

Color Refinement for Relational Structures

Benjamin Scheidt 

Humboldt-Universität zu Berlin, Germany

Nicole Schweikardt 

Humboldt-Universität zu Berlin, Germany

Abstract

Color Refinement, also known as Naive Vertex Classification, is a classical method to distinguish graphs by iteratively computing a coloring of their vertices. While it is traditionally used as an imperfect way to test for isomorphism, the algorithm has permeated many other, seemingly unrelated, areas of computer science. The method is algorithmically simple, and it has a well-understood distinguishing power: it has been logically characterized by Immerman and Lander (1990) and Cai, Fürer, Immerman (1992), who showed that it distinguishes precisely those graphs that can be distinguished by a sentence of first-order logic with counting quantifiers and only two variables. A combinatorial characterization was given by Dvořák (2010), who showed that it distinguishes precisely those graphs that differ in the number of homomorphisms from some tree.

In this paper, we introduce Relational Color Refinement (RCR, for short), a generalization of the Color Refinement method from graphs to arbitrary relational structures, whose distinguishing power admits the equivalent combinatorial and logical characterizations as Color Refinement has on graphs: we show that RCR distinguishes precisely those structures that differ in the number of homomorphisms from an acyclic connected relational structure. Further, we show that RCR distinguishes precisely those structures that are distinguished by a sentence of the guarded fragment of first-order logic with counting quantifiers. Additionally, we show that for every fixed finite relational signature, RCR can be implemented to run on structures of that signature in time $\mathcal{O}(N \cdot \log N)$, where N denotes the number of tuples present in the structure.

2012 ACM Subject Classification Theory of computation \rightarrow Finite Model Theory; Theory of computation \rightarrow Graph algorithms analysis; Mathematics of computing \rightarrow Graph theory

Keywords and phrases color refinement, counting logics, homomorphism counts, homomorphism indistinguishability, guarded logics, pebble games, relational structures, alpha-acyclicity, join-trees

Digital Object Identifier 10.4230/LIPIcs.MFCS.2025.88

Related Version *Full Version:* arxiv.org/abs/2407.16022 [35]

1 Introduction

Color Refinement (CR, for short) constitutes a simple procedure to classify the vertices of a graph G ; it is well-understood and widely used in many areas of computer science. The idea is a simple iteration: given a coloring γ of the vertices $V(G)$ of a graph G , one computes a new coloring γ' of $V(G)$ following a certain procedure. The new coloring γ' is then used to compute another coloring γ'' following the same procedure, and so on, until the coloring stabilizes, i.e., the partitioning of $V(G)$ induced by the new coloring is the same as the one induced by the previous coloring. The procedure to compute the new coloring is very simple: two vertices u, v shall get different colors if they already have different colors, or they have a different number of neighbors of some color. Otherwise, they receive the same color. To start the iteration, one either uses the colors of the vertices (if G is a colored graph), or the uniform coloring that assigns to every vertex the same color. This approach is sometimes also called “naive vertex classification” or the “1-dimensional Weisfeiler-Leman algorithm”. CR is often formalized using multisets in the following way, see e.g. [19, Chapter 3]. Considering an uncolored, undirected, simple graph G , we start with $\gamma_0(v) = 0$ for all $v \in V(G)$, and for



© Benjamin Scheidt and Nicole Schweikardt;
licensed under Creative Commons License CC-BY 4.0

50th International Symposium on Mathematical Foundations of Computer Science (MFCS 2025).

Editors: Paweł Gawrychowski, Filip Mazowiecki, and Michał Skrzypczak; Article No. 88; pp. 88:1–88:19

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

all $i \in \mathbb{N}$, we let $\gamma_{i+1}(v) := (\gamma_i(v), \{\{\gamma_i(w) : \{v, w\} \in E(G)\}\})$. Note that this formalizes the procedure described above: if $\gamma_i(u) \neq \gamma_i(v)$, then $\gamma_{i+1}(u) \neq \gamma_{i+1}(v)$; and if u and v disagree in the number of neighbors of some color, then we have $\{\{\gamma_i(w) : \{u, w\} \in E(G)\}\} \neq \{\{\gamma_i(w) : \{v, w\} \in E(G)\}\}$, hence, $\gamma_{i+1}(u) \neq \gamma_{i+1}(v)$. This formalization has the additional advantage that the colorings γ_i assign *canonical colors*, i.e., the colors themselves do not depend on the vertex set of G . Cardon and Crochemore [11] showed that CR can be implemented to run in time $\mathcal{O}((n + m) \cdot \log(n))$, where n denotes the number of vertices, and m the number of edges; and Berkholz, Bonsma, Grohe [5] showed that even a *canonical* coloring can be computed within the same running time.

Applications. An obvious application of CR is to test for graph isomorphism: if there is an $i \in \mathbb{N}$ and a c such that $|\{v \in V(G) : \gamma_i(v) = c\}| \neq |\{v \in V(H) : \gamma_i(v) = c\}|$ (we say CR *distinguishes* G and H if this is true), then G and H cannot be isomorphic. However, this test is not perfect, since there exist non-isomorphic pairs of graphs that are not distinguished by CR. Nevertheless, it is a common subroutine in practical isomorphism testers and even plays a part in Babai’s seminal result that graph isomorphism is decidable in quasi-polynomial time [3]. In recent years, the classification of “similar vertices” that CR establishes has been applied to other problems as well: it was used in [23, 22] to reduce the cost of solving linear programs, in [31] it was used to speed up the evaluation of binary acyclic conjunctive queries, and in the area of machine learning, it is used as a graph kernel [37] and was proven to be equivalent to so-called Graph Neural Networks (GNNs) w.r.t. vertex classification [38, 21].

The power of CR. The power of CR is well-understood [2, 5, 10, 15, 26, 28] – consult e.g. [19, 27] for an overview. Some key results on the distinguishing power of CR are a logical characterization due to Immerman and Lander [26] and Cai, Fürer, Immerman [10], and a combinatorial characterization w.r.t. the concept of “homomorphism indistinguishability” due to Dvořák [15] and Dell, Grohe, Rattan [13]. In [10, 26] it is shown that CR distinguishes G and H if, and only if, there is a sentence φ of first-order logic with counting quantifiers with at most 2 variables (\mathcal{C}^2 , for short) such that $G \models \varphi$ and $H \not\models \varphi$. In [15, 13] it is shown that CR distinguishes two graphs G and H if, and only if, G and H differ in the number of homomorphisms from some tree T into G and H . If no such T exists, one says that G and H are *homomorphism indistinguishable* over the class of trees. This result sparked active research in recent years exploring the concept of homomorphism indistinguishability over various graph classes, see e.g. [12, 16, 20, 29, 30, 32, 36, 33, 34]. These characterizations can explain the success of the CR method, and in particular, they give us a hint on why the vertex classification produced by CR is so powerful: two vertices $u, v \in V(G)$ get classified as “similar” by CR if the number of homomorphisms from every rooted tree T into G that map the root to u is equal to the number of homomorphisms from T into G that map the root to v .

Contributions. With the success of CR in mind, it is an obvious question how one could devise a method to color arbitrary finite relational structures, not just graphs. In particular, we would like a method that admits a combinatorial characterization w.r.t. homomorphism counts from the class of acyclic connected relational structures (for a sensible, broad definition of acyclicity) and a logical characterization for a sensible logic, analogously to the ones of CR mentioned above. We propose *Relational Color Refinement* (RCR, for short) as such a method and show that it indeed admits the desired characterizations and, at the same time, can be implemented to run in time comparable to the running time of classical CR.

Our main result reads as follows.

► **Theorem A.** *Let σ be a finite relational signature.*

- (a) *RCR can be implemented to run in time $\mathcal{O}(N \cdot \log N)$ upon input of a σ -structure \mathcal{A} , where N denotes the number of tuples present in \mathcal{A} .*
- (b) *For all σ -structures \mathcal{A} and \mathcal{B} , the following statements are equivalent:*
 - (1) *RCR distinguishes \mathcal{A} and \mathcal{B} .*
 - (2) *There exists an acyclic and connected σ -structure \mathcal{C} such that $\text{hom}(\mathcal{C}, \mathcal{A}) \neq \text{hom}(\mathcal{C}, \mathcal{B})$.*
 - (3) *There exists a sentence $\varphi \in \text{GF}(\mathcal{C})$ such that $\mathcal{A} \models \varphi$ and $\mathcal{B} \not\models \varphi$.*
 - (4) *Spoiler wins the Guarded Game on \mathcal{A}, \mathcal{B} .*

Here, $\text{hom}(\mathcal{C}, \mathcal{A})$ denotes the number of homomorphisms from \mathcal{C} to \mathcal{A} (cf. Section 4), while $\text{GF}(\mathcal{C})$ is the *guarded fragment* of the logic \mathcal{C} (cf. Section 5.1), and the *Guarded Game* is a particular variant of Ehrenfeucht-Fraïssé games (cf. Section 5.2). The technically most challenging parts are proving (a) and proving the equivalence of (1) and (2) in (b).

Section 2 introduces the necessary basic concepts and notations used throughout the paper. In Section 3 we introduce RCR, discuss its connection to CR, and provide a proof of Theorem A (a). Section 4 is devoted to the proof of the equivalence of statements (1) and (2) of Theorem A (b). Section 5 starts by introducing the logic $\text{GF}(\mathcal{C})$ and the Guarded Game, followed by the proof of the equivalence of statements (1), (3) and (4) of Theorem A (b). Finally, we conclude in Section 6 with a summary and an outlook on future work. Due to space limitations, many details had to be deferred to the paper’s full version.

Further related work. The articles [2, 28] studied related, but different questions that do not concern the power of distinguishing two given graphs, but rather the power of distinguishing one given graph from *all* other graphs. Arvind, Köbler, Rattan, Verbitsky [2] provided a characterization of those graphs G that are *amenable* by CR in the sense that CR distinguishes G from *every* graph H that is not isomorphic to G . Kiefer, Schweitzer, Selman [28] provided a characterization of those graphs and, more generally, finite relational structures that can be *identified* (i.e., described up to isomorphism) by a sentence of the logic \mathcal{C}^2 . In [2] it is noted that by the result of [26, 10] it follows that “amenability by CR” and “identifiability by \mathcal{C}^2 ” are equivalent notions.

The work by Dell, Grohe, Rattan [13] has been generalized to relational structures by Butti and Dalmau [8]. However, they apply classical CR on the incidence graph of the relational structure and use the weaker notion of Berge-acyclicity that is subsumed by our notion of acyclicity; consequently, the distinguishing power of their algorithm is considerably weaker than that of RCR.

There is also related work in this direction on hypergraphs, which are conceptually similar to relational structures. Böker [9] introduced a variant of CR that works on hypergraphs, and showed that it distinguishes two hypergraphs if, and only if, there is a Berge-acyclic hypergraph that has a different number of homomorphisms to them. Again, the distinguishing power yielded by Berge-acyclicity is considerably weaker than that of the more general acyclicity notion considered in our work. The connection between the logic \mathcal{C}^2 and homomorphism counts from trees due to [15, 13] was generalized to the logic GC (a logic similar to the guarded fragment $\text{GF}(\mathcal{C})$, but tailored towards hypergraphs) and homomorphism counts from acyclic hypergraphs by Scheidt and Schweikardt [34].

Riveros, Scheidt, Schweikardt [31] use CR to speed up the evaluation of acyclic conjunctive queries on edge- and vertex-labeled graphs. Beyond CR itself, there is no overlap in the technical contributions between this paper and [31].

2 Preliminaries

Basic notation. We write \mathbb{N} for the set of non-negative integers, and we let $\mathbb{N}_{\geq 1} := \mathbb{N} \setminus \{0\}$. For $n \in \mathbb{N}$ we write $[n]$ to denote the set $\{i \in \mathbb{N} : 1 \leq i \leq n\}$ (i.e., $[0] = \emptyset$, $[1] = \{1\}$, and $[n] = \{1, \dots, n\}$ for $n \geq 2$). For a set S we write 2^S to denote the power set (i.e., the set of all subsets) of S ; and for $k \in \mathbb{N}$ we let $\binom{S}{k} := \{X \subseteq S : |X| = k\}$. We use bold letters \mathbf{a} to denote tuples (a_1, \dots, a_k) . The tuple's *arity* $\text{ar}(\mathbf{a})$ is k , and a_i denotes the tuple's i -th entry (for $i \in [k]$). We let $\text{set}(\mathbf{a}) = \{a_1, \dots, a_k\}$. We write $\{a_1 \rightarrow b_1, \dots, a_k \rightarrow b_k\}$ to describe the function $f: \{a_1, \dots, a_k\} \rightarrow \{b_1, \dots, b_k\}$ with $f(a_i) = b_i$ for $i \in [k]$.

