Iterating Non-Aggregative Structure Compositions

Marius Bozga
Univ. Grenoble Alpes, CNRS, Grenoble INP, VERIMAG, 38000, France

Radu Iosif
Univ. Grenoble Alpes, CNRS, Grenoble INP, VERIMAG, 38000, France

Florian Zuleger
Institute of Logic and Computation, Technische Universität Wien, Austria

Abstract

An aggregative composition is a binary operation obeying the principle that the whole is determined by the sum of its parts. The development of graph algebras, on which the theory of formal graph languages is built, relies on aggregative compositions that behave like disjoint union, except for a set of well-marked interface vertices from both sides, that are joined. The same style of composition has been considered in the context of relational structures, that generalize graphs and use constant symbols to label the interface.

In this paper, we study a non-aggregative composition operation, called *fusion*, that joins non-deterministically chosen elements from disjoint structures. The sets of structures obtained by iteratively applying fusion do not always have bounded tree-width, even when starting from a tree-width bounded set. First, we prove that the problem of the existence of a bound on the tree-width of the closure of a given set under fusion is decidable, when the input set is described inductively by a finite *hyperedge-replacement* (HR) grammar, written using the operations of aggregative composition, forgetting and renaming of constants. Such sets are usually called *context-free*. Second, assuming that the closure under fusion of a context-free set has bounded tree-width, we show that it is the language of an effectively constructible HR grammar. A possible application of the latter result is the possiblity of checking whether all structures from a non-aggregatively closed set having bounded tree-width satisfy a given monadic second order logic formula.

2012 ACM Subject Classification Theory of computation → Logic and verification; Theory of computation → Grammars and context-free languages

Keywords and phrases Hyperedge replacement, Tree-width

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2025.18

Related Version *Full Version*: https://arxiv.org/abs/2510.06019

Funding Marius Bozga and Radu Iosif wish to acknowledge the support of the French National Research Agency project Non-Aggregative Resource COmpositions (NARCO) under grant number ANR-21-CE48-0011. Florian Zuleger wishes to acknowledge the support of the FWF project AUTOSARD: "Automated Sublinear Amortised Resource Analysis of Data Structures" No. P36623 and the project VASSAL: "Verification and Analysis for Safety and Security of Applications in Life" funded by the European Union under Horizon Europe WIDERA Coordination and Support Action/Grant Agreement No. 101160022.

1 Introduction

The *tree-width* of a graph is a numerical measure of how "tree-like" the graph is. This notion extends naturally to the relational structures, used to define the semantics of classical first and second-order logic. Relational structures generalize a broad range of graph-like objects, such as edge-labeled graphs, hypergraphs, and multi-edge graphs. Tree-width plays a foundational role in logic and verification. Courcelle's theorem [6] shows that *Monadic Second-order Logic* (MSO) is decidable on classes of structures having bounded tree-width, while Seese's theorem [19] asserts that unbounded tree-width leads to undecidability of MSO theories. Thus, proving that a given class of structures has bounded tree-width is tantamount for establishing the decidability of logical theories for that class.

In principle, one is interested in reasoning about infinite families of structures. These families are typically generated inductively from a finite set of basic building blocks, using operations such as composition and renaming. These operations are usually formalized by the Hyperedge Replacement (HR) algebra introduced by Courcelle [8]. The principle of inductive definition is captured by the notion of a *context-free* set, i.e., the language of a finite grammar written using HR operations, or equivalently, the set of evaluations of a set of ground HR terms recognized by a tree automaton.

For both graphs and relational structures, HR algebras are built on *aggregative composition* operations, in which the whole is determined by the sum of its parts. These compositions are typically defined as the disjoint union of the arguments, where the elements designated by shared constants on both sides are joined together. In other words, aggregative composition preserves the identity of the substructures while merging them at known interface points. Importantly, context-free sets of graphs and relational structures defined by grammars based on such bounded-interface[1] aggregative compositions have bounded tree-width.

In this paper, we investigate a more flexible but less controlled operation, called *non-aggregative fusion*. Fusion allows elements from two disjoint structures to be merged non-deterministically, even when they are not marked by constants. To maintain a certain level of semantic coherence, we require fusion to be constrained by a *coloring* discipline: elements can only be joined if their colors (i.e., sets of designated unary relations) are disjoint. This idea stems from existing work in the area of reasoning about the correctness of systems with dynamically reconfigurable connectivity, such as distributed protocols [1] or pointer structures with aliasing [15].

Due to its nondeterministic nature, fusion can cause a dramatic shift in structure: even if a set **S** consists of structures having bounded tree-width, by taking the closure of **S** under fusion one may introduce infinitely many structures of unbounded tree-width. This phenomenon raises two natural and fundamental questions:

1. Given a context-free set of relational structures, does the closure of this set under fusion have bounded tree-width ?
2. If the answer to the above question is yes, is this closure again a context-free set ?

The result this paper is that both questions have a positive answer (Theorem 9):

1. The existence of a bound on the tree-width of the closure by fusion of a context-free set is a decidable problem.
2. If the fusion-closure of context-free set has bounded tree-width, then it is the language of an effectively constructible context-free grammar, that uses only aggregative composition.

These results provide tools for reasoning about nondeterministic structural iteration. For instance, one can check MSO properties over the tree-width bounded fusion-closure of a context-free set, thereby extending algorithmic verification techniques to a broader class of systems. We sketch below two possible application domains that have motivated our work.

Separation Logic of Relations (SLR). A key motivation for studying non-aggregative fusion comes from SLR, a generalization of classical Separation Logic to relational structures. This logic has been first considered for relational databases and object-oriented languages [14].

[1] Vertex-replacement algebras [8] using disjoint union and edge addition between arbitrarily large interfaces may produce sets of unbounded tree-width.

More recently, SLR (combined with inductive definitions [12]) has been proposed as an assertion language for the verification of distributed reconfigurable systems [1]. Here, the *separating conjunction* $\phi * \psi$ means that the models of two formulæ ϕ and ψ must not have overlapping interpretations of the same relation symbol.

A subtle but crucial point is that, while the separating conjunction enforces *disjointness* of the tuples that interpret a relation symbol, i.e., that tuples cannot overlap in *all* positions, the tuples may overlap in *some* positions. However, by the disjointness of $*$, such overlapping is only possible if the variables at these positions do not occur within the same unary relation symbol [2]. From a semantic point of view, this behavior corresponds precisely to our notion of fusion: joining elements of two separate structures is allowed, as long as their sets of unary relation labels are disjoint. Thus, fusion abstracts the semantics of SLR, where aliasing is controlled implicitly by colors (i.e., sets of relation symbols) and the semantics of the separating conjunction. Moreover, considering inductive definitions on top of the basic SLR logic is akin to considering context-free sets generated by recursive grammars here.

Chemical and Biological Systems. We believe that fusion-like operations naturally arise in the modeling of chemical and biological systems, where complex structures (e.g., proteins or polymeric carbon chains) are formed by joining smaller components through local interactions. Here non-aggregative fusion abstracts the joining of components not only at fixed attachment points, but also through general, property-based interactions, modeled via color compatibility in our framework. Studying the tree-width of these structures is essential for enabling automated reasoning about their properties. We consider the exploration of this area as future work.

Related Work

The notion of composition is central to substructural logics [17]. One of the foremost such logics is Bunched Implications (BI), whose first definition of semantics is based on partially ordered monoids (i.e., the multiplicative connective is interpreted as the multiplication in the monoid) [16]. In particular, the monoidal semantics of BI (and many other follow-up logics) do not assume the composition to be aggregative. The advent of the more popular semantics of BI based on finite partial functions, called *heaps*, has made aggregative composition popular among the users of Separation Logic [13, 18]. We list below several substructural logics where composition is not aggregative.

Docherty and Pym developped Intuitionistic Layered Graph Logic (ILGL), a substructural logic tailored to reasoning about graph structures with a fixed, non-commutative and non-associative notion of layering [9]. Calcagno et al. [2] introduce Context Logic as a framework for local reasoning about structured data, emphasizing compositionality through structural connectives that describe data and context separately rather than flattening them into aggregates. In [3], they formalize these connectives as modal operators and demonstrate that such non-aggregative reasoning is essential for expressing weakest preconditions and verifying updates. Cardelli et al. [4] present a spatial logic for reasoning about graphs that, like our work, emphasizes local, non-aggregative composition via structural connectives such as spatial conjunction.

[2] For each unary relation symbol r in the alphabet, the SLR formula $\mathsf{r}(x) * \mathsf{r}(y)$ entails $x \neq y$.

2 Definitions

Given integers i and j, we write $[i..j]$ for the set $\{i, i+1, \ldots, j\}$, assumed to be empty if $i > j$. For a set A, we denote by $\mathrm{pow}(A)$ its powerset. By writing $B \subseteq_{\mathit{fin}} A$ we mean that B is a finite subset of A. The cardinality of a finite (multi)set A is written $\mathrm{card}(A)$. By writing $A = A_1 \uplus A_2$ we mean that A_1 and A_2 partition A, i.e., $A = A_1 \cup A_2$ and $A_1 \cap A_2 = \emptyset$. The n-times Cartesian product of A with itself is denoted A^n and the set of possibly empty (resp. nonempty) sequences of elements from A by A^* (resp. A^+). Multisets are denoted as $\{\!\{a, b, \ldots\}\!\}$, $\sqcup$ and $\sqcap$ denote the operations of multiset union and intersection, respectively. The multi-powerset (i.e., the set of multisets) of A is written $\mathrm{mpow}(A)$.

2.1 Relational Structures

Let $\mathbb{R}$ be a finite alphabet of *relation symbols* $\mathsf{r} \in \mathbb{R}$, of arities $\#\mathsf{r} \geq 1$, and $\mathbb{C}$ be a countably infinite set of *constants* $\mathsf{c} \in \mathbb{C}$ of arity zero. As usual, relation symbols of arity 1, 2 and 3 are called unary, binary and ternary, respectively. A $\mathcal{C}$*-structure*, for some finite set $\mathcal{C} \subseteq_{\mathit{fin}} \mathbb{C}$ of constants, is a pair $\mathsf{S} = (\mathsf{U}_\mathsf{S}, \sigma_\mathsf{S})$, where U_S is a finite set called *universe* and σ_S is an *interpretation* that maps each relation symbol $\mathsf{r} \in \mathbb{R}$ to a subset of $\mathsf{U}_\mathsf{S}^{\#\mathsf{r}}$ and each constant $\mathsf{c} \in \mathcal{C}$ to an element of U_S. The *sort* of the structure S is the set $\mathcal{C}$. Two structures are *disjoint* iff their universes are disjoint and *isomorphic* iff they are defined over the same alphabet and differ only by a renaming of their elements[3]. For a given sort $\mathcal{C} \subseteq_{\mathit{fin}} \mathbb{C}$, we denote by $\mathcal{S}(\mathcal{C})$ the set of $\mathcal{C}$-structures.

