

Fractional Homomorphism, Weisfeiler-Leman Invariance, and the Sherali-Adams Hierarchy for the Constraint Satisfaction Problem

Silvia Butti   

Department of Information and Communication Technologies,
Universitat Pompeu Fabra, Barcelona, Spain

Víctor Dalmau   

Department of Information and Communication Technologies,
Universitat Pompeu Fabra, Barcelona, Spain

Abstract

Given a pair of graphs \mathbf{A} and \mathbf{B} , the problems of deciding whether there exists either a homomorphism or an isomorphism from \mathbf{A} to \mathbf{B} have received a lot of attention. While graph homomorphism is known to be NP-complete, the complexity of the graph isomorphism problem is not fully understood. A well-known combinatorial heuristic for graph isomorphism is the Weisfeiler-Leman test together with its higher order variants. On the other hand, both problems can be reformulated as integer programs and various LP methods can be applied to obtain high-quality relaxations that can still be solved efficiently. We study so-called fractional relaxations of these programs in the more general context where \mathbf{A} and \mathbf{B} are not graphs but arbitrary relational structures. We give a combinatorial characterization of the Sherali-Adams hierarchy applied to the homomorphism problem in terms of fractional isomorphism. Collaterally, we also extend a number of known results from graph theory to give a characterization of the notion of fractional isomorphism for relational structures in terms of the Weisfeiler-Leman test, equitable partitions, and counting homomorphisms from trees. As a result, we obtain a description of the families of CSPs that are closed under Weisfeiler-Leman invariance in terms of their polymorphisms as well as decidability by the first level of the Sherali-Adams hierarchy.

2012 ACM Subject Classification Mathematics of computing → Graph theory; Theory of computation → Constraint and logic programming

Keywords and phrases Weisfeiler-Leman algorithm, Sherali-Adams hierarchy, Graph homomorphism, Constraint Satisfaction Problem

Digital Object Identifier 10.4230/LIPIcs.MFCS.2021.27

Related Version *Full Version:* <https://arxiv.org/abs/2107.02956>

Funding *Silvia Butti:* The project that gave rise to these results received the support of a fellowship from “la Caixa” Foundation (ID 100010434). The fellowship code is LCF/BQ/DI18/11660056. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 713673.

Víctor Dalmau: Víctor Dalmau was supported by MICCIN grant PID2019-109137GB-C22.

1 Introduction

The graph isomorphism and homomorphism problems, that is, the problems of deciding, given input graphs \mathbf{A} and \mathbf{B} , whether \mathbf{A} is isomorphic (respectively homomorphic) to \mathbf{B} , have been the subject of extensive research. Despite an important research effort, it is still an open problem to determine whether the isomorphism problem for graphs can be solved in polynomial time. The recent quasipolynomial algorithm for the problem presented by Babai [4] is widely regarded as a major breakthrough in theoretical computer science.



© Silvia Butti and Víctor Dalmau;

licensed under Creative Commons License CC-BY 4.0

46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021).

Editors: Filippo Bonchi and Simon J. Puglisi; Article No. 27; pp. 27:1–27:19

Leibniz International Proceedings in Informatics



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

In turn, the complexity of the graph homomorphism problem has also been intensively studied in combinatorics (see [21]) as well as in a more general setting, known as the Constraint Satisfaction Problem (CSP), where \mathbf{A} and \mathbf{B} are not required to be graphs but can be arbitrary relational structures. The CSP is general enough to encompass problems from areas as diverse as artificial intelligence, optimization, computer algebra, computational biology, computational linguistics, among many other. In contrast with the graph isomorphism problem it was quickly established that the homomorphism problem is NP-complete even for graphs, as it can encode graph coloring. Consequently, an important research effort has been put into identifying tractable fragments of the problem, in particular by fixing the target structure \mathbf{B} . This culminated in the recent major result of Bulatov [8] and Zhuk [37], which confirmed a conjecture by Feder and Vardi [15] that, given a fixed target structure \mathbf{B} , the homomorphism problem for \mathbf{B} is either solvable in polynomial time or NP-complete.

Linear programming relaxations, among other relaxations such as SDP-based, have been largely used in the study of both the isomorphism and the homomorphism problem. In fact, the isomorphism problem for graphs \mathbf{A} , \mathbf{B} can be reformulated as an integer program which asks whether there exists a permutation matrix X such that $XN_{\mathbf{A}} = N_{\mathbf{B}}X$, where $N_{\mathbf{A}}$ and $N_{\mathbf{B}}$ are the adjacency matrices of \mathbf{A} , \mathbf{B} respectively. If we relax this condition to only require that X is doubly stochastic, we obtain what is known as fractional isomorphism.

Fractional isomorphism has a combinatorial counterpart in the 1-dimensional Weisfeiler-Leman (1-WL) algorithm [26], also known as colour refinement. In particular, it was shown in [34, 35] and [30] that \mathbf{A} and \mathbf{B} are fractionally isomorphic if and only if 1-WL does not distinguish between them. 1-WL produces a sequence of colourings c_0, c_1, \dots of the nodes of a graph by means of an iterative refinement procedure which assigns a pair of nodes to the same colour class of c^i if they belong to the same class of c^{i-1} , and additionally they have the same number of neighbours of each colour in c^{i-1} . The algorithm keeps iterating until a fixed point is reached. The Weisfeiler-Leman algorithm is a very powerful heuristic to test for graph isomorphism: if two graphs are distinguished by 1-WL (that is, they give rise to distinct fixed-point colourings up to renaming of the vertices), then this is a witness that the graphs are not isomorphic. In fact, it was shown that 1-WL decides the isomorphism problem on almost all graphs (that is, all but $o(2^{\binom{n}{2}})$ graphs on n vertices for every n) [5].

However, it is also easy to see that 1-WL fails on some very simple instances, such as regular graphs. To address these limitations, the original Weisfeiler-Leman algorithm has been extended so that at every iteration on a graph \mathbf{A} it produces a colouring of the set of k -tuples ($k > 1$) of nodes of \mathbf{A} . It was initially conjectured that this hierarchy of increasingly powerful methods, known as the k -dimensional Weisfeiler-Leman (k -WL) algorithm, would provide a polynomial time graph isomorphism test at least for graphs of bounded degree. While this conjecture was proved to be incorrect [10], k -WL turns out to have a number of useful applications, such as the important role it plays in the aforementioned quasipolynomial algorithm of Babai.

In addition, the k -WL algorithm has proven to be very robust and has a number of equivalent formulations. In [14], a new characterization was given in terms of counting homomorphisms: for every $k \geq 1$, \mathbf{A} and \mathbf{B} are indistinguishable by the k -WL algorithm if for every graph \mathbf{T} of treewidth at most k , the number of homomorphisms from \mathbf{T} to \mathbf{A} is equal to the number of homomorphisms from \mathbf{T} to \mathbf{B} . Further, it has been shown in [10] that \mathbf{A} and \mathbf{B} are indistinguishable by the k -WL algorithm if and only if they are indistinguishable in the logic C^{k+1} consisting of all FO formulas with counting quantifiers that have at most $k + 1$ variables.

Similarly to the Weisfeiler-Leman hierarchy for fractional isomorphism, any LP relaxation of an integer 0/1-program, like that of graph isomorphism, can be further strengthened by sequentially applying so-called lift-and-project methods from mathematical programming in order to obtain a hierarchy of increasingly tighter relaxations which can still be solved efficiently. The main idea behind these is to add auxiliary variables and valid inequalities to an initial relaxation of a 0/1 integer program. These methods, which include Lovász-Schrijver [27] and Sherali-Adams [32], have been used to study classical problems in combinatorial optimization such as Max-Cut, Vertex Cover, Maximal Matching, among many others.

Quite surprisingly, Atserias and Maneva [3] and Malkin [28] were able to lift the connection between the 1-WL algorithm and fractional isomorphism to show a close correspondence between the higher levels of the SA hierarchy for the graph isomorphism problem and the k -WL algorithm (and, hence, with the logic C^{k+1}). This correspondence was further tightened in [20].

Let us now turn our attention to the homomorphism problem. Here, LP relaxations have been intensively used in the more general setting of the Constraint Satisfaction Problem and, most usually, in the approximation of its optimization versions such as MaxCSP, consisting of finding a map from the universe of \mathbf{A} to the universe of \mathbf{B} that maximizes the number of constraints satisfied, among other variants. For instance, a simple linear programming relaxation yields a 2-approximation algorithm for the Vertex Cover problem, and no better polynomial time approximation algorithm is known.

One of the simplest and most widespread LP relaxations of the homomorphism problem (see for example [24]) has an algebraic characterization that very much resembles that of fractional isomorphism. For the sake of simplicity we shall present it here in the restricted case of graphs. Let us start with the algebraic formulation of fractional isomorphism which can be, alternatively, expressed as the existence of a pair of doubly stochastic matrices X and Y such that $XM_{\mathbf{A}} = M_{\mathbf{B}}Y$ and $M_{\mathbf{A}}Y^T = X^TM_{\mathbf{B}}$, where $M_{\mathbf{A}}$ denotes the *incidence* matrix of \mathbf{A} . If we relax this condition to only require that X and Y are left stochastic and, additionally, we drop the second equation, then we obtain a relaxation of graph homomorphism, which we call fractional homomorphism. With some minor variations depending on whether the objective function is present (as in MaxCSP) or not and how repeated elements in a tuple are treated, this LP formulation has been extensively used [7, 12, 13, 18, 23, 24].

We consider relaxations arising from the application of the Sherali-Adams (SA) method. Giving an explicit description of the inequalities produced by the SA method for homomorphism might be relatively cumbersome (because the constraints of the input structures \mathbf{A} and \mathbf{B} must be encoded in the polytope-defining inequalities rather than in the objective function as in the optimization variants) even when the target structure \mathbf{B} is fixed. Hence, we consider a simpler family of LP inequalities, interleaved with the SA hierarchy, of which fractional homomorphism corresponds to the first level. Our hierarchy coincides with the usual SA hierarchy for MaxCSP [11, 17, 33, 36] where the objective function has been turned into a constraint.

Our main result is a combinatorial characterization of this family of LP relaxations which, abusing slightly notation, we still shall refer to as $SA^k, k \geq 1$. Along the way, we extend a number of the aforementioned results from graph isomorphism to general isomorphism between relational structures which yields a hierarchy of relaxations of (relational structure) isomorphism. Our results show that, for every $k \geq 1$, the k th level relaxation of the homomorphism problem, that is, SA^k , is tightly related with the corresponding k th level relaxation of isomorphism, which we denote using \equiv_k .

Let us consider the first level of the hierarchy, i.e, fractional homomorphism and fractional isomorphism. In the case of graphs \equiv_1 -equivalence coincides precisely with 1-WL equivalence. Here we show that a structure \mathbf{A} is fractionally homomorphic to a structure \mathbf{B} if and only if there exists a sequence of structures $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_n$ with $\mathbf{A} = \mathbf{X}_0$, $\mathbf{B} = \mathbf{X}_n$, and for every $i < n$, \mathbf{X}_i is either homomorphic or \equiv_1 -equivalent to \mathbf{X}_{i+1} . The correspondence for any higher level k is obtained by replacing \mathbf{X}_0 and \mathbf{X}_n by suitably defined structures \mathbf{A}_k^* , \mathbf{B}_k^* that allow to reduce the k^{th} level of the hierarchy to the first level.

In particular, it follows that feasibility in SA^k is preserved by structures that are \equiv_k -equivalent.