A *multiset* M is a tuple (S, f) , where S is a set and f is a function $f: S \rightarrow \mathbb{N}_{\geq 1}$; the number $f(s)$ indicates the multiplicity with which the element $s \in S$ occurs in the multiset M . We write $\text{mult}_M(x)$ to denote the multiplicity of x in the multiset M ; in particular, $\text{mult}_M(x) = 0$ denotes that $x \notin S$. We adopt the usual notation for multisets using brackets $\{\!\{ \dots \}\!\}$ in which each $s \in S$ is listed exactly $\text{mult}_M(s)$ times. E.g., $\{\!\{a, a, b\}\!\}$ denotes the multiset $(\{a, b\}, \{a \rightarrow 2, b \rightarrow 1\})$.

A *coloring* of a set S is a function $\gamma: S \rightarrow C$ for some set C . Let $\alpha: S \rightarrow C_\alpha$ and $\beta: S \rightarrow C_\beta$ be two colorings of the same set S . We say that α *refines* β , if for all $u, v \in S$: $\alpha(u) = \alpha(v) \implies \beta(u) = \beta(v)$. The colorings α and β are *equivalent*, if for all $u, v \in S$: $\alpha(u) = \alpha(v) \iff \beta(u) = \beta(v)$.

An (uncolored, undirected, simple) graph is a tuple $G := (V(G), E(G))$, where $V(G)$ is a finite set of vertices and $E(G) \subseteq \binom{V(G)}{2}$ is a set of edges. A *forest* is an acyclic graph; and a *tree* is a connected forest.

Relational Structures. A (*finite, relational*) *signature* σ is a finite, non-empty set; the elements in σ are called *relation symbols*. Every $R \in \sigma$ has an associated *arity* $\text{ar}(R) \in \mathbb{N}_{\geq 1}$. The *arity of* σ is defined as $\text{ar}(\sigma) := \max\{\text{ar}(R) : R \in \sigma\}$. By $\sigma|_k$, for $k \in \mathbb{N}_{\geq 1}$, we denote the subset $\{R \in \sigma : \text{ar}(R) = k\}$ of relation symbols of σ with arity exactly k .

A structure \mathcal{A} of signature σ (for short, σ -*structure*) consists of a finite, non-empty set $V(\mathcal{A})$ (called the *universe* of \mathcal{A}), and a relation $R^{\mathcal{A}} \subseteq V(\mathcal{A})^{\text{ar}(R)}$ for every $R \in \sigma$. We additionally require that every $v \in V(\mathcal{A})$ occurs as an entry in at least one tuple of at least one relation of \mathcal{A} – note that this assumption can easily be met, e.g. by inserting into σ a new relation symbol U of arity 1 with $U^{\mathcal{A}} = V(\mathcal{A})$ (here, we identify a tuple (v) of arity 1 with the element v). By \mathbf{A} we denote the set $\bigcup_{R \in \sigma} R^{\mathcal{A}}$ of all tuples that belong to some relation of \mathcal{A} . We define the *size*¹ of \mathcal{A} as $\|\mathcal{A}\| := |\mathbf{A}|$. We say that two σ -structures \mathcal{A}, \mathcal{B} have *strictly equal size*, if $|R^{\mathcal{A}}| = |R^{\mathcal{B}}|$ for every $R \in \sigma$.

The *Gaifman graph* of \mathcal{A} is defined as the (undirected, simple) graph G with $V(G) = V(\mathcal{A})$ and where $E(G)$ consists of all $\{u, v\} \in \binom{V(G)}{2}$ for which there is a tuple $\mathbf{a} \in \mathbf{A}$ with $u, v \in \text{set}(\mathbf{a})$. A σ -structure is called *connected* if its Gaifman graph is connected.

A *binary signature* is a signature σ where every $R \in \sigma$ has arity ≤ 2 . A *colored multigraph* \mathcal{G} is a structure of a binary signature. The binary relations of \mathcal{G} can be viewed as directed edge relations that carry specific labels, and the unary relations of \mathcal{G} can be viewed as assigning specific labels to the vertices of \mathcal{G} .

¹ In the literature, usually $\|\mathcal{A}\|$ denotes the size of a reasonable representation of \mathcal{A} as input for an algorithm. Since we consider σ to be a *fixed* signature and restrict attention to structures where each $v \in V(\mathcal{A})$ occurs in at least one tuple in \mathbf{A} , within the \mathcal{O} -notation our notion of $\|\mathcal{A}\|$ is equivalent to the one used in the literature.

Color Refinement (CR). Color Refinement (CR) can be adapted to colored multigraphs by including the vertex labels and the loops in the base color, and the edge labels in the iteration. This can be formalized as follows. Let σ be a binary signature, and let \mathcal{G} be a σ -structure. For every $v \in V(\mathcal{G})$, let $\gamma_0(v) = (\{C \in \sigma|_1 : v \in C^{\mathcal{G}}\}, \{R \in \sigma|_2 : (v, v) \in R^{\mathcal{G}}\})$, and for all $i \in \mathbb{N}$ let

$$\gamma_{i+1}(v) := (\gamma_i(v), \{\{\lambda(v, w), \gamma_i(w)\} : \{v, w\} \in E(G)\}),$$

where G denotes the Gaifman graph of \mathcal{G} and

$$\lambda(v, w) := \{R^+ : R \in \sigma|_2, (v, w) \in R^{\mathcal{G}}\} \cup \{R^- : R \in \sigma|_2, (w, v) \in R^{\mathcal{G}}\}.$$

Types. The notion of *atomic type* and the *similarity type* of tuples will be crucial for the definition of RCR. Let σ be an arbitrary signature. For a σ -structure \mathcal{A} and a tuple $\mathbf{a} \in V(\mathcal{A})^k$ of arity k , the *atomic type* $\text{atp}(\mathbf{a})$ is the set $\{R \in \sigma : \mathbf{a} \in R^{\mathcal{A}}\}$. For every tuple $\mathbf{b} \in V(\mathcal{A})^\ell$ of arity ℓ , the *similarity type* $\text{stp}(\mathbf{a}, \mathbf{b})$ between \mathbf{a} and \mathbf{b} is defined as the set $\{(i, j) \in [k] \times [\ell] : a_i = b_j\}$. We use $\text{stp}(\mathbf{a})$ as shorthand for $\text{stp}(\mathbf{a}, \mathbf{a})$.

In general, an *atomic type* ρ of arity k (over signature σ) is a subset of $\sigma|_k$. A *similarity type* τ of arity (k, ℓ) (over signature σ) is a subset of $[k] \times [\ell]$ that satisfies the following condition of “rectangularity”: for all $i, i' \in [k]$ and $j, j' \in [\ell]$, if $\{(i, j), (i', j), (i, j')\} \subseteq \tau$ then $(i', j') \in \tau$. We write $\text{ar}(\rho)$ and $\text{ar}(\tau)$ to denote the arity of ρ and τ . If τ has arity (k, k) , we simply say that τ has arity k and write $\text{ar}(\tau) = k$. We let STP_σ be the set of all similarity types of arity $(\text{ar}(R), \text{ar}(S))$ for any $R, S \in \sigma$. We say that a tuple $\mathbf{a} \in V(\mathcal{A})^k$ has atomic type ρ if $\rho = \text{atp}(\mathbf{a})$. Analogously, we say that \mathbf{a} has similarity type τ , if $\tau = \text{stp}(\mathbf{a})$; and we say that \mathbf{a}, \mathbf{b} have similarity type τ , if $\tau = \text{stp}(\mathbf{a}, \mathbf{b})$.

3 Color Refinement on Relational Structures

The goal of this section is to introduce RCR as a generalization of CR from graphs to relational structures. Let σ be an arbitrary (relational) signature; this σ will be fixed throughout the rest of the paper.

3.1 Relational Color Refinement (RCR, for short) – Definition

Consider an arbitrary σ -structure \mathcal{A} . The key idea of RCR is to color the tuples in \mathcal{A} and take into account the tuples’ atomic type and their mutual overlap (the latter is formalized by their similarity type). The details are as follows.

We iteratively compute colors for every tuple $\mathbf{a} \in \mathcal{A}$. For every $\mathbf{a} \in \mathcal{A}$, the initial color consists of the atomic type and the similarity type of \mathbf{a} , i.e., it is $\varrho_0(\mathbf{a}) := (\text{atp}(\mathbf{a}), \text{stp}(\mathbf{a}))$. For $i \in \mathbb{N}_{\geq 1}$, the color after i iterations is defined as $\varrho_i(\mathbf{a}) := (\varrho_{i-1}(\mathbf{a}), N_i^{\mathcal{A}}(\mathbf{a}))$, where

$$N_i^{\mathcal{A}}(\mathbf{a}) := \{\{\text{stp}(\mathbf{a}, \mathbf{b}), \varrho_{i-1}(\mathbf{b})\} : \mathbf{b} \in \mathcal{A}, \text{stp}(\mathbf{a}, \mathbf{b}) \neq \emptyset\}.$$

Note that $\text{stp}(\mathbf{a}, \mathbf{b}) \neq \emptyset \iff \text{set}(\mathbf{a}) \cap \text{set}(\mathbf{b}) \neq \emptyset$. By definition, ϱ_i refines ϱ_{i-1} for all $i \in \mathbb{N}_{\geq 1}$. The i -th coloring is *stable*, if for all $\mathbf{a}, \mathbf{b} \in \mathcal{A}$ we have $\varrho_i(\mathbf{a}) = \varrho_i(\mathbf{b}) \iff \varrho_{i+1}(\mathbf{a}) = \varrho_{i+1}(\mathbf{b})$. It is easy to see that for every σ -structure \mathcal{A} there is an $i \in \mathbb{N}$ such that the i -th coloring is stable; we let $i_{\mathcal{A}}$ be the smallest such number, and we write $\varrho_\infty(\mathbf{a})$ to denote $\varrho_{i_{\mathcal{A}}}(\mathbf{a})$.

For $i \in \mathbb{N}$ we write $\text{RC}_i(\mathcal{A})$ to denote the set of colors produced in the i -th refinement round, i.e., $\text{RC}_i(\mathcal{A}) = \{\varrho_i(\mathbf{a}) : \mathbf{a} \in \mathcal{A}\}$. For each $c \in \text{RC}_i(\mathcal{A})$ we let $\text{mult}_{\mathcal{A}}(c) := |\{\mathbf{a} \in \mathcal{A} : \varrho_i(\mathbf{a}) = c\}|$, i.e., $\text{mult}_{\mathcal{A}}(c)$ is the number of tuples with color c . We let $\text{RC}(\mathcal{A}) := \text{RC}_\infty(\mathcal{A}) := \text{RC}_{i_{\mathcal{A}}}(\mathcal{A})$; and we will call this the set of *stable colors on \mathcal{A} produced by RCR*.

We say that RCR *distinguishes* two σ -structures \mathcal{A} and \mathcal{B} in round i , if there is a color $c \in \text{RC}_i(\mathcal{A}) \cup \text{RC}_i(\mathcal{B})$ such that $\text{mult}_{\mathcal{A}}(c) \neq \text{mult}_{\mathcal{B}}(c)$. Furthermore, we say that RCR *distinguishes* \mathcal{A} and \mathcal{B} if there is an $i \leq \max\{i_{\mathcal{A}}, i_{\mathcal{B}}\}$ such that RCR distinguishes \mathcal{A} and \mathcal{B} in round i . It is straightforward to see that if \mathcal{A} and \mathcal{B} are *not* of strictly equal size, then RCR distinguishes \mathcal{A} and \mathcal{B} in round 0.

A run of RCR on particular example structures can be found in the paper's full version.

3.2 Connection between RCR and CR

For the signature $\sigma = \{E, U\}$ with $\text{ar}(E) = 2$ and $\text{ar}(U) = 1$, the following is straightforward to see. For any (simple, undirected) graph G let \mathcal{A}_G be the σ -structure \mathcal{A} that represents G as follows: $V(\mathcal{A}) = U^{\mathcal{A}} = V(G)$ and $E^{\mathcal{A}}$ consists of the tuples (u, v) and (v, u) for all $\{u, v\} \in E(G)$. RCR on \mathcal{A}_G produces a stable coloring γ that is equivalent to the stable coloring γ' produced by CR on G . Therefore, RCR can be viewed as a generalization of CR from graphs to σ -structures for arbitrary (relational) signatures σ .

Next, we point out that running RCR on a σ -structure \mathcal{A} produces the same result as running CR on a suitably defined colored multigraph representation $\mathcal{G}_{\mathcal{A}}$ of \mathcal{A} . The colored multigraph $\mathcal{G}_{\mathcal{A}}$ can be viewed as a suitably colored generalization, from graphs to relational structures, of the notion of the *line graph* $L(G)$ (cf. [14]) associated with an undirected graph G . It will be crucial for the proof of Theorem A (b).

► **Definition 3.1.** We represent a σ -structure \mathcal{A} by a colored multigraph $\mathcal{G}_{\mathcal{A}}$ of the signature

$$\hat{\sigma} := \{E_{i,j} : i, j \in [\text{ar}(\sigma)]\} \cup \{U_R : R \in \sigma\},$$

where $\text{ar}(E_{i,j}) = 2$ for all $i, j \in [\text{ar}(\sigma)]$ and $\text{ar}(U_R) = 1$ for all $R \in \sigma$.

