Electronic Colloquium on Computational Complexity, Report No. 77 (2011)

Graph Isomorphism, Sherali-Adams Relaxations and Expressibility in Counting Logics

Albert Atserias
Universitat Politècnica de Catalunya
Barcelona, Spain

Elitza Maneva
Universitat de Barcelona
Barcelona, Spain

May 6, 2011

Abstract

Two graphs with adjacency matrices $\mathbf{A}$ and $\mathbf{B}$ are isomorphic if there exists a permutation matrix $\mathbf{P}$ for which the identity $\mathbf{P}^{\mathrm{T}}\mathbf{A}\mathbf{P} = \mathbf{B}$ holds. Multiplying through by $\mathbf{P}$ and relaxing the permutation matrix to a doubly stochastic matrix leads to the notion of fractional isomorphism. We show that the levels of the Sherali-Adams hierarchy of linear programming relaxations applied to fractional isomorphism interleave in power with the levels of a well-known color-refinement heuristic for graph isomorphism called the Weisfeiler-Lehman algorithm. This tight connection has quite striking consequences. For example, it follows immediately from a deep result of Grohe in the context of logics with counting quantifiers, that a fixed number of levels of SA suffice to determine isomorphism of planar graphs. We also offer applications both in finite model theory and polyhedral combinatorics. First, we show that certain properties of graphs such as that of having a flow-circulation of a prescribed value, are definable in the infinitary logic with counting with a bounded number of variables. Second, we exploit a lower bound construction due to Cai, Fürer and Immerman in the context of counting logics to give simple explicit instances that show that the SA relaxations of the vertex-cover and cut polytopes do not reach their integer hulls for up to $\Omega(n)$ levels, where n is the number of vertices in the graph.

1 Introduction

Let $\mathbf{A}$ and $\mathbf{B}$ be the adjacency matrices of two labeled graphs on $\{1, \ldots, n\}$. The fact that the two graphs are isomorphic is equivalent to the existence of a permutation matrix $\mathbf{P}$ for which the relation $\mathbf{P}^{\mathrm{T}}\mathbf{A}\mathbf{P} = \mathbf{B}$ holds. Multiplying both sides by $\mathbf{P}$ gives the equivalent condition $\mathbf{A}\mathbf{P} = \mathbf{P}\mathbf{B}$. At this point a linear programming relaxation suggests itself: relax the condition that $\mathbf{P}$ is a permutation matrix to a doubly stochastic matrix. How much coarser is this than actual isomorphism?

The concept of fractional isomorphism as defined in the preceeding paragraph falls within the framework of linear programming relaxations of combinatorial problems. Other types of relaxations of isomorphism include color-refinement methods such as the Weisfeiler-Lehman algorithm (WL-algorithm). In this algorithm the vertices of the graphs are classified according to their degree, then according to the multi-set of degrees of their neighbors, and so on until a fixed-point is achieved. If the two graphs get partitions with different parameters, the graphs are definitely not isomorphic. As it turns out, fractional isomorphism and color-refinement yield one and the same relaxation: it was shown by Ramana, Scheinerman and Ullman [34] that two graphs are fractionally isomorphic if and only if they are not distinguished by the color-refinement algorithm.

ISSN 1433-8092

Despite their simplicity, color-refinement methods are known to behave very well in practice and are in fact one of the most commonly used heuristics for isomorphism testing. A sample result adding support to this claim is a classical result of Babai, Erdös and Selkow [5] showing that the color-refinement algorithm will end-up distinguishing every pair of vertices of a randomly chosen graph with high probability. That said, one obvious limitation of the method is that it will fail badly on regular graphs, as in such a case the algorithm cannot even start. To remedy this, the WL-algorithm has been extended to refinement of colorings of k-tuples of vertices (the k-WL algorithm) for $k = 1, 2, 3, \ldots$, thus yielding a hierarchy of increasingly powerful relaxations of isomorphism. The power of the resulting algorithms has also been studied in depth. For example, Kucera [24] shows that the algorithm for $k = 2$ decides isomorphism almost surely on random regular graphs. Another example of quite different nature is the result of Grohe showing that there exists a fixed constant k for which the k-WL-algorithm is able to distinguish any pair of non-isomorphic planar graphs [15]. This was extended recently to a much more general and breakthrough result showing that the same is true for any non-trivial minor-closed class of graphs [17].

Hierarchies of relaxations such as the k-WL-algorithm could also be considered in the context of fractional isomorphism through linear programming. The theory of lift-and-project methods in the mathematical programming literature provides such a framework. These are methods by which an initial relaxation P of an integral polytope $P^{\mathbb{Z}}$ is tightened into sharper and sharper polytopes, thus forming a hierarchy of relaxations:

$$P = P^1 \supseteq P^2 \supseteq \cdots \supseteq P^{\mathbb{Z}}.$$

Examples of these include the hierarchy of linear programming relaxations proposed by Lovász and Schrijver [30], the one by Sherali and Adams [38], and their semi-definite programming versions, including Lasserre [25]. See [26] for a survey and comparison. These have been applied to study classical polytopes of combinatorial optimization such as the stable-set polytope, the cut polytope, and the matching polytope, among others [30, 26, 40, 31].

In this paper we show that for $k \geq 2$, the k-th level of the Sherali-Adams hierarchy relaxation of graph isomorphism is sandwiched between the $(k-1)$-tuple version of the WL-algorithm and its k-tuple version. What this means is that if two graphs are distinguishable by the $(k-1)$-WL-algorithm, then the k-th level of SA vanishes, and that if they are indistinguishable by the k-WL-algorithm, then the k-th level of SA remains non-empty. Thus, the k-WL-algorithm provides a combinatorial characterization of the power of this lift-and-project method applied to graph isomorphism. We call this sandwiching property the Transfer Lemma.

1.1 Consequences

The Transfer Lemma, in combination with the above-mentioned strong results about the power of the WL-algorithm, already has consequences for the graph isomorphism problem itself. For example, it follows directly from Grohe's results that there exists a fixed level of SA relaxations that becomes empty on any pair of non-isomorphic planar graphs. A very notable feature of this consequence is the fact that the proof of Grohe's result relies very heavily on the interpretation of the WL-algorithm in the context of logic languages with counting quantifiers, which do not seem to be even remotely related to linear programming relaxations.

Less immediate applications of the Transfer Lemma arise from the link it sets between two different areas: polyhedral combinatorics and finite model theory. We offer applications going in both directions.

First, we show that several properties of graphs are definable in a logic known in the finite model theory literature as infinitary logic with counting quantifiers and a bounded number of variables (denoted by $C^{\omega}_{\infty\omega}$). These properties include "having a matching of a given size in bipartite graphs" and "having an *st*-flow of a given value in networks with unit capacities". We note that the definability of the first follows also from a result by Blass, Gurevich and Shelah [8], but that the second strengthens it and is new; see the section on related work for more on this.

As a second application we export the powerful inexpressibility results due to Cai, Fürer, and Immerman [9] in the context of counting logics to get instances with fractional solutions in the context of SA relaxations. From the existence of two non-isomorphic graphs of bounded degree that remain indistinguishable by the k-WL-algorithm up to $k = \Omega(n)$, we get explicit instances of the max-cut and vertex-cover problems whose linear programming relaxations do not reach their integer hulls after $\Omega(n)$ levels of SA. Let us note that in both cases stronger results are known since Schoenebeck [36] proved that a non-trivial integrality gap for vertex-cover resists $\Omega(n)$ levels of the Lasserre hierarchy, and hence of the SA hierarchy, and similar techniques would apply to max-cut. At any rate, the point we are trying to make with this application is not to get the strongest possible results, but to illustrate the power that the connection established by the Transfer Lemma gives for exporting methods from one field into the other.

Both these applications of the Transfer Lemma make use of a general statement we prove about the preservation of solutions between *k-local* linear programs: if two graphs have a non-empty k-level SA polytope of fractional isomorphisms, our result implies that solutions to the linear program of one graph translate to solutions of the linear program of the other.

1.2 Related work

For the origins of fractional isomorphism see the references in the monograph [35]. The connection between fractional isomorphism and the color-refinement algorithm for vertices was made in [34]. The extension to the levels of the Sherali-Adams hierarchy and to the tuple-version of the WL algorithm and the logic with counting quantifiers is, to our knowledge, new.

The logic $C^{\omega}_{\infty\omega}$ is well-studied in finite model theory [11, 28]. The connection between indistinguishability in this logic and the tuple-version of the WL algorithm is from [20]. Despite the negative results from [9], the expressive power of this logic is still the object of study. Somewhat unexpectedly, it was shown in [8] that the property of having a perfect matching in bipartite graphs is expressible in the uniform version of $C^{\omega}_{\infty\omega}$ called IFP + C. Here we revisit matchings in bipartite graphs and consider the more general problems of *st*-cuts and *st*-flows in networks with unit capacities. Our results show that the existence of such objects with prescribed values are expressible in $C^{3}_{\infty\omega}$. Our techniques and those in [8] are completely different. The open problem from [8] about perfect matchings in general graphs stays open.

Lift-and-project methods for combinatorial optimization problems have been the object of intense study. An optimal integrality gap of 2 for vertex-cover was shown to resist $\Omega(\log n)$ levels of the Lovász-Schrijver hierarchy (LS) in [1]. This was later improved in [41, 37, 14] to more levels and to the semi-definite version LS+. For the Sherali-Adams hierarchy, it was shown in [10] that optimal gaps of 2 for vertex-cover and max-cut resist $n^{\Omega(1)}$ levels. For vertex-cover, a gap of 7/6 resists $\Omega(n)$ levels of Lasserre and hence of SA [36], and a gap of 1.36 resists $n^{\Omega(1)}$ levels of Lasserre [42]. For max-cut, we could not find any published lower-bound on the number of SA-levels that are needed to reach the integer hull but Schoenebeck informs us that his methods would yield a non-trivial gap for up to $\Omega(n)$ levels of Lasserre and hence SA. See also [27] for related results.

2 Preliminaries

In this section we define Sherali-Adams relaxations of 0-1 linear programs in their full generality. We also give a brief overview the basic definitions in counting logics, their corresponding pebble games, and the Weisfeiler-Lehman algorithm refered to in the introduction.

2.1 Sherali-Adams relaxations

Let $P \subseteq [0,1]^n$ be a polytope defined by a system of linear inequalities, say:

$$P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \geq \mathbf{b},\ \mathbf{0} \leq \mathbf{x} \leq \mathbf{1}\}$$

for a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, and a column vector $\mathbf{b} \in \mathbb{R}^m$. We write $P^{\mathbb{Z}}$ for the convex hull of the 0-1-vectors in P. In symbols, $P^{\mathbb{Z}} = \mathrm{conv}(P \cap \{0,1\}^n)$. The sequence of Sherali-Adams relaxations of $P^{\mathbb{Z}}$ is a sequence of polytopes $P^1 \supseteq P^2 \supseteq \cdots$ starting at $P^1 = P$ and each containing $P^{\mathbb{Z}}$. The k-th polytope P^k is defined in three steps.

In the first step, a system of polynomial inequalities of degree k is obtained by multiplying both sides of each defining inequality $\mathbf{a}^{\mathrm{T}}\mathbf{x} \geq b$ of P by all possible multipliers of the form $\prod_{i \in I} x_i \prod_{j \in J}(1 - x_j)$, where I and J are subsets of $[n]$ such that $|I \cap J| \leq k-1$ and $I \cap J = \emptyset$. This leaves a system of polynomial inequalities, each of degree at most k. In the second step the polynomial system is *linearized*, and hence relaxed. What this means is that each square x_i^2 is replaced by x_i, and each resulting monomial of the form $\prod_{j \in K} x_i$ is replaced by a new variable y_K. The result is a system of linear inequalities defining a polytope P_L^k in $\mathbb{R}^{n_k}$ for $n_k = \sum_{i=0}^{k} \binom{n}{i}$. In the third step, the polytope is projected back to n dimensions by defining

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

The polytope P^k is called the k-th level Sherali-Adams relaxation of $P^{\mathbb{Z}}$. It is not hard to see that $P^k \supseteq P^{\mathbb{Z}}$. Indeed, the integer hull of P is achieved not later than after n steps [38]:

$$P = P^1 \supseteq P^2 \supseteq \cdots \supseteq P^n = P^{\mathbb{Z}}.$$

Thus, the Sherali-Adams hierarchy provides a sequence of tighter and tighter relaxations of the integral polytope $P^{\mathbb{Z}}$. The smallest k for which $P^k = P^{\mathbb{Z}}$ is called the Sherali-Adams *rank* of the polytope P.

2.2 Counting quantifiers

A *counting quantifier* has the form $\exists^{\geq m} x\phi$, where m is a non-negative integer. The meaning is that "there exist at least m distinct x satisfying ϕ". Although a counting quantifier as in $\exists^{\geq m} x\phi$ can always be replaced by an equivalent formula such as $\exists x_1 \cdots \exists x_m (\bigwedge_{i \neq j} x_i \neq x_j \wedge \phi(x/x_i))$, doing so requires introducing m new variables. For an example, the formula $\forall x(\exists^{\geq d} y(E(x,y)) \wedge \neg\exists^{\geq d+1} y(E(x,y)))$ says of a graph that it is d-regular, and it does so using exactly two variables. It is not hard to see that without counting quantifiers the same property would require $d+1$ variables. Saving variables in a logical expression is analogous to saving space in computation. As in the example, variables can even be re-used provided they are re-used with disjoint scopes. Complexity issues of this sort have been explored in depth in finite model theory.

For the rest of the paper, let C^k denote the collection of all logical formulas with counting quantifiers using at most k different variables. For a class $\mathcal{C}$ of structures, we write $\mathcal{C}^{=n}$ for the subclass of structures whose universe has cardinality n. We say that a subclass $\mathcal{D} \subseteq \mathcal{C}$ is non-uniformly C^k-definable on $\mathcal{C}$ if there exists a sequence of C^k-formulas $\varphi_1, \varphi_2, \ldots$ such that $\mathcal{D}^{=n}$ is precisely the class of $\mathbf{A}$ in $\mathcal{C}^{=n}$ on which φ_n is true, for each $n \geq 1$. We call it uniformly C^k-definable if a single C^k-formula $\varphi = \varphi_n$ does the job. For instance, the example above shows that the class of d-regular graphs is uniformly C^2-definable on the class of graphs. It is not hard to see that non-uniform C^k-definability corresponds to uniform definability in the logic $C^k_{\infty\omega}$ extensively studied in finite model theory. For more background on C^k and $C^k_{\infty\omega}$ we refer the reader to [32, 9, 11].

2.3 Pebble games

An essential concept from logic is that of indistinguishability by the formulas of a logical language. We say that two structures $\mathbf{A}$ and $\mathbf{B}$ are C^k-indistinguishable if every C^k-formula that is true in $\mathbf{A}$ is also true in $\mathbf{B}$, and vice-versa. Clearly this defines an equivalence relation on the class of structrures that we write $\mathbf{A} \equiv^k_{\mathrm{C}} \mathbf{B}$. The indistinguishability relation in a logical language usually admits an alternative interpretation in terms of a two-player game. For first-order logic, these sort of games go back to Ehrenfeucht and Fraïssé [12, 13], and for logics with restricted number of variables to Barwise [7] and Immerman [18]. For the logic C^k we follow [9, 19], but see also [22].

The game for C^k goes as follows. Let $\mathbf{A}$ and $\mathbf{B}$ be two structures. The game is played by two players: Spoiler and Duplicator. The goal of Spoiler is to show a difference between $\mathbf{A}$ and $\mathbf{B}$. The goal of Duplicator is to hide such a difference for as long as possible. There are $2k$ pebbles matched in pairs, say by having k different shapes. In each round, Spoiler chooses a pair of pebbles to play, say the i-th pair. Then he chooses a structure, $\mathbf{A}$ or $\mathbf{B}$, and a subset X of the universe of that structure. In response, Duplicator must choose a subset Y of the universe of the other structure such that $|Y| = |X|$; if she cannot do even that, she loses immediately. To complete the round, Spoiler places one of the pebbles of the i-th pair over an element of Y of his choice, and in response Duplicator places the other pebble of the i-th pair over an element of X of her choice. At the end of the round the sets X and Y are forgotten, but the pebbles are retained on the board. The goal of Spoiler is to exhibit a discrepancy between $\mathbf{A}$ and $\mathbf{B}$ in the form of a correspondence between pebbled elements $a_i \mapsto b_i$ for $i \in [k]$, that is not a partial isomorphism between the substructure of $\mathbf{A}$ induced by $\{a_i : i \in [k]\}$ and the substructure of $\mathbf{B}$ induced by $\{b_i : i \in [k]\}$. If he succeeds, the game ends and Duplicator loses. We say that Duplicator has a winning strategy for the C^k-pebble game if she has a strategy to play forever.

When the concept of winning strategy is appropriately formalized (we do that in the next section), the claim is that Duplicator has a winning strategy for the C^k-pebble game played on $\mathbf{A}$ and $\mathbf{B}$ if and only if $\mathbf{A} \equiv^k_{\mathrm{C}} \mathbf{B}$. For a proof see [19].

2.4 Weisfeiler-Lehman algorithm

For this subsection, let $\mathbf{A}$ and $\mathbf{B}$ denote colored directed graphs with vertex sets A and B, edge relations $E^{\mathbf{A}}$ and $E^{\mathbf{B}}$, and color classes $C^{\mathbf{A}}_1, \ldots, C^{\mathbf{A}}_r$ and $C^{\mathbf{B}}_1, \ldots, C^{\mathbf{B}}_r$, respectively. Let $k \geq 1$ be an integer. One way to determine if $\mathbf{A} \equiv^{k+1}_{\mathrm{C}} \mathbf{B}$ is by running the k-tuple WL-algorithm on each structure, and checking if the resulting parameters match. Let us now give the details of the algorithm. This exposition follows [9].