We define the *composition* of two relational structures as the component-wise union of disjoint isomorphic copies of the structures followed by joining the elements that interpret the common constants. This is the same as the *gluing* operation defined by Courcelle [7, Definition 2.1], that we recall below, for self-completeness:

▶ **Definition 1.** *Let $\mathsf{S} \in \mathcal{S}(\mathcal{C})$ be a structure and $\sim \subseteq \mathsf{U}_\mathsf{S} \times \mathsf{U}_\mathsf{S}$ be an equivalence relation, where $[u]_\sim$ denotes the equivalence class of $u \in \mathsf{U}_\mathsf{S}$. The* quotient *of S with respect to $\sim$ is the $\mathcal{C}$-structure $\mathsf{S}_{/\sim}$ defined as follows:*

$$\begin{aligned}
\mathsf{U}_{\mathsf{S}_{/\sim}} &\stackrel{\text{def}}{=} \{[u]_\sim \mid u \in \mathsf{U}_\mathsf{S}\} \\
\sigma_{\mathsf{S}_{/\sim}}(\mathsf{r}) &\stackrel{\text{def}}{=} \{([u_1]_\sim, \ldots, [u_{\#\mathsf{r}}]_\sim) \mid (u_1, \ldots, u_{\#\mathsf{r}}) \in \sigma_\mathsf{S}(\mathsf{r})\}, \text{ for each } \mathsf{r} \in \mathbb{R} \\
\sigma_{\mathsf{S}_{/\sim}}(\mathsf{c}) &\stackrel{\text{def}}{=} [\sigma_\mathsf{S}(\mathsf{c})]_\sim, \text{ for each } \mathsf{c} \in \mathcal{C}
\end{aligned}$$

Let $\mathsf{S}_i = (\mathsf{U}_i, \sigma_i)$ be disjoint $\mathcal{C}_i$-structures, for $i = 1, 2$, and $\approx \subseteq (\mathsf{U}_1 \uplus \mathsf{U}_2) \times (\mathsf{U}_1 \uplus \mathsf{U}_2)$ be the least equivalence relation such that $\sigma_1(\mathsf{c}) \approx \sigma_2(\mathsf{c})$, for all $\mathsf{c} \in \mathcal{C}_1 \cap \mathcal{C}_2$. The composition of S_1 with S_2 is the $\mathcal{C}_1 \cup \mathcal{C}_2$-structure $\mathsf{S}_1 * \mathsf{S}_2 = (\mathsf{U}, \sigma)$, where:

$$\begin{aligned}
\mathsf{U} &\stackrel{\text{def}}{=} \{[u]_\approx \mid u \in \mathsf{U}_1 \uplus \mathsf{U}_2\} \\
\sigma(\mathsf{r}) &\stackrel{\text{def}}{=} \{([u_1]_\approx, \ldots, [u_{\#\mathsf{r}}]_\approx) \mid (u_1, \ldots, u_{\#\mathsf{r}}) \in \sigma_1(\mathsf{r}) \uplus \sigma_2(\mathsf{r})\}, \text{ for each } \mathsf{r} \in \mathbb{R} \\
\sigma(\mathsf{c}) &\stackrel{\text{def}}{=} [\sigma_i(\mathsf{c})]_\approx, \text{ if } \mathsf{c} \in \mathcal{C}_i, \text{ for each } \mathsf{c} \in \mathcal{C}_1 \cup \mathcal{C}_2 \quad (\text{i.e., } [\sigma_1(\mathsf{c})]_\approx = [\sigma_2(\mathsf{c})]_\approx \text{ if } \mathsf{c} \in \mathcal{C}_1 \cap \mathcal{C}_2)
\end{aligned}$$

We remark that the composition of $\emptyset$-structures S_1 and S_2 is the same as their disjoint union, denoted $\mathsf{S}_1 \uplus \mathsf{S}_2$.

---

[3] See, e.g., [10, Section A3] for a formal definition of isomorphism between structures.

The composition of structures is *aggregative*, meaning that it keeps both structures separate except for the interpretation of the common constants. In the following, we define a non-aggregative *fusion* operation that matches also some of the elements which are not interpretations of common constants. In contrast to the deterministic composition, the equivalence relation that matches elements in the fusion operation is chosen nondeterministically.

Before formalizing the notion of non-aggregative fusion, we introduce a generic mechanism for controlling which pairs of elements are allowed to join. We assume a designated set of unary relation symbols $\mathfrak{C} \subseteq \mathbb{R}$. The sets of relation symbols $\gamma \in \mathrm{pow}(\mathfrak{C})$ are called *colors*. Given some $\mathcal{C}$-structure $\mathsf{S} = (\mathsf{U}_\mathsf{S}, \sigma_\mathsf{S})$, we denote by $\mathsf{col}_\mathsf{S}(u) = \{\mathsf{r} \in \mathfrak{C} \mid u \in \sigma_\mathsf{S}(\mathsf{r})\}$ the color of each element $u \in \mathsf{U}_\mathsf{S}$. Note that the empty set is a color.

Back to the definition of non-aggregative fusion, we use colors to prevent joining elements labeled with non-disjoint colors. This is captured by the following notion of compatibility:

▶ **Definition 2.** *Let $\mathsf{S} = (\mathsf{U}, \sigma)$ be a $\mathcal{C}$-structure. A relation $\sim \subseteq \mathsf{U} \times \mathsf{U}$ is* compatible *with* S *if and only if* $\mathsf{col}_\mathsf{S}(u_1) \cap \mathsf{col}_\mathsf{S}(u_2) = \emptyset$*, for each pair $u_1 \sim u_2$.*

We now define the non-aggregative fusion operation. The operation is non-deterministic, i.e., returns a (possibly empty) set of structures. For reasons of simplicity, the fusion operation is only defined for structures of sort $\emptyset$, i.e., for structures that do not interpret any constants. This restriction can be lifted at the expense of complexifying the definition below, by considering fusion in which the interpretation of common constants in both structures must always be joined, as in Definition 1.

Given disjoint sets A and B, a relation $\sim \subseteq A \times B$ is an A-B *matching* iff $\{a, b\} \cap \{a', b'\} = \emptyset$, for all distinct pairs $(a, b), (a', b') \in \sim$. The least equivalence relation that contains $\sim$ is denoted $\equiv_\sim \subseteq (A \uplus B) \times (A \uplus B)$. We say that an equivalence relation $\equiv_\sim$ is k*-generated* iff $\sim$ is a matching consisting of k pairs.

▶ **Definition 3.** *Let $\mathsf{S}_i = (\mathsf{U}_i, \sigma_i)$, for $i = 1, 2$, be two disjoint $\emptyset$-structures. The* fusion *of* S_1 *and* S_2 *is the following set of $\emptyset$-structures:*

$$\mathsf{F}(\mathsf{S}_1, \mathsf{S}_2) \stackrel{\text{def}}{=} \{(\mathsf{S}_1 * \mathsf{S}_2)_{/\equiv_\sim} \mid \sim \text{ non-empty } \mathsf{U}_1\text{-}\mathsf{U}_2 \text{ matching compatible with } \mathsf{S}_1 * \mathsf{S}_2\}$$

Let $\mathbf{S}$ be a set of $\emptyset$-structures. The closure of $\mathbf{S}$ under fusion is the least set $\mathsf{F}^*(\mathbf{S})$ such that $\mathbf{S} \cup \{\mathsf{F}(\mathsf{S}_1, \mathsf{S}_2) \mid \mathsf{S}_1, \mathsf{S}_2 \in \mathsf{F}^*(\mathbf{S})\} \subseteq \mathsf{F}^*(\mathbf{S})$.

Note that $\mathsf{F}(\mathsf{S}_1, \mathsf{S}_2) = \emptyset$ iff $\mathsf{col}_{\mathsf{S}_1}(u_1) \cap \mathsf{col}_{\mathsf{S}_1}(u_2) \neq \emptyset$, for all pairs $(u_1, u_2) \in \mathsf{U}_1 \times \mathsf{U}_2$. The problems considered in the rest of this paper concern the uses of fusion, in addition to the composition and the unary operations on structures introduced next.

2.2 An Algebra of Structures

We recall the definitions of sorted terms and algebras [6, Definition 1.1]. Let Σ be a countably infinite set of *sorts* and let $\mathbb{F}$ be a countably infinite set of *function symbols*, where the set $\mathbb{F}$ is called a *signature*. Each $f \in \mathbb{F}$ has an associated tuple of argument sorts $\alpha(f)$ and a value sort $\rho(f)$. The arity of f, denoted $\#f$, is the length of $\alpha(f)$. Moreover, each variable has a sort. A $\mathbb{F}$-term $t[x_1, \ldots, x_n]$ is built as usual from function symbols and variables $x_1, \ldots, x_n$ of matching sorts. A ground term is a term without variables. A trivial term consists of a single variable. A term t' is a subterm of t iff there exists a term $u[x]$ such that $t = u[t']$, where $u[t']$ denotes the replacement of x by t' in u. The sort of a term t, denoted $\rho(t)$ is the value sort of the top-most symbol, i.e., either the value sort $\rho(f)$ of the top-most function symbol f, in case of a non-trivial term t, or the sort $\rho(x)$ of the variable x, in case of a trivial term x. A position p in a term t is a node of the tree that uniquely represents t, in the usual way (see [5] for a formal definition). $\mathcal{T}(\mathcal{F})$ denotes the set of ground terms having function symbols taken from a finite set $\mathcal{F} \subseteq_{fin} \mathbb{F}$.

An $\mathbb{F}$-*algebra* $\mathcal{A} = (\{\mathsf{A}_s\}_{s\in\Sigma}, \{f^{\mathcal{A}}\}_{f\in\mathbb{F}})$ consists of domains A_s of each sort $s \in \Sigma$ and interprets the function symbols $f \in \mathbb{F}$ as functions $f^{\mathcal{A}} : \mathsf{A}_{s_1} \times \ldots \times \mathsf{A}_{s_n} \to \mathsf{A}_{\rho(f)}$, where $\alpha(f) = (s_1, \ldots, s_n)$. By the domain of $\mathcal{A}$ we understand the set $\mathsf{A} = \bigcup_{s\in\Sigma} \mathsf{A}_s$. We denote by $t^{\mathcal{A}}$ the interpretation of an $\mathbb{F}$-term t in $\mathcal{A}$, i.e., the function obtained by replacing each function symbol that occurs in t by its interpretation. In particular, $t^{\mathcal{A}}$ is an element of the domain of $\mathcal{A}$ if t is ground.

