Relational Colour Refinement for Non-Relational Signatures

Bachelor's Thesis

Theodor Jurij Teslia

August 28, 2025

RWTH Aachen University

Colour Refinement is an iterative graph algorithm that computes a colouring of the vertices, based on the colouring of their direct neighbours. Aside from useful applications in various fields it is theoretically interesting, due to its characterisations. It was proved that the distinguishing power of Colour Refinement is equivalent to that of counting logic with two variables and to that of counting homomorphisms from trees. Therefore, the question, whether such an algorithm is also possible for extensions of graphs was posed. Recently, Scheidt and Schweikardt defined an algorithm for relational structures with analogous properties. Especially the characterisations are similar to classical Colour Refinement. As such, its distinguishing power is equivalent to that of the guarded fragment of counting logic and to that of counting homomorphisms from α-acyclic relational structures.

In this thesis, we want to present the results of Scheidt and Schweikardt and investigate the question of how robust their algorithm and characterisations are. Concretely, we consider extending the class of relational structures to that of all structures and of structures with relations and unary functions, respectively. We find two logical characterisations but disprove the existence of a characterisation through homomorphism counting. Furthermore, we discuss for which restrictions on relational structures the characterisation through homomorphism counting remains and present two cases of restrictions, one for which the characterisation holds and one for this is does not.

1 Introduction

The graph isomorphism problem is a very interesting and important problem in both theoretical and applied computer science. The question, whether two graphs are structurally identical comes up in many fields of research. For example when trying to recognize identical chemical structures for patent verification [17] or finding symmetries for optimizing algorithms for problems such as the Boolean Satisfiability problem [14]. Furthermore, the graph isomorphism problem is interesting from a complexity theoretic perspective, as it is a prominent example for a problem, which has not been proven to be either NP-complete or solvable in polynomial time. [2]

One (incomplete) algorithm that is often used in practical isomorphism problem solvers [14] is *Colour Refinement*, short CR, also called the *1-dimensional Weisfeiler-Leman algorithm*. Given two graphs, it can decide, whether they might be isomorphic or whether they are not [4]. Concretely, CR is an iterative algorithm that, in the beginning, assigns every vertex the same colour and in following iterations assigns each one a new colour, based on the colours of its neighbours. This procedure is repeated, until the partition of the vertices induced by the colouring stays the same. We then say that Colour Refinement distinguishes two graphs, if there is some colour that appears differently often in the two graphs. It is easy to see that two isomorphic graphs are not distinguished by Colour Refinement. This is equivalent to the fact that if two graphs get distinguished by Colour Refinement, then they cannot be isomorphic. Furthermore, while it is not possible to infer the opposite direction, that is two non-isomorphic graphs always get distinguished by CR, it was shown by Babai, Erdős and Selkow that almost all graphs get distinguished by it [3]. However, there exist some classes of graphs, that cannot be distinguished by Colour Refinement, for example the classes of regular graphs with the same number of vertices.

Aside from isomorphism testing, Colour Refinement has applications in different fields. Incidentally, the first recorded occurrence of this algorithm appeared in 1965 and dealt with the description of chemical structures [16]. Its significance for computer science has been recognised later by Weisfeiler and Leman in 1968 [21]. One interesting application of Colour Refinement is in the reduction of the dimension of linear programs. By defining a variant of Colour Refinement on matrices which finds a partition of the rows and columns, it is possible to reformulate a linear program with a considerably smaller dimension. This method of first reducing the problem and then solving the reduced instance has been shown to be more performant than the standard way of solving linear programs. [13] Another application can be found in the field of machine learning, more precisely for kernel methods. There, it is possible to define a kernel for graphs, by counting the number of vertices that share the same colour for the first h rounds. This Weisfeiler-Leman Graph Kernel has an adequate ability to to classify graphs, while having a significantly better runtime than classical graph kernels. [12]

One importance of CR in theoretical computer science can be seen, when we consider other characterisations of its distinguishing power. Thus, CR can be equivalently characterised by counting homomorphisms from trees and by considering a certain logic. Due to Dvořák [10] and Dell, Grohe and Rattan [9] we have the following characterisation. Given two graphs G and H, we have that Colour Refinement distinguishes them if, and only if, there is some tree T such that the number of homomorphisms from T to G is different than to H. Such a characterisation can also be done through logic. We define C_2 as the logic that extends first-order logic by counting quantifiers and only uses up to two variables. Then, it was the shown by Cai and Immerman [8] and Immerman and Lander [15] that the following holds: Colour Refinement distinguishes G and H if, and only if, there is a C_2-sentence which is satisfied by G, but not by H.

From the above examples it can be seen that while Colour Refinement is a simple procedure, it can be applied in various situations. This versatility has been one of the reasons for its success and poses the question, whether it could be possible to formulate an analogous procedure for more than graphs. One obvious extension of graphs are hypergraphs. These are structures with a set of vertices and edges between those. However, while the edges of classical graphs connect two vertices, edges of

hypergraphs can include an arbitrary number of them. One interesting result due to Böker [5] is an extension of Colour Refinement to hypergraphs, which gives rise to an analogous characterisation using homomorphism counting. Concretely, Colour Refinement for a hypergraph G is defined like classical CR on a coloured variant of the incidence graph of G. We then consider connected Berge-acyclic hypergraphs, that is hypergraphs whose incidence graphs are trees. When counting homomorphisms from those to hypergraphs, we get that Colour Refinement distinguishes two hypergraphs G and H if, and only if, there is some connected Berge-acyclic hypergraph B, such that B has a different number of homomorphisms to G than to H. Another result with respect to hypergraphs has been achieved by Scheidt and Schweikardt in [18]. They devised a 2-sorted counting logic called GC^k and proved that two hypergraphs G and H satisfy exactly the same GC^k sentences if, and only if, all hypergraphs with generalised hypertree width k have the same number of homomorphisms to G as to H. For the case where $k = 1$, we then get indistinguishability over the class of all α-acyclic hypergraphs, which is a strictly stronger measure of acyclicity than Berge-acyclicity. Interestingly, we will encounter α-acyclicity and GC^1 in Section 3 for characterising relational structures, instead of hypergraphs. A first effort to extend Colour Refinement to relational structures has been made by Butti and Dalmau in [7]. They also defined Colour Refinement on the incidence graph of a relational structure and proved that this distinguishes two relational structures if, and only if, there is a connected Berge-acyclic relational structure with a different number of homomorphisms to the structures. A more recent result with respect to relational structures has been made by Scheidt and Schweikardt [19]. They defined an extension of Colour Refinement for relational structures, called Relational Colour Refinement, short RCR, which is stronger than the version in [7]. Thus it can distinguish structures that the variant in [7] cannot. Furthermore, they were able to define the logic $\mathsf{GF}(\mathsf{C})$, which characterises RCR in the same way as C_2 characterises classical CR. Additionally, the aforementioned α-acyclic structures characterise RCR as well. Concretely, we have that two relational structures of the same signature get distinguished by RCR if, and only if, there is an α-acyclic structure of the same signature such that is has a different amount of homomorphisms to the structures. A deeper discussion of the results from [19] can be found in Section 3.

It can be seen that there exist multiple comprehensive results for relational structures and hypergraphs. Especially the results from [19] seem like a very robust and usable extension for relational structures. Furthermore, we notice that functional, as well as non-relational signatures have not yet been investigated. As many practical and useful structures use non-relational signatures, for example all algebraic structures, we pose the question how robust the results from [19] are when trying to apply them to structures with (unary) functions. Concretely, we consider two possible ways of how Relational Colour Refinement can be adapted to signatures with functions. For both approaches we investigate, whether they can be characterised by a variant of $\mathsf{GF}(\mathsf{C})$ over signatures with functions and by counting homomorphisms from acyclic structures with non-relational signatures. We will see that such a characterisation through logic is in fact possible, while counting homomorphisms from acyclic structures is not. Additionally, we prove a similar result to the (not existing) characterisations by counting homomorphisms. We show that while it is not possible to require that the acyclic structure is functional (given a signature with functions), it is possible to require the acyclic structure to be symmetric (given that the two distinguished structures are symmetric as well). Therefore, we get that two symmetric structures are distinguished by RCR if, and only if, there is some symmetric, acyclic structure with different numbers of homomorphisms to the structures.

The results of this thesis are as follows. We define two variants of RCR for non-relational signatures, called naive RCR and RCR_k for a $k \in \mathbb{N}$, where the latter is only defined for unary functions. Naive RCR encodes n-ary functions f directly as a $(n+1)$-ary relations R_f, such that $(x_1, \ldots, x_n, y) \in R_f$ if, and only if, $f(x_1, \ldots, x_n) = y$ and then applies normal RCR to the relational structure. RCR_k encodes unary functions f as a family of relations R_{f^1}, R_{f^2}, $\ldots$, where $(x, y) \in R_{f^i}$ if, and only if, $f^i(x) = y$, that is the n-times application of f on x. Furthermore we define $\mathsf{nfGF}(\mathsf{C})$ as the logic that is defined as $\mathsf{GF}(\mathsf{C})$ over non-relational signatures but does not allow the nesting of terms. Additionally

we define $\mathsf{GF}(\mathsf{C})_k$ as the logic that is the normal extension of $\mathsf{GF}(\mathsf{C})$ to non-relational signatures but terms are only allowed to have an alternation-depth of function-applications of k. We further call an encoding of a non-relational structure as a relational one total, if the encoding of every function is defined for every value in the function-domain and we call the structure functional, if the encoding of every function defines at most one value for every element in the function-domain. We then achieve the following results.

Theorem 1.1 (Theorem 4.1)**.** *For a signature σ and two σ-structures $\mathfrak{A}$ and $\mathfrak{B}$, naive* RCR *distinguishes $\mathfrak{A}$ and $\mathfrak{B}$ if, and only if, there is a sentence $\varphi \in \mathsf{nfGF}(\mathsf{C})$ of signature σ such that $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \not\models \varphi$.*

Theorem 1.2 (Theorem 4.2)**.** *For a signature σ that only contains relation symbols and unary function symbols, two σ-structures $\mathfrak{A}$ and $\mathfrak{B}$ and a $k \in \mathbb{N}$,* RCR_k *distinguishes $\mathfrak{A}$ and $\mathfrak{B}$ if, and only if, there is a sentence $\varphi \in \mathsf{GF}(\mathsf{C})_k$ of signature σ such that $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \not\models \varphi$.*

Theorem 1.3 (Theorem 4.3)**.** *There exists a signature such that two structures of it get distinguished by naive* RCR *if, and only if, there exists an acyclic and functional structure that distinguishes the encodings by homomorphism count. Additionally, there cannot exist an acyclic, total encoding that has differently many homomorphisms to the encodings.*

Theorem 1.4 (Theorem 5.1)**.** *For two symmetric, relational structures $\mathfrak{A}$ and $\mathfrak{B}$,* RCR *distinguishes $\mathfrak{A}$ and $\mathfrak{B}$ if, and only if, there is a symmetric, acyclic structure that distinguishes $\mathfrak{A}$ and $\mathfrak{B}$ by number of homomorphism.*

The methods used to achieve the above results rely heavily on the proofs from [19]. For the logical characterisation of structures with functions we translate both structures to relational structures and translate any formula to a valid $\mathsf{GF}(\mathsf{C})$-formula over a relational signature. The same is done in reverse as well. For the characterisation through homomorphism counting, we give a family of counterexamples.

The structure of this thesis is as follows. We begin by defining notation and fundamental definitions in Section 2. Then in Section 3 we present and explain the results from [19]. Afterwards, we continue in Section 4 by considering structures with functions, where we will first show the logical characterisation for the two approaches, before then discussing the characterisation through homomorphism counting. Lastly, we discuss the restriction to symmetric structures in Section 5.

2 Preliminaries

Let A be a set. Then a $a_1, \dots, a_k \in A$ form a tuple with length k of the form $\mathbf{a} = (a_1, \dots, a_k)$ and with $\mathbf{a}_n$ we denote a_n. For any function $f : A \to B$, we write $f(\mathbf{a})$ for $(f(a_1), \dots, f(a_k))$. If π is a permutation, that is $\pi \in \mathcal{S}_k$, where $\mathcal{S}_k$ is the symmetric group for k elements, then $\pi(\mathbf{a})$ means $(a_{\pi(1)}, \dots, a_{\pi(k)})$. For a set A and a $k \in \mathbb{N}$, we write $\binom{A}{k}$, to denote all subsets of A with cardinality of exactly k. Let $\mathbf{a} = (a_1, \dots, a_k)$ be a tuple of length k and b a element. Then we write $(\mathbf{a}b) \coloneqq (a_1, \dots, a_k, b)$ for the tuple of length $k+1$ with b added to the end. For a tuple $\mathbf{a} = (a_1, \dots, a_k)$ we write $\mathrm{set}(\mathbf{a}) \coloneqq \{a_1, \dots, a_k\}$ for the set that contains the elements from $\mathbf{a}$. Usually, sets are denoted by uppercase, elements from sets as lowercase, and tuples as boldface and lowercase Latin letters.

By $\mathbb{N}$ we mean the set of natural numbers, including 0, and with $\mathbb{N}_{\geq k}$ for a $k \in \mathbb{N}$, we denote the set $\{k, k+1, \dots\} = \mathbb{N} \setminus \{0, \dots, k\}$. We use $[k]$ for any $k \in \mathbb{N}_{\geq 1}$ to mean $\{1, 2, \dots, k\}$.

A multiset $\mathcal{M}$ is a pair with a set A and a function $\mathrm{mult}_{\mathcal{M}} : A \to \mathbb{N}_{\geq 1}$. This denotes, that for every $a \in A$, there are exactly $\mathrm{mult}_{\mathcal{M}}(a)$ many copies. We also write multisets as $\{\!\{\dots\}\!\}$. As an example, the multiset $\mathcal{M} = \{\!\{a, b, b\}\!\}$ is equivalent to the set $A = \{a, b\}$, together with the function $\mathrm{mult}_{\mathcal{M}} = \{a \mapsto 1, b \mapsto 2\}$.

A signature is a set of relation and function symbols, paired with an arity for every symbol. We write relation symbols as uppercase and function symbols as lower case letters. For example, $\sigma = \{R/2, T/3, f/3\}$ represents a signature with a relation symbol R of arity 2, a relation symbol T with arity 3 and a function symbol f with arity 3. We call a signature relational if it only contains relation symbols and binary if it contains only unary and binary relation symbols. For a function or relation symbol $R \in \sigma$, we write $\operatorname{ar}(R)$ for the arity of R and we define $\operatorname{ar}(\sigma) \coloneqq \max_{R\in\sigma}\{\operatorname{ar}(R)\}$ as the maximal arity that appears in σ. For a signature $\sigma = \{R_1/\operatorname{ar}(R_1), R_2/\operatorname{ar}(R_2), \dots, f_1/\operatorname{ar}(f_1), f_2/\operatorname{ar}(f_2), \dots\}$ we define a σ-structure $\mathfrak{A}$ as a tuple $(A, R_1^{\mathfrak{A}}, R_2^{\mathfrak{A}}, \dots, f_1^{\mathfrak{A}}, f_2^{\mathfrak{A}}, \dots)$ where A, called the universe, is a set of elements and $R_i^{\mathfrak{A}} \subseteq A^{\operatorname{ar}(R_i)}$ is a relation of arity $\operatorname{ar}(R_i)$ and $f_i^{\mathfrak{A}} : A^{\operatorname{ar}(f_i)} \to A$ is a function of arity $\operatorname{ar}(f_i)$. We define $|\mathfrak{A}| = |A|$. A structure is always written as an uppercase Fraktur letter, its universe is written using the same uppercase Latin letter. For relation structures, we use an uppercase, boldface Larin letter to denote the union of all relations, formally $\bigcup_{R\in\sigma} R$. As an example, $\mathfrak{A}$ is a structure, A is its universe and $\mathbf{A}$ is the union of all relations. A graph G is a pair of a set of vertices, denoted as $V(G)$, and a binary and symmetric relation over that set of vertices, called the edges $E(V)$. If not specified, we only talk about finite signatures and finite universes.

For a graph $G = (V, E)$, we apply Colour Refinement by inductively defining for every $v \in V$ and every $i \in \mathbb{N}$ a colour $C_i(v)$, using the following method. The initial colouring is defined as $C_0(v) = 0$. For $i \in \mathbb{N}$, we define $C_{i+1}(v) = (C_i(v), \{\!\{C_i(u) : \{v, u\} \in E\}\!\})$. We say that Colour Refinement distinguishes two graphs G and H, if there is an $i \in \mathbb{N}$, such that

$$\{\!\{C_i(v) : v \in V(G)\}\!\} \neq \{\!\{C_i(u) : u \in V(H)\}\!\}.$$

For binary relational structures it is possible to adapt classical Colour Refinement to them, by using the unary relations in the initial colouring and annotating neighbours with the binary relations that connect the elements. Let $\sigma = \sigma_1 \dot\cup \sigma_2$, where σ_1 contains only unary and σ_2 contains only binary relations. We write $E(\mathcal{G})$ for $\bigcup_{E\in\sigma_2}\{\{v, w\} : (v, w) \in E^{\mathcal{G}}\}$. Then, for every element $e \in G$ we define

$$\gamma_0(e) = (\{C \in \sigma_1 : e \in C\}, \{E \in \sigma_2 : (e, e) \in E^{\mathcal{G}}\}).$$

For every $e \in G$, every $i \in \mathbb{N}$ we define a

$$\gamma_{i+1}(e) = (\{\gamma_i(e), \{\!\{(\lambda(e, f), \gamma_i(f)) : \{e, f\}) \in E(\mathcal{G})\}\!\}),$$

and

$$\lambda_{i+1}(e, f) = \{E^+ : E \in \sigma_2, (e, f) \in E^{\mathcal{G}}\} \cup \{E^- : E \in \sigma_2, (f, e) \in E^{\mathcal{G}}\},$$

where E^+ and E^- for every $E \in \sigma_2$ are new symbols that do not appear in σ. This definition can also be found in [19].

For a signature σ and two σ-structures $\mathfrak{A}$ and $\mathfrak{B}$, we write call $\varphi : A \to B$ a homomorphism, if:

- For every relation symbol $R \in \sigma$ of arity k and every $(x_1, \dots, x_k) \in R^{\mathfrak{A}}$, $(\varphi(x_1), \dots, \varphi(x_k)) \in R^{\mathfrak{B}}$ and
- for every function symbol $f \in \sigma$ of arity k and every $(x_1, \dots, x_k)$, $\varphi(f(x_1, \dots, x_k)) = f(\varphi(x_1), \dots, \varphi(x_k))$.

We then write $\operatorname{Hom}(\mathfrak{A}, \mathfrak{B})$ for the set of all homomorphisms from $\mathfrak{A}$ to $\mathfrak{B}$ and define $\hom(\mathfrak{A}, \mathfrak{B}) \coloneqq |\operatorname{Hom}(\mathfrak{A}, \mathfrak{B})|$. This applies analogously for relational structures and graphs.

For a $\varphi(x_1, \dots, x_k) \in \mathsf{C}$, where C is the logic that extends FO with counting quantifiers, we write $\mathfrak{A}, a_1, \dots, a_k \models \varphi(x_1, \dots, x_k)$ to mean $\mathfrak{A} \models \varphi(a_1, \dots, a_k)$. Similarly, for a $\psi(x_1, \dots, x_k, y_1, \dots, y_\ell) \in \mathsf{C}$, we write $\mathfrak{A}, a_1, \dots, a_k, b_1, \dots, b_\ell \models \psi(x_1, \dots, x_k, y_1, \dots, y_\ell)$ for $\mathfrak{A} \models \psi(a_1, \dots, a_k, b_1, \dots, b_\ell)$.

3 Relational Colour Refinement

Relational structures are a very important and interesting extension of graphs. They can be found in various situation, for example in relational database theory where they are used to model the structure of databases [1]. In this section we want to look at the Relational Colour Refinement algorithm, short RCR. It was defined by Scheidt and Schweikardt in [19] as an extension of Colour Refinement for relational structures. The algorithm is similar to the adaptation of classical Colour Refinement for binary structures and has characterisations analogous to classical Colour Refinement. Let us begin by defining RCR and applying it to a small example.

3.1 RCR and Binary Structures

The idea of RCR is to apply Colour Refinement on the tuples that appear in a relation. This can then be seen as a binary structure, on which the before mentioned Colour Refinement algorithm can be applied. To define RCR, we need the following definitions.

Definition 3.1 (Atomic and shared types)**.** Let σ be a relational signature and $\mathfrak{A}$ be a σ-structure. For every $\mathbf{a} \in \mathbf{A}$ we define $\operatorname{atp}(\mathbf{a}) \coloneqq \{R \in \sigma : \mathbf{a} \in R\}$ as the set of relations that include $\mathbf{a}$. For every tuple $\mathbf{a} \in \mathbf{A}$ of length n and every tuple $\mathbf{b} \in \mathbf{A}$ of length m we define $\operatorname{stp}(\mathbf{a}, \mathbf{b}) \coloneqq \{(i, j) \in [n] \times [m] : \mathbf{a}_i = \mathbf{b}_j\}$ as the set of all pairs of indices which denote the elements that appear in both $\mathbf{a}$ and $\mathbf{b}$. We call $\operatorname{atp}(\mathbf{a})$ the atomic type of $\mathbf{a}$ and call $\operatorname{stp}(\mathbf{a}, \mathbf{b})$ the shared type of $\mathbf{a}$ and $\mathbf{b}$. As shorthand notation we write $\operatorname{stp}(\mathbf{a})$ for $\operatorname{stp}(\mathbf{a}, \mathbf{a})$.

