then it is a *Σ-isomorphism* (and then α^{-1} is a Σ-homomorphism from $\mathcal{B}$ to $\mathcal{A}$), and if furthermore $\mathcal{A} = \mathcal{B}$ then α is a *Σ-automorphism* of $\mathcal{A}$. We write $1_\mathcal{A}$ for the identity Σ-automorphism of $\mathcal{A}$.

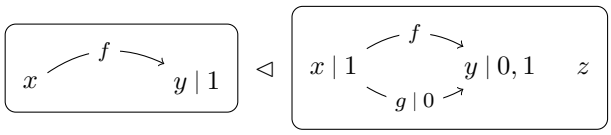
We assume a set $\mathscr{V}$ disjoint from Ω, whose elements are called *variables*, and a function $\tau' : \mathscr{V} \to S$. For any finite $X \subseteq \mathscr{V}$, the Σ-algebra $\mathscr{T}(\Sigma, X)$ of Σ-terms on X is defined as usual. $\mathscr{T}(\Sigma, X)$ is *free with generating set X in* the class of Σ-algebras, i.e., for all Σ-algebras $\mathcal{A}$ and all functions $\theta : X \to \lfloor\mathcal{A}\rfloor$ such that $\theta(v) \in \mathcal{A}_{\tau'(v)}$ for all $v \in X$, there exists a unique Σ-homomorphism from $\mathscr{T}(\Sigma, X)$ to $\mathcal{A}$ that extends θ. A Σ-algebra $\mathcal{A}$ is *reachable* if the unique Σ-homomorphism from $\mathscr{T}(\Sigma, \varnothing)$ to $\mathcal{A}$ is surjective (every element of $\lfloor\mathcal{A}\rfloor$ has a denotation).

2.2 Attributed graphs

An *attributed graph* (or *graph* for short) G is a tuple $(\dot{G}, \vec{G}, \acute{G}, \grave{G}, \mathscr{A}_G, \mathring{G})$ where $\dot{G}, \vec{G}$ are sets, $\acute{G}, \grave{G}$ are the *source* and *target* functions from $\vec{G}$ to $\dot{G}$, $\mathscr{A}_G$ is a Σ-algebra and $\mathring{G}$ is an *attribution of* G, i.e., a function from $\dot{G} \cup \vec{G}$ to $\mathscr{P}(\lfloor\mathscr{A}_G\rfloor)$. We assume that $\dot{G}$, $\vec{G}$ and $\lfloor\mathscr{A}_G\rfloor$ are pairwise disjoint; their elements are respectively called *vertices*, *arrows* and *attributes*. The *carrier* of G is the set $\lfloor G\rfloor \stackrel{\text{def}}{=} \dot{G} \cup \vec{G} \cup \lfloor\mathscr{A}_G\rfloor$. The *underlying graph* of G is $(\dot{G}, \vec{G}, \acute{G}, \grave{G})$ (this is a standard graph). G is *unlabelled* if $\mathring{G}(x) = \varnothing$ for all $x \in \dot{G} \cup \vec{G}$, it is *finite* if the sets $\dot{G}$, $\vec{G}$ and $\mathring{G}(x)$ are finite (note that $\lfloor G\rfloor$ may still be infinite).

A graph F is a *subgraph* of G, written $F \lhd G$, if $\dot{F} \subseteq \dot{G}$, $\vec{F} \subseteq \vec{G}$, $\acute{F} = \acute{G}|_{\vec{F}}$, $\grave{F} = \grave{G}|_{\vec{F}}$, $\mathscr{A}_F = \mathscr{A}_G$ and $\mathring{F}(x) \subseteq \mathring{G}(x)$ for all $x \in \dot{F} \cup \vec{F}$. The relation $\lhd$ is a partial order on graphs.

In the subgraph relation the names of vertices and arrows are important, for this reason graphs will usually be depicted with the names of the graph items, and their attributes will be listed after each name, separated from it by | (which is omitted if the attribute is $\varnothing$). Since graphs may not be connected, they will be surrounded by a rectangle with rounded corners, as in:



The arrows specify the source and target functions, and the associated Σ-algebra may be specified separately. Here, any algebra containing 0 and 1 in its carrier set would do.

Given two attributions l, l' of G, we write $l \cap l'$ for the attribution that maps any $x \in \dot{G} \cup \vec{G}$ to $l(x) \cap l'(x)$. The attributions $l \cup l'$ and $l \backslash l'$ are defined similarly.

Given $F \lhd G$ and an attribution l of F, we consider l as an attribution of G by extending it implicitly with empty sets, i.e., for all $x \in (\dot{G} \cup \vec{G}) \backslash (\dot{F} \cup \vec{F})$ we let $l(x) = \varnothing$. Conversely, given an attribution l' of G, the restriction of l' to $\dot{F} \cup \vec{F}$ is an attribution of F that will be written l' (by a slight abuse of notation).

[1] This condition is by no means essential; its aim is to simplify notations and definitions.

[2] This is particularly true for terms and would not be the case if the overloading of function symbols had been adopted, as in [12].

2.3 Morphisms

Morphisms between attributed graphs are needed for two reasons. The most obvious one is that graph rewriting rules must be matched with the input graph and that a matching is a morphism, though a special one in the present framework. Another, more fundamental reason is to compute not just with graphs, but with *abstract* graphs, e.g., in Graph Theory the abstract graph K_n is any graph that is a clique with n vertices, independently of their names. This is similar to working with λ-terms modulo α-conversion but one can get away with it by using de Bruijn indices; nothing of the sort is available on graphs. There is no other way than to consider the input graph as a representation of an abstract graph, and to use tools[3] that do not depend on this particular representation. We therefore need the notion of isomorphism that links the different representations of an abstract graph.

A morphism of attributed graphs is a standard graph morphism extended with a Σ-homomorphism that preserves the contents of the attributes. More precisely, given two graphs G and H, a *morphism* α from G to H, written $\alpha : G \to H$, is a function from $\lfloor G \rfloor$ to $\lfloor H \rfloor$ whose restriction to $\lfloor \mathscr{A}_G \rfloor$, denoted $\mathring{\alpha}$, is a Σ-homomorphism from $\mathscr{A}_G$ to $\mathscr{A}_H$, such that $\alpha(\dot{G}) \subseteq \dot{H}$, $\alpha(\vec{G}) \subseteq \vec{H}$, $\alpha \circ \acute{G} = \acute{H} \circ \alpha$, $\alpha \circ \grave{G} = \grave{H} \circ \alpha$ (i.e., adjacencies are preserved) and $\mathring{\alpha} \circ \mathring{G}(x) \subseteq \mathring{H} \circ \alpha(x)$ for all $x \in \dot{G} \cup \vec{G}$. Note that $\mathring{\alpha} \circ \mathring{G}(x) = \{\mathring{\alpha}(a) \mid a \in \mathring{G}(x)\}$ and that $\mathring{\alpha}$ associates a value to *all* elements of $\lfloor \mathscr{A}_G \rfloor$, not just to those occurring in $\bigcup_{x \in \dot{G} \cup \vec{G}} \mathring{G}(x)$. The *underlying graph morphism* of α, denoted α by abuse of notation, is the restriction of α to $\dot{G} \cup \vec{G}$. A *matching* is a morphism whose underlying graph morphism is injective.

For instance, if $G \lhd H$ then $\lfloor G \rfloor \subseteq \lfloor H \rfloor$ and the canonical injection j from $\lfloor G \rfloor$ to $\lfloor H \rfloor$ is a matching from G to H. Note that $\mathring{j} = 1_{\mathscr{A}_G}$.

The *image* $\alpha(F)$ of a subgraph $F \lhd G$ by α is the smallest subgraph of H (w.r.t. $\lhd$) such that $\alpha|_{\lfloor F \rfloor}$ is a morphism from F to $\alpha(F)$. It is easy to see that this graph always exists and that

$$\alpha(F) = (\, \alpha(\dot{F}),\, \alpha(\vec{F}),\, \acute{H}|_{\alpha(\vec{F})},\, \grave{H}|_{\alpha(\vec{F})},\, \mathscr{A}_H,\, l\,) \;\text{ where } l(y) = \bigcup_{x \in \alpha^{-1}(y)} \mathring{\alpha} \circ \mathring{F}(x)$$

for all $y \in \alpha(\dot{F}) \cup \alpha(\vec{F})$. In particular when α is a matching then $l = \mathring{\alpha} \circ \mathring{F} \circ \alpha^{-1}$. We see that $\alpha(F) \lhd \alpha(G) \lhd H$. The *image* of the underlying graph of F by α is the underlying graph of $\alpha(F)$ (and is denoted similarly).

An *isomorphism* α from G to H, written $\alpha : G \simeq H$, is a bijective morphism $\alpha : G \to H$ such that $\alpha^{-1} : H \to G$ is a morphism, i.e., such that $\mathring{\alpha} \circ \mathring{G} = \mathring{H} \circ \alpha$; α is an *automorphism* if $G = H$. We write $G \simeq H$ and say that G and H are *isomorphic* if there exists an isomorphism $\alpha : G \simeq H$. We write 1_G for the identity automorphism of the graph G. Note that $\mathring{1}_G = 1_{\mathscr{A}_G}$.

A binary relation $\Longrightarrow$ on graphs is *deterministic up to isomorphism* if, for all graphs G, G', H and H', the relations $G \simeq G'$, $G \Longrightarrow H$ and $G' \Longrightarrow H'$ entail $H \simeq H'$.

3 Operations on Graphs

In order to define parallel rewrite relations on graphs, it is convenient to take the union of possibly many different graphs that have a common part. This operation can only be defined for graphs that are compatible on this common part, and that we call *joinable* below. We start with a simpler notion of joinable functions. Basic properties are given without proofs.

Definition 3.1 (joinable functions)**.** Two functions $f : D \to C$ and $g : D' \to C'$ are *joinable* if $f(x) = g(x)$ for all $x \in D \cap D'$, i.e., both functions map common domain elements to same images.

If f and g are joinable, then the *meet* of f and g is the function $f \curlywedge g$ from $D \cap D'$ to $C \cap C'$ that is the restriction of f (or g) to $D \cap D'$, and the *join* $f \curlyvee g$ is the unique function from $D \cup D'$ to $C \cup C'$ such that $f = (f \curlyvee g)|_D$ and $g = (f \curlyvee g)|_{D'}$.

Similarly, given a set I, an I-indexed family $(f_i : D_i \to C_i)_{i \in I}$ of functions is *joinable* if its elements are pairwise joinable, and then let $\bigcurlyvee_{i \in I} f_i$ be the only function from $\bigcup_{i \in I} D_i$ to $\bigcup_{i \in I} C_i$ such that $f_j = \big(\bigcurlyvee_{i \in I} f_i\big)|_{D_j}$ for all $j \in I$.

For all sets of functions S and T, if the elements of S and T can be composed then we write $S \circ T$ for $\{f \circ g \mid f \in S,\, g \in T\}$, and $f \circ T$ for $S \circ T$ when $S = \{f\}$. Similarly, if these elements are joinable we write $S \curlyvee T$ for $\{f \curlyvee g \mid f \in S,\, g \in T\}$.

In particular, functions with disjoint domains are joinable, and every function is joinable with itself: $f \curlyvee f = f \curlywedge f = f$. More generally, any two restrictions $f|_A$ and $f|_B$ of the same function f are joinable, $f|_A \curlywedge f|_B = f|_{A \cap B}$ and $f|_A \curlyvee f|_B = f|_{A \cup B}$.

It is obvious that these operations are commutative. On triples of pairwise joinable functions, they are also associative and distributive over each other.

If two joinable functions are injective then so is their meet; if they are surjective then so is their join. But the join of injective functions may not be injective, and the meet of surjective functions may not be surjective.

[3] The tools of Category Theory have this property, but they are sometimes cumbersome.

Definition 3.2 (joinable graphs)**.** Two graphs G and H are *joinable* if $\mathscr{A}_G = \mathscr{A}_H$, $\dot{G} \cap \vec{H} = \vec{G} \cap \dot{H} = \varnothing$ and the functions $\acute{G}$ and $\acute{H}$ (and similarly $\grave{G}$ and $\grave{H}$) are joinable. We then define the graphs

$$\begin{aligned} G \sqcap H &\stackrel{\text{def}}{=} (\ \dot{G} \cap \dot{H},\ \vec{G} \cap \vec{H},\ \acute{G} \curlywedge \acute{H},\ \grave{G} \curlywedge \grave{H},\ \mathscr{A}_G,\ \mathring{G} \cap \mathring{H}\ ), \\ G \sqcup H &\stackrel{\text{def}}{=} (\ \dot{G} \cup \dot{H},\ \vec{G} \cup \vec{H},\ \acute{G} \curlyvee \acute{H},\ \grave{G} \curlyvee \grave{H},\ \mathscr{A}_G,\ \mathring{G} \cup \mathring{H}\ ), \end{aligned}$$

that we respectively call the *intersection* and *union* of G and H.

Similarly, for any set I an I-indexed family $(G_i)_{i\in I}$ of graphs is *joinable* if its elements are pairwise joinable, and then for any Σ-algebra $\mathcal{A}$ such that $\mathcal{A} = \mathscr{A}_{G_i}$ for all $i \in I$, let

$$\bigsqcup_{i\in I} G_i \stackrel{\text{def}}{=} (\ \bigcup_{i\in I} \dot{G}_i,\ \bigcup_{i\in I} \vec{G}_i,\ \bigcurlyvee_{i\in I} \acute{G}_i,\ \bigcurlyvee_{i\in I} \grave{G}_i,\ \mathcal{A},\ \bigcup_{i\in I} \mathring{G}_i\ ).$$

Note that the algebra $\mathcal{A}$ is uniquely determined whenever $I \neq \varnothing$, and otherwise it will be obvious from the context. It is easy to see that these structures are graphs: the sets of vertices and arrows are disjoint and the source and target functions have the correct domains and codomains. Note that if G and H are joinable then $G \sqcap H = H \sqcap G \lhd G \lhd G \sqcup H = H \sqcup G$. Similarly, if $(G_i)_{i\in I}$ is joinable then $G_j \lhd \bigsqcup_{i\in I} G_i$ for all $j \in I$. We also see that any two subgraphs of G are joinable, and that $H \lhd G$ iff $G \sqcap H = H$ iff $G \sqcup H = G$. As above, on triples of pairwise joinable graphs, these operations are associative and distributive over each other.

These operations are convenient to define a natural way of removing objects (graph items and attributes) from a graph.

Definition 3.3. For any graph G, sets V, A and attribution l of G, we say that G *is disjoint from* V, A, l if $\dot{G} \cap V = \varnothing$, $\vec{G} \cap A = \varnothing$ and $\mathring{G}(x) \cap l(x) = \varnothing$ for all $x \in \dot{G} \cup \vec{G}$. We write $G \backslash [V, A, l]$ for the largest subgraph of G (w.r.t. $\lhd$) that is disjoint from V, A, l.

Since for any two subgraphs $F, F' \lhd G$ that are disjoint from V, A, l, the subgraph $F \sqcup F'$ of G is also disjoint from V, A, l, then the graph $G \backslash [V, A, l]$ always exists: it is the union of all subgraphs of G disjoint from V, A, l. It cannot contain dangling arrows; all arrows adjacent to an element of V are necessarily removed, even if they do not belong to A. We will therefore not be restricted by the *gluing condition* that is necessary in the algebraic Double-Pushout approach to graph rewriting, see [2, p. 45]. Similarly, $G \backslash [V, A, l]$ cannot contain dangling attributes.

The rest of the section is devoted to proving some fundamental properties of the operations defined so far. It is easy to see that removing objects is compatible with intersections, i.e., that $(G \sqcap H) \backslash [V, A, l] = (G \backslash [V, A, l]) \sqcap (H \backslash [V, A, l])$. It is less obvious that the same holds for unions.

Lemma 3.4. *For all joinable families $(G_i)_{i\in I}$ of graphs, all sets V, A and all attributions l of $\bigsqcup_{i\in I} G_i$, we have*

$$\Big(\bigsqcup_{i\in I} G_i\Big) \backslash [V, A, l] = \bigsqcup_{i\in I} (G_i \backslash [V, A, l]).$$

Proof. Since $G_j \lhd \bigsqcup_{i\in I} G_i$ for all $j \in I$ then $G_j \backslash [V, A, l] \lhd \big(\bigsqcup_{i\in I} G_i\big) \backslash [V, A, l]$, hence $\bigsqcup_{j\in I} (G_j \backslash [V, A, l]) \lhd \big(\bigsqcup_{i\in I} G_i\big) \backslash [V, A, l]$.

Conversely, let $H \lhd \bigsqcup_{i\in I} G_i$ such that H is disjoint from V, A, l. For all $f \in \vec{H}$ and all $a \in \mathring{H}(f)$ there exists an $i \in I$ such that $f \in \vec{G}_i$ and $a \in \mathring{G}_i(f)$. Let $x = \acute{H}(f)$ and $y = \grave{H}(f)$, so that f is an arrow from x to y. Obviously $f \notin A$, $x, y \notin V$ and $a \notin l(f)$. Since $x, y \in \dot{G}_i$, then the graph

$$\boxed{x \xrightarrow{f \mid a} y}$$

(or the corresponding unlabelled graph if there is no such a) is a subgraph of G_i disjoint from V, A, l, hence is a subgraph of $G_i \backslash [V, A, l]$ and therefore of $\bigsqcup_{j\in I} (G_j \backslash [V, A, l])$. Similarly, for all $x \in \dot{H}$ and all $a \in \mathring{H}(x)$ the graph

$$\boxed{x \mid a}$$

is a subgraph of $\bigsqcup_{j\in I} (G_j \backslash [V, A, l])$. Since H is the union of all such graphs then $H \lhd \bigsqcup_{j\in I} (G_j \backslash [V, A, l])$, and this holds for $H = \big(\bigsqcup_{i\in I} G_i\big) \backslash [V, A, l]$. $\square$

We now see that the removal of objects in a graph can easily be transported by isomorphism.

Lemma 3.5. *For all isomorphisms $\alpha : G \simeq H$, sets $V \subseteq \dot{G}$, $A \subseteq \vec{G}$ and attributions l of G, we have $\alpha(G \backslash [V, A, l]) = \alpha(G) \backslash [\alpha(V), \alpha(A), \mathring{\alpha} \circ l \circ \alpha^{-1}]$.*

Proof. For any subgraph $F \lhd G$, the subgraph

$$\alpha(F) = (\ \alpha(\dot{F}),\ \alpha(\vec{F}),\ \alpha \circ \acute{F} \circ \alpha^{-1},\ \alpha \circ \grave{F} \circ \alpha^{-1},\ \mathcal{B},\ \mathring{\alpha} \circ \mathring{F} \circ \alpha^{-1}\ )$$

of H is disjoint from $\alpha(V), \alpha(A), \mathring{\alpha} \circ l \circ \alpha^{-1}$ iff F is disjoint from V, A, l.

□

It is similarly easy to see that unions and intersections can be transported by a *single* isomorphism, i.e., if $\alpha : G \simeq H$ and $F, F' \lhd G$ then $\alpha(F \sqcap F') = \alpha(F) \sqcap \alpha(F')$ and $\alpha(F \sqcup F') = \alpha(F) \sqcup \alpha(F')$. However, unions are not invariant constructions since images of joinable graphs by *different* isomorphisms may not be joinable, and if they are joinable their union may not be isomorphic to the union of the original graphs. To ensure that this is the case, we need some extra conditions. We first see how to build new morphisms by joining existing morphisms.

Lemma 3.6. *For all joinable families $(\alpha_i : G_i \to H_i)_{i \in I}$ of morphisms such that $(G_i)_{i \in I}$ and $(H_i)_{i \in I}$ are joinable, then $\curlyvee_{i \in I} \alpha_i : \bigsqcup_{i \in I} G_i \to \bigsqcup_{i \in I} H_i$ is a morphism and it is surjective if the α_i's are surjective.*

Proof. Let $\alpha = \curlyvee_{i \in I} \alpha_i$, $G = \bigsqcup_{i \in I} G_i$ and $H = \bigsqcup_{i \in I} H_i$. For all $f \in \vec{G}$ there is an $i \in I$ such that $f \in \vec{G}_i$, hence $\alpha \circ \acute{G}(f) = \alpha_i \circ \acute{G}_i(f) = \acute{H}_i \circ \alpha_i(f) = \acute{H} \circ \alpha(f)$ and similarly $\alpha \circ \grave{G}(f) = \grave{H} \circ \alpha(f)$. For all $x \in \dot{G} \cup \vec{G}$,

$$\mathring{\alpha} \circ \mathring{G}(x) = \mathring{\alpha}\Big(\bigcup_{i \in I} \mathring{G}_i(x)\Big) \subseteq \bigcup_{i \in I} \mathring{\alpha}_i \circ \mathring{G}_i(x) \subseteq \bigcup_{i \in I} \mathring{H}_i \circ \alpha_i(x) = \mathring{H} \circ \alpha(x)$$

hence α is a morphism from G to H. The join of surjective functions is surjective. □

The following lemma provides the conditions required for building new isomorphisms by joining existing isomorphisms.