We define an algebra of structures, called $\mathcal{HR}$, with sorts $\Sigma_{\mathcal{HR}} \stackrel{\text{def}}{=} \{\mathcal{C} \mid \mathcal{C} \subseteq_{fin} \mathbb{C}\}$, where each universe $\mathsf{HR}_{\mathcal{C}}$ is the set of $\mathcal{C}$-structures. The signature $\mathbb{F}_{\mathcal{HR}}$ consists of:

- constant symbols $\mathsf{r}(\mathcal{C}_1, \ldots, \mathcal{C}_{\#\mathsf{r}})$, for $\mathsf{r} \in \mathbb{R}$ and $\mathcal{C}_i \subseteq_{fin} \mathbb{C}$ such that either $\mathcal{C}_i = \mathcal{C}_j$ or $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ for all $1 \leq i < j \leq \#\mathsf{r}$, interpreted as $\mathsf{r}(\mathcal{C}_1, \ldots, \mathcal{C}_{\#\mathsf{r}})^{\mathcal{HR}} \stackrel{\text{def}}{=} (\mathsf{U}, \sigma)$, where:

$$\mathsf{U} \stackrel{\text{def}}{=} \{u_1, \ldots, u_{\#\mathsf{r}}\} \text{ for some (possibly equal) elements } u_1, \ldots, u_{\#\mathsf{r}}$$
$$\text{such that } u_i = u_j \iff \mathcal{C}_i = \mathcal{C}_j, \text{ for all } 1 \leq i < j \leq \#\mathsf{r}$$
$$\sigma(\mathsf{r}) \stackrel{\text{def}}{=} \{(u_1, \ldots, u_{\#\mathsf{r}})\} \text{ and } \sigma(\mathsf{r}') \stackrel{\text{def}}{=} \emptyset, \text{ for all } \mathsf{r}' \in \mathbb{R} \setminus \{\mathsf{r}\}$$
$$\sigma(\mathsf{c}) \stackrel{\text{def}}{=} u_i, \text{ for all } \mathsf{c} \in \mathcal{C}_i \text{ and } 1 \leq i \leq \#\mathsf{r}$$

- binary function symbols $\oplus_{\mathcal{C},\mathcal{C}'}$, for $\mathcal{C}, \mathcal{C}' \subseteq_{fin} \mathbb{C}$, interpreted by the composition operation $*$ from Definition 1 applied to structures of sorts $\mathcal{C}$ and $\mathcal{C}'$, respectively, see Figure 1 (a).
- unary function symbols $\mathsf{rename}^{\alpha}_{\mathcal{C}}$, for all $\mathcal{C}, \mathcal{C}' \subseteq_{fin} \mathbb{C}$ and surjective function $\alpha : \mathcal{C} \to \mathcal{C}'$, interpreted as the operations $\mathsf{rename}^{\alpha}_{\mathcal{C}}{}^{\mathcal{HR}} : \mathsf{HR}_{\mathcal{C}} \to \mathsf{HR}_{\mathcal{C}'}$ where, for each $\mathcal{C}$-structure S, the output $\mathsf{S}' = \mathsf{rename}^{\alpha}_{\mathcal{C}}(\mathsf{S})$ is defined below:

$$\mathsf{U}_{\mathsf{S}'} \stackrel{\text{def}}{=} \mathsf{U}_{\mathsf{S}} \quad \sigma_{\mathsf{S}'}(\mathsf{r}) \stackrel{\text{def}}{=} \sigma_{\mathsf{S}}(\mathsf{r}) \quad \sigma_{\mathsf{S}'}(\alpha(\mathsf{c})) \stackrel{\text{def}}{=} \sigma_{\mathsf{S}}(\mathsf{c}), \text{ for all } \mathsf{r} \in \mathbb{R} \text{ and } \mathsf{c} \in \mathcal{C} \setminus \mathcal{C}'$$

- unary function symbols $\mathsf{forget}^{\mathcal{C}'}_{\mathcal{C}}$, for $\mathcal{C} \subseteq_{fin} \mathbb{C}$ and $\mathcal{C}' \subseteq \mathcal{C}$, interpreted as the operations $\mathsf{forget}^{\mathcal{C}'}_{\mathcal{C}}{}^{\mathcal{HR}} : \mathsf{HR}_{\mathcal{C}} \to \mathsf{HR}_{\mathcal{C}\setminus\mathcal{C}'}$ where, for each $\mathcal{C}$-structure S, the output $\mathsf{S}' = \mathsf{forget}^{\mathcal{C}'}_{\mathcal{C}}{}^{\mathcal{HR}}(\mathsf{S})$ is defined below:

$$\mathsf{U}_{\mathsf{S}'} \stackrel{\text{def}}{=} \mathsf{U}_{\mathsf{S}} \quad \sigma_{\mathsf{S}'}(\mathsf{r}) \stackrel{\text{def}}{=} \sigma_{\mathsf{S}}(\mathsf{r}) \quad \sigma_{\mathsf{S}'}(\mathsf{c}) \stackrel{\text{def}}{=} \sigma_{\mathsf{S}}(\mathsf{c}), \text{ for all } \mathsf{r} \in \mathbb{R} \text{ and } \mathsf{c} \in \mathcal{C}$$

To ease the notation, we omit the sorts of the arguments from the $\mathcal{HR}$ function symbols $\mathbb{F}_{\mathcal{HR}}$ when they are understood from the context.

▶ **Example 4.** The two leftmost graphs in Figure 1 (a) are the values of the following terms, respectively (singleton sets are denoted by their elements, to avoid clutter):

$$t_1 \stackrel{\text{def}}{=} \mathsf{rename}^{\mathsf{c}_3\to\mathsf{c}_1}\left(\mathsf{forget}^{\mathsf{c}_1}\left(a(\mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2) \oplus c(\mathsf{c}_1, \mathsf{c}_3)\right) \oplus b(\mathsf{c}_0, \mathsf{c}_3)\right)$$
$$t_2 \stackrel{\text{def}}{=} b(\mathsf{c}_0, \mathsf{c}_1) \oplus \mathsf{forget}^{\mathsf{c}_2}\left(a(\mathsf{c}_1, \mathsf{c}_3, \mathsf{c}_2)\right)$$

The graph in Figure 1 (c) is the value of the term $t_0 \stackrel{\text{def}}{=} \mathsf{rename}^{\mathsf{c}_0\leftrightarrow\mathsf{c}_1}\left(\mathsf{forget}^{\mathsf{c}_2}(t_1 \oplus t_2)\right)$.

We comment on the relationship to the algebra of relational structures introduced in [7, Definition 2.3], as a generalization of both the hyperedge-replacement (HR) algebra of hypergraphs and the vertex-replacement (VR) algebra of binary graphs, to relational structures. As a matter of fact, we do not use this algebra. Instead, our algebra of relational structures algebra $\mathcal{HR}$ follows the standard HR algebra on hypergraphs [6]. This relationship can be easily understood by viewing relational structures as hypergraphs in the

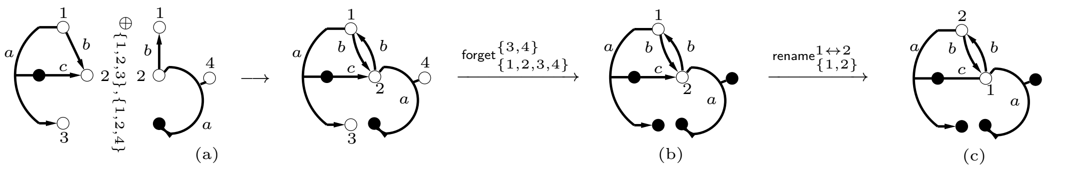

Figure 1 HR operations on structures over the alphabet $\{a, b, c\}$, where $\#a = 3$ and $\#b = \#c = 2$. The order of vertices attached to an edge is indicated by an arrow pointing to the last vertex.

adjacency encoding, i.e., each tuple $(u_1, \ldots, u_{\#\mathsf{r}})$ from the interpretation of a relation symbol r corresponds to a r-labeled hyperedge attached to the vertices $u_1, \ldots, u_{\#\mathsf{r}}$ in this order. For reasons of space, we omit further details.

We now introduce the subalgebras of $\mathcal{HR}$ that define the tree-width parameter of a structure. For this, we assume some enumeration of the constants $\mathbb{C} = \{\mathsf{c}_0, \mathsf{c}_1, \ldots\}$ and denote by $\mathbb{F}_{\mathcal{HR}^{\leq k}}$ the subset of $\mathbb{F}_{\mathcal{HR}}$ consisting of the function symbols whose argument and value sorts are all contained in $\{\mathsf{c}_0, \ldots, \mathsf{c}_k\}$.

▶ **Definition 5.** *The* tree-width *of a structure* S*, denoted* $\mathrm{tw}(\mathsf{S})$*, is the minimal integer* $k \geq 0$ *for which there exists a ground term* $t \in \mathcal{T}(\mathbb{F}_{\mathcal{HR}^{\leq k}})$ *such that* $t^{\mathcal{HR}} = \mathsf{S}$*. A set* $\mathbf{S}$ *of structures has* bounded tree-width *if and only if the set* $\{\mathrm{tw}(\mathsf{S}) \mid \mathsf{S} \in \mathbf{S}\}$ *is finite.*

In particular, for each tree-width bounded set $\mathbf{S}$, there exists a set $\mathcal{T}$ of ground terms and a finite set $\mathcal{C} \subseteq_{\mathit{fin}} \mathbb{C}$ of constants such that each term $t \in \mathcal{T}$ uses only constants from $\mathcal{C}$ and $\mathbf{S} = \{t^{\mathcal{HR}} \mid t \in \mathcal{T}\}$.

This algebraic definition of tree-width of a relational structure is analogous to the definition of the tree-width of a hypergraph using a subalgebra of HR defined by a restriction of sorts to finite sets of vertex labels (see, e.g., [8, Proposition 1.19] for a proof of equivalence between the graph-theoretic and algebraic definitions of tree-width for hypergraphs). Moreover, the tree-width of a structure can be equivalently defined in terms of the tree-width of its Gaifman-graph:

▶ **Definition 6.** *Let* S *be a* $\mathcal{C}$*-structure. The* Gaifman graph *of* S *is the simple undirected graph* $\mathsf{Gaif}(\mathsf{S}) = (V, E)$*, where* $V \stackrel{\text{def}}{=} \mathsf{U}_\mathsf{S}$ *and* $E \stackrel{\text{def}}{=} \{\{u_i, u_j\} \mid (u_1, \ldots, u_{\#\mathsf{r}}) \in \sigma_\mathsf{S}(\mathsf{r}), 1 \leq i \neq j \leq \#\mathsf{r}, \mathsf{r} \in \mathbb{R}\}$.

It is known that the tree-width of a structure equals the tree-width of its Gaifman graph [11, Proposition 11.27].

We recall below two standard notions of graph theory. Given binary graphs G and H, we say that H is a *minor* of G iff H is obtained from a subgraph of G by *edge contractions*, where the contraction of a binary edge $e \in E_G$ attached to vertices u and v means deleting e and joining u and v into a single vertex x (the edges attached to x are the ones attached to either u or v). It is well-known that the tree-width of each minor of a graph G is bounded by the tree-width of G.

A $n{\times}m$*-grid* is a binary graph whose vertices can be labeled with pairs $(i, j) \in [1..n] \times [1..m]$ such that there is an edge between (i, j) and (i', j') iff either $i < n$, $i' = i + 1$ and $j' = j$ or $j < m$, $i' = i$ and $j' = j + 1$. It is well-known[4] that each $n \times n$-grid has tree-width n. By bluring the distinction between isomorphic grids, we obtain the following:

[4] This can be shown by using the characterisation of tree-width in terms of the cops and robber game [20].

▶ **Proposition 7.** *A set of structures whose Gaifman graphs contain infinitely many non-isomorphic square grids has unbounded tree-width.*

2.3 Context-Free Sets of Structures

Context-free sets are usually defined as languages of grammars, i.e., finite sets of inductive rules written using nonterminals and function symbols from a given signature. To simplify some of the upcoming proofs, we use here an equivalent definition of contex-free sets based on recognisable sets of ground terms, defined using tree automata [5]. For self-containment reasons, we briefly introduce context-free grammars and discuss their equivalence with tree automata in Section 3.5 (see the statement of Theorem 24).

