Definable Ellipsoid Method, Sums-of-Squares Proofs, and the Graph Isomorphism Problem

Albert Atserias
Universitat Politècnica de Catalunya
Barcelona, Catalonia, Spain

Joanna Fijalkow
University of Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800
F-33400 Talence, France, and
Institute of Informatics, University of Warsaw
Warsaw, Poland

Abstract

The ellipsoid method is an algorithm that solves the (weak) feasibility and linear optimization problems for convex sets by making oracle calls to their (weak) separation problem. We observe that the previously known method for showing that this reduction can be done in fixed-point logic with counting (FPC) for linear and semidefinite programs applies to any family of explicitly-bounded convex sets. We further show that the exact feasibility problem for semidefinite programs is expressible in the infinitary version of FPC. As a corollary we get that, for the graph isomorphism problem, the Lasserre/Sums-of-Squares semidefinite programming hierarchy of relaxations collapses to the Sherali-Adams linear programming hierarchy, up to a small loss in the degree.

1 Introduction

Besides being the first algorithm to be discovered that could solve linear programs (LPs) in polynomial time, the ellipsoid method has at least two other features that make it an important tool for the computer science theoretician. The first is that it is able to handle implicit LPs given by exponentially many, or even infinitely many, linear inequalities. These include some of the most fundamental problems of combinatorial optimization and mathematical programming, such as the weighted matching problem on general graphs, the submodular function minimization problem, or approximately solving semidefinite programs. The second important feature of the ellipsoid method is that, for LPs, its running time is polynomial in the bit length of its input and is provably robust against issues of numerical instability (see, e.g., [16]).

There is a third emerging feature of the ellipsoid method that is of particular significance for the logician and the descriptive complexity theorist. The starting point is the important breakthrough result of Anderson, Dawar and Holm [2] who developed a method called *folding* to deal with symmetries in an LP. They used this method to show that, for the special case of LPs, the ellipsoid method can be implemented in fixed-point logic with counting (FPC), and hence in polynomial time, but *choicelessly*, i.e., in a way that the symmetries of the input are respected all along the computation, and in the output. As the main application of their result, they proved that the class of graphs that have a perfect matching could be defined in FPC, thus solving one of the open problems raised by Blass, Gurevich and Shelah in their work on Choiceless Polynomial Time [9]. The method of folding was extended further by Dawar and Wang to deal with explicitly-bounded and full-dimensional semidefinite programs (SDPs) [11].

Our first contribution is the observation that the method of folding can be used to capture the power of the ellipsoid method in its full strength. We observe that the fully general polynomial-time reduction that solves the weak feasibility problem given a weak separation oracle for an explicitly-bounded convex set can be implemented, choicelessly, in FPC. As in the earlier works that employed the folding method, our implementation uses the reduction algorithm as described in [16] as a black-box. The black-box is made into a choiceless procedure through a sequence of runs of the algorithm along a refining sequence of suitable quotients of the given convex set. It should be pointed out that while all the main ideas for doing this were already implicit in the earlier works by Anderson, Dawar and Holm, and by Dawar and Wang, working out the details requires a certain degree of care. For example, when we started this work it was not clear whether the earlier methods would be able to deal with separation oracles for families of convex sets that are *not* closed under the folding-quotient operations. We observe that such closure conditions, which happen to hold for LPs and SDPs, are not required. The details of this can be found in Section 3.

Aided by this new understanding, we develop three applications of folding.

1.1 The SDP exact feasibility problem

The first application concerns the semidefinite programming exact feasibility problem. A semidefinite set, also known as a spectrahedron, is a subset of Euclidean space that is defined as the intersection of the cone of positive semidefinite matrices with an affine subspace. Thus, semidefinite sets are the feasible regions of SDPs. The SDP exact feasibility problem asks, for an SDP given as input, whether its feasible region is non-empty. While the approximate and explicitly-bounded version of this problem is solvable in polynomial-time by the ellipsoid method, the computational complexity of *exact* feasibility is a well-known open problem in mathematical programming: it is decidable in polynomial space, by reduction to the existential theory of the reals, but its precise position in the complexity hierarchy is unknown. It has been shown that the problem is at least as hard as PosSLP, the positivity problem for integers represented as arithmetic circuits [30], and hence at least as hard as the famous square-root sum problem, but the exact complexity of these two problems is also largely unknown (see [1]).