The k-WL algorithm run on $\mathbf{A}$ starts with all k-tuples of elements of $\mathbf{A}$ classified into bags labeled by the isomorphism type that the tuples induce on $\mathbf{A}$, where the isomorphism type induced by a k-tuple $(a_1, \ldots, a_k)$ is the collection of all atomic formulas on the variables $x_1, \ldots, x_k$ that are satisfied by the assignment $x_i = a_i$.[1] At each iteration, the algorithm cycles through all possible k-tuples $(a_1, \ldots, a_k)$ and counts, for each isomorphism type of $(k+1)$-tuples T and each k-tuple of bags $(B_1, \ldots, B_k)$, the number of $a \in A$ for which the $(k+1)$-tuple $(a_1, \ldots, a_k, a)$ induces on $\mathbf{A}$ a substructure of isomorphism type T, and the k-tuple $(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_k)$ belongs to the bag B_i for every $i \in [k]$. Once these counts are over, it refines each bag of tuples into sub-bags labeled by the outcomes of these counts. When no further splitting is possible, the algorithm stops. To avoid the size of the labels to increase exponentially, after each iteration the bags are ordered in some standard way (lexicographically by their labels, say), and re-labeled by their position in this order. The parameters of the output are the counts that result at the final collection of bags. Note by the way that the splitting process must finish after no more than $|A|^k$ iterations since whenever a bag contains a single tuple it cannot split any further. When the k-WL algorithm is run on both $\mathbf{A}$ and $\mathbf{B}$, we say that the parameters match if the parameters of their outputs are the same. The claim is that for $k \geq 1$, it holds that $\mathbf{A} \equiv_{\mathrm{C}}^{k+1} \mathbf{B}$ if and only if the parameters match when the k-WL algorithm is run on $\mathbf{A}$ and $\mathbf{B}$. For a proof see [9, 33].

There is one subtle difference in our definition of the k-WL algorithm and the definition in [9] that is nonetheless relevant only if $k = 1$. The difference is that we introduce isomorphism types of $(k+1)$-tuples into the counts. In the case $k \geq 2$ it can be seen that these counts are redundant since the maximum arity of the relations in $\mathbf{A}$ is 2. The good news is that our definition unifies the algorithm and its proof of correctness for the cases $k = 1$ and $k > 1$. In contrast the definition in [9] required splitting into cases. Also the generality of working with isomorphism types is necessary to deal with directed graphs (in the case $k = 1$). Our definition of k-WL appeared first in [16].

3 Transfer Lemma

The statement of the Transfer Lemma relates two different notions of indistinguishability. One is defined through the k-pebble game and is denoted by $\equiv_{\mathrm{C}}^{k}$, and the other is the k-th level of Sherali-Adams of fractional isomorphism and is denoted by $\equiv_{\mathrm{SA}}^{k}$. These will be defined formally in the coming subsections. For more generality, and because we need it in the applications, we prove and state the lemma for colored directed graphs instead of plain graphs. A colored directed graph is a structure of the form $\mathbf{A} = (A, E^{\mathbf{A}}, C_1^{\mathbf{A}}, \ldots, C_r^{\mathbf{A}})$, where $E^{\mathbf{A}}$ is a binary relation on the set of vertices A, and each $C_i^{\mathbf{A}}$ is a subset of the vertices that represent the vertices colored $i \in \{1, \ldots, r\}$. The statement of our main result is the following:

Theorem 1. (Transfer Lemma) *Let $\mathbf{A}$ and $\mathbf{B}$ be colored directed graphs and $k \geq 1$ an integer. Then:*

$$\mathbf{A} \equiv_{\mathrm{SA}}^{k+1} \mathbf{B} \quad \Longrightarrow \quad \mathbf{A} \equiv_{\mathrm{C}}^{k+1} \mathbf{B} \quad \Longrightarrow \quad \mathbf{A} \equiv_{\mathrm{SA}}^{k} \mathbf{B}.$$

We do not know if either implication can be reversed. However, for $k = 1$, the second implication can be reversed as it is known that $\mathbf{A} \equiv_{\mathrm{SA}}^{1} \mathbf{B}$ is literally equivalent to $\mathbf{A} \equiv_{\mathrm{C}}^{2} \mathbf{B}$. Indeed, $\equiv_{\mathrm{SA}}^{1}$ is just plain fractional isomorphism as discussed in the introduction and $\equiv_{\mathrm{C}}^{2}$ is known to be equivalent to the vertex-refinement algorithm or the 1-WL-algorithm (see [20]). Thus, the equivalence between $\equiv_{\mathrm{SA}}^{1}$ and $\equiv_{\mathrm{C}}^{2}$ is exactly the result from [34], which was the starting point for our work.

[1] The atomic formulas are the formulas of the form $x_i = x_j$ or $E(x_i, x_j)$ or $C_c(x_i)$ for some $c \in [r]$.

3.1 Formal definition of the pebble game

Following the standard practice in finite model theory, we define winning strategies in the pebble game in terms of systems of partial isomorphisms that have the back and forth properties.

Let A and B be sets that do not contain the special element $\star$. Let $\mathbf{A} = (A, E^{\mathbf{A}}, C_1^{\mathbf{A}}, \ldots, C_r^{\mathbf{A}})$ and $\mathbf{B} = (B, E^{\mathbf{B}}, C_1^{\mathbf{B}}, \ldots, C_r^{\mathbf{B}})$ be colored directed graphs, i.e. $E^{\mathbf{A}} \subseteq A^2$ and $C_i^{\mathbf{A}} \subseteq A$ for $i \in [r]$. Let $(\mathbf{a}, \mathbf{b})$ be a pair of k-tuples, where $\mathbf{a} = (a_1, \ldots, a_k)$ with $a_i \in A \cup \{\star\}$ for every $i \in [k]$, and $\mathbf{b} = (b_1, \ldots, b_k)$ with $b_i \in B \cup \{\star\}$ for every $i \in [k]$. We say that $(\mathbf{a}, \mathbf{b})$ defines a partial k-isomorphism from A to B if the following conditions hold for every $i \in [k]$, every $j \in [k]$, and every $c \in [r]$:

1. $a_i = \star$ if and only if $b_i = \star$,
2. $a_i = a_j$ if and only if $b_i = b_j$,
3. $(a_i, a_j) \in E^{\mathbf{A}}$ if and only if $(b_i, b_j) \in E^{\mathbf{B}}$,
4. $a_i \in C_c^{\mathbf{A}}$ if and only if $b_i \in C_c^{\mathbf{B}}$.

For a k-tuple $\mathbf{a} = (a_1, \ldots, a_k)$, an index $i \in [k]$ and an element a, we write $\mathbf{a}[i/a]$ for the result of replacing the i-th component of $\mathbf{a}$ by a.

A *winning strategy for the Duplicator in the k-pebble game on* $\mathbf{A}$ *and* $\mathbf{B}$ is a non-empty $\mathcal{F} \subseteq (A \cup \{\star\})^k \times (B \cup \{\star\})^k$ such that every $(\mathbf{a}, \mathbf{b})$ in $\mathcal{F}$ defines a partial k-isomorphism from $\mathbf{A}$ to $\mathbf{B}$ and for every $i \in [k]$ the following properties are satisfied:

1. $(\mathbf{a}[i/\star], \mathbf{b}[i/\star])$ belongs to $\mathcal{F}$,
2. for every $X \subseteq A$ there exists $Y \subseteq B$ with $|Y| = |X|$ such that for every $b \in Y$ there exists $a \in X$ such that $(\mathbf{a}[i/a], \mathbf{b}[i/b])$ belongs to $\mathcal{F}$,
3. for every $Y \subseteq B$ there exists $X \subseteq A$ with $|X| = |Y|$ such that for every $a \in X$ there exists $b \in Y$ such that $(\mathbf{a}[i/a], \mathbf{b}[i/b])$ belongs to $\mathcal{F}$.

If there exists such a strategy, we write $\mathbf{A} \equiv_{\mathrm{C}}^{k} \mathbf{B}$.

3.2 Sherali-Adams levels of fractional isomorphism

For every $a, a' \in A$, let $A_{a,a'} = 1$ if (a, a') belongs to $E^{\mathbf{A}}$ and $A_{a,a'} = 0$ otherwise. Similarly, for every $b, b' \in B$, let $B_{b,b'} = 1$ if $(b, b') \in E^{\mathbf{B}}$ and $B_{b,b'} = 0$ otherwise. Thus, $(A_{a,a'})_{a,a' \in A}$ and $(B_{b,b'})_{b,b' \in B}$ are the adjacency matrices of $\mathbf{A}$ and $\mathbf{B}$, which we also denote by $\mathbf{A}$ and $\mathbf{B}$ whenever there is no risk that this could lead to confusion. For every $a \in A$ and $c \in [r]$, let $C_{a,c} = 1$ if $a \in C_c^{\mathbf{A}}$ and $C_{a,c} = 0$ otherwise. Similarly, for every $b \in B$, let $D_{b,c} = 1$ if $b \in C_c^{\mathbf{B}}$ and $D_{b,c} = 0$ otherwise. Thus, $(C_{a,c})_{a \in A, c \in [r]}$ and $(D_{b,c})_{b \in B, c \in [r]}$ are $|A| \times [r]$ and $|B| \times [r]$ matrices that encode the colors. We will write $\mathbf{C}$ and $\mathbf{D}$ for them.

For every pair $(a, b) \in A \times B$, let $X_{a,b}$ be a variable. Let $\mathbf{X}$ be the $|A| \times |B|$ matrix $(X_{a,b})_{a \in A, b \in B}$. The fractional relaxation of isomorphism is the following system of linear equalities and inequalities:

$$
\begin{array}{ll}
\mathbf{AX} = \mathbf{XB} & \mathbf{BX}^{\mathrm{T}} = \mathbf{X}^{\mathrm{T}}\mathbf{A} \\
\mathbf{C} = \mathbf{XD} & \mathbf{D} = \mathbf{X}^{\mathrm{T}}\mathbf{C} \\
\mathbf{Xe} = 1 & \mathbf{X}^{\mathrm{T}}\mathbf{e} = 1 \\
\multicolumn{2}{c}{\mathbf{X} \geq 0}
\end{array}
\tag{1}
$$

We write $F(\mathbf{A}, \mathbf{B})$ for this linear program.

It is worth noting that if $\mathbf{A}$ and $\mathbf{B}$ are graphs, then $\mathbf{A} = \mathbf{A}^{\mathrm{T}}$ and $\mathbf{B} = \mathbf{B}^{\mathrm{T}}$, and the equations $\mathbf{AX} = \mathbf{XB}$ and $\mathbf{BX}^{\mathrm{T}} = \mathbf{X}^{\mathrm{T}}\mathbf{A}$ in $F(\mathbf{A}, \mathbf{B})$ become equivalent. In the general case of directed colored graphs, the equation $\mathbf{BX}^{\mathrm{T}} = \mathbf{X}^{\mathrm{T}}\mathbf{A}$ is added for symmetry purposes.

For every integer $k \geq 0$, let R_k denote the collection of all subsets $p \subseteq A \times B$ with $|p| \leq k$. For $p \in R_k$ and $(a, b) \in A \times B$, we use the notation $p \cup ab$ as an abbreviation for $p \cup \{(a, b)\}$. For every $p \in R_k$, let X_p be a variable. If $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$, the k-th level of Sherali-Adams applied on $F(\mathbf{A}, \mathbf{B})$ is equivalent to the following system of linear equalities and inequalities:

$$A_{a,a_1} X_{q \cup a_1 b} + \cdots + A_{a,a_n} X_{q \cup a_n b} = X_{q \cup ab_1} B_{b_1,b} + \cdots + X_{q \cup ab_n} B_{b_n,b} \qquad \text{for } a \in A, b \in B, \tag{2}$$
$$B_{b,b_1} X_{q \cup ab_1} + \cdots + B_{b,b_n} X_{q \cup ab_n} = X_{q \cup a_1 b} A_{a_1,a} + \cdots + X_{q \cup a_n b} A_{a_n,a} \qquad \text{for } a \in A, b \in B, \tag{3}$$

together with

$$X_{q \cup ab_1} D_{b_1,c} + \cdots + X_{q \cup ab_n} D_{b_n,c} = X_q C_{a,c} \qquad \text{for } a \in A, c \in [r], \tag{4}$$
$$X_{q \cup a_1 b} C_{a_1,c} + \cdots + X_{q \cup a_n b} C_{a_n,c} = X_q D_{b,c} \qquad \text{for } b \in B, c \in [r], \tag{5}$$

and

$$X_{q \cup ab_1} + \cdots + X_{q \cup ab_n} = X_q \qquad \text{for } a \in A, \tag{6}$$
$$X_{q \cup a_1 b} + \cdots + X_{q \cup a_n b} = X_q \qquad \text{for } b \in B, \tag{7}$$
$$X_{q \cup ab} \geq 0 \qquad \text{for } a \in A, b \in B, \tag{8}$$
$$X_{\emptyset} = 1 \tag{9}$$

where in all places where it appears, q is an element of R_{k-1}. We obtained these inequalities by multiplying each equation in (1) by a term of the form $\prod_{ab \in I} X_{ab}$ for $I \subseteq A \times B$ with $|I| \leq k - 1$, and linearizing. Note that the factors of the form $\prod_{ab \in J}(1 - X_{ab})$ that are required in the official definition of Sherali-Adams relaxations, and appear to be missing here, are really implicit as they can be obtained as linear combinations of the ones given. This holds in this special case since all constraints are equalities instead of inequalities.

We write $F_k(\mathbf{A}, \mathbf{B})$ for this system. Note that setting $k = 1$ we recover $F(\mathbf{A}, \mathbf{B})$. If $F_k(\mathbf{A}, \mathbf{B})$ is satisfiable, we write $\mathbf{A} \equiv_{\mathrm{SA}}^{k} \mathbf{B}$.

4 Proof of the Transfer Lemma

Let $\mathbf{A}$ and $\mathbf{B}$ be colored directed graphs, and let $k \geq 1$ be a natural number. We will prove a longer chain of implications that entails the Transfer Lemma and involves two more notions of indistinguishability: $\equiv_{\mathrm{EP}}^{k}$ is an equivalence relation that extends the combinatorial notion known as "equitable partitions" (see [35]) to k-tuples, while $\equiv_{\mathrm{CS}}^{k}$ is defined by another pebble game that we refer to as the sliding game. The complete statement is the following:

$$\mathbf{A} \equiv_{\mathrm{SA}}^{k+1} \mathbf{B} \Longrightarrow \mathbf{A} \equiv_{\mathrm{C}}^{k+1} \mathbf{B} \Longrightarrow \mathbf{A} \equiv_{\mathrm{CS}}^{k} \mathbf{B} \Longrightarrow \mathbf{A} \equiv_{\mathrm{EP}}^{k} \mathbf{B} \Longrightarrow \mathbf{A} \equiv_{\mathrm{SA}}^{k} \mathbf{B}.$$

We prove the sequence of implications precisely in the order they appear. Before that, we need to define the two new notions of indistinguishability.

4.1 Formal definition of the sliding game

For a in $A \cup \{\star\}$, define $N^+(a)$ and $N^-(a)$ as follows:

1. $N^+(a) = \{a' \in A : (a, a') \in E^{\mathbf{A}}\}$, if $a \neq \star$,
2. $N^-(a) = \{a' \in A : (a', a) \in E^{\mathbf{A}}\}$, if $a \neq \star$,
3. $N^+(a) = N^-(a) = A$, if $a = \star$.

For b in $B \cup \{\star\}$, define $N^+(b)$ and $N^-(b)$ analogously.

A *winning strategy for the Duplicator in the k-pebble sliding game on* $\mathbf{A}$ *and* $\mathbf{B}$ is a non-empty $\mathcal{F} \subseteq (A \cup \{\star\})^k \times (B \cup \{\star\})^k$ such that every $(\mathbf{a}, \mathbf{b})$ in $\mathcal{F}$ defines a partial k-isomorphism from $\mathbf{A}$ to $\mathbf{B}$ and for every $i \in [k]$ and every $o \in \{+, -\}$, the following properties are satisfied:

1. $(\mathbf{a}[i/\star], \mathbf{b}[i/\star])$ belongs to $\mathcal{F}$,
2. for every $X \subseteq N^o(a_i)$ there exists $Y \subseteq N^o(b_i)$ with $|Y| = |X|$ such that for every $b \in Y$ there exists $a \in X$ such that $(\mathbf{a}[i/a], \mathbf{b}[i/b])$ belongs to $\mathcal{F}$,
3. for every $Y \subseteq N^o(b_i)$ there exists $X \subseteq N^o(a_i)$ with $|X| = |Y|$ such that for every $a \in X$ there exists $b \in Y$ such that $(\mathbf{a}[i/a], \mathbf{b}[i/b])$ belongs to $\mathcal{F}$.

If there exists such a strategy, we write $\mathbf{A} \equiv^k_{\mathrm{CS}} \mathbf{B}$.

4.2 Analogue of equitable partition for tuples

For an integer $k \geq 1$, we write S_k for the set of all permutations on $[k]$. For a permutation $\pi \in S_k$, we write $\mathbf{a} \circ \pi$ for the tuple $(a_{\pi(1)}, \ldots, a_{\pi(k)})$.

Let $\mathbf{a} = (a_1, \ldots, a_k)$ and $\mathbf{a}' = (a'_1, \ldots, a'_k)$ be tuples in $(A \cup \{\star\})^k$. For every $i \in [k]$ and $o \in \{+, -\}$, define:

$$d_i^o(\mathbf{a}, \mathbf{a}') = \begin{cases} 1 & \text{if } \mathbf{a} \neq \mathbf{a}' \text{ and there exists } a \in N^o(a_i) \cup \{\star\} \text{ such that } \mathbf{a}' = \mathbf{a}[i/a], \\ 0 & \text{otherwise.} \end{cases}$$

Note that $d_i^+(\mathbf{a}, \mathbf{a}') = d_i^-(\mathbf{a}', \mathbf{a})$. Let S and T be subsets of $(A \cup \{\star\})^k$. For every $i \in [k]$ and $o \in \{+, -\}$, define:

$$d_i^o(S, T) = \sum_{\mathbf{a} \in S} \sum_{\mathbf{a}' \in T} d_i^o(\mathbf{a}, \mathbf{a}').$$

Note that $d_i^+(S, T) = d_i^-(T, S)$. If S is a singleton $\{\mathbf{a}\}$, we write $d_i^o(\mathbf{a}, T)$ instead of $d_i^o(\{\mathbf{a}\}, T)$. We call $d_i^+(\mathbf{a}, T)$ the *out-degree of* $\mathbf{a}$ *in* T *on its i-th component*, and $d_i^-(\mathbf{a}, T)$ the *in-degree of* $\mathbf{a}$ *in* T *on its i-th component.*

Let $(P_1, \ldots, P_s)$ be a partition of $(A \cup \{\star\})^k$ into non-empty parts. For every $\mathbf{a} \in (A \cup \{\star\})^k$, let $c(\mathbf{a})$ be the unique $m \in [s]$ such that $\mathbf{a}$ belongs to P_m. The partition $(P_1, \ldots, P_s)$ is called a k-equitable partition of $\mathbf{A}$ if for every $m \in [s]$ and every $\mathbf{a}, \mathbf{a}' \in P_m$, the following conditions hold:

1. $(\mathbf{a}, \mathbf{a}')$ defines a partial k-isomorphism from $\mathbf{A}$ to $\mathbf{A}$,
2. $c(\mathbf{a}[i/\star]) = c(\mathbf{a}'[i/\star])$ for every $i \in [k]$,
3. $d_i^o(\mathbf{a}, P_n) = d_i^o(\mathbf{a}', P_n)$ for every $i \in [k]$, $o \in \{+, -\}$, and $n \in [s]$,
4. $|P_{c(\mathbf{a})}| = |P_{c(\mathbf{a} \circ \pi)}|$ for every permutation $\pi \in S_k$,

5. $c(\mathbf{a} \circ \pi) = c(\mathbf{a}' \circ \pi)$ for every permutation $\pi \in S_k$.