Let $\mathcal{F} \subseteq_{fin} \mathbb{F}$ be a finite signature of function symbols. A *tree automaton* over $\mathcal{F}$ is a tuple $\mathcal{A} = (Q, F, \rightarrow)$, where Q is a finite set of states, $F \subseteq Q$ is a set of accepting states and $\rightarrow$ is a set of transition rules of the form $(q_1, \ldots, q_{\#f}) \xrightarrow{f} q$, where $q_1, \ldots, q_{\#f}, q \in Q$ and $f \in \mathcal{F}$. A run π of $\mathcal{A}$ over a ground term $t \in \mathcal{T}(\mathcal{F})$ maps each position p within t to a state $q = \pi(p)$ if the automaton has a rule $(\pi(p_1), \ldots, \pi(p_{\#f})) \xrightarrow{f} q$, where $p_1, \ldots, p_{\#f}$ are the positions of the children of p in t. A ground term t is accepted by $\mathcal{A}$ iff $\mathcal{A}$ has a run that labels the root of t with an accepting state. The language of $\mathcal{A}$, denoted $\mathcal{L}(\mathcal{A})$, is the set of ground terms accepted by $\mathcal{A}$. A set $T \subseteq \mathcal{T}(\mathcal{F})$ is *recognisable* iff it is the language of a tree automaton over the finite signature $\mathcal{F}$.

▶ **Definition 8.** *A set* $\mathbf{S}$ *of structures is* context-free *if and only if there exists a recognisable set of ground terms* $T \subseteq \mathcal{T}(\mathcal{F})$*, over a finite signature* $\mathcal{F} \subseteq \mathbb{F}_{\mathcal{HR}}$*, such that* $\mathbf{S} = \{t^{\mathcal{HR}} \mid t \in T\}$.

Note that a context-free set of structures has finitely many sorts, because the signature of terms used to describe the set is finite. For this reason, any context-free set of structures has bounded tree-width. On the other hand, there are bounded tree-width sets which are not context-free, for instance the set of linear structures over the alphabet $\{a, b, c\}$ of binary relations of the form $a^n b^n c^n$, for all $n \geq 1$.

3 The Closure of Context-Free Sets under Fusion

This section is concerned with the statement and proof of the main result of the paper (Theorem 9). For simplicity, we assume that $\mathbf{S}$ is a set of *connected* structures, where connectivity of a structure S means that there exists an undirected path in $\mathsf{Gaif}(\mathsf{S})$ between each pair of elements $u_1, u_2 \in \mathsf{U}_\mathsf{S}$. We note that, because $\mathbf{S}$ contains only connected structures, each structure from its closure $\mathsf{F}^*(\mathbf{S})$ is necessarily connected.

The assumption of $\mathbf{S}$ being a context-free set of connected structures loses no generality, because it is possible, from a tree automaton $\mathcal{A}$ such that $\mathbf{S} = \mathcal{L}(A)^{\mathcal{HR}}$, to build a tree automaton $\mathcal{B}$ such that $\mathcal{L}(B)^{\mathcal{HR}}$ is the set of connected substructures of a structure in $\mathbf{S}$. Intuitively, $\mathcal{B}$ is obtained from $\mathcal{A}$ be labeling each state q with finite information concerning the existence of a path between each pair of constant symbols in each structure that is the value of a ground term recognized by q in $\mathcal{A}$. This construction can be used to generalize the statement of Theorem 9 below from connected to arbitrary structures. We will detail this construction in an extended version of the present article.

▶ **Theorem 9.** *Let* $\mathbf{S}$ *be a context-free set of connected* $\emptyset$*-structures.*

1. $\mathsf{F}^*(\mathbf{S})$ *has bounded tree-width if and only if* $\mathsf{F}^*(\mathbf{S})$ *is context-free.*
2. *It is decidable whether* $\mathsf{F}^*(\mathbf{S})$ *has bounded tree-width.*

An obvious consequence of this theorem is the decidability of the problem: *given a context-free set* $\mathbf{S}$*, is* $\mathsf{F}^*(\mathbf{S})$ *context-free?*

We give an overview of the proof before going into technical details. The core idea is the equivalence between the (A) tree-width boundedness of the closure $\mathsf{F}^*(\mathbf{S})$ of a context-free set $\mathbf{S}$ and (B) the non-existence of two structures $\mathsf{S}_1, \mathsf{S}_2 \in \mathsf{F}^*(\mathbf{S})$ having each at least three elements each $u_i, v_i, w_i \in \mathsf{U}_{\mathsf{S}_i}$, labeled with disjoint colors γ_i, for $i = 1, 2$. This equivalence is established via a third, more technical, condition about the disjointness relations between the colors that may occur in a structure from $\mathsf{F}^*(\mathbf{S})$. The latter implies that the matching relation of each fusion of two structures from $\mathsf{F}^*(\mathbf{S})$ is generated by at most two pairs of elements with compatible colors.

The equivalence between (A) and (B) is used to prove both points of Theorem 9. For point (1) suppose, for a contradiction, that (B) does not hold. Then, S_1 and S_2 can be composed by joining their u_i, v_i or w_i elements, for $i = 1, 2$, respectively, as in Figure 2. Consequently, the set of Gaifman graphs corresponding to the structures in $\mathsf{F}^*(\mathbf{S})$ contains an infinite set of grid minors, thus $\mathsf{F}^*(\mathbf{S})$ has unbounded tree-width (Proposition 7). Else, if (B) holds (i.e., such structures cannot be found), we prove that the matching relation considered in the fusion of any two structures is generated by either one or two pairs of elements. In each of these cases, by adding a finite number of constants to the signature of the $\mathbb{F}_{\mathcal{HR}}$-terms from the language of $\mathcal{A}$, we can build a tree automaton $\mathcal{A}^*$ such that $\mathcal{L}(\mathcal{A}^*)^{\mathcal{HR}} = \mathsf{F}^*(\mathbf{S})$, thus taking care of point (1) of the theorem.

To prove point (2), we rely on the equivalence of (A) and (B) and show that (B) is decidable. This is done by arguing that the existence of two structures with the above property is equivalent to the existence of two multiset abstractions of structures $\{\!\!\{\gamma_1, \gamma_1, \gamma_1\}\!\!\}$, $\{\!\!\{\gamma_2, \gamma_2, \gamma_2\}\!\!\}$ in the domain of multisets of colors having multiplicity at most three. These multiset abstractions of colors can be effectively computed by a finite fixpoint iteration over the rules of the tree automaton that recognizes the set of ground terms which evaluates to the elements of $\mathbf{S}$.

3.1 Color Multisets

In the following, let $\mathbf{S}$ be a context-free set of $\emptyset$-structures. We denote the set of colors by $\Gamma \stackrel{\text{def}}{=} \mathrm{pow}(\mathfrak{C})$, where $\mathfrak{C}$ is a fixed finite set of unary relation symbols. First, we define an abstraction of structures as finite multisets of colors:

▶ **Definition 10.** *The* multiset color abstraction $\mathsf{S}^\sharp \in \mathrm{mpow}(\Gamma)$ *of a structure* S *is* $\mathsf{S}^\sharp \stackrel{\text{def}}{=} \{\!\!\{\mathsf{cols}_\mathsf{S}(u) \mid u \in \mathsf{U}_\mathsf{S}\}\!\!\}$. *For an integer* $k \geq 0$, *the* k-multiset color abstraction $\mathsf{S}^{\sharp k} \subseteq \mathrm{mpow}(\Gamma)$ *is* $\mathsf{S}^{\sharp k} \stackrel{\text{def}}{=} \{M \subseteq \mathsf{S}^\sharp \mid \mathrm{card}(M) \leq k\}$. *These abstractions are lifted to sets of structures as sets of multisets* $\mathbf{S}^\sharp \stackrel{\text{def}}{=} \{\mathsf{S}^\sharp \mid \mathsf{S} \in \mathbf{S}\}$ *and* $\mathbf{S}^{\sharp k} \stackrel{\text{def}}{=} \bigcup_{\mathsf{S} \in \mathbf{S}} \mathsf{S}^{\sharp k}$.

Note that $\mathsf{S}^\sharp$ is a multiset, whereas $\mathsf{S}^{\sharp k}$ is a set of multisets. When lifted to sets of structures, both $\mathbf{S}^\sharp$ and $\mathbf{S}^{\sharp k}$ are sets of multisets.

The core of the proof of Theorem 9 is the equivalence between the following two conditions stated formally below:

$$\mathrm{tw}(\mathsf{F}^*(\mathbf{S})) \leq k, \text{ for some } k \geq 1 \tag{A}$$

$$\{\!\!\{\gamma_1, \gamma_1, \gamma_1\}\!\!\}, \{\!\!\{\gamma_2, \gamma_2, \gamma_2\}\!\!\} \in (\mathsf{F}^*(\mathbf{S}))^{\sharp 3} \Rightarrow \gamma_1 \cap \gamma_2 \neq \emptyset, \text{ for all } \gamma_1, \gamma_2 \in \Gamma \tag{B}$$

Note that condition (A) is more general than the premiss of Theorem 9. We prove the (A) $\Rightarrow$ (B) direction below.

▶ **Lemma 11.** *If* $\mathbf{S}$ *has bounded tree-width, then (A) implies (B).*

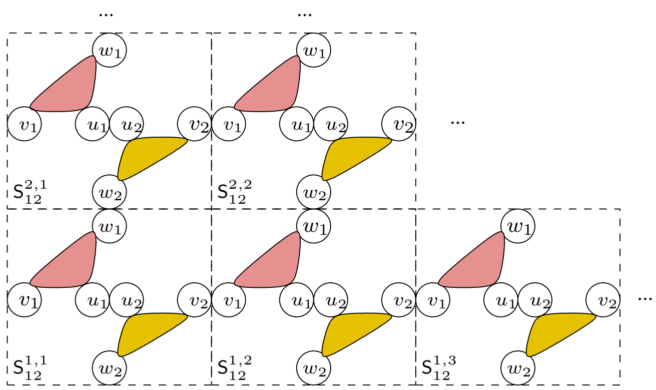

Figure 2 Building structures whose Gaifman graphs have infinitely large grid minors.

Proof. By contradiction, assume that there exist $\{\!\{\gamma_1, \gamma_1, \gamma_1\}\!\}, \{\!\{\gamma_2, \gamma_2, \gamma_2\}\!\} \in (\mathsf{F}^*(\mathbf{S}))^{\sharp 3}$ such that $\gamma_1 \cap \gamma_2 = \emptyset$. Then, there exist structures $\mathsf{S}_1, \mathsf{S}_2 \in \mathsf{F}^*(\mathbf{S})$ such that $\{\!\{\gamma_1, \gamma_1, \gamma_1\}\!\} \in \mathsf{S}_1^\sharp$ and $\{\!\{\gamma_2, \gamma_2, \gamma_2\}\!\} \in \mathsf{S}_2^\sharp$. We shall use S_1 and S_2 to build infinitely many structures whose Gaifman graphs have arbitrarily large square grid minors, as illustrated in Figure 2.