Our result on the SDP exact feasibility problem is that, when its input is represented suitably as a finite structure, it is definable in the logic $\mathrm{C}^{\omega}_{\infty\omega}$, i.e., bounded-variable infinitary logic with counting (see Section 2 for definitions and references for all logics appearing in this paper). In more recent terminology, we say that the SDP exact feasibility problem has *bounded counting width*: there is a fixed bound k so that the set of YES (and NO) instances of the problem is closed under indistinguishability by formulas of k-variable counting logic. Let us briefly discuss the new idea that goes into proving this.

First we show that the FPC-definability of the ellipsoid method can be combined with the techniques in [11] to give an FPC-formula ϕ that solves the weak feasibility problem for explicitly-bounded SDPs. This deviates from the result in [11] in that it removes one of their two assumptions: the full-dimensionality requirement is now dropped. Then we show how to reduce the exact feasibility problem for arbitrary SDPs to the weak feasibility problem for explicitly-bounded SDPs, and do so in bounded-variable infinitary logic. What drives this reduction is the observation that an arbitrary SDP is feasible if, and only if, there exists a large radius $R{>}0$ such that, for every small tolerance $\epsilon{>}0$, an ϵ-perturbation of the constraints of the original SDP restricted to solutions of magnitude at most R is non-empty. Verifying this last condition when an (R,ϵ)-pair is given in the input can be done in FPC through the formula ϕ: indeed, the resulting SDP is explicitly-bounded thanks to R, and it is enough to decide its weak feasibility thanks to ϵ. Hence, the reduction boils down to handling the $\exists R{>}0\ \forall\epsilon{>}0$ quantification in bounded-variable infinitary logic. To achieve this, the key observation is that the part of the input that corresponds to an (R,ϵ)-pair is independent of the original SDP. This allows us to construct a *Booleanized* version $\phi_{R,\epsilon}$ of the FPC-formula ϕ that works only for the fixed (R,ϵ)-pair. Finally, by replacing the $\exists R{>}0\ \forall\epsilon{>}0$ quantification by an infinite disjunction and conjunction, respectively, we obtain the $\mathrm{C}^{\omega}_{\infty\omega}$ formula $\bigvee_{R>0}\bigwedge_{\epsilon>0}\phi_{R,\epsilon}$. We analyze the exact form of ϕ and show that it allows for the operation of fixing R and ϵ while retaining the same number of variables. This is the subject of Section 4.

1.2 The SOS proof-existence problem

A Sums-of-Squares (SOS) proof that an n-variable polynomial inequality $p_0 \geq 0$ holds on a set defined by the polynomial constraints $p_1 \geq 0, \ldots, p_m \geq 0$ is an identity of the form $\sum_{j=1}^{m} p_j s_j + s_0 = p_0$, where each polynomial s_j is a sum of squares of polynomials. The sums-of-squares methodology for solving polynomial optimization problems advocated by Lasserre [20] and Parrilo [26] motivates the question of computing such proofs, when they exist. It is well-known that, in many settings, including in the case of polynomial inequalities over Boolean variables, the search-space of SOS proofs with polynomials bounded by degree d can be formulated as the feasible region of an SDP with $m \cdot n^{O(d)}$ variables and constraints, and small coefficients. This leads to a computational approach to finding low-degree SOS proofs by reduction to the SDP exact feasibility problem. It should be noted that a naive application of this method does *not* in general yield algorithms that are polynomial in n and m even for $d = O(1)$ due to results in [22] proving that SOS proofs suffer from blow-up phenomena in the coefficients of their polynomials. Remarkably, it was later shown in [27]

that in certain special cases of the problem the blow-up phenomena do not appear.

We recall the SDP that describes the search-space of low-degree SOS proofs and note that its representation as a finite relational structure is computable in the logic FPC from a natural representation of the input polynomials $p_0, p_1, \ldots, p_m$. Together with the definability of the SDP exact feasibility problem, this implies that the SOS proof system over Boolean variables is *weakly degree-automatable* in the logic $\mathrm{C}^{\omega}_{\infty\omega}$ in the following sense: there is a constant c such that, for each degree d, there is a formula ϕ_d of the logic $\mathrm{C}^{cd}_{\infty\omega}$ that tells whether a given polynomial inequality $p_0 \geq 0$ has a degree-d SOS proof from a given system $p_1 \geq 0, \ldots, p_m \geq 0$ of polynomial constraints over Boolean variables. The qualification *weakly* in degree-automatable distinguishes the problem from its search version in which an actual degree-d proof is sought. For refutations, where p_0 is the constant -1 polynomial, we note that the proof-existence problem can also be reduced, in FPC, to the weak feasibility problem for arbitrary SDPs (not necessarily explicitly-bounded). While less demanding, this weaker form of the problem is not known to be solvable in polynomial time, let alone FPC definable. All this can be found in Section 5.