Lemma 3.7. *For all joinable families $(\alpha_i : G_i \simeq H_i)_{i \in I}$ of isomorphisms such that $(G_i)_{i \in I}$ is joinable, if $\alpha_i \curlywedge \alpha_j$ is surjective for all $i, j \in I$ then*

1. *$(H_i)_{i \in I}$ is joinable,*
2. *$\alpha_i \curlywedge \alpha_j : G_i \sqcap G_j \simeq H_i \sqcap H_j$ for all $i, j \in I$,*
3. *$\curlyvee_{i \in I} \alpha_i : \bigsqcup_{i \in I} G_i \simeq \bigsqcup_{i \in I} H_i$.*

Proof. For all $i, j \in I$, let $\alpha_{i,j} = \alpha_i \curlywedge \alpha_j$. Since α_i and α_j are injective then $\alpha_{i,j}$ is bijective from $\lfloor G_i \rfloor \cap \lfloor G_j \rfloor$ to $\lfloor H_i \rfloor \cap \lfloor H_j \rfloor$, and $\mathring{\alpha}_{i,j} = \mathring{\alpha}_i = \mathring{\alpha}_j$ is a Σ-isomorphism.

1. We first prove that $\dot{H}_i \cap \vec{H}_j = \varnothing$. Assume this is not true, then there is a $y \in \dot{H}_i \cap \vec{H}_j = \alpha_i(\dot{G}_i) \cap \alpha_j(\vec{G}_j) \subseteq \lfloor H_i \rfloor \cap \lfloor H_j \rfloor$, hence there is a $x \in \lfloor G_i \rfloor \cap \lfloor G_j \rfloor$ such that $y = \alpha_{i,j}(x) = \alpha_i(x) = \alpha_j(x)$, hence $x = \alpha_i^{-1}(y) \in \dot{G}_i$ and $x = \alpha_j^{-1}(y) \in \vec{G}_j$. But G_i and G_j are joinable, hence $\dot{G}_i \cap \vec{G}_j = \varnothing$, a contradiction. By symmetry between i and j we also get $\vec{H}_i \cap \dot{H}_j = \varnothing$.

   For all $g \in \vec{H}_i \cap \vec{H}_j$, let $f = \alpha_{i,j}^{-1}(g) \in \vec{G}_i \cap \vec{G}_j$, so that $\alpha_i(f) = \alpha_j(f) = g$. Since G_i and G_j are joinable, then $\acute{G}_i(f) = \acute{G}_j(f) \in \dot{G}_i \cap \dot{G}_j$, hence

   $$\acute{H}_i(g) = \acute{H}_i \circ \alpha_i(f) = \alpha_i \circ \acute{G}_i(f) = \alpha_j \circ \acute{G}_j(f) = \acute{H}_j \circ \alpha_j(f) = \acute{H}_j(g).$$

   Similarly we get $\grave{H}_i(g) = \grave{H}_j(g)$, hence H_i and H_j are joinable.

2. For all $f \in \vec{G}_i \cap \vec{G}_j$ we have

   $$\alpha_{i,j} \circ (\acute{G}_i \curlywedge \acute{G}_j)(f) = \alpha_i \circ \acute{G}_i(f) = \acute{H}_i \circ \alpha_i(f) = (\acute{H}_i \curlywedge \acute{H}_j) \circ \alpha_{i,j}(f)$$

   and similarly $\alpha_{i,j} \circ (\grave{G}_i \curlywedge \grave{G}_j) = (\grave{H}_i \curlywedge \grave{H}_j) \circ \alpha_{i,j}$ and $\mathring{\alpha}_{i,j} \circ (\mathring{G}_i \cap \mathring{G}_j) = (\mathring{H}_i \cap \mathring{H}_j) \circ \alpha_{i,j}$, hence $\alpha_{i,j}$ is an isomorphism.

3. Let $\alpha = \curlyvee_{i \in I} \alpha_i$, $G = \bigsqcup_{i \in I} G_i$ and $H = \bigsqcup_{i \in I} H_i$, then by Lemma 3.6 $\alpha : G \to H$ is surjective. For all $x, y \in \lfloor G \rfloor$, if $\alpha(x) = \alpha(y)$ then there exist $i, j \in I$ such that $\alpha(x) = \alpha_i(x) = \alpha_j(y) = \alpha(y)$ and this image therefore belongs to $\lfloor H_i \rfloor \cap \lfloor H_j \rfloor$. Since $\alpha_{i,j}$ is surjective there is a $z \in \lfloor G_i \rfloor \cap \lfloor G_j \rfloor$ such that $\alpha_i(x) = \alpha_{i,j}(z) = \alpha_j(y)$, hence $x = z = y$ and α is a therefore bijective. For all $x \in \dot{G} \cup \vec{G}$ we have

   $$\mathring{\alpha} \circ \mathring{G}(x) = \mathring{\alpha}\Big(\bigcup_{i \in I} \mathring{G}_i(x)\Big) = \bigcup_{i \in I} \mathring{\alpha}_i \circ \mathring{G}_i(x) = \bigcup_{i \in I} \mathring{H}_i \circ \alpha_i(x) = \mathring{H} \circ \alpha(x).$$

   hence α is an isomorphism.

□

4 Parallel Transformations

The concept of rewrite rules and parallel transformations will be introduced and motivated on an example running throughout the section. This example is the simultaneous assignment $a, b \coloneqq b, a$ from the programming language Python, that specifies in a concise and elegant way the swapping of the contents of a and b. Our angle is to decompose this expression as $a \coloneqq b \parallel b \coloneqq a$ and view it as the parallel application of two rules on an environment where a and b have values, say 0 and 1. This environment can be represented as an attributed graph, say in the following way:

$$G \;=\; \boxed{x \mid a, 0 \qquad y \mid b, 1}$$

where a and b are constants of sort `idtf`.

4.1 Rules for parallel transformations

Rewrite rules will be expressed by means of graphs attributed with sets of Σ-terms. We assume two variables u and v of sort `nat`. As explained in Section 1, graph rewrite rules have a left- and a right-hand side L and R, and an interface $K \lhd L$ specifying the part of L that should not be removed.

The left-hand side of a rewrite rule corresponding to $a \coloneqq b$ should match the relevant part of the environment, hence it should be

$$L_1 \;=\; \boxed{x_1 \mid a, u \qquad y_1 \mid b, v}$$

Before replacing the value u of a by v it should be removed from a, and of course nothing else shall be removed, hence

$$K_1 \;=\; \boxed{x_1 \mid a \qquad y_1 \mid b, v} \;\lhd\; L_1.$$

The right-hand side may now specify that the resulting environment should attribute the value v to a. The obvious choice is the graph

$$\boxed{x_1 \mid a, v \qquad y_1 \mid b, v}$$

but then the rule would also state that b should keep its content v; this would obviously conflict with the rule representing $b \coloneqq a$. In order to consistently apply the rules in parallel, we need to scale down this right-hand side by dropping the unnecessary reference to b. Hence

$$R_1 \;=\; \boxed{x_1 \mid a, v}$$

We end up with the rules $r_1 = (L_1, K_1, R_1)$ for $a \coloneqq b$ and $r_2 = (L_2, K_2, R_2)$ for $b \coloneqq a$, where

$$L_2 = \boxed{x_2 \mid a, u \qquad y_2 \mid b, v} \quad K_2 = \boxed{x_2 \mid a, u \qquad y_2 \mid b} \quad R_2 = \boxed{y_2 \mid b, u}$$

Thus K_1 and K_2 are not subgraphs of R_1 and R_2, and the intersections $L_1 \sqcap R_1$ and $L_2 \sqcap R_2$ are strict subgraphs of K_1 and K_2 respectively. These rules therefore have the shape pictured below.

The semantics of such rules can be informally described by ascribing different roles to the four different areas. The existence of a matching from L to some graph G is of course a necessary condition for the transformation of G. The images (by this matching) of objects of L (vertices, arrows and attributes) that do not belong to K, i.e., the area $L \backslash K$, have to be removed from G. Then, in order to find an image of R in the result H of the transformation, the image of $L \sqcap R$ in G has to be preserved in H, and images of the objects of $R \backslash L$ have to be added to G to obtain H.

This leaves the area $K \backslash R$ of images of the objects of K that do not belong to R as "intermediate" objects that do not have to be deleted nor have to be preserved, hence they are free to be either preserved or deleted. But this is not deterministic, hence another semantic property is needed to determine the fate of these objects: the condition of locality. More precisely, we formulate the condition of locality in the following way: *every object of G that does not have to be removed has to be preserved.* Hence the intermediate objects will normally be preserved, unless they are explicitly removed by some rule. This will be made precise in Section 4.2.

In the following definition we consider rules as finite syntactic objects, in conformity with the spirit of rule-based programming frameworks where term algebras are often used. Terms on the left-hand side allow the selection of attributes, and on the right-hand side they allow the computation of new attributes (this will be used in Section 6).

Definition 4.1 ((Σ, X)-graphs, rules, matchings)**.** For any finite $X \subseteq \mathscr{V}$, we call (Σ, X)-*graph* any finite graph G such that $\mathscr{A}_G = \mathscr{T}(\Sigma, X)$. Let

$$\mathrm{Var}(G) \;=\; \bigcup_{x \in \dot{G} \cup \vec{G}} \left(\bigcup_{t \in \mathring{G}(x)} \mathrm{Var}(t) \right),$$

where $\mathrm{Var}(t)$ is the set of variables occurring in t, see e.g. [6, p. 37].

A *rule* r is a triple (L, K, R) of (Σ, X)-graphs such that L and R are joinable, $L \sqcap R \lhd K \lhd L$ and $\mathrm{Var}(L) = X$ (see Remark 4.2 below). If $L \sqcap R = K$ then r is said to be *standard*.

A *matching of* r *in* a graph G is a matching μ from L to G that is *consistent*, i.e., such that $\mathring{\mu}(\mathring{L}(x) \backslash \mathring{K}(x)) \cap \mathring{\mu}(\mathring{K}(x)) = \varnothing$ (or equivalently $\mathring{\mu}(\mathring{L}(x) \backslash \mathring{K}(x)) = \mathring{\mu}(\mathring{L}(x)) \backslash \mathring{\mu}(\mathring{K}(x))$) for all $x \in \dot{K} \cup \vec{K}$. We denote $\mathscr{M}(r, G)$ the set of all matchings of r in G (they all have domain $\lfloor L \rfloor$).

We consider sets $\mathcal{R}$ of rules with the following condition: for all pairs of rules $r, r' \in \mathcal{R}$, if $(L, K, R) = r \neq r' = (L', K', R')$ then $\lfloor L \rfloor \neq \lfloor L' \rfloor$ and hence $\mathscr{M}(r, G) \cap \mathscr{M}(r', G) = \varnothing$ for any graph G. Let $\mathscr{M}(\mathcal{R}, G) \stackrel{\text{def}}{=} \biguplus_{r \in \mathcal{R}} \mathscr{M}(r, G)$, then for any $\mu \in \mathscr{M}(\mathcal{R}, G)$ there is a unique rule $\mathrm{r}_\mu \in \mathcal{R}$ such that $\mu \in \mathscr{M}(\mathrm{r}_\mu, G)$, and its components are denoted $\mathrm{r}_\mu = (\mathrm{L}_\mu, \mathrm{K}_\mu, \mathrm{R}_\mu)$.

Remark 4.2. If X were allowed to contain a variable v not occurring in L, then v would freely match any element of $\mathscr{A}_G$ (that is possibly infinite) and the set $\mathscr{M}(r, G)$ would contain as many matchings with essentially the same effect. Also note that $\mathrm{Var}(R) \subseteq \mathrm{Var}(L)$ (since R is a $(\Sigma, \mathrm{Var}(L))$-graph), R and K are joinable and $R \sqcap K = L \sqcap R$.

Also note that for any matching μ from L to G, since $\mathring{\mu}$ may be non-injective (thus allowing distinct variables to match identical values), consistency is necessary to separate the attributes in G that should be removed from those that should be preserved by a rewrite step.

On the running example we see that $\mathscr{M}(\{r_1, r_2\}, G) = \{\mu_1, \mu_2\}$ where μ_1 and μ_2 are the matchings of r_1 and r_2 respectively in G given below.

	x_1	y_1	a	b	u	v
μ_1	x	y	a	b	0	1

	x_2	y_2	a	b	u	v
μ_2	x	y	a	b	0	1

4.2 Semantics of parallel transformations

For any set $M \subseteq \mathscr{M}(\mathcal{R}, G)$ of matchings in a graph G we now wish to define what is a parallel transformation from G to some graph H by the simultaneous application of all the rules specified by M, without assuming any order, while *preserving the semantics* of each of these rules. The set M of course provides the necessary condition for applying the rules (though not a sufficient one as we will see).

A first obvious property is that there should be matchings from the right-hand sides of these rules to H, i.e., for all $\mu \in M$ there exists a matching μ' from R_μ to H. Since R_μ intersects L_μ this matching should agree with μ on $\mathrm{R}_\mu \sqcap \mathrm{K}_\mu$, i.e., μ and μ' should be joinable. We therefore have $\mu'(\mathrm{R}_\mu) \lhd H$ and $\mathring{\mu}' = \mathring{\mu}$. On the running example let $M = \{\mu_1, \mu_2\}$, we see that μ'_1 and μ'_2 are given by

	x_1	a	b	u	v
μ'_1	x	a	b	0	1

	y_2	a	b	u	v
μ'_2	y	a	b	0	1

Note that R_1 and R_2 are $(\Sigma, \{u, v\})$-graphs. Thus

$$\mu'_1(R_1) \;=\; \boxed{x \mid a, 1} \qquad \mu'_2(R_2) \;=\; \boxed{y \mid b, 0}$$

The condition of locality now has to be interpreted for the whole of M. Therefore, the part of G that is not removed by *any* application of the rules has to be preserved in H. The vertices removed by r_μ for any $\mu \in M$ are the vertices of G matched by L_μ but not by K_μ, i.e., the elements of the set $\mu(\dot{\mathrm{L}}_\mu \backslash \dot{\mathrm{K}}_\mu)$. Hence all the vertices in $V = \bigcup_{\mu \in M} \mu(\dot{\mathrm{L}}_\mu \backslash \dot{\mathrm{K}}_\mu)$ are removed by the transformation, and similarly all arrows in a set $A \subseteq \vec{G}$ and all elements of an attribution l on G. We therefore have $G \backslash [V, A, l] \lhd H$. Note that, due to possible overlaps, some "intermediate" objects of some rule may belong to V, A or l, and should therefore be removed.

This can be illustrated on the running example, where $V = A = \varnothing$ since no vertex or arrow is removed, only attributes are removed; 0 is removed from $\mathring{G}(x)$ through μ_1 and 1 is removed from $\mathring{G}(y)$ through μ_2. The vertex y does not belong to $\mu_1(L_1 \sqcap R_1)$ and is therefore an intermediate object of μ_1, as are its attributes b and 1. Among these, only 1 is removed (by μ_2), b and therefore y are preserved. Symmetrically, x and its attributes a and 0 are intermediate objects of μ_2, of which 0 is removed by μ_1, while a and x are preserved in the result H.

The previous conditions only provide lower bounds for H. But H need not contain anything else than what is preserved from G and the images of the right-hand sides, hence $H = G\backslash[V, A, l] \sqcup \bigsqcup_{\mu\in M} \mu'(\mathrm{R}_\mu)$. Thus

$$H \;=\; \boxed{x \mid a \qquad y \mid b} \sqcup \boxed{x \mid a, 1} \sqcup \boxed{y \mid b, 0} \;=\; \boxed{x \mid a, 1 \qquad y \mid b, 0}$$

As such, this equation does not guarantee the effective creation of new vertices and arrows corresponding to the vertices or arrows in $\mathrm{R}_\mu\backslash\mathrm{K}_\mu$, hence the third item in Definition 4.3 below. This condition clearly depends on the images of the right-hand sides, hence on $(\mu')_{\mu\in M}$, and will be illustrated in Example 4.4 below. Since attributes are added by inclusion the condition of effective creation does not apply to them.

This equation does not guarantee either the effective deletion of all objects in V, A, l, that depend on M, since some r_ν may restore what was removed by r_μ (if this objects belongs to $\nu(\mathrm{L}_\nu \sqcap \mathrm{R}_\nu)$). Such conflicts should be avoided if the semantics of the individual rules is to be respected. This leads to a further condition on M, the fourth item in Definition 4.3. However, a distinction again has to be made between graph items and attributes. We see in our running example by examining the result H that 0 has been removed from $\mathring{G}(x)$ as required by r_1 through μ_1. But assume that the rules r_1 and r_2 are applied to the graph

$$G' \;=\; \boxed{x \mid a, 0 \qquad y \mid b, 0}$$

through the matchings

	x_1	y_1	a	b	u	v
ν_1	x	y	a	b	0	0

	x_2	y_2	a	b	u	v
ν_2	x	y	a	b	0	0

then the result of the transformation is $H' = G'$ and we do not observe that 0 has been deleted from $\mathring{G}'(x)$ as required by r_1. Yet this result is correct since 0 has been added to $\mathring{G}'(x)$ by the right-hand side of r_2. Hence the values (through matchings $\nu \in M$) of terms $t \in \mathring{\mathrm{R}}_\nu(x)\backslash\mathring{\mathrm{K}}_\nu(x)$ for some $x \in \dot{\mathrm{R}}_\nu \cup \vec{\mathrm{R}}_\nu$ should be allowed in H, even if they are deleted by some rule. A formal definition can now be endeavoured.

Definition 4.3 (parallel transformation $\Vdash_M$)**.** Let G, H be two graphs, $\mathcal{R}$ a set of rules and $M \subseteq \mathscr{M}(\mathcal{R}, G)$, there is a *parallel transformation from G to H by M*, and we write $G \Vdash_M H$ if for all $\mu \in M$ there exists a matching μ' from R_μ to H such that

- μ' and μ are joinable for all $\mu \in M$,
- $H \;=\; G\backslash[\mathrm{V}_M, \mathrm{A}_M, \ell_M] \sqcup \bigsqcup_{\mu\in M} \mu'(\mathrm{R}_\mu)$ where

$$\mathrm{V}_M \stackrel{\text{def}}{=} \bigcup_{\mu\in M} \mu(\dot{\mathrm{L}}_\mu\backslash\dot{\mathrm{K}}_\mu),\ \mathrm{A}_M \stackrel{\text{def}}{=} \bigcup_{\mu\in M} \mu(\vec{\mathrm{L}}_\mu\backslash\vec{\mathrm{K}}_\mu),\ \ell_M \stackrel{\text{def}}{=} \bigcup_{\mu\in M} \mathring{\mu} \circ (\mathring{\mathrm{L}}_\mu\backslash\mathring{\mathrm{K}}_\mu) \circ \mu^{-1},$$

- (*effective creation*) G is disjoint from $\mathrm{V}'_M, \mathrm{A}'_M, \varnothing$ where

$$\mathrm{V}'_M \stackrel{\text{def}}{=} \bigcup_{\mu\in M} \mu'(\dot{\mathrm{R}}_\mu\backslash\dot{\mathrm{K}}_\mu),\ \mathrm{A}'_M \stackrel{\text{def}}{=} \bigcup_{\mu\in M} \mu'(\vec{\mathrm{R}}_\mu\backslash\vec{\mathrm{K}}_\mu),$$

- (*effective deletion*) H is disjoint from $\mathrm{V}_M, \mathrm{A}_M, \ell_M\backslash\ell'_M$ where

$$\ell'_M \stackrel{\text{def}}{=} \bigcup_{\mu\in M} \mathring{\mu} \circ (\mathring{\mathrm{R}}_\mu\backslash\mathring{\mathrm{K}}_\mu) \circ \mu'^{-1}.$$

We then say that H is *obtained by* $(\mu')_{\mu\in M}$.

Note that ℓ'_M is only defined on the subgraph $\bigsqcup_{\mu\in M} \mu'(\mathrm{R}_\mu \sqcap \mathrm{K}_\mu)$ of H; as mentioned in Section 2.2, ℓ'_M is implicitly extended to the suitable domain by mapping other vertices and arrows to $\varnothing$.

Example 4.4. In order to illustrate the parallel deletion and creation of vertices and arrows, we consider only unlabelled graphs, and therefore a rule $r = (L, K, R)$ with no variable and

$$L \;=\; \boxed{x \text{ —— } f \text{ ⟶ } y} \quad K \;=\; \boxed{x} \quad R \;=\; \boxed{z \text{ —— } g \text{ ⟶ } x}$$

This is a standard rule that removes an arrow f and its target y, and creates a new vertex z and a new arrow g from z to the source x of f. The input graph is

$$G \;=\; \boxed{a_1 \text{ —— } f_1 \text{ ⟶ } b \text{ ⟵ } f_2 \text{ —— } a_2}$$

There are exactly two matchings μ_1 and μ_2 of r in G, given by

	x	y	f
μ_1	a_1	b	f_1

	x	y	f
μ_2	a_2	b	f_2

With $M = \{\mu_1, \mu_2\}$ we easily see that $\mathrm{V}_M = \mu_1(y) \cup \mu_2(y) = \{b\}$, $\mathrm{A}_M = \mu_1(f) \cup \mu_2(f) = \{f_1, f_2\}$ and $\ell_M = \varnothing$, so that

$$G \backslash [\mathrm{V}_M, \mathrm{A}_M, \ell_M] \;=\; \boxed{a_1 \quad a_2}$$

A first possibility is to choose μ_1' and μ_2' that yield the following images:

$$\mu_1'(R) \;=\; \boxed{a_1 \longleftarrow g_1 \longrightarrow c} \qquad \mu_2'(R) \;=\; \boxed{c \longrightarrow g_2 \longrightarrow a_2}$$

and are easily seen to be joinable with μ_1 and μ_2 respectively. This yields

$$H_1 \;=\; \boxed{a_1 \longleftarrow g_1 \longrightarrow c \longrightarrow g_2 \longrightarrow a_2}$$

Since $\mathrm{V}_M' = \mu_1'(z) \cup \mu_2'(z) = \{c\}$ and $\mathrm{A}_M' = \mu_1'(g) \cup \mu_2'(g) = \{g_1, g_2\}$ do not occur in G, the property of effective creation holds, and since H_1 is disjoint from $\{b\}, \{f_1, f_2\}, \varnothing$ then so does effective deletion. We conclude that $G \Vdash_M H_1$.