By 3., we note that the following identity holds for every $m, n \in [s]$, $a \in P_m$, $a' \in P_n$, and $i \in [k]$:

$$|P_m| d_i^+(\mathbf{a}, P_n) = d_i^+(P_m, P_n) = d_i^-(P_n, P_m) = |P_n| d_i^-(\mathbf{a}', P_m) \tag{10}$$

We say that $\mathbf{A}$ and $\mathbf{B}$ have a common k-equitable partition if there exist a k-equitable partition $(P_1, \ldots, P_s)$ of $\mathbf{A}$ and a k-equitable partition $(Q_1, \ldots, Q_t)$ of $\mathbf{B}$ such that the following conditions are satisfied:

1. $s = t$ and $|P_m| = |Q_m|$ for every $m \in [s]$,

and, for every $m \in [s]$, $\mathbf{a} \in P_m$ and $\mathbf{b} \in Q_m$:

2. $(\mathbf{a}, \mathbf{b})$ defines a partial k-isomorphism from $\mathbf{A}$ to $\mathbf{B}$,
3. $c(\mathbf{a}[i/\star]) = c(\mathbf{b}[i/\star])$ for every $i \in [k]$,
4. $d_i^o(\mathbf{a}, P_n) = d_i^o(\mathbf{b}, Q_n)$ for every $i \in [k]$, $o \in \{+, -\}$, and every $n \in [s]$,
5. $c(\mathbf{a} \circ \pi) = c(\mathbf{b} \circ \pi)$ for every permutation $\pi \in S_k$.

If there exists a common k-equitable partition we write $\mathbf{A} \equiv^k_{\mathrm{EP}} \mathbf{B}$.

4.3 From Sherali-Adams to pebble game

We show the first implication in the Transfer Lemma:

Lemma 1. *Let $k \geq 2$. If $\mathbf{A} \equiv^k_{\mathrm{SA}} \mathbf{B}$, then $\mathbf{A} \equiv^k_{\mathrm{C}} \mathbf{B}$.*

Proof. Let $(X_p)_{p \in R_k}$ be a feasible solution for $F_k(\mathbf{A}, \mathbf{B})$. Let $\mathcal{F}$ be the collection of all pairs of k-tuples $(\mathbf{a}, \mathbf{b}) \in (A \cup \{\star\})^k \times (B \cup \{\star\})^k$ for which the following two conditions are satisfied:

1. $a_i = \star$ if and only if $b_i = \star$, for every $i \in [k]$,
2. $p = \{(a_i, b_i) : i \in [k],\ a_i \neq \star,\ b_i \neq \star\}$ satisfies $X_p \neq 0$.

Note that $\mathcal{F}$ is non-empty as the pair of k-tuples $(\star^k, \star^k)$ satisfies the two conditions since in this case $p = \emptyset$ and $X_\emptyset \neq 0$ by equation (9). We proceed to show that each $(\mathbf{a}, \mathbf{b})$ in $\mathcal{F}$ defines a partial k-isomorphism from $\mathbf{A}$ to $\mathbf{B}$ and that the subtuple and back-and-forth properties are satisfied. We start with the subtuple property:

Claim 1. *Let $p, q \in R_k$. If $q \subseteq p$, then $X_p \leq X_q$.*

Proof. Assume $q \subseteq p$. We proceed by induction on the cardinality of the difference $|p - q|$. If $|p - q| = 0$, then $p = q$ and we are done. Assume $|p - q| > 0$. Let $(a, b) \in p - q$ and define $p' = p - \{(a, b)\}$. Then $q \subseteq p'$ and $|p' - q| < |p - q|$. By equation (6) with $p' \in R_{k-1}$ we have

$$X_{p'} = \sum_{b' \in B} X_{p' \cup ab'}.$$

Since each term in the sum is non-negative by equation (8) we get $X_{p' \cup ab} \leq X_{p'}$. Since $p' \cup ab = p$, the inequality $X_p \leq X_q$ follows from the induction hypothesis $X_{p'} \leq X_q$. ∎

Before we continue we need a definition. Let $p \in R_k$, where $p = \{(a_1, b_1), \dots, (a_s, b_s)\}$ with $s \leq k$ and $(a_h, b_h) \neq (a_\ell, b_\ell)$ for every $h, \ell \in [s]$, $h \neq \ell$. We say that p is a partial k-isomorphism from $\mathbf{A}$ to $\mathbf{B}$ if the following conditions are satisfied for every $h, \ell \in [s]$ and $c \in [r]$:

1. if $a_h = a_\ell$ then $b_h = b_\ell$ (and hence $h = \ell$),
2. if $b_h = b_\ell$ then $a_h = a_\ell$ (and hence $h = \ell$),
3. if $A_{a_h,a_\ell} = 1$ then $B_{b_h,b_\ell} = 1$,
4. if $B_{b_h,b_\ell} = 1$ then $A_{a_h,a_\ell} = 1$,
5. if $C_{a_h,c} = 1$ then $D_{b_h,c} = 1$,
6. if $D_{b_h,c} = 1$ then $C_{a_h,c} = 1$.

With this definition we are ready to state the second property of the solutions to $F_k(\mathbf{A}, \mathbf{B})$:

Claim 2. *Let $p \in R_k$. If $X_p \neq 0$, then p is a partial k-isomorphism from $\mathbf{A}$ to $\mathbf{B}$.*

Proof. Assume $X_p \neq 0$. Let $p = \{(a_1, b_1), \dots, (a_s, b_s)\}$ with $s \leq k$ and $(a_h, b_h) \neq (a_\ell, b_\ell)$ for every $h, \ell \in [s]$, $h \neq \ell$. We need to check all six conditions in the definition of partial k-isomorphism above.

For 1, assume for contradiction that $a_h = a_\ell$ and $b_h \neq b_\ell$. Let $q = p - \{(a_\ell, b_\ell)\}$ and note that $q \in R_{k-1}$. From equation (6) for this q and $a = a_h$ we get

$$X_{q \cup a_h b_h} = X_q - \sum_{\substack{b \in B \\ b \neq b_h}}^{n} X_{q \cup a_h b}.$$

Since (a_h, b_h) belongs to q we have $q \cup a_h b_h = q$ and therefore

$$\sum_{\substack{b \in B \\ b \neq b_h}}^{n} X_{q \cup a_h b} = 0.$$

Each term in the sum is non-negative by equation (8), hence each is 0. In particular, either $h = \ell$ and then we are done, or $X_{q \cup a_h b_\ell} = 0$. But $a_h = a_\ell$ and $q \cup a_\ell b_\ell = p$, hence $X_p = 0$; a contradiction.

For 2 argue as in 1 using equation (7) for $q = p - \{(a_h, b_h)\}$ and $b = b_\ell$.

For 3, assume for contradiction that $A_{a_h,a_\ell} = 1$ and $B_{b_h,b_\ell} = 0$. Let $q = \{(a_h, b_h)\}$. Note that $q \in R_1 \subseteq R_{k-1}$ since $k \geq 2$. From equation (2) for this q, $a = a_h$ and $b = b_\ell$ we get

$$X_{q \cup a_\ell b_\ell} = \sum_{b \in B} X_{q \cup a_h b} B_{b,b_\ell} - \sum_{\substack{a \in A \\ a \neq a_\ell}} A_{a_h,a} X_{q \cup a b_\ell}. \tag{11}$$

Since (a_h, b_h) belongs to q, by part 1 of this lemma we have $X_{q \cup a_h b} = 0$ whenever $b \neq b_h$. Moreover, whenever $b = j_h$ we have $B_{b,b_\ell} = 0$ by assumption. Both things together mean that the first sum in equation (11) vanishes. Since every term in the second sum in that same equation is non-negative by equation (8), we get $X_{q \cup a_\ell b_\ell} \leq 0$. Since $q \cup a_\ell b_\ell \subseteq p$, by Claim 1 we get $X_p \leq 0$. But also $X_p \geq 0$ by (8), so $X_p = 0$; a contradiction.

For 4 argue as in 3 using part 2 of this lemma.

For 5, assume for contradiction that $C_{a_h,c} = 1$ and $D_{b_h,c} = 0$. Let $q = p - \{(a_h, b_h)\}$. Note that $q \in R_{k-1}$. From equation (4) for this q and $a = a_h$ we get

$$X_q C_{a_h,c} = \sum_{b \in B} X_{q \cup a_h b} D_{b,c} \leq X_{q \cup a_h b_h} D_{b_h,c}. \tag{12}$$

But then the conditions $C_{a_h,c} = 1$ and $D_{b_h,c} = 0$ imply that $X_q \leq 0$. Since $q \subseteq p$, we get $X_p \leq 0$ from Claim 1, and hence $X_p = 0$; a contradiction.

For 6 argue as in 4 using equation (5) for the same q and $b = b_h$. □

The next claim states the forth property:

Claim 3. *Let $q \in R_{k-1}$. If $X_q \neq 0$, then for every $X \subseteq A$, there exists $Y \subseteq B$ with $|Y| = |X|$ such that for every $b \in Y$ there exists $a \in X$ such that $X_{q \cup ab} \neq 0$.*

Proof. Assume $X_q \neq 0$. For every $(a, b) \in A \times B$, define $Y_{a,b} = X_{q \cup ab}/X_q$ and let $\mathbf{Y}$ be the $|A| \times |B|$ matrix $(Y_{a,b})_{a \in A, b \in B}$. Equations (6), (8) and (9) imply that $\mathbf{Y}$ is a doubly stochastic matrix. Therefore $\mathbf{Y}$ is the convex combination of one or more permutation matrices: $\mathbf{Y} = \sum_{t=1}^{r} \alpha_t \Pi_t$ with $r \geq 1$ and $\alpha_t > 0$ for every $t \in \{1, \ldots, r\}$. Let π be the permutation underlying Π_1 interpreted like a bijection from A to B. For every $X \subseteq A$, define $Y = \pi(X)$. Obviously $|Y| = |X|$. Moreover, for every $b \in Y$, choose $a = \pi^{-1}(b) \in X$ and check:

$$Y_{a,b} = \sum_{t=1}^{r} \alpha_t \Pi_t(a, b) \geq \alpha_1 \Pi_1(a, b) = \alpha_1 > 0.$$

This implies $X_{q \cup ab} \neq 0$ and we are done. □

The final claim states the back property:

Claim 4. *Let $q \in R_{k-1}$. If $X_q \neq 0$, then for every $Y \subseteq B$, there exists $X \subseteq A$ with $|X| = |Y|$ such that for every $a \in X$ there exists $b \in Y$ such that $x_{q \cup ab} \neq 0$.*

Proof. This proof is the same is Claim 3 with the roles of X and Y, and a and b reversed. □

These claims complete the proof of the lemma. □

4.4 From pebble game to sliding game

For this section, fix $k \geq 1$. We show that if the Duplicator has a winning strategy in the non-sliding game with $k + 1$ pebbles, then she also has a winning strategy in the sliding game with k pebbles. Intuitively, the idea is that the Duplicator can use her stategy in the non-sliding game to simulate the moves of the sliding game by pretending that the Spoiler makes restricted use of pebble $k + 1$.

More precisely, if Spoiler slides pebble $i \in [k]$ from a to a' in the sliding game, then Duplicator pretends that Spoiler actually does the following: place pebble $k + 1$ on a' to force the sliding condition on the Duplicator side, then move pebble i from a to a' to actually get the move done, and finally remove pebble $k + 1$ out the board to leave it free for the next move. We make this argument formal in the next lemma:

Lemma 2. *Let $k \geq 1$. If $\mathbf{A} \equiv_{\mathrm{C}}^{k+1} \mathbf{B}$, then $\mathbf{A} \equiv_{\mathrm{CS}}^{k} \mathbf{B}$.*

Proof. Let $\mathcal{F}$ be a winning strategy witnessing that $\mathbf{A} \equiv_{\mathrm{C}}^{k+1} \mathbf{B}$. Let $\mathcal{H}$ be the collection of all pairs of k-tuples $(\mathbf{a}', \mathbf{b}')$, with $\mathbf{a}' = (a'_1, \ldots, a'_k) \in (A \cup \{\star\})^k$ and $\mathbf{b}' = (b'_1, \ldots, b'_k) \in (B \cup \{\star\})^k$, for which there exists $(\mathbf{a}, \mathbf{b})$ in $\mathcal{F}$, with $\mathbf{a} = (a_1, \ldots, a_{k+1})$ and $\mathbf{b} = (b_1, \ldots, b_{k+1})$, such that $a_i = a'_i$ and $b_i = b'_i$ for every $i \in [k]$. In words, $\mathbf{a}'$ and $\mathbf{b}'$ are the projections on the first k components of some pair of tuples $(\mathbf{a}, \mathbf{b})$ that belongs to $\mathcal{F}$. We claim that $\mathcal{H}$ is a winning strategy in the k-pebble sliding game.

First, $\mathcal{H}$ is non-empty since $\mathcal{F}$ is non-empty. Second, every $(\mathbf{a}', \mathbf{b}')$ in $\mathcal{H}$ is a partial k-isomorphism since the corresponding $(\mathbf{a}, \mathbf{b})$ in $\mathcal{F}$ is a partial $k+1$-isomorphism. Third, for every $(\mathbf{a}', \mathbf{b}')$ in $\mathcal{H}$ and every $i \in [k]$, the pair $(\mathbf{a}'[i/\star], \mathbf{b}'[i/\star])$ belongs to $\mathcal{H}$ by the closure under subtuples of $\mathcal{F}$. Next we argue that the back and forth property are satisfied. By symmetry, it suffices to check the forth property with $+$-orientation.

Fix $(\mathbf{a}', \mathbf{b}')$ in $\mathcal{H}$, with $\mathbf{a}' = (a'_1, \ldots, a'_k)$ and $\mathbf{b}' = (b'_1, \ldots, b'_k)$. Let $\mathbf{a} = (a'_1, \ldots, a'_k, a_{k+1})$ and $\mathbf{b} = (b'_1, \ldots, b'_k, b_{k+1})$ be the corresponding pair of tuples in $\mathcal{F}$. Fix $i \in [k]$ and $X \subseteq N^+(a'_i)$. By the closure under subtubles of $\mathcal{F}$ and the forth property of $\mathcal{F}$ applied to the pair $(\mathbf{a}[k+1/\star], \mathbf{b}[k+1/\star])$, component $k+1$, and set $X \subseteq N^+(\star) = A$, there exists $Y \subseteq N^+(\star) = B$ with $|Y| = |X|$ such that for every $b \in Y$ there exists $a \in X$ such that $(\mathbf{a}[k+1/a], \mathbf{b}[k+1/b])$ belongs to $\mathcal{F}$. Now let us show that:

Claim 5. $Y \subseteq N^+(b'_i)$.

Proof. If $b'_i = \star$ there is nothing to show since in that case $N^+(b'_i) = B$ and it is obvious that $Y \subseteq B$. Assume then that $b'_i \neq \star$. Fix an arbitrary element $b \in Y$. We want to show that (b'_i, b) is an edge in $\mathbf{B}$. By choice of Y, there exists $a \in X$ such that $(\mathbf{a}[k+1/a], \mathbf{b}[k+1/b])$ belongs to $\mathcal{F}$. In particular $(\mathbf{a}[k+1/a], \mathbf{b}[k+1/b])$ is a partial $k+1$-isomorphism, and since (a'_i, a) is an edge in $\mathbf{A}$, also (b'_i, b) must be an edge in $\mathbf{B}$. This shows that $Y \subseteq N^+(b'_i)$. $\square$

Next we show:

Claim 6. *For every $b \in Y$, there exists $a \in X$ such that $(\mathbf{a}'[i/a], \mathbf{b}'[i/b])$ belongs to $\mathcal{H}$.*

Proof. In the proof of the previous claim we argued that for every $b \in Y$, there exists $a \in X$ such that $(\mathbf{a}[k+1/a], \mathbf{b}[k+1/b])$ belongs to $\mathcal{F}$. By the forth property of $\mathcal{F}$ applied to the pair of tuples $(\mathbf{a}[k+1/a], \mathbf{b}[k+1/b])$, component i, and set $X' = \{a\} \subseteq N^+(a'_i)$, there exists $Y' \subseteq N^+(b'_i)$ with $|Y'| = |X'|$ such that for every $b' \in Y'$ there exists $a' \in X'$ such that $(\mathbf{a}[k+1/a, i/a'], \mathbf{b}[k+1/b, i/b'])$ belongs to $\mathcal{F}$. But since the members of $\mathcal{F}$ define partial $k+1$-isomorphisms and the only a' in X' is a, necessarily $Y' = \{b\}$ since otherwise the components i and $k+1$ would be equal in $\mathbf{a}[k+1/a, i/a']$ and different in $\mathbf{a}[k+1/b, i/b']$.