While interesting in its own right for its potential applications to proof complexity lower bounds along the lines of [14], we think of the weak degree-automatability result for SOS proofs as the required tool to develop the third and main application.

1.3 Hierarchies for the graph isomorphism problem

A variety of mathematical programming relaxations of the graph isomorphism problem have been proposed in the literature: the fractional isomorphism relaxation of Tinhofer [31], its strengthening via the Sherali-Adams hierarchy of LP relaxations [3, 21], its further strengthening via the Lasserre hierarchy of SDP relaxations [24], its relaxation via Groebner basis computations [8], and a few others. While it is known that no fixed level of any of these hierarchies of LP, SDP or Groebner-based relaxations solves the graph isomorphism problem [3, 21, 24, 8], their relative strength was not fully understood before our work. Since SDP is a proper generalization of LP, one may be tempted to guess that the Lasserre SDP hierarchy could perhaps distinguish more graphs than its Sherali-Adams LP sibling. Our main contribution is to prove that this is not the case: for the graph isomorphism problem, the strength of the Lasserre hierarchy collapses to that of the Sherali-Adams hierarchy.

Concretely, we prove in Section 6 that there exists a constant c such that if two graphs are distinguishable at level d of the Lasserre hierarchy, then they are also distinguishable at level cd of the Sherali-Adams hierarchy. The constant c loss comes from the number of variables that are needed to express the SDP exact feasibility problem in bounded-variable infinitary logic with counting. This collapse may sound surprising because it implies that, for distinguishing graphs, the spectral methods that underlie the Lasserre hierarchy are already available in low levels of the Sherali-Adams hierarchy. However, it agrees nicely with the fact that indistinguishability by 3-variable counting logic captures graph spectra [10] and the correspondence between k-variable counting logic and level k of the Sherali-Adams hierarchy from [3]. It also aligns well with the results in [23] where it is shown that certain spectral methods for approximating the number of constraints that can be satisfied in a constraint

satisfaction problem can be implemented directly in the Sherali-Adams hierarchy.

To get the collapse result we consider the standard 0-1 quadratic programming formulation $P(G, H)$ of the graph isomorphism problem for graphs G and H on disjoint sets of vertices. Assuming that the level-d Lasserre relaxation of $P(G, H)$ distinguishes G and H, our new insights on the expressibility of the SDP exact feasibility problem imply that G and H can be distinguished by a $\mathrm{C}^{cd}_{\infty\omega}$-sentence, where c is a constant independent of d. Hence, by the main result in [3] relating the levels of the Sherali-Adams hierarchy with indistinguishability in bounded-variable counting logic, the graphs G and H can be distinguished by level-cd Sherali-Adams relaxation, thus proving the collapse. It should be noted that, remarkably, this holds for any two graphs and *any* d, even if $d = d(n)$ is an arbitrary function of the number n of vertices of G and H. The details of this can be found in Section 6.

When stated in the language of proofs, the collapse has another interesting consequence. By combining the results in [3] and [8], it was already known that if there is a degree-d Sherali-Adams (SA) proof that G and H are not isomorphic, then there is also a degree-d monomial Polynomial Calculus (mon-PC) proof over the reals, hence also a degree-d Polynomial Calculus (PC) proof over the reals, which implies that there is a degree-$2d$ SOS proof by [7]. Thus, for the graph isomorphism problem, our collapse result completes a full cycle of simulations $\mathrm{SOS}_d \to \mathrm{SA}_{cd} \to \text{mon-PC}_{cd} \to \mathrm{PC}_{cd} \to \mathrm{SOS}_{2cd}$ to show that all these proof systems are equally powerful up to a $2c$-factor loss in the degree. It also confirms the belief expressed in [8] that the gap between PC and monomial PC is not large (a result obtained independently in [14]).