We can now define Relational Colour Refinement. For every $\mathbf{a} \in \mathbf{A}$ we iteratively construct a colour $\varrho_i(\mathbf{a})$. We begin by defining the initial colour using the atomic and shared type of $\mathbf{a}$ itself. Formally, $\varrho_0(\mathbf{a}) \coloneqq (\operatorname{atp}(\mathbf{a}), \operatorname{stp}(\mathbf{a}))$. For every $i \in \mathbb{N}_{\geq 1}$ the colour $\varrho_i(\mathbf{a})$ is made up of $\varrho_{i-1}(\mathbf{a})$ and the colour of the last round for every tuple that $\mathbf{a}$ shares an element with, in addition to the information which elements are shared. This is formalised as $\varrho_{i+1}(\mathbf{a}) \coloneqq (\varrho_i(\mathbf{a}), N^{\mathfrak{A}}_{i+1}(\mathbf{a}))$, where

$$N^{\mathfrak{A}}_{i+1} \coloneqq \{\!\{(\operatorname{stp}(\mathbf{a}, \mathbf{b}), \varrho_i(\mathbf{b})) : \mathbf{b} \in \mathbf{A}, \operatorname{stp}(\mathbf{a}, \mathbf{b}) \neq \emptyset\}\!\}.$$

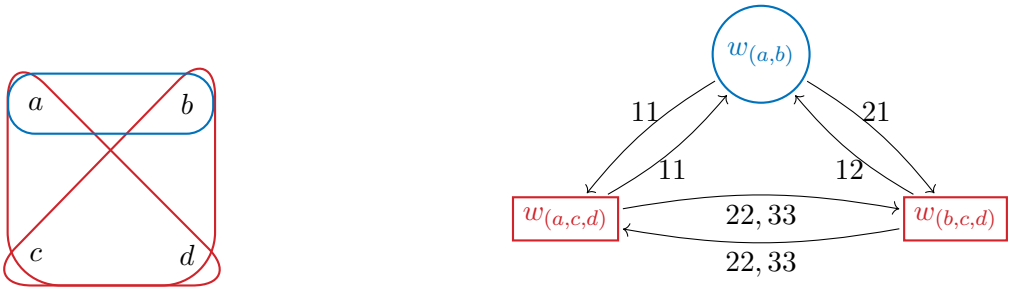
It can easily be verified that if $\varrho_i(\mathbf{a}) \neq \varrho_i(\mathbf{b})$, then $\varrho_{i+1}(\mathbf{a}) \neq \varrho_{i+1}(\mathbf{b})$, we therefore get a refinement. This also means there must be a smallest $i_{\mathfrak{A}} \leq \mathbf{A}$, such that for every $\mathbf{a}, \mathbf{b} \in \mathbf{A}$, we have that $\varrho_{i_{\mathfrak{A}}}(\mathbf{a}) = \varrho_{i_{\mathfrak{A}}}(\mathbf{b})$ if, and only if, $\varrho_{i_{\mathfrak{A}}+1}(\mathbf{a}) = \varrho_{i_{\mathfrak{A}}+1}(\mathbf{b})$, we then call this colouring the stable colouring. For a relational structure $\mathfrak{A}$ and $i \in \mathbb{N}$ we define $\operatorname{RC}_i(\mathfrak{A}) \coloneqq \{\varrho_i(\mathbf{a}) : \mathbf{a} \in \mathbf{A}\}$ as the set of colours that get constructed in the i-th round of RCR. Further, for a relational structure $\mathfrak{A}$, $i \in \mathbb{N}$ and $c \in \operatorname{RC}_i(\mathfrak{A})$ we write $\operatorname{mult}_{\mathfrak{A}}(c) \coloneqq |\{\mathbf{a} \in \mathbf{A} : \varrho_i(\mathbf{a}) = c\}|$ for the number of occurrences of the colour c. Finally, for two σ-structures $\mathfrak{A}$ and $\mathfrak{B}$ we say that Relational Colour Refinement distinguishes $\mathfrak{A}$ and $\mathfrak{B}$, if there is an $i \leq \max\{i_{\mathfrak{A}}, i_{\mathfrak{B}}\}$ and a colour $c \in \operatorname{RC}_i(\mathfrak{A}) \cup \operatorname{RC}_i(\mathfrak{B})$ such that $\operatorname{mult}_{\mathfrak{A}}(c) \neq \operatorname{mult}_{\mathfrak{B}}(c)$.

When comparing this algorithm with the Colour Refinement algorithm for binary structures, we notice similarities. In fact, it is possible to interpret a relational structure as a binary structure, on which classical Colour Refinement can be applied with an equivalent result to Relational Colour Refinement on the original structure. For a relational signature σ we define a signature $\widehat{\sigma} \coloneqq \{E_{i,j}/2 : i, j \in [\operatorname{ar}(\sigma)]\} \cup \{U_R/1 : R \in \sigma\}$. For a σ-structure $\mathfrak{A}$, we can now define a binary structure $\mathcal{G}_{\mathfrak{A}}$ of signature $\widehat{\sigma}$. The universe of $\mathcal{G}_{\mathfrak{A}}$ is defined as $V(\mathcal{G})_{\mathfrak{A}} = \{w_{\mathbf{a}} : \mathbf{a} \in \mathbf{A}\}$. In words, we create a new element $w_{\mathbf{a}}$ for every tuple $\mathbf{a} \in \mathbf{A}$. The relations are then defined as follows. For $R \in \sigma$ we set

$$U_R^{\mathcal{G}_{\mathfrak{A}}} \coloneqq \{w_{\mathbf{a}} : \mathbf{a} \in R\}$$

and for $i, j \in [\operatorname{ar}(\sigma)]$ we set

$$E_{i,j}^{\mathcal{G}_{\mathfrak{A}}} \coloneqq \{(w_{\mathbf{a}}, w_{\mathbf{b}}) : (i, j) \in \operatorname{stp}(\mathbf{a}, \mathbf{b})\}.$$



(a) The structure $\mathfrak{A}$, where the blue edge represents R and the red edges represent T.

(b) The binary structure $\mathcal{G}_{\mathfrak{A}}$, where the blue circle represents U_R and the red rectangles represent U_T.

Figure 1: A relational structure $\mathfrak{A}$ of signature $\sigma = \{R/2, T/3\}$ and the binary structure $\mathcal{G}_{\mathfrak{A}}$ that encodes it.

It was then proved by Scheidt and Schweikardt [19] that, when identifying $\mathbf{a}$ with $w_{\mathbf{a}}$ for every $\mathbf{a} \in \mathbf{A}$, Colour Refinement on $\mathcal{G}_{\mathfrak{A}}$ is equivalent, or rather induces the same partition of the tuples in every round, as Relational Colour Refinement on $\mathfrak{A}$.

3.1.1 An Example for Relational Colour Refinement

We want to illustrate this with a small example. We choose the signature $\sigma = \{R/2, T/3\}$ and a σ-structure $\mathfrak{A} = (A, R^{\mathfrak{A}}, T^{\mathfrak{A}})$ with $A = \{a, b, c, d\}$, $R^{\mathfrak{A}} = \{(a, b)\}$ and $T^{\mathfrak{A}} = \{(a, c, d), (b, c, d)\}$. A graphical representation can be found in Figure 1a. When applying the transformation described above, we get the signature $\widehat{\sigma} = \{E_{(i,j)} : i, j \in [3]\} \cup \{U_T, U_R\}$. The complete definition of $\mathcal{G}_{\mathfrak{A}}$ can be seen in Figure 1b, where an edge with the label ij from element a to b represents a tuple $(a, b) \in E_{i,j}^{\mathcal{G}_{\mathfrak{A}}}$, shared edges are contracted and self-loops are omitted.

We now want to apply RCR on $\mathfrak{A}$ and then classical CR on $\mathcal{G}_{\mathfrak{A}}$. By this we will see that both algorithms generate the same partition of elements. By the definition it is obvious that $\varrho_0((a, b)) = (\{R\}, \{(1, 1), (2, 2)\})$ and $\varrho_0((a, c, d)) = \varrho_0((b, c, d)) = (\{T\}, \{(1, 1), (2, 2), (3, 3)\})$. Thus (a, b) already has a different colour from the other two tuples. In the next step (a, c, d) and (b, c, d) will also receive different colours. Concretely, we have

$$\begin{aligned}\varrho_1((a, c, d)) = (\varrho_0((a, c, d)), \{\!\{&(\{(1, 1)\}, \varrho_0((a, b))),\\ &(\{(2, 2), (3, 3)\}, \varrho_0((b, c, d))),\\ &(\{(1, 1), (2, 2), (3, 3)\}, \varrho_0((a, c, d)))\}\!\})\end{aligned}$$

and

$$\begin{aligned}\varrho_1((b, c, d)) = (\varrho_0((b, c, d)), \{\!\{&(\{(1, 2)\}, \varrho_0((a, b))),\\ &(\{(2, 2), (3, 3)\}, \varrho_0((a, c, d))),\\ &(\{(1, 1), (2, 2), (3, 3)\}, \varrho_0((b, c, d)))\}\!\}).\end{aligned}$$

It can thus be seen that $\varrho_1((a, c, d)) \neq \varrho_1((b, c, d))$. The algorithm now terminates as every element has its own colour.

We will now get the same results when applying classical Colour Refinement to $\mathcal{G}_{\mathfrak{A}}$. Similarly to RCR, we have $\gamma_0(w_{(a,b)}) = (\{U_R\}, \{E_{1,1}, E_{2,2}\})$ and $\gamma_0(w_{(a,c,d)}) = \gamma_0(w_{(b,c,d)}) = (\{U_T\}, \{E_{1,1}, E_{2,2}, E_{3,3}\})$. As before, $w_{(a,c,d)}$ and $w_{(b,c,d)}$ share the initial colour, while $w_{(a,b)}$ has its own. For the second round

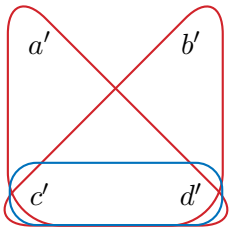


Figure 2: The σ-structure $\mathfrak{B}$ that gets distinguished by RCR from $\mathfrak{A}$.

we now get

$$\begin{aligned}\gamma_1(w_{(a,c,d)}) = (\gamma_0(w_{(a,c,d)}), \{\!\{&(\{E^+_{1,1}, E^-_{1,1}\}, \gamma_0(w_{(a,b)})),\\ &(\{E^+_{2,2}, E^+_{3,3}, E^-_{2,2}, E^-_{3,3}\}, \gamma_0(w_{(b,c,d)})),\\ &(\{E^+_{1,1}, E^+_{2,2}, E^+_{3,3}, E^-_{1,1}, E^-_{2,2}, E^-_{3,3}\}, \gamma_0(w_{(a,c,d)}))\}\!\})\end{aligned}$$

and

$$\begin{aligned}\gamma_1(w_{(b,c,d)}) = (\gamma_0(w_{(b,c,d)}), \{\!\{&(\{E^+_{1,2}, E^-_{2,1}\}, \gamma_0(w_{(a,b)})),\\ &(\{E^+_{2,2}, E^+_{3,3}, E^-_{2,2}, E^-_{3,3}\}, \gamma_0(w_{(a,c,d)})),\\ &(\{E^+_{1,1}, E^+_{2,2}, E^+_{3,3}, E^-_{1,1}, E^-_{2,2}, E^-_{3,3}\}, \gamma_0(w_{(b,c,d)}))\}\!\}).\end{aligned}$$

Again, $\gamma_1(w_{(a,c,d)}) \neq \gamma_1(w_{(b,c,d)})$ and the algorithm terminates. We see that both procedures act equally, which is what was proved by Scheidt and Schweikardt.

Let us now look at an example where RCR distinguishes two structures. For this, consider the σ-structure $\mathfrak{B} = (B, R^{\mathfrak{B}}, T^{\mathfrak{B}})$ with $B = (a', b', c', d')$, $R^{\mathfrak{B}} = \{(c', d')\}$ and $T^{\mathfrak{B}} = \{(a', c', d'), (b', c', d')\}$, which can be seen in Figure 2. It can easily be seen that every colour appears exactly as often in the colouring of the tuples of $\mathfrak{A}$ as of $\mathfrak{B}$. Thus RCR cannot distinguish the structures in round 0. However, as $\mathfrak{B}$ does not include a tuple with a' as the first element, the colour $\gamma_1((a, c, d))$ does not appear in round 1 of RCR in $\mathfrak{B}$. Therefore $\text{mult}_{\mathfrak{A}}(\gamma_1((a, c, d))) = 1 \neq 0 = \text{mult}_{\mathfrak{B}}(\gamma_1((a, c, d)))$ and RCR distinguishes $\mathfrak{A}$ and $\mathfrak{B}$ in round 1.

In their paper, Scheidt and Schweikardt consider a larger example, which is an extension of the classical non-distinguishable example for Colour Refinement. It uses a signature with a binary and a 6-ary relation and is comprised of the structure $\mathfrak{A}_1$ with one 6-cycle and the structure $\mathfrak{A}_2$ with two 3-cycles. Without the 6-ary relation the structure would be a regular graph and therefore could not be distinguished by Colour Refinement. Because of that, the 6-ary relation is added with one tuple, containing all 6 elements each. With this change, the structures can be distinguished, which is discussed in [19].

Furthermore, Scheidt and Schweikardt investigate other, seemingly simpler, possible variants of Colour Refinement which use the Gaifman-Graph and Incidence-Graph of a relational structure. However, these variants are not able to distinguish $\mathfrak{A}_1$ and $\mathfrak{A}_2$, which is why they are disregarded.

3.2 Logical Characterisation of RCR

Classical Colour Refinement gets characterised by counting logic with up to two variables, also called C_2. This means that two graphs G and H get distinguished by Colour Refinement if, and only if, there is a sentence $\varphi \in \mathsf{C}_2$, such that $G \models \varphi$ and $H \not\models \varphi$ [15]. Similarly, Relational Colour Refinement is characterised by the guarded fragment of counting logic, in short $\mathsf{GF}(\mathsf{C})$. This logic restricts first-order logic with counting quantifiers in the same way, as the guarded fragment of first-order logic GF restricts FO. An investigation of this notion of guards can be found in [11]. This means that the

bound on the number of variables is dropped, but the restriction that quantifiers need to be relativised by an atomic formula is added.

Definition 3.2 (The guarded fragment of counting logic)**.** For a relational signature σ we define the class $\mathsf{GF}(\mathsf{C})$ over σ inductively using the following rules:

1. Let $R \in \sigma$ with $\ell = \mathrm{ar}(R)$ and let $x_1, \ldots, x_\ell$ be variables. Then $R(x_1, \ldots, x_\ell) \in \mathsf{GF}(\mathsf{C})$.
2. Let a and y be two variables. Then $x = y \in \mathsf{GF}(\mathsf{C})$.
3. Let $\varphi \in \mathsf{GF}(\mathsf{C})$. Then $\neg(\varphi) \in \mathsf{GF}(\mathsf{C})$.
4. Let $\varphi, \psi, \in \mathsf{GF}(\mathsf{C})$. Then $(\varphi \wedge \psi) \in \mathsf{GF}(\mathsf{C})$.
5. For a formula built using Rule *1.* or *2.* Δ and a $\varphi \in \mathsf{GF}(\mathsf{C})$ we call Δ a guard for φ, if $\mathrm{free}(\Delta) \supseteq \mathrm{free}(\varphi)$. Let $\Delta, \varphi \in \mathsf{GF}(\mathsf{C})$, where Δ is a guard for φ, let $\mathbf{v}$ be a tuple of variables with $\mathrm{set}(\mathbf{v}) \subseteq \mathrm{free}(\Delta)$ and let $n \in \mathbb{N}_{\geq 1}$. Then $\exists^{\geq n}\mathbf{v}.(\Delta \wedge \varphi) \in \mathsf{GF}(\mathsf{C})$.

Formulae that are built using Rules *1.* and *2.* are called atomic formulae, we omit parenthesises in the usual way and use $\exists^{=n}\mathbf{v}.(\Delta \wedge \varphi)$ as shorthand notation for $\exists^{\geq n}\mathbf{v}.(\Delta \wedge \varphi) \wedge \neg\exists^{\geq n+1}\mathbf{v}.(\Delta \wedge \varphi)$.

The semantics of this logic are analogous to classical Counting Logic, see for example [8], and also are concretely defined in [19].

We find that this logic is equivalent to the logic GC^1 as it is defined by the same authors in [18].

Using a pebble game for $\mathsf{GF}(\mathsf{C})$, called the Guarded-Game, Scheidt and Schweikardt proved the following theorem:

Theorem 3.1 (Theorem B from [19])**.** *Let σ be a relational signature and let $\mathfrak{A}$ and $\mathfrak{B}$ be σ-structures. Then the three following statements are equivalent:*

1. *Relational Colour Refinement distinguishes $\mathfrak{A}$ and $\mathfrak{B}$.*
2. *There exists a sentence $\varphi \in \mathsf{GF}(\mathsf{C})$ such that $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \not\models \varphi$.*
3. *Spoiler wins the Guarded-Game on $\mathfrak{A}$ and $\mathfrak{B}$.*

Let us again consider the structures from Figures 1a and 2. In the preceding section we used the colour $\varrho_1((a, c, d))$ to distinguish $\mathfrak{A}$ and $\mathfrak{B}$, as it does not appear in the colouring of $\mathfrak{B}$. More precisely, there does not exist a tuple of length 3 in $\mathfrak{B}$, which is in the relation T and its first element is in the relation R with another element. This can be formalised using the formula

$$\varphi_1 \coloneqq \exists^{\geq 1}(x, y, z).\left(T(x, y, z) \wedge \exists^{\geq 1}(y).\,(R(x, y))\right).$$

It is easy to see that $\mathfrak{A} \models \varphi_1$ and $\mathfrak{B} \not\models \varphi_1$. Another formula can be obtained from the fact that in the stable colouring of $\mathfrak{B}$, the tuples (a, c, d) and (b, c, d) share the same colour. Therefore, the formula

$$\varphi_2 \coloneqq \exists^{\geq 2}(x, y, z).\,(T(x, y, z) \wedge R(y, z))$$

is also distinguishing.

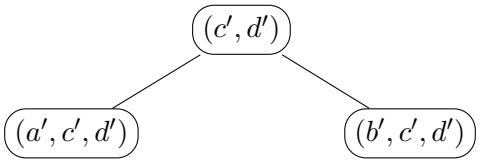


Figure 3: A join-tree for the structure $\mathfrak{B}$ from Figure 2

3.3 Characterising RCR by Homomorphism Counting

Another way to characterise classical CR is to count homomorphisms from trees. Due to [10] and [9] it is known that Colour Refinement distinguishes two graphs G and H if, and only if, there is a tree T, such that $\hom(T, G) \neq \hom(T, H)$. Again, there is an analogous characterisation for Relational Colour Refinement. One obstacle in defining such a characterisation is finding a class that generalises trees for relational structures. As a tree is a connected, acyclic graph, we have to find a fitting notion of acyclicity for relational structures. As can be seen in [6], there are multiple such definitions possible for hypergraphs, which can be applied to relational structures as well. When considering the results from [18] and that GC^1 characterises RCR, it becomes clear that hypergraphs of generalised hypertree width of 1, or equivalently α-acyclic hypergraphs, may be a possible candidate. This is in fact the case. Let us therefore define α-acyclic structures, or in the following just acyclic structures.

Definition 3.3 (Acyclic structures)**.** Let σ be relational signature and let $\mathfrak{C}$ be a σ-structure. A join-tree J for $\mathfrak{C}$ is a tree with a vertex for every tuple in $\mathfrak{C}$, thus $V(J) = \mathbf{C}$, which fulfils the join-tree-property: For any $c \in C$, the set $\{\mathbf{c} \in V(J) : c \in \mathrm{set}(\mathbf{c})\}$ induces a connected subgraph of J. This induced subgraph is also a tree and will be denoted as J_c. Finally, we call structure acyclic, if it has a join-tree.

There are multiple equivalent characterisations for this notion of acyclicity which can be found in [6]. Recall the definition for homomorphisms between structures and the definitions of Hom and hom. This then leads us to another main result from [19].

Theorem 3.2 (Theorem A from [19])**.** *Let σ be a relational signature and let $\mathfrak{A}$ and $\mathfrak{B}$ be σ-structures. Then the two following statements are equivalent:*

1. *Relational Colour Refinement distinguishes $\mathfrak{A}$ and $\mathfrak{B}$.*
2. *There exists an acyclic σ-structure $\mathfrak{C}$, such that $\hom(\mathfrak{C}, \mathfrak{A}) \neq \hom(\mathfrak{C}, \mathfrak{B})$.*

We again want to consider the structures from Figures 1a and 2 for a simple example. The structure $\mathfrak{B}$ is acyclic. This can be seen from its join tree depicted in Figure 3. Now consider the homomorphisms from $\mathfrak{B}$. The identity is always a homomorphism, therefore $\hom(\mathfrak{B}, \mathfrak{B}) \geq 1$. However, when trying to construct a homomorphism from $\mathfrak{B}$ to $\mathfrak{A}$, we encounter some issues. As $(c', d') \in R^{\mathfrak{B}}$ and (a, b) is the only tuple in $R^{\mathfrak{A}}$, we have to map c' to a and d' to b. But now consider the tuple $(a', c', d') \in T^{\mathfrak{B}}$ and let x be the element that a' gets mapped to. Then (x, a, b) must be in $T^{\mathfrak{A}}$. But there is no such tuple in $\mathfrak{A}$, independently of x. Therefore there cannot exist a homomorphism from $\mathfrak{B}$ to $\mathfrak{A}$, thus $\hom(\mathfrak{B}, \mathfrak{A}) = 0 \neq \hom(\mathfrak{B}, \mathfrak{B})$ and $\mathfrak{B}$ is an acyclic structure that distinguishes $\mathfrak{B}$ and $\mathfrak{A}$ by homomorphism count.