The previous paragraph shows that for every $b \in Y$ there exists $a \in X$ such that the pair $(\mathbf{a}[k+1/a, i/a], \mathbf{b}[k+1/b, i/b])$ belongs to $\mathcal{F}$. Since $(\mathbf{a}'[i/a], \mathbf{b}'[i/b])$ is precisely the pair of projections on the first k components of the tuples in $(\mathbf{a}[k+1/a, i/a], \mathbf{b}[k+1/b, i/b])$, this shows that for every $b \in Y$ there exists $a \in X$ such that $(\mathbf{a}'[i/a], \mathbf{b}'[i/b])$ belongs to $\mathcal{H}$. $\square$

The proof of the forth property of $\mathcal{H}$ is complete, and with it the proof of the lemma. $\square$

4.5 From sliding game to common equitable partition

For this section, fix $k \geq 1$. Let $\mathbf{a}$ and $\mathbf{b}$ be k-tuples in $(A \cup \{\star\})^k$ and $(B \cup \{\star\})^k$ respectively. Define $(\mathbf{a}, \mathbf{A}) \equiv (\mathbf{b}, \mathbf{B})$ if $(\mathbf{a}, \mathbf{b})$ belongs to some winning strategy for the Duplicator in the k-pebble sliding game on $\mathbf{A}$ and $\mathbf{B}$.

Lemma 3. *$\equiv$ is an equivalence relation.*

Proof. The symmetry of the relation follows from the symmetry of the game, and its reflexivity is clear. The only property that requires checking is transitivity. Assume $(\mathbf{a}, \mathbf{A}) \equiv (\mathbf{b}, \mathbf{B})$ and $(\mathbf{b}, \mathbf{B}) \equiv (\mathbf{c}, \mathbf{C})$. Let $\mathcal{F}$ and $\mathcal{F}'$ be the two winning strategies witnessing these facts. Let $\mathcal{G}$ be the collection of all pairs of k-tuples $(\mathbf{a}', \mathbf{c}')$ with $\mathbf{a}' \in (A \cup \{\star\})^k$ and $\mathbf{c}' \in (C \cup \{\star\})^k$ for which there exists a k-tuple $\mathbf{b}' \in (B \cup \{\star\})^k$ such that $(\mathbf{a}', \mathbf{b}')$ belongs to $\mathcal{F}$ and $(\mathbf{b}', \mathbf{c}')$ belongs to $\mathcal{F}'$. Clearly each $(\mathbf{a}', \mathbf{c}')$ in $\mathcal{G}$ defines a partial k-isomorphism from $\mathbf{A}$ to $\mathbf{C}$. Moreover, $(\mathbf{a}'[i/\star], \mathbf{c}'[i/\star])$ belongs to $\mathcal{G}$ by the closure under subtuples properties of $\mathcal{F}$ and $\mathcal{F}'$. Indeed $(\mathbf{a}'[i/\star], \mathbf{b}'[i/\star])$ belongs to $\mathcal{F}$ and $(\mathbf{b}'[i/\star], \mathbf{c}'[i/\star])$ belongs to $\mathcal{F}'$ for the $\mathbf{b}'$ that witnesses that $(\mathbf{a}', \mathbf{c}')$ belongs to $\mathcal{G}$. The back and forth properties of $\mathcal{G}$ are also easily derived from the back and forth properties of $\mathcal{F}$ and $\mathcal{F}'$. Finally, $\mathcal{G}$ contains the pair $(\mathbf{a}, \mathbf{c})$ by construction, which means that it is non-empty, and hence a winning strategy witnessing that $(\mathbf{a}, \mathbf{A}) \equiv (\mathbf{c}, \mathbf{C})$. □

In restriction to a single structure $\mathbf{A}$, the equivalence relation $\equiv$ can be thought as an equivalence relation on $(A \cup \{\star\})^k$.

Lemma 4. *The sequence of equivalence classes of $\equiv$ on $(A \cup \{\star\})^k$ is a k-equitable partition of $\mathbf{A}$.*

Proof. Let $(P_1, \ldots, P_s)$ be the equivalence classes of $\equiv$ on $(A \cup \{\star\})^k$. This forms a partition of $(A \cup \{\star\})^k$. Fix an index $m \in [s]$, and tuples $\mathbf{a} = (a_1, \ldots, a_k)$ and $\mathbf{a}' = (a'_1, \ldots, a'_k)$ in P_m. Since $\mathbf{a} \equiv \mathbf{a}'$, the pair $(\mathbf{a}, \mathbf{a}')$ belongs to some winning strategy $\mathcal{F}$. In particular it defines a partial k-isomorphism from $\mathbf{A}$ to $\mathbf{A}$.

To argue that $c(\mathbf{a}[i/\star]) = c(\mathbf{a}'[i/\star])$ for every $i \in [k]$, note that $(\mathbf{a}[i/\star], \mathbf{a}'[i/\star])$ also belongs to $\mathcal{F}$ by the closure under subtuples property in the definition of winning strategy.

Next we want to show that $d_i^o(\mathbf{a}, P_n) = d_i^o(\mathbf{a}', P_n)$ for every $i \in [k]$, $o \in \{+, -\}$, and $n \in [s]$. First we consider the case that $\mathbf{a}[i/\star]$ lands in P_n. In this case $d_i^o(\mathbf{a}, P_n) = 1$ since every tuple in P_n must be equivalent to $\mathbf{a}[i/\star]$ and hence have $\star$ in the i-th component, and $d_i^o(\mathbf{a}, P_n)$ is precisely the number of $a \in N^o(a_i) \cup \{\star\}$ such that $\mathbf{a}[i/a]$ belongs to P_n. Also $\mathbf{a}'[i/\star]$ lands in P_n by the previous paragraph, and hence $d_i^o(\mathbf{a}', P_n) = 1$ by the same argument.

Next we consider the case where $\mathbf{a}[i/\star]$ does not land in P_n. Let X be the set of all $a \in N^o(a_i)$ such that $\mathbf{a}[i/a]$ belongs to P_n. Then $|X| = d_i^o(\mathbf{a}, P_n)$. Similarly, let X' be the set of all $a' \in N^o(a'_i)$ such that $\mathbf{a}'[i/a']$ belongs to P_n. Since $\mathbf{a}'[i/\star]$ does not land in P_n either because $c(\mathbf{a}[i/\star]) = c(\mathbf{a}'[i/\star])$, we have $|X'| = d_i^o(\mathbf{a}', P_n)$. We show that $|X| = |X'|$.

Let $Y \subseteq N^o(a'_i)$ be the set guaranteed to exist by the forth property of $\mathcal{F}$ for the pair of tuples $(\mathbf{a}, \mathbf{a}')$, index i, and set X. Then $|Y| = |X|$. We claim that $Y \subseteq X'$. To show this, observe that for each $a' \in Y$ there exists some $a \in X$ such that $(\mathbf{a}[i/a], \mathbf{a}'[i/a'])$ belongs to $\mathcal{F}$. Hence $\mathbf{a}[i/a] \equiv \mathbf{a}'[i/a']$ which means that $\mathbf{a}'[i/a']$ belongs to the equivalence class P_n of $\mathbf{a}[i/a]$. This shows that $Y \subseteq X'$. Therefore $|Y| \leq |X'|$ and hence $|X| \leq |X'|$ because $|Y| = |X|$. The symmetric argument exchanging the roles of $\mathbf{a}$, X and $\mathbf{a}'$, X' would show that $|X'| \leq |X|$. Thus $|X| = |X'|$ as was to be shown.

To argue that $c(\mathbf{a} \circ \pi) = c(\mathbf{a}' \circ \pi)$ for every permutation $\pi \in S_k$, note that $\mathcal{F} \circ \pi$ defined as $\{(\mathbf{c} \circ \pi, \mathbf{c}' \circ \pi) : (\mathbf{c}, \mathbf{c}') \in \mathcal{F}\}$ is also a winning strategy. The same argument shows that $|P_{c(\mathbf{a})}| = |P_{c(\mathbf{a} \circ \pi)}|$. □

Lemma 5. *Let $k \geq 1$. If $\mathbf{A} \equiv^k_{\mathrm{CS}} \mathbf{B}$, then $\mathbf{A} \equiv^k_{\mathrm{EP}} \mathbf{B}$.*

Proof. Let $(P_1, \dots, P_s)$ be the k-equitable partition given by $\equiv$ on $\mathbf{A}$. Similarly, let $(Q_1, \dots, Q_t)$ be the k-equitable partition given by $\equiv$ on $\mathbf{B}$.

By hypothesis there exists a winning strategy for the Duplicator on $\mathbf{A}$ and $\mathbf{B}$. Let $\mathcal{F}$ be such a strategy. By the forth property of $\mathcal{F}$, for every $\mathbf{a}$ in A^k there exists $\mathbf{b} = \mathbf{b}(\mathbf{a})$ in B^k such that $(\mathbf{a}, \mathbf{b})$ belongs to $\mathcal{F}$, therefore $(\mathbf{a}, \mathbf{A}) \equiv (\mathbf{b}(\mathbf{a}), \mathbf{B})$. Moreover, by the transitivity of the equivalence relation and the fact that $(\mathbf{a}, \mathbf{A}) \equiv (\mathbf{b}(\mathbf{a}), \mathbf{B})$ for every $\mathbf{a} \in (A \cup \{\star\})^k$ it follows that $(\mathbf{a}, \mathbf{A}) \equiv (\mathbf{a}', \mathbf{A})$ if and only if $(\mathbf{b}(\mathbf{a}), \mathbf{B}) \equiv (\mathbf{b}(\mathbf{a}'), \mathbf{B})$. This means that there exists a well-defined injective mapping $\alpha : \{1, \dots, s\} \to \{1, \dots, t\}$ that takes $m \in [s]$ to the unique $n \in [t]$ such that every $\mathbf{a}$ in P_m is equivalent to every $\mathbf{b}$ in Q_n.

Claim 7. *$s = t$.*

Proof. The injective mapping $\alpha : \{1, \dots, s\} \to \{1, \dots, t\}$ shows that $s \leq t$. By symmetry we also get $t \leq s$ and hence $s = t$. ☐

Since α is indeed a bijection, we may assume that it is the identity by rearranging the partitions. In other words, from now on we assume that $(\mathbf{a}, \mathbf{A}) \equiv (\mathbf{b}, \mathbf{B})$ if and only if $c(\mathbf{a}) = c(\mathbf{b})$.

Claim 8. *$c(\mathbf{a}[i/\star]) = c(\mathbf{b}[i/\star])$ for every $i \in [k]$, $m \in [s]$, $\mathbf{a} \in P_m$ and $\mathbf{b} \in Q_m$.*

Proof. Since $(\mathbf{a}, \mathbf{A}) \equiv (\mathbf{b}, \mathbf{B})$, the pair $(\mathbf{a}, \mathbf{b})$ belongs to some winning strategy $\mathcal{F}$, but then the pair $(\mathbf{a}[i/\star], \mathbf{b}[i/\star])$ also belongs to $\mathcal{F}$ by the closure under subtuples of winning strategies. This shows that $c(\mathbf{a}[i/\star]) = c(\mathbf{b}[i/\star])$. ☐

Next we show that the degrees are the same:

Claim 9. *$d_i^o(\mathbf{a}, P_n) = d_i^o(\mathbf{b}, Q_n)$ for every $i \in [k]$, $o \in \{+, -\}$, $m, n \in [s]$, $\mathbf{a} \in P_m$ and $\mathbf{b} \in Q_m$.*

Proof. Let $\mathbf{a} = (a_1, \dots, a_k)$ and $\mathbf{b} = (b_1, \dots, b_k)$. First we consider the case that $\mathbf{a}[i/\star]$ lands in P_n. In this case $d_i^o(\mathbf{a}, P_n) = 1$ since every tuple in P_n must be equivalent to $\mathbf{a}[i/\star]$ and hence have $\star$ in the i-th component, and $d_i^o(\mathbf{a}, P_n)$ is precisely the number of $a \in N^o(a_i) \cup \{\star\}$ for which $\mathbf{a}[i/a]$ lands in P_n. By Claim 8, also $\mathbf{b}[i/\star]$ lands in Q_n. Hence $d_i^o(\mathbf{b}, Q_n) = 1$ by the same argument, which completes this case.

Next we consider the case that $\mathbf{a}[i/\star]$ does not land in P_n. Let X be the set $\{a \in N^o(a_i) : \mathbf{a}[i/a] \in P_n\}$. Thus $|X| = d_i^o(\mathbf{a}, P_n)$. By the definition of winning strategy for the Duplicator there exists a set $Y \subseteq N^o(b_i)$ with $|Y| = |X|$ such that for every $b \in Y$ there exists $a \in X$ such that $(\mathbf{a}[i/a], \mathbf{b}[i/b])$ belongs to $\mathcal{F}$. Since this implies $(\mathbf{a}[i/a], \mathbf{A}) \equiv (\mathbf{b}[i/b], \mathbf{B})$ we can conclude that $\mathbf{b}[i/b] \in Q_n$ for every $b \in Y$. Thus $d_i^o(\mathbf{b}, Q_n) \geq |Y| = |X| = d_i^o(\mathbf{a}, P_n)$.

The symmetric condition for winning strategy implies the opposite inequality and putting the two together we have $d_i^o(\mathbf{a}, P_n) = d_i^o(\mathbf{b}, Q_n)$. ☐

Next we show that the classes have the same sizes:

Claim 10. *$|P_m| = |Q_m|$ for every $m \in [s]$.*

Proof. First notice that the fact that there is a winning strategy for the Duplicator implies that $|A| = |B|$. To see this note first that the pair of k-tuples $(\star^k, \star^k)$ belongs to the winning strategy by the closure under subtuples of winning strategies, and that the forth property applied to this pair of tuples and any $i \in [k]$, $o \in \{+, -\}$ requires that for every $X \subseteq N^o(\star) = A$ there must exist a $Y \subseteq N^o(\star) = B$ such that $|Y| = |X|$, among other properties. In particular choosing $X = A$ we get $|B| \geq |A|$. By the symmetric condition we also get $|A| \geq |B|$. Using the equality between the sizes of A and B the statement of this claim follows easily from the previous one.

For every $m, n \in [s]$, $\mathbf{a} \in P_m$, $\mathbf{a}' \in P_n$, $\mathbf{b} \in Q_m$ and $\mathbf{b}' \in Q_n$ we have the identities

$$|P_m| d_i^+(\mathbf{a}, P_n) = d_i^+(P_m, P_n) = d_i^-(P_n, P_m) = |P_n| d_i^-(\mathbf{a}', P_m)$$
$$|Q_m| d_i^+(\mathbf{b}, Q_n) = d_i^+(Q_m, Q_n) = d_i^-(Q_n, Q_m) = |Q_n| d_i^-(\mathbf{b}', Q_m).$$

Therefore

$$\frac{|P_m|}{|P_n|} = \frac{d_i^-(\mathbf{a}', P_m)}{d_i^+(\mathbf{a}, P_n)} = \frac{d_i^-(\mathbf{b}', Q_m)}{d_i^+(\mathbf{b}, Q_n)} = \frac{|Q_m|}{|Q_n|}$$

where the second equality follows from the previous claim. This means that the ratio $r = |P_m|/|Q_m|$ does not depend on m, and since $|A| = \sum_{m=1}^{s} |P_m| = r \sum_{m=1}^{s} |Q_m| = r|B|$, it follows that $r = 1$. ☐

Claim 11. *$c(\mathbf{a} \circ \pi) = c(\mathbf{b} \circ \pi)$ for every permutation $\pi \in S_k$, $m \in [s]$, $\mathbf{a} \in P_m$ and $\mathbf{b} \in Q_m$.*

Proof. Since $(\mathbf{a}, \mathbf{A}) \equiv (\mathbf{b}, \mathbf{B})$, the pair $(\mathbf{a}, \mathbf{b})$ belongs to some winning strategy $\mathcal{F}$. But then the pair $(\mathbf{a} \circ \pi, \mathbf{b} \circ \pi)$ belongs to $\mathcal{F} \circ \pi$ defined by $\{(\mathbf{c} \circ \pi, \mathbf{c}' \circ \pi) : (\mathbf{c}, \mathbf{c}') \in \mathcal{F}\}$, which is again a winning strategy. This shows that $c(\mathbf{a} \circ \pi) = c(\mathbf{b} \circ \pi)$. ☐

These claims show that $(P_1, \ldots, P_s)$ and $(Q_1, \ldots, Q_s)$ witness that $\mathbf{A}$ and $\mathbf{B}$ have a common k-equitable partition. ☐

4.6 From common equitable partition to Sherali-Adams

We prove the last implication of the Transfer Lemma:

Lemma 6. *Let $k \geq 1$. If $\mathbf{A} \equiv_{\mathrm{EP}}^{k} \mathbf{B}$, then $\mathbf{A} \equiv_{\mathrm{SA}}^{k} \mathbf{B}$.*

Proof. Let $(P_1, \ldots, P_s)$ and $(Q_1, \ldots, Q_s)$ be the common k-equitable partition of $\mathbf{A}$ and $\mathbf{B}$.

For every $q \subseteq A \times B$ with $|q| \leq k$, if q is not a partial mapping define $X_q = 0$. If q is a partial mapping, define X_q as follows. Let $a_1, \ldots, a_r$ be an enumeration without repetitions of $\mathrm{Dom}(q)$. In particular $r \leq k$. Let $\mathbf{a} = (a_1, \ldots, a_r, \star, \ldots, \star)$ be the k-tuple that starts with $a_1, \ldots, a_r$ and is padded to length k by adding stars. Let $\mathbf{b} = (b_1, \ldots, b_k)$ be the k-tuple defined by $b_i = q(a_i)$ for every $i \in \{1, \ldots, r\}$ and $b_i = \star$ for every $i \in \{r+1, \ldots, k\}$. Let $m = c(\mathbf{a})$ and $n = c(\mathbf{b})$. If $m \neq n$, define $X_q = 0$. If $m = n$, define $X_q = 1/|P_m| = 1/|Q_m|$. Since $c(\mathbf{a} \circ \pi) = c(\mathbf{b} \circ \pi)$ and $|P_{c(\mathbf{a})}| = |P_{c(\mathbf{a} \circ \pi)}|$ hold for every permutation $\pi \in S_k$, this definition does not depend on the choice of the enumeration $a_1, \ldots, a_r$ and is hence well-defined.