It is remarkable that we proved these statements about the relative strength of proof systems and hierarchies through an excursion into the descriptive complexity of the ellipsoid method, the SDP exact feasibility problem, and bounded-variable infinitary logics. However, it should be noted that our proof is indirect as it relies on the correspondence between k-variable counting logic and the k-th level Sherali-Adams hierarchy from [3]. The question whether the collapses can be shown to hold *directly* by strengthening LP-solutions to SDP-ones for the primals, or by relaxing SDP-solutions to LP-ones for the duals, remains an interesting one.

2 Preliminaries

We use $[n]$ to denote the set $\{1, \ldots, n\}$.

Vectors and matrices If I is a non-empty index set, then an I-vector is an element of $\mathbb{R}^I$. The components of $u \in \mathbb{R}^I$ are written $u(i)$ or u_i, for $i \in I$. We identify $\mathbb{R}^n$ with $\mathbb{R}^{[n]}$. For I-vectors u and v, the *inner product* of u and v is $\langle u, v\rangle = \sum_{i\in I} u_i v_i$. We write $\|u\|_1 = \sum_{i\in I} |u_i|$ for the L_1-norm, $\|u\|_2 = \sqrt{\langle u, u\rangle}$ for the L_2-norm, and $\|u\|_\infty = \max\{|u_i| : i \in I\}$ for the L_∞-norm. For $K \subseteq \mathbb{R}^I$ and $\delta > 0$, we define the *δ-ball* around K by $S(K, \delta) := \{x \in \mathbb{R}^I : \|x - y\|_2 \leq \delta \text{ for some } y \in K\}$. For $K = \{x\}$, we set $S(x, \delta) := S(\{x\}, \delta)$. We define also $S(K, -\delta) := \{x \in \mathbb{R}^I : S(x, \delta) \subseteq K\}$. When we refer to the *volume* of a subset K of Euclidean space $\mathbb{R}^I$, we assume that K is Lebesgue measurable and that the volume is

defined as its Lebesgue measure (see, e.g., [28]). In particular, the volume of a 1-ball in the n-dimensional real vector space is $V_n = \pi^{n/2}/\Gamma(n/2+1)$, where Γ is the gamma function, i.e., the standard continuous extension of the factorial function.

If I and J are two non-empty index sets, then an $I \times J$-matrix is simply an $I \times J$-vector; i.e., an element of $\mathbb{R}^{I\times J}$. Accordingly, the components of $X \in \mathbb{R}^{I\times J}$ are written $X(i,j)$, or $X_{i,j}$, or X_{ij}. The L_1-, L_2- and L_∞-norms of a matrix $X \in \mathbb{R}^{I\times J}$ are defined as the respective norms of X seen as an $I \times J$-vector, and the inner product of the matrices $X, Y \in \mathbb{R}^{I\times J}$ is $\langle X, Y\rangle = \sum_{i\in I}\sum_{j\in J} X_{ij}Y_{ij}$. Matrix product is written by concatenation. A square matrix $X \in \mathbb{R}^{I\times I}$ is *positive definite*, denoted $X \succ 0$, if it is symmetric and satisfies $z^T X z > 0$, for every non-zero $z \in \mathbb{R}^I$. If it is symmetric but satisfies the weaker condition that $z^T X z \geq 0$, for every $z \in \mathbb{R}^I$, then it is *positive semidefinite*, which we denote by $X \succeq 0$. Equivalently, X is positive semidefinite if and only if $X = Y^T Y$ for some matrix $Y \in \mathbb{R}^{J\times I}$ if and only if all its eigenvalues are non-negative. By I we denote the square identity matrix of appropriate dimensions, i.e., $\mathrm{I}_{ij} = 1$ if $i = j$ and $\mathrm{I}_{ij} = 0$ if $i \neq j$. By J we denote the square all-ones matrix of appropriate dimensions, i.e., $\mathrm{J}_{ij} = 1$ for all i and j. For I and J we omit the reference to the index set in the notation (particularly so if the index set is called I or J, for obvious reasons).