First, construct the structure $\mathsf{S}_{12} \in \mathsf{F}^*(\mathbf{S})$ by fusing one pair (u_1, u_2), having colors γ_1 and γ_2, respectively. Let v_1 and w_1 (resp. v_2 and w_2) be the remaining distinct elements of S_{12}, having color γ_1 (resp. γ_2) from S_1 and S_2, respectively. For an arbitrarily large integer $n \geq 1$, consider $n \times n$ disjoint copies $(\mathsf{S}_{12}^{i,j})_{i,j=1,n}$ of S_{12}. Let $\approx^{1,j}$ be the equivalence relation generated by $\{(v_1^{1,j}, v_2^{1,j-1})\}$ and $\approx^{i,1}$ be generated by $\{(w_2^{i,1}, w_1^{i-1,1})\}$, $\approx^{i,j}$ be generated by $\{(v_1^{i,j}, v_2^{i,j-1}), (w_2^{i,j}, w_1^{i-1,j})\}$, for all $i, j = 2, n$. Second, construct the grid-like connected structure $X^{n,n} \in \mathsf{F}^*(\mathbf{S})$:

$$X^{n,n} = (\ldots(\ldots((\mathsf{S}_{12}^{1,1} * \mathsf{S}_{12}^{1,2})_{/\approx^{1,2}} * \mathsf{S}_{12}^{2,1})_{/\approx^{2,1}} * \ldots * \mathsf{S}_{12}^{i,j})_{/\approx^{i,j}} * \ldots * \mathsf{S}_{12}^{n,n})_{/\approx^{n,n}}$$

where structures $\mathsf{S}_{12}^{i,j}$ are added to the fusion in the increasing order of $i + j$. We can show that $\mathsf{Gaif}(X^{n,n})$ has an $n \times n$ square grid minor. Finally, as n can be taken arbitrarily large, we conclude that $\mathsf{F}^*(\mathbf{S})$ does not have bounded tree-width, which contradicts (A). ◀

3.2 Color Schemes

For the proof of the (A) "⇐" (B) direction, we first organize the set of colors using the *RGB color schemes* defined below:

▶ **Definition 12.** *A partition* $(\Gamma^{red}, \Gamma^{green}, \Gamma^{blue})$ *of* Γ *is an* RGB color scheme *if and only if:*

1. $\gamma_1 \cap \gamma_2 \neq \emptyset$*, for all* $\gamma_1, \gamma_2 \in \Gamma^{blue}$*,*
2. $\gamma_1 \cap \gamma_2 \neq \emptyset$*, for all* $\gamma_1 \in \Gamma^{green}$ *and all* $\gamma_2 \in \Gamma^{blue}$*,*
3. *for all* $\gamma_1 \in \Gamma^{red}$ *there exists* $\gamma_2 \in \Gamma^{blue}$ *such that* $\gamma_1 \cap \gamma_2 = \emptyset$*.*

Note that an RGB color scheme is fully specified by the set Γ^{blue}. Indeed, any color not in Γ^{blue} is unambiguously placed within Γ^{red} or Γ^{green}, depending on whether or not it is disjoint from some color in Γ^{blue}. In particular, if $\Gamma^{blue} = \emptyset$ then $\Gamma^{red} = \emptyset$ and $\Gamma^{green} = \Gamma$. For example, Figure 3 shows several RGB color schemes for the set $\mathfrak{C} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ of unary relation symbols.

Because a fusion operation only joins element with disjoint colors, blue elements can only be joined with red elements, green elements can be joined with green or red elements, whereas red elements can be joined with elements of any other color. We define below what is meant for a set of structures to conform to an RGB color scheme:

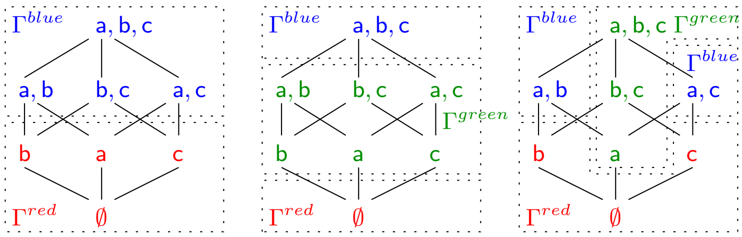

Figure 3 Examples of RGB color schemes.

Table 1 The types of structures obtained by fusion, where $\emptyset$ means that the result of the fusion is the empty set.

$\mathsf{F}(\mathsf{S}_1, \mathsf{S}_2)$	S_2 of R type	S_2 of G type	S_2 of B type
S_1 of R type	R, G, B	G, B	B
S_1 of G type	G, B	G, B	$\emptyset$
S_1 of B type	B	$\emptyset$	$\emptyset$

▶ **Definition 13.** *A set* $\mathbf{S}$ *of structures* conforms to $(\Gamma^{red}, \Gamma^{green}, \Gamma^{blue})$ *if and only if:*

1. *for all structures* $\mathsf{S} \in \mathbf{S}$*, if* $\mathsf{cols}_{\mathsf{S}}(u) \in \Gamma^{red}$*, for some element* $u \in \mathsf{U}_{\mathsf{S}}$*, then* $\mathsf{cols}_{\mathsf{S}}(u') \in \Gamma^{blue}$*, for all other elements* $u' \in \mathsf{U}_{\mathsf{S}} \setminus \{u\}$*, and*
2. $\mathsf{S}^\sharp \cap \Gamma^{green} \subseteq \{\!\{\gamma, \gamma \mid \gamma \in \Gamma^{green}\}\!\}$*, for all structures* $\mathsf{S} \in \mathsf{F}^*(\mathbf{S})$*.*

In other words, $\mathbf{S}$ conforms to a given color scheme if each structure from $\mathbf{S}$ has either a single red and the rest blue, or at most occurrences of the same green color and the rest blue elements. Moreover, the number of occurrences of a green color must not exceed two, for each structure obtained by taking fusions of some structures in $\mathbf{S}$. This observation justifies the following notion of *type* of a structure:

▶ **Definition 14.** *A structure* S *is of* type R *if it has exactly one red element and the rest blue,* G *if it has at least one green element and the rest blue and* B *if it has only blue elements.*

Note that there can be structures of neither R, G or B type, but these are the only types of interest, as justified by the following:

▶ **Lemma 15.** *Let* $\mathbf{S}$ *be a set of structures conforming to an RGB color scheme. Then, each structure* $\mathsf{S} \in \mathsf{F}^*(\mathbf{S})$ *is of type either* R, G *or* B*.*

Proof. By induction on the construction of $\mathsf{S} \in \mathsf{F}^*(\mathbf{S})$ from one or more structures from $\mathbf{S}$. Table 1 summarizes the possible types of $\mathsf{F}(\mathsf{S}_1, \mathsf{S}_2)$ on structures S_1 and S_2 of types R, G or B, respectively. ◀

The (B) "⇒" (A) direction will be established via a third condition (C), which is conformance to an RGB color scheme defined by taking the Γ^{blue} set to be the colors occurring three times in some structure from $\mathsf{F}^*(\mathbf{S})$:

▶ **Lemma 16.** *If (B) holds then:*

$$\mathbf{S} \text{ conforms to } (\Gamma^{red}, \Gamma^{green}, \Gamma^{blue}), \text{ where } \Gamma^{blue} \stackrel{\text{def}}{=} \{\gamma \in \Gamma \mid \{\!\{\gamma, \gamma, \gamma\}\!\} \in (\mathsf{F}^*(\mathbf{S}))^{\sharp 3}\} \qquad \text{(C)}$$

Proof. We show that $\mathbf{S}$ conforms to the $(\Gamma^{red}, \Gamma^{green}, \Gamma^{blue})$ RGB color scheme from the statement, by checking the two points of Definition 13:

(1) Let $\mathsf{S} \in \mathbf{S}$ and prove that for any two colors $\gamma_1, \gamma_2 \in \mathfrak{C}$, if $\{\!\{\gamma_1, \gamma_2\}\!\} \subseteq \mathsf{S}^\sharp$ and $\gamma_1 \in \Gamma^{red}$ then $\gamma_2 \in \Gamma^{blue}$. Since $\gamma_1 \in \Gamma^{red}$, there must exists a color $\gamma_1' \in \Gamma^{blue}$, such that $\gamma_1 \cap \gamma_1' = \emptyset$, by Definition 12. By the definition of Γ^{blue}, this further implies $\{\!\{\gamma_1', \gamma_1', \gamma_1'\}\!\} \in (\mathsf{F}^*(\mathbf{S}))^{\sharp 3}$. Henceforth, there exists a structure $\mathsf{S}' \in \mathsf{F}^*(\mathbf{S})$ such that $\{\!\{\gamma_1', \gamma_1', \gamma_1'\}\!\} \subseteq \mathsf{S}'^\sharp$. We can now use S' and three disjoint copies of S to build a new structure S'' by gluing progressively, each one of the three elements of color γ_1' in S' to the element of color γ_1 of S. Then, by construction, the structure S'' will also contain three elements of color γ_2, one from each disjoint copy of S. Therefore, $\{\!\{\gamma_2, \gamma_2, \gamma_2\}\!\} \in \mathsf{S}''^\sharp$ and because $\mathsf{S}'' \in \mathsf{F}^*(\mathbf{S})$ this implies $\{\!\{\gamma_2, \gamma_2, \gamma_2\}\!\} \in (\mathsf{F}^*(\mathbf{S}))^{\sharp 3}$ and therefore $\gamma_2 \in \Gamma^{blue}$.

(2) By contradiction, let $\mathsf{S} \in \mathsf{F}^*(\mathbf{S})$ be such that $\mathsf{S}^\sharp \sqcap \Gamma^{green} \not\subseteq \{\!\{\gamma, \gamma \mid \gamma \in \Gamma^{green}\}\!\}$. Then there exists $\gamma' \in (\mathsf{S}^\sharp \sqcap \Gamma^{green}) \setminus \{\!\{\gamma, \gamma \mid \gamma \in \Gamma^{green}\}\!\}$, i.e., $\gamma' \in \Gamma^{green}$ and $\{\!\{\gamma', \gamma', \gamma'\}\!\} \subseteq \mathsf{S}^\sharp$. The latter implies $\{\!\{\gamma', \gamma', \gamma'\}\!\} \in \mathsf{S}^{\sharp 3} \subseteq (\mathsf{F}^*(\mathbf{S}))^{\sharp 3}$. But this implies $\gamma' \in \Gamma^{blue}$ according to the definition of the RGB color scheme, contradicting $\gamma' \in \Gamma^{green}$. ◀

In the rest of this and the next subsections, we are concerned with the proof of the following implication, that establishes the equivalence of (A) and (B). As previously mentioned, this direction of the proof uses the third condition (C) that is, conformance to the RGB color scheme from the statement of Lemma 16:

▶ **Lemma 17.** *If $\mathbf{S}$ has bounded tree-width and (C) holds then (A) holds.*

The proof of the above lemma is split into two technical results (Lemmas 18 and 19). The first (Lemma 18) involves reasoning about the number of pairs of elements that are joined by the fusion operation in order to obtain a structure from $\mathsf{F}^*(\mathbf{S})$:

▶ **Lemma 18.** *If $\mathbf{S}$ conforms to an RGB scheme $(\Gamma^{red}, \Gamma^{green}, \Gamma^{blue})$ and $\mathsf{S} = (\mathsf{S}_1 \uplus \mathsf{S}_2)_{/\approx}$ for some $\mathsf{S}_i = (\mathsf{U}_i, \sigma_i) \in \mathsf{F}^*(\mathbf{S})$, for $i = 1, 2$, and some equivalence closure $\approx$ of a U_1-U_2 matching, then exactly one of the following holds:*