In essence, this is due to the fact that the LP relaxation of homomorphism inherits the symmetries of \mathbf{A} and \mathbf{B} . There are, indeed, several results exploring the symmetries of an LP program in a similar fashion. In [3] such types of symmetries are used to transfer results between the logic C^k and LP relaxations of several combinatorial problems, whereas [19] aims to identify the partition classes (as in the WL algorithm) of the variables and constraints of an LP program so that, by identifying those in the same class, the LP size is reduced. That being said, the main interest of our result lies precisely in the opposite direction: that is, the fact that the fractional homomorphism LP relaxation is able to certify that \mathbf{A} is not homomorphic to \mathbf{B} unless \mathbf{A} belongs to the backwards closure of \mathbf{B} under homomorphism and \equiv_1 equivalence, or, even more strongly, unless \mathbf{A} is homomorphic to a structure \mathbf{X}_1 which is \equiv_1 -equivalent to a structure \mathbf{X}_2 which in turn is homomorphic to \mathbf{B} .

We apply our results to study the following question: for which structures \mathbf{B} is the set $\text{CSP}(\mathbf{B})$, which contains all the structures homomorphic to \mathbf{B} , closed under 1-WL equivalence. This question arises in the context of solving CSPs in a distributed manner [9] where the elements of the input instance (nodes, edges) are distributed among agents which communicate with each other by sending messages through fixed communication channels. Using our main results we can rederive the classification obtained in [9] using substantially different techniques.

2 Preliminaries

Relational structures

For a positive integer n , We denote by $[n]$ the set $\{1, \dots, n\}$. We shall denote tuples in boldface. Let $\mathbf{a} = (a_1, \dots, a_k) \in A^k$. We use $\mathbf{a}[i]$ to denote a_i and $\{\mathbf{a}\}$ to denote the set of variables which occur in \mathbf{a} . For every tuple $\mathbf{i} = (i_1, \dots, i_n) \in [k]^n$ we use $\pi_{\mathbf{i}} \mathbf{a}$ to denote the *projection* of \mathbf{a} to \mathbf{i} , i.e, the tuple $(a_{i_1}, \dots, a_{i_n})$. If $I \subseteq [k]$ we might abuse slightly notation and use $\pi_I \mathbf{a}$ to refer to $\pi_{\mathbf{i}} \mathbf{a}$ where \mathbf{i} is the tuple that contains the elements of I in increasing order. For every function f on a domain containing $\{\mathbf{a}\}$, we denote by $f(\mathbf{a})$ the coordinate-wise application of f to \mathbf{a} .

Given a set A and a positive integer k , a k -ary *relation* over A is a subset of A^k . A *signature* σ is a finite collection of relation symbols, each with an associated arity. We shall use $\text{arity}(R)$ to denote the arity of a relation symbol R . A *relational structure* \mathbf{A} over σ , or simply a σ -structure, consists of a set A called the *universe* of \mathbf{A} , and a relation $R(\mathbf{A})$ over A for each $R \in \sigma$ of the corresponding arity. We denote by $\mathcal{C}_{\mathbf{A}}$ the set $\{(\mathbf{a}, R) \mid \mathbf{a} \in R(\mathbf{A}), R \in \sigma\}$. Elements (\mathbf{a}, R) of $\mathcal{C}_{\mathbf{A}}$ will alternatively be denoted $R(\mathbf{a})$ and will be referred to as *constraints*. We shall usually use the same boldface and (standard) capital letter to refer to a structure and its universe, respectively.

A *graph* is a relational structure whose signature consists of a single binary relation that is symmetric and non-reflexive.

Let \mathbf{A}, \mathbf{B} be σ -structures. A *homomorphism* from \mathbf{A} to \mathbf{B} is a map $h : A \rightarrow B$ such that for every $R \in \sigma$ and every $\mathbf{a} \in R(\mathbf{A})$ it holds that $h(\mathbf{a}) \in R(\mathbf{B})$. If there exists a homomorphism from \mathbf{A} to \mathbf{B} we say that \mathbf{A} is homomorphic to \mathbf{B} and we write $\mathbf{A} \rightarrow \mathbf{B}$. We shall use $\text{Hom}(\mathbf{A}; \mathbf{B})$ to denote the number of homomorphisms from \mathbf{A} to \mathbf{B} . The problem of deciding, given two similar structures \mathbf{A} and \mathbf{B} , whether \mathbf{A} is homomorphic to \mathbf{B} is known as the *Constraint Satisfaction Problem* (CSP). If we fix the target structure \mathbf{B} so that the input is only \mathbf{A} , then we obtain the Constraint Satisfaction Problem over \mathbf{B} , denoted $\text{CSP}(\mathbf{B})$.

An *isomorphism* from \mathbf{A} to \mathbf{B} is a bijective map $f : A \rightarrow B$ such that for every $R \in \sigma$ and every $\mathbf{a} \in A^{\text{arity}(R)}$ it holds that $\mathbf{a} \in R(\mathbf{A})$ if and only if $f(\mathbf{a}) \in R(\mathbf{B})$.

The *union* $\mathbf{A} \cup \mathbf{B}$ of two σ -structures \mathbf{A} and \mathbf{B} is the structure \mathbf{C} with $C = A \cup B$ and $R(\mathbf{C}) = R(\mathbf{A}) \cup R(\mathbf{B})$ for every $R \in \sigma$. The *disjoint union* of two structures \mathbf{A} and \mathbf{B} is the structure $\mathbf{A} \cup \mathbf{C}$ where \mathbf{C} is any σ -structure isomorphic to \mathbf{B} satisfying $A \cap C = \emptyset$. We say that a structure is *connected* if it cannot be expressed as the disjoint union of two structures. We say that \mathbf{A} is a *substructure* of \mathbf{B} if $\mathbf{A} \cup \mathbf{B} = \mathbf{B}$. If, in addition, $R(\mathbf{A}) = R(\mathbf{B}) \cap A^{\text{arity}(R)}$ for every $R \in \sigma$ then \mathbf{A} is the substructure of \mathbf{B} *induced* by A .

Tree-like structures

We define the *factor graph*¹ of a structure \mathbf{A} to be the bipartite graph with nodes $A \cup \mathcal{C}_{\mathbf{A}}$ and where every $R(\mathbf{a})$ in $\mathcal{C}_{\mathbf{A}}$ is joined by an edge with every $a \in \{\mathbf{a}\}$.

Then, we say that a structure \mathbf{T} is an *ftree* (factor tree) if its factor graph is a tree in the ordinary graph-theoretic sense. If \mathbf{Q} is a substructure of \mathbf{T} and \mathbf{Q} is an ftree then we say that \mathbf{Q} is a *subftree* of \mathbf{T} .

A *tree-decomposition* of a structure \mathbf{A} is a pair (G, β) where $G = (V, E)$ is a tree and $\beta : V \rightarrow \mathcal{P}(A)$ is a mapping such that the following conditions are satisfied:

1. For every constraint $R(\mathbf{a})$ in $\mathcal{C}_{\mathbf{A}}$ there exists a node $v \in V$ such that $\{\mathbf{a}\} \subseteq \beta(v)$
2. If $a \in \beta(u) \cap \beta(v)$ then $a \in \beta(w)$ for every node w in the unique path in G joining u to v .

The *width* of a tree-decomposition (G, β) is $\max\{|\beta(v)| \mid v \in V\} - 1$ and the *treewidth* of \mathbf{A} is defined as the smallest width among all its tree-decompositions.

The Sherali–Adams hierarchy

In this presentation we follow [3]. Let $P \subseteq [0, 1]^n$ be a polytope $\{\mathbf{x} \in \mathbb{R}^n : M\mathbf{x} \geq \mathbf{b}, 0 \leq \mathbf{x} \leq 1\}$ for a matrix $M \in \mathbb{R}^{m \times n}$, and a column vector $\mathbf{b} \in \mathbb{R}^m$. We denote the convex hull of the $\{0, 1\}$ -vectors in P by $P^{\mathbb{Z}}$. The sequence of Sherali-Adams relaxations of $P^{\mathbb{Z}}$ is the sequence of polytopes $P = P^1 \supseteq P^2 \supseteq \dots$ where P^k is defined in the following way.

Each inequality in $M\mathbf{x} \geq \mathbf{b}$ is multiplied by all possible terms of the form $\prod_{i \in I} x_i \prod_{j \in J} (1 - x_j)$ where $I, J \subseteq [n]$ satisfy $|I \cup J| \leq k - 1$ and $I \cap J = \emptyset$. This leaves a system of polynomial inequalities, each of degree at most k . Then, this system is *linearized* and hence relaxed in the following way: each square x_i^2 is replaced by x_i and each resulting monomial $\prod_{i \in K} x_i$ is replaced by a variable y_K . In this way we obtain a polytope P_L^k . Finally, P_L^k is projected back to n dimensions by defining

$$P^k := \{\mathbf{x} \in \mathbb{R}^n : \text{there exists } \mathbf{y} \in P_L^k \text{ such that } \mathbf{y}_{\{i\}} = \mathbf{x}_i \text{ for each } i \in [n]\}.$$

We note here that $P^{\mathbb{Z}} \subseteq P^k$ for every $k \geq 1$.

¹ We note that the definition of factor graph presented here differs from that in [9], in which the edges are labelled. We also note that the notion of factor graph, although similar, differs in several ways from the incidence multigraph (see [25]) as the latter allows for parallel edges.

In order to apply the SA method to the homomorphism problem there are different possible choices for the polytope P (encoding a relaxation of homomorphism) to start with, each one then yielding a different hierarchy.

Here we shall adapt a SA-based family of relaxations commonly used in optimization variants of CSP [11, 17, 33, 36] which we transform into a relaxation of (plain) CSP by just turning the objective function into a set of new restrictions. Hence, the resulting system of inequalities is not, strictly speaking, obtained using the SA method. Nonetheless, we shall abuse slightly notation and still use SA^k to refer to our system of inequalities.

In fact, giving an explicit description of all inequalities obtained using the SA method for any natural polytope P encoding the LP relaxation for a *general* CSP in our setting is a bit cumbersome (because the constraints of the CSP are encoded in the polytope-defining inequalities instead of the objective function as in CSP optimization variants). Hence, it seems sensible to settle for a good approximation as SA^k . Indeed, as it can be seen in Appendix A, the sequence of relaxations SA^k is tightly interleaved with the sequence P^k obtained by the SA method, in stricto sensu, for a natural choice of initial polytope P .

Given two structures \mathbf{A} and \mathbf{B} , the system of inequality $\text{SA}^k(\mathbf{A}, \mathbf{B})$ for the homomorphism problem over (\mathbf{A}, \mathbf{B}) contains a variable $p_V(f)$ for every $V \subseteq A$ with $1 \leq |V| \leq k$ and every $f : V \rightarrow B$, and a variable $p_{R(\mathbf{a})}(f)$ for every $R(\mathbf{a}) \in \mathcal{C}_{\mathbf{A}}$ and every $f : \{\mathbf{a}\} \rightarrow B$. Each variable must take a value in the range $[0, 1]$. The variables are constrained by the following conditions:

$$\sum_{f:V \rightarrow B} p_V(f) = 1 \quad V \subseteq A \text{ s.t. } |V| \leq k \quad (\text{SA1})$$

$$p_U(f) = \sum_{g:V \rightarrow B, g|_U=f} p_V(g) \quad U \subseteq V \subseteq A \text{ s.t. } |V| \leq k, f : U \rightarrow B \quad (\text{SA2})$$

$$p_U(f) = \sum_{g:V \rightarrow B, g|_U=f} p_{R(\mathbf{a})}(g) \quad R(\mathbf{a}) \in \mathcal{C}_{\mathbf{A}}, U \subseteq \{\mathbf{a}\} = V \text{ s.t. } |U| \leq k, f : U \rightarrow B \quad (\text{SA3})$$

$$p_{R(\mathbf{a})}(f) = 0 \quad R(\mathbf{a}) \in \mathcal{C}_{\mathbf{A}}, f : \{\mathbf{a}\} \rightarrow B \text{ s.t. } f(\mathbf{a}) \notin R(\mathbf{B}) \quad (\text{SA4})$$

For the particular case of $k = 1$ we shall use the simplified notation $p_v(f(v))$ to denote the variable $p_V(f)$ for a singleton set $V = \{v\}$ and a function $f : V \rightarrow B$.