Claim 12. *If $|q| < k$ and $a \in A$, then $X_q = \sum_{b \in B} X_{q \cup ab}$.*

Proof. If q is not a partial mapping, then $X_q = 0$ and $X_{q \cup ab} = 0$ for every $b \in B$, and the identity is obvious. Assume then that q is a partial mapping and that $|q| < k$. Let $a_1, \ldots, a_r$ be an enumeration without repetitions of $\mathrm{Dom}(q)$. In particular $r < k$. Let $\mathbf{a} = (a_1, \ldots, a_r, \star, \ldots, \star)$ be the k-tuple that starts with $a_1, \dots, a_r$ and is padded to length k by adding stars. Let $\mathbf{b} = (b_1, \dots, b_k)$ be the k-tuple defined by $b_i = q(a_i)$ for every $i \in \{1, \dots, r\}$ and $b_i = \star$ for every $i \in \{r+1, \dots, k\}$. Setting $i = r + 1$ for the rest of the proof, in particular $a_i = b_i = \star$.

Let $m = c(\mathbf{a})$ and $n = c(\mathbf{b})$. If $m \neq n$, we have $X_q = 0$ by definition, and also $X_{q \cup ab} = 0$ for every $b \in B$ since otherwise $c(\mathbf{a}[i/a]) = c(\mathbf{b}[i/b])$, which implies $c(\mathbf{a}) = c(\mathbf{b})$, and hence $m = n$, by the definition of common equitable partition. Since this makes the identity obvious, we may assume that $m = n$.

Recall $i = r + 1$ and let $\mathbf{a}' = \mathbf{a}[i/a]$ and $m' = c(\mathbf{a}')$. Note that none of the tuples $\mathbf{b}'$ in $Q_{m'}$ can have $\star$ in the i-th component since $(\mathbf{a}', \mathbf{b}')$ must define a partial k-isomorphism, $\mathbf{a}'$ does not have it. We claim that

$$\sum_{b \in B} X_{q \cup ab} = \frac{d_i^+(\mathbf{b}, Q_{m'})}{|Q_{m'}|} = \frac{d_i^-(Q_{m'}, Q_m)}{|Q_m||Q_{m'}|} = \frac{1}{|Q_m|} = X_q.$$

The first equality follows from the definition of $X_{q \cup ab}$ that sets $X_{q \cup ab} = 0$ if $\mathbf{b}[i/b]$ does not belong to $Q_{m'}$, and $X_{q \cup ab} = 1/|Q_{m'}|$ if $\mathbf{b}[i/b]$ belongs to $Q_{m'}$, together with the facts that $N^+(b_i) = B$ since $b_i = \star$, and that $\mathbf{b}[i/\star]$ does not land in $Q_{m'}$ since no tuple in $Q_{m'}$ has $\star$ in the i-th component. The second equality follows from the identity

$$|Q_m| d_i^+(\mathbf{b}, Q_{m'}) = d_i^+(Q_m, Q_{m'}) = d_i^-(Q_{m'}, Q_m).$$

For the third equality, let $b \in B$ be such that $\mathbf{b}[i/b]$ lands in $Q_{m'}$. Such a b must exist since $d_i^+(\mathbf{b}, Q_{m'}) = d_i^+(\mathbf{a}, P_{m'})$ and $d_i^+(\mathbf{a}, P_{m'}) \geq 1$ as $\mathbf{a}[i/a]$ lands in $P_{m'}$ and $a \in N^+(a_i) = N^+(\star) = A$. Again we are using the fact that no tuple in $Q_{m'}$ has $\star$ in the i-th component to make sure that the count $d_i^+(\mathbf{b}, Q_{m'})$ does not include $\star$. Now we have $d_i^-(\mathbf{b}[i/b], Q_m) = 1$ since $d_i^-(\mathbf{b}[i/b], Q_m)$ is precisely the number of b' in $N^-(b) \cup \{\star\}$ such that $\mathbf{b}[i/b']$ belongs to Q_m, but the only such b' is $\star$. Indeed, every tuple $\mathbf{b}'$ in Q_m has $\star$ in the i-th component since $(\mathbf{b}, \mathbf{b}')$ must define a partial k-isomorphism, and $\mathbf{b}$ has it. This together with the identity

$$d^-(Q_{m'}, Q_m) = |Q_{m'}| d_i^-(\mathbf{b}[i/b], Q_m)$$

proves the third equality and the claim. □

Claim 13. *If $|q| < k$ and $b \in B$, then $X_q = \sum_{a \in A} X_{q \cup ab}$.*

Proof. The proof is the same as above: exchange the roles of a and b, and $\mathbf{A}$ and $\mathbf{B}$. □

Claim 14. *If $|q| < k$, $a \in A$ and $b \in B$, then $\sum_{a' \in A} A_{a,a'} X_{q \cup a'b} = \sum_{b' \in B} X_{q \cup ab'} B_{b',b}$.*

Proof. If q is not a partial mapping, then $X_q = 0$ and $X_{q \cup a'b} = 0$ for every $b \in B$, and the identity is obvious. Assume then that q is a partial mapping and that $|q| < k$. Let $a_1, \dots, a_r$ be an enumeration without repetitions of $\mathrm{Dom}(q)$. In particular $r < k$ since we are assuming $|q| < k$. Let $\mathbf{a} = (a_1, \dots, a_r, a, \star, \dots, \star)$ be the k-tuple that starts with $a_1, \dots, a_r$, follows with a, and is padded to length k by adding stars. Similarly, let $\mathbf{b} = (q(a_1), \dots, q(a_r), b, \star, \dots, \star)$ be the k-tuple that starts with $q(a_1), \dots, q(a_r)$, follows with b, and is padded to length k by adding stars.

Set $i = r + 1$ for the rest of the proof and let $m = c(\mathbf{a})$ and $n = c(\mathbf{b})$. By the same argument as in Claim 12, note that none of the tuples in Q_m or P_n have $\star$ in the i-th component since neither $\mathbf{a}$ nor $\mathbf{b}$ have it. We claim that

$$\sum_{a' \in A} A_{a,a'} X_{q \cup a'b} \;=\; \sum_{a' \in N^+(a)} X_{q \cup a'b} \;=\; \frac{d_i^+(\mathbf{a}, P_n)}{|P_n|}. \tag{13}$$

The first equality is obvious. The second equality follows from the definition of $X_{q \cup a'b}$ that sets $X_{q \cup a'b} = 0$ if $\mathbf{a}[i/a']$ does not belong to P_n, and $X_{q \cup a'b} = 1/|P_n|$ if $\mathbf{a}[i/a']$ belongs to P_n, together with the fact that $\mathbf{a}[i/\star]$ does not land in P_n since none of the tuples in P_n has $\star$ in the i-th component.

At the same time we claim that

$$\sum_{b' \in B} X_{q \cup ab'} B_{b',b} \;=\; \sum_{b' \in N^-(b)} X_{q \cup ab'} \;=\; \frac{d_i^-(\mathbf{b}, Q_m)}{|Q_m|}. \tag{14}$$

Again the first equality is obvious, and the second equality follows from the definition of $X_{q \cup ab'}$, together with the fact that the tuples in Q_m do not have $\star$ in the i-th component.

Fix $\mathbf{a}' \in P_n$ and $\mathbf{b}' \in Q_m$. From the definition of common equitable partition we have $d_i^-(\mathbf{a}', P_m) = d_i^-(\mathbf{b}, Q_m)$ and $d_i^+(\mathbf{b}', Q_n) = d_i^+(\mathbf{a}, P_n)$. Moreover $|P_m| = |Q_m|$ and $|P_n| = |Q_n|$. These, together with any one of the two identities

$$|P_n| d_i^-(\mathbf{a}', P_m) = d_i^-(P_n, P_m) = d_i^+(P_m, P_n) = |P_m| d_i^+(\mathbf{a}, P_n)$$
$$|Q_m| d_i^+(\mathbf{b}', Q_n) = d_i^+(Q_m, Q_n) = d_i^-(Q_n, Q_m) = |Q_n| d_i^-(\mathbf{b}, Q_m),$$

give the identity

$$\frac{d_i^-(\mathbf{b}, Q_m)}{|Q_m|} = \frac{d_i^+(\mathbf{a}, P_n)}{|P_n|}.$$

This shows the equality between (13) and (14). □

Claim 15. *If $|q| < k$, $a \in A$ and $c \in [r]$, then $X_q C_{a,c} = \sum_{b \in B} X_{q \cup ab} D_{b,c}$.*

Proof. First assume that $C_{a,c} = 0$, so the left-hand side is 0. Then for every $b \in B$ we have either $D_{b,c} = 0$, or $D_{b,c} = 1$ and then $X_{q \cup ab} = 0$ since $q \cup ab$ cannot be a partial isomorphism in this case. Thus, each term in the right-hand side is 0.

Next assume that $C_{a,c} = 1$, so the left-hand side is X_q. Then $X_{q \cup ab} = 0$ whenever $D_{b,c} = 0$ since $q \cup ab$ cannot be a partial isomorphism in this case. Thus, the right-hand side can be written as

$$\sum_{b \in B} X_{q \cup ab}$$

which equals X_q by equation (6). □

Claim 16. *If $|q| < k$, $b \in B$ and $c \in [r]$, then $X_q D_{b,c} = \sum_{a \in A} X_{q \cup ab} C_{a,c}$.*

Proof. This proof is the same as in the previous claim exchanging the roles of a and b, and C and D. □

These claims show that the proposed assignment satisfies all the equations of $F_k(\mathbf{A}, \mathbf{B})$. Since the components are non-negative, the lemma follows. □

5 Preservation of Local Linear Programs

Many of the linear programs that appear in the combinatorial optimization literature are composed of linear inequalities that are in some sense *local*: the variables involved in the inequality talk about some small *neighborhood* of the graph or hypergraph, or whatever combinatorial structure the linear program refers to. In this section we isolate one such definition of *local linear program* and show that its polytope of feasible solutions is preserved by the SA-levels of fractional isomorphism. This will be of use in the applications of Sections 6 and 7.

5.1 Local linear programs

Let $\mathbf{A} = (A, E^{\mathbf{A}}, C_1^{\mathbf{A}}, \ldots, C_r^{\mathbf{A}})$ be a colored directed graph. Let the size of a tuple $\mathbf{a} \in A^k$, denoted by $|\mathbf{a}|$, be the number of distinct elements in the tuple. For a tuple $\mathbf{a} = (a_1, \ldots, a_k) \in A^k$, let us temporarily define $\gamma : \{a_1, \ldots, a_k\} \to \{1, \ldots, |\mathbf{a}|\}$ to be the unique bijective map such that $\gamma(a_i) \leq |(a_1, \ldots, a_i)|$ for every $i \in [k]$. We will denote by $[\mathbf{A}, \mathbf{a}]$ the generic colored directed graph isomorphic to the subgraph of $\mathbf{A}$ induced by $\{a_1, \ldots, a_k\}$ together with the tuple corresponding to $\mathbf{a}$, which we refer to as its *order-tuple*. Thus, in $[\mathbf{A}, \mathbf{a}]$:

1. the vertices are $\{1, \ldots, |\mathbf{a}|\}$,
2. the edges are $\{(\gamma(a), \gamma(a')) : (a, a') \in E^{\mathbf{A}}\}$,
3. the i-th color is $\{\gamma(a) : a \in C_i^{\mathbf{A}}\}$,
4. the order-tuple is $(\gamma(a_1), \gamma(a_2), \ldots, \gamma(a_k))$.

For two tuples $\mathbf{a} = (a_1, \ldots, a_m)$ and $\mathbf{b} = (b_1, \ldots, b_n)$, we write $\mathbf{ab}$ for the concatenation tuple $(a_1, \ldots, a_m, b_1, \ldots, b_n)$. If $m = n$, we write $\mathbf{a} \oplus \mathbf{b}$ for the tuple of pairs $((a_1, b_1), \ldots, (a_m, b_m))$.

A *basic k-local LP* is specified by rational numbers $d^{[\mathbf{C},\mathbf{c}]}$ and $M_r^{[\mathbf{C},\mathbf{c}]}$ for every generic colored digraph $\mathbf{C}$ of size at most k with order-tuple $\mathbf{c}$ of length at most $2k$, and every $r \leq k$. The instantiation of the system on $\mathbf{A}$ is the system of inequalities that has one variable $x_{\mathbf{a}}$ for every tuple $\mathbf{a} \in A^{\leq k}$, and for every $\mathbf{a}' \in A^{\leq k}$ one inequality of the form

$$\sum_{r=1}^{k} \sum_{\substack{\mathbf{a} \in A^r \\ |\mathbf{aa}'| \leq k}} M_r^{[\mathbf{A},\mathbf{aa}']} \, x_{\mathbf{a}} \leq d^{[\mathbf{A},\mathbf{a}']}.$$

A *k-local LP* is a union of basic k-local LPs. If L is a k-local LP, its instantiation on $\mathbf{A}$, denoted by $L(\mathbf{A})$, is the union of the instantiations of the basic k-local systems that compose L.

5.2 Examples

Before we go on to show that the feasible solutions to local linear programs are preserved by the SA-levels of fractional isomorphism, let us give a few examples of local LPs. These examples will actually play a role later in the paper.

Typical constraints All four examples discussed contain two types of constraints for which it is easy to check the condition of k-locality: that the coefficient of a variable indexed by a tuple $\mathbf{a}$ in an inequality indexed by a tuple $\mathbf{a}'$ depends only on $[\mathbf{A}, \mathbf{aa}']$ and the length of $\mathbf{a}$. One special case that satisfies the condition is an LP consisting of a single inequality with the same coefficient for all $x_{\mathbf{a}}$ in which $\mathbf{a}$ induces a particular colored directed subgraph or one in a set of colored directed subgraphs on the structure. We call such a basic local LP *homogeneous*. The objective functions of many natural linear programs are homogeneous local LPs, as we will see.

Another special case is when the coefficient in front of variable $x_{\mathbf{a}}$ in the inequality indexed by $\mathbf{a}'$ is non-zero only if the elements in $\mathbf{a}$ are contained within $\mathbf{a}'$. In this case we have $M_r^{[\mathbf{C},\mathbf{c}]} \neq 0$ only if the first r elements of $\mathbf{c}$ are included in the last $s-r$, where s is the length of $\mathbf{c}$. The non-zero coefficients are allowed to all be different since in the case that $\mathbf{a}_1$ and $\mathbf{a}_2$ are contained in $\mathbf{a}'$, we have $[\mathbf{A}, \mathbf{a}_1\mathbf{a}'] \neq [\mathbf{A}, \mathbf{a}_2\mathbf{a}']$ whenever $\mathbf{a}_1 \neq \mathbf{a}_2$, because they have different order-tuples. We call such a basic k-local LP *bounded*. In particular, any inequality in a linear program that only mentions the variables indexed by tuples over up to k points of A is a bounded k-local LP. We see examples below.

Matchings in bipartite graphs We write the fractional matching polytope for general graphs which, for bipartite graphs, is known to coincide with its integer hull.

Let $\mathbf{G} = (V, E)$ be an undirected graph. The classical way of writing the fractional matching polytope has one variable x_e for each edge $e \in E$ and two types of constraints:

$$\begin{array}{ll} \sum_{e \in \delta(u)} x_e \leq 1 & \text{for } u \in V, \\ 0 \leq x_e \leq 1 & \text{for } e \in E, \end{array}$$

where $\delta(u)$ denotes the set of edges of $\mathbf{G}$ that are incident on u. The classical objective function is

$$\text{maximize } \sum_{e \in E} x_e.$$

In order to write this linear program as a local LP, we introduce one variable x_{uv} for every pair of vertices $u, v \in V$, and add constraints that force these variable to 0 if $\{u, v\}$ is not an edge of the graph, and force $x_{uv} = x_{vu}$ for every $u, v \in V$. We also incorporate the objective function as one additional constraint:

$$\begin{array}{lll} \frac{1}{2} \sum_{u \neq v} x_{uv} \geq W & & (15) \\ \sum_{v \neq u} x_{uv} \leq 1 & \text{for } u \in V, & (16) \\ 0 \leq x_{uv} \leq 1 & \text{for } u, v \in V, & (17) \\ x_{uv} = 0 & \text{for } u, v \in V \text{ such that } \{u, v\} \notin E, & (18) \\ x_{uv} = x_{vu} & \text{for } u, v \in V \text{ with } u \neq v. & (19) \end{array}$$

We check that this is a 2-local LP. First, inequality (15) is a homogeneous 2-local system: set $d^{[\mathbf{C},\mathbf{c}]} = -W$ if $\mathbf{C}$ is the empty graph and $\mathbf{c}$ is the empty tuple, and $d^{[\mathbf{C},\mathbf{c}]} = 0$ otherwise; set $M_r^{[\mathbf{C},\mathbf{c}]} = -1/2$ if $r = 2$ and $\mathbf{C}$ is a graph on $\{1, 2\}$ and $\mathbf{c} = (1, 2)$, and $M_r^{[\mathbf{C},\mathbf{c}]} = 0$ otherwise. Second, inequality (16) is a local LP: set $d^{[\mathbf{C},\mathbf{c}]} = 1$ if $\mathbf{C}$ is a graph on $\{1\}$ and $\mathbf{c} = (1)$, and $d^{[\mathbf{C},\mathbf{c}]} = 0$ otherwise; set $M_r^{[\mathbf{C},\mathbf{c}]} = 1$ if $r = 2$ and $\mathbf{C}$ is a graph on $\{1, 2\}$ and $\mathbf{c} = (1, 2, 1)$, and $M_r^{[\mathbf{C},\mathbf{c}]} = 0$ otherwise. The remaining inequalities are bounded 2-local systems. Thus, the result is a union of basic 2-local LPs, and hence a 2-local LP.

Maximum flows A network is a directed graph without self-loops, and with two distinguished vertices s and t. We code these as colored directed graphs $\mathbf{G} = (V, E, S, T)$, with color S set to $\{s\}$ and color T set to $\{t\}$. Our networks have unit capacities at every edge.