1. *$\approx$ is 1-generated, or*

2. *$\approx$ is 2-generated, S is of type B, and either:*
   - **a.** *S_1, S_2 are both of type R, or*
   - **b.** *S_1, S_2 are both of type G and $\mathrm{card}({\mathsf{S}_1}^\sharp \sqcap \Gamma^{green}) = \mathrm{card}({\mathsf{S}_2}^\sharp \sqcap \Gamma^{green}) = 2$.*

Proof. We distinguish two cases:

- S_1 is of type R: If S_2 is of type B or G then S_1 and S_2 can be fused only by equivalences $\approx$ generated by a single pair, that contains the element from the support of S_1 with color in Γ^{red}, thus matching the case (1) from the statement. Else, if S_2 is of type R then S_1 and S_2 can be fused by equivalences generated by at most two pairs, each containing an element with color from Γ^{red}, from either S_1 or S_2, thus matching the case 2a from the statement. In this latter case, S is of type B because joining a red with a blue element always results in a blue element.
- S_1, S_2 are both of type G: By contradiction, assume they can be fused by an equivalence $\approx$ generated by three pairs of elements $(u_{1i}, u_{2i})_{i=1,2,3}$. Let $G_{1i} = \mathsf{col}_{\mathsf{S}_1}(u_{1i})$, $G_{2i} = \mathsf{col}_{\mathsf{S}_2}(u_{2i})$ be the colors from Γ^{green} of the matching elements in the two structures, for $i = 1, 2, 3$. Then, we can construct structures using S_1 and S_2 where any of these colors repeat strictly more than twice, henceforth, contradicting the conformance property to the RGB color scheme. The principle of the construction is depicted in Figure 4. Finally, note that the construction depicted in Figure 4 fuse actually only pairs of colors (G_{1i}, G_{2i}) for $i = 1, 2$. Henceforth, the conformance property is also contradicted if S_1 and S_2 can be fused by a 2-generated equivalence relation $\approx$, such that the support of either S_1 or S_2 contains more than three elements with colors in Γ^{green}. By the same argument, it follows that S is of type B, if $\approx$ is generated by two pairs of green elements. ◀

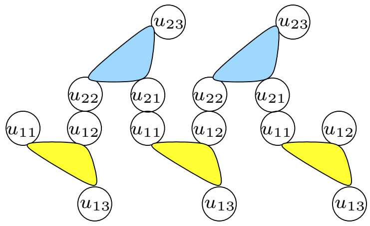

Figure 4 Fusion of G structures by 3-generated matchings.

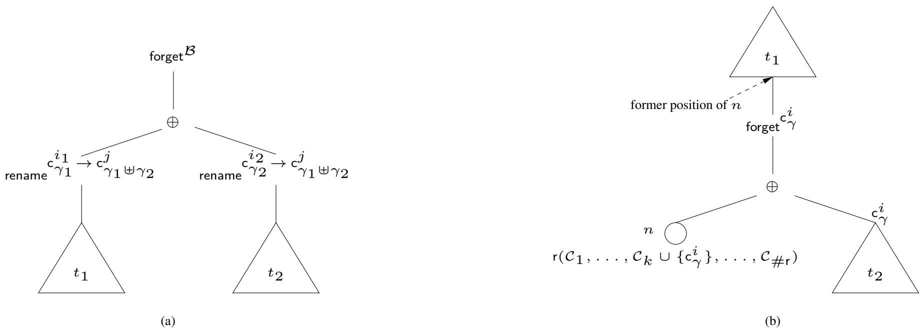

Figure 5 The $\mathsf{join}(t_1, t_2, \mathsf{c}^{i_1}_{\gamma_1}, \mathsf{c}^{i_2}_{\gamma_2})$ (a) and $\mathsf{append}(t_1, t_2, n, k, \mathsf{c}^{i}_{\gamma})$ (b) operations.

3.3 Fusions as Operations on Terms

The second technical result required for the proof of Lemma 17 is a characterization of the k-generated fusion, for $k = 1, 2$, via operations on witness $\mathbb{F}_{\mathcal{HR}}$-terms. We assume that $\mathbf{S}$ is a given tree-width bounded set of structures that conforms to a fixed RGB color scheme $(\Gamma^{red}, \Gamma^{green}, \Gamma^{blue})$. The goal is to prove that $\mathrm{tw}(\mathsf{F}^*(\mathbf{S})) \leq \mathrm{tw}(\mathbf{S}) + K$, for an integer $K \geq 0$.

Since $\mathbf{S}$ has bounded tree-width, there exists a set $\mathcal{T}$ of terms such that $\mathbf{S} = \{t^{\mathcal{HR}} \mid t \in \mathcal{T}\}$ and a finite set $\mathcal{C}$ of constants such that each term $t \in \mathcal{T}$ uses only constants from $\mathcal{C}$ and assume w.l.o.g. $\mathcal{C}$ to be the least such set. Moreover, consider the following set of *special* constants $\overline{\mathcal{C}} \stackrel{\text{def}}{=} \{\mathsf{c}^i_\gamma \mid \gamma \in \Gamma,\ 1 \leq i \leq 2\}$, with the following intuition. Recall that each element of a structure $t^{\mathcal{HR}}$ is the common interpretation of all the constants $\mathsf{c} \in \mathcal{C}_i$ in the label $\mathsf{r}(\mathcal{C}_1, \ldots, \mathcal{C}_{\#\mathsf{r}})$ of a leaf of t. By adding c^i_γ to the set $\mathcal{C}_i$, we mean that the color of that element in the structure is γ. A constant is *visible* in a term if it is not in the scope of a rename or forget operation. Let $K \stackrel{\text{def}}{=} \mathrm{card}(\overline{\mathcal{C}})$ and note that $\mathrm{card}(K) = 2 \cdot \mathrm{card}(\Gamma)$.

We shall prove that $\mathrm{tw}(\mathsf{S}) \leq \mathrm{tw}(\mathbf{S}) + K$ by building, for any $\mathsf{S} \in \mathsf{F}^*(\mathbf{S})$, a term t that uses only constants from $\mathcal{C} \uplus \overline{\mathcal{C}}$, such that $\mathsf{forget}^{\overline{\mathcal{C}}}\left(t^{\mathcal{HR}}\right) = \mathsf{S}$. We then note that S has tree-width at most $\mathrm{card}(\mathcal{C}) + K$ and obtain $\mathrm{tw}(\mathsf{S}) \leq \mathrm{card}(\mathcal{C}) + K = \mathrm{tw}(\mathbf{S}) + K$. Since the choice of $\mathsf{S} \in \mathsf{F}^*(\mathbf{S})$ was arbitrary, this leads to $\mathrm{tw}(\mathsf{F}^*(\mathbf{S})) \leq \mathrm{tw}(\mathbf{S}) + K$.

In the following, we understand terms as trees whose nodes are labeled by function symbols from $\mathbb{F}_{\mathcal{HR}}$. For each node n of a term t, we write $\mathsf{lab}(n)$ for its label. The children of each node n form an ordered sequence of length equal to the arity of $\mathsf{lab}(n)$. In order to the build the witness terms for the structures $\mathsf{S} \in \mathsf{F}^*(\mathbf{S})$, we make use of the following operations on terms t, t_1 and t_2 having constants in $\mathcal{C} \uplus \overline{\mathcal{C}}$:

1. $\mathsf{label}(t, n, k, \mathsf{c}^i_\gamma)$, where n is a leaf of t, $\mathsf{lab}(n) = \mathsf{r}(\mathcal{C}_1, \ldots, \mathcal{C}_{\#\mathsf{r}})$ and $k \in [1..\#\mathsf{r}]$: Let $1 \leq k_1 < \ldots < k_\ell \leq \#\mathsf{r}$ be the indices of the sets of constants from $\mathsf{lab}(n)$ that are equal to $\mathcal{C}_k$ (see the definition of $\mathcal{HR}$ in Section 2.2). The operation changes the label of n in t only, by replacing the set $\mathcal{C}_{k_j}$ with $\mathcal{C}_{k_j} \cup \{\mathsf{c}^i_\gamma\}$, for each $1 \leq j \leq \ell$.

2. $\mathsf{join}(t_1, t_2, \mathsf{c}^{i_1}_{\gamma_1}, \mathsf{c}^{i_2}_{\gamma_2})$: The result is the following term, for a nondeterministic choice of j, such that $\mathsf{c}^{j}_{\gamma_1 \uplus \gamma_2}$ is not visible in either t_1 or t_2 (the operation is undefined otherwise):

$$\mathsf{forget}^{\mathcal{B}}(\mathsf{rename}^{\mathsf{c}^{i_1}_{\gamma_1} \to \mathsf{c}^{j}_{\gamma_1 \uplus \gamma_2}}(t_1) \oplus \mathsf{rename}^{\mathsf{c}^{i_2}_{\gamma_2} \to \mathsf{c}^{j}_{\gamma_1 \uplus \gamma_2}}(t_2)),\ \mathcal{B} \stackrel{\text{def}}{=} \{\mathsf{c}^{j}_{\gamma_1 \uplus \gamma_2} \mid \gamma_1 \uplus \gamma_2 \in \Gamma^{blue}\}$$

We refer to Figure 5 (a) for an illustration. In addition, we consider an overloaded version of $\mathsf{join}(t_1, t_2, \mathsf{c}^{i_{11}}_{\gamma_{11}}, \mathsf{c}^{i_{12}}_{\gamma_{12}}, \mathsf{c}^{i_{21}}_{\gamma_{21}}, \mathsf{c}^{i_{22}}_{\gamma_{22}})$ that fuses the interpretation of $\mathsf{c}^{i_{1j}}_{\gamma_{1j}}$ with that of $\mathsf{c}^{i_{2j}}_{\gamma_{2j}}$, for both $j = 1, 2$. This definition is similar to the one above, thus omitted for brevity.

3. $\mathsf{append}(t_1, t_2, n, k, \mathsf{c}^{i}_{\gamma})$, where n is a leaf of t_1, $\mathsf{lab}(n) = \mathsf{r}(\mathcal{C}_1, \ldots, \mathcal{C}_{\#\mathsf{r}})$ and $k \in [1..\#\mathsf{r}]$: the result is the term $t_1[n/\mathsf{forget}^{\mathsf{c}^{i}_{\gamma}}(\mathsf{label}(n, n, k, \mathsf{c}^{i}_{\gamma}) \oplus t_2)]$, where $t[n/s]$ denotes the substitution of the leaf n by the term s in t. We refer to Figure 5 (b) for an illustration.