4 Relational Colour Refinement for Structures With Functions

Many interesting structures use functions. For example all algebraic signatures include function symbols for addition and multiplication, but also constants, which are understood as 0-ary functions. As Colour Refinement and variants of it have shown to be very useful in multiple applications,

a similar method for structures with functions seems desirable. This section aims to apply the results from [19], which were discussed in the previous section, to such classes of structures. We will define two different approaches of Colour Refinement Algorithms. Both will encode structures and signatures with functions as relational structures and relational signatures, respectively. On these relational structures the already discussed Relational Colour Refinement can be applied. We then study, whether the non-relational signatures can also be used in the characterisations. Concretely, can we use functions in the formulae which form the logic that characterises the algorithms, if such a logic exists? To what extend can these functions be used and what restrictions apply? The same will be done for the characterisation by homomorphism counting. Can we use the encodings to define acyclic, non-relational structures, and can the classes of those characterise the algorithms by homomorphism counting? We will begin by defining both approaches and then comparing those with a simple example. Afterwards we will look at the logical characterisation and end with the characterisation by homomorphism counting.

4.1 Naive Encoding of Functions

The simplest form to encode a non-relational structure as a relational one is to simply interpret functions as relations as follows. For a non-relational signature σ, we define a relational signature σ'. For every relation symbol $R \in \sigma$, we introduce a relation symbol $R \in \sigma'$ with the same arity and for every function symbol $f \in \sigma$ with arity k, we introduce a relational symbol $R_f \in \sigma'$ of arity $k+1$. Semantically, a structure $\mathfrak{A}$ of signature σ can then be encoded as a structure $\mathfrak{A}'$ of signature σ' and with the same universe as $\mathfrak{A}$. For every relational symbol $R \in \sigma$ we set $R^{\mathfrak{A}'} \coloneqq R^{\mathfrak{A}}$ and for every function symbol $f \in \sigma$ of arity k there exists a relation symbol $R_f \in \sigma'$ and we set $R_f^{\mathfrak{A}} \coloneqq \{(\mathbf{x}y) : f^{\mathfrak{A}}(\mathbf{x}) = y\}$ where $\mathbf{x}$ is a tuple of length k.

This procedure encodes a non-relational structure as a relational one, on which Relational Colour Refinement can now be performed. As such we say, that the Naive Relational Colour Refinement (nRCR) distinguishes two structures $\mathfrak{A}$ and $\mathfrak{B}$ if, and only if, RCR distinguishes their naive encodings $\mathfrak{A}'$ and $\mathfrak{B}'$. However, this results in a very weak logical characterisation, that does not allow nesting of terms, namely the nesting-free-fragment of $\mathsf{GF}(\mathsf{C})$, which we will discuss later.

4.2 Using the Transitive Expansion

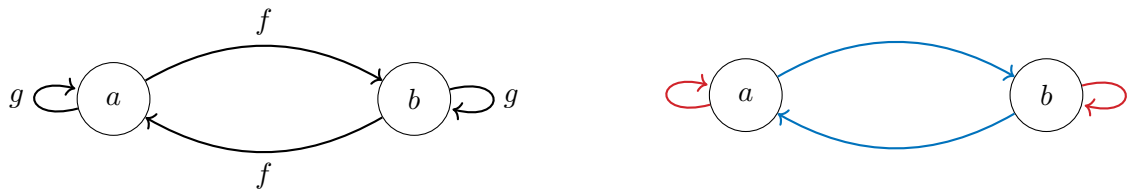
As a first remark we note that we only consider unary functions in this section. The key idea will be, to encode a function f as a family of relations, which then can capture the notion of nesting function applications. However, a bound on the alternation of different function symbols is necessary to ensure that the expanded signature is still finite, thus we will fixate a maximal alternation depth when discussing our new variant of RCR. Let us concretely define, how we expand the signature.

Definition 4.1 (Transitive expansion)**.** Let $\sigma \coloneqq \sigma_{\text{Rel}} \dot\cup \sigma_{\text{Func}}$ be a signature with relation symbols σ_{Rel} and unary function symbols σ_{Func} and let $\mathfrak{A}$ be a structure of signature σ with $|\mathfrak{A}| = n$. For readability, we define the family of sets of alternations of function applications $\text{Alters}_n^0(\sigma) \coloneqq \{\text{id}\}$ and

$$\begin{aligned}\text{Alters}_n^k(\sigma) \coloneqq \text{Alters}_n^{k-1}(\sigma) \cup \{ f_1^{m_1} f_2^{m_2} \dots f_k^{m_k} : & f_1, f_2, \dots, f_k \in \sigma_{\text{Func}} \\ & \wedge \forall i \in [k].m_i \in [n] \\ & \wedge \forall i \in [k-1].f_i \neq f_{i+1}\}.\end{aligned}$$

We will now fixate an arbitrary $k \in \mathbb{N}$ which will be our bound on the alternation depth and will define a new signature $\widetilde{\sigma}$ as well as a structure $\widetilde{\mathfrak{A}}$ of said signature, which will be the transitive expansion with alternation depth k of $\mathfrak{A}$. For $k \in \mathbb{N}$, $\alpha, \beta, \alpha_1, \dots, \alpha_\ell \in \text{Alters}_n^k(\sigma)$ and a $R \in \sigma_{\text{Rel}}$ with arity ℓ, we define the binary relation

$$\text{Eq}_{\alpha,\beta}^{\widetilde{\mathfrak{A}}} \coloneqq \{(a,b) : \alpha^{\mathfrak{A}}(a) = \beta^{\mathfrak{A}}(b)\},$$



(a) The structure $\mathfrak{A}$.

(b) The structure $\widetilde{\mathfrak{A}}$.

Figure 4: Graphical descriptions of $\mathfrak{A}$ and $\widetilde{\mathfrak{A}}$. The blue transitions represent the relations $\mathrm{Eq}_{\alpha,\beta}$ with $(\alpha,\beta) \in \{(\mathrm{id}, f), (f, \mathrm{id}), (f, f^2), (f, g), (f, g^2), (f^2, f), (g, f), (g^2, f)\}$, while the red transitions represent all other binary relations.

and the relation of arity ℓ

$$R^{\widetilde{\mathfrak{A}}}_{\alpha_1,\dots,\alpha_\ell} \coloneqq \{(a_1,\dots,a_\ell) : (\alpha_1^{\mathfrak{A}}(a_1),\dots,\alpha_\ell^{\mathfrak{A}}(a_\ell)) \in R^{\mathfrak{A}}\}.$$

We now define the transitive expansion with alternation depth k signature $\widetilde{\sigma}$, where

$$\begin{aligned}\widetilde{\sigma} \coloneqq &\{\mathrm{Eq}_{\alpha,\beta}/2 : \alpha,\beta \in \mathrm{Alters}^k_n(\sigma)\},\\ &\dot\cup\{R_{\alpha_1,\dots,\alpha_\ell}/\ell : R \in \sigma_{\mathrm{Rel}}, \mathrm{ar}(R) = \ell \text{ and } \alpha \in \mathrm{Alters}^k_n(\sigma)\}.\end{aligned}$$

Since the following definitions will depend on this construction, let us consider an example. We define the signature $\sigma = \{R, f, g\}$ where R is a unary relation symbol and f and g are unary function symbols. Now consider a σ structure $\mathfrak{A} = (A, \sigma)$ with $A = \{a, b\}$, $R^{\mathfrak{A}} = \{a\}$, $f^{\mathfrak{A}} = \{a \mapsto b, b \mapsto a\}$ and $g^{\mathfrak{A}} = \{a \mapsto a, b \mapsto b\}$. A graphical representation of $\mathfrak{A}$ can be found in Figure 4a. For the sake of simplicity, we will define the transitive expansion with alternation depth 1 and because $|\mathfrak{A}| = 2$ we will use $\mathrm{Alters}^1_2(\sigma)$ to do so. We see that $\mathrm{Alters}^1_2(\sigma) = \{\mathrm{id}, f, f^2, g, g^2\}$ and as such

$$\widetilde{\sigma} = \{R_{\mathrm{id}}, R_f, R_{f^2}, R_g, R_{g^2}, \mathrm{Eq}_{\mathrm{id},\mathrm{id}}, \mathrm{Eq}_{\mathrm{id},f}, \mathrm{Eq}_{\mathrm{id},f^2}, \dots, \mathrm{Eq}_{g^2,g^2}\}.$$

Because of the relatively large size of $\widetilde{\sigma}$, we will only give the formal definitions for a few relations, while the rest of the relations in $\widetilde{\mathfrak{A}}$ can be seen in Figure 4b. We find that $R^{\widetilde{\mathfrak{A}}}_{\mathrm{id}} = R^{\widetilde{\mathfrak{A}}}_{f^2} = R^{\widetilde{\mathfrak{A}}}_{g} = R^{\widetilde{\mathfrak{A}}}_{g^2} = \{a\}$ and that $R^{\widetilde{\mathfrak{A}}}_{f} = \{b\}$. Additionally, $\mathrm{Eq}^{\widetilde{\mathfrak{A}}}_{g,\mathrm{id}} = \mathrm{Eq}^{\widetilde{\mathfrak{A}}}_{g^2,\mathrm{id}} = \{(a,a),(b,b)\} = \mathrm{Eq}^{\widetilde{\mathfrak{A}}}_{\alpha,\alpha}$ for all $\alpha \in \mathrm{Alters}^1_2(\sigma)$. To give another example, we have $\mathrm{Eq}^{\widetilde{\mathfrak{A}}}_{g,f} = \mathrm{Eq}^{\widetilde{\mathfrak{A}}}_{g^2,f} = \{(a,b),(b,a)\}$. The definitions of all $\mathrm{Eq}^{\widetilde{\mathfrak{A}}}_{\alpha,\beta}$ can be found in Figure 4b.

We can now define RCR for signatures that include unary function symbols.

Definition 4.2 (RCR for structures with unary functions)**.** Let σ be a signature with relation and unary function symbols and let $\mathfrak{A}$ and $\mathfrak{B}$ be structures of signature σ. We say that $\mathfrak{A}$ and $\mathfrak{B}$ are being distinguished by RCR with alternation depth k (RCR_k), if $|\mathfrak{A}| \neq |\mathfrak{B}|$ or the transitive expansions with alternation depth k, $\widetilde{\mathfrak{A}}$ and $\widetilde{\mathfrak{B}}$, are being distinguished by RCR.

Just like nRCR, this algorithm allows for a logical characterisation, which will be investigated later.

4.3 Naive RCR versus RCR_k

We first want to discuss both of the given definitions and will see that RCR_k is stronger.

Consider the two structures $\mathfrak{A}$ and $\mathfrak{B}$ of signature $\sigma = \{f/1\}$ which can be seen in Figure 5. Formally they are defined as $\mathfrak{A} = (A, f^{\mathfrak{A}})$ and $\mathfrak{B} = (B, f^{\mathfrak{B}})$ where

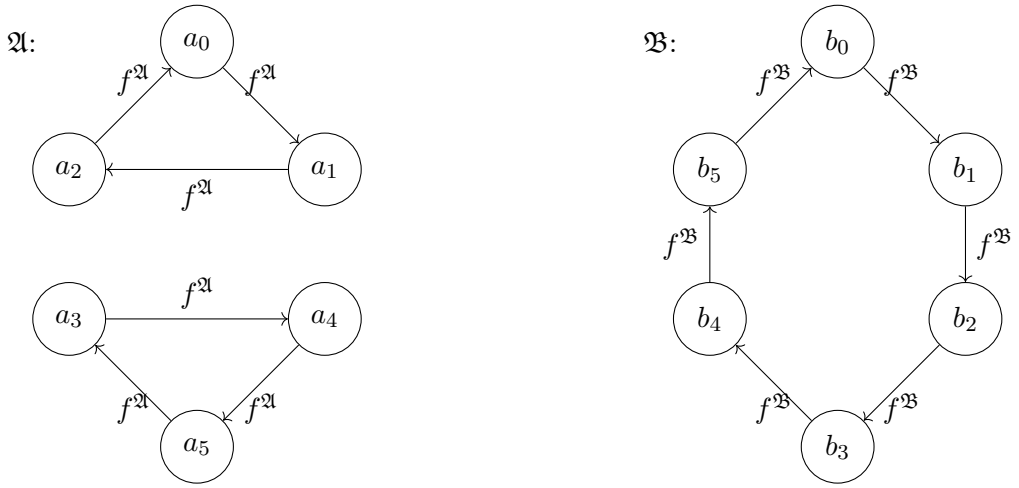


Figure 5: Two $\{f/1\}$-structures $\mathfrak{A}$ and $\mathfrak{B}$ which are not distinguished by naive RCR, but by RCR_k.

$$\begin{aligned} A =& \{a_0, a_1, a_2, a_3, a_4, a_5\}, & \qquad B =& \{b_0, b_1, b_2, b_3, b_4, b_5\}, \\ f^{\mathfrak{A}} =& \{a_0 \mapsto a_1, a_1 \mapsto a_2, a_2 \mapsto a_0, & \text{and} \qquad f^{\mathfrak{B}} =& \{b_0 \mapsto b_1, b_1 \mapsto b_2, b_2 \mapsto b_3, \\ & a_3 \mapsto a_4, a_4 \mapsto a_5, a_5 \mapsto a_3\} & & b_3 \mapsto b_4, b_4 \mapsto b_5, b_5 \mapsto b_0\} \end{aligned}$$

When applying nRCR to these structures, we find that they cannot be distinguished. This can also be seen when considering the encodings $\mathfrak{A}'$ and $\mathfrak{B}'$, which represent directed, regular graphs, which also cannot be distinguished by classical Colour Refinement.

We now want to see, whether RCR_1 distinguishes the structures $\mathfrak{A}$ and $\mathfrak{B}$ from Figure 5. First we compute $\widetilde{\sigma}$ as $\{\mathrm{Eq}_{f^i,f^j}, \mathrm{Eq}_{f^i,\mathrm{id}}, \mathrm{Eq}_{\mathrm{id},f^j} : 0 < i, j \leq 6\} \cup \{\mathrm{Eq}_{\mathrm{id},\mathrm{id}}\}$. For easier readability, we will only give the definitions for the symbols in $\{\mathrm{Eq}_{f^i,\mathrm{id}} : 0 < i \leq n\}$. We find that

$$\mathrm{Eq}^{\widetilde{\mathfrak{A}}}_{f^i,\mathrm{id}} = \{(a_j, a_{j+i \bmod 3}), (a_{j+3}, a_{(j+i \bmod 3)+3}) : j \in \{0,1,2\}\}$$

and

$$\mathrm{Eq}^{\widetilde{\mathfrak{B}}}_{f^i,\mathrm{id}} = \{(a_j, a_{j+i \mod 6}) : j \in \{0, ..5\}\}.$$

By using [19], we know that RCR distinguishes $\widetilde{\mathfrak{A}}$ and $\widetilde{\mathfrak{B}}$ if, and only if, there is a formula $\widetilde{\varphi} \in \mathsf{GF}(\mathsf{C})$ of signature $\widetilde{\sigma}$ that distinguishes them. Notice that $\mathrm{Eq}^{\widetilde{\mathfrak{A}}}_{f^3,\mathrm{id}} = \mathrm{Eq}^{\widetilde{\mathfrak{A}}}_{f^6,\mathrm{id}}$, $\mathrm{Eq}^{\widetilde{\mathfrak{A}}}_{f^1,\mathrm{id}} = \mathrm{Eq}^{\widetilde{\mathfrak{A}}}_{f^4,\mathrm{id}}$ and $\mathrm{Eq}^{\widetilde{\mathfrak{A}}}_{f^2,\mathrm{id}} = \mathrm{Eq}^{\widetilde{\mathfrak{A}}}_{f^5,\mathrm{id}}$, while only $\mathrm{Eq}^{\widetilde{\mathfrak{B}}}_{f^0,\mathrm{id}} = \mathrm{Eq}^{\widetilde{\mathfrak{B}}}_{f^6,\mathrm{id}}$. Therefore the sentence

$$\exists^{\geq 6}(x,y).\left(\mathrm{Eq}_{f^1,\mathrm{id}}(x,y) \wedge \mathrm{Eq}_{f^4,\mathrm{id}}(x,y)\right) \in \mathsf{GF}(\mathsf{C})$$

is satisfied by $\widetilde{\mathfrak{A}}$, but not $\widetilde{\mathfrak{B}}$. We can easily derive another formula $\varphi' \in \mathsf{GF}(\mathsf{C})$ to distinguish the transitive expansions, for example $\varphi' = \exists^{\geq 1} x.\, \mathrm{Eq}_{f^3,\mathrm{id}}(x,x)$, which expresses the existence of a 3-cycle.

We see that RCR_k distinguishes structures that were not distinguished by nRCR.

4.4 Characterisation Through Logic

In this section we will discuss, how the above approaches of Colour Refinement for non-relational structures can be characterised by logic. We will see that for both cases we will get a logic which extends the given definition of $\mathsf{GF}(\mathsf{C})$, see Definition 3.2, but restricts the usage of functions inside of terms.

4.4.1 Characterising Naive RCR Using Logic

The nRCR algorithm will be characterised by the nesting-free guarded fragment of counting logic $\mathsf{nfGF}(\mathsf{C})$. This logics forbids the nesting of functions and only allows the usage of them as $f(\mathbf{x}) = y$. This way, a function symbol acts like a relation symbol, which is analogous to the encoding used in nRCR. This leads to the characterisation of nRCR by $\mathsf{nfGF}(\mathsf{C})$, which we will prove in the following.

Definition 4.3 ($\mathsf{nfGF}(\mathsf{C})$)**.** Consider the definition of $\mathsf{GF}(\mathsf{C})$ given in 3.2. We obtain the nesting-free fragment, by allowing $f(\mathbf{x}) = y$ as a further atomic formula. Concretely, the only allowed atomic formulae are of the form $R(x_1, \ldots, x_\ell)$, $x = y$ and $f(x_1, \ldots, x_\ell) = y$, where f has arity ℓ.

The remaining definitions stay the same.

We now show that nRCR and $\mathsf{nfGF}(\mathsf{C})$ can distinguish exactly the same structures.

Theorem 4.1. *The two following statements are equivalent:*

1. *nRCR distinguishes $\mathfrak{A}$ and $\mathfrak{B}$.*
2. *There exists a sentence $\varphi \in \mathsf{nfGF}(\mathsf{C})$ such that $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \not\models \varphi$.*

Proof. 1. $\Rightarrow$ 2.: By definition, $\mathfrak{A}$ and $\mathfrak{B}$ are distinguished by nRCR if, and only if, $\mathfrak{A}'$ and $\mathfrak{B}'$ are distinguished by RCR. Using the result of [19], we obtain a sentence $\varphi' \in \mathsf{GF}(\mathsf{C})$ that distinguishes the encoded structures. Via a structural induction on the formula, we can now translate φ' into a formula $\varphi \in \mathsf{nfGF}(\mathsf{C})$. This can be achieved by replacing formulae $R_f(x_1, \ldots, x_\ell, y)$ by $f(x_1, \ldots, x_\ell) = y$ for function symbols $f \in \sigma$ and letting everything else stay the same.

2. $\Rightarrow$ 1.: When considering $\mathsf{nfGF}(\mathsf{C})$, one can find that the transformation done at the end of the first direction can be applied in reverse. This then leads to a distinguishing sentence in $\mathsf{GF}(\mathsf{C})$ and with [19] to a distinguishing colouring of the encoded structures, which by definition is a distinguishing colouring for the structures themselves. □

While the above theorem results in a nice characterisation of the naive encoding, the nesting of terms is often very desired when using functions. However, it can be shown that nesting is too powerful for the naive encoding.

Consider the structures $\mathfrak{A}$ and $\mathfrak{B}$ from Figure 5 and the formula $\varphi = \exists^{\geq 1} x.(f(f(f(x))) = x)$ which utilizes term nesting to find a cycle of length three. It is obvious that $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \not\models \varphi$. However, as it was discussed above, one finds that nRCR cannot distinguish them. Therefore, term nesting is too powerful for the naive encoding.

4.4.2 Logical Characterisation of RCR_k

A first idea that may come to mind when looking at the definition of the transitive expansion, is to use the classical notion of atomic formula for guards, fixate a maximal alternation depth for terms and only allow $|\mathfrak{A}|$ applications of the same function symbol in series. Concretely, for a function symbol f and a term $s(x)$, only terms of the form $f^m(s(x))$ where $m \leq |\mathfrak{A}|$ are allowed. However, we prove that we can allow a term $f^m(s(x))$ for an arbitrary $m \in \mathbb{N}$, while the bounded alternation depth is still needed. The reason why this is possible, hinges on the pigeonhole principle. When considering $f(x)$, $f^2(x)$, $f^3(x)$ and so forth, until $f^m(x)$, where $m > |\mathfrak{A}|$, there have to be to numbers i and j, such that $f^i(x) = f^j(x)$. Therefore, we can decompose the path into a path to a cycle, the cycle itself, and a last part of that cycle. To allow the following proofs to be more readable, we first want to define the set of all such valid decompositions.