The universe $V(\mathcal{G}_{\mathcal{A}})$ of $\mathcal{G}_{\mathcal{A}}$ consists of a new element $w_{\mathbf{a}}$ for every tuple $\mathbf{a} \in \mathbf{A}$. Furthermore, $(U_R)^{\mathcal{G}_{\mathcal{A}}} = \{w_{\mathbf{a}} : \mathbf{a} \in R^{\mathcal{A}}\}$, for all $R \in \sigma$. And for all $i, j \in [\text{ar}(\sigma)]$ we have $(E_{i,j})^{\mathcal{G}_{\mathcal{A}}} := \{(w_{\mathbf{a}}, w_{\mathbf{b}}) : \mathbf{a}, \mathbf{b} \in \mathbf{A}, (i, j) \in \text{stp}(\mathbf{a}, \mathbf{b})\}$.

► **Example 3.2.** Consider $\sigma_1 := \{E, R\}$ with $\text{ar}(E) = 2$ and $\text{ar}(R) = 6$, and let \mathcal{A}_1 be the σ_1 -structure with the universe $\{1, 2, 3, u, v, w\}$, where $E^{\mathcal{A}_1} := \{(1, 2), (2, 3), (3, 1), (u, v), (v, w), (w, u)\}$, and $R^{\mathcal{A}_1} := \{(1, 2, 3, u, v, w)\}$. The representation $\mathcal{G}_{\mathcal{A}_1}$ of \mathcal{A}_1 as a colored multigraph is depicted in Figure 1a. To keep the figure easy to grasp, we labeled the vertices with the tuples they represent, omitted the self-loops, and contracted multi-edges into a single one with a combined edge label, where xy denotes the tuple (x, y) .

It is easy to see that running RCR on a σ -structure \mathcal{A} produces (in the same number of rounds) a stable coloring of \mathbf{A} that is equivalent (via identifying $\mathbf{a} \in \mathbf{A}$ with $w_{\mathbf{a}} \in V(\mathcal{G}_{\mathcal{A}})$) to the stable coloring produced by classical CR on the colored multigraph $\mathcal{G}_{\mathcal{A}}$.

Let us define the *cohesion* of \mathcal{A} as the number $\Gamma(\mathcal{A})$ of all tuples $(\mathbf{a}, \mathbf{b}) \in \mathbf{A} \times \mathbf{A}$ with $\mathbf{a} \neq \mathbf{b}$ and $\text{set}(\mathbf{a}) \cap \text{set}(\mathbf{b}) \neq \emptyset$. Obviously, $\Gamma(\mathcal{A}) < \|\mathcal{A}\|^2$.

It is known that classical CR can be implemented to run in time $\mathcal{O}((n + m) \cdot \log(n))$ on colored multigraphs, where n denotes the number of vertices and m denotes the total number of edges (cf., [11, 5]). Thus, on the colored multigraph $\mathcal{G}_{\mathcal{A}}$ representing \mathcal{A} , CR runs in time $\mathcal{O}((n + m) \cdot \log(n))$ where $n = |\mathbf{A}| = \|\mathcal{A}\|$ and $m = \sum_{i,j \in [\text{ar}(\sigma)]} |(E_{i,j})^{\mathcal{G}_{\mathcal{A}}}| = \mathcal{O}(\Gamma(\mathcal{A}))$ (for each fixed signature σ). Thus, by running CR on $\mathcal{G}_{\mathcal{A}}$, we obtain an implementation of RCR on \mathcal{A} that, for each fixed signature σ , runs in time $\mathcal{O}((\|\mathcal{A}\| + \Gamma(\mathcal{A})) \cdot \log(\|\mathcal{A}\|)) = \mathcal{O}(\|\mathcal{A}\|^2 \cdot \log(\|\mathcal{A}\|))$. In the next subsection, we will improve the running time to $\mathcal{O}(\|\mathcal{A}\| \cdot \log\|\mathcal{A}\|)$ by using a representation of \mathcal{A} by a colored multigraph different from $\mathcal{G}_{\mathcal{A}}$.



3.3 Implementing RCR in Time $\mathcal{O}(\|\mathcal{A}\| \cdot \log(\|\mathcal{A}\|))$

► **Theorem B.** *For each fixed signature σ , RCR can be implemented to run in time $\mathcal{O}(\|\mathcal{A}\| \cdot \log(\|\mathcal{A}\|))$ upon input of a σ -structure \mathcal{A} .*

Recall from Section 3.2 that by performing classical CR on the colored multigraph $\mathcal{G}_{\mathcal{A}}$ we can implement RCR on a σ -structure \mathcal{A} with runtime $\mathcal{O}((n + m) \cdot \log(n))$, where n denotes the number of vertices and m denotes the total number of edges of $\mathcal{G}_{\mathcal{A}}$. Note that the nodes $w_{\mathbf{a}}$ for all tuples $\mathbf{a} \in \mathbf{A}$ that share an element form a clique in $\mathcal{G}_{\mathcal{A}}$. This causes a blow-up of the number of edges in $\mathcal{G}_{\mathcal{A}}$. We will alleviate this by resolving every clique by inserting a constant number of *fresh* vertices that are connected to all tuples participating in a clique. This will drastically reduce the number of edges, yielding a new colored multigraph $\mathcal{H}_{\mathcal{A}}$ whose number of nodes and edges is in $\mathcal{O}(\|\mathcal{A}\|)$. For the precise definition of $\mathcal{H}_{\mathcal{A}}$ we need the following notation.

► **Definition 3.4.** Let \mathcal{A} be a σ -structure. Let $\mathcal{H}_{\mathcal{A}}$ be the colored multigraph of signature $\widehat{\sigma}$ defined as follows. The universe $V(\mathcal{H}_{\mathcal{A}})$ consists of the nodes w_a for all $a \in \mathbf{A}$ and a new node v_s for every slice $s \in \mathcal{S}(\mathbf{A}) := \bigcup_{a \in \mathbf{A}} \mathcal{S}(a)$. I.e., $V(\mathcal{H}_{\mathcal{A}}) = \{w_a : a \in \mathbf{A}\} \dot{\cup} \{v_s : s \in \mathcal{S}(\mathbf{A})\}$. Furthermore, $(U_R)^{\mathcal{H}_{\mathcal{A}}} := \{w_a : a \in R^{\mathcal{A}}\}$, for all $R \in \sigma$. And for all $i, j \in [\text{ar}(\sigma)]$ we let $(E_{i,j})^{\mathcal{H}_{\mathcal{A}}} := \{(w_a, v_s) : a \in \mathbf{A}, s \in \mathcal{S}(a), (i, j) \in \text{stp}(a, s)\} \cup \{(v_s, w_b) : b \in \mathbf{A}, s \in \mathcal{S}(b), (i, j) \in \text{stp}(s, b)\}$.

► **Example 3.5.** Consider the signature $\sigma_2 := \{R\}$ with $\text{ar}(R) = 3$. Then, $\hat{\sigma}_2$ consists of the unary relation symbol U_R and binary relation symbols $E_{i,j}$ for $i, j \in \{1, 2, 3\}$.

Let \mathcal{A}_2 be the σ_2 -structure where $V(\mathcal{A}_2) := \{1, 2, 3\}$ and $R^{\mathcal{A}_2} := \{(1, 1, 2), (2, 3, 2)\}$. Then, $\mathbf{A}_2 = R^{\mathcal{A}_2}$ and $\mathcal{S}(\mathbf{A}_2) = \{(1), (2), (3), (1, 2), (2, 1), (2, 3), (3, 2)\}$. The colored multigraph $\mathcal{H}_{\mathcal{A}_2}$ is the $\widehat{\sigma}_2$ -structure with $V(\mathcal{H}_{\mathcal{A}_2}) = \{w_{(1,1,2)}, w_{(2,3,2)}\} \cup \{v_s : s \in \mathcal{S}(\mathbf{A}_2)\}$. The unary symbol U_R is interpreted by the set $(U_R)^{\mathcal{H}_{\mathcal{A}_2}} = \{w_{(1,1,2)}, w_{(2,3,2)}\}$. See Figure 1b for an illustration of $\mathcal{H}_{\mathcal{A}_2}$.

Note that for each $\mathbf{a} \in \mathbf{A}$ of arity k , the number of slices of \mathbf{a} is $|\mathcal{S}(\mathbf{a})| \leq k \cdot k!$. Thus, for $k := \text{ar}(\sigma) = \max\{\text{ar}(R) : R \in \sigma\}$ we have $|V(\mathcal{H}_{\mathcal{A}})| = |\mathbf{A}| + |\mathcal{S}(\mathbf{A})| \leq (1 + k \cdot k!) \cdot |\mathbf{A}|$ and $|(E_{i,j})^{\mathcal{H}_{\mathcal{A}}}| \leq 2 \cdot k \cdot k! \cdot |\mathbf{A}|$, for all $i, j \in [k]$. Hence, the total number of edges of $\mathcal{H}_{\mathcal{A}}$ is at most $2 \cdot k^3 \cdot k! \cdot |\mathbf{A}|$. I.e., for the fixed relational signature σ , the number of nodes and edges of the colored multigraph $\mathcal{H}_{\mathcal{A}}$ associated with a σ -structure \mathcal{A} is of size $\mathcal{O}(\|\mathcal{A}\|)$, where the factor hidden by the \mathcal{O} -notation is bounded by $2 \cdot k^3 \cdot k! = 2^{\mathcal{O}(k \cdot \log k)}$ for $k := \text{ar}(\sigma)$.

Since the number of vertices and the number of edges in $\mathcal{H}_{\mathcal{A}}$ both are of size $\mathcal{O}(\|\mathcal{A}\|)$, Theorem B is obtained as an immediate consequence of the following Theorem 3.6 and the known running time of CR (cf. Sections 1 and 3.2):

► **Theorem 3.6.** *Let \mathcal{A} be a σ -structure. Let $\varrho_i(\mathbf{a})$ be the color assigned to tuple $\mathbf{a} \in \mathbf{A}$ in the i -th round of Relational Color Refinement RCR on \mathcal{A} , and let $\gamma_i(u)$ be the color assigned to node u of $\mathcal{H}_{\mathcal{A}}$ in round i of conventional Color Refinement CR on the colored multigraph $\mathcal{H}_{\mathcal{A}}$. For all $i \in \mathbb{N}$ and all $\mathbf{a}, \mathbf{b} \in \mathbf{A}$ we have: $\varrho_i(\mathbf{a}) = \varrho_i(\mathbf{b}) \iff \gamma_{2i+1}(w_{\mathbf{a}}) = \gamma_{2i+1}(w_{\mathbf{b}})$.*

The remainder of this section is devoted to the proof of Theorem 3.6. See the paper's full version for detailed proofs of all subsequent lemmas and claims. We start with a lemma that summarizes some obvious facts.

► **Lemma 3.7.** *Let \mathcal{A} be a σ -structure. Let $k, k', \ell \geq 1$ and let $\mathbf{a} = (a_1, \dots, a_k)$ and $\mathbf{b} = (b_1, \dots, b_{k'})$ be elements in \mathbf{A} , and let $\mathbf{s} = (s_1, \dots, s_\ell)$ be a slice over $V(\mathcal{A})$.*

- (a) $\text{stp}(\mathbf{a}, \mathbf{b}) \neq \emptyset \iff \text{set}(\mathbf{a}) \cap \text{set}(\mathbf{b}) \neq \emptyset$.
- (b) $\text{stp}(\mathbf{a}) = \text{stp}(\mathbf{b}) \iff \text{ar}(\mathbf{a}) = \text{ar}(\mathbf{b})$ and the function $\beta: \text{set}(\mathbf{a}) \rightarrow \text{set}(\mathbf{b})$ with $\beta(a_i) := b_i$ for all $i \in [k]$ is well-defined and bijective.
- (c) $\mathbf{s} \in \mathcal{S}(\mathbf{a}) \iff$ for every $i \in [\ell]$ there exists a $j \in [k]$ such that $(i, j) \in \text{stp}(\mathbf{s}, \mathbf{a}) \iff$ for every $j \in [\ell]$ there exists an $i \in [k]$ such that $(i, j) \in \text{stp}(\mathbf{a}, \mathbf{s})$.
- (d) Let $\mathbf{s} \in \mathcal{S}(\mathbf{a})$. For all $\mathbf{s}' \in \mathcal{S}(\mathbf{a})$ we have: $\text{stp}(\mathbf{a}, \mathbf{s}) = \text{stp}(\mathbf{a}, \mathbf{s}') \iff \mathbf{s} = \mathbf{s}'$.

Two nodes $u, u' \in V(\mathcal{H}_{\mathcal{A}})$ are called *neighbors in $\mathcal{H}_{\mathcal{A}}$* if $(u, u') \in (E_{i,j})^{\mathcal{H}_{\mathcal{A}}}$ for some $i, j \in [\text{ar}(\sigma)]$.