The transformation $*_k$

For every $k > 0$ we define an operator $*_k$ that maps a structure into a new structure. As we shall see later this operator will allow to reduce SA^k feasibility to SA^1 feasibility.

Let \mathbf{A} be a σ -structure. Then we define the universe of \mathbf{A}_k^* to be $A_k^* := \cup_{j \leq k} A^j \cup \mathcal{C}_{\mathbf{A}}$. Additionally, \mathbf{A}_k^* contains the following relations:

$$\begin{aligned} T_{j,S}(\mathbf{A}_k^*) &= \{(a_1, \dots, a_j) \in A^j \mid a_i = a_{i'} \forall i, i' \in S\} & j \leq k, S \subseteq [j] \\ T_{j,\mathbf{i}}(\mathbf{A}_k^*) &= \{(\mathbf{a}, \pi_{\mathbf{i}} \mathbf{a}) \mid \mathbf{a} \in A^j\} & j', j \leq k, \mathbf{i} \in [j]^{j'} \\ R_S(\mathbf{A}_k^*) &= \{(a_1, \dots, a_{\text{arity}(R)}) \in R(\mathbf{A}) \mid a_i = a_{i'} \forall i, i' \in S\} & R \in \sigma, S \subseteq [\text{arity}(R)] \\ R_{\mathbf{i}}(\mathbf{A}_k^*) &= \{(R(\mathbf{a}), \pi_{\mathbf{i}} \mathbf{a}) \mid \mathbf{a} \in R(\mathbf{A})\} & R \in \sigma, j \leq k, \mathbf{i} \in [\text{arity}(R)]^j. \end{aligned}$$

Then we have:

► **Lemma 1.** *Let \mathbf{A}, \mathbf{B} be σ -structures. Then $\text{SA}^k(\mathbf{A}, \mathbf{B})$ is feasible if and only if $\text{SA}^1(\mathbf{A}_k^*, \mathbf{B}_k^*)$ is feasible.*

Iterated degree, fractional isomorphism, and equitable partitions

In order to prove our main result we need to lift the known equivalence between fractional isomorphism and the WL algorithm from graphs to relational structures.

Let $L := \{(S, R) \mid R \in \sigma, S \subseteq [\text{arity}(R)]\}$ be a set, the elements of which we shall call *labels*. We construct the *matrix representation* $M_{\mathbf{A}}$ of a σ -structure \mathbf{A} as follows (it will be convenient to assume that the indices of the rows and columns of a matrix are arbitrary sets). $M_{\mathbf{A}}$ is an $A \times \mathcal{C}_{\mathbf{A}}$ matrix whose entries are elements of L . In particular, for all $a \in A$ and $R(\mathbf{a}) \in \mathcal{C}_{\mathbf{A}}$, we have that $M_{\mathbf{A}}[a, R(\mathbf{a})] = (S, R)$ where S is the set containing all elements $i \in [\text{arity}(R)]$ such that $a = \mathbf{a}[i]$.

In a nutshell we are lifting from graph isomorphism to matrix isomorphism, where two matrices are isomorphic if they are identical modulo a permutation of the rows and columns. To formalize this, it will be convenient to associate \mathbf{A} with a set of 0-1 incidence matrices. In particular, for every $\ell \in L$ we define $M_{\mathbf{A}}^{\ell} \in \{0, 1\}^{A \times \mathcal{C}_{\mathbf{A}}}$ as follows: $M_{\mathbf{A}}^{\ell}[a, R(\mathbf{a})] = 1$ if $M_{\mathbf{A}}[a, R(\mathbf{a})] = \ell$ and $M_{\mathbf{A}}^{\ell}[a, R(\mathbf{a})] = 0$ otherwise. In [19], relaxations of matrix isomorphism are also considered although in that setting the matrices have real entries and the goal is different from ours.

We now describe a procedure akin to the 1-dimensional Weisfeiler-Leman algorithm to calculate iterative refinements of a colouring of the universe and constraint set of a relational structure. While there are syntactical differences, when run on graphs this procedure is equivalent for all purposes to 1-WL. For every $k \geq 0$ and $x \in A \cup \mathcal{C}_{\mathbf{A}}$, we define inductively the *iterated degree* $\delta_k^{\mathbf{A}}(x)$ of x on \mathbf{A} as follows. We set $\delta_0^{\mathbf{A}}(x)$ to be one of two arbitrary symbols that distinguish elements of A from elements of $\mathcal{C}_{\mathbf{A}}$. For $k \geq 1$ we set $\delta_k^{\mathbf{A}}(a) = \{ \{ (\ell, \delta_{k-1}^{\mathbf{A}}(\mathbf{a}, R)) \mid M_{\mathbf{A}}^{\ell}[a, R(\mathbf{a})] = 1 \} \}$ and $\delta_k^{\mathbf{A}}(\mathbf{a}, R) = \{ \{ (\ell, \delta_{k-1}^{\mathbf{A}}(a)) \mid M_{\mathbf{A}}^{\ell}[a, R(\mathbf{a})] = 1 \} \}$, where double curly brackets denote that $\delta_k^{\mathbf{A}}(x)$ is a multiset.

We say that \mathbf{A} and \mathbf{B} have the same iterated degree sequence if for every $k \geq 0$, $\{ \{ \delta_k^{\mathbf{A}}(a) \mid a \in A \cup \mathcal{C}_{\mathbf{A}} \} \} = \{ \{ \delta_k^{\mathbf{B}}(b) \mid b \in B \cup \mathcal{C}_{\mathbf{B}} \} \}$. Note that if there exists a matrix isomorphism from \mathbf{A} to \mathbf{B} (or alternatively, \mathbf{A} and \mathbf{B} are isomorphic) then \mathbf{A} and \mathbf{B} have the same iterated degree sequence, but the converse does not hold.

The notion of equitable partition is key in the proof of the equivalence of the different characterizations of fractional isomorphism. We present its adaptation to relational structures. A *partition* of a σ -structure \mathbf{A} is a pair (P, Q) where $P = \{P_i \mid i \in I\}$ is a partition of A and $Q = \{Q_j \mid j \in J\}$ is a partition of $\mathcal{C}_{\mathbf{A}}$. We say that (P, Q) is *equitable* if for every $i \in I$, $j \in J$, and $\ell \in L$, there are integers $c_{i,j}^{\ell}, d_{j,i}^{\ell}$, called the *parameters* of the partition, such that for every every $i \in I$, every $a \in P_i$, every $\ell \in L$, and every $j \in J$, we have

$$|\{(\mathbf{a}, R) \in Q_j \mid M_{\mathbf{A}}[a, R(\mathbf{a})] = \ell\}| = c_{i,j}^{\ell} \quad (\text{P1})$$

and, similarly, for every $j \in J$, every $R(\mathbf{a}) \in Q_j$, every $\ell \in L$, and every $i \in I$ we have

$$|\{a \in P_i \mid M_{\mathbf{A}}[a, (\mathbf{a}, R)] = \ell\}| = d_{j,i}^{\ell}. \quad (\text{P2})$$

We say that two structures \mathbf{A}, \mathbf{B} have a *common equitable partition* if there are equitable partitions $(\{P_i^{\mathbf{A}} \mid i \in I\}, \{Q_j^{\mathbf{A}} \mid j \in J\})$ and $(\{P_i^{\mathbf{B}} \mid i \in I\}, \{Q_j^{\mathbf{B}} \mid j \in J\})$ of \mathbf{A} and \mathbf{B} with the same parameters satisfying $|P_i^{\mathbf{A}}| = |P_i^{\mathbf{B}}|$ for every $i \in I$ and $|Q_j^{\mathbf{A}}| = |Q_j^{\mathbf{B}}|$ for every $j \in J$. We note that if \mathbf{A} and \mathbf{B} are connected it is not necessary to verify this latter requirement and, instead, it is enough to check that $|A| = |B|$.

A matrix M of non-negative real numbers is said to be *left* (resp. *right*) *stochastic* if all its columns (resp. rows) sum to 1. Note that we do not require M to be square. A *doubly stochastic matrix* is a square matrix that is both left and right stochastic.

► **Theorem 2.** *Let \mathbf{A}, \mathbf{B} be σ -structures. The following are equivalent:*

1. *There exist doubly stochastic matrices X, Y such that $XM_{\mathbf{A}}^{\ell} = M_{\mathbf{B}}^{\ell}Y$ and $M_{\mathbf{A}}^{\ell}Y^T = X^T M_{\mathbf{B}}^{\ell}$ for every $\ell \in L$;*
2. *\mathbf{A} and \mathbf{B} have the same iterated degree sequence;*
3. *\mathbf{A} and \mathbf{B} have a common equitable partition;*
4. *$\text{Hom}(\mathbf{T}, \mathbf{A}) = \text{Hom}(\mathbf{T}, \mathbf{B})$ for all σ -ftrees \mathbf{T} .*

If, additionally, \mathbf{A} and \mathbf{B} are graphs, then the following is also equivalent:

5. *There exists a doubly stochastic matrix X such that $XN_{\mathbf{A}} = N_{\mathbf{B}}X$ where $N_{\mathbf{A}}$ and $N_{\mathbf{B}}$ denote the adjacency matrices of \mathbf{A} and \mathbf{B} respectively.*

We refer the reader to the full version of this paper for the proof. The equivalence between (1), (2) and (3) is an immediate generalization of the equivalence between the algebraic and combinatorial characterizations of fractional graph isomorphism (see for example [31]), while (4) generalizes the corresponding characterization of graphs from [14] in terms of counting homomorphisms from trees. Also, note that condition (5) shows that our definition coincides with the standard notion of fractional isomorphism when \mathbf{A} and \mathbf{B} are graphs.

Similarly to the case of graphs, the notion of fractional isomorphism captured in Theorem 2 can be strengthened giving rise to a hierarchy of increasingly tighter relaxations. In particular, for every $k \geq 1$, we shall denote $\mathbf{A} \equiv_k \mathbf{B}$ whenever \mathbf{A}_k^* and \mathbf{B}_k^* satisfy the conditions of Theorem 2.

It is easy to see that the case $k = 1$ would be unchanged if one replaces \mathbf{A}_1^* and \mathbf{B}_1^* by \mathbf{A} and \mathbf{B} respectively and, hence, Theorem 2 characterizes \equiv_1 . For other small values of k other than $k = 1$, \equiv_k is a bit more difficult to characterize. However, as long as k is at least as large as the arity of any relation in the signature then we have the following result.

► **Lemma 3.** *Let r be the maximum arity among all relations in σ and assume that $r \leq k$. Then for every pair of structures \mathbf{A}, \mathbf{B} the following are equivalent:*

1. $\mathbf{A} \equiv_k \mathbf{B}$
2. $\text{Hom}(\mathbf{Q}; \mathbf{A}) = \text{Hom}(\mathbf{Q}; \mathbf{B})$ for every structure \mathbf{Q} of treewidth $< k$.