Another possibility is to choose μ_1' and μ_2' so that

$$\mu_1'(R) \;=\; \boxed{a_1 \longleftarrow g_1 \longrightarrow c_1} \qquad \mu_2'(R) \;=\; \boxed{c_2 \longrightarrow g_2 \longrightarrow a_2}$$

and that are also joinable with μ_1 and μ_2 respectively. This yields

$$H_2 \;=\; \boxed{a_1 \longleftarrow g_1 \longrightarrow c_1 \qquad c_2 \longrightarrow g_2 \longrightarrow a_2}$$

Since $\mathrm{V}_M' = \mu_1'(z) \cup \mu_2'(z) = \{c_1, c_2\}$ and $\mathrm{A}_M' = \mu_1'(g) \cup \mu_2'(g) = \{g_1, g_2\}$ do not occur in G, the property of effective creation holds, and since H_2 is also disjoint from $\{b\}, \{f_1, f_2\}, \varnothing$ then so does effective deletion. We conclude that $G \Vdash_M H_2$. There are no other possibilities than these two, up to isomorphism.

This example shows that the graphs $\mu_1'(R)$ and $\mu_2'(R)$ may or may not overlap. This is a situation similar to amalgamation (see Section 10); the result depends on the choice of μ''s hence the relation $\Vdash_M$ is not deterministic.

Example 4.5. We now apply the rule r of Example 4.4 to the graph

$$G \;=\; \boxed{a \longrightarrow f_1 \longrightarrow b \longrightarrow f_2 \longrightarrow c}$$

There are exactly two matchings μ_1 and μ_2 of r in G, given by

	x	y	f
μ_1	a	b	f_1

	x	y	f
μ_2	b	c	f_2

With $M = \{\mu_1, \mu_2\}$ we see that $\mathrm{V}_M = \mu_1(y) \cup \mu_2(y) = \{b, c\}$. However, since μ_2 and μ_2' must be joinable then $\mu_2'(x) = \mu_2(x) = b$ and therefore b belongs to $\mu_2'(R) \lhd H$. This means that effective deletion cannot hold, hence there is no H such that $G \Vdash_M H$. The reason is that the two applications of r clash on b: it should be removed according to μ_1 and preserved according to μ_2.

More generally, if there is a rule that removes a vertex (or an arrow), and a rule that preserves a vertex or an arrow, it is easy to build a graph and two matchings that similarly clash. Note that in Example 4.5 the graph G can be obtained by gluing together two copies of L.

5 Parallel Rewriting

In order to obtain a deterministic parallel transformation we need to determine the μ''s from M. In Example 4.4 we see that H_1 can be obtained as a homomorphic image of H_2 but not the reverse. It seems therefore reasonable to favour H_2 over H_1, hence to minimize the amount of overlap among the images of right-hand sides. Such overlaps cannot be completely avoided since these images may intersect with G. But it is possible to define the images of the right-hand sides (and the corresponding matchings) by ensuring that they only overlap in G. For this we create new vertices or arrows of the form (x, μ) for all $\mu \in M$.

Definition 5.1 (graph $G^{\uparrow}_{\mu}$ and matching $\mu\uparrow$)**.** For any rule $r = (L, K, R)$, graph G and matching $\mu \in \mathscr{M}(r, G)$ we define a graph $G^{\uparrow}_{\mu}$ together with a matching $\mu\uparrow : R \to G^{\uparrow}_{\mu}$.

Let $\dot{G}^{\uparrow}_{\mu} \stackrel{\text{def}}{=} \mu(\dot{R} \cap \dot{K}) \cup ((\dot{R} \backslash \dot{K}) \times \{\mu\})$ and $\vec{G}^{\uparrow}_{\mu} \stackrel{\text{def}}{=} \mu(\vec{R} \cap \vec{K}) \cup ((\vec{R} \backslash \vec{K}) \times \{\mu\})$. Then, let $\mu\uparrow$ be defined by: $\mathring{\mu}\uparrow \stackrel{\text{def}}{=} \mathring{\mu}$ and the underlying graph morphism (also denoted $\mu\uparrow$) is the function from $\dot{R} \cup \vec{R}$ to $\dot{G}^{\uparrow}_{\mu} \cup \vec{G}^{\uparrow}_{\mu}$ such that for all x, if $x \in \dot{K} \cup \vec{K}$ then $\mu\uparrow(x) \stackrel{\text{def}}{=} \mu(x)$ else $\mu\uparrow(x) \stackrel{\text{def}}{=} (x, \mu)$.

Finally, let

$$G^{\uparrow}_{\mu} \stackrel{\text{def}}{=} (\, \dot{G}^{\uparrow}_{\mu} \,,\ \vec{G}^{\uparrow}_{\mu} \,,\ \mu\uparrow \circ \acute{R} \circ \mu\uparrow^{-1} \,,\ \mu\uparrow \circ \grave{R} \circ \mu\uparrow^{-1} \,,\ \mathscr{A}_G \,,\ \mathring{\mu} \circ \mathring{R} \circ \mu\uparrow^{-1} \,).$$

Obviously $\mu\uparrow$ is joinable with μ and is a matching from R to $G^{\uparrow}_{\mu}$ such that $\mu\uparrow(R) = G^{\uparrow}_{\mu}$. We now prove that the graphs G and $G^{\uparrow}_{\mu}$'s are pairwise joinable and examine their intersection.

Lemma 5.2. *For every rule $r = (L, K, R)$, graph G and $\mu \in \mathscr{M}(r, G)$, the graphs G and $G^{\uparrow}_{\mu}$ are joinable, $\mu(R \sqcap K) \lhd G \sqcap G^{\uparrow}_{\mu}$ and $G \sqcap G^{\uparrow}_{\mu}$ has the same underlying graph as $\mu(R \sqcap K)$.*

Proof. It is obvious[4] that $\dot{G} \cap \vec{G}^{\uparrow}_{\mu} = \vec{G} \cap \dot{G}^{\uparrow}_{\mu} = \varnothing$ and $\vec{G} \cap \vec{G}^{\uparrow}_{\mu} = \mu(\vec{R} \cap \vec{K})$, hence for all $g \in \vec{G} \cap \vec{G}^{\uparrow}_{\mu}$ there is a $f \in \vec{R} \cap \vec{K}$ such that $g = \mu(f) = \mu\uparrow(f)$, hence

$$\acute{G}^{\uparrow}_{\mu}(g) = \acute{G}^{\uparrow}_{\mu} \circ \mu\uparrow(f) = \mu\uparrow \circ \acute{R}(f) = \mu \circ \acute{K}(f) = \acute{G} \circ \mu(f) = \acute{G}(g)$$

so that $\acute{G}^{\uparrow}_{\mu}$ and $\acute{G}$ are joinable and similarly for $\grave{G}^{\uparrow}_{\mu}$ and $\grave{G}$, hence $G^{\uparrow}_{\mu}$ and G are joinable.

We have $\mu(R \sqcap K) \lhd \mu(K) \lhd G$ and $\mu(R \sqcap K) = \mu\uparrow(R \sqcap K) \lhd \mu\uparrow(R) = G^{\uparrow}_{\mu}$, hence $\mu(R \sqcap K) \lhd G \sqcap G^{\uparrow}_{\mu}$. Besides, for all $y \in \dot{G} \cap \dot{G}^{\uparrow}_{\mu} = \mu(\dot{R} \cap \dot{K})$ there exists a $x \in \dot{R} \cap \dot{K}$ such that $\mu(x) = y$, hence $\dot{G} \cap \dot{G}^{\uparrow}_{\mu} \subseteq \mu(\dot{R} \cap \dot{K})$ and similarly $\vec{G} \cap \vec{G}^{\uparrow}_{\mu} \subseteq \mu(\vec{R} \cap \vec{K})$, hence $G \sqcap G^{\uparrow}_{\mu}$ and $\mu(R \sqcap K)$ have the same underlying graph. □

We next prove that $\left(G^{\uparrow}_{\mu}\right)_{\mu \in M}$ is joinable and examine their intersections.

Corollary 5.3. *For all $\mu, \nu \in \mathscr{M}(\mathcal{R}, G)$, the graphs $G^{\uparrow}_{\mu}$ and $G^{\uparrow}_{\nu}$ are joinable and, if $\mu \neq \nu$ then $G^{\uparrow}_{\mu} \sqcap G^{\uparrow}_{\nu}$ and $\mu(\mathrm{R}_{\mu} \sqcap \mathrm{K}_{\mu}) \sqcap \nu(\mathrm{R}_{\nu} \sqcap \mathrm{K}_{\nu})$ have the same underlying graph.*

Proof. If $\mu = \nu$ this is obvious, hence we may assume that $\mu \neq \nu$ so that $\vec{G}^{\uparrow}_{\mu} \cap \vec{G}^{\uparrow}_{\nu} = \mu(\vec{\mathrm{R}}_{\mu} \cap \vec{\mathrm{K}}_{\mu}) \cap \nu(\vec{\mathrm{R}}_{\nu} \cap \vec{\mathrm{K}}_{\nu}) \subseteq \vec{G}$, and since $G^{\uparrow}_{\mu}$ and $G^{\uparrow}_{\nu}$ are both joinable with G then they are joinable with each other and $G^{\uparrow}_{\mu} \sqcap G^{\uparrow}_{\nu} \lhd G$, so that $G^{\uparrow}_{\mu} \sqcap G^{\uparrow}_{\nu} = (G \sqcap G^{\uparrow}_{\mu}) \sqcap (G \sqcap G^{\uparrow}_{\nu})$, hence the result. □

We may therefore adopt the matchings $\mu\uparrow$ as matchings μ', yielding the following graph transformation.

Definition 5.4 ($G\|_M$, effective deletion property, $\Longrightarrow_{\mathcal{R}}$, $\Longmapsto_{\mathcal{R}}$)**.** For any set $\mathcal{R}$ of rules, any graph G and any $M \subseteq \mathscr{M}(\mathcal{R}, G)$, let

$$G\|_M \stackrel{\text{def}}{=} G \backslash [\mathrm{V}_M, \mathrm{A}_M, \ell_M] \sqcup \bigsqcup_{\mu \in M} G^{\uparrow}_{\mu}$$

with the same V_M, A_M and ℓ_M as above (see Definition 4.3). M has the *effective deletion property* if $G\|_M$ is disjoint from $\mathrm{V}_M, \mathrm{A}_M, \ell_M \backslash \ell^{\uparrow}_M$, where

$$\ell^{\uparrow}_M \stackrel{\text{def}}{=} \bigcup_{\mu \in M} \mathring{\mu} \circ (\mathring{\mathrm{R}}_{\mu} \backslash \mathring{\mathrm{K}}_{\mu}) \circ \mu\uparrow^{-1}.$$

Let $\Longrightarrow_{\mathcal{R}}$ be the relation of *parallel rewriting* defined by, for all graphs G and all $M \subseteq \mathscr{M}(\mathcal{R}, G)$ such that M has the effective deletion property,

$$G \Longrightarrow_{\mathcal{R}} G\|_M.$$

Let $\Longmapsto_{\mathcal{R}}$ be the relation of *full parallel rewriting* defined by, for all graphs G such that $\mathscr{M}(\mathcal{R}, G)$ has the effective deletion property,

$$G \Longmapsto_{\mathcal{R}} G\|_{\mathscr{M}(\mathcal{R}, G)}.$$

Example 5.5. We consider the same graph G, rule r and matchings μ_1, μ_2 as in Example 4.4. We have $\mathscr{M}(r, G) = \{\mu_1, \mu_2\}$, hence

$$G \Longmapsto_{\mathcal{R}} \boxed{a_1 \longleftarrow (g, \mu_1) \text{ —— } (z, \mu_1) \qquad (z, \mu_2) \text{ —— } (g, \mu_2) \longrightarrow a_2}$$

This graph is isomorphic to H_2.

[4] We assume that, as in the category **Sets**, functions are given with their codomains, so that μ "contains" its codomain $\lfloor G \rfloor$ and hence, by the axiom of regularity, G cannot "contain" μ.

The rest of the section is devoted to proving the properties of this particular transformation, in particular that it meets the requirements of a parallel transformation given in Definition 4.3. We also see that it is a most general one in the sense that, if M has the effective deletion property then any graph that can be obtained as a parallel transformation from G by M can also be obtained as a homomorphic image of $G\|_M$.

Theorem 5.6. *For any graphs G, H and any $M \subseteq \mathscr{M}(\mathcal{R}, G)$ that has the effective deletion property, we have $G \Vdash_M G\|_M$ and if $G \Vdash_M H$ is obtained by $(\mu')_{\mu \in M}$ then there exists $\alpha : G\|_M \to H$ surjective such that $\alpha \circ \mu\uparrow = \mu'$ for all $\mu \in M$.*

Proof. For all $\mu \in M$, $\mu\uparrow$ is a matching from R_μ to $G\|_M$ joinable with μ. Effective creation holds since

$$\mu\uparrow(\dot{\mathrm{R}}_\mu \backslash \dot{\mathrm{K}}_\mu) \cap \dot{G} = ((\dot{\mathrm{R}}_\mu \backslash \dot{\mathrm{K}}_\mu) \times \{\mu\}) \cap \dot{G} = \varnothing$$

and similarly $\mu\uparrow(\vec{\mathrm{R}}_\mu \backslash \vec{\mathrm{K}}_\mu) \cap \vec{G} = \varnothing$, hence G is disjoint from $\mathrm{V}'_M, \mathrm{A}'_M, \varnothing$ (with $\mu' = \mu\uparrow$). Effective deletion holds by hypothesis, hence $G \Vdash_M G\|_M$.

We now assume $G \Vdash_M H$ where H is obtained by $(\mu')_{\mu \in M}$, hence

$$G\|_M = G' \sqcup \bigsqcup_{\mu \in M} G^\uparrow_\mu \text{ and } H = G' \sqcup \bigsqcup_{\mu \in M} \mu'(\mathrm{R}_\mu) \text{ where } G' = G \backslash [\mathrm{V}_M, \mathrm{A}_M, \ell_M].$$

According to Lemma 3.6, we can build a morphism from $G\|_M$ to H by joining suitable morphisms $\alpha_\mu : G^\uparrow_\mu \to \mu'(\mathrm{R}_\mu)$ for all $\mu \in M$, together with a morphism from G' to itself; this will be $1_{G'}$.

Let α_μ have $\mu' \circ \mu\uparrow^{-1}$ as underlying graph morphism and $\mathring{\alpha}_\mu = 1_{\mathscr{A}_G}$. This is a bijective function that preserves adjacencies and such that

$$\mathring{G}^\uparrow_\mu \circ \alpha_\mu = (\mathring{\mu}' \circ \mathring{\mathrm{R}}_\mu \circ \mu'^{-1}) \circ \alpha_\mu = \mathring{\mu} \circ \mathring{\mathrm{R}}_\mu \circ \mu\uparrow^{-1} = \mathring{G}^\uparrow_\mu = \mathring{\alpha}_\mu \circ \mathring{G}^\uparrow_\mu,$$

hence α_μ is an isomorphism. We now see that $(1_{G'}, \alpha_\mu)_{\mu \in M}$ is joinable.

For all $\mu \in M$ and all $y \in \dot{G}' \cap \dot{G}^\uparrow_\mu = \mu(\dot{\mathrm{R}}_\mu \cap \dot{\mathrm{K}}_\mu)$, there exists a $x \in \dot{\mathrm{R}}_\mu \cap \dot{\mathrm{K}}_\mu$ such that $y = \mu(x) = \mu\uparrow(x)$, hence by joinability of μ and μ' we have $\alpha_\mu(y) = \mu'(x) = \mu(x) = y$. Similarly, $\alpha_\mu(f) = f$ for all $f \in \vec{G}' \cap \vec{G}^\uparrow_\mu$, hence $1_{G'}$ and α_μ are joinable.

For all $\mu, \nu \in M$ such that $\mu \neq \nu$ and for all $y \in \dot{G}^\uparrow_\mu \cap \dot{G}^\uparrow_\nu$, by Corollary 5.3 $y \in \mu(\dot{\mathrm{R}}_\mu \cap \dot{\mathrm{K}}_\mu) \cap \nu(\dot{\mathrm{R}}_\nu \cap \dot{\mathrm{K}}_\nu)$ hence $\alpha_\mu(y) = y = \alpha_\nu(y)$ and similarly for arrows, hence α_μ and α_ν are joinable.

Lemma 3.6 thus yields that $1_{G'} \curlyvee \bigvee_{\mu \in M} \alpha_\mu : G\|_M \to H$ is surjective. □

The morphism that yields H as a homomorphic image of $G\|_M$ may not be unique, unless the algebra used in G and H is reachable.

Corollary 5.7. *If $\mathscr{A}_G$ is reachable then α is the unique morphism such that $\alpha \circ \mu\uparrow = \mu'$ for all $\mu \in M$.*

Proof. The underlying graph morphism of α is determined by G', the μ''s and $\mu\uparrow$'s, hence it is unique. For all $\mu \in M$, $\mathring{\mu}$ is a Σ-homomorphism from $\mathscr{T}(\Sigma, \mathrm{Var}(\mathrm{L}_\mu))$ to $\mathscr{A}_G$ that extends the unique Σ-homomorphism from $\mathscr{T}(\Sigma, \varnothing)$ to $\mathscr{A}_G$, that is surjective by hypothesis, hence $\mathring{\mu}$ is surjective. But $\alpha \circ \mu\uparrow = \mu'$ entails $\mathring{\alpha} \circ \mathring{\mu} = \mathring{\mu}$, hence $\mathring{\alpha} = 1_{\mathscr{A}_G}$ and α is therefore unique. □

By analogy with a notion from category theory, we can say in this case that $G\|_M$ is the initial graph among those that can be obtained as parallel transformations from G by M.

It is now possible to show that the construction of $G\|_M$ is invariant, in the sense that if $\alpha : G \simeq H$ and $M \subseteq \mathscr{M}(\mathcal{R}, G)$, then $G_M \simeq H_{\alpha \circ M}$ (note that $\alpha \circ M = \{\alpha \circ \mu \mid \mu \in M\} \subseteq \mathscr{M}(\mathcal{R}, H)$).

Lemma 5.8. *If $\alpha : G \simeq H$ and $M \subseteq \mathscr{M}(\mathcal{R}, G)$ then $G\|_M \simeq H\|_{\alpha \circ M}$.*

Proof. We are going to use Lemma 3.7 to build an isomorphism from $G' = G \sqcup \bigsqcup_{\mu \in M} G^\uparrow_\mu$ to $H' = H \sqcup \bigsqcup_{\mu \in M} H^\uparrow_{\alpha \circ \mu}$ by joining to α suitable isomorphisms $\alpha_\mu : G^\uparrow_\mu \simeq H^\uparrow_{\alpha \circ \mu}$ for all $\mu \in M$. Note that $(G, G^\uparrow_\mu)_{\mu \in M}$ is joinable by Lemma 5.2 and Corollary 5.3.

Let α_μ have $(\alpha \circ \mu)\uparrow \circ \mu\uparrow^{-1}$ as underlying graph isomorphism (from $G^\uparrow_\mu = \mu\uparrow(\mathrm{R}_\mu)$ to $H^\uparrow_{\alpha \circ \mu} = (\alpha \circ \mu)\uparrow(\mathrm{R}_\mu)$ underlying graphs) and $\mathring{\alpha}_\mu = \mathring{\alpha}$. Since

$$\mathring{H}^\uparrow_{\alpha \circ \mu} \circ \alpha_\mu = \mathring{\alpha} \circ \mathring{\mu} \circ \mathring{\mathrm{R}}_\mu \circ (\alpha \circ \mu)\uparrow^{-1} \circ \alpha_\mu = \mathring{\alpha} \circ \mathring{\mu} \circ \mathring{\mathrm{R}}_\mu \circ \mu\uparrow^{-1} = \mathring{\alpha}_\mu \circ \mathring{G}^\uparrow_\mu$$

then α_μ is an isomorphism. It is joinable with α since for all vertices and arrows y of $G \sqcap G_\mu^\uparrow$, by Lemma 5.2 there is a vertex or arrow x of $\mathrm{R}_\mu \sqcap \mathrm{K}_\mu$ such that $y = \mu(x) = \mu\uparrow(x)$, and we have

$$\alpha_\mu(y) = \alpha_\mu \circ \mu\uparrow(x) = (\alpha \circ \mu)\uparrow(x) = \alpha \circ \mu(x) = \alpha(y).$$

Besides, it is easy to see that $\alpha(G \sqcap G_\mu^\uparrow) = \alpha(G) \sqcap \alpha(G_\mu^\uparrow) = H \sqcap H_{\alpha\circ\mu}^\uparrow$ hence $\alpha \curlywedge \alpha_\mu$ is surjective.