► **Remark 3.8.** Note that for all $\mathbf{a} \in \mathbf{A}$ and $\mathbf{s} \in \mathcal{S}(\mathbf{A})$ we have: $w_{\mathbf{a}}$ and $v_{\mathbf{s}}$ are neighbors in $\mathcal{H}_{\mathcal{A}} \iff \mathbf{s} \in \mathcal{S}(\mathbf{a})$. Also, by Lemma 3.7 (d) we obtain: For all $\mathbf{a} \in \mathbf{A}$ and all $\mathbf{s}, \mathbf{s}' \in \mathcal{S}(\mathbf{a})$ with $\mathbf{s} \neq \mathbf{s}'$, we have $\text{stp}(\mathbf{a}, \mathbf{s}) \neq \text{stp}(\mathbf{a}, \mathbf{s}')$.

The following characterization of tuples \mathbf{a}, \mathbf{b} with $\text{stp}(\mathbf{a}) = \text{stp}(\mathbf{b})$ will be crucial for our proof of Theorem 3.6.

► **Lemma 3.9.** *Let \mathcal{A} be a σ -structure. For all $\mathbf{a}, \mathbf{b} \in \mathbf{A}$ we have: $\text{stp}(\mathbf{a}) = \text{stp}(\mathbf{b}) \iff$ there is a bijection $\pi_{\mathcal{S}}: \mathcal{S}(\mathbf{a}) \rightarrow \mathcal{S}(\mathbf{b})$ such that for all $\mathbf{s} \in \mathcal{S}(\mathbf{a})$ we have $\text{stp}(\mathbf{a}, \mathbf{s}) = \text{stp}(\mathbf{b}, \pi_{\mathcal{S}}(\mathbf{s}))$.*

It follows from Remark 3.8 that the bijection $\pi_{\mathcal{S}}$ is unique, if it exists. The following lemma summarizes straightforward properties of the mapping $\pi_{\mathcal{S}}$ obtained from Lemma 3.9.

► **Lemma 3.10.** *Let \mathcal{A} be a σ -structure, let $\mathbf{a}, \mathbf{b} \in \mathbf{A}$ with $\text{stp}(\mathbf{a}) = \text{stp}(\mathbf{b})$, and let $\pi_{\mathcal{S}}: \mathcal{S}(\mathbf{a}) \rightarrow \mathcal{S}(\mathbf{b})$ be the bijection obtained from Lemma 3.9. For all $\mathbf{s}, \mathbf{s}' \in \mathcal{S}(\mathbf{a})$ and for $\mathbf{t} := \pi_{\mathcal{S}}(\mathbf{s})$ and $\mathbf{t}' := \pi_{\mathcal{S}}(\mathbf{s}')$ we have: (1) $\text{ar}(\mathbf{s}) = \text{ar}(\mathbf{t})$, and (2) $\text{set}(\mathbf{s}) \subseteq \text{set}(\mathbf{s}') \iff \text{set}(\mathbf{t}) \subseteq \text{set}(\mathbf{t}')$, and (3) $\text{set}(\mathbf{s}) = \text{set}(\mathbf{s}') \iff \text{set}(\mathbf{t}) = \text{set}(\mathbf{t}')$.*

For $\mathbf{a} \in \mathbf{A}$ let $N(\mathbf{a}) := \{\mathbf{c}' \in \mathbf{A} : \text{stp}(\mathbf{a}, \mathbf{c}') \neq \emptyset\} = \{\mathbf{c}' \in \mathbf{A} : \text{set}(\mathbf{a}) \cap \text{set}(\mathbf{c}') \neq \emptyset\}$. We proceed with the main technical lemma that will enable us to prove Theorem 3.6.

► **Lemma 3.11.** *Let \mathcal{A} be a σ -structure. Let Z be a non-empty set and let f be a mapping $f: \mathbf{A} \rightarrow Z$. Consider $\mathbf{a}, \mathbf{b} \in \mathbf{A}$ with $f(\mathbf{a}) = f(\mathbf{b})$ and $\text{stp}(\mathbf{a}) = \text{stp}(\mathbf{b})$. Let $\pi_S: \mathcal{S}(\mathbf{a}) \rightarrow \mathcal{S}(\mathbf{b})$ be the bijection obtained by Lemma 3.9. The following are equivalent:*

1. $\{\{\text{stp}(\mathbf{a}, \mathbf{c}), f(\mathbf{c})\} : \mathbf{c} \in N(\mathbf{a})\} = \{\{\text{stp}(\mathbf{b}, \mathbf{c}), f(\mathbf{c})\} : \mathbf{c} \in N(\mathbf{b})\}$.
2. For all $\mathbf{s} \in \mathcal{S}(\mathbf{a})$ we have:
 $\{\{\text{stp}(\mathbf{s}, \mathbf{c}), f(\mathbf{c})\} : \mathbf{c} \in \mathcal{S}^{-1}(\mathbf{s})\} = \{\{\text{stp}(\pi_S(\mathbf{s}), \mathbf{d}), f(\mathbf{d})\} : \mathbf{d} \in \mathcal{S}^{-1}(\pi_S(\mathbf{s}))\}$.

The proof makes heavy use of Lemma 3.10 and is combinatorially quite involved; in particular, the proof of direction $2 \Rightarrow 1$ proceeds by an intricate induction that starts with tuples \mathbf{c} in $N(\mathbf{a})$ with $\text{set}(\mathbf{c}) = \text{set}(\mathbf{a})$, and the induction step considers tuples \mathbf{c} in $N(\mathbf{a})$ with a decreasing size of the intersection of $\text{set}(\mathbf{c})$ and $\text{set}(\mathbf{a})$.

The following fact will be helpful for the subsequent proofs.

► **Fact 3.12.** *For all $\mathbf{a}, \mathbf{b} \in \mathbf{A}$ we have*

- (a) $\text{atp}(\mathbf{a}) = \text{atp}(\mathbf{b}) \iff \gamma_0(w_{\mathbf{a}}) = \gamma_0(w_{\mathbf{b}})$.
- (b) For all $\mathbf{s} \in \mathcal{S}(\mathbf{a})$ and $\mathbf{t} \in \mathcal{S}(\mathbf{b})$ we have: $\text{stp}(\mathbf{a}, \mathbf{s}) = \text{stp}(\mathbf{b}, \mathbf{t}) \iff \lambda(w_{\mathbf{a}}, v_{\mathbf{s}}) = \lambda(w_{\mathbf{b}}, v_{\mathbf{t}})$
- (c) For all $\mathbf{s} \in \mathcal{S}(\mathbf{a})$, $\mathbf{t} \in \mathcal{S}(\mathbf{b})$ and for all $\mathbf{c} \in \mathcal{S}^{-1}(\mathbf{s})$, $\mathbf{d} \in \mathcal{S}^{-1}(\mathbf{t})$ we have:
 $\text{stp}(\mathbf{s}, \mathbf{c}) = \text{stp}(\mathbf{t}, \mathbf{d}) \iff \lambda(v_{\mathbf{s}}, w_{\mathbf{c}}) = \lambda(v_{\mathbf{t}}, w_{\mathbf{d}})$.
- (d) For all nodes v of $\mathcal{H}_{\mathcal{A}}$ we have:
 v and $w_{\mathbf{a}}$ are neighbors in $\mathcal{H}_{\mathcal{A}} \iff v = v_{\mathbf{s}}$ for some $\mathbf{s} \in \mathcal{S}(\mathbf{a})$.

The first three statements follow immediately from the definition of $\mathcal{H}_{\mathcal{A}}$, see Definition 3.4. The last statement follows from Remark 3.8 and the definition of $\mathcal{H}_{\mathcal{A}}$. The following lemma relates the color that CR assigns to the node $w_{\mathbf{a}}$ to the colors it assigns to the nodes $v_{\mathbf{s}}$ for the slices \mathbf{s} of \mathbf{a} .

► **Lemma 3.13.** *For all $\mathbf{a}, \mathbf{b} \in \mathbf{A}$ with $\text{stp}(\mathbf{a}) = \text{stp}(\mathbf{b})$ and all $i \in \mathbb{N}_{\geq 1}$ we have:*

- $$\gamma_i(w_{\mathbf{a}}) = \gamma_i(w_{\mathbf{b}}) \iff$$
- (a) $\gamma_{i-1}(w_{\mathbf{a}}) = \gamma_{i-1}(w_{\mathbf{b}})$ and (b) $\gamma_{i-1}(v_{\mathbf{s}}) = \gamma_{i-1}(v_{\mathbf{t}})$, for all $\mathbf{s} \in \mathcal{S}(\mathbf{a})$ and $\mathbf{t} := \pi_S(\mathbf{s})$.
- Here, $\pi_S: \mathcal{S}(\mathbf{a}) \rightarrow \mathcal{S}(\mathbf{b})$ is the bijection from Lemma 3.9.

Finally, we are ready for the proof of Theorem 3.6.

Proof of Theorem 3.6. We proceed by induction on i . For the **induction base** $i=0$, the proof is straightforward by using Fact 3.12 and Lemma 3.9.

For the **inductive Step**, consider an $i \in \mathbb{N}_{\geq 1}$, and let $\mathbf{a}, \mathbf{b} \in \mathbf{A}$.

Induction hypothesis: For all $j < i$ and $\mathbf{c}, \mathbf{d} \in \mathbf{A}$ we have: $\varrho_j(\mathbf{c}) = \varrho_j(\mathbf{d}) \iff \gamma_{2j+1}(w_{\mathbf{c}}) = \gamma_{2j+1}(w_{\mathbf{d}})$.

Induction Claim: $\varrho_i(\mathbf{a}) = \varrho_i(\mathbf{b}) \iff \gamma_{2i+1}(w_{\mathbf{a}}) = \gamma_{2i+1}(w_{\mathbf{b}})$.

In case that $\text{stp}(\mathbf{a}) \neq \text{stp}(\mathbf{b})$, we have $\varrho_0(\mathbf{a}) \neq \varrho_0(\mathbf{b})$, and hence, by the definition of RCR, $\varrho_j(\mathbf{a}) \neq \varrho_j(\mathbf{b})$ holds for all $j \geq 0$. Furthermore, by the induction base, $\varrho_0(\mathbf{a}) \neq \varrho_0(\mathbf{b})$ implies that $\gamma_1(w_{\mathbf{a}}) \neq \gamma_1(w_{\mathbf{b}})$. Hence, by the definition of CR, $\gamma_j(w_{\mathbf{a}}) \neq \gamma_j(w_{\mathbf{b}})$ holds for all $j \geq 1$. This yields: $\varrho_i(\mathbf{a}) \neq \varrho_i(\mathbf{b})$ and $\gamma_{2i+1}(w_{\mathbf{a}}) \neq \gamma_{2i+1}(w_{\mathbf{b}})$, completing the induction step.

In the following, we consider the case where $\text{stp}(\mathbf{a}) = \text{stp}(\mathbf{b})$. From Lemma 3.9 we obtain a bijection $\pi_S: \mathcal{S}(\mathbf{a}) \rightarrow \mathcal{S}(\mathbf{b})$ satisfying $\text{stp}(\mathbf{a}, \mathbf{s}) = \text{stp}(\mathbf{b}, \pi_S(\mathbf{s}))$ for all $\mathbf{s} \in \mathcal{S}(\mathbf{a})$.

If $\varrho_{i-1}(\mathbf{a}) \neq \varrho_{i-1}(\mathbf{b})$, then $\varrho_i(\mathbf{a}) \neq \varrho_i(\mathbf{b})$ by definition of RCR and $\gamma_{2(i-1)+1}(w_{\mathbf{a}}) \neq \gamma_{2(i-1)+1}(w_{\mathbf{b}})$ by induction hypothesis. It follows from the definition of CR that $\gamma_{2i+1}(w_{\mathbf{a}}) \neq \gamma_{2i+1}(w_{\mathbf{b}})$ as well. Thus, from now on we consider the case that $\varrho_{i-1}(\mathbf{a}) = \varrho_{i-1}(\mathbf{b})$.

Using Lemma 3.13 we get that $\gamma_{2i+1}(w_a) = \gamma_{2i+1}(w_b)$ if, and only if, $\gamma_{2i}(w_a) = \gamma_{2i}(w_b)$ and for all $s \in \mathcal{S}(a)$ and $t := \pi_{\mathcal{S}}(s)$ it holds that $\gamma_{2i}(v_s) = \gamma_{2i}(v_t)$. Applying the same lemma again yields that $\gamma_{2i+1}(w_a) = \gamma_{2i+1}(w_b)$ if, and only if $\gamma_{2i-1}(w_a) = \gamma_{2i-1}(w_b)$ and for all $s \in \mathcal{S}(a)$ and $t := \pi_{\mathcal{S}}(s)$ it holds that $\gamma_{2i}(v_s) = \gamma_{2i}(v_t)$ and $\gamma_{2i-1}(v_s) = \gamma_{2i-1}(v_t)$.