This result, which follows easily from condition (4) in Theorem 2, is inspired by a similar result [14] which states that two graphs \mathbf{A}, \mathbf{B} are indistinguishable by the k -WL algorithm if and only if $\text{Hom}(\mathbf{Q}; \mathbf{A}) = \text{Hom}(\mathbf{Q}; \mathbf{B})$ for every graph \mathbf{Q} of treewidth $\leq k$. It is not that surprising that a similar result can be shown for \equiv_k since, after all, the k -WL algorithm can be seen as the 1-WL algorithm applied to k -ary tuples. However, note that the bound on the treewidth differs in one unit between k -WL and \equiv_k . Still, using Lemma 3 it can be shown that for $r \leq k$, \equiv_k can be alternatively characterized in logical terms extending, again, an analogous result for k -WL [22]. More precisely, $\mathbf{A} \equiv_k \mathbf{B}$ if and only if \mathbf{A} and \mathbf{B} satisfy the same formulas in the k -variable fragment of first-order logic with counting quantifiers, denoted C^k (we omit the definition as it will not be needed).

2.1 Main results

The main result of this paper is a new characterization in terms of fractional isomorphism of the Sherali-Adams relaxation of the homomorphism problem.

► **Theorem 4.** *Let \mathbf{A}, \mathbf{B} be relational structures. Then, the following are equivalent:*

1. $\text{SA}^k(\mathbf{A}, \mathbf{B})$ is feasible;
2. *There exists a sequence of structures $\mathbf{X}_0, \dots, \mathbf{X}_n$ such that $\mathbf{X}_0 = \mathbf{A}_k^*$, $\mathbf{X}_n = \mathbf{B}_k^*$, and for all $i = 0, \dots, n-1$ we have that $\mathbf{X}_i \rightarrow \mathbf{X}_{i+1}$ or $\mathbf{X}_i \equiv_1 \mathbf{X}_{i+1}$;*
3. *There exist structures $\mathbf{X}_1, \mathbf{X}_2$ such that $\mathbf{A}_k^* \rightarrow \mathbf{X}_1$, $\mathbf{X}_1 \equiv_1 \mathbf{X}_2$, and $\mathbf{X}_2 \rightarrow \mathbf{B}_k^*$.*

Theorem 4 is an immediate consequence, using Lemma 1, of the following Theorem.

► **Theorem 5.** *Let \mathbf{A}, \mathbf{B} be relational structures. Then, the following are equivalent:*

1. $\text{SA}^1(\mathbf{A}, \mathbf{B})$ is feasible;
2. There exist left stochastic matrices X, Y such that for every $\ell = (S, R) \in L$, it holds that $XM_{\mathbf{A}}^{\ell} \leq \sum_{\ell'} M_{\mathbf{B}}^{\ell'} Y$ where ℓ' ranges over all $(S', R) \in L$ with $S \subseteq S'$;
3. There exists a sequence of structures $\mathbf{X}_0, \dots, \mathbf{X}_n$ such that $\mathbf{X}_0 = \mathbf{A}$, $\mathbf{X}_n = \mathbf{B}$, and for all $i = 0, \dots, n-1$ we have that $\mathbf{X}_i \rightarrow \mathbf{X}_{i+1}$ or $\mathbf{X}_i \equiv_1 \mathbf{X}_{i+1}$;
4. There exist structures $\mathbf{X}_1, \mathbf{X}_2$ such that $\mathbf{A} \rightarrow \mathbf{X}_1$, $\mathbf{X}_1 \equiv_1 \mathbf{X}_2$, and $\mathbf{X}_2 \rightarrow \mathbf{B}$.

If in addition \mathbf{A} and \mathbf{B} have no loops (meaning that there are no repeated elements in any constraint) then the following condition is also equivalent:

5. There exist left stochastic matrices X, Y such that for every $\ell \in L$ it holds that $XM_{\mathbf{A}}^{\ell} = M_{\mathbf{B}}^{\ell} Y$.

Note that for graphs condition (5) of the above theorem is naturally seen as the homomorphism counterpart of the notion of fractional isomorphism (see condition (1) in Lemma 2). Consequently, we shall say that \mathbf{A} is fractionally homomorphic to \mathbf{B} whenever \mathbf{A} and \mathbf{B} satisfy the conditions of Theorem 5.

While Theorems 4 and 5 have applications, for instance in the field of Constraint Satisfaction Problems (see Section 4), we believe that their main interest is that they shed light on the LP relaxations for homomorphism as they show that fractional homomorphism and its higher order counterparts can be decomposed into a sequence of basic, better studied morphisms, a fact which we find interesting on its own. Additionally, they establish a close link between LP relaxations for isomorphism and homomorphism, which were introduced initially in different fields.

3 Proof of Theorem 5

The equivalence (1) \Leftrightarrow (2) is merely syntactic. In particular we shall show that there is a one-to-one satisfiability-preserving correspondence between pairs of matrices and variable assignments of $\text{SA}^1(\mathbf{A}, \mathbf{B})$. However, we first need to massage a bit the two formulations. First, we can assume that for every $R^A(\mathbf{a}) \in \mathcal{C}_{\mathbf{A}}$ and $R^B(\mathbf{b}) \in \mathcal{C}_{\mathbf{B}}$, the corresponding entry in Y is null unless $R^A = R^B$ and $f(\mathbf{a}) = \mathbf{b}$ for some $f : \{\mathbf{a}\} \rightarrow \{\mathbf{b}\}$, since otherwise there is no way that Y can be part of a feasible solution. Secondly, we note that the feasibility of $\text{SA}^1(\mathbf{A}, \mathbf{B})$ does not change if in (SA3) we replace $=$ by \leq obtaining a new set of inequalities (which to avoid confusion we shall denote by (SA3')) and, in addition, we add for every $R(\mathbf{a}) \in \mathcal{C}_{\mathbf{A}}$ the equality

$$\sum_{f: \{\mathbf{a}\} \rightarrow B} p_{R(\mathbf{a})}(f) = 1. \quad (\text{SA5})$$

Finally, note that in $\text{SA}^1(\mathbf{A}, \mathbf{B})$ we can ignore (SA2).

Then we can establish the following correspondence between pairs of matrices X, Y and assignments $\text{SA}^1(\mathbf{A}, \mathbf{B})$: for every $a \in A$ and $b \in B$, we set $p_a(b) = X[b, a]$ and for every $R(\mathbf{a}) \in \mathcal{C}_{\mathbf{A}}$ and $f : \{\mathbf{a}\} \rightarrow B$ we define $p_{R(\mathbf{a})}(f) = Y[R(f(\mathbf{a})), R(\mathbf{a})]$. Then, it is easy to see that (SA3') corresponds to $XM_{\mathbf{A}}^{\ell} \leq \sum_{\ell' \in L_{\ell}} M_{\mathbf{B}}^{\ell'} Y$ for every $\ell \in L$ (where $L_{(S,R)} := \{(S', R) \in L \mid S \subseteq S'\}$), X being left stochastic corresponds to (SA1), and Y being left stochastic corresponds to (SA5).

The equivalence (1) \Leftrightarrow (5) is obtained as in (1) \Leftrightarrow (2). We just need to notice that when \mathbf{A} and \mathbf{B} have no loops, then none of the entries in $M_{\mathbf{A}}$ and $M_{\mathbf{B}}$ contain any label $\ell = (S, R) \in L$ where $|S| > 1$ and hence it is only necessary to consider labels $\ell = (S, R) \in L$ where S is a singleton. Observe that, in this case, the equation in (2) becomes $XM_{\mathbf{A}}^{\ell} \leq M_{\mathbf{B}}^{\ell} Y$

since for every label $\ell = (S, R)$ where S is a singleton, the only label (S', R) with $S \subseteq S'$ and $M_{\mathbf{B}}^{\ell}$ not a zero matrix is ℓ itself. Finally, in order to replace \leq by $=$ in the previous equation we just need to use (SA3) instead of (SA3').

Notice that (4) \implies (3) is trivial.

The proof of (3) \implies (2) is by induction on n . If $n = 0$ the claim is immediate, so assume that $n \geq 1$. Let $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_n$ be a sequence of structures satisfying (3). By the induction hypothesis, there exist left stochastic matrices X, Y such that $XM_{\mathbf{X}_1}^{\ell} \leq \sum_{\ell' \in L_{\ell}} M_{\mathbf{X}_n}^{\ell'} Y$ for all $\ell \in L$.

If $\mathbf{X}_0 \equiv_1 \mathbf{X}_1$ then it follows from Theorem 2 that there exist doubly stochastic matrices X' and Y' such that $X'M_{\mathbf{X}_0}^{\ell} = M_{\mathbf{X}_1}^{\ell} Y'$ for all $\ell \in L$, and so it is easy to verify that XX', YY' are such that (2) holds. Assume that $\mathbf{X}_0 \not\equiv_1 \mathbf{X}_1$. We shall show that there exist left stochastic matrices X' and Y' such that for all $b \in B$, for all $R(\mathbf{a}) \in \mathcal{C}_{\mathbf{A}}$, and for all $\ell \in L$ there exists $\hat{\ell} = \hat{\ell}(b, R(\mathbf{a}), \ell) \in L_{\ell}$ such that $XA^{\ell}[b, R(\mathbf{a})] \leq B^{\hat{\ell}}Y[b, R(\mathbf{a})]$. Assuming that this holds, again it follows by the induction hypothesis that XX', YY' are left stochastic matrices such that $XX'M_{\mathbf{X}_0}^{\ell} \leq \sum_{\ell' \in L_{\ell}} M_{\mathbf{X}_n}^{\ell'} YY'$ for all $\ell \in L$.

Let h be a homomorphism from \mathbf{A} to \mathbf{B} . We define $X'[b, a] = 1$ if $b = h(a)$ and $X'[b, a] = 0$ otherwise. Similarly, we set $Y'[R^B(\mathbf{b}), R^A(\mathbf{a})] = 1$ if $\mathbf{b} = h(\mathbf{a})$ and $R^B = R^A$ and $Y'[R^B(\mathbf{b}), R^A(\mathbf{a})] = 0$ otherwise. It is easy to see that X' and Y' are left stochastic. Now let $\ell = (S, R) \in L$, $b \in B$ and $R(\mathbf{a}) \in \mathcal{C}_{\mathbf{A}}$. If $M_{\mathbf{A}}^{\ell}[a, R(\mathbf{a})] = 0$ for all $a \in A$ then $XM_{\mathbf{A}}^{\ell}[b, R(\mathbf{a})] = 0$ and there is nothing to prove. So we can assume that there is $a \in A$ such that for all $i \in [\text{arity}(R)]$, $\mathbf{a}[i] = a$ if and only if $i \in S$. Then we have that $XM_{\mathbf{A}}^{\ell}[b, R(\mathbf{a})] = 1$ if $b = h(a)$, and $XM_{\mathbf{A}}^{\ell}[b, R(\mathbf{a})] = 0$ otherwise. Again in the latter case there is nothing to prove so let us assume that $b = h(a)$. It follows that $h(\mathbf{a}[i]) = b$ for all $i \in S$ and hence there exists $\hat{\ell} = (R, S')$ with $S \subseteq S'$ such that $M_{\mathbf{B}}^{\hat{\ell}}[b, R(h(\mathbf{a}))] = 1$, which completes the proof.