Then, Lemma 17 is an immediate consequence of the following lemma:

▶ **Lemma 19.** *For each structure $\mathsf{S} \in \mathsf{F}^*(\mathbf{S})$ there exists a $\mathbb{F}_{\mathcal{HR}}$-term t using only constants from $\mathcal{C} \cup \overline{\mathcal{C}}$, such that (i) $\mathsf{forget}^{\overline{\mathcal{C}}}\left(t^{\mathcal{HR}}\right) = \mathsf{S}$ and (ii) for each element $u \in \mathsf{U}_\mathsf{S}$ such that $\gamma = \mathsf{cols}_\mathsf{S}(u) \in \Gamma^{red} \uplus \Gamma^{green}$ there exists a special constant $\mathsf{c}^{i}_{\gamma} \in \overline{\mathcal{C}}$ such that $\sigma_\mathsf{S}(\mathsf{c}^{i}_{\gamma}) = u$.*

Proof *(sketch)*. We build t by induction of the derivation of $\mathsf{S} = (\mathsf{U}, \sigma) \in \mathsf{F}^*(\mathbf{S})$. For the base case $\mathsf{S} \in \mathbf{S}$, let $t' \in \mathcal{T}$ be a term such that $\mathsf{S} = t'^{\mathcal{HR}}$. By repeating the $\mathsf{label}(t, n, k, \mathsf{c}^{i}_{\gamma})$ operation, we add a special constant c^{i}_{γ} to each leaf n of t, on the appropriate position $1 \leq k \leq \#\mathsf{r}$, where $\mathsf{lab}(n) = \mathsf{r}(\mathcal{C}_1, \ldots, \mathcal{C}_{\#\mathsf{r}})$, such that $\sigma(\mathcal{C}_k) = \{u\}$ and $\gamma \stackrel{\text{def}}{=} \mathsf{cols}_\mathsf{S}(u) \in \Gamma^{red} \uplus \Gamma^{green}$. The choice of $1 \leq i \leq 2$ is nondeterministic. The result of applying these labeling operations to t' is t. Then, $\mathsf{forget}^{\overline{\mathcal{C}}}\left(t^{\mathcal{HR}}\right) = \mathsf{S}$ and (ii) holds, by construction. For the inductive step, let $\mathsf{S} = (\mathsf{S}_1 \uplus \mathsf{S}_2)_{/\approx}$, where $\mathsf{S}_1, \mathsf{S}_2 \in \mathsf{F}^*(\mathbf{S})$ and $\approx$ is an equivalence relation that is generated by the set of pairs $\{(u_{1i}, u_{2i})\}_{i \in I}$, where I is either $\{1\}$ or $\{1, 2\}$. Let $\gamma_{ji} \stackrel{\text{def}}{=} \mathsf{cols}_{\mathsf{S}_j}(u_{ji})$ for all $1 \leq j \leq 2$ and $i \in I$. By the inductive hypothesis, there exist terms t_j and integers $1 \leq k_{ji} \leq 2$, such that $(\mathsf{U}_j, \sigma_j) \stackrel{\text{def}}{=} t_j^{\mathcal{HR}}$ and $\mathsf{forget}^{\overline{\mathcal{C}}}\left(t_j^{\mathcal{HR}}\right) = \mathsf{S}_j$, for all $1 \leq j \leq 2$ and $i \in I$.

1. $I = \{1\}$, i.e., $\approx$ is 1-generated.
   a. $\gamma_{11}, \gamma_{21} \in \Gamma^{red} \uplus \Gamma^{green}$: by the inductive hypothesis (ii), there exist $\mathsf{c}^{i_1}_{\gamma_{11}}, \mathsf{c}^{i_2}_{\gamma_{21}} \in \overline{\mathcal{C}}$ such that $u_{j1} = \sigma_j(\mathsf{c}^{i_j}_{\gamma_{j1}})$, for both $1 \leq j \leq 2$. Suppose that $\mathsf{c}^{\ell}_{\gamma_{11} \uplus \gamma_{21}}$ is visible in t_1 (visibility in t_2 is a symmetric case), for some $1 \leq \ell \leq 2$. Then $\gamma_{11} \uplus \gamma_{21}$ must belong to $\Gamma^{red} \uplus \Gamma^{green}$, by the inductive hypothesis (ii). If $\mathsf{c}^{3-\ell}_{\gamma_{11} \uplus \gamma_{21}}$ is not visible in t_2, suppose first that $\mathsf{c}^{3-\ell}_{\gamma_{11} \uplus \gamma_{21}}$ is visible in t_1. Then, $\mathsf{c}^{i_1}_{\gamma_{11}}$, $\mathsf{c}^{\ell}_{\gamma_{11} \uplus \gamma_{21}}$ and $\mathsf{c}^{3-\ell}_{\gamma_{11} \uplus \gamma_{21}}$ are visible in t_1. By Lemma 15, $\gamma_{11}, \gamma_{11} \uplus \gamma_{21} \in \Gamma^{green}$ and $\gamma_{11} \uplus \gamma_{21}$ occurs 3 times in S, thus contradicting the definition of Γ^{blue}. Hence $\mathsf{c}^{3-\ell}_{\gamma_1 \uplus \gamma_2}$ is not visible in either t_1 or t_2 and $t \stackrel{\text{def}}{=} \mathsf{join}(t_1, t_2, \mathsf{c}^{i_1}_{\gamma_{11}}, \mathsf{c}^{i_2}_{\gamma_{21}})$ is well-defined. Else, if $\mathsf{c}^{3-\ell}_{\gamma_{11} \uplus \gamma_{21}}$ is visible in t_2, then $\gamma_{11} \uplus \gamma_{21}$ occurs 3 times in S, thus contradicting the definition of Γ^{blue}.
   b. $\gamma_{11} \in \Gamma^{blue}$ and $\gamma_{21} \in \Gamma^{red}$ ($\gamma_{11} \in \Gamma^{red}$ and $\gamma_{21} \in \Gamma^{blue}$ is a symmetric case): Let n be the leaf of t_1 such that $\mathsf{lab}(n) = \mathsf{r}(\mathcal{C}_1, \ldots, \mathcal{C}_{\#\mathsf{r}})$ and $\mathcal{C}_k$ be a set of constants that are (all) interpreted as u_{11}, for some $1 \leq k \leq \#\mathsf{r}$. Let $\mathsf{c}^{\ell}_{\gamma_{21}}$ be the special constant such that $u_{21} = \sigma_2(\mathsf{c}^{\ell}_{\gamma_{21}})$, for some $1 \leq \ell \leq 2$. We can assume w.l.o.g. that $\mathsf{c}^{\ell}_{\gamma_{21}}$ is not visible in n. If this were not to be the case, then n must have been involved in a previous join of a term t_3 with another term t_1', such that t_1 is the outcome of this join. In this case, we change the construction, by first joining t_2 with t_3, as in the previous case, then joining the result with t_1',. Note that this is possible due of the associativity of the $\oplus$ operation. Finally, we define $t \stackrel{\text{def}}{=} \mathsf{append}(t_1, t_2, n, k, \mathsf{c}^{\ell}_{\gamma_{21}})$.
2. $I = \{1, 2\}$, i.e., $\approx$ is 2-generated. By Lemma 18, either one of the following holds:
   a. S_1 and S_2 are of type G: let $\mathsf{c}^{k_{ji}}_{\gamma_{ji}}$ be special constants such that $u_{ji} = \sigma_j(\mathsf{c}^{k_{ji}}_{\gamma_{ji}})$, for all $1 \leq i, j \leq 2$. Then, we define $t \stackrel{\text{def}}{=} \mathsf{join}(t_1, t_2, (\mathsf{c}^{k_i}_{\gamma_{1i}})_{i \in I}, (\mathsf{c}^{k_i}_{\gamma_{2i}})_{i \in I})$ and check that the operation is well-defined, following a similar argument as in case (1a).

b. S_1 and S_2 are of type R: we can assume w.l.o.g. that u_{ii} is the interpretation of a special constant $\mathsf{c}^{j_i}_{\gamma_{ii}}$, for some $1 \leq j_i \leq 2$, where $\gamma_{ii} \in \Gamma^{red}$, for both $i = 1, 2$. Let n_1 be the leaf of t_1 such that $\mathsf{lab}(n_1) = \mathsf{r}(\mathcal{C}_1, \ldots, \mathcal{C}_{\#\mathsf{r}})$ and u_{11} is the interpretation of (all) constants from $\mathcal{C}_{k_1}$, for some $1 \leq k_1 \leq \#\mathsf{r}$. Analogously, we consider n_2 to be the leaf of t_2 and k_2 the position of the constants from its label, that are interpreted as u_{22}. Under similar assumptions as in the case (2a), ensuring that the result is well defined, we let $t \stackrel{\text{def}}{=} \mathsf{forget}^{c^{j_1}_{\gamma_{11}}}(\mathsf{append}(t_1, \mathsf{label}(t_2, n_2, k_2, \mathsf{c}^{j_1}_{\gamma_{11}}), n_1, k_1, \mathsf{c}^{j_2}_{\gamma_{22}}))$. The only difference with the previous case is that the indices j_1 and j_2 must be different to avoid name clashes, hence we require 2 special constants c^1_γ and c^2_γ, for each color $\gamma \in \Gamma^{red}$. ◀

3.4 Tree-width Bounded Fusion-closed Sets are Context-free

This subsection completes the proof of the first point of Theorem 9. The final ingredient is the following lemma, whose proof relies on the lifting of the construction that simulates the 1- or 2-generated fusion of structures from terms to tree automata recognising sets of terms:

▶ **Lemma 20.** *Let* $\mathbf{S}$ *be a context-free set of structures conforming to some RGB color scheme. Then, the set* $\mathsf{F}^*(\mathbf{S})$ *is context-free.*

Proof of Theorem 9 (1). "⇒" Since $\mathbf{S}$ is a context-free set, it has bounded tree-width. If $\mathsf{F}^*(\mathbf{S})$ has bounded tree-width, then $\mathbf{S}$ conforms to an RGB color scheme, by the combined results of Lemmas 11 and 16. Because $\mathbf{S}$ is context-free, we obtain that $\mathsf{F}^*(\mathbf{S})$ is context-free, by Lemma 20. "⇐" Because each context-free set has bounded tree-width. ◀

3.5 Color Abstractions

This section is concerned with the proof of the second point of Theorem 9. Let $\mathbf{S}$ be a context-free set of structures given by a tree automaton $\mathcal{A}$ over a finite signature $\mathcal{F} \subseteq \mathbb{F}_{\mathcal{HR}}$. In the rest of this section, $\mathbf{S}$, $\mathcal{F}$ and $\mathcal{A}$ are considered to be fixed.

Note that the equivalence between (A) and (B) follows from (A) ⇒ (B) (Lemma 11), (B) ⇒ (C) (Lemma 16) and (C) ⇒ (A) (Lemma 17). Hence, it is sufficient to establish the decidability of the condition (B) for $\mathbf{S}$. To this end, we compute the k-multiset abstraction $(\mathsf{F}^*(\mathbf{S}))^{\sharp k}$, for an arbitrary given integer $k \geq 1$ (note that checking (B) requires $k = 3$). First, we reduce the computation of $(\mathsf{F}^*(\mathbf{S}))^{\sharp k}$ to that of $\mathbf{S}^{\sharp k}$. Second, we sketch the argument behind the effective computability of $\mathbf{S}^{\sharp k}$.

The following lemma shows that, because we are interested only in k-multisets color abstractions, we can restrict fusion to 1-generated equivalence relations, while preserving the k-multiset color abstraction. We denote by $\mathsf{F}^*_1(\mathbf{S})$ the set of structures obtained by taking the closure of $\mathbf{S}$ only with respect to fusions induced by 1-generated matchings.

▶ **Lemma 21.** $(\mathsf{F}^*(\mathbf{S}))^{\sharp k} = (\mathsf{F}^*_1(\mathbf{S}))^{\sharp k}$ *for any set* $\mathbf{S}$ *of structures and integer* $k \geq 1$*.*

The set $(\mathsf{F}^*_1(\mathbf{S}))^{\sharp k}$ can be computed by a least fixpoint iteration of the following abstract operation on the domain of k-multiset color abstractions. As the later domain is finite, this fixpoint computation is guaranteed to terminate.