For all $\nu \in M$ such that $\nu \neq \mu$ and for all vertices and arrows y of $G_\mu^\uparrow \sqcap G_\nu^\uparrow$, by Corollary 5.3 y belongs to $\mu(\mathrm{R}_\mu \sqcap \mathrm{K}_\mu) \sqcap \nu(\mathrm{R}_\nu \sqcap \mathrm{K}_\nu)$ hence $\alpha_\mu(y) = \alpha(y) = \alpha_\nu(y)$, so that α_μ and α_ν are joinable. Since

$$(\alpha_\mu \curlywedge \alpha_\nu)(G_\mu^\uparrow \sqcap G_\nu^\uparrow) = \alpha(G_\mu^\uparrow \sqcap G_\nu^\uparrow) = H_{\alpha\circ\mu}^\uparrow \sqcap H_{\alpha\circ\nu}^\uparrow,$$

then $\alpha_\mu \curlywedge \alpha_\nu$ is surjective. Lemma 3.7 thus yields $\beta = \alpha \curlyvee \bigvee_{\mu\in M} \alpha_\mu : G' \simeq H'$. We also see that

$$\alpha(\mathrm{V}_M) = \alpha\big(\bigcup_{\mu\in M} \mu(\dot{\mathrm{L}}_\mu \backslash \dot{\mathrm{K}}_\mu)\big) = \bigcup_{\mu\in M} \alpha \circ \mu(\dot{\mathrm{L}}_\mu \backslash \dot{\mathrm{K}}_\mu) = \mathrm{V}_{\alpha\circ M}$$

(note that $\mathrm{r}_{\alpha\circ\mu} = \mathrm{r}_\mu$ since $\alpha \circ \mu \in \mathscr{M}(\mathrm{r}_\mu, H)$) and similarly $\alpha(\mathrm{A}_M) = \mathrm{A}_{\alpha\circ M}$ and $\alpha \circ \ell_M \circ \alpha^{-1} = \ell_{\alpha\circ M}$. Hence by Lemma 3.5

$$\begin{aligned}
\beta(G\|_M) &= \alpha(G\backslash[\mathrm{V}_M, \mathrm{A}_M, \ell_M]) \sqcup \bigsqcup_{\mu\in M} \alpha_\mu(G_\mu^\uparrow) \\
&= H\backslash[\mathrm{V}_{\alpha\circ M}, \mathrm{A}_{\alpha\circ M}, \ell_{\alpha\circ M}] \sqcup \bigsqcup_{\mu\in M} H_{\alpha\circ\mu}^\uparrow \\
&= H\|_{\alpha\circ M}.
\end{aligned}$$

□

It is then easy to see that the exhaustive application of rules entails the expected determinism property.

Theorem 5.9. *The relation $\Longmapsto_{\mathcal{R}}$ is deterministic up to isomorphism.*

Proof. Let G and H be graphs and $\alpha : G \simeq H$, then $\mathscr{M}(\mathcal{R}, H) = \alpha \circ \mathscr{M}(\mathcal{R}, G)$, hence $H\|_{\mathscr{M}(\mathcal{R},H)} = H\|_{\alpha\circ\mathscr{M}(\mathcal{R},G)} \simeq G\|_{\mathscr{M}(\mathcal{R},G)}$ by Lemma 5.8. □

6 Cellular Automata

As illustrated in Example 4.5, the effective deletion property in Definition 5.4 restricts, for most rules $\mathcal{R}$, the graphs on which the transformation $\Longmapsto_{\mathcal{R}}$ can be applied. The problem is therefore to examine what kind of parallelism can be expressed under this restriction. The subject of the current section is to prove that the model of parallelism allowed for by Definition 5.4 encompasses the popular model of cellular automata.

Definition 6.1. A *cellular automaton* is a tuple $\mathfrak{a} = (S, f, v_1, \ldots, v_n)$ where S is a finite set of *states*, $n \in \mathbb{N}$, the *local rule* f is a function from S^{n+1} to S, and $v_1, \ldots, v_n$ are n distinct elements of $\mathbb{Z}^d \backslash \{(0, \ldots, 0)\}$ called the *neighborhood frame*, for some $d \geqslant 1$.

The elements of $\mathbb{Z}^d$ are called *cells*. A *configuration* of $\mathfrak{a}$ is a function c from $\mathbb{Z}^d$ to S, and we write $\mathscr{C}_\mathfrak{a}$ for the set of configurations. The *global transition map* of $\mathfrak{a}$ is the function $\mathrm{T}_\mathfrak{a}$ from $\mathscr{C}_\mathfrak{a}$ to $\mathscr{C}_\mathfrak{a}$ that maps every configuration c to the configuration $c' = \mathrm{T}_\mathfrak{a}(c)$ defined by $c'(v) \stackrel{\text{def}}{=} f(c(v), c(v + v_1), \ldots, c(v + v_n))$ for all $v \in \mathbb{Z}^d$.

In this definition we have assumed w.l.o.g. that the local rule may always depend on the state of the local cell. An algebra and a rule can now be associated to every cellular automaton $\mathfrak{a}$, and a graph to any configuration $c \in \mathscr{C}_\mathfrak{a}$.

Definition 6.2 (algebra $\mathcal{S}_\mathfrak{a}$, rule $\mathrm{r}_\mathfrak{a}$, graphs G^c)**.** For any cellular automaton $\mathfrak{a} = (S, f, v_1, \ldots, v_n)$, let $\Sigma_\mathfrak{a}$ be the signature with two sorts `state` and `neighbour`, with a constant s of sort `state` for every state $s \in S$, with n constants $\mathrm{n}_1, \ldots, \mathrm{n}_n$ of sort `neighbour` and with a function symbol t of type $\texttt{state}^{n+1} \to \texttt{state}$. We also consider the set $X = \{u_0, \ldots, u_n\}$ of $n + 1$ variables of sort `state`.

Let $\mathcal{S}_\mathfrak{a}$ be the $\Sigma_\mathfrak{a}$-algebra where `state` is interpreted by S, `neighbour` by $\{\mathrm{n}_1, \ldots, \mathrm{n}_n\}$, every constant by itself and t by the local rule f.

The rule associated to $\mathfrak{a}$ is $\mathrm{r}_{\mathfrak{a}} = (\mathrm{L}_{\mathfrak{a}}, \mathrm{K}_{\mathfrak{a}}, \mathrm{R}_{\mathfrak{a}})$ with the $(\Sigma_{\mathfrak{a}}, X)$-graphs

$$\mathrm{L}_{\mathfrak{a}} = \left[\begin{array}{ccc} & & x_1 \mid u_1 \\ & f_1 \mid \mathrm{n}_1 \nearrow & \\ x_0 \mid u_0 & & \vdots \\ & f_n \mid \mathrm{n}_n \searrow & \\ & & x_n \mid u_n \end{array}\right] \quad \mathrm{K}_{\mathfrak{a}} = \left[\begin{array}{ccc} & & x_1 \mid u_1 \\ & f_1 \mid \mathrm{n}_1 \nearrow & \\ x_0 & & \vdots \\ & f_n \mid \mathrm{n}_n \searrow & \\ & & x_n \mid u_n \end{array}\right]$$

$$\mathrm{R}_{\mathfrak{a}} = \boxed{x_0 \mid t(u_0, \ldots, u_n)}$$

Note that $\mathrm{r}_{\mathfrak{a}}$ is a non-standard rule.

Finally, for every configuration $c \in \mathscr{C}_{\mathfrak{a}}$ let G^c be the graph such that $\mathscr{A}_{\mathrm{G}^c} = \mathcal{S}_{\mathfrak{a}}$ and that is the union for all $v \in \mathbb{Z}^d$ of the graphs

$$\left[\begin{array}{ccc} & & v + v_1 \mid c(v + v_1) \\ & (v,1) \mid \mathrm{n}_1 \nearrow & \\ v \mid c(v) & & \vdots \\ & (v,n) \mid \mathrm{n}_n \searrow & \\ & & v + v_n \mid c(v + v_n) \end{array}\right]$$

Theorem 6.3. *For any cellular automaton $\mathfrak{a}$, configuration $c \in \mathscr{C}_{\mathfrak{a}}$ and graph H, we have*

$$\mathrm{G}^c \Longmapsto_{\mathrm{r}_{\mathfrak{a}}} H \quad \textit{iff} \quad H = \mathrm{G}^{\mathrm{T}_{\mathfrak{a}}(c)}.$$

Proof. Let $M = \mathscr{M}(\mathrm{r}_{\mathfrak{a}}, \mathrm{G}^c)$, for any cell $v \in \mathbb{Z}^d$ there is an obvious matching μ_v of $\mathrm{r}_{\mathfrak{a}}$ in G^c defined by

	x_0	x_i	f_i	u_0	u_i
μ_v	v	$v + v_i$	(v, i)	$c(v)$	$c(v + v_i)$

for all $1 \leqslant i \leqslant n$. There is no other matching that maps x_0 to v, hence $M = \{\mu_v \mid v \in \mathbb{Z}^d\}$. The rule $\mathrm{r}_{\mathfrak{a}}$ only removes the attribute of cells, hence $\mathrm{V}_M = \mathrm{A}_M = \ell_M((v,i)) = \varnothing$ and $\ell_M(v) = \{c(v)\}$ for all $v \in \mathbb{Z}^d$ and $1 \leqslant i \leqslant n$. Hence $\mathrm{G}^c \backslash [\mathrm{V}_M, \mathrm{A}_M, \ell_M]$ has the same underlying graph and arrow attributes as G^c, and its vertices are attributed with $\varnothing$. The right-hand side $\mathrm{R}_{\mathfrak{a}}$ adds to every vertex $\mu_v(x_0) = v$ the attribute

$$\begin{aligned} \mathring{\mu}_v(t(u_0, \ldots, u_n)) &= f(\mathring{\mu}_v(u_0), \ldots, \mathring{\mu}_v(u_n)) \\ &= f(c(v), c(v + v_1), \ldots, c(v + v_n)) \\ &= c'(v) \end{aligned}$$

where $c' = \mathrm{T}_{\mathfrak{a}}(c)$, hence $\mathrm{G}^c\|_M = \mathrm{G}^c \backslash [\mathrm{V}_M, \mathrm{A}_M, \ell_M] \sqcup \bigsqcup_{v \in \mathbb{Z}^d} \mu_v{\uparrow}(\mathrm{R}_{\mathfrak{a}}) = \mathrm{G}^{c'}$; this proves the only if part.

Finally, we have $\ell_M^{\uparrow}(v) = \{c'(v)\}$, i.e., the attribute added by $\mu_v{\uparrow}(\mathrm{R}_{\mathfrak{a}})$ for all $v \in \mathbb{Z}^d$, hence $\mathrm{G}^{c'}$ is disjoint from $\mathrm{V}_M, \mathrm{A}_M, \ell_M \backslash \ell_M^{\uparrow}$. This proves that M has the effective deletion property and therefore that $\mathrm{G}^c \Longmapsto_{\mathrm{r}_{\mathfrak{a}}} \mathrm{G}^{\mathrm{T}_{\mathfrak{a}}(c)}$. □

Example 6.4. Conway's Game of Life [13] is a 2-dimensional cellular automaton $\mathfrak{g}$ with two states 0 (for *dead*) and 1 (for *alive*) and the Moore neighborhood frame, i.e., $v_1, \ldots, v_8$ are the non-zero elements of $\{-1, 0, 1\}^2$. The automaton is defined by the local rule

$$f(s_0, \ldots, s_8) = \begin{cases} 1 & \text{if } \sum_{i=1}^8 s_i = 3 \\ s_0 & \text{if } \sum_{i=1}^8 s_i = 2 \\ 0 & \text{otherwise.} \end{cases}$$

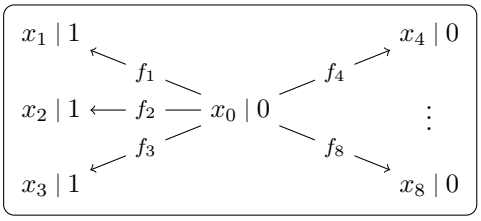
It can be translated into a single rule $\mathrm{r}_{\mathfrak{g}}$ as above, with $n = 8$. However, most problems in computer science involving the Game of Life are restricted to *finite* configurations, i.e., those configurations c where $c^{-1}(1)$ is finite. Yet for all configurations c the set $\mathscr{M}(\mathrm{r}_{\mathfrak{g}}, \mathrm{G}^c)$ is infinite. For the sake of computability we therefore need a set $\mathcal{R}$ of rules with the property that $\mathscr{M}(\mathcal{R}, \mathrm{G}^c)$ is finite whenever c is finite.

For any matching μ of $\mathrm{r}_{\mathfrak{g}}$ we write $\mu(\mathrm{r}_{\mathfrak{g}})$ for the triple $(\mu(\mathrm{L}_{\mathfrak{g}}), \mu(\mathrm{K}_{\mathfrak{g}}), \mu(\mathrm{R}_{\mathfrak{g}}))$, that may or may not be a rule; note however that $\mu(\mathrm{R}_{\mathfrak{g}})$ is well defined even though $\mathrm{R}_{\mathfrak{g}}$ is not a subgraph of $\mathrm{L}_{\mathfrak{g}}$. We use the fact that $f(0, \ldots, 0) = 0$, hence a cell may become alive only if it is in the neighborhood of a live cell. It is therefore possible to use rules whose left-hand sides match only those parts of c that contain a live cell. We thus let $\mathcal{R} = \{\mu_0(\mathrm{r}_{\mathfrak{g}}), \ldots, \mu_8(\mathrm{r}_{\mathfrak{g}})\}$ where $\mu_0, \ldots, \mu_8$ are matchings

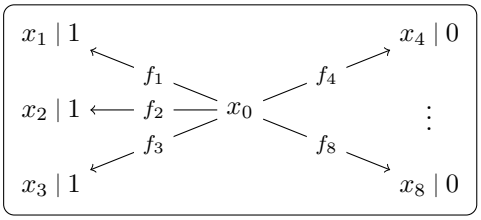
such that $\mathring{\mu}_0$ is the substitution $\{u_0 \mapsto 1\}$ and $\mathring{\mu}_i$ is the substitution $\{u_0 \mapsto 0, u_i \mapsto 1\}$ for all $1 \leqslant i \leqslant 8$ (and such that $\mathcal{R}$ is a valid set of rules, see Definition 4.1). It is obvious that every cell other than dead cells with a dead neighborhood are matched by exactly one rule in $\mathcal{R}$, hence $\mathscr{M}(\mathcal{R}, \mathrm{G}^c)$ is finite if c is finite, and these matchings always extend to matchings of $\mathrm{r}_{\mathfrak{g}}$ and hence yield the same transformation as $\mathrm{r}_{\mathfrak{g}}$.

Relying on further properties of f it is possible to achieve the same goal with fewer rules. More precisely, we use the facts that $f(0, s_1, \ldots, s_8)$ does not depend on the order of the s_i's, and that a cell can only be born if it has 3 live and 5 dead neighbours. We thus consider the *rule of birth* $\mathrm{r}_b = (\mathrm{L}_b, \mathrm{K}_b, \mathrm{R}_b)$ where:

$\mathrm{L}_b =$



$\mathrm{K}_b =$



$\mathrm{R}_b = \boxed{x_0 \mid 1}$

Let $\mathcal{R}' = \{\mu_0(\mathrm{r}_{\mathfrak{g}}), \mathrm{r}_b\}$, it is easy to see that r_b accounts for all births and $\mu_0(\mathrm{r}_{\mathfrak{g}})$ for all deaths (and survivals), so that $\mathcal{R}'$ again yields the same transformation as $\mathrm{r}_{\mathfrak{g}}$. However, there is no longer one matching of r_b on each relevant cell, but $3! \times 5! = 720$, still finite but not quite optimal. This problem is addressed in Sections 7 and 8.

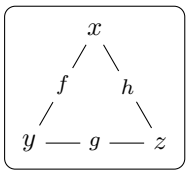
Compared to cellular automata, full parallel rewriting also allows to remove and create cells. The fact that graphs are non-geometric in nature, since vertices are not attached to coordinates, enables the use of more exotic topologies than $\mathbb{Z}^d$, e.g., the cells could be placed on a sphere or a torus. The topology defined by the neighborhoods need not be uniform and it can evolve from one configuration to the next.

7 Automorphism Groups of Rules

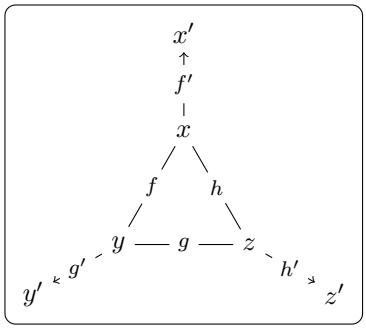
In this section, the notion of *automorphism group of a rule* is introduced, that will be used later to define a new refined parallel graph transformation. Indeed, using the full set of matchings might not be desirable in some cases, as illustrated below.

Example 7.1. Consider the rule $r = (L, K, R)$, where

$L = K =$

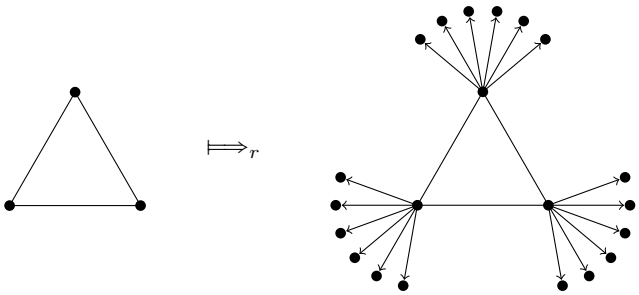


$R =$



Here, each edge f, g, h represents a pair of opposite arrows. We have $L = K \lhd R$, hence $L \sqcap R = K$ (this is a standard

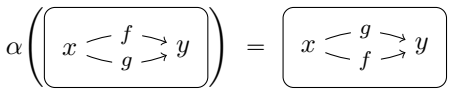
rule) and $L \sqcup R = R$. Obviously, this rule has six matchings in a triangle, hence



Thus the transformation of a simple triangle using the six matchings consists in adding to each vertex of the considered triangle six new adjacent vertices. However, this transformation may be counterintuitive because one may expect to add just one new adjacent vertex to each of the three vertices of the transformed triangle as depicted in the rewrite rule. To obtain such a result one may exploit the symmetries that occur in the graphs L, K and R. These symmetries are actually the automorphisms of the corresponding graphs. Indeed, for any graph G, the automorphisms α of G are the bijective morphisms from G to G such that $\mathring{\alpha} \circ \mathring{G} = \mathring{G} \circ \alpha$ (see Section 2), hence such that

$$\alpha(G) = (\, \alpha(\dot{G}),\, \alpha(\vec{G}),\, \acute{G},\, \grave{G},\, \mathscr{A}_G,\, \mathring{\alpha} \circ \mathring{G} \circ \alpha^{-1}\,) = G.$$

In group-theoretic terms (see, e.g., [14]), the set $\mathrm{Aut}(G)$ of automorphisms of G is the subgroup of the *symmetric group* $\mathrm{Sym}(\lfloor G \rfloor)$ of all permutations α of $\lfloor G \rfloor$ such that $\alpha(G) = G$. But for all such α we have $\alpha(\dot{G}) = \dot{G}$, $\alpha(\vec{G}) = \vec{G}$ and $\mathring{\alpha}$ is a Σ-automorphism of $\mathscr{A}_G$, i.e., an element of the set $\mathrm{Aut}(\mathscr{A}_G)$ of Σ-automorphisms of $\mathscr{A}_G$, that is again a group. Hence every α in $\mathrm{Aut}(G)$ is the join $\dot{\alpha} \curlyvee \vec{\alpha} \curlyvee \mathring{\alpha}$ of some $\dot{\alpha} \in \mathrm{Sym}(\dot{G})$, some $\vec{\alpha} \in \mathrm{Sym}(\vec{G})$ and some $\mathring{\alpha} \in \mathrm{Aut}(\mathscr{A}_G)$. In other words, $\mathrm{Aut}(G)$ is a subgroup of $\mathrm{Sym}(\dot{G}) \curlyvee \mathrm{Sym}(\vec{G}) \curlyvee \mathrm{Aut}(\mathscr{A}_G)$, that is indeed a subgroup of $\mathrm{Sym}(\lfloor G \rfloor)$. For example, if we take α such that $\dot{\alpha} = ()$, $\vec{\alpha} = (f\ g)$, $\mathring{\alpha} = ()$ (in cycle notation), and apply it to the following graph, we get



that is just another way of drawing exactly the same graph, hence $(f\ g)$ is an automorphism (a symmetry) of this graph. Note that symmetries may not be apparent in the drawing of a graph.

We also need to examine how the symmetries of the left and right hand sides of a rule interact; this is not obvious since the permuted sets are different and may intersect (as is the case in Example 7.1). To this purpose the notion of automorphism groups of graphs is now extended to their subgraphs.