It only remains to prove (1) \implies (4). Assume that \mathbf{A} and \mathbf{B} satisfy (1). Further, we can assume that there exists an integer $m > 0$ such that all variables in the feasible solution of $\text{SA}^1(\mathbf{A}, \mathbf{B})$ take rational values of the form n/m for some integer n . Let Y be the set of all tuples $((b_1, c_1), \dots, (b_m, c_m)) \in (B \times [m])^m$ satisfying the following conditions:

- $(b_i, c_i) \neq (b_{i'}, c_{i'})$ for every $i \neq i' \in [m]$ (i.e., the tuple has no repeated elements), and
- for every $i \in [m]$, c_i is at most $|\{j \in [m] \mid b_j = b_i\}|$.

In other words, Y is the set of tuples that can be obtained if in every tuple $(b_1, \dots, b_m) \in B^m$ we replace all occurrences, say n , of any symbol $b \in B$ by n ‘‘copies’’ $(b, 1), \dots, (b, n)$, in any possible order. Let $\Pi : Y \rightarrow B^m$ be the function mapping $((b_1, c_1), \dots, (b_m, c_m))$ to (b_1, \dots, b_m) .

Let $X = A \times Y$. For every permutation τ on $[m]$ and any m -ary tuple $\mathbf{z} = (z_1, \dots, z_m) \in Z^m$ where Z is any arbitrary set we shall use $\mathbf{z} \circ \tau$ to denote the tuple $(z_{\tau(1)}, \dots, z_{\tau(m)})$. This notation is justified by the fact that we can see formally \mathbf{z} as a mapping from $[m]$ to Z . For any $x = (a, \mathbf{y}) \in X$, we shall abuse notation and write $x \circ \tau$ to denote $(a, \mathbf{y} \circ \tau)$.

For any two \mathbf{z}, \mathbf{z}' in any of the sets B^m, Y or X , we shall write $\mathbf{z} \sim \mathbf{z}'$ iff there exists some permutation τ on $[m]$ such that $\mathbf{z}' = \mathbf{z} \circ \tau$.

For every multiset $U = \{\{b_1, \dots, b_m\}\}$ of m elements from B , we shall fix one arbitrary element of Y , denoted $\langle U \rangle$, satisfying that the multiset of variables in $\Pi(\langle U \rangle)$ coincides with U . We shall abuse slightly notation and for every $\mathbf{b} = (b_1, \dots, b_m) \in B^m$ we shall also use $\langle \mathbf{b} \rangle$ to denote $\langle \{\{b_1, \dots, b_m\}\} \rangle$. Clearly if $\mathbf{b} \sim \mathbf{b}'$ then $\langle \mathbf{b} \rangle = \langle \mathbf{b}' \rangle$. Moreover, for every $\mathbf{b} \in B^m$ we shall denote by $\mathbf{y}^{\mathbf{b}}$ the tuple $((b_1, c_1), \dots, (b_m, c_m)) \in Y$ obtained by choosing $c_i < c_j$ whenever $b_i = b_j$ and $i < j$. Notice that $\Pi(\mathbf{y}) = \mathbf{b}$ and $\mathbf{y}^{\mathbf{b}} \sim \langle \mathbf{b} \rangle$.

Now we shall show how to construct two structures \mathbf{X}_1 and \mathbf{X}_2 which satisfy condition (4). The domain of both \mathbf{X}_1 and \mathbf{X}_2 is X . The constraints of \mathbf{X}_1 and \mathbf{X}_2 are constructed as follows. Let J be the set of all triplets (R, T, \mathbf{a}) where $R \in \sigma$, $\mathbf{a} \in A^{\text{arity}(R)}$, and T is a multiset of size m of tuples in $R(\mathbf{B})$ such that for every $s, s' \in [\text{arity}(R)]$:

$$\mathbf{a}[s] = \mathbf{a}[s'] \Rightarrow \forall \mathbf{t} \in T(\mathbf{t}[s] = \mathbf{t}[s']).$$

In other words, T only contains tuples $\mathbf{t} \in R(\mathbf{B})$ such that $\mathbf{t} = f(\mathbf{a})$ for some function $f : \{\mathbf{a}\} \rightarrow B$. Then for every $(R, T, \mathbf{a}) \in J$ and $d \in [2]$, we define Q_j^d to be the set of $m!$ constraints obtained in the following way. Fix any arbitrary ordering $\mathbf{t}_1, \dots, \mathbf{t}_m$ of the tuples in T and let $\mathbf{b}_1, \dots, \mathbf{b}_r \in B^m$, $r = \text{arity}(R)$ such that $\mathbf{b}_s[i] = \mathbf{t}_i[s]$ for every $i \in [m]$ and $s \in [r]$. Then, for every permutation $\tau \in [m]$, we include constraint $R((a_1, \langle \mathbf{b}_1 \rangle) \circ \tau, \dots, (a_r, \langle \mathbf{b}_r \rangle) \circ \tau)$ in Q_j^1 and constraint $R((a_1, \mathbf{y}^{\mathbf{b}_1}) \circ \tau, \dots, (a_r, \mathbf{y}^{\mathbf{b}_r}) \circ \tau)$ in Q_j^2 , where $\mathbf{a} = (a_1, \dots, a_r)$.

Finally, we define $C_{\mathbf{X}_d}$, $d \in [2]$ to be $\cup_{j \in J} Q_j^d$ (note that for each $d \in [2]$, all sets Q_j^d , $j \in J$ are disjoint). To complete the proof it only remains to show that \mathbf{X}_1 and \mathbf{X}_2 satisfy condition (4).

▷ **Claim 6.** \mathbf{A} is homomorphic to \mathbf{X}_1 .

Proof. Let p be a feasible solution of $\text{SA}^1(\mathbf{A}, \mathbf{B})$. We define a mapping $h : A \rightarrow X$ by setting $h(a) = (a, \langle \mathbf{c}_a \rangle)$, where $\mathbf{c}_a \in B^m$ is any tuple satisfying that every element $b \in B$ occurs exactly $m \cdot p_a(b)$ times. We shall show that h is a homomorphism from \mathbf{A} to \mathbf{X}_1 . Let $R \in \sigma$, $\mathbf{a} = (a_1, \dots, a_r) \in R(\mathbf{A})$ where $r = \text{arity}(R)$, and consider the multiset $T = \{\mathbf{t}_1, \dots, \mathbf{t}_m\}$ of tuples in $R(\mathbf{B})$ obtained by picking each tuple $\mathbf{t}_i = f_i(\mathbf{a}) \in R(\mathbf{B})$ exactly $m \cdot p_{R(\mathbf{a})}(f_i)$ times (note that we are using implicitly (SA4) to guarantee the existence of T). For every $s, s' \in [r]$ satisfying $a_s = a_{s'}$ it follows from the construction of T that $\mathbf{t}[s] = \mathbf{t}[s']$ for every $\mathbf{t} \in T$, and hence $(R, T, \mathbf{a}) \in J$. Let $\mathbf{t}_1, \dots, \mathbf{t}_m$ be the ordering of the elements in T associated to (R, T, \mathbf{a}) in the construction of \mathbf{X}_1 and $\mathbf{b}_1, \dots, \mathbf{b}_r$ be the tuples obtained by setting $\mathbf{b}_s[i] = \mathbf{t}_i[s]$ for $i \in [m]$ and $s \in [r]$. Then, it follows that \mathbf{X}_1 contains the constraint $R((a_1, \langle \mathbf{b}_1 \rangle), \dots, (a_r, \langle \mathbf{b}_r \rangle))$. It follows again from (SA3) that for every $s \in [r]$ $\mathbf{c}_{a_s} \sim \mathbf{b}_s$ and hence $(a_s, \langle \mathbf{b}_s \rangle)$ is precisely $h(a_s)$ as desired. ◁

▷ **Claim 7.** \mathbf{X}_2 is homomorphic to \mathbf{B} .

Proof. Let $h : X \rightarrow B$ be the function mapping any (a, \mathbf{y}) to $\pi_1(\Pi(\mathbf{y}))$. It is immediate that h defines a homomorphism from \mathbf{X}_2 to \mathbf{B} . ◁

▷ **Claim 8.** \mathbf{X}_1 and \mathbf{X}_2 have a common equitable partition.

Proof. We shall prove that (P, Q^1) and (P, Q^2) define a common equitable partition of \mathbf{X}_1 and \mathbf{X}_2 where $P = \{P_i \mid i \in I\}$ is the partition given by the equivalence relation \sim on X and $Q^d = \{Q_j^d \mid j \in J\}$, $d \in [2]$ are as defined in the construction of \mathbf{X}_1 and \mathbf{X}_2 .

This follows immediately from the following fact. Let $d \in [2]$, $i \in I$, $j \in J$, and $\ell \in L$. Then exactly one of the following conditions holds:

1. There is no $x \in P_i$ and $C \in Q_j^d$ such that $M_{\mathbf{X}_d}[x, C] = \ell$;
2. There exists a one-to-one correspondence between the elements of P_i and Q_j^d such that for every pair (x, C) of associated elements, $M_{\mathbf{X}_d}[x, C] = \ell$.

Furthermore, for every $i \in I$, $j \in J$, and $\ell \in L$, condition (1) holds for $d = 1$ if and only if it holds for $d = 2$.

Let us prove it. Let $j = (R, T, \mathbf{a})$ and let $\mathbf{b}_1, \dots, \mathbf{b}_r$ and $\mathbf{y}^{\mathbf{b}_1}, \dots, \mathbf{y}^{\mathbf{b}_r}$ be as in the construction of Q_j^d (where $r = \text{arity}(R)$). We first observe that for every $x \in P_i$, every $C = R(x_1, \dots, x_r) \in Q_j^d$, and every permutation τ on $[m]$, $M_{\mathbf{X}_d}[x, C] = \ell \Leftrightarrow M_{\mathbf{X}_d}[x \circ \tau, C \circ \tau] = \ell$ where we use $C \circ \tau$ to denote $R(x_1 \circ \tau, \dots, x_r \circ \tau)$. It then follows that if (1) fails then (2) must hold.

Now it only remains to see that (1) holds for $d = 1$ if and only if (1) holds for $d = 2$. Clearly, this follows immediately if the relation symbol in ℓ is different from R , so we can assume that $\ell = (R, S)$ for some $S \subseteq [r]$. We then note that if there is some pair (x, C) violating (1), then $M_{\mathbf{X}_d}[x \circ \tau, C \circ \tau] = \ell$ for all permutations τ on $[m]$, and hence C can be chosen to be any constraint in Q_j^d . Hence, if we choose $R((a_1, \langle \mathbf{b}_1 \rangle), \dots, (a_r, \langle \mathbf{b}_r \rangle))$ for $d = 1$ and $R((a_1, \mathbf{y}^{\mathbf{b}_1}), \dots, (a_r, \mathbf{y}^{\mathbf{b}_r}))$ for $d = 2$, in order to complete the proof it is enough to show that for every $s, s' \in [r]$, $(a_s, \langle \mathbf{b}_s \rangle) = (a_{s'}, \langle \mathbf{b}_{s'} \rangle) \Leftrightarrow (a_s, \mathbf{y}^{\mathbf{b}_s}) = (a_{s'}, \mathbf{y}^{\mathbf{b}_{s'}})$.

The direction (\Leftarrow) is immediate. For the direction (\Rightarrow) assume that s, s' satisfy the left-hand side. Since $a_s = a_{s'}$ it follows that $\mathbf{b}_s = \mathbf{b}_{s'}$. Since $\mathbf{y}^{\mathbf{b}_s}$ and $\mathbf{y}^{\mathbf{b}_{s'}}$ are determined in a unique way from \mathbf{b}_s and $\mathbf{b}_{s'}$ respectively, we are done. \triangleleft

4 Some Applications

As an immediate consequence of Theorem 4 we obtain that feasibility of the k^{th} Sherali-Adams relaxation of the homomorphism problem is closed under \equiv_k .