▶ **Definition 22.** *The* single-pair multiset fusion *is defined below, for all* $M_1, M_2 \in \mathrm{mpow}(\Gamma)$*:*

$$\mathsf{f}^\sharp_1(M_1, M_2) \stackrel{\text{def}}{=} \{M \in \mathrm{mpow}(\Gamma) \mid \exists \gamma_1 \in M_1 \,.\, \exists \gamma_2 \in M_2 \,.\, \gamma_1 \cap \gamma_2 = \emptyset, \; M = \{\!\{\gamma_1 \cup \gamma_2\}\!\} \cup \textstyle\bigcup_{i=1,2}(M_i \setminus \{\!\{\gamma_i\}\!\})\}$$

Given an integer $k \geq 1$, the single-pair k-multiset fusion *is defined for M_1, $M_2 \in \mathrm{mpow}(\Gamma)$, such that* $\mathrm{card}(M_1) \leq k$ *and* $\mathrm{card}(M_2) \leq k$*:*

$$\mathsf{f}_1^{\sharp k}(M_1, M_2) \stackrel{\text{def}}{=} \{M \mid \exists M' \in \mathsf{f}_1^{\sharp}(M_1, M_2).\ M \subseteq M',\ \mathrm{card}(M) \leq k\}$$

For a set $\mathcal{M}$ of multisets (resp. k-multisets) of colors, let $\mathsf{f}_1^{\sharp *}(\mathcal{M})$ (resp. $\mathsf{f}_1^{\sharp k *}(\mathcal{M})$) be the closure of $\mathcal{M}$ under taking single-pair fusion on multisets (resp. k-multisets).

This operation is used to compute $(\mathsf{F}_1^*(\mathbf{S}))^{\sharp k}$ by a iterating $\mathsf{f}_1^{\sharp k *}$ starting with $\mathbf{S}^{\sharp k}$ until a fixed point is reached. Since there are finitely many colors, the domain of multisets of colors having multiplicity at most k is finite, hence this iteration is guaranteed to compute $\mathsf{f}_1^{\sharp k *}(\mathbf{S}^{\sharp k})$ in finitely many steps.

▶ **Lemma 23.** $(\mathsf{F}_1^*(\mathbf{S}))^{\sharp k} = \mathsf{f}_1^{\sharp k *}(\mathbf{S}^{\sharp k})$*, for any set $\mathbf{S}$ of structures and integer $k \geq 1$.*

The last step concerns the effective computation of $\mathbf{S}^{\sharp k}$, i.e., the set of color multisets of multiplicity at most k that occur in the multiset abstraction of some structure $\mathsf{S} \in \mathbf{S}$. We leverage from the fact that $\mathbf{S}$ is a context-free set described by the set of terms that forms the language of a tree automaton $\mathcal{A}$.

We assume basic acquaintance with *context-free grammars*, i.e., finite sets of rules of the form either $q \leftarrow t[q_1, \ldots, q_k]$ or $\leftarrow q$, where $q, q_1, \ldots, q_k$ denote nonterminals and t is a $\mathbb{F}_{\mathcal{HR}}$-term with variables $q_1, \ldots, q_k$. The language $\mathcal{L}(\Gamma)$ of a grammar Γ is the set of interpretations in $\mathcal{HR}$ of the terms produced by derivations starting with an axiom $\leftarrow q$. It is well known that a tree automaton can be transformed into a context-free grammar having the same language, by turning each transition $(q_1, \ldots, q_k) \xrightarrow{f} q$ into a rule $q \leftarrow f[q_1, \ldots, q_k]$, for $k \geq 1$, and adding an axiom $\leftarrow q$ for each final state q of the tree automaton.

By *first-order logic* we understand the set of formulæ consisting of equalities between variables, relation atoms of the form $\mathsf{r}(x_1, \ldots, x_{\#\mathsf{r}})$, for some relation symbol r, composed via boolean operations and quantifiers. A first-order logic sentence φ (i.e., a formula without free variables) is interpreted over a structure S by the satisfiability relation $\mathsf{S} \models \varphi$, defined inductively on the structure of φ, as usual.

Over words, it is a well-known result that the non-emptiness of the intersection of a context-free set (e.g., given by a context-free grammar) and a regular set (e.g., given by a regular grammar, DFA, or a MSO formula) is decidable. This result has been generalized by Courcelle to context-free grammars over the HR-algebra of structures and FO-definable sets of structures[5]:

▶ **Theorem 24** (Theorem 3.6 in [7])**.** *For each grammar Γ and first-order sentence φ, one can decide the existence of a structure $\mathsf{S} \in \mathcal{L}(\Gamma)$ such that $\mathsf{S} \models \varphi$.*

In order to compute $\mathbf{S}^{\sharp k}$, we build a first-order sentence φ_M for each multiset of colors $M : \Gamma \to [1..k]$ such that $\mathsf{S} \models \varphi_M$ iff $\mathsf{S}^{\sharp k}$. As there are only finitely many such multi-sets M, we are able to construct $\mathbf{S}^{\sharp k}$ by finitely many calls to the above decision procedure. We now state the details for the construction of φ_M. For each color $\gamma \in \Gamma$, we denote by $\varphi_\gamma(X)$ the formula $\bigwedge_{\mathsf{r} \in \mathfrak{C}} \mathsf{r}(x) \wedge \bigwedge_{\mathsf{r} \notin \mathfrak{C}} \neg \mathsf{r}(x)$. We then obtain φ_M as the conjunction of the formulæ $\exists^{=M(\gamma)} x \,.\, \varphi_\gamma(x)$, if $M(\gamma) < k$, and $\exists^{\geq M(\gamma)} x \,.\, \varphi_\gamma(x)$, if $M(\gamma) = k$, for all colors $\gamma \in \Gamma$. As usual, the quantifier $\exists^{=n} x \,.\, \phi(x)$ (resp. $\exists^{\geq n} x \,.\, \phi(x)$) means "there exists exactly n (resp. at least n) elements x that satisfy $\phi(x)$". It is now easy to verify that $\mathsf{S} \models \varphi_M$ iff $M \in \mathsf{S}^{\sharp k}$. This concludes the proof of Theorem 9 (2).

[5] [7, Theorem 3.6] is actually given for Monadic Second Order Logic, which subsumes first-order logic.

4 Conclusions and Future Work

We have defined a non-aggregative and nondeterministic fusion operation on logical structures, that is controlled by a coloring of structures using unary relations. We study the tree-width of the closure of a context-free set under fusion. We prove that it is decidable whether the closure of a context-free set has bounded tree-width. Moreover, if this is the case, we show that the closure set is context-free as well, described by an effectively constructible grammar.

Future work involves considering more general notions of coloring, e.g., coloring functions defined by MSO-definable transductions. Moreover, we plan to investigate generalizations of Theorem 9 for other notions of width, such as generalizations of clique-width and rank-width from graphs and hypergraphs to relational structures.

References

1 Emma Ahrens, Marius Bozga, Radu Iosif, and Joost-Pieter Katoen. Reasoning about distributed reconfigurable systems. *Proc. ACM Program. Lang.*, 6(OOPSLA2):145–174, 2022. doi:10.1145/3563293.

2 Cristiano Calcagno, Philippa Gardner, and Uri Zarfaty. Context logic and tree update. In Jens Palsberg and Martín Abadi, editors, *Proceedings of the 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2005, Long Beach, California, USA, January 12-14, 2005*, pages 271–282. ACM, 2005. doi:10.1145/1040305.1040328.

3 Cristiano Calcagno, Philippa Gardner, and Uri Zarfaty. Context logic as modal logic: completeness and parametric inexpressivity. In Martin Hofmann and Matthias Felleisen, editors, *Proceedings of the 34th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2007, Nice, France, January 17-19, 2007*, pages 123–134. ACM, 2007. doi:10.1145/1190216.1190236.

4 Luca Cardelli, Philippa Gardner, and Giorgio Ghelli. A Spatial Logic for Querying Graphs. In Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan Eidenbenz, and Ricardo Conejo, editors, *Proceedings of the 29th International Colloquium on Automata, Languages and Programming (ICALP'02)*, volume 2380 of *Lecture Notes in Computer Science*, pages 597–610. Springer, July 2002. doi:10.1007/3-540-45465-9_51.

5 Hubert Comon, Max Dauchet, Rémi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, and Marc Tommasi. *Tree Automata Techniques and Applications*. HAL, 2008. URL: https://inria.hal.science/hal-03367725.

6 Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

7 Bruno Courcelle. The monadic second-order logic of graphs VII: graphs as relational structures. *Theor. Comput. Sci.*, 101(1):3–33, 1992. doi:10.1016/0304-3975(92)90148-9.

8 Bruno Courcelle and Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2012. doi:10.1017/CBO9780511977619.

9 Simon Docherty and David J. Pym. Intuitionistic layered graph logic: Semantics and proof theory. *Log. Methods Comput. Sci.*, 14(4), 2018. doi:10.23638/LMCS-14(4:11)2018.

10 Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite model theory*. Perspectives in Mathematical Logic. Springer, 1995.

11 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. doi:10.1007/3-540-29953-X.

12 Radu Iosif and Florian Zuleger. Expressiveness results for an inductive logic of separated relations. In Guillermo A. Pérez and Jean-François Raskin, editors, *34th International Conference on Concurrency Theory (CONCUR 2023)*, volume 279 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:20, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CONCUR.2023.20.

13 Samin S. Ishtiaq and Peter W. O'Hearn. BI as an assertion language for mutable data structures. In Chris Hankin and Dave Schmidt, editors, *Conference Record of POPL 2001: The 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, London, UK, January 17-19, 2001*, pages 14–26. ACM, 2001. doi:10.1145/360204.375719.

14 Viktor Kuncak and Martin Rinard. Generalized records and spatial conjunction in role logic. In *Static Analysis*, pages 361–376, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. doi:10.1007/978-3-540-27864-1_26.

15 Peter W. O'Hearn, John C. Reynolds, and Hongseok Yang. Local reasoning about programs that alter data structures. In *Proceedings of the 15th International Workshop on Computer Science Logic*, CSL '01, pages 1–19, 2001. doi:10.1007/3-540-44802-0_1.

16 David J. Pym, Peter W. O'Hearn, and Hongseok Yang. Possible worlds and resources: the semantics of bi. *Theoretical Computer Science*, 315(1):257–305, 2004. Mathematical Foundations of Programming Semantics. doi:10.1016/j.tcs.2003.11.020.

17 Greg Restall. Substructural Logics. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2024 edition, 2024.

18 John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *17th IEEE Symposium on Logic in Computer Science (LICS 2002), 22-25 July 2002, Copenhagen, Denmark, Proceedings*, pages 55–74. IEEE Computer Society, 2002. doi:10.1109/LICS.2002.1029817.

19 D. Seese. The structure of the models of decidable monadic theories of graphs. *Annals of Pure and Applied Logic*, 53(2):169–195, 1991. doi:10.1016/0168-0072(91)90054-P.

20 P.D. Seymour and R. Thomas. Graph searching and a min-max theorem for tree-width. *Journal of Combinatorial Theory, Series B*, 58(1):22–33, 1993. doi:10.1006/jctb.1993.1027.