The classical linear program for st-flows has one variable x_e for every $e \in E$ and two types of constraints:

$$\begin{aligned} &\textstyle\sum_{e\in\delta^-(u)} x_e - \sum_{e\in\delta^+(u)} x_e = 0 && \text{for } u \in V \setminus \{s,t\}, \\ &0 \le x_e \le 1 && \text{for } e \in E, \end{aligned}$$

where $\delta^-(u)$ denotes the set of edges of $\mathbf{G}$ *entering* u, and $\delta^+(u)$ denotes the set of edges of $\mathbf{G}$ *leaving* u. The objective is to maximize the flow going out of s:

$$\text{maximize } \textstyle\sum_{e\in\delta^+(s)} x_e.$$

In order to write this linear program as a local LP, we introduce one variable x_{uv} for every pair of vertices $u, v \in V$, and add constraints that force x_{uv} to be non-zero only on edges $(u, v) \in E$. We also incorporate the objective function as a constraint:

$$\begin{aligned} &\textstyle\sum_{v\neq s} x_{sv} \ge W && && (20) \\ &\textstyle\sum_{v\neq u} x_{vu} - \sum_{v\neq u} x_{uv} = 0 && \text{for } u \in V \setminus \{s,t\}, && (21) \\ &0 \le x_{uv} \le 1 && \text{for } u, v \in V, && (22) \\ &x_{uv} = 0 && \text{for } u, v \in V \text{ such that } (u,v) \notin E. && (23) \end{aligned}$$

Inequality (20) is a homogeneous 2-local LP: set $d^{[\mathbf{C},\mathbf{c}]} = -W$ if $\mathbf{C}$ is the empty graph and $\mathbf{c}$ is the empty tuple, and $d^{[\mathbf{C},\mathbf{c}]} = 0$ otherwise; set $M_r^{[\mathbf{C},\mathbf{c}]} = -1$ if $r = 2$ and $\mathbf{C}$ is a graph on $\{1, 2\}$ with color S on vertex 1 and $\mathbf{c} = (1, 2)$, and $M_r^{[\mathbf{C},\mathbf{c}]} = 0$ otherwise. Equation (21) is a union of two basic 2-local LPs with opposite signs: one for $\le$ and one for $\ge$. In the first, set $d^{[\mathbf{C},\mathbf{c}]} = 0$ for every $\mathbf{C}$ and $\mathbf{c}$, and $M_r^{[\mathbf{C},\mathbf{c}]} = 1$ if $r = 2$ and $\mathbf{C}$ is a graph on $\{1, 2\}$ where 2 is not colored S or T and $\mathbf{c} = (1, 2, 2)$, and set $M_r^{[\mathbf{C},\mathbf{c}]} = -1$ if $r = 2$ and $\mathbf{C}$ is a graph on $\{1, 2\}$ where 1 is not colored S or T and $\mathbf{c} = (1, 2, 1)$, and $M_r^{[\mathbf{C},\mathbf{c}]} = 0$ otherwise. The remaining inequalities are bounded 2-local LPs.

SA-levels of vertex-cover Let $\mathbf{G} = (V, E)$ be an undirected graph. The vanilla linear program for vertex cover is:

$$\begin{aligned} &x_u + x_v \ge 1 && \text{for } \{u, v\} \in E, \\ &0 \le x_u \le 1 && \text{for } u \in V. \end{aligned}$$

The objective function is

$$\text{minimize } \textstyle\sum_{u\in V} x_u.$$

The corresponding t-level Sherali-Adams system is defined on the variables y_I for every $I \subseteq V$ with $|I| \le t$. For $I, J \subseteq V$, let $S(I, J) = \sum_{J'\subseteq J} (-1)^{|J'|} y_{I\cup J'}$. In particular $S(I, J) = 0$ if I and J are not disjoint. In the following definition I and J range over all disjoint subsets of V such that

$|I \cup J| \leq t-1$. We also incorporate the objective function as a constraint:

$$\begin{aligned}
&\textstyle\sum_{u \in V} y_{\{u\}} \leq W, && && (24)\\
&y_\emptyset = 1, && && (25)\\
&S(I,J) \geq 0, && && (26)\\
&S(I,J) \geq S(I \cup \{u\}, J) && \text{for } u \in V, && (27)\\
&S(I \cup \{u\}, J) + S(I \cup \{v\}, J) \geq S(I,J) && \text{for } \{u,v\} \in E. && (28)
\end{aligned}$$

To put it in the form of a local LP we need the variables to be indexed by tuples, so we replace y_I for $I = \{v_1, \ldots, v_r\}$, $r \leq t$, by $y_{\mathbf{a}}$ for $\mathbf{a} = (v_1, \ldots, v_r)$ and for every permutation $\pi : [r] \to [r]$ we add the constraint $y_{\mathbf{a}} = y_{\mathbf{a} \circ \pi}$. Constraints should also be indexed by tuples, so (28) is really a pair of (equivalent) constraints: one for (u,v) and one for (v,u).

Inequality (24), which comes from the objective function, is a homogeneous 1-local LP, which implies it is also a homogeneous $(t+1)$-local LP. The remaining constraints are bounded $(t+1)$-local LPs.

SA-levels of max-cut Again, let $\mathbf{G} = (V,E)$ be an undirected graph. The linear program relaxation for max-cut known as the metric polytope has one variable x_{uv} for every pair of vertices $u, v \in V$, and the constraints below:

$$\begin{aligned}
&0 \leq x_{uv} \leq 1 && \text{for } u,v \in V && (29)\\
&x_{uv} = x_{vu} && \text{for } u,v \in V, && (30)\\
&x_{uw} \leq x_{uv} + x_{vw} && \text{for } u,v,w \in V, && (31)\\
&x_{uv} + x_{vw} \leq 2 - x_{uw} && \text{for } u,v,w \in V. && (32)
\end{aligned}$$

The objective function is

$$\text{maximize } \tfrac{1}{2} \textstyle\sum_{\{u,v\} \in E} x_{uv}.$$

The corresponding t-level Sherali-Adams system is defined on the variables y_I for every $I \subseteq V^2$ with $|I| \leq t$. In the following system I and J range over all disjoint subsets of V^2 such that $|I \cup J| \leq t-1$. We also incorporate the objective function as a constraint.

$$\begin{aligned}
&\tfrac{1}{2} \textstyle\sum_{\{u,v\} \in E} y_{\{uv\}} \geq W, && (33)\\
&y_\emptyset = 1, && (34)\\
&S(I,J) \geq 0, && (35)\\
&S(I,J) \geq S(I \cup \{uv\}, J), && (36)\\
&S(I \cup \{uv\}, J) = S(I \cup \{vu\}, J), && (37)\\
&S(I \cup \{uw\}, J) \leq S(I \cup \{uv\}, J) + S(I \cup \{vw\}, J), && (38)\\
&S(I \cup \{uv\}, J) + S(I \cup \{vw\}, J) \leq 2S(I,J) - S(I \cup \{uw\}, J). && (39)
\end{aligned}$$

To put it in the form required by Theorem 2 we need the variables to be indexed by tuples, so we replace y_I for $I = \{(v_1, v'_1), \ldots, (v_r, v'_r)\}$, $r \leq t$, by $y_{\mathbf{a}}$ for $\mathbf{a} = (a_1, \ldots, a_r)$ with $a_i = (v_i, v'_i)$, and for every permutation $\pi : [r] \to [r]$ we add the constraint $y_{\mathbf{a}} = y_{\mathbf{a} \circ \pi}$.

Similarly to the case of vertex cover, the first constraint, which comes from the objective function, is a homogeneous 2-local LP, which implies it is also a homogeneous $(2t+1)$-local LP. The remaining constraints are bounded $(2t+1)$-local LPs.

5.3 Preservation of feasible solutions

Next we show that local LPs have the good feature that their polyhedra of feasible solutions are preserved by sufficiently high levels of the Sherali-Adams relaxation of fractional isomorphism. More precisely, to preserve k-local LPs, k levels suffice. The full statement is the following:

Theorem 2. *Let L be a k-local LP, and let $\mathbf{A}$ and $\mathbf{B}$ be colored digraphs such that $\mathbf{A} \equiv_{\mathrm{SA}}^{k} \mathbf{B}$. Then, $L(\mathbf{A})$ is feasible if and only if $L(\mathbf{B})$ is feasible. Furthermore, if $x_{\mathbf{a}}$ is a solution of $L(\mathbf{A})$ then $x_{\mathbf{b}} = \sum_{\mathbf{a} \in A^r} X_{\mathbf{a} \oplus \mathbf{b}}\, x_{\mathbf{a}}$ is a solution of $L(\mathbf{B})$, where $\mathbf{X}$ denotes the solution witnessing $\mathbf{A} \equiv_{\mathrm{SA}}^{k} \mathbf{B}$, and r is the length of $\mathbf{b}$.*

The rest of this section is devoted to the proof of Theorem 2. Let us fix $\mathbf{A}$ and $\mathbf{B}$ such that $\mathbf{A} \equiv_{\mathrm{SA}}^{k} \mathbf{B}$, and let $\mathbf{X}$ be a solution of $F_k(\mathbf{A}, \mathbf{B})$ witnessing this fact. We start with a straightforward lemma about the properties of $\mathbf{X}$ that we will use several times.

Lemma 7. *Let $0 \leq r, s \leq k$ be integers, let $\mathbf{a} \in A^r$ and $\mathbf{a}' \in A^s$ be such that $|\mathbf{a}\mathbf{a}'| \leq k$, and let $\mathbf{b} \in B^r$. Then*

$$X_{\mathbf{a} \oplus \mathbf{b}} = \sum_{\substack{\mathbf{b}' \in B^s \\ [\mathbf{A}, \mathbf{a}\mathbf{a}'] = [\mathbf{B}, \mathbf{b}\mathbf{b}']}} X_{\mathbf{a}\mathbf{a}' \oplus \mathbf{b}\mathbf{b}'}.$$

Proof. The proof is a simple induction on s. For $s = 0$ the statement trivially holds. Next, for $s \geq 1$, suppose $\mathbf{a}' = \mathbf{a}'' a$ where $\mathbf{a}'' \in A^{s-1}$ and $a \in A$. Applying Claim 2 from Section 4 we have

$$\sum_{\substack{\mathbf{b}'' \in B^{s-1}, b \in B \\ [\mathbf{A}, \mathbf{a}\mathbf{a}'' a] = [\mathbf{B}, \mathbf{b}\mathbf{b}'' b]}} X_{\mathbf{a}\mathbf{a}'' a \,\oplus\, \mathbf{b}\mathbf{b}'' b} \quad = \quad \sum_{\substack{\mathbf{b}'' \in B^{s-1} \\ [\mathbf{A}, \mathbf{a}\mathbf{a}''] = [\mathbf{B}, \mathbf{b}\mathbf{b}'']}} \sum_{b \in B} X_{\mathbf{a}\mathbf{a}'' a \,\oplus\, \mathbf{b}\mathbf{b}'' b}.$$

Equation (6) of the Sherali-Adams system shows that the right-hand side is

$$\sum_{\substack{\mathbf{b}'' \in B^{s-1} \\ [\mathbf{A}, \mathbf{a}\mathbf{a}''] = [\mathbf{B}, \mathbf{b}\mathbf{b}'']}} X_{\mathbf{a}\mathbf{a}'' \oplus \mathbf{b}\mathbf{b}''},$$

and the induction hypothesis gives that this is precisely $X_{\mathbf{a} \oplus \mathbf{b}}$. ☐

We proceed with the proof of the theorem. It is sufficient to prove the statement for a basic k-local LP L given by $M_r^{[\mathbf{C}, \mathbf{c}]}$ and $d^{[\mathbf{C}, \mathbf{c}]}$. Let $x_{\mathbf{a}}$ be a feasible solution for $L(\mathbf{A})$. Thus for every $\mathbf{a}' \in A^{\leq k}$ we have

$$\sum_{r=1}^{k} \sum_{\substack{\mathbf{a} \in A^r \\ |\mathbf{a}\mathbf{a}'| \leq k}} M_r^{[\mathbf{A}, \mathbf{a}\mathbf{a}']}\, x_{\mathbf{a}} \leq d^{[\mathbf{A}, \mathbf{a}']}. \tag{40}$$

We need to show that for every $\mathbf{b}' \in B^{\leq k}$ it holds that

$$\sum_{r=1}^{k} \sum_{\substack{\mathbf{b} \in B^r \\ |\mathbf{b}\mathbf{b}'| \leq k}} M_r^{[\mathbf{B}, \mathbf{b}\mathbf{b}']} \sum_{\mathbf{a} \in A^r} X_{\mathbf{a} \oplus \mathbf{b}}\, x_{\mathbf{a}} \leq d^{[\mathbf{B}, \mathbf{b}']}. \tag{41}$$

In the following, let $0 \leq s \leq k$ be such that $\mathbf{b}' \in B^s$. Using Lemma 7 the left-hand side of equation (41) becomes

$$\sum_{r=1}^{k} \sum_{\substack{\mathbf{b} \in B^r \\ |\mathbf{b}\mathbf{b}'| \leq k}} M_r^{[\mathbf{B}, \mathbf{b}\mathbf{b}']} \sum_{\mathbf{a} \in A^r} \sum_{\substack{\mathbf{a}' \in A^s \\ [\mathbf{A}, \mathbf{a}\mathbf{a}'] = [\mathbf{B}, \mathbf{b}\mathbf{b}']}} X_{\mathbf{a}\mathbf{a}' \oplus \mathbf{b}\mathbf{b}'}\, x_{\mathbf{a}}.$$

Rearranging the sums with care we can rewrite this as

$$\begin{aligned}
&\sum_{\substack{\mathbf{a}'\in A^s\\ [\mathbf{A},\mathbf{a}']=[\mathbf{B},\mathbf{b}']}} \sum_{r=1}^{k} \sum_{\substack{\mathbf{b}\in B^r\\ |\mathbf{bb}'|\le k}} \sum_{\substack{\mathbf{a}\in A^r\\ [\mathbf{A},\mathbf{aa}']=[\mathbf{B},\mathbf{bb}']}} M_r^{[\mathbf{B},\mathbf{bb}']}\ X_{\mathbf{aa}'\oplus\mathbf{bb}'}\ x_{\mathbf{a}} \\
&= \sum_{\substack{\mathbf{a}'\in A^s\\ [\mathbf{A},\mathbf{a}']=[\mathbf{B},\mathbf{b}']}} \sum_{r=1}^{k} \sum_{\substack{\mathbf{a}\in A^r\\ |\mathbf{aa}'|\le k}} \sum_{\substack{\mathbf{b}\in B^r\\ [\mathbf{A},\mathbf{aa}']=[\mathbf{B},\mathbf{bb}']}} M_r^{[\mathbf{B},\mathbf{bb}']}\ X_{\mathbf{aa}'\oplus\mathbf{bb}'}\ x_{\mathbf{a}} \\
&= \sum_{\substack{\mathbf{a}'\in A^s\\ [\mathbf{A},\mathbf{a}']=[\mathbf{B},\mathbf{b}']}} \sum_{r=1}^{k} \sum_{\substack{\mathbf{a}\in A^r\\ |\mathbf{aa}'|\le k}} M_r^{[\mathbf{A},\mathbf{aa}']}\ x_{\mathbf{a}} \sum_{\substack{\mathbf{b}\in B^r\\ [\mathbf{A},\mathbf{aa}']=[\mathbf{B},\mathbf{bb}']}} X_{\mathbf{aa}'\oplus\mathbf{bb}'}.
\end{aligned}$$

In the last line we used the fact that the condition $[\mathbf{A},\mathbf{aa}'] = [\mathbf{B},\mathbf{bb}']$ implies $M_r^{[\mathbf{A},\mathbf{aa}']} = M_r^{[\mathbf{B},\mathbf{bb}']}$. Using again Lemma 7 the last expression becomes

$$\begin{aligned}
&\sum_{\substack{\mathbf{a}'\in A^s\\ [\mathbf{A},\mathbf{a}']=[\mathbf{B},\mathbf{b}']}} \sum_{r=1}^{k} \sum_{\substack{\mathbf{a}\in A^r\\ |\mathbf{aa}'|\le k}} M_r^{[\mathbf{A},\mathbf{aa}']}\ x_{\mathbf{a}}\ X_{\mathbf{a}'\oplus\mathbf{b}'} \\
&= \sum_{\substack{\mathbf{a}'\in A^s\\ [\mathbf{A},\mathbf{a}']=[\mathbf{B},\mathbf{b}']}} X_{\mathbf{a}'\oplus\mathbf{b}'} \sum_{r=1}^{k} \sum_{\substack{\mathbf{a}\in A^r\\ |\mathbf{aa}'|\le k}} M_r^{[\mathbf{A},\mathbf{aa}']}\ x_{\mathbf{a}} \\
&\le \sum_{\substack{\mathbf{a}'\in A^s\\ [\mathbf{A},\mathbf{a}']=[\mathbf{B},\mathbf{b}']}} X_{\mathbf{a}'\oplus\mathbf{b}'}\ d^{[\mathbf{A},\mathbf{a}']} = d^{[\mathbf{B},\mathbf{b}']}.
\end{aligned}$$

In the last line we used equation (40), together with the fact that the condition $[\mathbf{A},\mathbf{a}'] = [\mathbf{B},\mathbf{b}']$ implies $d^{[\mathbf{A},\mathbf{a}']} = d^{[\mathbf{B},\mathbf{b}']}$, another application of Lemma 7, and $X_\emptyset = 1$ by (9). This completes the proof of Theorem 2.

6 Applications to Logics with Counting

In this section we discuss the applications of the Transfer Lemma and the preservation of local linear programs to get new results on the expressive power of the logic with counting quantifiers.

6.1 Examples revisited

In the following, let MAX-FLOW denote the linear program for st-flows in st-networks as discussed in Section 5.2. Similarly, let BIPARTITE-MATCHING denote the linear program for matchings in bipartite graphs. For every integer $t \ge 1$, let VERTEX-COVER^t denote the t-th level of SA of the standard linear programming relaxation of vertex-cover, and let MAX-CUT^t denote the t-th level of SA of the metric polytope relaxation of max-cut.

For a local LP L, we say that L is *preserved* by an equivalence $\equiv$ if, whenever $\mathbf{A} \equiv \mathbf{B}$ and $L(\mathbf{A})$ has a solution, also $L(\mathbf{B})$ has a solution. More generally, if L is a local LP with an associated objective function max $\mathbf{c}^{\mathrm{T}}\mathbf{x}$ for which the constraint $\mathbf{c}^{\mathrm{T}}\mathbf{x} \ge W$ is also a local LP for every value W, then we say that *the optimum value of L is preserved by* $\equiv$ if the expanded local LP $L \cup \{\mathbf{c}^{\mathrm{T}}\mathbf{x} \ge W\}$ is preserved by $\equiv$ for every W.