Definition 7.2 (groups $\mathrm{Aut}_G(H_1, \ldots, H_n)$ and $\mathcal{S}|_H$)**.** For any $n \geqslant 1$ and any graphs $H, H_1, \ldots, H_n \lhd G$, let

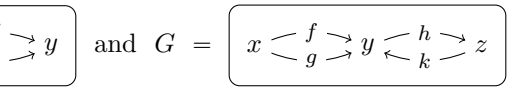
$$\mathrm{Aut}_G(H) \stackrel{\text{def}}{=} \{\alpha \in \mathrm{Sym}(\dot{G}) \curlyvee \mathrm{Sym}(\vec{G}) \curlyvee \mathrm{Aut}(\mathscr{A}_G) \mid \alpha(H) = H\}$$

$$\text{and } \mathrm{Aut}_G(H_1, \ldots, H_n) \stackrel{\text{def}}{=} \bigcap_{i=1}^{n} \mathrm{Aut}_G(H_i).$$

For any $\alpha \in \mathrm{Aut}_G(H)$, we write $\alpha|_H$ for $\alpha|_{\lfloor H \rfloor}$, and for any subgroup $\mathcal{S}$ of $\mathrm{Aut}_G(H)$, let $\mathcal{S}|_H \stackrel{\text{def}}{=} \{\alpha|_H \mid \alpha \in \mathcal{S}\}$; this is a subgroup of $\mathrm{Aut}(H)$.

It is obvious that $\mathrm{Aut}_G(G) = \mathrm{Aut}(G)$. We see that $\mathrm{Aut}_G(H)$ is a permutation group on $\lfloor G \rfloor$, but only the structure of H is involved in the constraint $\alpha(H) = H$, not the structure of G.

Example 7.3. Take for instance the unlabelled graphs



We have

$$\mathrm{Aut}(H) = \{1_H,\, (x)(y)(f\ g)\} \text{ and } \mathrm{Aut}(G) = \{1_G,\, (x)(y)(z)(f\ g)(h)(k)\}$$

(we write fixpoints in order to make the domains explicit). However, in $\mathrm{Aut}_G(H)$ the permutations of objects that do not belong to H are free, hence

$$\begin{aligned}\mathrm{Aut}_G(H) &= \{1_G,\ (x)(y)(z)(f\ g)(h)(k),\ (x)(y)(z)(f)(g)(h\ k),\\ &\qquad\qquad (x)(y)(z)(f\ g)(h\ k)\}\\ &= \mathrm{Aut}(H) \curlyvee \{(z)(h)(k),\ (z)(h\ k)\}\\ &= \mathrm{Aut}(H) \curlyvee \{(z)\} \curlyvee \{(h)(k),\ (h\ k)\}\\ &= \mathrm{Aut}(H) \curlyvee \mathrm{Sym}(z) \curlyvee \mathrm{Sym}(h,\ k).\end{aligned}$$

It is easy to see that $\mathrm{Aut}_G(H) = \mathrm{Aut}(H) \curlyvee \mathrm{Sym}(\dot{G}\backslash\dot{H}) \curlyvee \mathrm{Sym}(\vec{G}\backslash\vec{H})$ always holds and hence that $\mathrm{Aut}_G(H)|_H = \mathrm{Aut}(H)$. This means that, compared to the elements of $\mathrm{Aut}(H)$ that are only permutations of $\lfloor H \rfloor$, the elements of $\mathrm{Aut}_G(H)$ are *all* possible extensions of the elements of $\mathrm{Aut}(H)$ to permutations of $\lfloor G \rfloor$. This allows us to conveniently intersect the automorphism groups of joinable graphs, as are the graphs L, K and R involved in a rule (see Definition 4.1).

One last point that should be accounted for is that we are interested in the matchings from L to G, hence the symmetries of a rule are only relevant through their action on L. This leads to the following definition.

Definition 7.4 (group $\mathrm{Aut}(r)$)**.** For any rule $r = (L, K, R)$, the *automorphism group* of r is

$$\mathrm{Aut}(r) \stackrel{\text{def}}{=} \mathrm{Aut}_{L \sqcup R}(L, K, R)|_L.$$

It is obvious that $\mathrm{Aut}(r)$ is a subgroup of $\mathrm{Aut}(L)$.

Example 7.5. Let $r = (L, K, R)$ be the rule of example 7.1, where $L = K \lhd R$, hence $L \sqcup R = R$. It is well-known that the symmetric group $\mathrm{Sym}(1, 2, 3)$ has 6 elements and is generated by the permutations (1 2) and (1 3). Thus $\mathrm{Aut}(L)$ is the group generated by the permutations $(x\ y)(h\ g)$ and $(x\ z)(f\ g)$ (for the sake of simplicity the edges f, g, h are not explicitly expanded as pairs of arrows), and has 6 elements. Hence $\mathrm{Aut}_R(L)$ is generated by $(x\ y)(h\ g)$, $(x\ z)(f\ g)$, $(x'\ y')$, $(x'\ z')$, $(f'\ g')$ and $(g'\ h')$; it has 6^3 elements. However, the group $\mathrm{Aut}(R)$ is generated by $(x\ y)(x'\ y')(h\ g)(f'\ g')$ and $(x\ z)(x'\ z')(f\ g)(f'\ h')$; it has 6 elements and is included in $\mathrm{Aut}_R(L)$, thus $\mathrm{Aut}_R(L, R) = \mathrm{Aut}(R)$. Finally, the group $\mathrm{Aut}(r) = \mathrm{Aut}(R)|_L$ is generated by $(x\ y)(x'\ y')(h\ g)(f'\ g')|_L = (x\ y)(h\ g)$ and $(x\ z)(x'\ z')(f\ g)(f'\ h')|_L = (x\ z)(f\ g)$, hence $\mathrm{Aut}(r) = \mathrm{Aut}(L)$.

In this example the groups are finite, but the possibly infinite algebra of terms that occur in a rule has not been considered. It is however easy to see that these groups are always finite.

Lemma 7.6. *For any (Σ, X)-graph G and rule r, the set $\mathscr{M}(r, G)$ is finite.*

Proof. Let L be the left part of r and $Y = \mathrm{Var}(L)$. All elements of $\mathscr{M}(r, G)$ can be obtained as $\mu \curlyvee \mathring{\mu}$, where μ is an injective morphism between the underlying graphs of L and G, and $\mathring{\mu}$ a Σ-homomorphism from $\mathscr{T}(\Sigma, Y)$ to $\mathscr{T}(\Sigma, X)$. μ belongs to the finite set of injections from $\dot{L} \cup \vec{L}$ to $\dot{G} \cup \vec{G}$. Since $\mathscr{T}(\Sigma, Y)$ is free with generating set Y in the class of Σ-algebras, every Σ-homomorphism $\mathring{\mu}$ is determined by $\mathring{\mu}|_Y$. For every $v \in Y$ there is an $x \in \dot{L} \cup \vec{L}$ and a $t \in \mathring{L}(x)$ such that $v \in \mathrm{Var}(t)$, and since $\mathring{\mu}(t) \in \mathring{G}(\mu(x))$ then $\mathring{\mu}(v)$ belongs to the set of subterms of the elements of $\mathring{G}(\mu(x))$, that is finite. Hence there is a finite set of possible functions $\mathring{\mu}|_Y$. □

Note that if there were a variable v in the term algebra of a rule that did not occur in its left hand side (see Remark 4.2), then v could freely be matched to any element of the algebra of the input (Σ, X)-graph G, and $\mathscr{M}(r, G)$ could then be infinite.

Corollary 7.7. *The group* $\mathrm{Aut}(r)$ *is finite.*

Proof. $\mathrm{Aut}(r)$ is a subgroup of $\mathrm{Aut}(L)$ that is a subset of $\mathscr{M}(r, L)$. □

Corollary 7.8. *If $\mathcal{R}$ is finite and G is a (Σ, X)-graph then such is $G\|_M$ for all $M \subseteq \mathscr{M}(\mathcal{R}, G)$.*

Proof. $M \subseteq \biguplus_{r \in \mathcal{R}} \mathscr{M}(r, G)$ is finite hence $G\|_M = G\backslash[\mathrm{V}_M, \mathrm{A}_M, \ell_M] \sqcup \bigsqcup_{\mu \in M} G_\mu^{\uparrow}$ is obtained as a finite union of finite graphs, it is therefore finite. □

If G is not a (Σ, X)-graph then the set $\mathscr{M}(r, G)$ may not be finite, even if G is finite. In practice it is often necessary to apply the rules of $\mathcal{R}$ to graphs G where $\mathscr{A}_G$ is not an algebra of terms, e.g., the additive algebra of integers, but it is then possible to recover finiteness by imposing ad-hoc restrictions on the terms that occur in the rules, e.g., the term $u + v$ (where u and v are variables) cannot be allowed since it has infinite matchings with any integer.

8 Parallel Rewriting modulo Automorphism

Considering again Example 7.1, we would like to consider rewrite steps in which only one matching is selected among the 6 possible ones ; and more generally to select a subset M of $\mathscr{M}(\mathcal{R}, G)$ for defining a rewrite relation that yields more natural and concise graphs. The difficulty is to maintain determinism, i.e., to avoid an arbitrary choice of matchings. A key point is that the elements of M need not be selected in a deterministic way. Indeed, if a non-deterministic procedure for computing $M \subseteq \mathscr{M}(\mathcal{R}, G)$ is used, and if it is guaranteed that $G\|_{M'} \simeq G\|_M$ for all possible output M', then the corresponding rewrite relation is deterministic up to isomorphism.

Definition 8.1 (equivalence relation $\sim$, sets $\overline{M}$, relation $\overset{\sim}{\Longmapsto}_{\mathcal{R}}$)**.** For any graph G, let $\sim$ be the equivalence relation on $\mathscr{M}(\mathcal{R}, G)$ defined by

$$\mu \sim \nu \text{ iff } \mu \circ \mathrm{Aut}(\mathrm{r}_\mu) = \nu \circ \mathrm{Aut}(\mathrm{r}_\nu).$$

(See Definition 4.1 for r_μ and r_ν). The equivalence class of μ is denoted $\overline{\mu}$. For any subset $M \subseteq \mathscr{M}(\mathcal{R}, G)$ we write $\overline{M}$ for the set $\bigcup_{\mu \in M} \overline{\mu}$.

For any set $\mathcal{R}$ of rules, let $\overset{\sim}{\Longmapsto}_{\mathcal{R}}$ be the relation of *(full) parallel rewriting modulo automorphism* defined by, for all graphs G,

$$G \overset{\sim}{\Longmapsto}_{\mathcal{R}} G\|_M$$

where M is any minimal set that has the effective deletion property and such that $\overline{M} = \mathscr{M}(\mathcal{R}, G)$.

The minimality of M means that it contains exactly one undetermined representative per equivalence class modulo $\sim$. These classes are described below.

Lemma 8.2. *For any graph G and $\mu \in \mathscr{M}(\mathcal{R}, G)$ we have $\overline{\mu} = \mu \circ \mathrm{Aut}(\mathrm{r}_\mu)$.*

Proof. For all $\nu \in \overline{\mu}$ we obviously have $\nu \in \nu \circ \mathrm{Aut}(\mathrm{r}_\nu) = \mu \circ \mathrm{Aut}(\mathrm{r}_\mu)$, hence $\overline{\mu} \subseteq \mu \circ \mathrm{Aut}(\mathrm{r}_\mu)$.

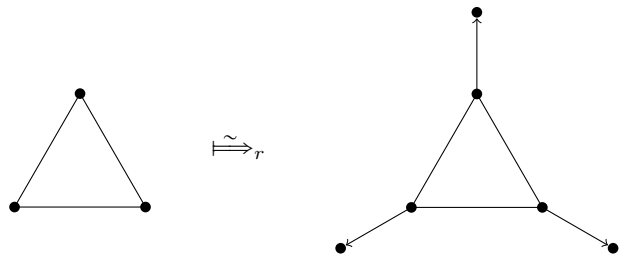
Conversely, assume that $\nu \in \mu \circ \mathrm{Aut}(\mathrm{r}_\mu)$, i.e., that there exists a $\sigma \in \mathrm{Aut}(\mathrm{r}_\mu)$ such that $\nu = \mu \circ \sigma$. Since $\mathrm{Aut}(\mathrm{r}_\mu)$ is a subgroup of $\mathrm{Aut}(\mathrm{L}_\mu)$ then $\mu \circ \sigma = \nu$ is a matching from L_μ to G. But ν is also a matching from L_ν to G, and according to the convention on $\mathcal{R}$ given in Definition 4.1 this entails $\mathrm{r}_\mu = \mathrm{r}_\nu$ and hence $\mathrm{Aut}(\mathrm{r}_\mu) = \mathrm{Aut}(\mathrm{r}_\nu)$. Since $\mu = \nu \circ \sigma^{-1} \in \nu \circ \mathrm{Aut}(\mathrm{r}_\mu)$, we have

$$\nu \circ \mathrm{Aut}(\mathrm{r}_\nu) \subseteq \mu \circ \mathrm{Aut}(\mathrm{r}_\mu) \circ \mathrm{Aut}(\mathrm{r}_\nu) = \mu \circ \mathrm{Aut}(\mathrm{r}_\mu) \subseteq \nu \circ \mathrm{Aut}(\mathrm{r}_\mu) = \nu \circ \mathrm{Aut}(\mathrm{r}_\nu),$$

hence $\mu \sim \nu$ and $\nu \in \overline{\mu}$; this proves that $\mu \circ \mathrm{Aut}(\mathrm{r}_\mu) \subseteq \overline{\mu}$. □

Note that matchings μ can only be $\sim$-equivalent if they are matchings of the same rule and have the same image $\mu(\mathrm{L}_\mu)$ in G. We also see that $|\overline{\mu}| \leqslant |\mathrm{Aut}(\mathrm{r}_\mu)|$ and that the equality holds if μ is injective. The more symmetric a rule is, the more matchings are likely to occur in the equivalence classes of matchings of this rule.

Example 8.3. Following Example 7.1, the 6 matchings from L to a triangle T are all equivalent, since for any such matching μ, $\overline{\mu} = \mu \circ \mathrm{Aut}(r)$ has the same number of elements as $\mathrm{Aut}(r)$, because μ is injective, and this number is 6 by Example 7.5. Hence $\overline{\mu} = \mathscr{M}(r, T)$ and obviously $\{\mu\}$ has the deletion property and is minimal, so that $T \overset{\sim}{\Longmapsto}_r T\|_{\{\mu\}}$. Graphically,



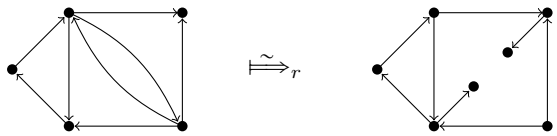
Example 8.4. We consider the rule $r = (L, K, R)$ with

$$L = \left[\begin{tikzcd} y \arrow[r, dash, "f'"] \arrow[d, dash] & x' \\ f \arrow[d] \arrow[ur, dash, bend left, "h"] \arrow[dr, dash, bend right, "h'"] & g' \arrow[u] \arrow[d, dash] \\ x & y' \arrow[l, "g"] \end{tikzcd}\right] \quad K = \left[\begin{tikzcd} y \arrow[r, dash, "f'"] \arrow[d, dash] & x' \\ f \arrow[d] & g' \arrow[u] \arrow[d, dash] \\ x & y' \arrow[l, "g"] \end{tikzcd}\right] \quad R = \left[\begin{tikzcd} y \arrow[r, dash, "f'"] \arrow[d, dash] & x' \\ f \arrow[d] \arrow[ur, "z"] & g' \arrow[u] \arrow[d, dash] \\ x & y' \arrow[l, "g"] \arrow[ul, "k"] \end{tikzcd}\right]$$

L has 4 symmetries, generated by $(x\ x')(f\ f')(g\ g')$ and $(y\ y')(f\ g)(f'\ g')(h\ h')$. K has 4 symmetries, but $\mathrm{Aut}_{L \sqcup R}(K) = \mathrm{Aut}(K) \curlyvee \mathrm{Sym}(z) \curlyvee \mathrm{Sym}(h, h', k)$ has 24 elements, generated by $(x\ x')(f\ f')(g\ g')$, $(y\ y')(f\ g)(f'\ g')$, $(h\ h')$ and $(h\ k)$. R

has two symmetries and $\mathrm{Aut}_{L\sqcup R}(R) = \mathrm{Aut}(R) \curlyvee \mathrm{Sym}(h, h')$ has 4 elements generated by $(y\ y')(f\ g)(f'\ g')$ and $(h\ h')$. The intersection of these groups is generated by $(y\ y')(f\ g)(f'\ g')(h\ h')$ hence $\mathrm{Aut}(r)$ has only two elements.

The rule r has 4 matchings in the graph below (because of the 4 symmetries of L), divided in two equivalence classes, hence



Example 8.5. Following Example 6.4, and as above we see that $\mathrm{Aut}(\mathrm{r}_b) = \mathrm{Aut}(\mathrm{L}_b)$ is the group generated by permutations $(x_1\ x_2)(f_1\ f_2)$, $(x_1\ x_3)(f_1\ f_3)$, $(x_4\ x_5)(f_4\ f_5)$ and $(x_4\ x_5\ x_6\ x_7\ x_8)(f_4\ f_5\ f_6\ f_7\ f_8)$; it is isomorphic to the group $\mathrm{Sym}(1,2,3) \times \mathrm{Sym}(1,2,3,4,5)$ and its cardinality is $3! \times 5! = 720$. For any finite configuration c, if there is a matching $\mu \in \mathscr{M}(\mathrm{r}_b, \mathrm{G}^c)$ then $\overline{\mu} = \mu \circ \mathrm{Aut}(\mathrm{r}_\mu) \subseteq \mathscr{M}(\mathrm{r}_b, \mathrm{G}^c)$. Since $\mathring{\mu} = 1_{\mathscr{T}(\Sigma,\varnothing)}$ is injective, the set $\overline{\mu}$ has 720 elements, i.e., all matchings $\nu \in \mathscr{M}(\mathrm{r}_b, \mathrm{G}^c)$ such that $\nu(x_0) = \mu(x_0)$ belong to the same equivalence class $\overline{\mu}$; the 720 matchings per birth cell are reduced to one. Since r_b does not create or remove vertices or arrows, we see that $\mathrm{G}^c\|_{\{\mu\}} = \mathrm{G}^c\|_{\overline{\mu}}$. Hence $\mathrm{G}^c \overset{\sim}{\Longmapsto}_{\mathcal{R}'} \mathrm{G}^{\mathrm{T}_{\mathfrak{g}}(c)}$ with the same number of matchings as $\mathrm{G}^c \Longmapsto_{\mathcal{R}} \mathrm{G}^{\mathrm{T}_{\mathfrak{g}}(c)}$.

If, as is the case in Example 8.5, the transformations $\Longmapsto_{\mathcal{R}}$ and $\overset{\sim}{\Longmapsto}_{\mathcal{R}}$ yield the same result, then rewriting modulo automorphism may improve the efficiency of the transformation compared to full parallel rewriting, since performance depends on the number of matchings. Other factors affecting performance are obviously the size of the input graphs and the number and size of rules, but also the complexity of computing and applying matchings. This cannot be determined in the present framework since the functions possibly involved in the Σ-algebra of the input graph are not even assumed to be computable.

The rest of the section is devoted to proving the properties of parallel rewriting modulo automorphism. We first examine the differences in rewriting a graph by $\overline{M}$ or by M. As shown by Example 7.1, the graph $G\|_{\overline{M}}$ may be quite different from (much bigger than) $G\|_M$. However, some objects relevant to computing $G\|_{\overline{M}}$ can be shown to depend only on the equivalence classes represented in M.

Lemma 8.6. *For any graph G and set $M \subseteq \mathscr{M}(\mathcal{R}, G)$ we have*

1. $\mathrm{V}_{\overline{M}} = \mathrm{V}_M$, $\mathrm{A}_{\overline{M}} = \mathrm{A}_M$, $\ell_{\overline{M}} = \ell_M$,
2. $\ell_{\overline{M}} \backslash \ell^{\uparrow}_{\overline{M}} = \ell_M \backslash \ell^{\uparrow}_M$,
3. $G \sqcap G\|_{\overline{M}} = G \sqcap G\|_M$.

Proof. For all $\mu \in M$ and $\nu \in \overline{\mu}$, by Lemma 8.2 there exists a $\sigma \in \mathrm{Aut}(\mathrm{r}_\mu)$ such that $\nu = \mu \circ \sigma$, thus $\mathrm{r}_\nu = \mathrm{r}_\mu$ and $\sigma \in \mathrm{Aut}(\mathrm{L}_\mu) \cap \mathrm{Aut}_{\mathrm{L}_\mu}(\mathrm{K}_\mu)$. By definition of $\mathrm{Aut}(\mathrm{r}_\mu)$ there also exists a $\tau \in \mathrm{Aut}_{\mathrm{L}_\mu \sqcup \mathrm{R}_\mu}(\mathrm{L}_\mu, \mathrm{K}_\mu, \mathrm{R}_\mu)$ such that $\sigma = \tau|_{\mathrm{L}_\mu}$, and let $\rho = \tau|_{\mathrm{R}_\mu}$ so that $\rho \in \mathrm{Aut}(\mathrm{R}_\mu)$ and $\mathring{\sigma} = \mathring{\tau} = \mathring{\rho}$ is a Σ-isomorphism.