Let

$$\begin{aligned} \mathcal{I}(n, m) = \{(k, \ell, p) \in [n]^3 \quad : \quad & k + p < k + \ell \leq n \wedge \\ & k + r \cdot \ell + p = m \text{ for some } r \in \mathbb{N}\}. \end{aligned}$$

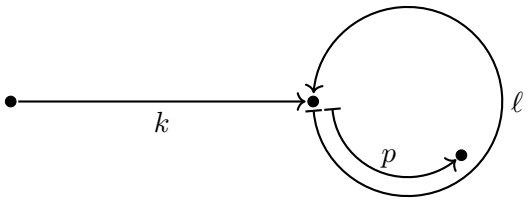


Figure 6: A description of how a path can be decomposed into a cycle, the path to it and a last part of it.

This set will represents all the possible ways, to decompose a path into a cycle and the path to and from it. This means, that the triple (k, ℓ, p) will represent a path, that has a beginning part of length k, then a cycle of length ℓ and a last part that consists of the first p elements of the cycle. One can see that in a structure $\mathfrak{A}$ with a unary function f and n elements, any path along of f with length $m > n$ can be decomposed into a triple in the set $\mathcal{I}(n, m)$. A graphical description of such a triple (k, ℓ, p) can be found in Figure 6.

In the beginning we remarked that we have to fixate an alternation depth. This bound can be seen in the definition of the transitive expansion and will be used in the logic that will characterise the Colour Refinement algorithm. Therefore we can only reason about a fragment of $\mathsf{GF}(\mathsf{C})$, where the terms do not alternate too often. This is formally stated in the following definition.

Definition 4.4 (Alternation bounded $\mathsf{GF}(\mathsf{C})$)**.** For a $k \in \mathbb{N}$, we define the fragment of $\mathsf{GF}(\mathsf{C})$ with a bounded alternation depth of k ($\mathsf{GF}(\mathsf{C})_k$) as $\mathsf{GF}(\mathsf{C})$ with the constraint that for all formulae $\varphi \in \mathsf{GF}(\mathsf{C})_k$ of signature σ and every term t that appears in φ, there is an $n \in \mathbb{N}$ and an $\alpha \in \mathrm{Alters}_n^k(\sigma)$ such that $\alpha = t$. Atomic formulae are defined as usual, that is, the formulae $R(t_1(x_1), t_2(x_2), \ldots, t_n(x_n))$ and $t_1(x_1) = t_2(x_2)$ for terms $t_1, t_2, \ldots, t_n$ and variables $x_1, x_2, \ldots, x_n$ are atomic formulae.

With this, we can prove the first result, which allows us to use every $f^m(x) = y$ for every $m \in \mathbb{N}$ in a formula.

Lemma 4.1. *Let $\psi(x_1, x_2) \in \mathsf{GF}(\mathsf{C})_1$ be of the form $f^m(x_1) = x_2$. Then there exists a formula $\vartheta(x_1, x_2) \in \mathsf{GF}(\mathsf{C})_1$ such that for any $\mathfrak{A}$ with $|\mathfrak{A}| = n$ it holds*

$$\mathfrak{A}, a_1, a_2 \models \psi(x_1, x_2) \text{ if, and only if, } \mathfrak{A}, a_1, a_2 \models \vartheta(x_1, x_2)$$

and for any $f^{m'}(x)$ that appears in ϑ we have $m' \leq n$. Furthermore, $\vartheta(x_1, x_2)$ is of the form $\bigvee \Phi(x_1, x_2)$, and if $\mathfrak{A}, a_1, a_2 \models \vartheta(x_1, x_2)$, then there is exactly one $\varphi(x_1, x_2) \in \Phi$, such that $\mathfrak{A}, a_1 \models \exists^{\geq 1} x_2.\varphi(x_1, x_2)$.

Proof. If $m \leq n$, we let $\vartheta \coloneqq \psi$ and the claim follows.

Otherwise, we define

$$\vartheta(x_1, x_2) \coloneqq \bigvee_{(k,\ell,p) \in \mathcal{I}(n,m)} \zeta_{(k,\ell,p)}(x_1, x_2)$$

where

$$\begin{aligned} \zeta_{(k,\ell,p)}(x_1, x_2) \coloneqq & f^{k+p}(x_1) = x_2 \wedge f^k(x_1) = f^{k+\ell}(x_1) \\ & \wedge \mathrm{E}_f^{k,\ell}(x_1) \\ & \wedge \bigwedge_{\ell' < \ell} f^k(x_1) \neq f^{k+\ell'}(x_1) \end{aligned}$$

and for some term $t(x_1)$ we have

$$\mathrm{E}_f^{k,\ell}(t(x_1)) = \begin{cases} \top & \text{if } k = 0 \\ f^{k-1}(t(x_1)) \neq f^{k-1+\ell}(t(x_1)) & \text{otherwise.} \end{cases}$$

Due to the definition of $\mathcal{I}(n,m)$ it is obvious that only $f^{m'}$ with $m' \leq n$ appears. We now proceed to the proof of the equivalence. For the purpose of readability, we will write $f_{\mathfrak{A}}$ instead of $f^{\mathfrak{A}}$.

We will show that if $\mathfrak{A}, a_1, a_2 \models \vartheta(x_1, x_2)$, then $\mathfrak{A}, a_1, a_2 \models \psi(x_1, x_2)$. Let $\mathfrak{A}, a_1, a_2 \models \vartheta(x_1, x_2)$. By definition of ϑ, there are $(k, \ell, p) \in \mathcal{I}(n, m)$ with $\mathfrak{A}, a_1, a_2 \models \zeta_{(k,\ell,p)}(x_1, x_2)$. In particular $f^k_{\mathfrak{A}}(a_1) = f^{k+\ell}_{\mathfrak{A}}(a_1)$. It follows that

$$f^k_{\mathfrak{A}}(a_1) = f^{k+\ell}_{\mathfrak{A}}(a_1) = f^{k+2\ell}_{\mathfrak{A}}(a_1) = f^{k+3\ell}_{\mathfrak{A}}(a_1) = \cdots = f^{k+r\cdot\ell}_{\mathfrak{A}}(a_1)$$

for all $r \in \mathbb{N}$. By using the definition of $\mathcal{I}(n, m)$, we get

$$a_2 = f^{k+p}_{\mathfrak{A}}(a_1) = f^{k+r\cdot\ell+p}_{\mathfrak{A}}(a_1) = f^m_{\mathfrak{A}}(a_1).$$

From this we can deduce $\mathfrak{A}, a_1, a_2 \models \psi(x_1, x_2)$, where $\psi(x_1, x_2)$ has the form $f^m(x_1) = x_2$.

Now we prove that if $\mathfrak{A}, a_1, a_2 \models \psi(x_1, x_2)$, then $\mathfrak{A}, a_1, a_2 \models \vartheta(x_1, x_2)$. Let $\mathfrak{A}, a_1, a_2 \models \psi(x_1, x_2)$. By assumption $m > n$ and by the pigeonhole principle there have to be distinct i and j such that $f^i_{\mathfrak{A}}(a_1) = f^j_{\mathfrak{A}}(a_1)$. Choose such i, j such that they are lexicographically minimal. Now choose $k \coloneqq i$, $\ell \coloneqq j - i$ and $p \coloneqq (m - i) \mod (j - i) = (m - i) \mod \ell$. Obviously $(k, \ell, p) \in \mathcal{I}(n, m)$ and what remains to be shown is that $\mathfrak{A}, a_1, a_2 \models \zeta_{(k,\ell,p)}(x_1, x_2)$. For that, we consider the parts of the conjunction and show for each one that it is satisfied.

- $f^{k+p}(x_1) = x_2$ is satisfied. We use the fact that $a = b \mod c \Leftrightarrow b = r \cdot c + a$ for some $r \in \mathbb{N}$. Then
$$f^{k+p}_{\mathfrak{A}}(a_1) = f^{i+(m-i)-r\cdot\ell}_{\mathfrak{A}}(a_1) = f^{i+r\cdot\ell+m-i-r\cdot\ell}_{\mathfrak{A}}(a_1) = f^m_{\mathfrak{A}}(a_1) = a_2.$$
Therefore $\mathfrak{A}, a_1, a_2 \models f^{k+p}(x_1) = x_2$.
- $f^k(x_1) = f^{k+\ell}(x_1)$ is satisfied. Consider that
$$f^k_{\mathfrak{A}}(a_1) = f^i_{\mathfrak{A}}(a_1) = f^j_{\mathfrak{A}}(a_1) = f^{j+i-i}_{\mathfrak{A}}(a_1) = f^{i+j-i}_{\mathfrak{A}}(a_1) = f^{k+\ell}_{\mathfrak{A}}(a_1).$$
This leads to $\mathfrak{A}, a_1, a_2 \models f^k(x_1) = f^{k+\ell}(x_1)$.
- $\mathrm{E}^{k,\ell}_f(x_1)$ is satisfied. Otherwise $f^{k-1}_{\mathfrak{A}}(a_1) = f^{k-1+\ell}_{\mathfrak{A}}(a)$, but then $(k-1, \ell)$ would be lexicographically smaller than (i, j).
- The same reasoning applies to $\bigwedge_{\ell'<\ell} f^k(x_1) \neq f^{k+\ell'}(x_1)$. If it weren't satisfied, there would be a (i, j') with $j' < j$ and $f^i_{\mathfrak{A}}(a_1) = f^{i+j'}_{\mathfrak{A}}(a_1)$ which would be lexicographically smaller than (i, j).

Thus we have shown that every subformula of the conjunction and therefore the formula is being fulfilled.

Lastly, it remains to prove that if ϑ is satisfied, then there is exactly one $(k, \ell, p) \in \mathcal{I}(n, m)$ such that $\exists^{\geq 1} x_2.\zeta_{(k,\ell,p)}(x_1, x_2)$ is fulfilled. We prove this by contradiction. Assume that $\mathfrak{A}, a_1, a_2 \models \vartheta(x_1, x_2)$ and that there are $\zeta_{(k,\ell,p)}(x_1, x_2)$ and $\zeta_{(k',\ell',p')}(x_1, x_2)$ with $(k, \ell, p) \neq (k', \ell', p')$, such that $\mathfrak{A}, a_1 \models \exists^{\geq 1} x_2.\zeta_{(k,\ell,p)}(x_1, x_2)$ and $\mathfrak{A}, a_1 \models \exists^{\geq 1} x_2.\zeta_{(k',\ell',p')}(x_1, x_2)$.

We proceed with a case distinction. Let $k = k'$ and $\ell = \ell'$. Then there are $r, r' \in \mathbb{N}$ such that

$$k + r \cdot \ell + p = k' + r' \cdot \ell' + p' = m.$$

Thus we can infer that $r \cdot \ell + p = r' \cdot \ell' + p'$. By definition of $\mathcal{I}(n, m)$ we know that $p, p' < \ell = \ell'$ and as such

$$r \cdot \ell + p, r' \cdot \ell' + p' \in \{r \cdot \ell, r \cdot \ell + 1, \ldots, r \cdot \ell + (\ell - 1)\}$$

and because p is a non-negative integer, $r = r'$ has to follow and further we get $p = p'$. However this would contradict that $(k, \ell, p) \neq (k', \ell', p')$. Now assume that $\ell \neq \ell'$ and without loss of generality assume that $\ell < \ell'$. But then $\mathfrak{A}, a_1 \not\models \bigwedge_{\hat{\ell}<\ell'} f^{k'}(x_1) \neq f^{k'+\hat{\ell}}(x_1)$, because

$$f_{\mathfrak{A}}^{k'+\ell}(a_1) = f_{\mathfrak{A}}^{k+\ell} = f_{\mathfrak{A}}^{k}(a_1) = f_{\mathfrak{A}}^{k'}(a_1)$$

and $k' + \ell < k' + \ell'$. Thus this cannot be the case as well.

Consider that $k \neq k'$ and without loss of generality assume that $k < k'$. If $\ell = \ell'$, then by the principle of induction, we get that $f_{\mathfrak{A}}^{k}(a_1) = f_{\mathfrak{A}}^{k+\ell}(a_1)$, $f_{\mathfrak{A}}^{k+1}(a_1) = f_{\mathfrak{A}}^{k+1+\ell}(a_1)$ and then $f_{\mathfrak{A}}^{k'}(a_1) = f_{\mathfrak{A}}^{k'+\ell'}(a_1)$. But this contradicts $\mathfrak{A}, a_1 \models \mathrm{E}_f^{k',\ell'}(x_1)$. If $\ell < \ell'$, then

$$f_{\mathfrak{A}}^{k'}(a_1) = f_{\mathfrak{A}}^{k+(k'-k)}(a_1) = f_{\mathfrak{A}}^{k+(k'-k)+\ell}(a_1) = f_{\mathfrak{A}}^{k'+\ell}(a_1),$$

but this again contradicts $\mathfrak{A}, a_1 \models \bigwedge_{\hat{\ell}<\ell'} f^{k'}(x_1) \neq f^{k'+\hat{\ell}}(x_1)$. If $\ell' < \ell$, then there exists a $t \in \mathbb{N}$ such that

$$k + t \cdot \ell < k' \leq k + (t+1) \cdot \ell.$$

We now define $r \coloneqq k + (t+1) \cdot \ell - k'$ and get $f_{\mathfrak{A}}^{k'+r}(a_1) = f_{\mathfrak{A}}^{k'+r+\ell'}$ and by using $f_{\mathfrak{A}}^{k'+r}(a_1) = f_{\mathfrak{A}}^{k+(t+1)\cdot\ell}(a_1) = f_{\mathfrak{A}}^{k}(a_1)$ it follows that $f_{\mathfrak{A}}^{k}(a_1) = f_{\mathfrak{A}}^{k+\ell'}(a_1)$. This contradicts $\mathfrak{A}, a_1 \models \bigwedge_{\hat{\ell}<\ell} f^{k}(x_1) \neq f^{k'+\hat{\ell}}(x_1)$.

One can see that we did not use x_2 or a_2. Therefore its interpretation is irrelevant, which is why we can existentially quantify it in the claim. As all possible cases lead to a contradiction, the first assumption cannot be true and we proved the claim. □

The above proof allows for the translation of a formula $f^m(x) = y$ to a formula $\vartheta(x, y)$ that is equivalent for structures with n elements. A natural extension would be, to allow alternation of functions, for example formulae like $g^m(f^{m'}(x)) = y$. This is also possible and will be proved in the following.

Lemma 4.2. *Let $d \in \mathbb{N}$ and $\psi(x_1, x_2) \in \mathsf{GF}(\mathsf{C})_d$ be of the form $t(x_1) = x_2$ for a term t. Then there exists a formula $\vartheta_t(x_1, x_2) \in \mathsf{GF}(\mathsf{C})_d$, such that for any structure $\mathfrak{A}$ with $|\mathfrak{A}| = n$ it holds*

$$\mathfrak{A}, a_1, a_2 \models \psi(x_1, x_2) \text{ if, and only if, } \mathfrak{A}, a_1, a_2 \models \vartheta_t(x_1, x_2).$$

Furthermore, $\vartheta_t(x_1, x_2)$ is of the form $\bigvee \Phi(x_1, x_2)$ where all $\varphi(x_1, x_2) \in \Phi(x_1, x_2)$ are of the form

$$t'(x_1) = x_2 \wedge \bigwedge \Psi(x_1)$$

for some term $t'(x_1)$, and for every function symbol f in the signature, there does not appear a term of the form $f^m(s(x))$ where $m > n$. Additionally, if $\mathfrak{A}, a_1, a_2 \models \vartheta_t(x_1, x_2)$, then there is exactly one $\varphi \in \Phi$, such that $\mathfrak{A}, a_1 \models \exists^{\geq 1} x_2.\varphi(x_1, x_2)$.

Proof. We prove this via an induction on the term $t(x_1)$.

Base case: If $t(x_1)$ is of the form $f^m(x_1)$ for a unary function symbol f and $m \in \mathbb{N}$, we use the formula constructed in the proof of Lemma 4.1. It can easily be verified that it is in the correct form and from the same proof we get that if the translated formula is fulfilled, exactly one subformula of the disjunction is satisfied.

Inductive step: Assume that $t(x_1)$ is of the form $g^m(s(x_1))$ for a unary function symbol g, $m \in \mathbb{N}$ and term s. By the induction hypothesis, there is a formula $\vartheta_s(x_1, x_2) \in \mathsf{GF}(\mathsf{C})_{d-1}$ of the form $\bigvee \Phi_s(x_1, x_2)$ defined above with $\mathfrak{A}, a_1, a_2 \models s(x_1) = x_2$ if, and only if, $\mathfrak{A}, a_1, a_2 \models \vartheta_s(x_1, x_2)$.

If $m \leq n$, we set $\vartheta_t(x_1, x_2)$ to

$$\bigvee \Phi'(x_1, x_2),$$

where $\Phi'(x_1,x_2) \coloneqq \{g^m(t'(x_1)) = x_2 \wedge \bigwedge \Psi(x_1) : t'(x_1) = x_2 \wedge \bigwedge \Psi(x_1) \in \Phi_s(x_1,x_2)\}$.

If $m > n$, then we set $\vartheta_t(x_1,x_2)$ to

$$\bigvee_{(k,\ell,p)\in\mathcal{I}(n,m)} \bigvee \Phi'_{(k,\ell,p)}(x_1,x_2),$$

where

$$\begin{aligned}\Phi'_{(k,\ell,p)} \coloneqq \{&g^{k+p}(t'(x_1)) = x_2 \wedge g^k(t'(x_1)) = g^{k+l}(t'(x_1))\\ &\wedge \mathrm{E}_g^{k,l}(t'(x_1)) \wedge \bigwedge_{\ell'<\ell} g^k(t'(x_1)) \neq g^{k+\ell'}(t'(x_1))\\ &\wedge \Psi(x_1) : t'(x_1) = x_2 \wedge \bigwedge \Psi(x_1) \in \Phi_s(x_1,x_2)\}\end{aligned}$$

By using the above definitions, we get $\mathfrak{A}, a_1, a_2 \models s(x_1) = x_2$ if, and only if, $\mathfrak{A}, a_1, a_2 \models \varphi_s(x_1,x_2)$ for some $\varphi_s \in \Phi_s$ where $\varphi_s(x_1,x_2)$ is of the form $t'(x_1) = x_2 \wedge \bigwedge\ \Psi(x_1)$. Therefore

$$\mathfrak{A}, a_1, a_2 \models s(x_1) = x_2 \text{ if, and only if, } \mathfrak{A}, a_1, a_2 \models t'(x_1) = x_2 \wedge \bigwedge \Psi(x_1). \qquad (\star)$$

We now prove that

$$\mathfrak{A}, a_1, a_2 \models t(x_1) = x_2 \text{ if, and only if, } \mathfrak{A}, a_1, a_2 \models \vartheta_t(x_1,x_2).$$

Assume $m \leq n$. Let $\mathfrak{A}, a_1, a_2 \models \vartheta_t$. Then there is some $\varphi(x_1,x_2)$ of the form $g^m(t'(x_1)) = x_2 \wedge \bigwedge \Psi(x_1)$ such that $\mathfrak{A}, a_1, a_2 \models \varphi(x_1,x_2)$. We then get

$$\begin{aligned}&\mathfrak{A}, a_1, a_2 \models g^m(t'(x_1)) = x_2 \wedge \bigwedge \Psi(x_1)\\ \Leftrightarrow\ &\mathfrak{A}, a_1, a_2, a_3 \models g^m(x_3) = x_2 \wedge \bigwedge \Psi(x_1) \wedge t'(x_1) = x_3 \text{ for some } a_3 \in A\\ \overset{(\star)}{\Leftrightarrow}\ &\mathfrak{A}, a_1, a_2, a_3 \models g^m(x_3) = x_2 \wedge s(x_1) = x_3 \text{ for some } a_3 \in A\\ \Leftrightarrow\ &\mathfrak{A}, a_1, a_2 \models g^m(s(x_1)) = x_2.\end{aligned}$$

Now let $m > n$. Then there is a

$$\begin{aligned}\varphi(x_1,x_2) \coloneqq\ &g^{k+p}(t'(x_1)) = x_2 \wedge g^k(t'(x_1)) = g^{k+l}(t'(x_1))\\ &\wedge \mathrm{E}_g^{k,l}(t'(x_1)) \wedge \bigwedge_{\ell'<\ell} g^k(t'(x_1)) \neq g^{k+\ell'}(t'(x_1))\\ &\wedge \bigwedge \Psi(x_1)\end{aligned}$$

for some $(k,\ell,p) \in \mathcal{I}(n,m)$ with $\mathfrak{A}, a_1, a_2 \models \varphi(x_1,x_2)$. And now

$$\begin{aligned}&\mathfrak{A}, a_1, a_2 \models \varphi(x_1,x_2)\\ \Leftrightarrow\ &A, a_1, a_2, a_3 \models g^{k+p}(x_3) = x_2 \wedge g^k(x_3) = g^{k+l}(x_3)\\ &\qquad \wedge \mathrm{E}_g^{k,l}(x_3) \wedge \bigwedge_{\ell'<\ell} g^k(x_3) \neq g^{k+\ell'}(x_3)\\ &\qquad \wedge \bigwedge \Psi(x_1) \wedge t'(x_1) = x_3 \text{ for some } a_3 \in A\\ \overset{4.1}{\Leftrightarrow}\ &\mathfrak{A}, a_1, a_2, a_3 \models g^m(x_3) = x_2 \wedge t'(x_1) = x_3 \wedge \bigwedge \Psi(x_1) \text{ for some } a_3 \in A\\ \overset{(\star)}{\Leftrightarrow}\ &\mathfrak{A}, a_1, a_2, a_3 \models g^m(x_3) = x_2 \wedge s(x_1) = x_3 \text{ for some } a_3 \in A\\ \Leftrightarrow\ &\mathfrak{A}, a_1, a_2 \models g^m(s(x_1)) = x_2.\end{aligned}$$

The other direction follows in both cases, as only equivalent steps have been used and it is obvious that the disjunction of a set is being fulfilled, if a formula of the set is satisfied.