Vocabularies, structures and logics A many-sorted (relational) vocabulary L is a set of sort symbols $D_1, \ldots, D_s$ together with a set of relation symbols $R_1, \ldots, R_m$. Each relation symbol R in the list has an associated *type* of the form $D_{i_1} \times \cdots \times D_{i_r}$, where $r \geq 0$ is the *arity* of the symbol, and $i_1, \ldots, i_r \in [s]$ are not necessarily distinct. A structure $\mathbb{A}$ of vocabulary L, or an L-structure, is given by s disjoint sets $D_1, \ldots, D_s$ called *domains*, one for each sort symbol $D_i \in L$, and one relation $R \subseteq D_{i_1} \times \cdots \times D_{i_r}$ for each relation symbol $R \in L$ of type $D_{i_1} \times \cdots \times D_{i_r}$. We use $D(\mathbb{A})$ or D to denote the domain associated to the sort symbol D, and $R(\mathbb{A})$ or R to denote the relation associated to the relation symbol R. In practice, the overloading of the notation should never be an issue. The domain of a sort symbol is also called a *sort*. If $\mathbb{A}$ is an L-structure and L' is a many-sorted vocabulary obtained from L by removing some sort and relation symbols, then an *L'-reduct* of $\mathbb{A}$, denoted $L'(\mathbb{A})$, is the L'-structure obtained from $\mathbb{A}$ by omitting the domains and relations associated to the sort and relation symbols which are not present in L'.

A logic for a many-sorted vocabulary L has an underlying set of *individual variables* for each different sort in L. When interpreted on an L-structure, the variables are supposed to range over the domain of its sort; i.e., the variables are typed. Besides the equalities $x = y$ between variables of the same type, the atomic L-formulas are the formulas of the form $R(x_1, \ldots, x_r)$, where R is a relation symbol of arity r and $x_1, \ldots, x_r$ are variables of types that match the type of R. The formulas of first-order logic over L are built from the atomic formulas by negations, disjunctions, conjunctions, and existential and universal quantification of individual variables. For detailed background on first-order logic see, e.g., [13].

The syntax of First-Order Logic with Counting FOC is defined by adjoining one more sort N to the underlying vocabulary, adding one binary relation symbol $\leq$ of type $N \times N$ and

two ternary relation symbols + and $\times$ of types $N \times N \times N$, as well as extending the syntax to allow quantification of the form $\exists^{\geq y} x(\varphi)$, where φ is a formula, x is a variable of any type and y is a variable of type N. In the semantics of FOC, each L-structure $\mathbb{A}$ is expanded to an $L \cup \{N, \leq, +, \times\}$-structure with $N(\mathbb{A}) = \{0, \ldots, n\}$, where $n = \max\{|D_i(\mathbb{A})| : i = 1, \ldots, s\}$, and $\leq$, +, and $\times$ are interpreted by the standard arithmetic relations on $\{0, \ldots, n\}$. The meaning of $\exists^{\geq y} x(\varphi)$, for a concrete assignment $y \mapsto i \in \{0, \ldots, n\}$, is that there exist at least i many witnesses a for the variable x within its sort such that the assignment $x \mapsto a$ satisfies the formula φ. Numbers up to n^c, where $c > 1$ is an integer, are represented by c-tuples of numbers in $\{0, \ldots, n-1\}$. The arithmetic relations on such numbers, and the quantifiers counting up to such numbers, are both definable in FPC, the logic that we introduce next.

The syntax of Fixed-Point Logic with Counting FPC extends the syntax of FOC by allowing the formation of *inflationary fixed-point* formulas $\mathrm{ifp}_{x,X}\varphi(x, X)$. On a structure $\mathbb{A}$ of the appropriate vocabulary, such formulas are interpreted as defining the least fixed-point of the monotone operator $A \mapsto A \cup \{a \in D_{i_1} \times \cdots \times D_{i_r} : \mathbb{A} \models \varphi(a, A)\}$, where $D_{i_1} \times \cdots \times D_{i_r}$ is the type of the relation symbol X in $\varphi(x, X)$.

The syntax of Infinitary Logic with Counting $\mathrm{C}_{\infty\omega}$ extends the syntax of first-order logic by allowing quantifiers of the form $\exists^{\geq i} x(\varphi)$ which say that there are at least i many witnesses for the variable x, where i is a (concrete) natural number, as well as infinite disjunctions and conjunctions; i.e., formulas of the form $\bigvee_{i \in I} \phi_i$ and $\bigwedge_{i \in I} \phi_i$ where I is a possibly infinite index set, and $\{\phi_i : i \in I\}$ is an indexed set of formulas. The fragment of $\mathrm{C}_{\infty\omega}$ with k variables, denoted $\mathrm{C}^k_{\infty\omega}$, is the set of formulas that use at most k variables of any type. In the formulas of $\mathrm{C}^k_{\infty\omega}$ the variables can be reused and hence there is no finite bound on the quantification depth of the formulas. We write $\mathrm{C}^\omega_{\infty\omega}$ for the union of the $\mathrm{C}^k_{\infty\omega}$ over all natural numbers k. It is well-known that for every natural number k, every many-sorted vocabulary L, and every L-formula φ of FPC that uses k variables, there exists an L-formula ψ of $\mathrm{C}^{2k}_{\infty\omega}$ such that φ and ψ define the same relations over all finite L-structures. While all the published proofs that we are aware of give the statement for single-sorted vocabularies, it is clear that the case of many-sorted vocabularies is analogous. For the proof and more on FPC and $\mathrm{C}^\omega_{\infty\omega}$, we refer to [25].