► **Corollary 9.** *Let \mathbf{A} , \mathbf{A}' and \mathbf{B} be σ -structures and suppose that $\mathbf{A} \equiv_k \mathbf{B}$. Then, $\text{SA}^k(\mathbf{A}, \mathbf{B})$ is feasible if and only if $\text{SA}^k(\mathbf{A}', \mathbf{B})$ is feasible.*

We note that if the maximum arity on the signature σ is at most k , then the previous corollary can be alternatively stated in the following way: if \mathbf{A} and \mathbf{A}' are C^k -equivalent then $\text{SA}^k(\mathbf{A}, \mathbf{B})$ is feasible if and only if $\text{SA}^k(\mathbf{A}', \mathbf{B})$ is feasible. Then, one could be tempted to use this observation to transfer results from LP relaxations to logical definability. In particular one could infer that if SA^k decides correctly $\text{CSP}(\mathbf{B})$ then it is definable in the logic $C_{\infty, \omega}^k$ which is the extension of C^k consisting of all formulas made from atomic formulas and equality by means of finitary and infinitary conjunctions, negations, and standard and counting quantifiers. However, this is of limited interest as it follows by combining [2] and [6] that $\text{CSP}(\mathbf{B})$ would also be definable in the much weaker logic $L_{\infty, \omega}^{\max(k, 3)}$ where counting quantifiers are not allowed. Consequently, the previous corollary is most likely to find applications in obtaining lower bounds on the Sherali-Adams rank for concrete instances of CSP.

However, the principal novelty in our result is precisely the opposite direction, which leads to an alternative combinatorial characterization of the Sherali-Adams relaxation. A concrete application is the answer to the following question: for which structures \mathbf{B} is $\text{CSP}(\mathbf{B})$ closed under \equiv_1 -equivalence? This question arises in the context of the distributed CSP [9] where the variables and constraints of an instance are distributed among agents which communicate with each other by sending messages through fixed communication channels. In fact, the connection between the Weisfeiler-Leman algorithm and distributed computation goes back to the influential paper of Angluin on networks of processors [1]. For the distributed CSP, one of the most natural configurations for the communication network is essentially identical to the factor graph [16]. It then follows, under some technical requirements (agent anonymity and synchronicity) that any distributed message passing algorithm will necessarily behave in an identical manner on every two input instances that are \equiv_1 -equivalent and, hence, it follows that $\text{CSP}(\mathbf{B})$ can only be solved by a distributed algorithm if $\text{CSP}(\mathbf{B})$ is closed under \equiv_1 -equivalence.

This question was already answered in [9] where it was shown that $\text{CSP}(\mathbf{B})$ is closed under \equiv_1 if and only if \mathbf{B} has symmetric polymorphisms of all arities, where a k -ary symmetric polymorphism of a σ -structure \mathbf{B} is any homomorphism h from \mathbf{B}^k to \mathbf{B} that is invariant under the permutation of its arguments (the reader can safely ignore the definition of symmetric polymorphism as we will be using it as a black-box). The proof in [9] makes use of a result from [24] stating that $\text{CSP}(\mathbf{B})$ has symmetric polymorphisms of all arities if and only if it is solvable by an LP relaxation known as the basic linear programming relaxation (BLP). Although BLP is slightly different from SA^1 , both coincide over instances (\mathbf{A}, \mathbf{B}) where \mathbf{A} has no loops (i.e., every constraint has no repeated elements). It is then very easy to obtain the following characterization:

► **Lemma 10.** *Let \mathbf{B} be a fixed finite σ -structure. The following are equivalent:*

1. $\text{CSP}(\mathbf{B})$ is closed under \equiv_1 -equivalence;
2. SA^1 decides $\text{CSP}(\mathbf{B})$;
3. BLP decides $\text{CSP}(\mathbf{B})$;
4. \mathbf{B} has symmetric polymorphisms of all arities.

Proof (Sketch). (1) \Leftrightarrow (2) follows from Theorem 5 and (3) \Leftrightarrow (4) from [24]. Hence, it is only necessary to verify (2) \Leftrightarrow (3). We use the following fact which follows from the Sparse Incomparability Lemma [29]: for every instance \mathbf{A} of $\text{CSP}(\mathbf{B})$, there exists a structure \mathbf{A}' with no loops such that $\mathbf{A}' \rightarrow \mathbf{A}$ and $\mathbf{A} \rightarrow \mathbf{B}$ iff $\mathbf{A}' \rightarrow \mathbf{B}$. Now, assume that BLP does not solve $\text{CSP}(\mathbf{B})$. This means that there exists a structure \mathbf{A} not homomorphic to \mathbf{B} and such that $\text{BLP}(\mathbf{A}, \mathbf{B})$ is feasible. Now, let \mathbf{A}' be the structure given by the Sparse Incomparability Lemma. Since $\mathbf{A}' \rightarrow \mathbf{A}$ it follows that $\text{BLP}(\mathbf{A}', \mathbf{B})$ is feasible, and, since \mathbf{A}' has no loops, $\text{SA}^1(\mathbf{A}', \mathbf{B})$ is feasible as well. Since \mathbf{A}' is not homomorphic to \mathbf{B} it follows that SA^1 does not solve $\text{CSP}(\mathbf{B})$. The same argument can be used to show the converse although it is not necessary as it also follows immediately by comparing the inequalities of SA^1 and BLP. ◀

References

- 1 Dana Angluin. Local and global properties in networks of processors (extended abstract). In *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*, STOC '80, page 82–93, New York, NY, USA, 1980. Association for Computing Machinery. doi:10.1145/800141.804655.
- 2 Albert Atserias, Andrei A. Bulatov, and Anuj Dawar. Affine systems of equations and counting infinitary logic. *Theor. Comput. Sci.*, 410(18):1666–1683, 2009. doi:10.1016/j.tcs.2008.12.049.
- 3 Albert Atserias and Elitza Maneva. Sherali–adams relaxations and indistinguishability in counting logics. *SIAM Journal on Computing*, 42(1):112–137, 2013. doi:10.1137/120867834.
- 4 László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 684–697, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2897518.2897542.
- 5 László Babai, Paul Erdős, and Stanley M. Selkow. Random graph isomorphism. *SIAM J. Comput.*, 9(3):628–635, 1980. doi:10.1137/0209047.
- 6 Libor Barto. The collapse of the bounded width hierarchy. *Journal of Logic and Computation*, 26(3):923–943, 2016.
- 7 Joshua Brakensiek, Venkatesan Guruswami, Marcin Wrochna, and Stanislav Zivný. The power of the combined basic linear programming and affine relaxation for promise constraint satisfaction problems. *SIAM J. Comput.*, 49(6):1232–1248, 2020. doi:10.1137/20M1312745.

- 8 A. A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–330, October 2017. doi:10.1109/FOCS.2017.37.
- 9 Silvia Butti and Victor Dalmau. The complexity of the distributed constraint satisfaction problem. In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference)*, volume 187 of *LIPICs*, pages 20:1–20:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.STACS.2021.20.
- 10 Jin-yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identifications. *Comb.*, 12(4):389–410, 1992. doi:10.1007/BF01305232.
- 11 Siu On Chan, James R. Lee, Prasad Raghavendra, and David Steurer. Approximate constraint satisfaction requires large LP relaxations. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 350–359. IEEE Computer Society, 2013. doi:10.1109/FOCS.2013.45.
- 12 Víctor Dalmau and Andrei A. Krokhin. Robust satisfiability for csp: Hardness and algorithmic results. *ACM Trans. Comput. Theory*, 5(4):15:1–15:25, 2013. doi:10.1145/2540090.
- 13 Víctor Dalmau, Andrei A. Krokhin, and Rajsekar Manokaran. Towards a characterization of constant-factor approximable finite-valued CSPs. *J. Comput. Syst. Sci.*, 97:14–27, 2018. doi:10.1016/j.jcss.2018.03.003.
- 14 Holger Dell, Martin Grohe, and Gaurav Rattan. Lovász meets weisfeiler and leman. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPICs*, pages 40:1–40:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.ICALP.2018.40.
- 15 Tomás Feder and Moshe Y Vardi. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.
- 16 Ferdinando Fioretto, Enrico Pontelli, and William Yeoh. Distributed constraint optimization problems and applications: A survey. *J. Artif. Int. Res.*, 61(1):623–698, 2018.
- 17 Konstantinos Georgiou, Avner Magen, and Madhur Tulsiani. Optimal sherali-adams gaps from pairwise independence. In Irit Dinur, Klaus Jansen, Joseph Naor, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 12th International Workshop, APPROX 2009, and 13th International Workshop, RANDOM 2009, Berkeley, CA, USA, August 21-23, 2009. Proceedings*, volume 5687 of *Lecture Notes in Computer Science*, pages 125–139. Springer, 2009. doi:10.1007/978-3-642-03685-9_10.
- 18 Mrinalkanti Ghosh and Madhur Tulsiani. From weak to strong linear programming gaps for all constraint satisfaction problems. *Theory Comput.*, 14(1):1–33, 2018. doi:10.4086/toc.2018.v014a010.
- 19 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 – 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings*, volume 8737 of *Lecture Notes in Computer Science*, pages 505–516. Springer, 2014. doi:10.1007/978-3-662-44777-2_42.
- 20 Martin Grohe and Martin Otto. Pebble games and linear equations. *J. Symb. Log.*, 80(3):797–844, 2015. doi:10.1017/jsl.2015.28.
- 21 Pavol Hell and Jaroslav Nešetřil. *Graphs and Homomorphisms*. Oxford University Press, 2004.
- 22 Neil Immerman and Eric Lander. Describing graphs: A first-order approach to graph canonization. In *Complexity Theory Retrospective: In Honor of Juris Hartmanis on the Occasion of His Sixtieth Birthday*, pages 59–81, 1990.
- 23 Amit Kumar, Rajsekar Manokaran, Madhur Tulsiani, and Nisheeth K. Vishnoi. On lp-based approximability for strict csp. In Dana Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1560–1573. SIAM, 2011. doi:10.1137/1.9781611973082.121.