Lastly, we show that if $\mathfrak{A}, a_1, a_2 \models \vartheta_t(x_1, x_2)$, where ϑ_t is of the form $\bigvee \Phi$, there is exactly one $\varphi \in \Phi$, such that $\mathfrak{A}, a_1 \models \exists^{\geq 1} x_2.\varphi(x_1, x_2)$. As in the proof of Lemma 4.1, we are going to use a proof by contradiction and we will look at the cases where $m \leq n$ and $m > n$ separately. If $m \leq n$, assume that $\mathfrak{A}, a_1, a_2 \models \vartheta_t(x_1, x_2)$ and that there are $\varphi_1, \varphi_2 \in \Phi'(x_1, x_2)$ with $\varphi_1 \neq \varphi_2$, $\mathfrak{A}, a_1 \models \exists^{\geq 1} x_2.\varphi_1(x_1, x_2)$ and $\mathfrak{A}, a_1 \models \exists^{\geq 1} x_2.\varphi_2(x_1, x_2)$. It is easy to see that

$$\mathfrak{A}, a_1, a_2, a_3 \models g^m(t'_1(x_1)) = x_2 \wedge \bigwedge \Psi_1(x_1) \wedge g^m(t'_2(x_1)) = x_3 \wedge \bigwedge \Psi_2(x_1)$$

for some a_2 and a_3, which is equivalent to

$$\begin{aligned} \mathfrak{A}, a_1, a_2, a_3, a_4, a_5 \models \quad & g^m(x_4) = x_2 \wedge t'_1(x_1) = x_4 \wedge \bigwedge \Psi_1(x_1) \\ & \wedge g^m(x_5) = x_3 \wedge t'_2(x_1) = x_4 \wedge \bigwedge \Psi_2(x_1) \end{aligned}$$

for some a_4 and a_5. However, $t'_1(x_1) = x_2 \wedge \bigwedge \Psi_1(x_1), t'_2(x_1) = x_s \wedge \bigwedge \Psi_2(x_1) \in \Phi_s$ and thus there would be two different $\psi_1(x_1, x_2), \psi_2(x_1, x_2) \in \Phi_s$ such that $\mathfrak{A}, a_1 \models \exists^{\geq 1} x_2.\psi_1(x_1, x_2)$ and $\mathfrak{A}, a_1 \models \exists^{\geq 1} x_2.\psi_2(x_1, x_2)$. But this contradicts the induction hypothesis that Φ_s conforms to the properties stated in the lemma.

If $m > n$, we again assume that $\mathfrak{A}, a_1, a_2 \models \vartheta_t(x_1, x_2)$ and that there are $\varphi_1(x_1, x_2) \in \Phi'_{(k,\ell,p)}(x_1, x_2)$ and $\varphi_2(x_1, x_2) \in \Phi'_{(k',\ell',p')}(x_1, x_2)$ with $\varphi_1 \neq \varphi_2$, $\mathfrak{A}, a_1 \models \exists^{\geq 1} x_2.\varphi_1(x_1, x_2)$ and $\mathfrak{A}, a_1 \models \exists^{\geq 1} x_2.\varphi_2(x_1, x_2)$. By looking at the structure of the formulae as they are defined in this proof and by substituting terms and variables like in the first case, we again find that

$$\mathfrak{A}, a_1, a_3, a_4 \models t'_1(x_1) = x_3 \wedge \bigwedge \Psi_1(x_1) \wedge t'_2(x_1) = x_4 \wedge \bigwedge \Psi_2(x_1),$$

where $t'_1(x_1) = x_2 \wedge \bigwedge \Psi_1(x_1), t'_2(x_1) = x_2 \wedge \bigwedge \Psi_2(x_1) \in \Phi_s$. By using the same arguments as before, we as well arrive at a contradiction. As such, the assumption must be false and we have finished the proof. □

Finally, we want to be able to use terms inside relations. This is also possible and will be shown in the following lemma.

Lemma 4.3. *Let $d \in \mathbb{N}$ and $\psi(x_1, \ldots, x_m) \coloneqq R(t_1(x_1), \ldots, t_m(x_m)) \in \mathsf{GF}(\mathsf{C})_d$ be an atomic formula. Then there exists a formula $\vartheta_\psi \in \mathsf{GF}(\mathsf{C})_d$, such that for any given structure (of fitting signature) $\mathfrak{A}$ with $|\mathfrak{A}| = n$ it holds*

$$\mathfrak{A}, a_1, \ldots, a_m \models \psi(x_1, \ldots, x_m) \text{ if, and only if, } \mathfrak{A}, a_1, \ldots, a_m \models \vartheta_\psi(x_1, \ldots, x_m).$$

Furthermore, $\vartheta_\psi(x_1, \ldots, x_m)$ is of the form $\bigvee \Phi(x_1, \ldots, x_m)$ where all $\varphi \in \Phi$ are of the form

$$R(t'_1(x_1), \ldots, t'_m(x_m)) \wedge \bigwedge \Psi_1(x_1) \wedge \cdots \wedge \bigwedge \Psi_m(x_m),$$

and for every $f^m(s(x))$ that appear in ϑ_ψ, where f is a unary function symbol and s is a term we have $m \leq n$. Additionally, if $\mathfrak{A}, a_1, \ldots, a_m \models \vartheta_\psi(x_1, \ldots, x_m)$, then there exists exactly one $\varphi(x_1, \ldots, x_m) \in \Phi(x_1, \ldots, x_m)$, such that $\mathfrak{A}, a_1, \ldots, a_m \models \varphi(x_1, \ldots, x_m)$.

Proof. Let $\mathfrak{A}, a_1, \ldots, a_m \models \psi(x_1, \ldots, x_m)$. This is equivalent to

$$\mathfrak{A}, a_1, \ldots, a_m, b_1, \ldots, b_m \models R(b_1, \ldots, b_m) \wedge t_1(x_1) = b_1 \wedge \cdots \wedge t_m(x_m) = b_m$$

for some $b_1, \dots, b_m \in A$. By applying the previous lemma, we get the equivalent statement

$$\begin{aligned}\mathfrak{A}, a_1, \dots, a_m, b_1, \dots, b_m \models R(y_1, \dots, y_m) \wedge & \bigvee_{i_1} \left(t'_{1,i_1}(x_1) = y_1 \wedge \bigwedge \Psi_{1,i_1}(x_1) \right) \\ & \wedge \dots \\ & \wedge \bigvee_{i_m} \left(t'_{m,i_m}(x_m) = y_m \wedge \bigwedge \Psi_{m,i_m}(x_m) \right).\end{aligned}$$

Through distribution of boolean formulae we get

$$\begin{aligned}\mathfrak{A}, a_1, \dots, a_m, b_1, \dots, b_m \models \bigvee_{i_1} \cdots \bigvee_{i_m} (R(y_1, \dots, y_m) \wedge & t'_{1,i_1}(x_1) = y_1 \wedge \bigwedge \Psi_{1,i_1}(x_1) \\ & \wedge \dots \\ & \wedge t'_{m,i_m}(x_m) = y_m \wedge \bigwedge \Psi_{m,i_m}(x_m)).\end{aligned} \tag{1}$$

Finally, we can resubstitute variables and get

$$\begin{aligned}\mathfrak{A}, a_1, \dots, a_m \models \bigvee_{i_1} \cdots \bigvee_{i_m} (R(t'_{1,i_1}(x_1), \dots, t'_{m,i_m}(x_m)) & \\ \wedge \bigwedge \Psi_{1,i_1}(x_1) & \\ \wedge \dots & \\ \wedge \bigwedge \Psi_{m,i_m}(x_m)) =: \vartheta_\psi(x_1, \dots, x_m). &\end{aligned}$$

One can see that ϑ_ψ is of the correct form. The equality follows from the fact that only equivalences have been used to derive ϑ_ψ from ψ.

Lastly, we prove that if ϑ_ψ is satisfied, there is exactly one formula of the disjunction that is satisfied. For this, consider the equivalent formula from Equation (1). Assume that $\mathfrak{A}, a_1, \dots, a_m \models \vartheta_\psi$ and that there are two subformulae φ_1 and φ_2 of the formula in Equation (1), where φ_1 is of the form

$$\begin{aligned}R(y_1, \dots, y_m) \wedge & t'_{1,i_1}(x_1) = y_1 \wedge \bigwedge \Psi_{1,i_1}(x_1) \\ & \wedge \dots \\ & \wedge t'_{m,i_m}(x_m) = y_m \wedge \bigwedge \Psi_{m,i_m}(x_m)\end{aligned}$$

and φ_2 is of the form

$$\begin{aligned}R(y_1, \dots, y_m) \wedge & s'_{1,i_1}(x_1) = y_1 \wedge \bigwedge \Psi'_{1,i_1}(x_1) \\ & \wedge \dots \\ & \wedge s'_{m,i_m}(x_m) = y_m \wedge \bigwedge \Psi'_{m,i_m}(x_m),\end{aligned}$$

such that $\varphi_1 \neq \varphi_2$, $\mathfrak{A}, a_1, \dots, a_m, b_1, \dots, b_m \models \varphi_1$ and $\mathfrak{A}, a_1, \dots, a_m, b_1, \dots, b_m \models \varphi_2$. As $\varphi_1 \neq \varphi_2$, there must be a j such that ψ_1 is of the form $t'_{j,i_j}(x_j) = y_j \wedge \bigwedge \Psi_{j,i_j}(x_j)$, ψ_2 is of the form $s'_{j,i_j}(x_j) = y_j \wedge \bigwedge \Psi'_{j,i_j}(x_j)$ and $\psi_1 \neq \psi_2$. From the construction of the formula we know, that there is a term t_j, a formula ϑ_{t_j} of the form $\bigvee \Phi_{t_j}$ and $\psi_1, \psi_2 \in \Phi_{t_j}$. However, $\mathfrak{A}, a_j \models \exists^{\geq 1} y_j.\psi_1(x_j, y_j)$ and $\mathfrak{A}, a_j \models \exists^{\geq 1} y_j.\psi_2(x_j, y_j)$ would contradict the claim that has been proved in Lemma 4.2. □

To illustrate how this translation works, let us consider the formula ψ of the form $g^4(f^3(x)) = y$ for a structure with 2 elements. As in the proof, we inductively translate the inner terms and as such get for the formula $f^3(x) = y$, the formula φ of the form

$$\bigvee_{(k,\ell,p) \in \mathcal{I}(2,3)} \left(f^k(x) = f^{k+\ell}(x) \wedge f^{k+p}(x) = y \wedge \mathrm{E}_f^{k,l}(x) \wedge \bigwedge_{\hat{\ell} < \ell} f^k(x) \neq f^{k+\hat{\ell}}(x) \right)$$

and with $\mathcal{I}(2,3) = \{(0,2,1),(1,1,0),(0,1,0)\}$ we get that φ equals

$$\begin{aligned}&\Big(x = f^2(x) \wedge f(x) = y \wedge x \neq f(x)\Big)\\ \vee &\Big(f(x) = f^2(x) \wedge f(x) = y \wedge x \neq f(x)\Big)\\ \vee &\left(x = f(x) \wedge x = y \wedge \top\right).\end{aligned}$$

Now we can construct ϑ_ψ from ψ. From the proof, we know that ϑ_ψ is of the form

$$\begin{aligned}\bigvee_{(k',\ell',p')\in\mathcal{I}(2,4)} \bigvee_{(k,\ell,p)\in\mathcal{I}(2,3)} &\left(f^k(x) = f^{k+\ell}(x) \wedge \mathrm{E}_f^{k,l}(x) \wedge \bigwedge_{\hat{\ell}<\ell} f^k(x) \neq f^{k+\hat{\ell}}(x)\right.\\ &\wedge g^{k'}(f^{k+p}(x)) = g^{k'+\ell'}(f^{k+p}(x)) \wedge g^{k'+p'}(f^{k+p}(x)) = y\\ &\left.\wedge E_g^{k',\ell'}(f^{k+p}(x)) \wedge \bigwedge_{\hat{\ell}<\ell'} g^{k'}(f^{k+p}(x)) \neq g^{k'+\hat{\ell}}(f^{k+p}(x))\right)\end{aligned}$$

and with $\mathcal{I}(2,4) = \{(1,1,0),(0,1,0),(0,2,0)\}$ we find ϑ_ψ analogously.

This now allows us to prove the logical characterisation of our Colour Refinement Algorithm.

Theorem 4.2. *Let $\mathfrak{A}$ and $\mathfrak{B}$ be two structures of the same signature σ with relation and unary function symbols and let $k \in \mathbb{N}$. Then the two following statements are equivalent:*

1. RCR_k *distinguishes $\mathfrak{A}$ and $\mathfrak{B}$.*
2. *There exists a sentence $\varphi \in \mathsf{GF}(\mathsf{C})_k$ such that $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \not\models \varphi$.*

Proof. We prove that *1.* implies *2.*. Let $\mathfrak{A}$ and $\mathfrak{B}$ be distinguished by RCR_k. If they are of different sizes, assume without loss of generality that

$$|\mathfrak{A}| = n > n' = |\mathfrak{B}|.$$

Then define $\varphi \coloneqq \exists^{\geq n} x.\top \in \mathsf{GF}(\mathsf{C})_k$, which obviously distinguishes the structures.

Now assume $|\mathfrak{A}| = |\mathfrak{B}| = n$. By definition, RCR distinguishes $\widetilde{\mathfrak{A}}$ and $\widetilde{\mathfrak{B}}$. When using the proof from [19], we obtain a formula $\widetilde{\varphi} \in \mathsf{GF}(\mathsf{C})$ of signature $\widetilde{\sigma}$ that distinguishes the expansions. This formula $\widetilde{\varphi}$ can then be translated to a formula $\varphi \in \mathsf{GF}(\mathsf{C})_k$ of signature σ. Replace every atomic subformula $\mathrm{Eq}_{\alpha,\beta}(x,y)$, where $\alpha,\beta \in \mathrm{Alters}_n^k(\sigma)$, by the formula $\alpha(x) = \beta(y)$. Similarly, replace every atomic subformula $R_{\alpha_1,\ldots,\alpha_\ell}(x_1,\ldots,x_\ell)$ by the formula $R(\alpha_1(x_1),\ldots,\alpha_\ell(x_\ell))$. Obviously, if a structure's expansion satisfied $\widetilde{\varphi}$, it also satisfies φ and vice versa. Therefore, we get a formula $\varphi \in \mathsf{GF}(\mathsf{C})_k$ that distinguishes $\mathfrak{A}$ and $\mathfrak{B}$.

Now we prove that *2.* implies *1.*. Let $\varphi \in \mathsf{GF}(\mathsf{C})_k$ such that $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \not\models \varphi$. Using Lemma 4.3 we can obtain a formula ϑ_ψ for every atomic subformula ψ of φ with $\mathfrak{A} \models \psi$ if, and only if, $\mathfrak{A} \models \vartheta_\psi$. With this we can construct an equivalent formula $\varphi' \in \mathsf{GF}(\mathsf{C})_k$, which then allows us, to easily translate it to $\widetilde{\sigma}$. We will construct this formula φ' inductively and directly prove the equivalence.

Claim 4.1. *The two formulae φ and φ' are equivalent.*

Proof. **Base cases:** If φ is an atomic formula, that is, either a term equivalence or a relation, then set φ' to ϑ_φ. The equivalence follows directly from the above Lemmas 4.2 and 4.3.

Inductive cases: In the cases where φ is of the form $\neg\vartheta$ or $\vartheta_1 \wedge \vartheta_2$, we set φ' to $\neg\vartheta'$ or $\vartheta_1' \wedge \vartheta_2'$ and the claim follows directly using the induction hypothesis.

Let φ be of the form $\exists^{\geq \ell}\mathbf{v}.\Delta \wedge \vartheta$. In addition to translating Δ and ϑ to ϑ_Δ and ϑ', respectively, we also will need to transform the formula, so that it still is a valid formula in $\mathsf{GF}(\mathsf{C})_k$. When looking at

the possible translations from the atomic formula $\Delta(x_1, \ldots, x_m)$, we see that it must be of the form $\bigvee_{i\in[o]}(\Delta'_i(x_1, \ldots, x_m) \wedge \bigwedge \Psi_i(x_1, \ldots, x_m))$. When considering the transformed formula

$$\exists^{\geq \ell}\mathbf{v}. \left(\bigvee_{i\in[o]} \left(\Delta'_i \wedge \bigwedge \Psi_i \right) \wedge \vartheta' \right),$$

we then will distribute ϑ' over the disjunction and thus define

$$\psi \coloneqq \exists^{\geq \ell}\mathbf{v}. \left(\bigvee_{i\in[o]} \left(\Delta'_i \wedge \bigwedge \Psi_i \wedge \vartheta' \right) \right)$$

In the following we prove the equivalence of φ and ψ. Let $\mathfrak{A} \models \varphi$. This means there are at least ℓ tuples $\mathbf{a} \in A$, such that $(\mathfrak{A}, \mathbf{a}) \models \Delta(\mathbf{v}) \wedge \vartheta(\mathbf{v})$. Using the induction hypothesis we get that this is equivalent to $(\mathfrak{A}, \mathbf{a}) \models \bigvee(\Delta' \wedge \bigwedge \Psi) \wedge \vartheta'$, which, using the distributive law of propositional logic, is equivalent to $(\mathfrak{A}, \mathbf{a}) \models \bigvee(\Delta' \wedge \bigwedge \Psi \wedge \vartheta')$. Therefore the number of tuples that satisfy $\Delta \wedge \vartheta$ must be the same as for $\bigvee(\Delta' \wedge \bigwedge \Psi \wedge \vartheta')$ and $\mathfrak{A} \models \exists^{\geq \ell}\mathbf{v}. \bigvee(\Delta' \wedge \bigwedge \Psi \wedge \vartheta')$ follows.

However, we are not finished, because $\psi \notin \mathsf{GF}(\mathsf{C})_k$. We will solve this, by considering all possible segmentations of the disjunction. Informally, for $o, \ell \in \mathbb{N}$ we define $\mathrm{Parts}(o, \ell)$ as the set of all multisets with exactly ℓ elements of $[o]$, respecting their multiplicity. We then define φ' as

$$\bigvee_{(M,\mathrm{mult}_M)\in\mathrm{Parts}(o,\ell)} \bigwedge_{i\in M} \exists^{\geq \mathrm{mult}_M(i)}\mathbf{v}.(\Delta'_i \wedge \bigwedge \Psi_i \wedge \vartheta')$$

and will prove the equivalence between ψ and φ' in the following.

Let $\mathfrak{A} \models \psi$. Then there are ℓ different tuples $\mathbf{a}$, such that $\mathfrak{A}, \mathbf{a} \models \bigvee_{i\in[o]}(\Delta'_i \wedge \bigwedge \Psi_i \wedge \vartheta')$. From the above lemmas we know that for every such tuple, there is exactly one i such that $\mathfrak{A}, \mathbf{a} \models \Delta'_i \wedge \bigwedge \Psi_i \wedge \vartheta'$. Now construct a multiset (M, mult_M) with exactly these i that are being satisfied and with the multiplicity of the amount of tuples satisfying them. One can see that $(M, \mathrm{mult}_M) \in \mathrm{Parts}(o, n)$ and that

$$\mathfrak{A} \models \bigwedge_{i\in M} \exists^{\geq \mathrm{mult}_M(i)}\mathbf{v}.(\Delta'_i \wedge \bigwedge \Psi_i \wedge \vartheta').$$

It directly follows that $\mathfrak{A} \models \varphi'$.

Let $\mathfrak{A} \models \varphi'$. From the construction we know, that every $\mathbf{a}$ that is being quantified satisfies only the $\Delta'_i \wedge \bigwedge \Psi_i \wedge \vartheta'$ they are being quantified for. By the definition of $\mathrm{Parts}(o, \ell)$, we thus get exactly ℓ tuples that satisfy some $\Delta'_i \wedge \bigwedge \Psi_i \wedge \vartheta'$ and $\mathfrak{A} \models \psi$ follows. We therefore have proved the claim. □

Note that for every term α that appears in φ', it holds that $\alpha \in \mathrm{Alters}^k_n(\sigma)$. This follows from the properties of the translation in Lemma 4.3. Furthermore, for every atomic subformula, we have a corresponding relation symbol in $\widetilde{\sigma}$. With this, we can transform φ' to a formula $\widetilde{\sigma} \in \mathsf{GF}(\mathsf{C})$ of signature $\widetilde{\sigma}$, such that $\mathfrak{A} \models \varphi'$ if, and only if, $\widetilde{\mathfrak{A}} \models \widetilde{\varphi}$.

It can be seen that the only subformulae that need to be changed are atomic. Let ψ be an atomic formula that appears in φ'. If ψ is a term equation, that is, it is of the form $t(x) = s(y)$, we know through the construction of φ' and the definition of the transitive expansion, that there are $\alpha, \beta \in \mathrm{Alters}^k_n(\sigma)$ with $\alpha = t$ and $\beta = s$. As such, we can replace ψ with $\mathrm{Eq}_{\alpha,\beta}(x, y)$.