1. Since ν and μ are injective on $\dot{\mathrm{L}}_\nu = \dot{\mathrm{L}}_\mu$ then
$$\nu(\dot{\mathrm{L}}_\nu \backslash \dot{\mathrm{K}}_\nu) = \nu(\dot{\mathrm{L}}_\nu) \backslash \nu(\dot{\mathrm{K}}_\nu) = \mu(\sigma(\dot{\mathrm{L}}_\mu) \backslash \sigma(\dot{\mathrm{K}}_\mu)) = \mu(\dot{\mathrm{L}}_\mu \backslash \dot{\mathrm{K}}_\mu)$$
and therefore
$$\mathrm{V}_{\overline{M}} = \bigcup_{\mu \in M} \bigcup_{\nu \in \overline{\mu}} \nu(\dot{\mathrm{L}}_\nu \backslash \dot{\mathrm{K}}_\nu) = \bigcup_{\mu \in M} \bigcup_{\nu \in \overline{\mu}} \mu(\dot{\mathrm{L}}_\mu \backslash \dot{\mathrm{K}}_\mu) = \mathrm{V}_M,$$
and similarly $\mathrm{A}_{\overline{M}} = \mathrm{A}_M$ and $\ell_{\overline{M}} = \ell_M$ (using consistency, see Definition 4.1).
2. The function $\ell_{\overline{M}}$ is an attribution on G hence is empty on all other vertices or arrows than those of G. For all $y \in \dot{G} \cup \vec{G}$, if y does not belong to $G^{\uparrow}_\nu$ then $\mathring{\nu} \circ (\mathring{\mathrm{R}}_\mu \backslash \mathring{\mathrm{K}}_\mu) \circ \nu\!\uparrow^{-1}(y) = \varnothing$, otherwise by Lemma 5.2 there is a vertex or arrow x of $\mathrm{R}_\mu \sqcap \mathrm{K}_\mu$ such that $\nu(x) = y = \nu\!\uparrow(x)$. Hence
$$\begin{aligned}\mathring{\nu} \circ (\mathring{\mathrm{R}}_\mu \backslash \mathring{\mathrm{K}}_\mu) \circ \nu\!\uparrow^{-1}(y) &= \mathring{\mu} \circ (\mathring{\rho} \circ \mathring{\mathrm{R}}_\mu(x) \backslash \mathring{\sigma} \circ \mathring{\mathrm{K}}_\mu(x)) \\ &= \mathring{\mu} \circ (\mathring{\mathrm{R}}_\mu \circ \rho(x) \backslash \mathring{\mathrm{K}}_\mu \circ \sigma(x)) \\ &= \mathring{\mu} \circ (\mathring{\mathrm{R}}_\mu \backslash \mathring{\mathrm{K}}_\mu) \circ \sigma(x) \\ &= \mathring{\mu} \circ (\mathring{\mathrm{R}}_\mu \backslash \mathring{\mathrm{K}}_\mu) \circ \mu\!\uparrow^{-1}(y)\end{aligned}$$
since $\rho(x) = \sigma(x)$ belongs to $\mathrm{R}_\mu \sqcap \mathrm{K}_\mu$ and $\mu\!\uparrow(\sigma(x)) = \mu \circ \sigma(x) = y$. Thus
$$(\ell_{\overline{M}} \backslash \ell^{\uparrow}_{\overline{M}})(y) = \ell_M(y) \backslash \bigcup_{\mu \in M} \bigcup_{\nu \in \overline{\mu}} \mathring{\nu} \circ (\mathring{\mathrm{R}}_\mu \backslash \mathring{\mathrm{K}}_\mu) \circ \nu\!\uparrow^{-1}(y) = (\ell_M \backslash \ell^{\uparrow}_M)(y).$$

3. By Lemma 5.2, $G \sqcap G^{\uparrow}_{\nu}$ has the same underlying graph as

$$\nu(\mathrm{R}_\mu \sqcap \mathrm{K}_\mu) = \mu(\rho(\mathrm{R}_\mu) \sqcap \sigma(\mathrm{K}_\mu)) = \mu(\mathrm{R}_\mu \sqcap \mathrm{K}_\mu),$$

hence as $G \sqcap G^{\uparrow}_{\mu}$. For any vertex or arrow y of this graph we have as above

$$\mathring{G}^{\uparrow}_{\nu}(y) = \mathring{\nu} \circ \mathring{\mathrm{R}}_\mu \circ \nu^{-1}(y) = \mathring{\mu} \circ \mathring{\mathrm{R}}_\mu \circ \mu^{-1}(y) = \mathring{G}^{\uparrow}_{\mu}(y)$$

hence $G \sqcap G^{\uparrow}_{\nu} = G \sqcap G^{\uparrow}_{\mu}$ and therefore

$$G \sqcap G\|_{\overline{M}} = G\backslash[\mathrm{V}_M, \mathrm{A}_M, \ell_M] \sqcup \bigsqcup_{\mu \in M} \bigsqcup_{\nu \in \overline{\mu}} G \sqcap G^{\uparrow}_{\nu} = G \sqcap G\|_M.$$

□

Corollary 8.7. *M has the effective deletion property iff $\overline{M}$ does.*

Proof. Since V_M (resp. A_M, resp. ℓ_M) is a subset of $\dot{G}$ (resp. a subset of $\vec{G}$, resp. an attribution on G) then $G\|_M$ is disjoint from $\mathrm{V}_M, \mathrm{A}_M, \ell_M \backslash \ell^{\uparrow}_M$ iff such is $G \sqcap G\|_M$, hence the result. □

Thus any minimal set M in Definition 8.1 has the effective deletion property if and only if $\mathscr{M}(\mathcal{R}, G)$ does; conflicts cannot be eliminated by factoring out equivalent matchings.

The case of the graph $G\|_M$ is more complicated, but it can be shown to depend only on the *number* of representatives in M of each equivalence class.

Lemma 8.8. *For any graph G and any $M, N \subseteq \mathscr{M}(\mathcal{R}, G)$, if there is a bijection ι from M to N such that $\iota(\mu) \sim \mu$ for all $\mu \in M$, then $G\|_M \simeq G\|_N$.*

Proof. We write ι_μ for $\iota(\mu)$. For all $\mu \in M$, $\iota_\mu \sim \mu$ entails $\mathrm{r}_{\iota_\mu} = \mathrm{r}_\mu$ and by Lemma 8.2 there is a $\sigma_\mu \in \mathrm{Aut}(\mathrm{r}_\mu)$ such that $\iota_\mu = \mu \circ \sigma_\mu$, and then a $\tau_\mu \in \mathrm{Aut}_{\mathrm{L}_\mu \sqcup \mathrm{R}_\mu}(\mathrm{L}_\mu, \mathrm{K}_\mu, \mathrm{R}_\mu)$ such that $\sigma_\mu = \tau_\mu|_{\mathrm{L}_\mu}$; let $\rho_\mu = \tau_\mu|_{\mathrm{R}_\mu}$ so that $\rho_\mu \in \mathrm{Aut}(\mathrm{R}_\mu)$ and $\mathring{\rho}_\mu = \mathring{\tau}_\mu = \mathring{\sigma}_\mu$.

We are going to use Lemma 3.7 to build an isomorphism from $F = G \sqcup \bigsqcup_{\mu \in M} G^{\uparrow}_{\mu}$ to $F' = G \sqcup \bigsqcup_{\mu \in M} G^{\uparrow}_{\iota_\mu}$ by joining to 1_G suitable isomorphisms $\alpha_\mu : G^{\uparrow}_{\mu} \simeq G^{\uparrow}_{\iota_\mu}$ for all $\mu \in M$.

Let α_μ have $\iota_\mu\!\uparrow \circ \rho_\mu^{-1} \circ \mu\!\uparrow^{-1}$ as underlying graph isomorphism (from $G^{\uparrow}_{\mu} = \mu\!\uparrow(\mathrm{R}_\mu)$ to $G^{\uparrow}_{\iota_\mu} = \iota_\mu\!\uparrow(\mathrm{R}_\mu)$ underlying graphs) and $\mathring{\alpha}_\mu = 1_{\mathscr{A}_G}$. We have

$$\begin{aligned}
\mathring{G}^{\uparrow}_{\iota_\mu} \circ \alpha_\mu &= \mathring{\iota}_\mu \circ \mathring{\mathrm{R}}_\mu \circ \iota_\mu\!\uparrow^{-1} \circ \alpha_\mu \\
&= \mathring{\iota}_\mu \circ \mathring{\mathrm{R}}_\mu \circ \rho_\mu^{-1} \circ \mu\!\uparrow^{-1} \\
&= \mathring{\iota}_\mu \circ \mathring{\rho}_\mu^{-1} \circ \mathring{\mathrm{R}}_\mu \circ \mu\!\uparrow^{-1} \text{ (since } \rho_\mu^{-1} \in \mathrm{Aut}(\mathrm{R}_\mu)) \\
&= \mathring{\iota}_\mu \circ \mathring{\sigma}_\mu^{-1} \circ \mathring{\mathrm{R}}_\mu \circ \mu\!\uparrow^{-1} \\
&= \mathring{\mu} \circ \mathring{\mathrm{R}}_\mu \circ \mu\!\uparrow^{-1} \\
&= \mathring{G}^{\uparrow}_{\mu} \\
&= \mathring{\alpha}_\mu \circ \mathring{G}^{\uparrow}_{\mu},
\end{aligned}$$

hence α_μ is an isomorphism. We next prove that it is joinable with 1_G.

Let U_μ be the underlying graph of $\mathrm{R}_\mu \sqcap \mathrm{K}_\mu$, then by Lemma 5.2 the underlying graphs of $G \sqcap G^{\uparrow}_{\mu}$ and $G \sqcap G^{\uparrow}_{\iota_\mu}$ are respectively $\mu(U_\mu)$ and $\iota_\mu(U_\mu)$. Since

$$\sigma_\mu(\mathrm{R}_\mu \sqcap \mathrm{K}_\mu) = \tau_\mu(\mathrm{R}_\mu \sqcap \mathrm{K}_\mu) = \rho_\mu(\mathrm{R}_\mu) \sqcap \sigma_\mu(\mathrm{K}_\mu) = \mathrm{R}_\mu \sqcap \mathrm{K}_\mu$$

then $\sigma_\mu(U_\mu) = U_\mu = \sigma_\mu^{-1}(U_\mu)$. For every vertex or arrow y of $G \sqcap G^{\uparrow}_{\mu}$ there exists $x \in \dot{U}_\mu \cup \vec{U}_\mu$ such that $y = \mu(x) = \mu\!\uparrow(x)$, thus

$$\alpha_\mu(y) = \iota_\mu\!\uparrow \circ \rho_\mu^{-1} \circ \mu\!\uparrow^{-1}(y) = \iota_\mu\!\uparrow \circ \rho_\mu^{-1}(x) = \iota_\mu \circ \sigma_\mu^{-1}(x) = \mu(x) = y$$

and hence 1_G and α_μ are joinable. But

$$(1_G \curlywedge \alpha_\mu)(\mu(U_\mu)) = \iota_\mu\!\uparrow \circ \rho_\mu^{-1}(U_\mu) = \iota_\mu\!\uparrow \circ \sigma_\mu^{-1}(U_\mu) = \iota_\mu\!\uparrow(U_\mu) = \iota_\mu(U_\mu),$$

(and $\mathring{1}_G \curlywedge \mathring{\alpha}_\mu = 1_{\mathscr{A}_G}$) hence $1_G \curlywedge \alpha_\mu$ is surjective.

For all $\nu \in M$ such that $\nu \neq \mu$ and for every vertex or arrow y of $G^{\uparrow}_{\mu} \sqcap G^{\uparrow}_{\nu}$, by Corollary 5.3 y belongs to $\mu(U_\mu) \sqcap \nu(U_\nu)$ hence $\alpha_\mu(y) = y = \alpha_\nu(y)$, so that α_μ and α_ν are also joinable. The function $\alpha_\mu \curlywedge \alpha_\nu$ is injective, hence

$$(\alpha_\mu \curlywedge \alpha_\nu)(\mu(U_\mu) \sqcap \nu(U_\nu)) = \alpha_\mu(\mu(U_\mu)) \sqcap \alpha_\nu(\nu(U_\nu)) = \iota_\mu(U_\mu) \sqcap \iota_\nu(U_\nu),$$

but ι is also injective so that $\iota_\nu \neq \iota_\mu$, hence by Corollary 5.3 this is the underlying graph of $G^{\uparrow}_{\iota_\mu} \sqcap G^{\uparrow}_{\iota_\nu}$. This proves that $\alpha_\mu \curlywedge \alpha_\nu$ is surjective.

Lemma 3.7 therefore yields $\alpha = 1_G \curlyvee \bigvee_{\mu \in M} \alpha_\mu : F \simeq F'$. We finally see that $\overline{M} = \overline{N}$ since $\iota_\mu \sim \mu$ for all $\mu \in M$ and ι is bijective from M to N, hence by Lemma 8.6 we have $\mathrm{V}_M = \mathrm{V}_{\overline{M}} = \mathrm{V}_{\overline{N}} = \mathrm{V}_N$ and similarly $\mathrm{A}_M = \mathrm{A}_N$ and $\ell_M = \ell_N$. Hence

$$\begin{aligned}
\alpha(G\|_M) &= 1_G(G\backslash[\mathrm{V}_M, \mathrm{A}_M, \ell_M]) \sqcup \bigsqcup_{\mu \in M} \alpha_\mu(G^{\uparrow}_{\mu}) \\
&= G\backslash[\mathrm{V}_N, \mathrm{A}_N, \ell_N] \sqcup \bigsqcup_{\mu \in M} G^{\uparrow}_{\iota_\mu} \\
&= G\|_N.
\end{aligned}$$

□

We conclude with the expected determinism property.

Theorem 8.9. *The relation $\overset{\sim}{\Longrightarrow}_{\mathcal{R}}$ is deterministic up to isomorphism.*

Proof. Let G and H be two graphs, $\alpha : G \simeq H$ and M, N any two sets such that $G \overset{\sim}{\Longrightarrow}_{\mathcal{R}} G\|_M$ and $H \overset{\sim}{\Longrightarrow}_{\mathcal{R}} H\|_N$. Since $\alpha \circ M \subseteq \mathscr{M}(\mathcal{R}, H) = \overline{N}$ then $\alpha \circ \overline{M} = \overline{\alpha \circ M} \subseteq \overline{N}$. By symmetry we also have $\alpha^{-1} \circ \overline{N} \subseteq \overline{M}$ hence $\overline{N} \subseteq \alpha \circ \overline{M}$ and therefore $\overline{\alpha \circ M} = \overline{N}$. The minimality of M such that $\overline{M} = \mathscr{M}(\mathcal{R}, G)$ obviously implies the minimality of $\alpha \circ \overline{M}$ such that $\alpha \circ \overline{M} = \mathscr{M}(\mathcal{R}, H)$ (since $\alpha \circ \mu \sim \alpha \circ \nu$ entails $\mu \sim \nu$), and since N is also minimal then each equivalence class of $\mathscr{M}(\mathcal{R}, H)$ is represented exactly once in $\alpha \circ M$ and once in N. Hence there exists a suitable bijection from $\alpha \circ M$ to N so that $H\|_{\alpha \circ M} \simeq H\|_N$ by Lemma 8.8. By Lemma 5.8 we have $G\|_M \simeq H\|_{\alpha \circ M}$, hence $G\|_M \simeq H\|_N$. □

9 Sequential and Parallel Independence

The parallel transformation defined as the graph $G\|_M$ in Definition 5.4 reduces to a standard notion of direct transformation when M contains only one matching, and hence to sequential rewriting, see Definition 9.1 below. Note that all singletons have the effective deletion property.

An obvious link between sequential and parallel rewriting is that when the matchings in M do not overlap, the graph $G\|_M$ can be obtained from G by sequential rewriting (up to isomorphism). In fact, it can be obtained in many different ways, by applying in sequence the elements of M in any order. But this property may still be true when the matchings do overlap. This is the case of the graph G' from the running example in Section 4.2. Indeed, when $a = b$ the assignments $a \coloneqq b$ and $b \coloneqq a$ can be evaluated in any order, they will yield the same result as the simultaneous assignment $a, b \coloneqq b, a$. Yet $\nu_1(L_1) = \nu_2(L_2) = G'$.

It is therefore possible for matchings in M to overlap and yet for the graph $G\|_M$ to be reachable from G by sequential rewriting. Of course, if there is a rule in $\mathcal{R}$ that has the same effect as this particular transformation, this is not surprising. In order to characterize the kind of overlaps that can be handled similarly by sequential and parallel rewriting, we need to express the (sequential) reachability of $G\|_M$ by the matchings in M (no other rule is allowed) in any order. This is known as sequential independence (see [11]).

One common difficulty with this notion is that the matchings applied in a sequence of direct transformations are matchings of rules in many different graphs, which makes it difficult to recognize that two sequences correspond to the same matchings in different order (see the notion of *shift equivalence* in [15, Section 3.5]). However, if this notion means that sequential and parallel rewriting meet, then we may use parallel transformations to express the result of sequences of direct transformations in any order. This leads to the following definition.

Definition 9.1 (sequential rewriting, sequential independence)**.** For any finite set of rules $\mathcal{R}$, we define the relation $\longrightarrow_{\mathcal{R}}$ by stating that, for all graphs G and all $\mu \in \mathscr{M}(\mathcal{R}, G)$,

$$G \longrightarrow_{\mathcal{R}} G\|_\mu$$

where $G\|_\mu$ stands for $G\|_{\{\mu\}}$, and similarly we will write V_μ for $\mathrm{V}_{\{\mu\}}$, etc.

We call *sequential rewriting* the relation $\longrightarrow^{\star}_{\mathcal{R}}$, i.e., the reflexive and transitive closure of $\longrightarrow_{\mathcal{R}}$.

For any graph G and set $M \subseteq \mathscr{M}(\mathcal{R}, G)$, we say that M is *sequential independent* if for all $M' \subseteq M$ and all $\mu \in M \backslash M'$,

- $\mu(\mathrm{L}_\mu) \lhd G\|_{M'}$, hence there is a canonical injection $j : \mu(\mathrm{L}_\mu) \to G\|_{M'}$,
- there exists an isomorphism α such that $\alpha(G\|_{M' \cup \{\mu\}}) = \big(G\|_{M'}\big)\|_{j \circ \mu}$ and $\alpha|_{\lfloor G \rfloor} = 1_G$.

The isomorphism α in Definition 9.1 is necessary to account for the difference between the isomorphic graphs $\mu{\uparrow}(\mathrm{R}_\mu)$ and $(j \circ \mu){\uparrow}(\mathrm{R}_\mu)$. It is possible to define precisely this isomorphism, but there is no need to be more specific than $\alpha|_{\lfloor G \rfloor} = 1_G$ in order to rule out any fortuitous isomorphism.

Note that this definition is designed so that any subset of a sequential independent set is obviously sequential independent. This simply expresses the fact that sequential independence cannot be lost by discarding matchings. A more surprising feature is that it tolerates infinite sets, although it is not generally possible to define infinite sequences of rewriting steps.

It is easy to see that if M is sequential independent and finite then $G\|_M$ can be obtained by sequential rewriting.

Lemma 9.2. *For all $M \subseteq \mathscr{M}(\mathcal{R}, G)$ finite and sequential independent there is a graph H such that $G \longrightarrow^\star_{\mathcal{R}} H$ and $H \simeq G\|_M$.*

Proof. By induction on the cardinality of M. It is trivial for $M = \varnothing$. Assume that $M = M' \uplus \{\mu\}$ then M' is finite sequential independent hence by induction hypothesis there exists H and $\beta : G\|_{M'} \simeq H$ such that $G \longrightarrow^\star_{\mathcal{R}} H$. By Definition 9.1 there is $j : \mu(\mathrm{L}_\mu) \to G\|_{M'}$ and $\alpha : G\|_M \simeq (G\|_{M'})\|_{\mu'}$ where $\mu' = j \circ \mu$. By Lemma 5.8 there exists $\gamma : (G\|_{M'})\|_{\mu'} \simeq H\|_{\beta \circ \mu'}$, hence $\gamma \circ \alpha(G\|_M) = \gamma((G\|_{M'})\|_{\mu'}) = H\|_{\beta \circ \mu'}$, hence $G \longrightarrow^\star_{\mathcal{R}} \gamma \circ \alpha(G\|_M)$ and the induction is complete. □

Of course there is usually more than one sequence of rewriting steps from G to the isomorphism class of $G\|_M$, since under the hypothesis of sequential independence these rewriting steps can be swapped; but without this hypothesis there is generally no sequence of rewriting steps from G to $G\|_M$. Besides, it remains to be seen wether $G\|_M$ may then also be obtained by parallel rewriting.

Sequential independence is not obviously a condition on overlaps, it relies on properties of the results of many graph transformations (exponentially many when M is finite). In order to characterize this property as a condition on overlaps, or more precisely as a condition pertaining to pairs of matchings in M and known as parallel independence, a careful analysis is required.

Parallel independence usually requires that the overlap of two matchings $\mu, \nu \in M$, i.e., the graph $\mu(\mathrm{L}_\mu) \sqcap \nu(\mathrm{L}_\nu)$ should be preserved by both direct transformations r_μ and r_ν, since if an object in the overlap were removed say by r_μ then r_ν would no longer match; this can be expressed as $\mu(\mathrm{L}_\mu) \sqcap \nu(\mathrm{L}_\nu) \lhd \mu(\mathrm{K}_\mu) \sqcap \nu(\mathrm{K}_\nu)$. Since this condition is required for all pairs of matchings, including (μ, ν) and (ν, μ), then it is equivalent to the condition $\mu(\mathrm{L}_\mu) \sqcap \nu(\mathrm{L}_\nu) \lhd \mu(\mathrm{K}_\mu)$, i.e., that the overlap is preserved by r_μ for all $\mu, \nu \in M$. However, this condition is not necessary for sequential independence, as the example of Section 4.2 shows. Indeed, we see that $\nu_1(L_1) \sqcap \nu_2(L_2) = G'$ is not a subgraph of

$$\nu_1(K_1) \;=\; \boxed{x \mid a \qquad y \mid b, 0}$$

Yet sequential independence holds because the object that is removed (the attribute 0 of x) is recovered by the right-hand side of r_1. It is therefore necessary to involve the graphs $\mu{\uparrow}(\mathrm{R}_\mu)$ and $\nu{\uparrow}(\mathrm{R}_\nu)$ in the condition on overlaps.