The examples under consideration were all shown to be k-local LPs, for appropriate k, with the objective function incorporated as a constraint. Thus, a direct corollary to Theorem 2 is the following:

Corollary 1. *For every $t \geq 1$, the following hold:*

1. *the optimum value of* BIPARTITE-MATCHING *is preserved by* $\equiv^2_{\mathrm{SA}}$*,*
2. *the optimum value of* MAX-FLOW *is preserved by* $\equiv^2_{\mathrm{SA}}$*,*
3. *the optimum value of* $\mathsf{VERTEX\text{-}COVER}^t$ *is preserved by* $\equiv^{t+1}_{\mathrm{SA}}$*,*
4. *the optimum value of* $\mathsf{MAX\text{-}CUT}^t$ *is preserved by* $\equiv^{2t+1}_{\mathrm{SA}}$*.*

By the Transfer Lemma, the optimum values of these LPs are also preserved by $\equiv^3_{\mathrm{C}}$ in the first two cases, and by $\equiv^{t+2}_{\mathrm{C}}$ and $\equiv^{2t+2}_{\mathrm{C}}$ in the last two. This will be used in the sequel.

6.2 Definability results

One consequence of the fact that the optimum value of BIPARTITE-MATCHING is preserved by $\equiv^2_{\mathrm{SA}}$ and hence by $\equiv^3_{\mathrm{C}}$ is, for example, that the class of bipartite graphs that have a perfect matching is definable by a $C^3_{\infty\omega}$-sentence. Let us see why.

First, the preservation under $\equiv^3_{\mathrm{C}}$ means that the class of bipartite graphs that have a perfect matching is a union of $\equiv^3_{\mathrm{C}}$-equivalence classes of bipartite graphs. Now, it is a standard result in finite model theory that each $\equiv^3_{\mathrm{C}}$-equivalence class is definable by a $C^3_{\infty\omega}$-sentence. More precisely, what this means is that for every structure $\mathbf{A}$, there exists a sentence $\phi_{\mathbf{A}}$ of the logic $C^3_{\infty\omega}$ that is true precisely on the structures that are $\equiv^3_{\mathrm{C}}$-equivalent to $\mathbf{A}$ (see Lemma 1.39 in [32]). Thus, since $C^3_{\infty\omega}$ is closed under infinitary disjunctions, it suffices to take the disjunction of all the $\phi_{\mathbf{A}}$'s as $\mathbf{A}$ ranges over the bipartite graphs whose $\equiv^3_{\mathrm{C}}$-equivalence classes partition those that have a perfect matching.

It goes without saying that the same sort of argument carries over to MAX-FLOW in st-networks. For the following statement, let a *saturable network* be an st-network in which enough flow can be pushed through it to fill the capacity of all edges leaving the source.

Corollary 2. *The following hold:*

1. *the class of graphs that have a perfect matching is $C^3_{\infty\omega}$-definable on bipartite graphs,*
2. *the class of saturable networks is $C^3_{\infty\omega}$-definable on st-networks with unit capacities.*

We have chosen to state the result for saturable networks, but we are free to prefer the class of networks in which a 1/3-fraction, say, of the capacity leaving the source can be filled. Similarly, we are free to prefer the class of bipartite graphs that have a matching pairing a 2/3-fraction, say, of the vertices.

A less direct application of Corollary 1 concerns the max-cut problem on $\mathbf{K}_5$-minor free graphs. A non-trivial result in polyhedral combinatorics states that for graphs $\mathbf{G}$ that do not have $\mathbf{K}_5$ as a minor, optimizing over the projection of the metric polytope to the edges of $\mathbf{G}$ yields the integral optimal cut of $\mathbf{G}$ [6]. Since optimizing over the projection to the edges of $\mathbf{G}$ of the metric polytope is precisely what the linear program MAX-CUT is, we get the following consequence to Corollary 1 and the Transfer Lemma:

Corollary 3. *The class of graphs that have a partition that cuts at least half the edges is $C^4_{\infty\omega}$-definable on the class of $\mathbf{K}_5$-minor free graphs.*

Obviously, the choice to cut half the edges is arbitrary; a 1/3-fraction would work equally well. Let us note that from the results in [17] on counting logics being able to express all polynomial-time properties on classes of minor-free graphs, Corollary 3 would follow for $C^k_{\infty\omega}$ replacing $C^4_{\infty\omega}$ for *some* k (that is very likely big). This is because optimizing a linear function over the metric polytope can be done in polynomial time by linear programming. Our argument shows that $k = 4$ is enough. It is interesting that the two proofs are very different.

To conclude this section, we note for contrast that the linear programming formulation of the matching polytope for non-bipartite graphs due to Edmonds is not k-local for any constant k. This is because its inequalities are indexed by the odd-size subsets of vertices (see [29]). However, this does not rule out the possibility that it is nonetheless preserved by $\equiv^k_{\mathrm{SA}}$ for some constant k, and therefore that the class of general graphs having a perfect matching is definable in $C^k_{\infty\omega}$ for some constant k, but this stays open.

7 Applications to Sherali-Adams Rank

The goal in this section is to exploit the known lower-bound results in the context of counting logics to build explicit instances of the vertex-cover and max-cut problems that show that the SA-rank of their relaxations is $\Omega(n)$, where n is the number of vertices in the graphs. The known result that we will use is the celebrated construction of Cai, Fürer and Immerman [9] showing that there exist graphs with n vertices and bounded degree that cannot be distinguished by a formula of the counting logic C^k for $k = \Omega(n)$.

7.1 The Cai-Fürer-Immerman construction

Instead of indistinguishable pairs of graphs as in [9], it will be convenient to start with indistinguishable pairs of instances of a constraint satisfaction problem. We use systems of linear equations over GF(2) as in [3]. Let $\mathbf{H} = (V, E)$ be a 3-regular graph, and let $\mathbf{d} = (d_v : v \in V)$ be a GF(2)-labeling of its vertices. We call $\mathbf{d}$ odd if $\sum_{v \in V} d_v = 1 \mod 2$; otherwise we call it even. We build a system of linear equations $\mathbf{S}(\mathbf{H}, \mathbf{d})$ over GF(2) as follows. For every vertex $v \in V$ and every edge $e \in E$ that is incident on v, the system has two variables $x^{v,e}_0$ and $x^{v,e}_1$. For every vertex $v \in V$ with incident edges e_1, e_2 and e_3, the system includes the following eight equations, one for each choice of $i, j, k \in \{0, 1\}$:

$$x^{v,e_1}_i + x^{v,e_2}_j + x^{v,e_3}_k = d_v + i + j + k.$$

For every edge $e \in V$ with end-points u and v, the system includes the following four equations, one for each choice of $i, j \in \{0, 1\}$:

$$x^{u,e}_i + x^{v,e}_j = i + j.$$

We refer to equations of the first type as *vertex equations*, and to equations of the second type as *edge equations*.

These systems were introduced in [2] to re-interpret the construction due to Cai, Fürer, and Immerman in the context of constraint satisfaction problems. Interestingly, these systems are very closely related to the so-called Tseitin systems that appear so often in propositional proof complexity [43]. The following fact is straightforward:

Lemma 8. *The system* $\mathbf{S}(\mathbf{H}, \mathbf{d})$ *is satisfiable if and only if* $\mathbf{d}$ *is even.*

Proof. For the *if* direction, set $x_i^{v,e}$ to $d_v + i$. For the *only if* direction, consider the subsystem induced by the variables $x_0^{v,e}$. The left-hand side of all equations in this subsystem adds-up to 0, and the right-hand side adds-up to $\sum_{v \in V} d_v$. □

Systems of linear equations as above can be encoded as relational structures in several standard ways. In the one we use below, the universe of the structure is the set of variables, and the equations of the type $x_1 + \cdots + x_k = a$, where $x_1, \ldots, x_k$ are variables, $a \in \mathrm{GF}(2)$ and $k \geq 1$, are thought of as tuples of a k-ary relation $R_{k,a}$. Note that every such structure can also be viewed as a system of linear equations over GF(2) (perhaps with repeated variables in some equations, but this point is minor and not really relevant for what follows). From now on we identify linear systems of GF(2) with the relational structures that encode them in this way.

In this framework we can state the following result from [3] whose proof builds on the main result in [9].

Theorem 3 ([3])**.** *Let* $k \geq 1$ *be an integer, let* $\mathbf{H} = (V, E)$ *be a 3-regular graph, and let* $\mathbf{d}, \mathbf{d}' \in \{0,1\}^V$ *be labelings of its vertices. If the treewidth of* $\mathbf{H}$ *is more than* k*, then* $\mathbf{S}(\mathbf{H}, \mathbf{d}) \equiv_{\mathrm{C}}^{k} \mathbf{S}(\mathbf{H}, \mathbf{d}')$.

In particular, if $\mathbf{G}$ is a 3-regular expander graph with n vertices, then its treewidth is at least ϵn for some constant $\epsilon > 0$, and hence $\mathbf{S}(\mathbf{G}, \mathbf{d}) \equiv_{\mathrm{C}}^{\lfloor \epsilon n \rfloor} \mathbf{S}(\mathbf{G}, \mathbf{d}')$. If we take $\mathbf{d}$ an even labeling and $\mathbf{d}'$ an odd one, we get a pair of $\Omega(n)$-indistinguishable instances, one of which is satisfiable and the other is unsatisfiable. Our next goal is to turn this pair of instances into pairs of $\Omega(n)$-indistinguishable graphs, one of which has a small vertex cover or a large cut and the other does not.

7.2 Method of interpretations

The method of interpretations is the logic version of the concept of reducibility in computational complexity. Informally, we say that a structure $\mathbf{B}$ is interpretable in another structure $\mathbf{A}$, if the universe and the relations of $\mathbf{B}$ are *definable* by means of formulas interpreted in the structure $\mathbf{A}$. The important point of interpretations for us is that they propagate indistinguishability: if $\mathbf{B}$ and $\mathbf{B}'$ are interpretable in $\mathbf{A}$ and $\mathbf{A}'$, respectively, then $\mathbf{B}$ and $\mathbf{B}'$ stay indistinguishable if $\mathbf{A}$ and $\mathbf{A}'$ were.

Let us make this concept formal. Let Θ be a class of formulas. A Θ-interpretation of width w and p parameters is a sequence of Θ-formulas $I = (\varphi_0(\mathbf{x}_0; \mathbf{y}), \varphi_1(\mathbf{x}_1, \ldots, \mathbf{x}_{r_1}; \mathbf{y}), \ldots, \varphi_s(\mathbf{x}_1, \ldots, \mathbf{x}_{r_s}; \mathbf{y}))$, where each $\mathbf{x}_i$ is a w-tuple of distinct variables and $\mathbf{y}$ is a p-tuple of distinct variables. For a structure $\mathbf{A}$ and a p-tuple of parameters $\mathbf{c} = (c_1, \ldots, c_p) \in A^p$ such that $c_i \neq c_j$ for $i \neq j$, the outcome of the interpretation is the structure $I(\mathbf{A}, \mathbf{c})$ with universe $\{\mathbf{a} \in A^w : \mathbf{A} \models \varphi_0(\mathbf{a}; \mathbf{c})\}$ and relations $\{(\mathbf{a}_1, \ldots, \mathbf{a}_{r_i}) \in (A^w)^{r_i} : \mathbf{A} \models \varphi_i(\mathbf{a}_1, \ldots, \mathbf{a}_{r_i}; \mathbf{c})\}$ for $i = 1, \ldots, s$. Let $\mathbf{B}$ be a structure. We say that I interprets $\mathbf{B}$ in $\mathbf{A}$ if, for some and every p-tuple of parameters $\mathbf{c} = (c_1, \ldots, c_p) \in A^p$ such that $c_i \neq c_j$ for $i \neq j$, it holds that $I(\mathbf{A}, \mathbf{c})$ is isomorphic to $\mathbf{B}$.

As mentioned earlier, our interest in interpretations is the following well-known result. We give the straightforward proof for completeness.

Lemma 9. *Let* I *be a* C^k*-interpretation of width* w *and* p *parameters. Let* $\mathbf{A}$ *and* $\mathbf{A}'$ *be structures and let* $\mathbf{B}$ *and* $\mathbf{B}'$ *be structures interpreted by* I *in* $\mathbf{A}$ *and* $\mathbf{A}'$*, respectively. If* $\mathbf{A} \equiv_{\mathrm{C}}^{mw+k+p} \mathbf{A}'$*, then* $\mathbf{B} \equiv_{\mathrm{C}}^{m} \mathbf{B}'$.

Proof. Let $I = (\varphi_0, \varphi_1, \ldots, \varphi_s)$ be the C^k-formulas in the interpretation. We prove the contrapositive. Assume that $\mathbf{B} \not\equiv_C^m \mathbf{B}'$ and let ψ be a C^m-sentence such that $\mathbf{B} \models \psi$ and $\mathbf{B}' \not\models \psi$. Let $\{x_1, \ldots, x_m\}$ be the set of variables that appear in ψ, which we may assume disjoint from the set of variables in the formulas φ_i, and let $\mathbf{y} = (y_1, \ldots, y_p)$ be a set of p new variables. For every formula θ with all variables in $\{x_1, \ldots, x_m\} \cup \{y_1, \ldots, y_p\}$, let $T(\theta)$ be the result of replacing each occurrence of $R_i(x_{i_1}, \ldots, x_{i_{r_i}})$ in θ by $\varphi_i(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{r_i}}; \mathbf{y})$, each occurrence of $x_i = x_j$ in θ by $\mathbf{x}_i = \mathbf{x}_j$, and each quantifier $\exists^{\geq t} x_i \tau$ by[2] $\exists^{\geq t} \mathbf{x}_i (\varphi_0(\mathbf{x}_i; \mathbf{y}) \wedge T(\tau))$. In this transformation we use w new variables $\mathbf{x}_i$ for each variable x_i in ψ. Note that $T(\psi)$ carries $\mathbf{y}$ as free-variables, and that its total number of variables is $wm + k + p$.

Now we claim that the C^{wm+k+p}-sentence $\delta := \exists y_1 \cdots \exists y_p (\bigwedge_{i \neq j} y_i \neq y_j \wedge T(\psi))$ distinguishes $\mathbf{A}$ from $\mathbf{A}'$. Indeed, $\mathbf{B}$ is isomorphic to $I(\mathbf{A}, \mathbf{c})$ for some and every $\mathbf{c} = (c_1, \ldots, c_p) \in A^p$ with $c_i \neq c_j$ for $i \neq j$, and $T(\psi)$ is designed so that $\mathbf{A} \models T(\psi)(\mathbf{c})$ if and only if $I(\mathbf{A}, \mathbf{c}) \models \psi$. Similarly, $\mathbf{B}'$ is isomorphic to $I(\mathbf{A}', \mathbf{c})$ for some and every $\mathbf{c}' = (c'_1, \ldots, c'_p) \in A'^p$ with $c'_i \neq c'_j$ for $i \neq j$, and $T(\psi)$ is designed so that $\mathbf{A}' \not\models T(\psi)(\mathbf{c}')$ if and only if $I(\mathbf{A}', \mathbf{c}') \not\models \psi$. We conclude that $\mathbf{A} \models \delta$ and $\mathbf{A}' \not\models \delta$. □

7.3 Reductions to vertex-cover and max-cut

With Lemma 9 in hand, the only thing left to do is to apply the standard reductions from constraint satisfaction problems to vertex-cover and max-cut, and show that these reductions are indeed C^k-interpretations for some small k. In both cases we proceed by first reducing the constraint satisfaction problem to the independent set problem, and from there we branch into reductions to vertex-cover and max-cut.

The reduction from constraint satisfaction problems to the independent set problem is simple. For every constraint with variables $x_1, \ldots, x_k$, create as many vertices as there are assignments to these variables that satisfy the constraint. In the case of linear systems over GF(2), there are exactly 2^{k-1} assignments satisfying each equation on k distinct variables. Then add an edge between any two vertices that represent incompatible assignments: that is, assignments that give two different values to some variable. In particular, the cloud of vertices that corresponds to a particular constraint will always be a clique. For a system of linear equations $\mathbf{S}$, let $\mathbf{G}(\mathbf{S})$ be the resulting graph. The straightforward claim is that $\mathbf{S}$ is satisfiable if and only if $\mathbf{G}(\mathbf{S})$ contains an independent set of size m, where m is the number of constraints in $\mathbf{S}$.

Let us show that this construction is a C^{14}-interpretation of width 6 and 2 parameters. We define the formulas $\varphi_0(\mathbf{x}; \mathbf{p})$ for the set of vertices of $\mathbf{G}(\mathbf{S})$ and $\varphi_1(\mathbf{x}, \mathbf{y}; \mathbf{p})$ for the edge relation of $\mathbf{G}(\mathbf{S})$. We give these formulas right away and discuss the intuition behind them after their definition. In the following, $\mathbf{p}$ is a tuple of two parameter-variables (p_0, p_1), and $\mathbf{x}$ and $\mathbf{y}$ are two 6-tuples of variables $(x_1, x_2, x_3, x'_1, x'_2, x'_3)$ and $(y_1, y_2, y_3, y'_1, y'_2, y'_3)$. Thus, the set of vertices of the defined graph will be a subset of the 6-th power of $\mathbf{S}$. For $i \in \{0, 1\}$ and $k \geq 1$, let E_i^k be the set of k-tuples $\mathbf{a} = (a_1, \ldots, a_k) \in \{0, 1\}^k$ such that $a_1 + \cdots + a_k = i \mod 2$. The formula $\varphi_0(\mathbf{x}; \mathbf{p})$ is the disjunction of the following four formulas:

1. $R_{3,0}(x_1, x_2, x_3) \wedge \bigvee_{\mathbf{a} \in E_0^3} (x'_1 = p_{a_1} \wedge x'_2 = p_{a_2} \wedge x'_3 = p_{a_3})$,
2. $R_{3,1}(x_1, x_2, x_3) \wedge \bigvee_{\mathbf{a} \in E_1^3} (x'_1 = p_{a_1} \wedge x'_2 = p_{a_2} \wedge x'_3 = p_{a_3})$,
3. $R_{2,0}(x_1, x_2) \wedge x_3 = p_0 \wedge \bigvee_{\mathbf{a} \in E_0^2} (x'_1 = p_{a_1} \wedge x'_2 = p_{a_2}) \wedge x'_3 = p_0$.

[2]Here we use the fact that C^k can simulate counting quantifiers over tuples. See Lemma 4.9 in [32] for details.

4. $R_{2,1}(x_1, x_2) \wedge x_3 = p_0 \wedge \bigvee_{\mathbf{a} \in E_1^2} (x'_1 = p_{a_1} \wedge x'_2 = p_{a_2}) \wedge x'_3 = p_0$.