Thus, we must show that $\varrho_i(a) = \varrho_i(b)$ if, and only if, $\gamma_{2i-1}(w_a) = \gamma_{2i-1}(w_b)$ and for all $s \in \mathcal{S}(a)$ and $t := \pi_{\mathcal{S}}(s)$ it holds that $\gamma_{2i}(v_s) = \gamma_{2i}(v_t)$ and $\gamma_{2i-1}(v_s) = \gamma_{2i-1}(v_t)$. Recall that we have $\varrho_{i-1}(a) = \varrho_{i-1}(b)$. Since $2i-1 = 2(i-1)+1$, we get that $\gamma_{2i-1}(w_a) = \gamma_{2i-1}(w_b)$ from the induction hypothesis. Hence, it remains to show that $\varrho_i(a) = \varrho_i(b)$ if, and only if, for all $s \in \mathcal{S}(a)$ and $t := \pi_{\mathcal{S}}(s)$ it holds that $\gamma_{2i}(v_s) = \gamma_{2i}(v_t)$ and $\gamma_{2i-1}(v_s) = \gamma_{2i-1}(v_t)$. It is now easy to see that the following two claims finish the proof.

▷ **Claim 3.14.** If $\varrho_i(a) = \varrho_i(b)$, then the following holds for all $s \in \mathcal{S}(a)$ and $t := \pi_{\mathcal{S}}(s)$:
If $\gamma_{2i-1}(v_s) = \gamma_{2i-1}(v_t)$, then $\gamma_{2i}(v_s) = \gamma_{2i}(v_t)$.

▷ **Claim 3.15.** $\varrho_i(a) = \varrho_i(b) \iff$ for all $s \in \mathcal{S}(a)$ and $t := \pi_{\mathcal{S}}(s)$ we have $\gamma_{2i-1}(v_s) = \gamma_{2i-1}(v_t)$.

To prove both claims we use Lemma 3.11, Fact 3.12 and the definitions of CR and RCR. ◀

4 Connection to Homomorphism Counts

This section is devoted to proving the equivalence of statements (1) and (2) of Theorem A (b). I.e., we relate the distinguishing power of RCR to distinguishability via homomorphism counts from *acyclic* σ -structures.

Acyclic σ -structures. Let \mathcal{C} be a σ -structure. A *join-tree* for \mathcal{C} is a tree (i.e., an undirected, simple graph that is connected and acyclic) J with vertex set $V(J) := \mathbf{C}$ (i.e., the tuples in \mathbf{C} serve as vertices of J) and which satisfies the following connectivity condition: for all $v \in V(\mathcal{C})$ the set $\{c \in V(J) : v \in \text{set}(c)\}$ induces a connected subgraph of J ; we will denote this subgraph (which in fact is a tree) by J_v .

We call a σ -structure \mathcal{C} *acyclic* if there exists a join-tree for \mathcal{C} . This definition of acyclicity of σ -structures is equivalent to acyclicity as defined in the textbook [1], it is equivalent to the notion of *alpha-acyclicity* as defined in [4, 6] and, finally, is also equivalent to \mathcal{C} having (generalized or fractional) hypertree width 1 as defined in [17, 18, 24]. In the literature, also other notions of acyclicity for relational structures (and hypergraphs) have been considered; but alpha-acyclicity arguably is the most common and the least restrictive one. Consult [7] for a detailed survey on this topic.

For the special case of *binary* signatures $\hat{\sigma}$, i.e., where $\hat{\sigma}$ -structures are colored multigraphs, it is known [7] that a $\hat{\sigma}$ -structure \mathcal{C} is acyclic if, and only if, its Gaifman graph is acyclic (w.r.t. the usual notion of acyclicity of undirected simple graphs). It is well-known that for non-binary signatures σ there exist acyclic σ -structures whose Gaifman graph is not acyclic.

Homomorphisms. A homomorphism from a σ -structure \mathcal{C} to a σ -structure \mathcal{A} is a mapping $h: V(\mathcal{C}) \rightarrow V(\mathcal{A})$ such that for all $R \in \sigma$, for $k := \text{ar}(R)$, and all $c = (c_1, \dots, c_k) \in R^{\mathcal{C}}$ we have $(h(c_1), \dots, h(c_k)) \in R^{\mathcal{A}}$. We write $\text{Hom}(\mathcal{C}, \mathcal{A})$ for the set of homomorphisms from \mathcal{C} to \mathcal{A} , and we let $\text{hom}(\mathcal{C}, \mathcal{A}) := |\text{Hom}(\mathcal{C}, \mathcal{A})|$ denote the number of homomorphisms from \mathcal{C} to \mathcal{A} .

The remainder of this section is dedicated to proving the equivalence of statements (1) and (2) of Theorem A (b), i.e., proving the following theorem.

► **Theorem C.** *For all σ -structures \mathcal{A} and \mathcal{B} , the following statements are equivalent.*

1. *RCR distinguishes \mathcal{A} and \mathcal{B} .*
2. *There exists an acyclic and connected σ -structure \mathcal{C} such that $\text{hom}(\mathcal{C}, \mathcal{A}) \neq \text{hom}(\mathcal{C}, \mathcal{B})$.*

This can be viewed as a generalization of the following result by Dvořák [15] and Dell, Grohe, Rattan [13] to arbitrary signatures σ . While [15, 13] state the theorem just for graphs, it easily extends to colored multigraphs (as noted in [13]). A *colored multitree* is an acyclic and connected colored multigraph, i.e., a colored multigraph whose Gaifman graph is a tree.

► **Theorem 4.1** ([15, 13]). *Let \mathcal{G} and \mathcal{H} be colored multigraphs. The following statements are equivalent.*

1. *CR distinguishes \mathcal{G} and \mathcal{H} .*
2. *There exists a colored multitree \mathcal{T} such that $\text{hom}(\mathcal{T}, \mathcal{G}) \neq \text{hom}(\mathcal{T}, \mathcal{H})$.*

Theorem 4.1 will serve as the first key ingredient of our proof of Theorem C. The second key ingredient is to use the following notion of a *join-tree representation* \mathcal{G}_C^J . Recall from Definition 3.1 the binary signature $\hat{\sigma} := \{E_{i,j} : i, j \in [\text{ar}(\sigma)]\} \cup \{U_R : R \in \sigma\}$ and the colored multigraph \mathcal{G}_A of signature $\hat{\sigma}$ that represents a σ -structure \mathcal{A} . For an acyclic σ -structure \mathcal{C} and a join-tree J for \mathcal{C} we define the colored multigraph \mathcal{G}_C^J of signature $\hat{\sigma}$ to have universe $V(\mathcal{G}_C^J) := \{v_c : c \in \mathcal{C}\}$ where v_c is a new vertex for each tuple $c \in \mathcal{C}$, and

$$(U_R)^{\mathcal{G}_C^J} := \{v_c : c \in R^C\} \text{ and } (E_{i,j})^{\mathcal{G}_C^J} := \{(v_b, v_c) : \{b, c\} \in E(J) \text{ and } (i, j) \in \text{stp}(b, c)\}$$

for all $R \in \sigma$ and all $i, j \in [\text{ar}(\sigma)]$. Via identifying c and v_c for all $c \in \mathcal{C}$, the Gaifman graph of \mathcal{G}_C^J is isomorphic to a subgraph of J . Thus, since J is a tree, \mathcal{G}_C^J is acyclic.

The last two ingredients for our proof of Theorem C are the following lemmas:

► **Lemma 4.2.** *For σ -structures \mathcal{A} and \mathcal{C} and any join-tree J for \mathcal{C} we have:*
 $\text{hom}(\mathcal{C}, \mathcal{A}) = \text{hom}(\mathcal{G}_C^J, \mathcal{G}_A)$.

► **Lemma 4.3.** *Let \mathcal{A} and \mathcal{B} be σ -structures, and let \mathcal{T} be a colored multitree of signature $\hat{\sigma}$ such that $\text{hom}(\mathcal{T}, \mathcal{G}_A) \neq \text{hom}(\mathcal{T}, \mathcal{G}_B)$. There exists an acyclic and connected σ -structure \mathcal{C} and a join-tree J for \mathcal{C} such that $\text{hom}(\mathcal{G}_C^J, \mathcal{G}_A) \neq \text{hom}(\mathcal{G}_C^J, \mathcal{G}_B)$.*

Before proving these lemmas, we first show how to use the four key ingredients for proving Theorem C.

Proof of Theorem C. As pointed out in Section 3.2, running RCR on a σ -structure \mathcal{A} produces a stable coloring that is equivalent (via identifying $a \in \mathcal{A}$ with $w_a \in V(\mathcal{G}_A)$) to the stable coloring produced by the classical CR on the colored multigraph \mathcal{G}_A . Thus, RCR distinguishes the σ -structures \mathcal{A} and \mathcal{B} if, and only if, classical CR distinguishes \mathcal{G}_A and \mathcal{G}_B . According to Theorem 4.1 the latter is the case if, and only if, there is a colored multitree \mathcal{T} with $\text{hom}(\mathcal{T}, \mathcal{G}_A) \neq \text{hom}(\mathcal{T}, \mathcal{G}_B)$.

Hence, for the direction “1 \Rightarrow 2”, if RCR distinguishes \mathcal{A} and \mathcal{B} , then there exists a colored multitree \mathcal{T} with $\text{hom}(\mathcal{T}, \mathcal{G}_A) \neq \text{hom}(\mathcal{T}, \mathcal{G}_B)$. By Lemma 4.3, there also exists an acyclic and connected σ -structure \mathcal{C} and a join-tree J for \mathcal{C} such that $\text{hom}(\mathcal{G}_C^J, \mathcal{G}_A) \neq \text{hom}(\mathcal{G}_C^J, \mathcal{G}_B)$. According to Lemma 4.2, this implies that $\text{hom}(\mathcal{C}, \mathcal{A}) \neq \text{hom}(\mathcal{C}, \mathcal{B})$.

For the direction “2 \Rightarrow 1”, if there exists an acyclic and connected σ -structure \mathcal{C} such that $\text{hom}(\mathcal{C}, \mathcal{A}) \neq \text{hom}(\mathcal{C}, \mathcal{B})$, then according to Lemma 4.2 we have $\text{hom}(\mathcal{G}_C^J, \mathcal{G}_A) \neq \text{hom}(\mathcal{G}_C^J, \mathcal{G}_B)$, for any join-tree J for \mathcal{C} . By construction, the Gaifman graph of \mathcal{G}_C^J is a subgraph of J . In fact, the Gaifman graph of \mathcal{C} being connected implies that the Gaifman graph of \mathcal{G}_C^J is exactly J . Hence, \mathcal{G}_C^J is a colored multitree. Thus, as pointed out in the first paragraph of the proof, RCR distinguishes \mathcal{A} and \mathcal{B} . ◀

The remainder of this section is devoted to proving the Lemmas 4.2 and 4.3. The following notation will be convenient. If f is a mapping from set V to a set V' , and $\mathbf{a} = (a_1, \dots, a_k)$ is a tuple in V^k for some $k \in \mathbb{N}_{\geq 1}$, then we write $f(\mathbf{a})$ for the tuple $(f(a_1), \dots, f(a_k))$.

Proof of Lemma 4.2. We prove the lemma by providing a bijection π between $\text{Hom}(\mathcal{C}, \mathcal{A})$ and $\text{Hom}(\mathcal{G}_{\mathcal{C}}^J, \mathcal{G}_{\mathcal{A}})$. Recall that $V(\mathcal{G}_{\mathcal{C}}^J) = \{v_{\mathbf{c}} : \mathbf{c} \in \mathcal{C}\}$ and $V(\mathcal{G}_{\mathcal{A}}) = \{w_{\mathbf{a}} : \mathbf{a} \in \mathcal{A}\}$.