If ψ is a relation, that is, it is of the form $R(t_1(x_1), \ldots, t_m(x_m))$, we again have $\alpha_1, \ldots, \alpha_m \in \mathrm{Alters}^k_n(\sigma)$, such that $\alpha_i = t_i$ for $i \in [m]$. We then can replace ψ with $R_{\alpha_1,\ldots,\alpha_m}(x_1, \ldots, x_m)$. From the semantic definition of the transitive expansion, it can be easily seen that $\mathfrak{A} \models \varphi'$ if, and only if, $\widetilde{\mathfrak{A}} \models \widetilde{\varphi}$.

With this, we have obtained a formula $\widetilde{\varphi} \in \mathsf{GF}(\mathsf{C})$ of signature $\widetilde{\sigma}$, where $\widetilde{\mathfrak{A}} \models \widetilde{\varphi}$ and $\widetilde{\mathfrak{B}} \not\models \widetilde{\varphi}$. Using [19], we thus know that RCR distinguishes $\widetilde{\mathfrak{A}}$ and $\widetilde{\mathfrak{B}}$ and by definition we can deduce that RCR_k distinguishes $\mathfrak{A}$ and $\mathfrak{B}$. □

We thus can characterise RCR_k with the logic $\mathsf{GF}(\mathsf{C})_k$.

4.5 Characterisation Through Homomorphism Counting

One very interesting property of classical, as well as Relational Colour Refinement is that aside from its logical characterisation, it can be characterised by counting homomorphisms from certain structures. As we showed, the logical characterisation of Relational Colour Refinement has two possible extensions to structures with functions. Thus we now want to consider, whether those extensions can also be characterised by counting homomorphisms.

In the following we will see that, using the two established approaches, it is in general not possible, to find acyclic structures with functions that divide two structures by homomorphism count. We will concentrate on the encoding defined in Section 4.1, where we will show one positive and one negative result.

To begin, let us define two concepts for relational structures that encode a non-relational structure.

Definition 4.5 (Total structures)**.** Let $\sigma := \sigma_{\text{Rel}} \dot\cup \sigma_{\text{Func}}$ be a signature where σ_{Func} contains exactly all function symbols and is not empty. Now let σ' be the relational encoding of σ and let $\mathfrak{A}'$ be a σ'-structure. We call $\mathfrak{A}'$ total, if for every $R_f \in \widehat{\sigma}$ with arity $n+1$ where $f \in \sigma_{\text{Func}}$ has arity n, and every tuple $\mathbf{x}$ of length n, there is a y such that $(\mathbf{x}y) \in R_f^{\mathfrak{A}'}$.

Total structures therefore capture the notion that every relation that encodes a function is defined for the complete value-domain. However, functions also only valuate to exactly one value. This idea is captured by the following definition.

Definition 4.6 (Functional structures)**.** We define σ, σ' and $\mathfrak{A}'$ exactly as in Definition 4.5. We call $\mathfrak{A}'$ functional, if for every $R_f \in \widehat{\sigma}$ with arity $n+1$ where $f \in \sigma_{\text{Func}}$ has arity n, we have that if $(\mathbf{x}y) \in R_f^{\mathfrak{A}'}$ there is no $y' \neq y$ such that $(\mathbf{x}y') \in R_f^{\mathfrak{A}'}$.

To continue, we have to define what it means to be acyclic for a structure with functions.

Definition 4.7 (Acyclic, non-relational structures)**.** Let σ be a signature with function symbols and σ' its encoding as it is defined in Section 4.1. Let $\mathfrak{C}$ be a σ-structure and $\mathfrak{C}'$ be its encoding of signature σ'. We then call $\mathfrak{C}$ acyclic, if $\mathfrak{C}'$ is acyclic, with respect to acyclicity as it is defined in Definition 3.3.

We now want to find the equivalence between the existence of an acyclic structure with functions and an encoding of an acyclic structure with the above properties (with respect to homomorphism counting).

Lemma 4.4. *Let σ be a signature with function symbols and σ' be the relational encoding of it. Let $\mathfrak{A}$ and $\mathfrak{B}$ be σ-structures and $\mathfrak{A}'$ and $\mathfrak{B}'$ be their respective encodings of signature σ'. Then the two following statements are equivalent.*

1. *There exists an acyclic structure $\mathfrak{C}$ of signature σ such that* $\hom(\mathfrak{C}, \mathfrak{A}) \neq \hom(\mathfrak{C}, \mathfrak{B})$.
2. *There exists an acyclic, total and functional structure $\mathfrak{C}'$ of signature σ' such that* $\hom(\mathfrak{C}', \mathfrak{A}') \neq \hom(\mathfrak{C}', \mathfrak{B}')$.

Proof. We begin by proving that *1.* implies *2.*. Let $\mathfrak{C}$ be such a σ-structure. Now let $\mathfrak{C}'$ be its encoding as a σ'-structure. By definition $\mathfrak{C}'$ is acyclic and when considering the definition of the encoding, we find that it also has to be total and functional. Furthermore we have $\text{Hom}(\mathfrak{C}, \mathfrak{A}) = \text{Hom}(\mathfrak{C}', \mathfrak{A}')$ and respectively with $\mathfrak{B}$ and $\mathfrak{B}'$.

Let $\varphi \in \text{Hom}(\mathfrak{C}, \mathfrak{A})$. We show that $\varphi \in \text{Hom}(\mathfrak{C}', \mathfrak{A}')$. Let $(\mathbf{x}y) \in R_f^{\mathfrak{C}'}$ for a function symbol $f \in \sigma$. Then by definition of the encoding $f^{\mathfrak{C}}(\mathbf{x}) = y$ and then $f^{\mathfrak{A}}(\varphi(\mathbf{x})) = \varphi(f^{\mathfrak{C}}(\mathbf{x})) = \varphi(y)$. Thus $(\varphi(\mathbf{x})\varphi(y)) \in R_f^{\mathfrak{A}'}$, but this is equal to $\varphi(\mathbf{x}y) \in R_f^{\mathfrak{A}'}$. Let $\mathbf{x} \in R^{\mathfrak{C}'}$ where $R \neq R_f$ for all function

symbols $f \in \sigma$. Then $R^{\mathfrak{C}'} = R^{\mathfrak{C}}$ and $R^{\mathfrak{A}} = R^{\mathfrak{A}'}$. Therefore because $\mathbf{x} \in R^{\mathfrak{C}}$, we have $\varphi(\mathbf{x}) \in R^{\mathfrak{A}} = R^{\mathfrak{A}'}$. This was to be shown

Let $\varphi \in \mathrm{Hom}(\mathfrak{C}', \mathfrak{A}')$. We show that $\varphi \in \mathrm{Hom}(\mathfrak{C}, \mathfrak{A})$. Let $\mathbf{x} \in R^{\mathfrak{C}}$ for a relational-symbol R. By the construction of the encoding we know that $R^{\mathfrak{C}} = R^{\mathfrak{C}'}$ and $R^{\mathfrak{A}'} = R^{\mathfrak{A}}$ and with the same argument as before we can conclude that $\varphi(\mathbf{x}) \in R^{\mathfrak{A}}$. Let $\mathbf{x}$ and y be such that $f^{\mathfrak{C}}(\mathbf{x}) = y$. Then by construction we know that $(\mathbf{x}y) \in R_f^{\mathfrak{C}'}$. Because φ is a homomorphism we also know that $\varphi(\mathbf{x}y) \in R_f^{\mathfrak{A}'}$ and because $\mathfrak{A}'$ is also an encoding, we have that $f^{\mathfrak{A}}(\varphi(\mathbf{x})) = \varphi(y) = \varphi(f^{\mathfrak{C}}(\mathbf{x}))$. This was to be shown. $\mathfrak{A}$ and $\mathfrak{A}'$ can be replaced by $\mathfrak{B}$ and $\mathfrak{B}'$, respectively, to get the analogous result for the other structure.

We now show that *2.* implies *1.*. Let $\mathfrak{C}'$ be an acyclic, total and functional σ'-structure. We can now construct a σ-structure $\mathfrak{C}$ by decoding $\mathfrak{C}'$ and will also get that $\mathrm{Hom}(\mathfrak{C}', \mathfrak{A}') = \mathrm{Hom}(\mathfrak{C}, \mathfrak{A})$. For a relation symbol $R \in \sigma$ we can define $R^{\mathfrak{C}} \coloneqq R^{\mathfrak{C}'}$. For a function symbol $f \in \sigma$ we can define $f^{\mathfrak{C}}$ as follows: Let f be of arity n. Then for a tuple $\mathbf{x}$ of length n, there must be a y such that $(\mathbf{x}y) \in R_f^{\mathfrak{C}'}$ because $\mathfrak{C}'$ is total and there must be exactly one such y because $\mathfrak{C}'$ is functional. Therefore we define $f^{\mathfrak{C}}(\mathbf{x}) = y$. The claim that the sets of homomorphisms is equal can be verified using exactly the same arguments as in the analogous proof of the other direction. □

We can now continue to put this in relation to the statement regarding RCR. Naive RCR distinguishes two σ-structures $\mathfrak{A}$ and $\mathfrak{B}$ if, and only if, RCR distinguishes the encodings $\mathfrak{A}'$ and $\mathfrak{B}'$ of signature σ'. Due to the results of [19] this is the case if, and only if, there is an acyclic σ'-structure $\mathfrak{C}'$ such that $\hom(\mathfrak{C}', \mathfrak{A}') \neq \hom(\mathfrak{C}', \mathfrak{B}')$. We would like to achieve the result that this is equivalent to there being an acyclic σ structure that distinguishes $\mathfrak{A}$ and $\mathfrak{B}$ by homomorphism count (however this will not be the case). By the above lemma, the latter is equivalent to there being an acyclic, total and functional σ'-structure $\mathfrak{C}''$ that distinguishes $\mathfrak{A}'$ and $\mathfrak{B}'$ by homomorphism count. Our goal will therefore be, to study the relationship between the two following statements.

1. There is an acyclic σ'-structure $\mathfrak{C}'$ such that $\hom(\mathfrak{C}', \mathfrak{A}') \neq \hom(\mathfrak{C}', \mathfrak{B}')$.
2. There is an acyclic, total and functional σ'-structure $\mathfrak{C}''$ such that $\hom(\mathfrak{C}'', \mathfrak{A}') \neq \hom(\mathfrak{C}'', \mathfrak{B}')$.

It is obvious that *2.* implies *1.*, however the other direction will not hold in general. In fact, we will be able to construct a functional structure from $\mathfrak{C}'$, but totality will not be able to be constructed. We will in the following prove the former claim and then show the latter claim, using a family of counterexamples.

Lemma 4.5. *Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures of signature σ and $\mathfrak{A}'$, $\mathfrak{B}'$ and σ' the respective encodings. If there is an acyclic structure $\mathfrak{C}'$ of signature σ' with $\hom(\mathfrak{C}', \mathfrak{A}') \neq \hom(\mathfrak{C}', \mathfrak{B}')$, then we can construct a functional, acyclic structure $\mathfrak{C}''$ of signature σ' such that $\hom(\mathfrak{C}'', \mathfrak{A}') = \hom(\mathfrak{C}', \mathfrak{A}')$ and $\hom(\mathfrak{C}'', \mathfrak{B}') = \hom(\mathfrak{C}', \mathfrak{B}')$.*

Proof. The proof for the above lemma will work as follows. If $\mathfrak{C}'$ is not functional, then there is a function symbol $f \in \sigma$ and two different tuples $(\mathbf{x}y), (\mathbf{x}z) \in R_f^{\mathfrak{C}'}$, we call this a collision. We will give a procedure to iteratively remove such collisions. The procedure will reduce the number of elements of $\mathfrak{C}'$ by one, will keep the acyclicity property and will result in a structure with the same amount of homomorphisms to $\mathfrak{A}'$ and $\mathfrak{B}'$, respectively. Thus, by continuously applying that procedure to all collisions, we will get $\mathfrak{C}''$. The algorithm must terminate, as the number of elements strictly decreases and we only consider finite structures.

The remainder of this proof will be dedicated to describing the procedure and proving the above claims. Assume that there is a function symbol $f \in \sigma$ and two different tuples $(\mathbf{x}y), (\mathbf{x}z) \in R_f^{\mathfrak{C}'}$. Our goal will be to construct a structure $\mathfrak{C}''$ of signature σ' without this particular collision and with the following properties:

a. $|\mathfrak{C}''| = |\mathfrak{C}'| - 1$

b. $\mathfrak{C}''$ is acyclic

c. $\hom(\mathfrak{C}'', \mathfrak{A}') = \hom(\mathfrak{C}', \mathfrak{A}')$ and $\hom(\mathfrak{C}'', \mathfrak{B}') = \hom(\mathfrak{C}', \mathfrak{B}')$.

We define $\mathfrak{C}'' \coloneqq ((C' \setminus \{y, z\}) \cup \{v_{y,z}\}, \sigma)$ and will define the relations using the following function. The function $\chi : \mathbf{C}' \to \mathbf{C}''$ which maps tuples of $\mathfrak{C}'$ to tuples of $\mathfrak{C}''$ with the same length. Concretely, for an arbitrary tuple $\mathbf{c} \in \mathbf{C}'$ of length k and for every $i \in [k]$, we have

$$\chi(\mathbf{c})_i \coloneqq \begin{cases} \mathbf{c}_i & \text{if } \mathbf{c}_i \notin \{y, z\} \\ v_{y,z} & \text{if } \mathbf{c}_i \in \{y, z\}. \end{cases}$$

Then, for all $R \in \sigma$, we define $R^{\mathfrak{C}''} \coloneqq \{\chi(\mathbf{c}) : \mathbf{c} \in R^{\mathfrak{C}'}\}$. We will now proceed by proving the above properties.

Property a.: This property follows directly from the definition of C''. We have $y \neq z$, $y, z \in C'$ and thus $|C' \setminus \{y, z\}| = |C'| - 2$. Furthermore, we have $v_{y,z} \notin C'$ and therefore $|(C' \setminus \{y, z\}) \cup \{v_{y,z}\}| = |C'| - 1$. This was to be shown.

Property b.: To show that $\mathfrak{C}''$ is acyclic, we first will define an undirected graph J'', will prove that it is connected and cycle free, thus a tree, and that it fulfils the join tree property for $\mathfrak{C}''$. By the assumption we know that $\mathfrak{C}'$ is acyclic and thus has a join tree J'. We further notice that $\mathbf{C}'' = \{\chi(\mathbf{c}) : \mathbf{c} \in \mathbf{C}'\}$. Now we define $V(J'') \coloneqq \mathbf{C}''$ and $E(J'') \coloneqq \{\{\chi(\mathbf{u}), \chi(\mathbf{v})\} : \{\mathbf{u}, \mathbf{v}\} \in E(J')\}$.

Claim 4.2. *J'' is connected.*

Proof. Consider $\mathbf{u}, \mathbf{v} \in V(J'')$. Then there are $\mathbf{a}, \mathbf{b} \in V(J')$, such that $\chi(\mathbf{a}) = \mathbf{u}$ and $\chi(\mathbf{b}) = \mathbf{v}$. By assumption, J' is a tree, so there are $\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_k$ with $\mathbf{a}_0 = \mathbf{a}$, $\mathbf{a}_k = \mathbf{b}$ and $\{\mathbf{a}_{i-1}, \mathbf{a}_i\} \in E(J')$ for all $i \in [k]$. By definition, we have $\chi(\mathbf{a}_0), \chi(\mathbf{a}_1), \ldots, \chi(\mathbf{a}_k) \in V(J'')$ with $\chi(\mathbf{a}_0) = \chi(\mathbf{a}) = \mathbf{u}$, $\chi(\mathbf{a}_k) = \chi(\mathbf{b}) = \mathbf{v}$ and $\{\chi(\mathbf{a}_{i-1}), \chi(\mathbf{a}_i)\} \in E(J'')$ for all $i \in [k]$. Thus $\mathbf{u}$ and $\mathbf{v}$ are connected. □

In the following, for an arbitrary $e \in C'$, we define the set $V_e \coloneqq \{\mathbf{c} \in V(J') : e \in \mathbf{c}\}$. One can see that $V(J'_e) = V_e$, where J'_e is the subgraph of J', induced by all elements containing e.

Claim 4.3. *J'' is cycle-free.*

Proof. Assume that there were $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_k \in V(J'')$ with $\{\mathbf{u}_{i-1}, \mathbf{u}_i\} \in E(J'')$ for all $i \in [k]$ and $\{\mathbf{u}_k, \mathbf{u}_0\} \in E(J'')$. We now define directed edges such that $e_0 = (\mathbf{u}_0, \mathbf{u}_1)$, $e_1 = (\mathbf{u}_1, \mathbf{u}_2)$, $\ldots$, $e_k = (\mathbf{u}_k, \mathbf{u}_0)$, therefore we have $e_i = (\mathbf{u}_i, \mathbf{u}_{i+1 \bmod k})$. For every edge e_i choose two elements $\mathbf{a}_i, \mathbf{b}_i \in V(J')$ such that $\{\mathbf{a}_i, \mathbf{b}_i\} \in E(J')$, $\chi(\mathbf{a}_i) = \mathbf{u}_i$ and $\chi(\mathbf{b}_i) = \mathbf{u}_{i+1 \bmod k}$. These elements must exist by the definition of $E(J'')$.

We now prove that there exists a cycle in J', which would contradict our assumption of J' being a join tree for $\mathfrak{C}'$. To show this, we prove that for all $i \in \{0\} \cup [k]$, the elements $\mathbf{b}_i$ and $\mathbf{a}_{i+1 \bmod k}$ are connected in J'. We see that $\chi(\mathbf{b}_i) = \mathbf{u}_{i+1 \bmod k} = \chi(\mathbf{a}_{i+1 \bmod k})$.

If there is a $c \in \operatorname{set}(\mathbf{u}_{i+1 \bmod k})$ with $c \neq v_{y,z}$, then by definition of χ, we have $c \in \operatorname{set}(\mathbf{b}_i) \cap \operatorname{set}(\mathbf{a}_{i+1 \bmod k})$. Therefore $\mathbf{b}_i, \mathbf{a}_{i+1 \bmod k} \in V_c$ and because J' is a join tree, $\mathbf{b}_i$ and $\mathbf{a}_{i+1 \bmod k}$ have to be connected.

If $\operatorname{set}(\mathbf{u}_{i+1 \bmod k}) = \{v_{y,z}\}$, we have four possible cases. If $y \in \operatorname{set}(\mathbf{b}_i) \cap \operatorname{set}(\mathbf{a}_{i+1 \bmod k})$ or $z \in \operatorname{set}(\mathbf{b}_i) \cap \operatorname{set}(\mathbf{a}_{i+1 \bmod k})$, then we can do the same as before by setting $c = y$ or $c = z$, respectively. Otherwise we have $y \in \operatorname{set}(\mathbf{b}_i)$ and $z \in \operatorname{set}(\mathbf{a}_{i+1 \bmod k})$, or $z \in \operatorname{set}(\mathbf{b}_i)$ and $y \in \operatorname{set}(\mathbf{a}_{i+1 \bmod k})$. We will only consider the former option, as the latter can be proven analogously. From our beginning assumption we know that $(\mathbf{x}y), (\mathbf{x}z) \in R_f$. Choose some $x \in \mathbf{x}$, and $(\mathbf{x}y), (\mathbf{x}z) \in V_x$ follows. Furthermore, we have $\mathbf{b}_i, (\mathbf{x}y) \in V_y$ and $\mathbf{a}_{i+1 \bmod k}, (\mathbf{x}z) \in V_z$. Since J' is a join tree, we thus know

that $\mathbf{b}_i$ is connected with $(\mathbf{x}y)$, which in turn is connected with $(\mathbf{x}z)$, which again is connected with $\mathbf{a}_{i+1 \bmod k}$. Therefore $\mathbf{b}_i$ and $\mathbf{a}_{i+1 \bmod k}$ are connected.

Thus we have found a cycle in J', which is a contradiction to it being a join tree. Therefore our assumption of the existence of the elements $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_k$ has to be false. □

The last missing piece to prove the acyclicity of $\mathfrak{C}''$ is to show that J'' fulfils the join tree property. That is, for any $c \in C''$, the set $\{\mathbf{c} \in V(J'') : c \in \mathrm{set}(\mathbf{c})\}$ induces a connected subgraph of J''.

Claim 4.4. *J'' is a valid join tree.*

Proof. Consider any $c \in C''$ and two elements $\mathbf{u}$ and $\mathbf{v}$ from the set $S \coloneqq \{\mathbf{c} \in V(J'') : c \in \mathrm{set}(\mathbf{c})\}$. We show that there is a path from $\mathbf{u}$ to $\mathbf{v}$ in S. If $c \neq v_{y,z}$, then $c \in C'$, thus consider V_c. By definition there are $\mathbf{a}, \mathbf{b} \in V_c$ such that $\chi(\mathbf{a}) = \mathbf{u}$ and $\chi(\mathbf{b}) = \mathbf{v}$. Because J' is a join tree, V_c must induce a connected subtree. Thus there must be $\mathbf{a}_0, \ldots, \mathbf{a}_k \in V_c$ such that $\mathbf{a}_0 = \mathbf{a}$, $\mathbf{a}_k = \mathbf{b}$ and $\{\mathbf{a}_{i-1}, \mathbf{a}_i\} \in E(J')$ for all $i \in [k]$. By definition we then get, that there must be $\chi(\mathbf{a}_0), \chi(\mathbf{a}_k) \in S$ and we get $\chi(\mathbf{a}_0) = \chi(\mathbf{a}) = \mathbf{u}$, $\chi(\mathbf{a}_k) = \chi(\mathbf{b}) = \mathbf{v}$ and $\{\chi(\mathbf{a}_{i-1}, \mathbf{a}_i)\} \in E(J''))$ for all $i \in [k]$. Therefore, $\mathbf{u}$ and $\mathbf{v}$ are connected in S.