- 24 Gabor Kun, Ryan O'Donnell, Suguru Tamaki, Yuichi Yoshida, and Yuan Zhou. Linear programming, width-1 CSPs, and robust satisfaction. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, ITCS '12*, page 484–495, New York, NY, USA, 2012. Association for Computing Machinery. doi:10.1145/2090236.2090274.
- 25 Benoît Larose, Cynthia Loten, and Claude Tardif. A characterisation of first-order constraint satisfaction problems. *Log. Methods Comput. Sci.*, 3(4), 2007. doi:10.2168/LMCS-3(4:6)2007.
- 26 AA Leman and B Weisfeiler. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya*, 2(9):12–16, 1968.
- 27 László Lovász and Alexander Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*, 1(2):166–190, 1991. doi:10.1137/0801013.
- 28 Peter Malkin. Sherali–adams relaxations of graph isomorphism polytopes. *Discrete Optimization*, 12:73–97, 2014.
- 29 Jaroslav Nešetřil and Vojtěch Rödl. Chromatically optimal rigid graphs. *Journal of Combinatorial Theory, Series B*, 46(2):133–141, 1989. doi:10.1016/0095-8956(89)90039-7.
- 30 Motakuri V. Ramana, Edward R. Scheinerman, and Daniel Ullman. Fractional isomorphism of graphs. *Discrete Mathematics*, 132(1):247–265, 1994. doi:10.1016/0012-365X(94)90241-0.
- 31 Edward R Scheinerman and Daniel H Ullman. *Fractional graph theory: a rational approach to the theory of graphs*. Courier Corporation, 2011.
- 32 Hanif D. Sherali and Warren P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discret. Math.*, 3(3):411–430, 1990. doi:10.1137/0403036.
- 33 Johan Thapper and Stanislav Zivný. The power of sherali-adams relaxations for general-valued csps. *SIAM J. Comput.*, 46(4):1241–1279, 2017. doi:10.1137/16M1079245.
- 34 Gottfried Tinhofer. Graph isomorphism and theorems of birkhoff type. *Computing*, 36(4):285–300, 1986. doi:10.1007/BF02240204.
- 35 Gottfried Tinhofer. A note on compact graphs. *Discret. Appl. Math.*, 30(2-3):253–264, 1991. doi:10.1016/0166-218X(91)90049-3.
- 36 Yuichi Yoshida and Yuan Zhou. Approximation schemes via sherali-adams hierarchy for dense constraint satisfaction problems and assignment problems. In Moni Naor, editor, *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 423–438. ACM, 2014. doi:10.1145/2554797.2554836.
- 37 Dmitriy Zhuk. A proof of CSP dichotomy conjecture. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 331–342, October 2017. doi:10.1109/FOCS.2017.38.

A Applying the SA method to the Homomorphism problem

Here we shall show that the family of relaxations SA^k considered in the present paper is closely interleaved with the system of relaxations obtained by applying the SA method to a natural choice of initial polytope P .

Let \mathbf{A} and \mathbf{B} be σ -structures. We define polytope $P = P(\mathbf{A}, \mathbf{B})$ using a system of inequalities. The variables of the system are $x_{a,b}$ for each $a \in A$ and $b \in B$. Each variable must take a value in the range $[0, 1]$. We remark that by fixing some arbitrary ordering on the variables in $x_{a,b}$ we can represent any assignment on the variables $x_{a,b}$ with a tuple $\mathbf{x} \in \mathbb{R}^n$ with $n = |A| \cdot |B|$. Therefore we shall abuse notation and use $\mathbf{x}_{a,b}$ to refer to the value in \mathbf{x} corresponding to variable $x_{a,b}$.

The variables are subject to the following inequalities.

$$\sum_{b \in B} x_{a,b} = 1 \text{ for every } a \in A, \tag{1}$$

$$\sum_{\mathbf{a} \in \{\mathbf{a}\}} x_{\mathbf{a}, f(\mathbf{a})} \leq |\{\mathbf{a}\}| - 1 \quad \begin{array}{l} \text{for each } R \in \sigma, \mathbf{a} \in R(\mathbf{A}), \\ \text{and } f : \{\mathbf{a}\} \rightarrow B \text{ with } f(\mathbf{a}) \notin R(\mathbf{B}). \end{array} \quad (2)$$

Note that if h is a homomorphism from \mathbf{A} to \mathbf{B} then the assignment setting $x_{\mathbf{a}, h(\mathbf{a})} = 1$ for every $\mathbf{a} \in A$ and the rest of variables to zero is feasible.

Now let $P^k, k \geq 1$ be the sequence of polytopes obtained using the SA method. The next lemma shows that the sequence of relaxations defined by SA^k and P^k are interleaved.

► **Lemma 11.** *Let $k \geq 1$ and let r be the maximum arity of a relation in σ . Then*

1. *If $P^k \neq \emptyset$ and $r \leq k$ then SA^k is feasible.*
2. *If SA^{k+r-1} is feasible then $P^k \neq \emptyset$*

Proof. (1). Assume that $P^k \neq \emptyset$ and let \mathbf{y} be a feasible solution of P_L^k . We shall construct a feasible solution of SA^k . First, set every variable of the form $p_V(f)$ to y_K where $K = \{(a, f(a)) \mid a \in V\}$. We first observe that this assignment satisfies (SA1) and (SA2). Indeed, let $U \subseteq A$ with $|U| < k$, let $f : U \rightarrow B$, and let $I = \{(u, f(u)) \mid u \in U\}$. Then, multiplying the equality (1) with $a \in A \setminus U$ by $\prod_{i \in I} x_i$ and linearizing we obtain equality (SA2) for U, f , and $V = U \cup \{a\}$. In this way we can obtain all equalities in (SA2) for $|U| + 1 = |V|$. We note here that the rest of equalities in (SA2) along all equalities in (SA1) can be obtained as a linear combination.

Secondly, let us set the rest of variables. For every $(\mathbf{a}, R) \in \mathcal{C}_{\mathbf{A}}$ and $f : \{\mathbf{a}\} \rightarrow B$, set $p_{(\mathbf{a}, R)}(f)$ to be y_K where $K = \{(a, f(a)) \mid a \in \{\mathbf{a}\}\}$ (note that we are using implicitly the fact that $r \leq k$). Then, (SA3) follows directly from (SA2). Finally, it only remains to show that (SA4) is also satisfied. Indeed, for every $f(\mathbf{a}) \notin R(\mathbf{B})$ we obtain equality $p_{(\mathbf{a}, R)}(f) = 0$ if we multiply (2) by the term $\prod_{i \in K}$ and linearize. We want to note that, in fact, (1) also holds under the weaker assumption $r \leq k + 1$, but the proof is slightly more involved.

(2). Assume that SA^{k+r-1} is feasible. We construct a feasible solution \mathbf{y} of P_L^k as follows. For every $K \subseteq A \times B$ which satisfies $K = \{(a, f(a)) \mid a \in U\}$ for some $U \subseteq A$ with $|U| \leq k$ and $f : U \rightarrow B$, we set $y_K := p_U(f)$. Otherwise, we set y_K to zero.

Let us show that this assignment satisfies all inequalities in P_L^k . Let

$$\mathbf{c}^T \mathbf{y} \leq \mathbf{d} \quad (3)$$

be any inequality defining P_L^k . Since (3) is obtained by multiplying an inequality which contains at most r variables by a term of at most $k - 1$ variables, there exists a set $V \subseteq A$ with $|V| \leq r + k - 1$ such that for every variable y_K appearing in (3), V satisfies $K \subseteq V \times B$. Note that, by (SA1), variables $p_V(g), g : V \rightarrow B$ define a probability distribution. For every $g : V \rightarrow B$ in the support of this distribution, consider the assignment \mathbf{x}^g that sets $\mathbf{x}_{v,b}^g = 1$ if $v \in V$ and $b = g(v)$ and $x_{v,b} = 0$ otherwise.

Inequality (3) has been obtained by multiplying an inequality from (1) or (2) by a term and linearizing. We claim that in both cases, the inequality that has generated (3) is satisfied by \mathbf{x}^g . If the inequality generating (3) is $\sum_{b \in B} x_{a,b} = 1$ for some $a \in A$ this follows simply from the fact that $a \in V$. Assume now that (3) has been generated by inequality $\sum_{\mathbf{a} \in \{\mathbf{a}\}} x_{\mathbf{a}, f(\mathbf{a})} \leq |\{\mathbf{a}\}| - 1$. In this case note that $\{\mathbf{a}\} \subseteq V$ and then the claim follows from (SA3) and (SA4). This finalizes the proof of the claim.

Consequently, since \mathbf{x}^g is integral it follows that the assignment \mathbf{y}^g defined as $\mathbf{y}_K^g = \prod_{i \in K} \mathbf{x}_i^g$ satisfies (3). Finally, note that if we set $\alpha^g = p_V(g)$, then for every $K \subseteq V \times B$, \mathbf{y}_K is precisely given by the convex combination $\sum_g \alpha^g \mathbf{y}_K^g$. ◀

B Proof of Lemma 1

► **Lemma 1.** *Let \mathbf{A}, \mathbf{B} be σ -structures. Then $SA^k(\mathbf{A}, \mathbf{B})$ is feasible if and only if $SA^1(\mathbf{A}_k^*, \mathbf{B}_k^*)$ is feasible.*

Proof (Sketch). The proof is purely syntactical although it is convenient to massage first a bit the LP formulations $SA^k(\mathbf{A}, \mathbf{B})$ and $SA^1(\mathbf{A}_k^*, \mathbf{B}_k^*)$. We shall refer to the solutions of $SA^k(\mathbf{A}, \mathbf{B})$ and $SA^1(\mathbf{A}_k^*, \mathbf{B}_k^*)$ by appropriately indexed sets of variables p, q respectively.

- In $SA^k(\mathbf{A}, \mathbf{B})$, it follows from (SA4) that we can safely replace all variables $p_{R(\mathbf{a})}(f)$ with $f(\mathbf{a}) \notin R(\mathbf{B})$ by 0.
- In $SA^1(\mathbf{A}_k^*, \mathbf{B}_k^*)$ we are required a bit more of work. First, for each $j \leq k$ and each $\mathbf{a} \in A^j$, it follows from conditions (SA3) and (SA4) for $T_{j,S}$ ($S \subseteq [j]$) that for every $x \in B_k^*$, $q_{\mathbf{a}}(x)$ must take value 0 unless $x = f(\mathbf{a})$ for some function $f : \{\mathbf{a}\} \rightarrow B$. Hence, in a first stage we set $q_{\mathbf{a}}(x)$ to zero for each $j \leq k$, each $\mathbf{a} \in A^j$ and each x that is not a tuple of the form $f(\mathbf{a})$ for some function $f : \{\mathbf{a}\} \rightarrow B$.

Furthermore, it follows from condition (SA3) for $T_{j,i}$ that $q_{\mathbf{a}}(f(\mathbf{a})) = q_{\mathbf{a}'}(f(\mathbf{a}'))$ for every \mathbf{a}, \mathbf{a}' satisfying $\{\mathbf{a}\} = \{\mathbf{a}'\}$ and every $f : \{\mathbf{a}\} \rightarrow B$. Hence, in a second stage, for each $V \subseteq A$ with $|V| \leq k$ and every $f : V \rightarrow B$ we identify all variables $q_{\mathbf{a}}(f(\mathbf{a}))$ which satisfy $\{\mathbf{a}\} = V$.

Then, consider now the variables of the form $q_{R(\mathbf{a})}(x)$, $x \in B_k^*$. It follows from conditions (SA3) and (SA4) for R_S ($S \subseteq [\text{arity}(R)]$) that $q_{R(\mathbf{a})}(x)$ must be set to 0 unless $x = R(f(\mathbf{a}))$ for some function $f : \{\mathbf{a}\} \rightarrow B$.

The other variables in $SA^1(\mathbf{A}_k^*, \mathbf{B}_k^*)$ are of the form $q_C(f)$ where $C \in \mathcal{C}_{\mathbf{A}_k^*}$. As we shall see they can always safely be identified with some of the other variables. Let us start first with the case in which C is a unary constraint. If $C = T_{j,S}(\mathbf{a})$ or $C = R_S(\mathbf{a})$, then it follows from (SA3) that $q_C(f) = q_{\mathbf{a}}(f(\mathbf{a}))$. Assume now that C is a binary constraint, that is $C = T_{j,i}(\mathbf{a}, \pi_i \mathbf{a})$ or $C = R_i(\mathbf{a}, \pi_i \mathbf{a})$. It follows again from (SA3) that $q_C(f) = q_{\mathbf{a}}(f(\mathbf{a}))$

Now we are ready to prove the lemma. In particular, consider the following one-to-one correspondence between the assignments in $SA^k(\mathbf{A}, \mathbf{B})$ and $SA^1(\mathbf{A}_k^*, \mathbf{B}_k^*)$:

- Every variable $p_V(f)$ in $SA^k(\mathbf{A}, \mathbf{B})$ is assigned as variable $q_{\mathbf{a}}(f(\mathbf{a}))$ in $SA^1(\mathbf{A}_k^*, \mathbf{B}_k^*)$ where \mathbf{a} is any tuple satisfying $\{\mathbf{a}\} = V$.
- Every variable $p_{R(\mathbf{a})}(f)$ in $SA^k(\mathbf{A}, \mathbf{B})$ is assigned as variable $q_{R(\mathbf{a})}(R(f(\mathbf{a})))$ in $SA^1(\mathbf{A}_k^*, \mathbf{B}_k^*)$.