For all $h \in \text{Hom}(\mathcal{C}, \mathcal{A})$ let $\pi(h) := h'$, where h' is defined by $h'(v_{\mathbf{c}}) := w_{h(\mathbf{c})}$ for all $\mathbf{c} \in \mathcal{C}$. Since $\mathbf{c} \in \mathcal{C}$ and $h \in \text{Hom}(\mathcal{C}, \mathcal{A})$, we obtain that $h(\mathbf{c}) \in \mathcal{A}$. Hence, $h'(v_{\mathbf{c}}) = w_{h(\mathbf{c})} \in V(\mathcal{G}_{\mathcal{A}})$. To prove the lemma, it suffices to verify that: (a) $\text{img}(\pi) \subseteq \text{Hom}(\mathcal{G}_{\mathcal{C}}^J, \mathcal{G}_{\mathcal{A}})$, i.e., $\pi(h)$ is a homomorphism for every $h \in \text{Hom}(\mathcal{C}, \mathcal{A})$; (b) π is injective; and (c) π is surjective. The proofs of statements (a) and (b) are straightforward. For the proof of statement (c) let $h'' \in \text{Hom}(\mathcal{G}_{\mathcal{C}}^J, \mathcal{G}_{\mathcal{A}})$. Our aim is to find an $h \in \text{Hom}(\mathcal{C}, \mathcal{A})$ such that $h'' = \pi(h)$. By definition of \mathcal{C} (recall our assumption on σ -structures described in Section 2), for every $z \in V(\mathcal{C})$ there exists an $R \in \sigma$ and a tuple $\mathbf{c} \in R^{\mathcal{C}}$ such that $z \in \text{set}(\mathbf{c})$. For each $z \in V(\mathcal{C})$ let us choose arbitrary, but from now on fixed such R and \mathbf{c} which we henceforth will denote by R_z and \mathbf{c}_z , and let us fix an $i_z \in [\text{ar}(R_z)]$ such that z is the i_z -th component of the tuple \mathbf{c}_z . Since $\mathbf{c}_z \in (R_z)^{\mathcal{C}}$, by definition of $\mathcal{G}_{\mathcal{C}}^J$ we have $v_{\mathbf{c}_z} \in (U_{R_z})^{\mathcal{G}_{\mathcal{C}}^J}$. Since $h'' \in \text{Hom}(\mathcal{G}_{\mathcal{C}}^J, \mathcal{G}_{\mathcal{A}})$, we obtain $h''(v_{\mathbf{c}_z}) \in (U_{R_z})^{\mathcal{G}_{\mathcal{A}}}$. By definition of $\mathcal{G}_{\mathcal{A}}$ there is a tuple $\mathbf{a}_z \in (R_z)^{\mathcal{A}}$ such that $h''(v_{\mathbf{c}_z}) = w_{\mathbf{a}_z}$. Let us write x_z to denote the i_z -th component of the tuple \mathbf{a}_z . Clearly, $x_z \in V(\mathcal{A})$. Using these notions, we define the mapping $h: V(\mathcal{C}) \rightarrow V(\mathcal{A})$ by letting $h(z) := x_z$ for every $z \in V(\mathcal{C})$.

Claim 1: For all $\mathbf{c} \in \mathcal{C}$ and $\mathbf{a} \in \mathcal{A}$ with $h''(v_{\mathbf{c}}) = w_{\mathbf{a}}$ we have: $h(\mathbf{c}) = \mathbf{a}$.

Claim 2: $h \in \text{Hom}(\mathcal{C}, \mathcal{A})$.

See the paper's full version for proofs of both claims. To complete the proof of Lemma 4.2 it suffices to show that $\pi(h) = h''$. By definition of π we have: $\pi(h) = h'$, where h' is defined by $h'(v_{\mathbf{c}}) := w_{h(\mathbf{c})}$ for all $\mathbf{c} \in \mathcal{C}$. From Claim 1 we obtain that $h''(v_{\mathbf{c}}) = h'(v_{\mathbf{c}})$ for all $\mathbf{c} \in \mathcal{C}$. Hence, $h'' = h' = \pi(h)$. This completes the proof of statement (c) and the proof of Lemma 4.2. \blacktriangleleft

Proof of Lemma 4.3. By assumption, $\text{hom}(\mathcal{T}, \mathcal{G}_{\mathcal{A}}) \neq \text{hom}(\mathcal{T}, \mathcal{G}_{\mathcal{B}})$, for a colored multitree \mathcal{T} of signature $\hat{\sigma}$ and σ -structures \mathcal{A}, \mathcal{B} . We will show that the homomorphisms from \mathcal{T} provide us with “templates” for acyclic connected σ -structures, one of which must have a number of homomorphisms into \mathcal{A} that is different from its number of homomorphisms into \mathcal{B} .

Let us write T to denote the Gaifman graph of \mathcal{T} . Since \mathcal{T} is a colored multitree, T is a tree. For an $h \in \text{Hom}(\mathcal{T}, \mathcal{G}_{\mathcal{A}})$ the *print* P_h of h in $\mathcal{G}_{\mathcal{A}}$ is the colored multigraph of signature $\hat{\sigma}$ defined by $V(P_h) := V(\mathcal{T}) = V(T)$ and, for all $R \in \sigma$ and all $i, j \in [\text{ar}(\sigma)]$:

$$\begin{aligned} (U_R)^{P_h} &:= \{v \in V(T) : h(v) \in (U_R)^{\mathcal{G}_{\mathcal{A}}}\}, \\ (E_{i,j})^{P_h} &:= \{(u, v) : (h(u), h(v)) \in (E_{i,j})^{\mathcal{G}_{\mathcal{A}}} \text{ and } (u=v \text{ or } \{u, v\} \in E(T))\}. \end{aligned}$$

Note that T is also the Gaifman graph of P_h . We let $P_{\mathcal{A}} = \{P_h : h \in \text{Hom}(\mathcal{T}, \mathcal{G}_{\mathcal{A}})\}$. The notion of the print P_h of h in $\mathcal{G}_{\mathcal{B}}$ for $h \in \text{Hom}(\mathcal{T}, \mathcal{G}_{\mathcal{B}})$ and the set $P_{\mathcal{B}}$ are defined analogously. Note that $P_{\mathcal{A}}$ and $P_{\mathcal{B}}$ are not necessarily disjoint, and that different homomorphisms may have the same print.

For every $P \in P_{\mathcal{A}} \cup P_{\mathcal{B}}$ we let $\#(P, \mathcal{A}) := |\{h \in \text{Hom}(\mathcal{T}, \mathcal{G}_{\mathcal{A}}) : P_h = P\}|$. The number $\#(P, \mathcal{B})$ is defined analogously. Note that $\text{hom}(\mathcal{T}, \mathcal{G}_{\mathcal{A}}) = \sum_{P \in P_{\mathcal{A}}} \#(P, \mathcal{A})$ and $\text{hom}(\mathcal{T}, \mathcal{G}_{\mathcal{B}}) = \sum_{P \in P_{\mathcal{B}}} \#(P, \mathcal{B})$ – see the paper's full version for a proof.

For any two prints P, P' we say that P is a *subprint* of P' (for short: $P \preceq P'$) if $(U_R)^P \subseteq (U_R)^{P'}$ and $(E_{i,j})^P \subseteq (E_{i,j})^{P'}$, for all $R \in \sigma$ and all $i, j \in [\text{ar}(\sigma)]$. Obviously, \preceq is a partial order on $P_{\mathcal{A}} \cup P_{\mathcal{B}}$.

It can be verified that for every print P we have $\text{hom}(P, \mathcal{G}_A) = \sum_{P': P \preceq P'} \#(P', A)$ and $\text{hom}(P, \mathcal{G}_B) = \sum_{P': P \preceq P'} \#(P', B)$.

Since $\text{hom}(\mathcal{T}, \mathcal{G}_A) = \sum_{P \in P_A} \#(P, A)$ and $\text{hom}(\mathcal{T}, \mathcal{G}_B) = \sum_{P \in P_B} \#(P, B)$ and, by assumption, $\text{hom}(\mathcal{T}, \mathcal{G}_A) \neq \text{hom}(\mathcal{T}, \mathcal{G}_B)$, there must be a $P \in P_A \cup P_B$ such that $\#(P, A) \neq \#(P, B)$. We choose a largest such P w.r.t. the partial order \preceq . I.e., $\#(P, A) \neq \#(P, B)$, but $\#(P', A) = \#(P', B)$ for all P' with $P \preceq P'$ and $P' \neq P$. Combining this with the fact that $\text{hom}(P, \mathcal{G}_A) = \sum_{P': P \preceq P'} \#(P', A)$ and $\text{hom}(P, \mathcal{G}_B) = \sum_{P': P \preceq P'} \#(P', B)$, we obtain: $\text{hom}(P, \mathcal{G}_A) \neq \text{hom}(P, \mathcal{G}_B)$.

Now, all that remains to be done is to show that there exists an acyclic and connected σ -structure \mathcal{C} and a join-tree J for \mathcal{C} (J will have exactly the same shape as T) such that \mathcal{G}_C^J is isomorphic to P – then $\text{hom}(P, \mathcal{G}_A) \neq \text{hom}(P, \mathcal{G}_B)$ implies that $\text{hom}(\mathcal{G}_C^J, \mathcal{G}_A) \neq \text{hom}(\mathcal{G}_C^J, \mathcal{G}_B)$. Details on how to construct \mathcal{C} are given in the paper’s full version. This then completes the proof of Lemma 4.3. \blacktriangleleft

5 Connection to Logic

This section’s goal is to provide a logical characterization of the distinguishing power of RCR. We aim for a theorem that is analogous to the following result due to Immerman and Lander [26] and Cai, Fürer, Immerman [10] concerning the logic \mathcal{C} , a syntactic extension of first-order logic with counting quantifiers of the form $\exists^{\geq n} x \psi$ (for every fixed $n \in \mathbb{N}_{\geq 1}$), expressing “there exist at least n values for x such that ψ holds”. The restriction of \mathcal{C} to two variables is denoted by \mathcal{C}^2 .

► **Theorem 5.1** ([10, 26]). *Let G and H be graphs. The following statements are equivalent:*

1. *CR distinguishes G and H .*
2. *There exists a sentence $\varphi \in \mathcal{C}^2$ such that $G \models \varphi$ and $H \not\models \varphi$.*

5.1 The Guarded Fragment of Counting Logic

This section introduces the *guarded fragment of the logic* \mathcal{C} , for short: $\text{GF}(\mathcal{C})$. Its definition is in the same spirit as the logic GF (the guarded fragment of first-order logic; see [25]) and the logic $\text{GF}(\mathcal{L})$ (the guarded fragment of any logic \mathcal{L} that is a subset of first-order logic; see [18]). Here, we use a similar notation as in [18], but adapt it in order to obtain a reasonable notion of “guarded fragment of \mathcal{C} ”. As in [25, 18], the term “guarded” refers to the fact that quantifiers are appropriately relativized by relational atoms.

We have available a countably infinite set $\text{Var} := \{v_i : i \in \mathbb{N}_{\geq 1}\}$ of *variables*. We call a tuple \mathbf{v} of m distinct variables of the form $(v_{i_1}, \dots, v_{i_m}) \in \text{Var}^m$ a *variable tuple*, and we let $\text{vars}(\mathbf{v}) := \{v_{i_1}, \dots, v_{i_m}\}$. Recall that at the beginning of Section 3 we have chosen an arbitrary (relational) signature that is fixed throughout the rest of this paper.

► **Definition 5.2** (Syntax of $\text{GF}(\mathcal{C})$). The logic $\text{GF}(\mathcal{C})$ is inductively defined along with the *free variables* and the *guard-depth*, formalized by the functions $\text{free}: \text{GF}(\mathcal{C}) \rightarrow 2^{\text{Var}}$ and $\text{gd}: \text{GF}(\mathcal{C}) \rightarrow \mathbb{N}$.

Atomic Formulas: For all $R \in \sigma$ with $\ell := \text{ar}(R)$, all $x_1, \dots, x_\ell \in \text{Var}$ and all $x, y \in \text{Var}$, the following formulas φ (of signature σ) are in $\text{GF}(\mathcal{C})$: φ is of the form

1. $R(x_1, \dots, x_\ell)$ with $\text{free}(\varphi) := \{x_1, \dots, x_\ell\}$ and $\text{gd}(\varphi) := 0$;
2. $x = y$ with $\text{free}(\varphi) := \{x, y\}$ and $\text{gd}(\varphi) := 0$.

Inductive Rules: Let χ, ψ be formulas (of signature σ) in $\text{GF}(\mathcal{C})$. The following formulas φ (of signature σ) are in $\text{GF}(\mathcal{C})$: φ is of the form

3. $\neg \chi$ with $\text{free}(\varphi) := \text{free}(\chi)$ and $\text{gd}(\varphi) := \text{gd}(\chi)$;
4. $(\chi \wedge \psi)$ with $\text{free}(\varphi) := \text{free}(\chi) \cup \text{free}(\psi)$ and $\text{gd}(\varphi) := \max(\text{gd}(\chi), \text{gd}(\psi))$.

An atomic formula Δ (of signature σ) of the form $R(x_1, \dots, x_\ell)$ in $\text{GF}(\mathbf{C})$ is called a *guard* for ψ , if $\text{free}(\psi) \subseteq \text{free}(\Delta)$. Let $n \in \mathbb{N}_{\geq 1}$ and let Δ be a guard for ψ . For every *variable tuple* \mathbf{v} with $\text{vars}(\mathbf{v}) \subseteq \text{free}(\Delta)$, the following formula φ (of signature σ) is in $\text{GF}(\mathbf{C})$: φ is of the form

$$5. \quad \exists^{\geq n} \mathbf{v}. (\Delta \wedge \psi) \quad \text{with } \text{free}(\varphi) := \text{free}(\Delta) \setminus \text{vars}(\mathbf{v}) \text{ and } \text{gd}(\varphi) := \text{gd}(\psi) + 1.$$

In this paper we assume w.l.o.g. that the variable tuple $\mathbf{v} = (v_{i_1}, \dots, v_{i_m})$ after a quantifier $\exists^{\geq n}$ is *ordered*, i.e., $i_1 < \dots < i_m$. This has no effect on the semantics, but simplifies some arguments. We write $\exists^{\geq n} \mathbf{v}. (\Delta \wedge \varphi)$ as shorthand for $(\exists^{\geq n} \mathbf{v}. (\Delta \wedge \varphi) \wedge \neg \exists^{\geq n+1} \mathbf{v}. (\Delta \wedge \varphi))$. We omit parentheses in the usual way.