If $v = v_{y,z}$, we again define $\mathbf{a}, \mathbf{b} \in V(J')$ such that $\chi(\mathbf{a}) = \mathbf{u}$ and $\chi(\mathbf{b}) = \mathbf{v}$. If $\mathbf{a}, \mathbf{b} \in V_y$ or $\mathbf{a}, \mathbf{b} \in V_z$, then we can proceed exactly as in the former case with $v = y$ or $v = z$, respectively. If that is not the case, then either $\mathbf{a} \in V_y$ and $\mathbf{b} \in V_z$, or the other way round. We will only prove the former case, as the latter case can proven analogously. Since the following will depend on it, we will now prove the following equality: $S = \{\chi(\mathbf{c}) : \mathbf{c} \in V_y\} \cup \{\chi(\mathbf{c}) : \mathbf{c} \in V_z\}$.

$\supseteq$: Let $\mathbf{u} \in \{\chi(\mathbf{c}) : \mathbf{c} \in V_y\} \cup \{\chi(\mathbf{c}) : \mathbf{c} \in V_z\}$. We then get that $\mathbf{u} = \chi(\mathbf{a})$ for an $\mathbf{a} \in V(J')$. From the definition it follows that $y \in \mathrm{set}(\mathbf{a})$ or $z \in \mathrm{set}(\mathbf{a})$ has to hold. Thus we get that $v_{y,z} \in \mathrm{set}(\chi(\mathbf{a})) = \mathrm{set}(\mathbf{u})$ and therefore $\mathbf{u} \in S$ has to hold.

$\subseteq$: Let $\mathbf{u} \in S$, then $v_{y,z} \in \mathrm{set}(\mathbf{u})$ follows. By definition there must exist an $\mathbf{a} \in V(J')$ such that $\chi(\mathbf{a}) = \mathbf{u}$ and $y \in \mathrm{set}(\mathbf{a})$ or $z \in \mathrm{set}(\mathbf{a})$ has to hold. Thus we get $\chi(\mathbf{a}) = \mathbf{u} \in \{\chi(\mathbf{c}) : \mathbf{c} \in V_y\} \cup \{\chi(\mathbf{c}) : \mathbf{c} \in V_z\}$.

It is obvious that $(\mathbf{x}y) \in V_y$ and $(\mathbf{x}z) \in V_z$. We now get a path in V_y with the elements $\mathbf{a}_0, \ldots, \mathbf{a}_k \in V_y$ such that $\mathbf{a}_0 = \mathbf{a}$, $\mathbf{a}_k = (\mathbf{x}y)$ and $\{\mathbf{a}_{i-1}, \mathbf{a}_i\} \in E(J')$ for all $i \in [k]$. We also get a path in V_z with the elements $\mathbf{b}_0, \ldots, \mathbf{b}_\ell \in V_z$ such that $\mathbf{b}_0 = (\mathbf{x}z)$, $\mathbf{b}_\ell = \mathbf{b}$ and $\{\mathbf{b}_{i-1}, \mathbf{b}_i\} \in E(J')$. Therefore with the above equation, we have two paths in S: $\chi(\mathbf{a}_0), \ldots, \chi(\mathbf{a}_k)$ and $\chi(\mathbf{b}_0), \ldots, \chi(\mathbf{b}_\ell)$, where $\chi(\mathbf{a}_0) = \mathbf{u}$ and $\chi(\mathbf{b}_\ell) = \mathbf{v}$. However, $\chi(\mathbf{a}_k) = \chi((\mathbf{x}y)) = (\mathbf{x}' v_{y,z}) = \chi((\mathbf{x}z)) = \chi(\mathbf{b}_0)$. Therefore, we get one path from $\mathbf{u}$ to $\mathbf{v}$ in S. □

Property c.: To show that $\hom(\mathfrak{C}'', \mathfrak{A}') = \hom(\mathfrak{C}', \mathfrak{A}')$ and $\hom(\mathfrak{C}'', \mathfrak{B}') = \hom(\mathfrak{C}', \mathfrak{B}')$, we will give a mapping $\pi : \mathrm{Hom}(\mathfrak{C}', \mathfrak{A}') \to \mathrm{Hom}(\mathfrak{C}'', \mathfrak{A}')$ from homomorphisms from $\mathfrak{C}'$ to $\mathfrak{A}'$ to homomorphisms from $\mathfrak{C}''$ to $\mathfrak{A}'$. We will then show that π is a bijection, from which follows that both sets have the same cardinality, which then proves the claim for $\mathfrak{A}'$. For $\mathfrak{B}'$, the proof is completely analogous, which is why it will be omitted. Let $\varphi \in \mathrm{Hom}(\mathfrak{C}', \mathfrak{A}')$. We now define $\varphi' \coloneqq \pi(\varphi)$ as

$$\varphi'(x) = \begin{cases} \varphi(x) & \text{if } x \neq v_{y,z} \\ \varphi(y) & \text{if } x = v_{y,z}. \end{cases}$$

In the following we will be using that for any $\varphi \in \mathrm{Hom}(\mathfrak{C}', \mathfrak{A}')$, we have that $\varphi(y) = \varphi(z)$. Otherwise, from $(\mathbf{x}y), (\mathbf{x}z) \in R_f^{\mathfrak{C}'}$, it follows that $(\varphi(\mathbf{x})\varphi(y)), (\varphi(\mathbf{x})\varphi(z)) \in R_f^{\mathfrak{A}'}$ for two different tuples $(\varphi(\mathbf{x})\varphi(y))$ and $(\varphi(\mathbf{x})\varphi(z))$. However, this would contradict that, by definition of the encoding, $\mathfrak{A}'$ is functional.

Claim 4.5. *For all $\varphi \in \mathrm{Hom}(\mathfrak{C}', \mathfrak{A}')$, we have $\pi(\varphi) \in \mathrm{Hom}(\mathfrak{C}'', \mathfrak{A}')$.*

Proof. Let $\varphi \in \mathrm{Hom}(\mathfrak{C}', \mathfrak{A}')$ and $\varphi' \coloneqq \pi(\varphi)$. Consider a relation symbol R and a tuple $\mathbf{c} \in R^{\mathfrak{C}''}$. If $v_{y,z} \notin \mathrm{set}(\mathbf{c})$, then $\chi(\mathbf{c}) = \mathbf{c}$, $\varphi(\mathbf{c}) = \varphi'(\mathbf{c})$ and $\mathbf{c} \in R^{\mathfrak{C}'}$ follows. Then we get $\varphi(\mathbf{c}) \in R^{\mathfrak{A}'}$ and further $\varphi'(\mathbf{c}) \in R^{\mathfrak{A}'}$.

If $v_{y,z} \in \mathrm{set}(\mathbf{c})$, then there exists a $\mathbf{c}' \in R^{\mathfrak{C}'}$ such that $\chi(\mathbf{c}') = \mathbf{c}$. We thus get $\varphi(\mathbf{c}') \in R^{\mathfrak{A}'}$. We also have that $\varphi(\mathbf{c}') = \varphi'(\mathbf{c})$, because for all $x \in \mathrm{set}(\mathbf{c}) \setminus \{v_{y,z}\}$, we also have $x \in \mathrm{set}(\mathbf{c}')$ and $\varphi(x) = \varphi'(x)$. For $v_{y,z}$ we have that $\varphi'(v_{y,z}) = \varphi(y) = \varphi(z)$. Therefore, $\varphi'(\mathbf{c}) \in R^{\mathfrak{A}'}$ follows, which was to be shown. □

This shows that π is correctly defined as a mapping between homomorphisms. The two following proofs will show that π is bijective.

Claim 4.6. *π is injective.*

Proof. Let $\varphi_1, \varphi_2 \in \mathrm{Hom}(\mathfrak{C}', \mathfrak{A}')$, $\varphi_1 \neq \varphi_2$ and define $\varphi_1' \coloneqq \pi(\varphi_1)$ and $\varphi_2' \coloneqq \pi(\varphi_2)$. Our goal is to show that $\varphi_1' \neq \varphi_2'$. There has to be a $u \in C'$ such that $\varphi_1(u) \neq \varphi_2(u)$, otherwise they would be the same function. We now do a case distinction.

Case 1: $u \notin \{y, z\}$. Then we have

$$\varphi_1'(u) = \varphi_1(u) \neq \varphi_2(u) = \varphi_2'(u).$$

Case 2: $u \in \{y, z\}$. Then we have

$$\varphi_1'(v_{y,z}) = \varphi_1(y) = \varphi_1(z) = \varphi_1(u) \neq \varphi_2(u) = \varphi_2(z) = \varphi_2(y) = \varphi_2'(v_{y,z}).$$

In both cases we have found an element that gets mapped differently, thus $\varphi_1' \neq \varphi_2'$ must follow. □

Claim 4.7. *π is surjective.*

Proof. Let $\varphi' \in \mathrm{Hom}(\mathfrak{C}'', \mathfrak{A}')$. We now construct a $\varphi \in \mathrm{Hom}(\mathfrak{C}', \mathfrak{A}')$ such that $\pi(\varphi) = \varphi'$. We define

$$\varphi(x) = \begin{cases} \varphi'(x) & \text{if } x \notin \{y, z\} \\ \varphi'(v_{y,z}) & \text{if } x \in \{y, z\}. \end{cases}$$

Using that $\psi(y) = \psi(z)$ for all $\psi \in \mathrm{Hom}(\mathfrak{C}', \mathfrak{A}')$, it can easily be verified that $\pi(\varphi) = \varphi'$. We now only have to show that $\varphi \in \mathrm{Hom}(\mathfrak{C}', \mathfrak{A}')$. Let $\mathbf{c} \in R^{\mathfrak{C}'}$ for a relation symbol R. We now define $\mathbf{c}' = \chi(\mathbf{c})$ and get by construction that $\mathbf{c}' \in R^{\mathfrak{C}''}$. Because we know that φ' is a homomorphism, it follows that $\varphi'(\mathbf{c}') \in R^{\mathfrak{A}'}$. But since $\varphi'(a) = \varphi(a)$ for all $a \notin \{y, z, v_{y,z}\}$ and $\varphi'(v_{y,z}) = \varphi(y) = \varphi(z)$, it follows that $\varphi'(\mathbf{c}') = \varphi(\mathbf{c})$. Therefore $\varphi(\mathbf{c}) \in R^{\mathfrak{A}'}$. □

We now have proved all properties. Therefore, we showed that it is possible to eliminate collisions, while keeping the number of homomorphisms to $\mathfrak{A}'$ and $\mathfrak{B}'$ the same and reducing the number of elements by one and still have an acyclic structure. We thus can apply this procedure to any collision, until there are no collisions left, but then the structure is functional. □

From the above lemma we get that given an acyclic, distinguishing structure, we can construct a new structure, which is also acyclic and distinguishing but in addition also functional. One example how this can be applied can be seen in Figure 7.

Now that we have proved that it is generally possible to enforce a distinguishing, acyclic structure to be functional, we now want to proceed to the proof that totality cannot be enforced. This proof will work as follows. We will give two families of structures over the signature $\sigma = \{E/2, f/1\}$. Any two elements from these families with the same size can be distinguished by naive RCR, thus there exists an acyclic relational structure that distinguishes their encodings by homomorphism count. Therefore, the first statement from above is true. However, we will then prove that there cannot be an acyclic and total structure that also distinguishes them by homomorphism count.

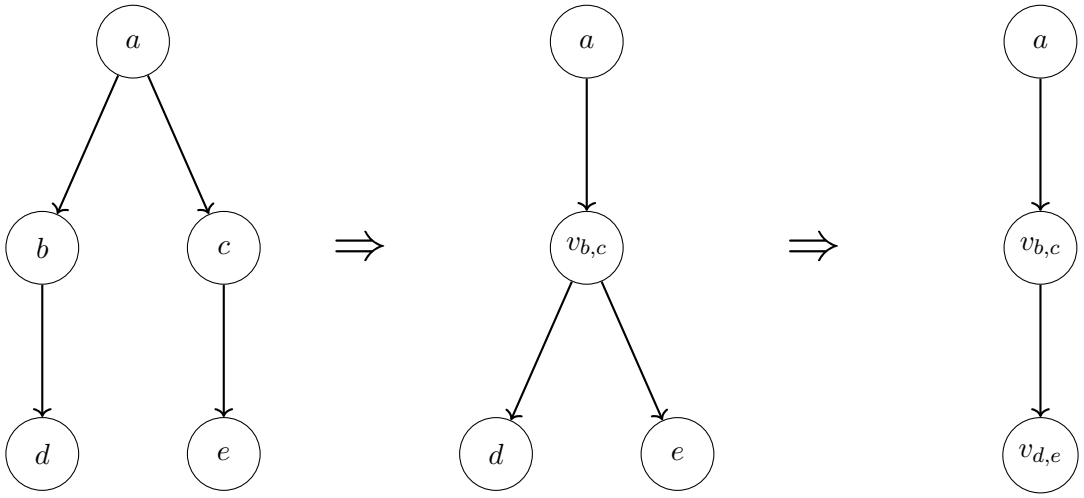


Figure 7: Two successive applications of the procedure described in the proof of Lemma 4.5. The arrows represent a binary relation R_f, which encodes a unary function f.

Let $(\mathfrak{A}_i)_{i\in\mathbb{N}_{\geq 4}}$ be a family of σ-structures, defined as $\mathfrak{A}_n = (A_n, \mathrm{E}^{\mathfrak{A}_n}, f^{\mathfrak{A}_n})$, where $A_n = [n]$, $\mathrm{E}^{\mathfrak{A}_n} = \{(i, i+1), (i+1, i) : i \in [n-2]\}$ and $f^{\mathfrak{A}_n} = \{i \mapsto i+1 \bmod n-1 : i \in [n-1]\} \cup \{n \mapsto n-1\}$. Let $(\mathfrak{B}_i)_{i\in\mathbb{N}_{\geq 4}}$ also be a family of σ-structures, defined as $\mathfrak{B}_n = (B_n, \mathrm{E}^{\mathfrak{B}_n}, f^{\mathfrak{B}_n})$, where $B_n = [n]$, $\mathrm{E}^{\mathfrak{B}_n} = \{(i, i+1), (i+1, i) : i \in [n-1]\}$ and $f^{\mathfrak{B}_n} = \{i \mapsto i+1 \bmod n : i \in [n]\}$. Graphical representations for a few examples of these structures can be seen in Figure 8.

For every $n \in \mathbb{N}_{\geq 4}$ it is clear that naive RCR distinguishes $\mathfrak{A}_n$ and $\mathfrak{B}_n$. The formula

$$\varphi_n = \exists^{\geq 2\cdot(n-1)}(x, y).\, \mathrm{E}(x, y) \in \mathsf{nfGF}(\mathsf{C})$$

distinguishes $\mathfrak{A}_n$ and $\mathfrak{B}_n$ and using Theorem 4.1, this is equivalent to naive RCR distinguishing $\mathfrak{A}_n$ and $\mathfrak{B}_n$. It now only remains to show that there cannot be a total and acyclic structure that distinguishes the encodings $\mathfrak{A}'_n$ and $\mathfrak{B}'_n$ by homomorphism count.

Lemma 4.6. *Let $n \in \mathbb{N}_{\geq 4}$ and let $\mathfrak{C}'_n$ be an acyclic σ'-structure that distinguishes $\mathfrak{A}'_n$ and $\mathfrak{B}'_n$ by homomorphism counts. Then $\mathfrak{C}'_n$ cannot be both acyclic and total.*

Proof. Using basic boolean equivalences, we prove the equivalent statement that if $\mathfrak{C}'_n$ is total, than it either is not distinguishing or not acyclic. As such, let $\mathfrak{C}'_n$ be a total σ'-structure.

We observe that if $\mathfrak{C}'_n$ contains R_f-loops, that is an element $x \in C'_n$ such that $(x, x) \in R_f^{\mathfrak{C}'_n}$, then $\mathfrak{C}'_n$ is not distinguishing. In fact, then we have $\hom(\mathfrak{C}'_n, \mathfrak{A}'_n) = \hom(\mathfrak{C}'_n, \mathfrak{B}'_n) = 0$. Otherwise, there would exist a homomorphism φ to either $\mathfrak{A}'_n$ or $\mathfrak{B}'_n$. Let $c \in C'_n$ be such that $(c, c) \in R_f^{\mathfrak{C}'_n}$. Then we would have $(\varphi(c), \varphi(c)) \in R_f^{\mathfrak{A}'_n}$ or $(\varphi(c), \varphi(c)) \in R_f^{\mathfrak{B}'_n}$, respectively. But in either structure, there does not exist such a loop.

Using similar arguments, we can reason that if $\mathfrak{C}'_n$ contains R_f-2-cycles, that is two elements c and d such that $(c, d), (d, c) \in R_f^{\mathfrak{C}'_n}$, then $\mathfrak{C}'_n$ cannot be distinguishing, as it does not have any homomorphisms to neither $\mathfrak{A}'_n$ nor $\mathfrak{B}'_n$. Otherwise there again would be a homomorphism φ to either $\mathfrak{A}'_n$ or $\mathfrak{B}'_n$. Let $c, d \in C'_n$ form such a R_f-2-cycle. Then we would have $(\varphi(c), \varphi(d)), (\varphi(d), \varphi(c)) \in R_f^{\mathfrak{A}'_n}$ or $(\varphi(c), \varphi(d)), (\varphi(d), \varphi(c)) \in R_f^{\mathfrak{B}'_n}$, respectively. But in either structure there does not exist such a 2-cycle.

From the two observations above it follows that if $\mathfrak{C}'_n$ has cardinality 1 or 2, then it cannot be distinguishing, as it is total and therefore would need to contain either R_f-loops or R_f-2-cycles. When now considering that $\mathfrak{C}'_n$ is total, thus for every c there is a d such that $(c, d) \in R_f^{\mathfrak{C}'_n}$, it is obvious that

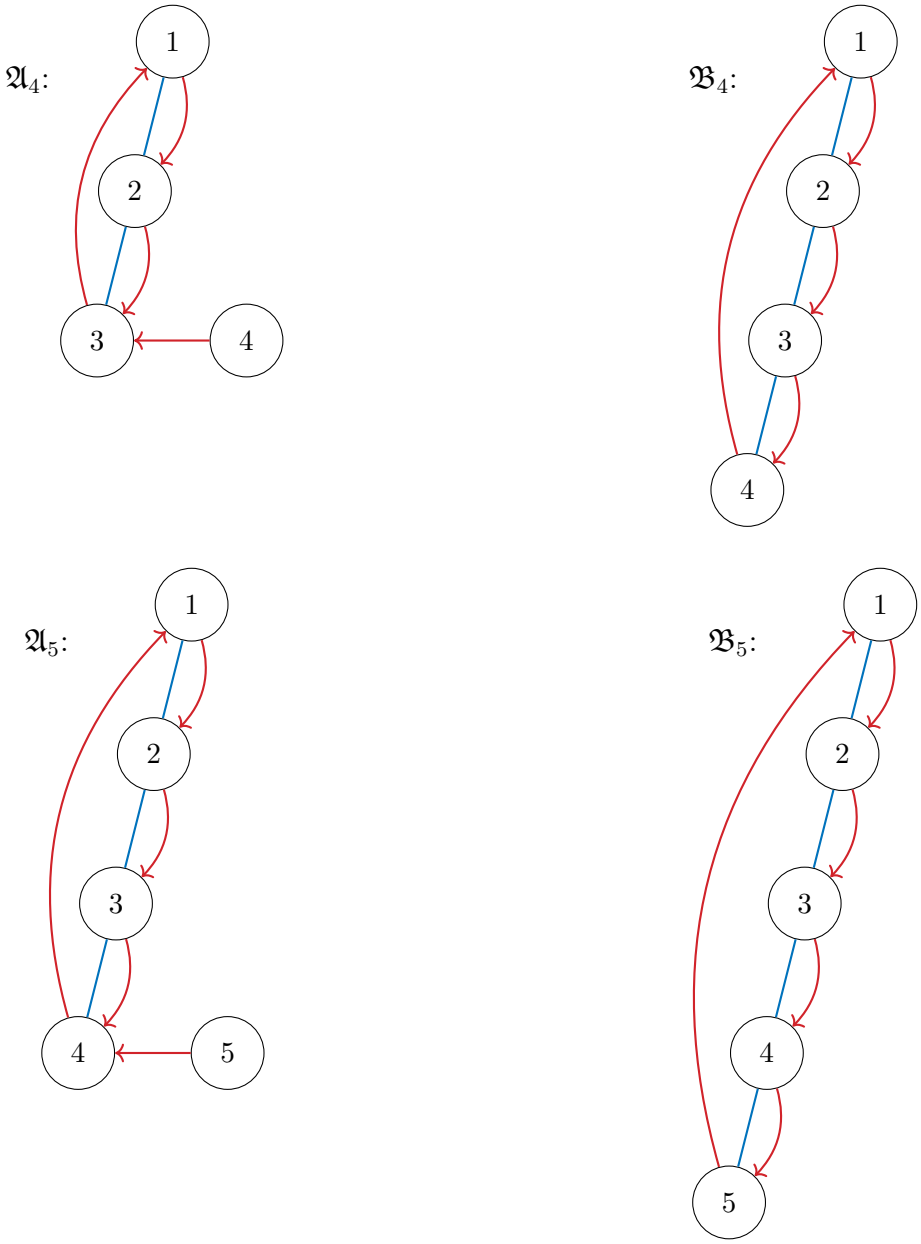


Figure 8: Examples for the structures $\mathfrak{A}_4$, $\mathfrak{A}_5$, $\mathfrak{B}_4$ and $\mathfrak{B}_5$. The blue edges represent the binary relation E, while the red arrows constitute the unary function f.