It is not difficult to see that this correspondence preserves feasibility. ◀

C Proof of Lemma 3

► **Lemma 3.** *Let r be the maximum arity among all relations in σ and assume that $r \leq k$. Then for every pair of structures \mathbf{A}, \mathbf{B} the following are equivalent:*

1. $\mathbf{A} \equiv_k \mathbf{B}$
2. $\text{Hom}(\mathbf{Q}; \mathbf{A}) = \text{Hom}(\mathbf{Q}; \mathbf{B})$ for every structure \mathbf{Q} of treewidth $< k$.

From Theorem 2 it follows that for a pair of σ -structures \mathbf{A} and \mathbf{B} , $\mathbf{A}_k^* \equiv_k \mathbf{B}_k^*$ if and only if $\text{Hom}(\mathbf{T}, \mathbf{A}_k^*) = \text{Hom}(\mathbf{T}, \mathbf{B}_k^*)$ for every σ_k^* -ftree \mathbf{T} . So it only remains to prove the following.

▷ **Claim 12.** Assume that $r \leq k$. Then the following are equivalent:

1. $\text{Hom}(\mathbf{Q}, \mathbf{A}) = \text{Hom}(\mathbf{Q}, \mathbf{B})$ for every σ -structure \mathbf{Q} of treewidth $< k$;
2. $\text{Hom}(\mathbf{T}, \mathbf{A}_k^*) = \text{Hom}(\mathbf{T}, \mathbf{B}_k^*)$ for every σ_k^* -ftree \mathbf{T} .

Proof. (1) \Rightarrow (2). Let \mathbf{T} be a σ^* -ftree. It follows immediately that if (1) holds then both \mathbf{A} and \mathbf{B} must have the same number of elements and constraints. It then follows that (2) holds for \mathbf{T} if it consists of a single element and no constraints at all. Consequently we can safely assume that all elements in \mathbf{T} participate in at least one constraint.

In what follows $\mathbf{D}_k^* \in \{\mathbf{A}_k^*, \mathbf{B}_k^*\}$. Let t be any node in T . Since t participates in a constraint it follows that the possible image of t in homomorphism from \mathbf{T} is heavily restricted. In particular, if the image of t according to some homomorphism from \mathbf{T} to \mathbf{D}_k^* is in D^j for some $j \leq k$ then necessarily the image of t in any homomorphism from \mathbf{T} to any structure $\mathbf{C}_k^* \in \{\mathbf{A}_k^*, \mathbf{B}_k^*\}$ must be in C^j . This means that we can safely add constraint $T_{j,\emptyset}(t)$ to \mathbf{T} without altering $\text{Hom}(\mathbf{T}, \mathbf{A}_k^*)$ or $\text{Hom}(\mathbf{T}, \mathbf{B}_k^*)$.

Likewise, if some homomorphism from \mathbf{T} to a structure \mathbf{D}_k^* maps t to a constraint $R(\mathbf{d})$, then likewise we can assume that constraint $R_\emptyset(t)$ belongs to \mathbf{T} .

To complete the proof we shall show that it is always possible to construct from \mathbf{T} a σ -structure \mathbf{Q} of treewidth $< k$ such that $|\text{Hom}(\mathbf{T}, \mathbf{D}_k^*)| = |\text{Hom}(\mathbf{Q}, \mathbf{D})|$. It is convenient to construct \mathbf{Q} in two stages. First, let us construct a σ -structure \mathbf{P} (not necessarily of treewidth $< k$) satisfying that $|\text{Hom}(\mathbf{T}, \mathbf{D}_k^*)| = |\text{Hom}(\mathbf{P}, \mathbf{D})|$. We shall allow to use equalities in \mathbf{P} , i.e., constraints of the form $p_1 = p_2$, indicating that p_1 and p_2 must be assigned to the same element in D .

We shall define \mathbf{P} along with a function α mapping every element t of \mathbf{T} to a j -ary tuple of elements in \mathbf{P} ($j \leq k$) inductively on the number of elements of \mathbf{T} as follows.

Assume (base case) that \mathbf{T} contains a unique element t . As discussed above we can assume that \mathbf{T} contains constraint $T_{j,\emptyset}(t)$ for some $j \leq k$ or $R_\emptyset(t)$ for some $R \in \sigma$. In the first case, we set the universe of \mathbf{P} to contain j new elements p_1, \dots, p_j . Furthermore, for every unary constraint $T_{j,S}(t)$ in \mathbf{T} and every $i, i' \in S$, we include in \mathbf{P} the equality $p_i = p_{i'}$, and we define $\alpha(t) = (p_1, \dots, p_j)$. In the second case, we set the universe of \mathbf{P} to contain $\text{arity}(R)$ new elements $p_1, \dots, p_{\text{arity}(R)}$ and we include in \mathbf{P} the constraint $R(p_1, \dots, p_{\text{arity}(R)})$. Similarly to the previous case, for every unary constraint $R_S(t)$ in \mathbf{T} and every $i, i' \in S$, we include in \mathbf{P} the equality $p_i = p_{i'}$. Finally, we set $\alpha(t) = (p_1, \dots, p_{\text{arity}(R)})$.

Let us consider now the inductive case. Let t_1 and t_2 be nodes that participate in a binary constraint $U(t_1, t_2)$ (recall that U is either $T_{j,\mathbf{i}}$ or $R_{\mathbf{i}}$) in \mathbf{T} . By removing this constraint \mathbf{T} gets divided in two ftrees \mathbf{T}_1 and \mathbf{T}_2 such that \mathbf{T}_1 contains t_1 and \mathbf{T}_2 contains t_2 . Now, assume that \mathbf{P}_i and α_i are already constructed for \mathbf{T}_i , $i = 1, 2$. We are ready to define \mathbf{P} . First, we compute the disjoint union of \mathbf{P}_1 and \mathbf{P}_2 . Then, we add some further equalities depending on constraint $U(t_1, t_2)$. Consider first the case that $U = T_{j_1,\mathbf{i}}$, $\mathbf{i} = (i_1, \dots, i_{j_2}) \in [j_1]^{j_2}$ for some $j_1, j_2 \leq k$ and let $\alpha_i(t_i) = (p_{i_1}^{i_1}, \dots, p_{i_{j_2}}^{i_{j_2}})$, $i = 1, 2$. Then, for every $\ell \leq j_2$ we add the equality $p_\ell^2 = p_\ell^1$. Finally, for every $t \in T$ we define $\alpha(t)$ to be $\alpha_i(t)$ where \mathbf{T}_i contains t . The procedure is identical for $U = R_{\mathbf{i}}$, where we just substitute j_1 by $\text{arity}(R)$. It follows immediately from the definition that $\text{Hom}(\mathbf{P}, \mathbf{D}) = \text{Hom}(\mathbf{T}, \mathbf{D}_k^*)$.

Finally, let us define \mathbf{Q} to be the structure obtained by identifying (i.e., merging into a single element) all elements in \mathbf{P} joined by a chain of equalities. It is immediate that $\text{Hom}(\mathbf{P}, \mathbf{D}) = \text{Hom}(\mathbf{Q}, \mathbf{D})$.

We shall conclude by giving a tree-decomposition (G, β) of \mathbf{Q} of width $< k$. In particular, let G be the tree where the vertex set is precisely the universe of \mathbf{T} and two different nodes are adjacent if both participate in some common constraint in \mathbf{T} and let $\beta(t) = \{\alpha(t)\}$.

(2) \Rightarrow (1). Let \mathbf{Q} be a σ -structure of treewidth $< k$ and let $\mathbf{D} \in \{\mathbf{A}, \mathbf{B}\}$. We note here that we can assume that \mathbf{Q} is connected since if \mathbf{Q} is the disjoint union of structures \mathbf{Q}_1 and \mathbf{Q}_2 then $\text{Hom}(\mathbf{Q}, \mathbf{D}) = \text{Hom}(\mathbf{Q}_1, \mathbf{D}) \cdot \text{Hom}(\mathbf{Q}_2, \mathbf{D})$. We shall show that there exists a σ_k^* -ftree \mathbf{T} such that $\text{Hom}(\mathbf{T}, \mathbf{D}_k^*) = \text{Hom}(\mathbf{Q}, \mathbf{D})$. Let (G, β) be a tree-decomposition of

width at most k of \mathbf{Q} . It is well-known and easy to prove that, since \mathbf{Q} is connected we can always construct (G, β) in such a way that for every pair u, v of adjacent nodes in G , $\beta(u) \subseteq \beta(v)$ or $\beta(v) \subseteq \beta(u)$. Furthermore, it is easy to see that we can further enforce that for every constraint $R(\mathbf{q})$ in \mathbf{Q} there exists a node $v \in G$ such that $\beta(v) = \{\mathbf{q}\}$.

The universe T of \mathbf{T} is $V \cup \mathcal{C}_{\mathbf{Q}}$ where V is the node-set of G . Furthermore \mathbf{T} contains the following constraints.

Let us start with the unary constraints. Let t be an element in \mathbf{T} . If $t = v \in G$ then we include in \mathbf{T} a constraint $T_{j_v, \emptyset}(t)$ where $j_v = |\beta(v)|$. Otherwise, if $t = R(\mathbf{q}) \in \mathcal{C}_{\mathbf{Q}}$ then we include in \mathbf{T} all constraints $R_{\{i, i'\}}(t)$ where $\mathbf{q}[i] = \mathbf{q}[i']$.

Now, let us turn our attention to the binary constraints. Fix some arbitrary ordering on Q and for every $v \in V$ let $\mathbf{q}^v = (q_1^v, \dots, q_{j_v}^v)$ be an array containing the nodes in $\beta(v)$ following this fixed order.

Then, for every edge (u, v) in G include constraint $T_{j_u, \mathbf{i}}(u, v)$ where $\mathbf{i} = (i_1, \dots, i_{j_v})$ is defined as follows. First, we assume without loss of generality that $\beta(v) \subseteq \beta(u)$. Then, for every $\ell \leq j_v$, i_ℓ is defined to be such that $\mathbf{q}^u[i_\ell] = \mathbf{q}^v[\ell]$.

Finally, for every constraint $t = R(\mathbf{q})$ in \mathbf{Q} we pick some element $v \in V$ satisfying $\{\mathbf{q}\} = \beta(v)$ and we add the constraint $R_{\mathbf{i}}(t, v)$ with $\mathbf{i} = (i_1, \dots, i_{j_v})$ where i_ℓ satisfies $\mathbf{q}[i_\ell] = \mathbf{q}^v[\ell]$. It is immediate to see that \mathbf{T} is an ftree and that $\text{Hom}(\mathbf{Q}, \mathbf{D}) = \text{Hom}(\mathbf{T}, \mathbf{D}_k^*) \triangleleft$