► **Definition 5.3** (Semantics of $\text{GF}(\mathbf{C})$). A σ -*interpretation* is a tuple $\mathcal{I} = (\mathcal{A}, \alpha)$ consisting of a σ -structure \mathcal{A} and a function $\alpha: \text{Var} \rightarrow V(\mathcal{A})$. Formulas (of signature σ) in $\text{GF}(\mathbf{C})$ are evaluated on σ -interpretations \mathcal{I} . We write $\mathcal{I} \models \varphi$ to denote that \mathcal{I} *satisfies* φ , and $\mathcal{I} \not\models \varphi$ to denote that \mathcal{I} does not satisfy φ . By $\mathcal{I}_{\frac{(a_1, \dots, a_\ell)}{(v_{i_1}, \dots, v_{i_\ell})}}$ we denote the σ -interpretation (\mathcal{A}, α') with $\alpha'(v_{i_j}) := a_j$ for all $j \in [\ell]$, and $\alpha'(x) := \alpha(x)$ for all $x \in \text{Var} \setminus \{v_{i_1}, \dots, v_{i_\ell}\}$. The semantics of formulas in $\text{GF}(\mathbf{C})$ are inductively defined as follows:

Rule 1: $\mathcal{I} \models R(x_1, \dots, x_\ell) \iff (\alpha(x_1), \dots, \alpha(x_\ell)) \in R^{\mathcal{A}}$.

Rule 2: $\mathcal{I} \models x = y \iff \alpha(x) = \alpha(y)$.

Rules 3 and 4: $\mathcal{I} \models \neg \chi \iff \mathcal{I} \not\models \chi$. $\mathcal{I} \models (\chi \wedge \psi) \iff \mathcal{I} \models \chi$ and $\mathcal{I} \models \psi$.

Rule 5: $\mathcal{I} \models \exists^{\geq n} \mathbf{v}. (\Delta \wedge \psi) \iff$ there are at least n tuples $\mathbf{a} \in V(\mathcal{A})^{\text{ar}(\mathbf{v})}$ such that $\mathcal{I}_{\frac{\mathbf{a}}{\mathbf{v}}} \models (\Delta \wedge \psi)$.

We will use the following conventions throughout the paper: $\varphi(x_1, \dots, x_k)$ denotes that $\text{free}(\varphi) \subseteq \{x_1, \dots, x_k\}$; and $\mathcal{A}, (a_1, \dots, a_k) \models \varphi(x_1, \dots, x_k)$ denotes that $(\mathcal{A}, \alpha) \models \varphi$ where α is an assignment where $\alpha(x_i) = a_i$ holds for all $i \in [k]$. A *sentence* is a formula $\varphi \in \text{GF}(\mathbf{C})$ that has no free variable, i.e., $\text{free}(\varphi) = \emptyset$. If φ is a sentence, we write $\mathcal{A} \models \varphi$ to denote that $(\mathcal{A}, \alpha) \models \varphi$ for *any* assignment α (since α does not matter in this case). We write $\mathcal{A} \equiv_{\text{GF}(\mathbf{C})} \mathcal{B}$ to denote that the σ -structures \mathcal{A} and \mathcal{B} satisfy the same sentences (of signature σ) in $\text{GF}(\mathbf{C})$. Finally, we say that \mathcal{A} and \mathcal{B} are *distinguishable* in $\text{GF}(\mathbf{C})$ if $\mathcal{A} \not\equiv_{\text{GF}(\mathbf{C})} \mathcal{B}$.

► **Example 5.4.** Consider the formula $\varphi :=$

$$\exists^{\geq 1} (v_1, v_2, v_3, v_4, v_5, v_6) \cdot (R(v_1, v_2, v_3, v_4, v_5, v_6) \wedge (E(v_1, v_2) \wedge (E(v_2, v_3) \wedge E(v_3, v_1))))),$$

the σ_1 -structure \mathcal{A}_1 from Example 3.2 and the σ_1 -structure \mathcal{B}_1 with $V(\mathcal{B}_1) = \{1, 2, 3, u, v, w\}$, $E^{\mathcal{B}_1} := \{(1, 2), (2, w), (w, u), (u, v), (v, 3), (3, 1)\}$ and $R^{\mathcal{B}_1} := \{(1, 2, 3, u, v, w)\}$.

Clearly, $\psi := (E(v_1, v_2) \wedge (E(v_2, v_3) \wedge E(v_3, v_1)))$ is a formula in $\text{GF}(\mathbf{C})$ with $\text{free}(\psi) = \{v_1, v_2, v_3\}$ and $\text{gd}(\psi) = 0$. Thus, $R(v_1, v_2, v_3, v_4, v_5, v_6)$ is a guard for ψ . Hence, $\varphi \in \text{GF}(\mathbf{C})$ with $\text{free}(\varphi) = \emptyset$, and $\text{gd}(\varphi) = 1$.

The formula φ states that there is at least one tuple (of arity 6) in R such that the first 3 entries of this tuple form a triangle w.r.t. relation E . Hence, $\mathcal{A}_1 \models \varphi$ and $\mathcal{B}_1 \not\models \varphi$, which means \mathcal{A}_1 and \mathcal{B}_1 are distinguishable in $\text{GF}(\mathbf{C})$.

5.2 The Guarded Game

Our ultimate goal in Section 5 is to prove for any two σ -structures \mathcal{A} and \mathcal{B} that RCR distinguishes \mathcal{A} and \mathcal{B} if, and only if, $\mathcal{A} \not\equiv_{\text{GF}(\mathbf{C})} \mathcal{B}$. Similarly to the proof of Theorem 5.1, our proof will use, as an intermediate step, a game characterization of (in)distinguishability of two σ -structures in $\text{GF}(\mathbf{C})$. We call this game the *Guarded Game*; it is defined as follows. It is played on two σ -structures \mathcal{A}, \mathcal{B} . A *configuration* of the Guarded Game is a tuple of the form

$((\mathcal{A}, \mathbf{a}), (\mathcal{B}, \mathbf{b}))$, where \mathcal{A} and \mathcal{B} are the given σ -structures and $\mathbf{a} \in V(\mathcal{A})^k$, $\mathbf{b} \in V(\mathcal{B})^k$ for some $k \in \mathbb{N}$. A configuration $((\mathcal{A}, \mathbf{a}), (\mathcal{B}, \mathbf{b}))$ is called *distinguishing*, if $\text{stp}(\mathbf{a}) \neq \text{stp}(\mathbf{b})$ or there are an $\ell \in [\text{ar}(\sigma)]$ and indices $i_1, \dots, i_\ell \in [k]$ such that $\text{atp}((a_{i_1}, \dots, a_{i_\ell})) \neq \text{atp}((b_{i_1}, \dots, b_{i_\ell}))$. We may omit parentheses if they are clear from the context. If $k = 0$, we write \mathcal{A}, \mathcal{B} for the configuration $((\mathcal{A}, ()), (\mathcal{B}, ()))$, and we call this the *empty* configuration; note that this configuration is *not* distinguishing.

A *round* of the Guarded Game is played as follows: consider $(\mathcal{A}, \mathbf{a}), (\mathcal{B}, \mathbf{b})$ to be the configuration at the beginning of the round. Spoiler picks a relation symbol $R \in \sigma$. Then, Duplicator provides a bijection π between $R^{\mathcal{A}}$ and $R^{\mathcal{B}}$. If no such bijection exists (i.e., $|R^{\mathcal{A}}| \neq |R^{\mathcal{B}}|$), the round ends and Spoiler wins this round; otherwise the round proceeds as follows. Spoiler picks some $\mathbf{a}' \in R^{\mathcal{A}}$ and creates the new configuration $(\mathcal{A}, \mathbf{a}'), (\mathcal{B}, \mathbf{b}')$ where $\mathbf{b}' := \pi(\mathbf{a}')$. Duplicator wins this round if the new configuration is not distinguishing and $\text{stp}(\mathbf{a}, \mathbf{a}') = \text{stp}(\mathbf{b}, \mathbf{b}')$. Otherwise, Spoiler wins this round.

Duplicator has a 0-round *winning strategy* on $(\mathcal{A}, \mathbf{a}), (\mathcal{B}, \mathbf{b})$ if the configuration is not distinguishing. For $i \geq 1$, Duplicator has an i -round winning strategy on $(\mathcal{A}, \mathbf{a}), (\mathcal{B}, \mathbf{b})$ if this configuration is not distinguishing, and she can provide a bijection π for every $R \in \sigma$ that Spoiler may pick, such that for every \mathbf{a}' and $\mathbf{b}' := \pi(\mathbf{a}')$ that Spoiler may choose, she wins the current round and has an $(i-1)$ -round winning strategy on the resulting configuration $(\mathcal{A}, \mathbf{a}'), (\mathcal{B}, \mathbf{b}')$. Spoiler has an i -round winning-strategy on $(\mathcal{A}, \mathbf{a}), (\mathcal{B}, \mathbf{b})$, if Duplicator does not have one. In particular, if Spoiler has a winning strategy for i rounds, he also has a winning strategy for more than i rounds. If \mathcal{A} and \mathcal{B} are not of strictly equal size, then Spoiler has a trivial 1-round winning strategy, because Duplicator is unable to give a bijection in the first round. We say that Duplicator *wins* the Guarded Game on $(\mathcal{A}, \mathbf{a}), (\mathcal{B}, \mathbf{b})$ if she has an i -round winning strategy on $(\mathcal{A}, \mathbf{a}), (\mathcal{B}, \mathbf{b})$ for every $i \in \mathbb{N}$.

5.3 RCR is equivalent to GF(C) and the Guarded Game

This section is devoted to proving the equivalence of statements (1), (3) and (4) of Theorem A (b), i.e., we prove the following theorem.

► **Theorem D.** *For all σ -structures \mathcal{A} and \mathcal{B} , the following statements are equivalent.*

1. *RCR distinguishes \mathcal{A} and \mathcal{B} .*
2. *There exists a sentence $\varphi \in \text{GF}(\text{C})$ such that $\mathcal{A} \models \varphi$ and $\mathcal{B} \not\models \varphi$.*
3. *Spoiler wins the Guarded Game on \mathcal{A}, \mathcal{B} .*

We prove the theorem by showing that the implication chain $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1$ holds. For this, we use the following three lemmas; their proofs are inductive and quite similar to the way the analogous result on graphs is shown, thus we defer their proofs to the paper's full version. The *arity* $\text{ar}(c)$ (atomic type $\text{atp}(c)$, similarity type $\text{stp}(c)$) of a color $c \in \text{RC}_i(\mathcal{A})$ is defined as the arity (atomic type, similarity type) of the tuples in \mathcal{A} that receive this color. Recall that we denote the color of a tuple $\mathbf{a} \in \mathcal{A}$ after i iterations of RCR as $\varrho_i(\mathbf{a})$.