A first approximation is to require $\mu(\mathrm{L}_\mu) \sqcap \nu(\mathrm{L}_\nu) \lhd \mu(\mathrm{K}_\mu) \sqcup \mu{\uparrow}(\mathrm{R}_\mu)$ (an overlap object can be removed by r_μ if it is recovered by r_μ). This condition turns out to be necessary for sequential independence, but not sufficient. The reason is that an attribute removed (and therefore matched) by μ may not be matched by ν but can still be matched by $\nu{\uparrow}$.

Example 9.3. Consider a rule r_1 that removes the attribute 1 to a vertex, and a rule r_2 that adds the attribute 1 to a vertex.

$$r_1 = \Big(\boxed{x \mid 1}, \boxed{x}, \boxed{x}\Big)$$

$$G = \boxed{z \mid 1} \qquad H = \boxed{z}$$

$$r_2 = \Big(\boxed{y}, \boxed{y}, \boxed{y \mid 1}\Big)$$

The matchings of these rules in G are not sequential independent; if the first is applied before the second then the result is G, but if they are applied the other way round the result is H. Yet the overlap of the left-hand sides (the graph H) is a subgraph of the images of both right-hand sides.

It is therefore necessary to consider the overlap not only between left-hand sides but also with right-hand sides. Thanks to the symmetry between μ and ν this can be expressed in the following way.

Definition 9.4 (parallel independence)**.** For any graph G and set $M \subseteq \mathscr{M}(\mathcal{R}, G)$, we say that M is *parallel independent* if

$$\mu(\mathrm{L}_\mu) \sqcap (\nu(\mathrm{L}_\nu) \sqcup \nu{\uparrow}(\mathrm{R}_\nu)) \lhd \mu(\mathrm{K}_\mu) \sqcup \mu{\uparrow}(\mathrm{R}_\mu) \text{ for all } \mu, \nu \in M \text{ such that } \mu \neq \nu.$$

It is easy to see that this condition is violated in Example 9.3. Note that we only consider pairs of distinct matchings. The reason is that the parallel transformation does not allow to apply twice the same matching to a graph. It would of course be possible to do so (by considering multisets of matchings) but this would obviously hinder determinism, since there would be no limit to the number of applications of matchings. Thus any singleton is parallel independent.

Of course we expect to be able to rewrite in parallel with those sets of matchings that are parallel independent. But this is not obvious and requires to be proved.

Lemma 9.5. *For any graph G and set $M \subseteq \mathscr{M}(\mathcal{R}, G)$ if M is parallel independent then M has the effective deletion property.*

Proof. Let $H = G\|_M$. Since $\mathrm{V}_M \subseteq \dot{G}$ then by Lemma 5.2 we have

$$\begin{aligned}
\dot{H} \cap \mathrm{V}_M &= \bigcup_{\nu \in M} \nu(\dot{\mathrm{R}}_\nu \cap \dot{\mathrm{K}}_\nu) \cap \mathrm{V}_M \\
&= \bigcup_{\mu,\nu \in M} \nu(\dot{\mathrm{R}}_\nu \cap \dot{\mathrm{K}}_\nu) \cap \mu(\dot{\mathrm{L}}_\mu) \backslash \mu(\dot{\mathrm{K}}_\mu) \\
&\subseteq \bigcup_{\mu \neq \nu \in M} \nu(\dot{\mathrm{L}}_\nu) \cap \mu(\dot{\mathrm{L}}_\mu) \backslash \mu(\dot{\mathrm{K}}_\mu),
\end{aligned}$$

since $\nu(\dot{\mathrm{R}}_\nu \cap \dot{\mathrm{K}}_\nu) \subseteq \nu(\dot{\mathrm{K}}_\nu) \subseteq \nu(\dot{\mathrm{L}}_\nu)$.

Since M is parallel independent then $\mu(\mathrm{L}_\mu) \sqcap (\nu(\mathrm{L}_\nu) \sqcup G_\nu^{\uparrow}) \lhd \mu(\mathrm{K}_\mu) \sqcup G_\mu^{\uparrow}$ for all $\mu \neq \nu$, hence $\mu(\mathrm{L}_\mu) \sqcap \nu(\mathrm{L}_\nu) \lhd \mu(\mathrm{K}_\mu) \sqcup (G_\mu^{\uparrow} \sqcap G)$ and again by Lemma 5.2 $\mu(\dot{\mathrm{L}}_\mu) \cap \nu(\dot{\mathrm{L}}_\nu) \subseteq \mu(\dot{\mathrm{K}}_\mu) \cup \mu(\dot{\mathrm{R}}_\mu \cap \dot{\mathrm{K}}_\mu) = \mu(\dot{\mathrm{K}}_\mu)$. Hence $\dot{H} \cap \mathrm{V}_M = \varnothing$ and similarly $\vec{H} \cap \mathrm{A}_M = \varnothing$.

In order to prove that H is disjoint from V_M, A_M, $\ell_M \backslash \ell_M^{\uparrow}$, it remains to prove that $\mathring{H}(x) \cap \ell_M(x) \backslash \ell_M^{\uparrow}(x) = \varnothing$ for all $x \in \dot{H} \cup \vec{H}$. This is true if $x \notin \dot{G} \cup \vec{G}$ since then $\ell_M(x) = \varnothing$, hence we assume that $x \in \dot{G} \cup \vec{G}$, so that $\mathring{H}(x) \cap \ell_M(x) = \bigcup_{\mu \in M} \mathring{G}_\mu^{\uparrow} \cap \ell_M(x) = \bigcup_{\mu \in M} \mathring{\mu} \circ \mathring{\mathrm{R}}_\mu \circ \mu^{-1}(x) \cap \ell_M(x)$ and we need to prove that $\mathring{\mu} \circ \mathring{\mathrm{R}}_\mu \circ \mu^{-1}(x) \cap \ell_M(x) \backslash \ell_M^{\uparrow}(x) = \varnothing$ for all $\mu \in M$, or equivalently

$$\bigcup_{\nu \in M} \mathring{\mu} \circ \mathring{\mathrm{R}}_\mu \circ \mu^{-1}(x) \cap \mathring{\nu} \circ \mathring{\mathrm{L}}_\nu \circ \nu^{-1}(x) \backslash \mathring{\nu} \circ \mathring{\mathrm{K}}_\nu \circ \nu^{-1}(x) \subseteq \ell_M^{\uparrow}(x).$$

For any sets A and B we have $A = (A \cap B) \cup (A \backslash B)$, hence for all $\nu \in M$,

$$\begin{aligned}
&\mathring{\mu} \circ \mathring{\mathrm{R}}_\mu \circ \mu^{-1}(x) \cap \mathring{\nu} \circ \mathring{\mathrm{L}}_\nu \circ \nu^{-1}(x) \\
&= \big(\mathring{\mu} \circ (\mathring{\mathrm{R}}_\mu \cap \mathring{\mathrm{K}}_\mu) \circ \mu^{-1}(x) \cap \mathring{\nu} \circ \mathring{\mathrm{L}}_\nu \circ \nu^{-1}(x)\big) \\
&\qquad \cup \big(\mathring{\nu} \circ \mathring{\mathrm{L}}_\nu \circ \nu^{-1}(x) \cap \mathring{\mu} \circ (\mathring{\mathrm{R}}_\mu \backslash \mathring{\mathrm{K}}_\mu) \circ \mu^{-1}(x)\big) \\
&\subseteq \big(\mathring{\mu} \circ (\mathring{\mathrm{R}}_\mu \cap \mathring{\mathrm{K}}_\mu) \circ \mu^{-1}(x) \cap \mathring{\nu} \circ \mathring{\mathrm{L}}_\nu \circ \nu^{-1}(x)\big) \cup \ell_M^{\uparrow}(x).
\end{aligned}$$

If $\nu \neq \mu$, since M is parallel independent then

$$\begin{aligned}
&\mathring{\mu} \circ (\mathring{\mathrm{R}}_\mu \cap \mathring{\mathrm{K}}_\mu) \circ \mu^{-1}(x) \cap \mathring{\nu} \circ \mathring{\mathrm{L}}_\nu \circ \nu^{-1}(x) \\
&\subseteq \mathring{\mu} \circ \mathring{\mathrm{L}}_\mu \circ \mu^{-1}(x) \cap \mathring{\nu} \circ \mathring{\mathrm{L}}_\nu \circ \nu^{-1}(x) \\
&\subseteq \mathring{\nu} \circ \mathring{\mathrm{K}}_\nu \circ \nu^{-1}(x) \cup \mathring{\nu} \circ \mathring{\mathrm{R}}_\nu \circ \nu^{-1}(x) \\
&\subseteq \mathring{\nu} \circ \mathring{\mathrm{K}}_\nu \circ \nu^{-1}(x) \cup \mathring{\nu} \circ (\mathring{\mathrm{R}}_\nu \backslash \mathring{\mathrm{K}}_\nu) \circ \nu^{-1}(x) \\
&\subseteq \mathring{\nu} \circ \mathring{\mathrm{K}}_\nu \circ \nu^{-1}(x) \cup \ell_M^{\uparrow}(x),
\end{aligned}$$

hence $\mathring{\mu} \circ \mathring{\mathrm{R}}_\mu \circ \mu^{-1}(x) \cap \mathring{\nu} \circ \mathring{\mathrm{L}}_\nu \circ \nu^{-1}(x) \subseteq \mathring{\nu} \circ \mathring{\mathrm{K}}_\nu \circ \nu^{-1}(x) \cup \ell_M^{\uparrow}(x)$. We notice that this is also true when $\nu = \mu$ since $\mathrm{L}_\mu \sqcap \mathrm{R}_\mu \lhd \mathrm{K}_\mu$, hence

$$\mathring{\mu} \circ \mathring{\mathrm{R}}_\mu \circ \mu^{-1}(x) \cap \mathring{\nu} \circ \mathring{\mathrm{L}}_\nu \circ \nu^{-1}(x) \backslash \mathring{\nu} \circ \mathring{\mathrm{K}}_\nu \circ \nu^{-1}(x) \subseteq \ell_M^{\uparrow}(x)$$

for all $\nu \in M$. □

We now see that this notion of parallel independence is correct in the sense that it characterizes sequential independence. Note that the following result does not assume that M is finite.

Theorem 9.6. *For any graph G and set $M \subseteq \mathscr{M}(\mathcal{R}, G)$, M is parallel independent iff M is sequential independent.*

Proof. Only if part. For all $M' \subseteq M$ and $\mu \in M \backslash M'$, let $R = \bigsqcup_{\nu \in M'} G^{\uparrow}_{\nu}$ so that $G\|_{M'} = G \backslash [\mathrm{V}_{M'}, \mathrm{A}_{M'}, \ell_{M'}] \sqcup R$. For all $\nu \in M'$ we have $\mu(\mathrm{L}_\mu) \sqcap \nu(\mathrm{L}_\nu) \lhd \nu(\mathrm{K}_\nu) \sqcup G^{\uparrow}_{\nu}$ and $\mu(\mathrm{L}_\mu) \sqcap \nu(\mathrm{L}_\nu) \lhd G$, hence by Lemma 5.2

$$\mu(\dot{\mathrm{L}}_\mu) \cap \nu(\dot{\mathrm{L}}_\nu) \subseteq \nu(\dot{\mathrm{K}}_\nu) \cup \nu(\dot{\mathrm{R}}_\nu \cap \dot{\mathrm{K}}_\nu) = \nu(\dot{\mathrm{K}}_\nu)$$

or equivalently $\mu(\dot{\mathrm{L}}_\mu) \cap \nu(\dot{\mathrm{L}}_\nu) \backslash \nu(\dot{\mathrm{K}}_\nu) = \varnothing$. Thus

$$\mu(\dot{\mathrm{L}}_\mu) \cap \mathrm{V}_{M'} = \bigcup_{\nu \in M'} \mu(\dot{\mathrm{L}}_\mu) \cap \nu(\dot{\mathrm{L}}_\nu) \backslash \nu(\dot{\mathrm{K}}_\nu) = \varnothing$$

and therefore $\mu(\dot{\mathrm{L}}_\mu) \subseteq \dot{G}\|_{M'}$. Similarly we get $\mu(\vec{\mathrm{L}}_\mu) \subseteq \vec{G}\|_{M'}$. Then, for all $x \in \mu(\dot{\mathrm{L}}_\mu) \cup \mu(\vec{\mathrm{L}}_\mu)$, we have

$$\mathring{\mu} \circ \mathring{\mathrm{L}}_\mu \circ \mu^{-1}(x) \cap \mathring{\nu} \circ \mathring{\mathrm{L}}_\nu \circ \nu^{-1}(x) \subseteq \mathring{\nu} \circ \mathring{\mathrm{K}}_\nu \circ \nu^{-1}(x) \cup \mathring{G}^{\uparrow}_{\nu}(x)$$

hence

$$\mathring{\mu} \circ \mathring{\mathrm{L}}_\mu \circ \mu^{-1}(x) \cap \mathring{\nu} \circ \mathring{\mathrm{L}}_\nu \circ \nu^{-1}(x) \backslash \mathring{\nu} \circ \mathring{\mathrm{K}}_\nu \circ \nu^{-1}(x) \subseteq \mathring{G}^{\uparrow}_{\nu}(x) \subseteq \mathring{R}(x).$$

Thus

$$\mathring{\mu} \circ \mathring{\mathrm{L}}_\mu \circ \mu^{-1}(x) \cap \ell_{M'}(x) = \bigcup_{\nu \in M'} \mathring{\mu} \circ \mathring{\mathrm{L}}_\mu \circ \mu^{-1}(x) \cap \mathring{\nu} \circ \mathring{\mathrm{L}}_\nu \circ \nu^{-1}(x) \backslash \mathring{\nu} \circ \mathring{\mathrm{K}}_\nu \circ \nu^{-1}(x) \subseteq \mathring{R}(x)$$

and then

$$\mathring{\mu} \circ \mathring{\mathrm{L}}_\mu \circ \mu^{-1}(x) \subseteq \mathring{\mu} \circ \mathring{\mathrm{L}}_\mu \circ \mu^{-1}(x) \backslash \ell_{M'}(x) \cup \mathring{R}(x) \subseteq \mathring{G}\|_{M'}(x).$$

Therefore, $\mu(\mathrm{L}_\mu) \lhd G\|_{M'}$.

Let $j : \mu(\mathrm{L}_\mu) \to G\|_{M'}$ be the canonical injection and $\mu' = j \circ \mu$, so that $\mu' \in \mathscr{M}(\mathrm{r}_\mu, G\|_{M'})$, $\mu'(\mathrm{L}_\mu) = \mu(\mathrm{L}_\mu)$ and $\mu'(\mathrm{K}_\mu) = \mu(\mathrm{K}_\mu)$, hence $\mathrm{V}_{\mu'} = \mathrm{V}_\mu$, $\mathrm{A}_{\mu'} = \mathrm{A}_\mu$ and $\ell_{\mu'} = \ell_\mu$. Let $N = M' \cup \{\mu\}$, $H = G \sqcup R \sqcup \mu\uparrow(\mathrm{R}_\mu)$ and $H' = G \sqcup R \sqcup \mu'\uparrow(\mathrm{R}_\mu)$. Note that $G\|_N = G \backslash [\mathrm{V}_N, \mathrm{A}_N, \ell_N] \sqcup R \sqcup \mu\uparrow(\mathrm{R}_\mu) \lhd H$, and also that $\mathrm{R}_{\mu'} = \mathrm{R}_\mu$ hence $\mu'\uparrow(\mathrm{R}_\mu) = (G\|_{M'})^{\uparrow}_{\mu'}$ and (using Lemma 3.4)

$$\begin{aligned}
(G\|_{M'})\|_{\mu'} &= \big(G \backslash [\mathrm{V}_{M'}, \mathrm{A}_{M'}, \ell_{M'}] \sqcup \bigsqcup_{\nu \in M'} G^{\uparrow}_{\nu}\big) \backslash [\mathrm{V}_{\mu'}, \mathrm{A}_{\mu'}, \ell_{\mu'}] \sqcup \mu'\uparrow(\mathrm{R}_\mu) \\
&= G \backslash [\mathrm{V}_N, \mathrm{A}_N, \ell_N] \sqcup \bigsqcup_{\nu \in M'} G^{\uparrow}_{\nu} \backslash [\mathrm{V}_\mu, \mathrm{A}_\mu, \ell_\mu] \sqcup \mu'\uparrow(\mathrm{R}_\mu) \\
&\lhd H'.
\end{aligned}$$

By Lemma 9.5 M has the effective deletion property, i.e., $G\|_M$ is disjoint from V_M, A_M, $\ell_M \backslash \ell^{\uparrow}_M$ hence in particular $G^{\uparrow}_{\nu}$ is disjoint from V_μ, A_μ, $\ell_\mu \backslash \ell^{\uparrow}_M$ for all $\nu \in M'$, so that

$$G^{\uparrow}_{\nu} \backslash [\mathrm{V}_\mu, \mathrm{A}_\mu, \ell_\mu] = G^{\uparrow}_{\nu} \backslash [\mathrm{V}_\mu \backslash \mathrm{V}_\mu, \mathrm{A}_\mu \backslash \mathrm{A}_\mu, \ell_\mu \backslash (\ell_\mu \backslash \ell^{\uparrow}_M)] = G^{\uparrow}_{\nu} \backslash [\varnothing, \varnothing, \ell_\mu \cap \ell^{\uparrow}_M].$$

For all $x \in \dot{G}^{\uparrow}_{\nu} \cup \vec{G}^{\uparrow}_{\nu}$, if $x \notin \dot{G} \cup \vec{G}$ then $\ell_\mu(x) = \varnothing$, otherwise $\mathring{G}^{\uparrow}_{\mu}(x) = \mathring{\mu} \circ \mathring{\mathrm{R}}_\mu \circ \mu^{-1}(x) = \mathring{\mu}' \circ \mathring{\mathrm{R}}_{\mu'} \circ \mu'^{-1}(x)$. Since $\mu(\mathrm{L}_\mu) \sqcap G^{\uparrow}_{\nu} \lhd \mu(\mathrm{K}_\mu) \sqcup G^{\uparrow}_{\mu}$ we have

$$\mathring{G}^{\uparrow}_{\nu}(x) \cap \mathring{\mu} \circ \mathring{\mathrm{L}}_\mu \circ \mu^{-1}(x) \subseteq \mathring{\mu} \circ \mathring{\mathrm{K}}_\mu \circ \mu^{-1}(x) \cup \mathring{G}^{\uparrow}_{\mu}(x)$$

or equivalently $\mathring{G}^{\uparrow}_{\nu}(x) \cap \ell_\mu(x) \subseteq \mathring{G}^{\uparrow}_{\mu}(x)$, and we therefore have

$$\mathring{G}^{\uparrow}_{\nu}(x) \cap \ell_\mu(x) \cap \ell^{\uparrow}_M(x) \subseteq \mathring{\mu}' \circ \mathring{\mathrm{R}}_{\mu'} \circ \mu'^{-1}(x).$$

We thus see that $G^{\uparrow}_{\nu} \backslash [\mathrm{V}_\mu, \mathrm{A}_\mu, \ell_\mu]$ has all the vertices and arrows of $G^{\uparrow}_{\nu}$, and the attributes that are removed are all in the graph $\mu'\uparrow(\mathrm{R}_\mu)$, hence

$$G^{\uparrow}_{\nu} \backslash [\mathrm{V}_\mu, \mathrm{A}_\mu, \ell_\mu] \sqcup \mu'\uparrow(\mathrm{R}_\mu) = G^{\uparrow}_{\nu} \sqcup \mu'\uparrow(\mathrm{R}_\mu)$$

and therefore $(G\|_{M'})\|_{\mu'} = G \backslash [\mathrm{V}_N, \mathrm{A}_N, \ell_N] \sqcup R \sqcup \mu'\uparrow(\mathrm{R}_\mu)$. In order to build an isomorphism $\alpha : H \simeq H'$, let α_μ have $\mu'\uparrow \circ \mu\uparrow^{-1}$ as underlying graph isomorphism (from $\mu\uparrow(\mathrm{R}_\mu)$ to $\mu'\uparrow(\mathrm{R}_\mu)$ underlying graph) and $\mathring{\alpha}_\mu = 1_{\mathscr{A}_G}$. Since

$$\mathring{\mu}' \circ \mathring{\mathrm{R}}_\mu \circ \mu'\uparrow^{-1} \circ \alpha_\mu = \mathring{\mu} \circ \mathring{\mathrm{R}}_\mu \circ \mu\uparrow^{-1} = \mathring{\alpha}_\mu \circ \mathring{\mu} \circ \mathring{\mathrm{R}}_\mu \circ \mu\uparrow^{-1}$$

then $\alpha_\mu : \mu\uparrow(\mathrm{R}_\mu) \simeq \mu'\uparrow(\mathrm{R}_\mu)$.