The formula $\varphi_1(\mathbf{x}, \mathbf{y}; \mathbf{p})$ is the disjunction of the following formulas:

1. $\bigvee_{i=0}^{1} R_{3,i}(x_1, x_2, x_3) \wedge \bigvee_{i=0}^{1} R_{3,0}(y_1, y_2, y_3) \wedge \bigvee_{i=1}^{3} \bigvee_{j=1}^{3} (x_i = y_j \wedge x'_i \neq y'_j)$,
2. $\bigvee_{i=0}^{1} R_{3,i}(x_1, x_2, x_3) \wedge \bigvee_{i=0}^{1} R_{2,i}(y_1, y_2) \wedge \bigvee_{i=1}^{3} \bigvee_{j=1}^{2} (x_i = y_j \wedge x'_i \neq y'_j)$,
3. $\bigvee_{i=0}^{1} R_{2,i}(x_1, x_2) \wedge \bigvee_{i=0}^{1} R_{2,i}(y_1, y_2) \wedge \bigvee_{i=1}^{2} \bigvee_{j=1}^{2} (x_i = y_j \wedge x'_i \neq y'_j)$.

For an intuition, the 6-tuple $(x_1, x_2, x_3, x'_1, x'_2, x'_3)$ encodes the vertex of $\mathbf{G}(\mathbf{S})$ that corresponds to the equation that has x_1, x_2, x_3 as variables, and x'_1, x'_2, x'_3 as a satisfying assignment. We use the parameters p_0, p_1 to encode both truth values and placeholders, as in the case of clauses 3 and 4 in the definition of φ_0 in which x_3 and x'_3 should not take any particular value. With this explanation, the first clause in the definition of φ_1 could read: "x_1, x_2, x_3 and y_1, y_2, y_3 appear together in two different equations, and happen to share a variable that gets inconsistent values in their respective assignments x'_1, x'_2, x'_3 and y'_1, y'_2, y'_3".

The claim that these formulas define $\mathbf{G}(\mathbf{S})$ is formalized as follows:

Claim 17. *Let $\mathbf{S}$ be a linear system over* GF(2) *with at least two variables, and let $\mathbf{c} = (c_0, c_1)$ be any two of these variables. Then $I(\mathbf{S}, \mathbf{c})$ is a graph isomorphic to $\mathbf{G}(\mathbf{S})$.*

At this point we have a reduction from systems of linear equations over GF(2) to the independent set problem. Next we argue the reduction from this problem to vertex-cover and max-cut.

For vertex-cover there is not much more left to do: the same graph $\mathbf{G}(\mathbf{S})$ will do the job. Let us compute the size of the smallest vertex-covers in $\mathbf{G}(\mathbf{S})$ and in $\mathbf{G}(\mathbf{S}')$, when $\mathbf{S} = \mathbf{S}(\mathbf{H}, \mathbf{d})$ for an even $\mathbf{d}$, and $\mathbf{S}' = \mathbf{S}(\mathbf{H}, \mathbf{d}')$ for an odd $\mathbf{d}'$. Let n be the number of vertices of $\mathbf{H}$. Since $\mathbf{H}$ is 3-regular, the number of edges is $3n/2$. The number of variables in the systems $\mathbf{S}$ and $\mathbf{S}'$ is $14n$ since there are eight vertex-equations for every vertex, and four edge-equations for every edge. Both graphs $\mathbf{G}(\mathbf{S})$ and $\mathbf{G}(\mathbf{S}')$ have $44n$ vertices since each of the $8n$ vertex-equations contributes four vertices, and each of the $6n$ edge-equations contributes two vertices. Since $\mathbf{S}$ is satisfiable, the graph $\mathbf{G}(\mathbf{S})$ has an independent set of size $14n$, and therefore a vertex-cover of size $30n$. On the other hand, since $\mathbf{S}'$ is unsatisfiable, the graph $\mathbf{G}(\mathbf{S})$ does not have an independent set of size $14n$, and therefore it does not have a vertex-cover of size $30n$. Let us note that $\mathbf{G}(\mathbf{S})$ and $\mathbf{G}(\mathbf{S}')$ both have degree at most 11.

If we write $\mathsf{VERTEX\text{-}COVER}(\mathbf{G})$ for the optimum value of $\mathsf{VERTEX\text{-}COVER}$ applied to graph $\mathbf{G}$, we have established the following:

Lemma 10. *There exist constants $\epsilon > 0$ and $c, d > 0$ such that for every large enough even n, there exist graphs $\mathbf{G}_0$ and $\mathbf{G}_1$, with at most cn vertices and dn edges each and such that:*

1. $\mathsf{VERTEX\text{-}COVER}(\mathbf{G}_0) \neq \mathsf{VERTEX\text{-}COVER}(\mathbf{G}_1)$,
2. $\mathbf{G}_0 \equiv_{\mathrm{C}}^{\lfloor \epsilon n \rfloor} \mathbf{G}_1$.

Next we give the reduction from the independent set problem to max-cut. This will require a small gadget construction that we found in [39]; other local reductions could work as well.

Starting at a graph $\mathbf{G} = (V, E)$, we add one new vertex v_0 and edges $\{v_0, v\}$ for every vertex v in $\mathbf{G}$. Next, each edge $\{u, v\}$ in $\mathbf{G}$ is replaced by a gadget that introduces two new vertices that we call (u, v) and (v, u). The gadget corresponding to edge $\{u, v\}$ connects (u, v) to u and (v, u)

to v, and both (u,v) and (v,u) to v_0 and to each other. Let $\mathbf{G}' = (V', E')$ be the resulting graph, which has $|V| + 2|E| + 1$ vertices and $|V| + 5|E|$ edges. The claim is that $\mathbf{G}$ has an independent set of size m if and only if $\mathbf{G}'$ has a partition that cuts at least $m + 4|E|$ edges. Indeed, if $A \subseteq V$ is an independent set of size m in $\mathbf{G}$, then the number of edges of $\mathbf{G}'$ with exactly one end-point in $A \cup \{(u,v) : \{u,v\} \in E,\ u \notin A\} \subseteq V'$ is at least $m + 4|E|$. Conversely, if $\mathbf{G}'$ has a partition that cuts at least $m + 4|E|$ edges of $\mathbf{G}'$, then the set of vertices in V that are on the side opposite to v_0 can be made into an independent set of size m in $\mathbf{G}$ by arbitrarily removing one vertex from each violated edge.

Showing that this construction is a C^k-interpretation for a small k is an easy exercise. If original vertices u in V are represented by triples (u, u, p_0), for a parameter p_0, and the new vertices (u, v) for $\{u, v\}$ in E are represented by triples (u, v, p_0), for the same parameter p_0, then we can represent the new vertex v_0 by the triple (p_1, p_1, p_1), for another parameter $p_1 \neq p_0$. The edge relation is then defined by a straightforward quantifier-free formula. The result is a C^3-interpretation of width 3 with 2 parameters.

If we write $\mathsf{MAX\text{-}CUT}(\mathbf{G})$ for the optimum value of $\mathsf{MAX\text{-}CUT}$ applied to graph $\mathbf{G}$, this shows:

Lemma 11. *There exist constants $\epsilon > 0$ and $c, d > 0$ such that for every large enough even n, there exist graphs $\mathbf{G}_0$ and $\mathbf{G}_1$, with at most cn vertices and dn edges each such that:*

1. $\mathsf{MAX\text{-}CUT}(\mathbf{G}_0) \neq \mathsf{MAX\text{-}CUT}(\mathbf{G}_1)$,
2. $\mathbf{G}_0 \equiv_{\mathrm{C}}^{\lfloor \epsilon n \rfloor} \mathbf{G}_1$.

And finally this gives the promised lower-bound on the SA-rank of the linear programming relaxations of the integer programs for max-cut and vertex-cover. In both cases take $\mathbf{G}_0$ and $\mathbf{G}_1$ as in the lemmas. For $k = \lfloor \epsilon n \rfloor$, consider the k-th SA-levels P_0^k and P_1^k of the linear programming relaxation of the corresponding integer programs for $\mathbf{G}_0$ and $\mathbf{G}_1$. Since $\mathbf{G}_0 \equiv_{\mathrm{C}}^{k} \mathbf{G}_1$, we have $\mathbf{G}_0 \equiv_{\mathrm{SA}}^{k-1} \mathbf{G}_1$ by the Transfer Lemma. From Corollary 1, the optimum values over $P_0^{k'}$ and $P_1^{k'}$ are the same for $k' = k - 2$ in one case and $k' = \lfloor (k-2)/2 \rfloor$ in the other. However, the optimum values over $P_0^{\mathbb{Z}}$ and $P_1^{\mathbb{Z}}$ are not the same since $\mathsf{VERTEX\text{-}COVER}(\mathbf{G}_0) \neq \mathsf{VERTEX\text{-}COVER}(\mathbf{G}_1)$ and $\mathsf{MAX\text{-}CUT}(\mathbf{G}_0) \neq \mathsf{MAX\text{-}CUT}(\mathbf{G}_1)$. Thus, the SA-rank is at least $k' + 1$ in both cases, which is $\Omega(n)$.

8 Discussion and Open Problems

Isomorphism is in a sense the finest of all binary relations on finite structures. There are other interesting relations between structures, such as embeddings and homomorphisms, that could well be phrased as 0-1 linear programs and then appropriately relaxed. The SA-levels of these relaxations would then yield tighter and tighter approximations. On the combinatorial side, embeddings and homomorphisms also admit relaxations through corresponding pebble games with modified winning conditions. In the case of homomorphisms, this is the existential k-pebble game popularized by Kolaitis and Vardi in the context of constraint satisfaction problems [23]. Does a version of the Transfer Lemma apply in this case too? While it would not be hard to establish one of the directions (namely, from solutions to the linear program to winning strategies in the game), and a version of this was actually anticipated in [4], it seems that the lack of counting in the homomorphism game could be a serious obstacle at establishing the other. One could try to add some counting mechanism to the game, but that would require a definition that mixes well with the concept of homomorphism, which looks challenging.

On a different line of thought, perhaps the most promising outcome of the main result in this paper is the connection it establishes between polyhedral combinatorics and finite model theory. In Sections 6 and 7 we have shown how rather elementary arguments are able to exploit the knowledge in one field to get results in the other. We hope that more sophisticated arguments could lead to stronger results. Let us point out two interesting possibilities.

In the direction from polyhedral combinatorics to finite model theory, it would be interesting to exploit the sophisticated constructions of integrality gap instances in the world of lift-and-project methods. One of the admitted bottlenecks of the pebble-game technique for proving inexpressibility results is the lack of general methods for building pairs of structures with different properties that stay sufficiently indistinguishable. Perhaps the methods for building integrality gap instances, say as in [10] through metric-embedding arguments from functional analysis, could be of use for building such objects. A concrete example where this could be applied is to the problem of perfect matchings on general graphs. In short, the question reduces to building, for every constant $k \geq 2$, a pair of $\equiv^k_{\mathrm{C}}$-equivalent graphs $\mathbf{G}_0$ and $\mathbf{G}_1$ in which $\mathbf{G}_0$ has a perfect matching but $\mathbf{G}_1$ does not. This would show that the class of general graphs having a perfect matching is not definable in the logic $C^{\omega}_{\infty\omega}$, thus solving a problem in [8]. The recent progress in understanding the SA-levels of the matching polytope could perhaps be also useful here [31].

In the direction from finite model theory to polyhedral combinatorics, new results could follow if the indistinguishability result in [9] were strengthened to a pair of indistinguishable instances of the unique-games problem with a large gap in their optimal values. With such a lower bound in hand, it would be conceivable that the flexibility that the logic approach gives at handling reductions could yield new and optimal integrality gap instances. For example, one would likely be able to exploit the reductions from unique-games to vertex-cover in [21] to get instances where an optimal integrality gap of 2 could resist up to $\Omega(n)$ levels of SA. At any rate, exploring the gap-creating reductions underlying the proofs of the PCP-theorem and the applications of the unique-games conjecture in the context of finite model theory appears to be an attractive line of research worth pursuing in itself.

Acknowledgments.

We are grateful to Grant Schoenebeck for discussions on what is known about the integrality gap instances for vertex-cover and max-cut in lift-and-project hierarchies. We are also grateful to Martin Grohe for references on the Weisfeiler-Lehman algorithm. Both authors partially supported by CICYT TIN2010-20967-C04-04 (TASSAT). Second author supported in part by MICINN Ramon y Cajal.

References

[1] S. Arora, B. Bollobás, L. Lovász, and I. Tourlakis. Proving integrality gaps without knowing the linear program. *Theory of Computing*, 2(1):19–51, 2006.

[2] A. Atserias, A. Bulatov, and A. Dawar. Affine systems of equations and counting infinitary logic. In *34th International Colloquium on Automata, Languages and Programming (ICALP)*, Lecture Notes in Computer Science, 2007.

[3] A. Atserias, A. A. Bulatov, and A. Dawar. Affine systems of equations and counting infinitary logic. *Theoretical Computer Science*, 410(18):1666–1683, 2009.

[4] A. Atserias, Ph. G. Kolaitis, and M. Y. Vardi. Linear programming and the existential pebble game. Unpublished, 2006.

[5] L. Babai, P. Erdös, and S. M. Selkow. Random graph isomorphism. *SIAM Journal on Computing*, 9:628–635, 1980.

[6] F. Barahona and A.R. Mahjoub. On the cut polytope. *Mathematical Programming*, 36:157–173, 1986.

[7] J. Barwise. On Moschovakis closure ordinals. *Journal of Symbolic Logic*, 42(2):292–296, 1977.

[8] A. Blass, Y. Gurevich, and S. Shelah. On polynomial time computation over unordered structures. *Journal of Symbolic Logic*, 67(3):1093–1125, 2002.

[9] J. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12:389–410, 1992.

[10] M. Charikar, K. Makarychev, and Y. Makarychev. Integrality gaps for Sherali-Adams relaxations. In *41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 283–292, 2009.

[11] H. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, 1995.

[12] A. Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49:129–141, 1961.

[13] R. Fraïssé. Sur une nouvelle classification des systèmes de relations. *Comptes Rendus*, 230:1022–1024, 1950.

[14] K. Georgiou, A. Magen, T. Pitassi, and I. Tourlakis. Integrality gaps of 2 - o(1) for Vertex Cover SDPs in the Lovász-Schrijver Hierarchy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 702–712, 2007.

[15] M. Grohe. Fixed-point logics on planar graphs. In *Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 6–15, 1998.

[16] M. Grohe. Isomorphism testing for embeddable graphs through definability. In *32nd Annual ACM Symposium on the Theory of Computing (STOC)*, 2000.

[17] M. Grohe. Fixed-point definability and polynomial time on graphs with excluded minors. In *25th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 179–188, 2010.

[18] N. Immerman. Upper and lower bounds for first order expressibility. *Journal of Computer and System Sciences*, 25:76–98, 1982.

[19] N. Immerman. *Descriptive Complexity*. Springer-Verlag, 1999.

[20] N. Immerman and E. Lander. *Describing Graphs: A First-Order Approach to Graph Canonization*, pages 59–81. Springer-Verlag, 1990.

[21] S. Khot and O. Regev. Vertex cover might be hard to approximate to within 2-epsilon. *Journal of Computer and System Sciences*, 74(3):335–349, 2008.

[22] Ph. G. Kolaitis and J. A. Väänänen. Generalized quantifiers and pebble games on finite structures. *Annals of Pure Applied Logic*, 74(1):23–75, 1995.

[23] Ph. G. Kolaitis and M. Y. Vardi. Conjunctive-query containment and constraint satisfaction. *Journal of Computer and System Sciences*, 61(2):302–332, 2000.

[24] L. Kucera. Canonical labeling of regular graphs in linear average time. In *28th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 271–279, 1987.

[25] J. B. Lasserre. Global optimization with polynomials and the problems of moments. *SIAM Journal on Optimization*, 11:296–317, 2011.

[26] M. Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserr e relaxations for 0-1 programming. *Mathematics of Operations Research*, 28:470–496, 2001.

[27] M. Laurent. Lower bound for the number of iterations in semidefinite hierarchies for the cut polytope. *Mathematics of Operations Research*, 28(4):871–883, 2003.

[28] L. Libkin. *Elements of Finite Model Theory.* Springer-Verlag, 2004.

[29] L. Lovász and M. D. Plummer. *Matching Theory*, volume 367. AMS Chelsea Publishing, 2009.

[30] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1:166–190, 1991.

[31] C. Mathieu and A. Sinclair. Sherali-Adams relaxations of the matching polytope. In *41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 293–302, 2009.

[32] M. Otto. *Bounded variable logics and counting – A study in finite models*, volume 9. Springer-Verlag, 1997.

[33] O. Pikhurko and O. Verbitsky. Logical complexity of graphs: a survey. 55pp.arXiv:1003.4865v3 [math.CO], accepted for publication in "Model Theoretic Methods in Finite Combinatorics", M.Grohe and J.Makowsky (editors), AMS Comtemporary Mathematics Series, 2011.

[34] M. V. Ramana, E. R. Scheinerman, and D. Ullman. Fractional isomorphism of graphs. *Discrete Mathematics*, 132(1–3):247–265, 1994.

[35] E. R. Scheinerman and D. H. Ullman. *Fractional Graph Theory.* Wiley-Interscience, 1992.

[36] G. Schoenebeck. Linear Level Lasserre Lower Bounds for Certain k-CSPs. In *49th Annual IEEE Symposium on Foundations of Computer Science (F OCS)*, pages 593–602, 2008.

[37] G. Schoenebeck, L. Trevisan, and M. Tulsiani. Tight integrality gaps for Lovasz-Schrijver LP relaxations of vertex cover and max cut. In *39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 302–310, 2007.

[38] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990.

[39] Matt Stallmann. Lecture Notes on Design and Analysis of Algorithms. North Carolina State University.

[40] T. Stephen and L. Tunçel. On a representation of the matching polytope via semi-definite liftings. *Mathematics of Operations Research*, 24:1–7, 1999.

[41] I. Tourlakis. New lower bounds for vertex cover in the lovasz-schrijver hierarchy. In *21st Annual IEEE Conference on Computational Complexity (CCC)*, pages 170–182, 2006.

[42] M. Tulsiani. CSP gaps and reductions in the Lasserre hierarchy. In *41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 303–312, 2009.

[43] A. Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, 1987.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de