► **Lemma 5.5.** *For every σ -structure \mathcal{A} , every $i \in \mathbb{N}$, and every $c \in \text{RC}_i(\mathcal{A})$ of arity k , there exists a formula $\varphi_c^i(\mathbf{x}) \in \text{GF}(\text{C})$ with $\mathbf{x} = (x_1, \dots, x_k)$ such that for every σ -structure \mathcal{B} of size strictly equal to \mathcal{A} and every $\mathbf{b} \in \mathcal{B}$ of arity k we have: $\mathcal{B}, \mathbf{b} \models \varphi_c^i(\mathbf{x}) \iff \varrho_i(\mathbf{b}) = c$.*

► **Lemma 5.6.** *Let \mathcal{A} and \mathcal{B} be σ -structures of strictly equal size and let $\mathbf{a} \in V(\mathcal{A})^k$, $\mathbf{b} \in V(\mathcal{B})^k$ be arbitrary tuples of arity k . Let $\mathbf{x} = (x_1, \dots, x_k)$ be a tuple of k distinct variables. If there exists a formula $\varphi \in \text{GF}(\text{C})$ with $\text{free}(\varphi) \subseteq \{x_1, \dots, x_k\}$ such that $\mathcal{A}, \mathbf{a} \models \varphi(\mathbf{x}) \iff \mathcal{B}, \mathbf{b} \not\models \varphi(\mathbf{x})$, then Spoiler has a $\text{gd}(\varphi)$ -round winning strategy for the Guarded Game on $(\mathcal{A}, \mathbf{a}), (\mathcal{B}, \mathbf{b})$.*

► **Lemma 5.7.** *Let $i \in \mathbb{N}$, let \mathcal{A} and \mathcal{B} be σ -structures of strictly equal size, and let $\mathbf{a} \in \mathbf{A}$, $\mathbf{b} \in \mathbf{B}$ be tuples of arity k . If $\varrho_1(\mathbf{a}) = \varrho_1(\mathbf{b})$, then the configuration $(\mathcal{A}, \mathbf{a})$, $(\mathcal{B}, \mathbf{b})$ is not distinguishing. Further, if $\varrho_{i+1}(\mathbf{a}) = \varrho_{i+1}(\mathbf{b})$ and $\text{mult}_{\mathcal{A}}(c) = \text{mult}_{\mathcal{B}}(c)$ for all $c \in \text{RC}_i(\mathcal{A}) \cup \text{RC}_i(\mathcal{B})$, then Duplicator has an i -round winning strategy for the Guarded Game on $(\mathcal{A}, \mathbf{a})$, $(\mathcal{B}, \mathbf{b})$.*

The proof of Theorem D proceeds as follows. If \mathcal{A} and \mathcal{B} are *not* of strictly equal size, it is straightforward to see that each of the theorem's three statements is fulfilled. For the case where \mathcal{A} and \mathcal{B} are of strictly equal size, “ $1 \Rightarrow 2$ ” easily follows from Lemma 5.5, and “ $2 \Rightarrow 3$ ” is obtained from Lemma 5.6. Concerning “ $3 \Rightarrow 1$ ”, one proves the contraposition and uses Lemma 5.7 to obtain a winning strategy for Duplicator in the Guarded Game; this winning strategy ensures that after each round, the configuration is of the form $(\mathcal{A}, \mathbf{a})$, $(\mathcal{B}, \mathbf{b})$ where \mathbf{a} and \mathbf{b} have the same color in the stable coloring produced by RCR on \mathcal{A} and \mathcal{B} .

6 Final Remarks

We introduced Relational Color Refinement (RCR) as an adaptation of the classical Color Refinement (CR) procedure for arbitrary relational structures. We showed that it can be implemented with the same running time as CR (Theorem B). Furthermore, we showed that the distinguishing power of RCR admits an analogous combinatorial (Theorem C) and logical (Theorem D) characterization as CR. Combining the Theorems B, C and D yields our main result, Theorem A, formulated in Section 1.

There are multiple directions for further research:

- One interesting task is to lift the results of [2, 28] from CR and C^2 to RCR and $\text{GF}(\text{C})$. This is non-trivial, because $\text{GF}(\text{C})$ is capable of identifying certain σ -structures that cannot be identified in C^2 : consider, e.g., the signature $\sigma_3 := \{R\}$ with $\text{ar}(R) = 4$ and the σ_3 -structure \mathcal{A}_3 with $V(\mathcal{A}_3) = \{1, 2, 3\}$ and $R^{\mathcal{A}_3} = \{(1, 2, 3, 3)\}$. It is easy to construct a $\text{GF}(\text{C})$ -sentence φ that is satisfied by \mathcal{A}_3 but by no σ_3 -structure \mathcal{B}_3 that is not isomorphic to \mathcal{A}_3 and where every node is contained in some relation (recall the assumption on structures we adopted in Section 2). But there does not exist any C^2 -sentence φ with the same property – in fact, according to [28, Proof of Corollary 5.9], the logic C^2 does not identify any σ -structure whose universe contains ≥ 3 elements and where σ contains a relation symbol of arity ≥ 3 .
- Considering that CR is equivalent to the 1-dimensional Weisfeiler-Leman algorithm (WL), RCR might be a good basis to devise a generalization of the k -dimensional WL to arbitrary relational structures.
- Given the close relationship between relational structures and hypergraphs, a coloring method similar to RCR should exist for hypergraphs, too. However, RCR relies heavily on the order that the tuples provide – and this order is absent in hyperedges. Thus, it is not clear how to adapt the refinement of the colors in an iteration step from tuples to hyperedges.
- Another promising direction is to think about the applications of RCR, given that there are so many applications of classical CR, also apart from isomorphism testing (cf. Section 1). In particular, we conjecture that some of the techniques developed in this paper can be used to lift the result by Riveros, Scheidt, Schweikardt [31] from binary structures to arbitrary structures. Further, the tight connection between CR and Graph Neural Networks suggests interesting applications for RCR in machine learning as well. We plan to investigate this.

References

- 1 Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995. URL: <http://webdam.inria.fr/Alice/>.
- 2 V. Arvind, Johannes Köbler, Gaurav Rattan, and Oleg Verbitsky. Graph Isomorphism, Color Refinement, and Compactness. *Computational Complexity*, 26(3):627–685, September 2017. doi:10.1007/s00037-016-0147-6.
- 3 László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 684–697. ACM, 2016. doi:10.1145/2897518.2897542.
- 4 Catriel Beeri, Ronald Fagin, David Maier, and Mihalis Yannakakis. On the desirability of acyclic database schemes. *J. ACM*, 30(3):479–513, 1983. doi:10.1145/2402.322389.
- 5 Christoph Berkholz, Paul S. Bonsma, and Martin Grohe. Tight lower and upper bounds for the complexity of canonical colour refinement. *Theory Comput. Syst.*, 60(4):581–614, 2017. doi:10.1007/S00224-016-9686-0.
- 6 Philip A. Bernstein and Nathan Goodman. Power of natural semijoins. *SIAM J. Comput.*, 10(4):751–771, 1981. doi:10.1137/0210059.
- 7 Johann Brault-Baron. Hypergraph Acyclicity Revisited. *ACM Comput. Surv.*, 49(3):54:1–54:26, 2016. doi:10.1145/2983573.
- 8 Silvia Butti and Víctor Dalmau. Fractional Homomorphism, Weisfeiler-Leman Invariance, and the Sherali-Adams Hierarchy for the Constraint Satisfaction Problem. In Filippo Bonchi and Simon J. Puglisi, editors, *46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021)*, volume 202 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:19, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2021.27.
- 9 Jan Böker. Color Refinement, Homomorphisms, and Hypergraphs. In Ignas Sau and Dimitrios M. Thilikos, editors, *Graph-Theoretic Concepts in Computer Science - 45th International Workshop, WG 2019, Vall de Núria, Spain, June 19-21, 2019, Revised Papers*, volume 11789 of *Lecture Notes in Computer Science*, pages 338–350. Springer, 2019. doi:10.1007/978-3-030-30786-8_26.
- 10 Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, December 1992. doi:10.1007/BF01305232.
- 11 A. Cardon and Maxime Crochemore. Partitioning a Graph in $O(|A| \log_2 |V|)$. *Theor. Comput. Sci.*, 19:85–98, 1982. doi:10.1016/0304-3975(82)90016-0.
- 12 Anuj Dawar, Tomáš Jakl, and Luca Reggio. Lovász-type theorems and game comonads. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–13. IEEE, 2021. doi:10.1109/LICS52264.2021.9470609.
- 13 Holger Dell, Martin Grohe, and Gaurav Rattan. Lovász meets Weisfeiler and Leman. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 40:1–40:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.40.
- 14 Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Heidelberg, 6th edition, 2025. URL: <https://diestel-graph-theory.com/>.
- 15 Zdeněk Dvořák. On recognizing graphs by numbers of homomorphisms. *Journal of Graph Theory*, 64(4):330–342, 2010. doi:10.1002/jgt.20461.
- 16 Eva Fluck, Tim Seppelt, and Gian Luca Spitzer. Going deep and going wide: Counting logic and homomorphism indistinguishability over graphs of bounded treedepth and treewidth. In Aniello Murano and Alexandra Silva, editors, *32nd EACSL Annual Conference on Computer Science Logic, CSL 2024, February 19-23, 2024, Naples, Italy*, volume 288 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:17, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CSL.2024.27.

- 17 Georg Gottlob, Nicola Leone, and Francesco Scarcello. Hypertree decompositions and tractable queries. *J. Comput. Syst. Sci.*, 64(3):579–627, 2002. doi:10.1006/jcss.2001.1809.
- 18 Georg Gottlob, Nicola Leone, and Francesco Scarcello. Robbers, marshals, and guards: Game theoretic and logical characterizations of hypertree width. *Journal of Computer and System Sciences*, 66(4):775–808, 2003. doi:10.1016/S0022-0000(03)00030-8.
- 19 Martin Grohe. *Descriptive Complexity, Canonisation, and Definable Graph Structure Theory*, volume 47 of *Lecture Notes in Logic*. Cambridge University Press, 2017. doi:10.1017/9781139028868.
- 20 Martin Grohe. Counting Bounded Tree Depth Homomorphisms. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '20*, pages 507–520, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3373718.3394739.
- 21 Martin Grohe. Word2vec, Node2vec, Graph2vec, X2vec: Towards a Theory of Vector Embeddings of Structured Data. In Dan Suciu, Yufei Tao, and Zhewei Wei, editors, *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS'20*, pages 1–16. ACM, 2020. doi:10.1145/3375395.3387641.
- 22 Martin Grohe, Kristian Kersting, Martin Mladenov, and Pascal Schweitzer. Color Refinement and Its Applications. In Guy Van den Broeck, Kristian Kersting, Sriraam Natarajan, and David Poole, editors, *An Introduction to Lifted Probabilistic Inference*. The MIT Press, 2021. doi:10.7551/mitpress/10548.003.0023.
- 23 Martin Grohe, Kristian Kersting, Martin Mladenov, and Erkal Selman. Dimension Reduction via Colour Refinement. In Andreas S. Schulz and Dorothea Wagner, editors, *Algorithms - ESA 2014*, Lecture Notes in Computer Science, pages 505–516, Berlin, Heidelberg, 2014. Springer. doi:10.1007/978-3-662-44777-2_42.
- 24 Martin Grohe and Dániel Marx. Constraint solving via fractional edge covers. *ACM Trans. Algorithms*, 11(1):4:1–4:20, 2014. doi:10.1145/2636918.
- 25 Erich Grädel. On the Restraining Power of Guards. *Journal of Symbolic Logic*, 64(4):1719–1742, 1999. doi:10.2307/2586808.
- 26 Neil Immerman and Eric Lander. Describing Graphs: A First-Order Approach to Graph Canonization. In Alan L. Selman, editor, *Complexity Theory Retrospective: In Honor of Juris Hartmanis on the Occasion of His Sixtieth Birthday, July 5, 1988*, pages 59–81. Springer, New York, NY, 1990. doi:10.1007/978-1-4612-4478-3_5.
- 27 Sandra Kiefer. The Weisfeiler-Leman Algorithm: An Exploration of Its Power. *ACM SIGLOG News*, 7(3):5–27, November 2020. doi:10.1145/3436980.3436982.
- 28 Sandra Kiefer, Pascal Schweitzer, and Erkal Selman. Graphs Identified by Logics with Counting. *ACM Transactions on Computational Logic*, 23(1):1:1–1:31, October 2021. doi:10.1145/3417515.
- 29 Laura Mančinska and David E. Roberson. Quantum isomorphism is equivalent to equality of homomorphism counts from planar graphs. In Sandy Irani, editor, *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 661–672. IEEE, 2020. doi:10.1109/FOCS46700.2020.00067.
- 30 Yoav Montacute and Nihil Shah. The Pebble-Relation Comonad in Finite Model Theory. In Christel Baier and Dana Fisman, editors, *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science*, number 13 in LICS '22, pages 1–11, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3531130.3533335.
- 31 Cristian Riveros, Benjamin Scheidt, and Nicole Schweikardt. Using Color Refinement to Boost Enumeration and Counting for Acyclic CQs of Binary Schemas. *CoRR*, 2024. doi:10.48550/arXiv.2405.12358.
- 32 David E. Roberson. Oddomorphisms and homomorphism indistinguishability over graphs of bounded degree. *CoRR*, 2022. arXiv:2206.10321.
- 33 Benjamin Scheidt. On Homomorphism Indistinguishability and Hypertree Depth. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International*

- Colloquium on Automata, Languages, and Programming (ICALP 2024)*, volume 297 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 152:1–152:18, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2024.152.
- 34 Benjamin Scheidt and Nicole Schweikardt. Counting Homomorphisms from Hypergraphs of Bounded Generalised Hypertree Width: A Logical Characterisation. In Jérôme Leroux, Sylvain Lombardy, and David Peleg, editors, *48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023)*, volume 272 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 79:1–79:15, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2023.79.
- 35 Benjamin Scheidt and Nicole Schweikardt. Color Refinement for Relational Structures. *CoRR*, 2024. doi:10.48550/arXiv.2407.16022.
- 36 Tim Seppelt. Logical equivalences, homomorphism indistinguishability, and forbidden minors. In Jérôme Leroux, Sylvain Lombardy, and David Peleg, editors, *48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023)*, volume 272 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 82:1–82:15, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2023.82.
- 37 Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-Lehman Graph Kernels. *J. Mach. Learn. Res.*, 12:2539–2561, 2011. doi:10.5555/1953048.2078187.
- 38 Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: <https://openreview.net/forum?id=ryGs6iA5Km>.