$\mathfrak{C}'_n$ must contain a R_f cycle of size greater than 2. Thus the Gaifman-Graph of $\mathfrak{C}'_n$ is not acyclic. As it can be found in [19] and [6], this is equivalent to $\mathfrak{C}'_n$ itself not being acyclic. So if $\mathfrak{C}'_n$ is total and does not contain R_f-loops and R_f-2-cycles, then it cannot be acyclic. □

We thus have showed that there cannot exist an acylic and total $\mathfrak{C}'_n$ that distinguishes $\mathfrak{A}'_n$ and $\mathfrak{B}'_n$ by homomorphism count. However, in Lemma 4.4 we showed that the existence of an acyclic, total and functional $\mathfrak{C}'_n$ that distinguishes $\mathfrak{A}'_n$ and $\mathfrak{B}'_n$ is equivalent to the existence of an acyclic $\mathfrak{C}_n$ that distinguishes $\mathfrak{A}_n$ and $\mathfrak{B}_n$. Since there is no acyclic and total $\mathfrak{C}'_n$, there also is no acyclic, total and functional $\mathfrak{C}''_n$ that distinguishes, and therefore there is no $\mathfrak{C}_n$ that distinguishes $\mathfrak{A}_n$ and $\mathfrak{B}_n$ by homomorphism count. Let us formalize our findings.

Theorem 4.3. *Let σ be a signature that contains function symbols, $\mathfrak{A}$ and $\mathfrak{B}$ two σ-structures and let σ', $\mathfrak{A}'$ and $\mathfrak{B}'$ their respective encodings. Now consider the following statements:*

1. *Naive RCR distinguishes $\mathfrak{A}$ and $\mathfrak{B}$.*
2. *There exists an acyclic and functional σ'-structure $\mathfrak{C}'$ such that $\hom(\mathfrak{C}', \mathfrak{A}') \neq \hom(\mathfrak{C}', \mathfrak{B}')$.*
3. *There exists an acyclic σ-structure $\mathfrak{C}$ such that $\hom(\mathfrak{C}, \mathfrak{A}) \neq \hom(\mathfrak{C}, \mathfrak{B})$.*

Then statement 1. is equivalent to statement 2., but while statement 3. implies statement 1., the converse does not hold.

Proof. ***1.* is equivalent to *2.*** Naive RCR distinguishes $\mathfrak{A}$ and $\mathfrak{B}$ if, and only if, there is an acyclic σ'-structure $\mathfrak{C}'$ such that $\hom(\mathfrak{C}', \mathfrak{A}') \neq \hom(\mathfrak{C}', \mathfrak{B}')$. Using Lemma 4.5, we then can construct an acyclic and function σ'-structure that also distinguishes $\mathfrak{A}'$ and $\mathfrak{B}'$ by homomorphism count.

3.* implies *1. The existence of such a σ-structure is, by Lemma 4.4, equivalent to there being an acyclic, total and functional σ'-structure that distinguishes the encodings. Using the results of [19], this implies that RCR distinguishes the encodings and by definition this means that naive RCR distinguishes $\mathfrak{A}$ and $\mathfrak{B}$.

1.* does not imply *3. By Lemma 4.4, Statement *3.* is equivalent to there being an acyclic, total and functional σ'-structure that distinguishes the encodings by homomorphism count. However as we showed in Lemma 4.6, this is not implied by the existence of just an acyclic σ'-structure which distinguishes. This is however equivalent to statement *1.*. □

We found that naive RCR cannot be characterised by acyclic structures, as functions inherently result in cycles. When prohibiting loops and 2-cycles, which would not violate the acyclicity, there thus have to be larger cycles which result in the structure not being acyclic. The notion of acyclicity with respect to distinguishing by homomorphism count that was used here is therefore strictly weaker than naive RCR. However, there could be other classes of structures, which characterise naive RCR in the desired way.

5 Relational Colour Refinement for Symmetric Structures

One interesting question that can be posed when looking at the characterisation of RCR through homomorphism counting is, what restriction to the class of structures can be made. Concretely, which subclass $\mathcal{S}$ of the relational structures can be chosen, such that we get the following equivalence: Two structures $\mathfrak{A}$ and $\mathfrak{B}$ from $\mathcal{S}$ get distinguished by RCR if, and only if, there is an acyclic structure $\mathfrak{C}$ from $\mathcal{S}$ with $\hom(\mathfrak{C}, \mathfrak{A}) \neq \hom(\mathfrak{C}, \mathfrak{B})$.

We have seen that this cannot be done for the class of all total structures. Another possible restriction is the class of all symmetric structures.

Definition 5.1 (Symmetric Structures)**.** Let σ be a relational signature. A structure $\mathfrak{A}$ of signature σ is a symmetric structure, if for every relation and every tuple in those relations, the order of the elements is irrelevant. This means, that every relation R with arity k is a subset of all possible subsets of A with exactly k elements. Formally, that means

$$R \subseteq \binom{A}{k}.$$

An equivalent characterisation uses the symmetric group $\mathcal{S}_k$. We call a σ structure $\mathfrak{A}$ symmetric, if for every $R \in \sigma$ of arity k, every k-tuple $\mathbf{x} = (x_1, x_2, \ldots, x_k) \in R^{\mathfrak{A}}$ and every k-permutation $\pi \in \mathcal{S}_k$

$$(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(k)}) \in R^{\mathfrak{A}}.$$

We will prove in the following that the restriction to symmetric structures in indeed possible. However, before we prove this, we have to show a lemma which will be used in the proof. As a reminder on notation, for a k-tuple $\mathbf{x} = (x_1, x_2, \ldots, x_k)$, a homomorphism φ and a permutation π, we write $\varphi(\mathbf{x})$ for $(\varphi(x_1), \varphi(x_2), \ldots, \varphi(x_k))$ and $\pi(\mathbf{x})$ for $(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(k)})$.

Lemma 5.1. *Let $\pi \in \mathcal{S}_k$, φ be a homomorphism, R a relation of arity k and $\mathbf{x} = (x_1, x_2, \ldots, x_k) \in R$. Then $\varphi(\pi(\mathbf{x})) = \pi(\varphi(\mathbf{x}))$.*

Proof. We show that $\varphi(\pi(\mathbf{x}))_i = \pi(\varphi(\mathbf{x}))_i$ for all $i \in [k]$. Note that the definitions of $\varphi(\pi(\mathbf{x}))$ and $\pi(\varphi(\mathbf{x}))$ are

$$\varphi(\pi(\mathbf{x})) = (\varphi(x_{\pi(1)}), \varphi(x_{\pi(2)}), \ldots, \varphi(x_{\pi(k)}))$$

and

$$\pi(\varphi(\mathbf{x})) = (\varphi(\mathbf{x})_{\pi(1)}, \varphi(\mathbf{x})_{\pi(2)}, \ldots, \varphi(\mathbf{x})_{\pi(k)}).$$

From these, we directly get

$$\varphi(\pi(\mathbf{x}))_i = \varphi(x_{\pi(i)}) = (\varphi(x_1), \varphi(x_2), \ldots, \varphi(x_2))_{\pi(i)} = \varphi(\mathbf{x})_{\pi(i)} = \pi(\varphi(\mathbf{x}))_i.$$

Therefore the lemma must hold. □

We can now prove the main result of this section.

Theorem 5.1. *Let σ be a relational signature and $\mathfrak{A}$ and $\mathfrak{B}$ be two σ structures. Then the following two statements are equivalent:*

1. RCR *distinguishes $\mathfrak{A}$ and $\mathfrak{B}$.*
2. *There exists an acyclic, symmetric σ structure $\mathfrak{C}$ with* $\hom(\mathfrak{C}, \mathfrak{A}) \neq \hom(\mathfrak{C}, \mathfrak{B})$.

Proof. We first prove that *2.* implies *1.* Let $\mathfrak{C}$ be an acyclic, symmetric σ structure with $\hom(\mathfrak{C}, \mathfrak{A}) \neq \hom(\mathfrak{C}, \mathfrak{B})$. As $\mathfrak{C}$ is acyclic, using [19], we get that RCR must distinguish $\mathfrak{A}$ and $\mathfrak{B}$.

We now prove that *1.* implies *2.* Assume that RCR distinguishes $\mathfrak{A}$ and $\mathfrak{B}$. From the results of [19] we know that there exists an acyclic structure $\mathfrak{C}'$ with $\hom(\mathfrak{C}', \mathfrak{A}) \neq \hom(\mathfrak{C}', \mathfrak{B})$. Our goal will be to construct a σ structure $\mathfrak{C}$ from $\mathfrak{C}'$ that is both acyclic and symmetric. Informally, $\mathfrak{C}'$ will have the same elements as $\mathfrak{C}$ and for every tuple that appears in some relation, we will add all possible permutations of that tuple to the relation as well. Formally, we define $\mathfrak{C} \coloneqq (C', \sigma)$ and for all $R \in \sigma$ with arity k, we have

$$R^{\mathfrak{C}} \coloneqq \{(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(k)}) : \text{for every } (x_1, x_2, \ldots, x_k) \in R^{\mathfrak{C}'} \text{ and every } \pi \in \mathcal{S}_k\}.$$

From the second characterisation of symmetric structures given above, it is obvious that $\mathfrak{C}$ is symmetric.

Claim 5.1. *$\mathfrak{C}$ is acyclic.*

Proof. We define a join-tree J for $\mathfrak{C}$. Since $\mathfrak{C}'$ is acyclic, we have a join-tree J' for $\mathfrak{C}'$. From the definition we know that $V(J) = \mathbf{C}$, thus we only have to define the set of edges. Let $\mathbf{x} \in V(J) \setminus V(J')$. From the construction there exists a permutation $\pi_{\mathbf{x}}$, such that $\pi_{\mathbf{x}}(\mathbf{x}) \in V(J')$. We now define $E(J) \coloneqq E(J') \cup \{\{\pi_{\mathbf{x}}(\mathbf{x}), \mathbf{x}\} : \mathbf{x} \in \mathbf{C} \setminus \mathbf{C}'\}$. This construction can be seen in Figure 9.

The connectedness and cycle-freeness follows directly from the fact that J' is also a tree. As such, it only remains to show the join-tree property. Consider an arbitrary $v \in C$. Since $C = C'$, it follows that $v \in C'$ and the set of all $\mathbf{x} \in \mathbf{C}'$ with $v \in \text{set}(\mathbf{x})$ induces a connected subgraph in J'. Let $\mathbf{x} \in \mathbf{C} \setminus \mathbf{C}'$ and $v \in \text{set}(\mathbf{x})$. Then $\pi_{\mathbf{x}}(\mathbf{x}) \in \mathbf{C}'$ and $\{\pi_{\mathbf{x}}(\mathbf{x}), \mathbf{x}\} \in E(J)$, thus $\mathbf{x}$ is also connected and the set $\{\mathbf{x} \in V(J) : v \in \text{set}(\mathbf{x})\}$ also induces a connected subgraph. This was to be shown. □

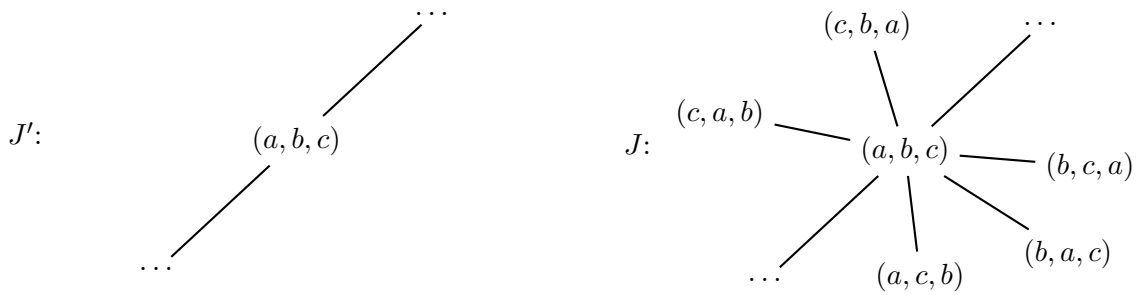


Figure 9: A section from the join tree J' and the join tree J generated from it. We consider a tuple $\mathbf{x} = (a, b, c)$, for which no other permutation appears in $\mathbf{C}'$.

It now remains to prove, that $\mathfrak{C}$ also has a different number of homomorphisms to $\mathfrak{A}$, than to $\mathfrak{B}$. In fact, we will show that $\mathfrak{C}$ and $\mathfrak{C}'$ have exactly the same homomorphisms to $\mathfrak{A}$ and $\mathfrak{B}$, respectively. Formally, we will prove that $\mathrm{Hom}(\mathfrak{C}', \mathfrak{A}) = \mathrm{Hom}(\mathfrak{C}, \mathfrak{A})$ and $\mathrm{Hom}(\mathfrak{C}', \mathfrak{B}) = \mathrm{Hom}(\mathfrak{C}, \mathfrak{B})$. However, we will only prove the claim for $\mathfrak{A}$, as the case for $\mathfrak{B}$ can be proven completely analogously.

Let $\varphi \in \mathrm{Hom}(\mathfrak{C}', \mathfrak{A})$. Then for every $R \in \sigma$, if $\mathbf{x} \in R^{\mathfrak{C}'}$, then $\varphi(\mathbf{x}) \in R^{\mathfrak{A}}$. Now consider $\mathbf{x} \in R^{\mathfrak{C}}$ for a $R \in \sigma$ with arity k and we will proceed with a case distinction. If $\mathbf{x} \in \mathbf{C}'$, then we have $\mathbf{x} \in R^{\mathfrak{C}'}$ and by assumption $\varphi(\mathbf{x}) \in R^{\mathfrak{A}}$. If $\mathbf{x} \in \mathbf{C} \setminus \mathbf{C}'$, then there must be a $\pi \in \mathcal{S}_k$, such that $\pi(\mathbf{x}) \in \mathbf{C}'$ and further $\pi(\mathbf{x}) \in R^{\mathfrak{C}'}$. Then by assumption we get that $\varphi(\pi(\mathbf{x})) \in R^{\mathfrak{A}}$. Using Lemma 5.1, we know that $\varphi(\pi(\mathbf{x})) = \pi(\varphi(\mathbf{x})) \in R^{\mathfrak{A}}$. Now let $\pi' \in \mathcal{S}_k$, such that $\pi' \circ \pi = \mathrm{id} \in \mathcal{S}_k$. As $\mathfrak{A}$ is symmetric, we know that $\pi'(\pi(\varphi(\mathbf{x}))) \in R^{\mathfrak{A}}$ and further we get that $\pi'(\pi(\varphi(\mathbf{x}))) = \varphi(\mathbf{x})$. Therefore $\varphi(\mathbf{x}) \in R^{\mathfrak{A}}$ and $\varphi \in \mathrm{Hom}(\mathfrak{C}, \mathfrak{A})$ follows.

Now let $\varphi \notin \mathrm{Hom}(\mathfrak{C}', \mathfrak{A})$. Then there is a $R \in \sigma$ with arity k and a $\mathbf{x} \in R^{\mathfrak{C}'}$ with $\varphi(\mathbf{x}) \notin R^{\mathfrak{A}}$. From the definition we get that $\mathbf{x} \in \mathbf{C}$ and thus $\mathbf{x} \in R^{\mathfrak{C}}$ and from the assumption we get that $\varphi(\mathbf{x}) \notin R^{\mathfrak{A}}$. Therefore $\varphi \notin \mathrm{Hom}(\mathfrak{C}, \mathfrak{A})$. □

With this, we have proven that it is possible to only consider symmetric acyclic structures with a different homomorphism count, when trying to distinguish symmetric structures. Therefore, we can restrict the class of structures to only consider symmetric structures, while retaining the characterisation through homomorphism counting.

6 Conclusion

In this thesis, we presented the results of Scheidt and Schweikardt [19] and discussed how they can be extended for the use with non-relational structures. We presented two possible ways on how non-relational structures can be encoded as relational structures and investigated the logical and combinatorial characterisations. Naive Relational Colour Refinement, where a function is directly interpreted as a relation, is characterised by the logic $\mathsf{nfGF}(\mathsf{C})$, which uses function symbols like relation symbols. By using the transitive expansion of a function as its encoding, we can allow arbitrarily many function applications, with a bound on the number of alternations. This notion is captured by the logic $\mathsf{GF}(\mathsf{C})_k$. However, while we find a logical characterisation, the combinatorial characterisation is not possible. We showed that the existence of an acyclic, total, functional relational structure that distinguishes two encoded structures by homomorphism count is equivalent to the existence of a non-relational structure that also distinguishes the structures. But while it is possible to construct a functional structure, it is not possible to enforce totality. This then lead us to the question of investigating restrictions of the class of structures and for which restrictions the characterisation by homomorphism counting remains. Aside from the negative result in that regard for total structures,

we showed that it is possible to restrict the class to symmetric structures.

There are multiple questions that are still unanswered.

- When characterising RCR_k logically, we were able to show that the number of applications of a single function symbol does not need a bound. The same was not done for the alternation depth. One interesting question would be to investigate, whether it is possible to only consider the transitive expansion up to a certain alternation depth d. Furthermore, would it then be possible to allow any alternation depth in a formula to characterise this algorithm? The result would then be that $\mathsf{GF}(\mathsf{C})$ with the standard definition atomic formulae would characterise RCR over non-relational structures.

- Our two algorithms operate on different classes of structures. While nRCR is defined for any structure, RCR_k only works for structures over signatures with relation and unary function symbols. It is not clear, whether the used approach can be adapted for functions with arity ≥ 2. However, an iterative colouring algorithm that is stronger than nRCR but still works on any structure is desirable.

- We have considered two possible restrictions on the class of relational structures for characterisation by homomorphism counting. This poses the question, which other restriction could be characterised this way.

References

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, Reading, Mass., reprinted with corr edition, 1996.

[2] László Babai. Graph Isomorphism in Quasipolynomial Time, January 2016. arXiv:1512.03547, doi:10.48550/arXiv.1512.03547.

[3] László Babai, Paul Erdős, and Stanley M. Selkow. Random Graph Isomorphism. *SIAM Journal on Computing*, 9(3):628–635, August 1980. doi:10.1137/0209047.

[4] Christoph Berkholz, Paul Bonsma, and Martin Grohe. Tight Lower and Upper Bounds for the Complexity of Canonical Colour Refinement. *Theory of Computing Systems*, 60(4):581–614, May 2017. doi:10.1007/s00224-016-9686-0.

[5] Jan Böker. Color Refinement, Homomorphisms, and Hypergraphs. In Ignasi Sau and Dimitrios M. Thilikos, editors, *Graph-Theoretic Concepts in Computer Science*, pages 338–350, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-30786-8_26.

[6] Johann Brault-Baron. Hypergraph Acyclicity Revisited, March 2014. arXiv:1403.7076, doi:10.48550/arXiv.1403.7076.

[7] Silvia Butti and Victor Dalmau. Fractional homomorphism, Weisfeiler-Leman invariance, and the Sherali-Adams hierarchy for the Constraint Satisfaction Problem, July 2021. arXiv:2107.02956, doi:10.48550/arXiv.2107.02956.

[8] 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.

[9] Holger Dell, Martin Grohe, and Gaurav Rattan. Lovász Meets Weisfeiler and Leman, May 2018. arXiv:1802.08876, doi:10.48550/arXiv.1802.08876.

[10] Zdeněk Dvořák. On recognizing graphs by numbers of homomorphisms. *Journal of Graph Theory*, 64(4):330–342, 2010. doi:10.1002/jgt.20461.

[11] Erich Grädel. On the Restraining Power of Guards. *The Journal of Symbolic Logic*, 64(4):1719–1742, 1999. arXiv:2586808, doi:10.2307/2586808.

[12] Martin Grohe, Kristian Kersting, Martin Mladenov, and Pascal Schweitzer. Color Refinement and Its Applications. August 2021. doi:10.7551/mitpress/10548.003.0023.

[13] 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*, pages 505–516, Berlin, Heidelberg, 2014. Springer. doi:10.1007/978-3-662-44777-2_42.

[14] Martin Grohe and Pascal Schweitzer. The graph isomorphism problem. *Commun. ACM*, 63(11):128–134, October 2020. doi:10.1145/3372123.

[15] 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.

[16] H. L. Morgan. The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *Journal of Chemical Documentation*, 5(2):107–113, May 1965. doi:10.1021/c160017a018.

[17] Louis C. Ray and Russell A. Kirsch. Finding Chemical Records by Digital Computers. *Science*, 126(3278):814–819, October 1957. doi:10.1126/science.126.3278.814.

[18] Benjamin Scheidt and Nicole Schweikardt. Counting Homomorphisms from Hypergraphs of Bounded Generalised Hypertree Width: A Logical Characterisation, August 2023. arXiv:2303.10980, doi:10.48550/arXiv.2303.10980.

[19] Benjamin Scheidt and Nicole Schweikardt. Color Refinement for Relational Structures, January 2025. arXiv:2407.16022, doi:10.48550/arXiv.2407.16022.

[20] S Vichy N Vishwanathan, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt. Graph kernels. *The Journal of Machine Learning Research*, 11:1201–1242, 2010.

[21] Boris Weisfeiler and Andrei Leman. The reduction of a graph to canonical form and the algebra which appears therein. *nti, Series*, 2(9):12–16, 1968.