We now see that $1_{G\sqcup R}$ and α_μ are joinable. For all $y \in (\dot{G} \cup \dot{R}) \cap \mu\uparrow(\dot{\mathrm{R}}_\mu)$, by Corollary 5.3 we have $\dot{R} \cap \mu\uparrow(\dot{\mathrm{R}}_\mu) \subseteq \dot{G}$, hence $y \in \dot{G} \cap \mu\uparrow(\dot{\mathrm{R}}_\mu)$. By Lemma 5.2 there exists $x \in \dot{\mathrm{R}}_\mu \cap \dot{\mathrm{K}}_\mu$ such that $y = \mu(x) = \mu\uparrow(x)$, hence $\alpha_\mu(y) = \mu'\uparrow(x) = \mu'(x) = \mu(x) = y$. The same holds for arrows and trivially for attributes.

Since similarly $\dot{R} \cap \mu'\uparrow(\dot{\mathrm{R}}_\mu) \subseteq \dot{G}$ it is easy to see that $1_{G\sqcup R} \curlywedge \alpha_\mu$ is surjective. By Lemma 3.7 $\alpha = 1_{G\sqcup R} \curlyvee \alpha_\mu : H \simeq H'$. Obviously $\alpha|_{\lfloor G \rfloor} = 1_G$, and

$$\begin{aligned}\alpha(G\|_N) &= \alpha(G\backslash[\mathrm{V}_N, \mathrm{A}_N, \ell_N] \sqcup R \sqcup \mu\uparrow(\mathrm{R}_\mu)) \\ &= G\backslash[\mathrm{V}_N, \mathrm{A}_N, \ell_N] \sqcup R \sqcup \mu'\uparrow(\mathrm{R}_\mu) \\ &= \big(G\|_{M'}\big)\|_{\mu'}.\end{aligned}$$

If part. For all $\mu, \nu \in M$ such that $\mu \neq \nu$, we have

$$\nu(\mathrm{L}_\nu) \lhd G\|_\mu = G\backslash[\mathrm{V}_\mu, \mathrm{A}_\mu, \ell_\mu] \sqcup G^{\uparrow}_\mu.$$

Since $\mu(\mathrm{K}_\mu) \lhd \mu(\mathrm{L}_\mu) \lhd G$, then

$$\begin{aligned}\nu(\mathrm{L}_\nu) \sqcap \mu(\mathrm{L}_\mu) \lhd G\|_\mu \sqcap \mu(\mathrm{L}_\mu) &= \mu(\mathrm{L}_\mu)\backslash[\mathrm{V}_\mu, \mathrm{A}_\mu, \ell_\mu] \sqcup (G^{\uparrow}_\mu \sqcap \mu(\mathrm{L}_\mu)) \\ &= \mu(\mathrm{K}_\mu) \sqcup (G^{\uparrow}_\mu \sqcap \mu(\mathrm{L}_\mu)) \\ &\lhd \mu(\mathrm{K}_\mu) \sqcup G^{\uparrow}_\mu.\end{aligned}$$

Besides, we also have $\mu(\mathrm{L}_\mu) \lhd G\|_\nu$, let $j : \mu(\mathrm{L}_\mu) \to G\|_\nu$ be the canonical injection, $\mu' = j \circ \mu \in \mathscr{M}(\mathrm{r}_\mu, G\|_{M'})$ (hence $\mathrm{V}_{\mu'} = \mathrm{V}_\mu$, $\mathrm{A}_{\mu'} = \mathrm{A}_\mu$ and $\ell_{\mu'} = \ell_\mu$) and $N = \{\mu, \nu\}$, there is an isomorphism α such that $\alpha(G\|_N) = \big(G\|_\nu\big)\|_{\mu'}$ and $\alpha|_{\lfloor G \rfloor} = 1_G$. Let $H = G\|_N \sqcap \mu(\mathrm{L}_\mu)$ and $H' = \big(G\|_\nu\big)\|_{\mu'} \sqcap \mu(\mathrm{L}_\mu)$, since $\mu(\mathrm{L}_\mu) \lhd G$ then $H' = \alpha(G\|_N) \sqcap 1_G(\mu(\mathrm{L}_\mu)) = \alpha(H) = H$. We see that

$$H = \mu(\mathrm{K}_\mu)\backslash[\mathrm{V}_\nu, \mathrm{A}_\nu, \ell_\nu] \sqcup (G^{\uparrow}_\nu \sqcap \mu(\mathrm{L}_\mu)) \sqcup (G^{\uparrow}_\mu \sqcap \mu(\mathrm{L}_\mu))$$

and similarly (using Lemma 3.4) that

$$\begin{aligned}H' &= \mu(\mathrm{K}_\mu)\backslash[\mathrm{V}_\nu, \mathrm{A}_\nu, \ell_\nu] \sqcup (G^{\uparrow}_\nu\backslash[\mathrm{V}_\mu, \mathrm{A}_\mu, \ell_\mu] \sqcap \mu(\mathrm{L}_\mu)) \sqcup (\mu'\uparrow(\mathrm{R}_\mu) \sqcap \mu(\mathrm{L}_\mu)) \\ &= \mu(\mathrm{K}_\mu)\backslash[\mathrm{V}_\nu, \mathrm{A}_\nu, \ell_\nu] \sqcup (G^{\uparrow}_\nu \sqcap \mu(\mathrm{K}_\mu)) \sqcup (\mu'\uparrow(\mathrm{R}_\mu) \sqcap \mu(\mathrm{L}_\mu)).\end{aligned}$$

By Lemma 5.2 we have $\dot{G}^{\uparrow}_\nu \cap \mu(\dot{\mathrm{L}}_\mu) = \nu(\dot{\mathrm{R}}_\nu \cap \dot{\mathrm{K}}_\nu) \cap \mu(\dot{\mathrm{L}}_\mu)$ and $\mu'\uparrow(\dot{\mathrm{R}}_\mu) \cap \mu(\dot{\mathrm{L}}_\mu) = \dot{G}^{\uparrow}_\mu \cap \mu(\dot{\mathrm{L}}_\mu) = \mu(\dot{\mathrm{R}}_\mu \cap \dot{\mathrm{K}}_\mu) \subseteq \mu(\dot{\mathrm{K}}_\mu)$. Hence $\dot{H}'\backslash\mu(\dot{\mathrm{K}}_\mu) = \varnothing$ and $\dot{H}\backslash\mu(\dot{\mathrm{K}}_\mu) = \dot{G}^{\uparrow}_\nu \cap \mu(\dot{\mathrm{L}}_\mu)\backslash\mu(\dot{\mathrm{K}}_\mu)$. Since $\dot{H} = \dot{H}'$ then $\dot{G}^{\uparrow}_\nu \cap \mu(\dot{\mathrm{L}}_\mu) \subseteq \mu(\dot{\mathrm{K}}_\mu)$. Similarly, we get $\vec{G}^{\uparrow}_\nu \cap \mu(\vec{\mathrm{L}}_\mu) \subseteq \mu(\vec{\mathrm{K}}_\mu)$.

For all $x \in \dot{H} \cup \vec{H}$ we have $\mathring{G}^{\uparrow}_\mu(x) = \mathring{\mu} \circ \mathring{\mathrm{R}}_\mu \circ \mu^{-1}(x) = \mathring{\mu}' \circ \mathring{\mathrm{R}}_\mu \circ \mu'^{-1}(x)$, hence obviously $\mathring{H}'(x)\backslash(\mathring{\mu} \circ \mathring{\mathrm{K}}_\mu \circ \mu^{-1}(x) \cup \mathring{G}^{\uparrow}_\mu(x)) = \varnothing$ and

$$\mathring{H}(x)\backslash(\mathring{\mu} \circ \mathring{\mathrm{K}}_\mu \circ \mu^{-1}(x) \cup \mathring{G}^{\uparrow}_\mu(x)) = \mathring{G}^{\uparrow}_\nu(x) \cap \mathring{\mu} \circ \mathring{\mathrm{L}}_\mu \circ \mu^{-1}(x)\backslash(\mathring{\mu} \circ \mathring{\mathrm{K}}_\mu \circ \mu^{-1}(x) \cup \mathring{G}^{\uparrow}_\mu(x)).$$

Since $\mathring{H}(x) = \mathring{H}'(x)$ then $\mathring{G}^{\uparrow}_\nu(x) \cap \mathring{\mu} \circ \mathring{\mathrm{L}}_\mu \circ \mu^{-1}(x) \subseteq \mathring{\mu} \circ \mathring{\mathrm{K}}_\mu \circ \mu^{-1}(x) \cup \mathring{G}^{\uparrow}_\mu(x)$.

We conclude that $G^{\uparrow}_\nu \sqcap \mu(\mathrm{L}_\mu) \lhd \mu(\mathrm{K}_\mu) \sqcup G^{\uparrow}_\mu$. □

Corollary 9.7. *If M is finite and parallel independent then there exists a graph H such that $G \longrightarrow^\star_{\mathcal{R}} H$, $G \Longrightarrow_{\mathcal{R}} G\|_M$ and $H \simeq G\|_M$.*

Proof. By Theorem 9.6 M is sequential independent, hence by Lemma 9.2 there exists H such that $G \longrightarrow^\star_{\mathcal{R}} H$ and $H \simeq G\|_M$. By Lemma 9.5 M has the effective deletion property, hence $G \Longrightarrow_{\mathcal{R}} G\|_M$. □

Hence parallel rewriting can always be applied with parallel independent sets, and then always yields a result reachable by sequential rewriting. This can be interpreted as a result of correction of parallel rewriting w.r.t. sequential rewriting.

10 Related Work and Conclusion

The graphs considered in this paper are structures whose items, namely nodes and arrows, can be assigned with sets of values (attributes). There exist different notions of attributed graphs in the literature. For example, in [16, 17], the authors consider attributed graphs where graph items can hold at most one attribute. This is a particular case of the attributed graphs considered in the present paper.

A notion of conditional rules has been introduced in [16] whose aim is to consider only matchings that satisfy a boolean condition associated with the attributes of a rule. This notion could be included in our framework as a way to select particular matchings. However, such conditions would have to be accounted for in the notion of automorphism group of a rule.

In [18], a notion of attributed graphs has been used to define Dynamic Abstract Data Types. In that paper, an attributed graph is defined as an algebra where graph items can be assigned, as in [16], with zero or one value in a given carrier set.

In [2], the notion of E-graphs is introduced so that several values can be attached to graph items. These values are considered as particular vertices, and are linked to standard graph items by means of dedicated edges. Thus E-graphs can always accomodate enough space for new data, a feature convenient to parallelism (see Section 1). The notion of E-graphs has been extended to symbolic graphs, see e.g.,[19] where E-graphs are endowed with variables constrained by a first-order formula. Our definition of parallel graph rewriting could also be adapted to symbolic graphs, but again it would be more difficult to adapt parallel rewriting modulo automorphism.

Parallel graph rewriting has already been considered in the literature. In the mid-seventies, H. Ehrig and H.-J. Kreowski [11] tackled the problem of parallel graph transformations and introduced the condition of *parallel independence* under which parallel graph transformations could be sequentialized and that of *sequential independence* under which a sequence of graph transformations could be parallelized. This pioneering work has been considered for several algebraic graph transformation approaches, see, e.g., [15, 20, 21] or the more recent contributions [22, 23, 24].

Another notion of non-independent parallelism has been considered in the Double-Pushout approach, see e.g. [25], where rules can be *amalgamated* by agreeing on common deletions, preservations and creations. However, amalgamation is restricted to standard rules and does not impose effective creation or deletion, i.e., amalgamated rules may not yield parallel transformations in the sense of Definition 4.3.

A classical result that is related to non-independence is the Concurrency Theorem, see [2]. This theorem states that a sequence of two rewrite steps with an overlap E between the right-hand side of the first and the left-hand side of the second rule, can be represented as one rewrite step of a new rule, called an *E-concurrent production*. Applying this new rule to a graph does not mean that the second rule can be applied to this graph, whether in parallel or not with the first. The Concurrency Theorem refers to sequential dependence, not to parallel dependence.

Parallelism in graph rewriting has been considered in many other contexts. In [26, chapter 14], parallelism is used to improve the operational semantics of the functional programming language CLEAN [27]. In that contribution, the authors do not deal with true parallelism but rather have an interleaving semantics, hence their parallel rewrite steps can be simulated by sequential ones. This is also the case for other frameworks where massive parallel graph transformations is defined so that it can be simulated by sequential rewriting e.g., [28, 24, 29].

In [30], a framework based on the algebraic Single-Pushout approach has been proposed where parallel transformations only involve matchings provided by a control flow mapping. The users can solve the possible conflicts between the rules by providing the right control flow. More recently, a parallel graph rewrite relation has been defined in [31] for a special kind of graphs called port-graphs. Unfortunately, such graphs are not closed under parallel graph transformation, in the sense that a port-graph can be rewritten in a structure that is not a port-graph. Besides, conditions for avoiding conflicts in parallel transformations have been defined over the considered rewrite rules rather than on the matchings they induce; this limits drastically the shape of these rules.

Graph transformations have also been used to model distributed systems through the Hyperedge Replacement approach, see [32, 33]. The parallel replacement of individual hyperedges by rooted hypergraphs is a natural way of avoiding conflicts since overlaps are restricted to common nodes, but this supposes that at most one rule applies to every hyperedge. In [34] these nodes represent communication channels between hyperedges (representing processes), and a synchronization algebra is used to decide which hyperedges can be replaced simultaneously. By representing cells by hyperedges and neighborhoods by their nodes it is then possible to represent cellular automata on finite configurations (the Game of Life however requires 2^9 rules due to the lack of variables, see [34, Example 5.2]).

An algebraic parallel transformation defined on production rules of the form $L \leftarrow K \leftarrow I \rightarrow R$ has been presented in [35], where a notion of *parallel coherence* ensures that direct transformations of an object G do not conflict, and thus enables a *parallel coherent transformation*. The characterization of parallel independence in this approach has been carried out in [36].

Future work includes applications and implementation issues. The proposed rewrite relations may be used in several contexts such as extensions of L-systems to dynamic graph structures, see [37]. The parallel rewrite relation up to automorphism raises the question of using group-theoretic algorithms for efficiently computing and using generating sets for the automorphism groups of the considered rules, see [14]. The present framework may also be enriched with extra features such as node and edge cloning as proposed in [38, 39].

References

[1] G. Rozenberg (Ed.), Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations, World Scientific, 1997.

[2] H. Ehrig, K. Ehrig, U. Prange, G. Taentzer, Fundamentals of Algebraic Graph Transformation, Monographs in Theoretical Computer Science. An EATCS Series, Springer, 2006.

[3] J. Engelfriet, G. Rozenberg, Node replacement graph grammars, in: Rozenberg [1], pp. 1–94.

[4] R. Echahed, Inductively sequential term-graph rewrite systems, in: ICGT 2008, Vol. 5214 of LNCS, Springer, 2008, pp. 84–98.

[5] R. V. Book, F. Otto, String-Rewriting Systems, Texts and Monographs in Computer Science, Springer, 1993.

[6] F. Baader, T. Nipkow, Term Rewriting and All That, Cambridge University Press, 1998.

[7] D. Plump, Confluence of graph transformation revisited, in: Processes, Terms and Cycles: Steps on the Road to Infinity, Essays Dedicated to Jan Willem Klop, on the Occasion of His 60th Birthday, Vol. 3838 of LNCS, Springer, 2005, pp. 280–308. doi:10.1007/11601548_16.

[8] D. Plump, Checking graph-transformation systems for confluence, ECEASST 26. doi:10.14279/tuj.eceasst.26.367.

[9] D. Plump, From imperative to rule-based graph programs, J. Log. Algebr. Meth. Program. 88 (2017) 154–173.

[10] O. Andrei, M. Fernández, H. Kirchner, G. Melançon, O. Namet, B. Pinaud, PORGY: strategy-driven interactive transformation of graphs, in: 6th International Workshop TERMGRAPH 2011, Vol. 48 of EPTCS, 2011, pp. 54–68.

[11] H. Ehrig, H. Kreowski, Parallelism of manipulations in multidimensional information structures, in: Mathematical Foundations of Computer Science, Vol. 45 of LNCS, Springer, 1976, pp. 284–293.

[12] D. Sannella, A. Tarlecki, Foundations of Algebraic Specification and Formal Software Development, Monographs in Theoretical Computer Science. An EATCS Series, Springer, 2012.

[13] M. Gardner, Mathematical games – the fantastic combinations of John Conway's new solitaire game "life", Scientific American 223 (1970) 120–123.

[14] C. M. Hoffmann, Group-Theoretic Algorithms and Graph Isomorphism, Vol. 136 of Lecture Notes in Computer Science, Springer, 1982.

[15] A. Corradini, U. Montanari, F. Rossi, H. Ehrig, R. Heckel, M. Löwe, Algebraic approaches to graph transformation - part I: basic concepts and double pushout approach, in: G. Rozenberg (Ed.), Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations, World Scientific, 1997, pp. 163–246.

[16] D. Plump, S. Steinert, Towards graph programs for graph algorithms, in: Second International Conference, ICGT 2004, Vol. 3256 of LNCS, 2004, pp. 128–143.

[17] D. Duval, R. Echahed, F. Prost, L. Ribeiro, Transformation of attributed structures with cloning, in: S. Gnesi, A. Rensink (Eds.), 17th International Conference FASE, Vol. 8411 of LNCS, Springer, 2014, pp. 310–324.

[18] H. Ehrig, M. Löwe, F. Orejas, Dynamic abstract data types based on algebraic graph transformations, in: E. Astesiano, G. Reggio, A. Tarlecki (Eds.), COMPASS/ADT, Vol. 906 of LNCS, Springer, 1994, pp. 236–254.

[19] F. Orejas, L. Lambers, Symbolic attributed graphs for attributed graph transformation, ECEASST 30.

[20] H. Ehrig, M. Löwe, Parallel and distributed derivations in the single-pushout approach, Theor. Comput. Sci. 109 (1&2) (1993) 123–143.

[21] H. Ehrig, H.-J. Kreowski, U. Montanari, G. Rozenberg (Eds.), Handbook of Graph Grammars and Computing by Graph Transformations, Volume 3: Concurrency, Parallelism and Distribution, World Scientific, 1999.

[22] A. Corradini, D. Duval, M. Löwe, L. Ribeiro, R. Machado, A. Costa, G. G. Azzi, J. S. Bezerra, L. M. Rodrigues, On the essence of parallel independence for the double-pushout and sesqui-pushout approaches, in: Graph Transformation, Specifications, and Nets - In Memory of Hartmut Ehrig, Vol. 10800 of LNCS, Springer, 2018, pp. 1–18.

[23] M. Löwe, Characterisation of parallel independence in AGREE-rewriting, in: 11th ICGT, Vol. 10887 of LNCS, Springer, 2018, pp. 118–133.

[24] H. Kreowski, S. Kuske, A. Lye, A simple notion of parallel graph transformation and its perspectives, in: Graph Transformation, Specifications, and Nets - In Memory of Hartmut Ehrig, Vol. 10800 of LNCS, Springer, 2018, pp. 61–82.

[25] G. Taentzer, Parallel high-level replacement systems, TCS: Theoretical Computer Science 186 (1997) 43–81.

[26] R. Plasmeijer, M. V. Eekelen, Functional Programming and Parallel Graph Rewriting, 1st Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1993.

[27] S. T. R. Group, The Clean Home Page, Radboud University, Nijmegen.

[28] R. Echahed, J. Janodet, Parallel admissible graph rewriting, in: Recent Trends in Algebraic Development Techniques, 13th International Workshop WADT'98, Selected Papers, Vol. 1589 of LNCS, Springer, 1999, pp. 122–137.

[29] H. Kreowski, S. Kuske, Graph multiset transformation: a new framework for massively parallel computation inspired by DNA computing, Natural Computing 10 (2) (2011) 961–986.

[30] O. Kniemeyer, G. Barczik, R. Hemmerling, W. Kurth, Relational growth grammars - A parallel graph transformation approach with applications in biology and architecture, in: Third International Symposium AGTIVE, Revised Selected and Invited Papers, 2007, pp. 152–167.

[31] R. Echahed, A. Maignan, Parallel graph rewriting with overlapping rules, CoRR (abs/1701.06790).

[32] P. Degano, U. Montanari, A model for distributed systems based on graph rewriting, Journal of the ACM 34 (2) (1987) 411–449.

[33] F. Drewes, H.-J. Kreowski, A. Habel, Hyperedge replacement, graph grammars, in: Rozenberg [1], pp. 95–162.

[34] I. Lanese, U. Montanari, Synchronization algebras with mobility for graph transformations, Electr. Notes Theor. Comput. Sci 138 (1) (2005) 43–60.

[35] T. Boy de la Tour, R. Echahed, Parallel coherent graph transformations, in: Proceedings of WADT 2020, the 25th International Workshop on Algebraic Development Techniques, LNCS, Springer, to appear.

[36] T. Boy de la Tour, Parallelism theorem and derived rules for parallel coherent transformations, CoRR (abs/1907.06585).

[37] S. Wolfram, A new kind of science, Wolfram-Media, 2002.

[38] J. H. Brenas, R. Echahed, M. Strecker, Verifying graph transformation systems with description logics, in: 11th ICGT, Vol. 10887 of LNCS, Springer, 2018, pp. 155–170.

[39] A. Corradini, D. Duval, R. Echahed, F. Prost, L. Ribeiro, AGREE - algebraic graph rewriting with controlled embedding, in: 8th ICGT, Vol. 9151 of LNCS, Springer, 2015, pp. 35–51.