Interpretations and reductions Let L and K be two many-sorted vocabularies, and let Θ be a class of K-formulas. A Θ-interpretation of L in K is given by: two Θ-formulas $\delta_D(x)$ and $\epsilon_D(x, y)$ for each sort symbol D of L, and one Θ-formula $\psi_R(x_1, \ldots, x_r)$ for each relation symbol $R \in L$ of arity r. In all these formulas, the displayed x's and y's are tuples of distinct variables of the same length m, called the arity of the interpretation. We say that the interpretation takes a K-structure $\mathbb{A}$ as input and produces an L-structure $\mathbb{B}$ as output if for each sort symbol D in L there exists a surjective partial map $f_D : A^m \to D(\mathbb{B})$, where A is the domain of $\mathbb{A}$, such that $f_D^{-1}(D(\mathbb{B})) = \{a \in A^m : \mathbb{A} \models \delta_D(a)\}$, $f_D^{-1}(\{(b, b) : b \in D(\mathbb{B})\}) = \{(a, b) \in (A^m)^2 : \mathbb{A} \models \epsilon_D(a, b)\}$, and $f_R^{-1}(R(\mathbb{B})) = \{(a_1, \ldots, a_r) \in (A^m)^r : \mathbb{A} \models \psi_R(a_1, \ldots, a_r)\}$ where $f_R = f_{D_1} \times \ldots \times f_{D_r}$ and $D_1 \times \cdots \times D_r$ is the type of R. The composition of two interpretations, one of L in K, and another one of K in J, is an interpretation

of L in J defined in the obvious way. Similarly, the composition of an interpretation of L in K with an L-formula is a K-formula defined in the obvious way. In all these compositions, the number of variables in the resulting formulas *multiply*. For example, the composition of a $\mathrm{C}^k_{\infty\omega}$-interpretation with a $\mathrm{C}^\ell_{\infty\omega}$-formula is a $\mathrm{C}^{k\ell}_{\infty\omega}$-formula. A reduction from a computational problem to another is a pair of maps f and g, where f takes an input x for the first problem and produces an input $y = f(x)$ for the second problem, and g takes x and a solution y' for y in the second problem and produces a solution $x' = g(x, y')$ for x in the first problem. The reduction is called a Θ-reduction if the maps can be produced by Θ-interpretations when their inputs are represented as structures of appropriate vocabularies. For more on interpretations and logical reductions see, e.g., [12].

Numbers, vectors and matrices as structures Since we are interested in definability in logics, we represent mathematical objects which serve as inputs and outputs of algorithms as finite relational structures. The details of the chosen representation are not essential, but we provide them for concreteness.

A natural number $n \in \mathbb{N}$ is represented by a structure, with a domain $\{0, \ldots, N-1\}$ of *bit positions* where $N \geq \lfloor \log_2(n+1) \rfloor$, of a vocabulary $L_{\mathbb{N}}$ that contains a binary relation symbol $\leq$ for the natural *linear order* on the bit positions, and a unary relation symbol P for the *actual bits*, i.e., the bit positions i that carry a 1-bit in the unique binary representation of n of length N. Single bits $b \in \{0, 1\}$ are represented as natural numbers with at least one bit position. Thus the vocabulary $L_{\mathbb{B}}$ for representing single bits is really the same as $L_{\mathbb{N}}$, but we still give it a separate name. A rational $q = (-1)^b n/d$, where $b \in \{0, 1\}$ and $n, d \in \mathbb{N}$, is represented by a structure with domain $\{0, \ldots, N-1\}$ of bit positions, where N is large enough to encode both the numerator n and the denominator d in binary. The vocabulary $L_{\mathbb{Q}}$ of this structure has one binary relation symbol $\leq$ for the natural linear order on the bit positions, and three unary relation symbols P_s, P_n and P_d that are used to encode the sign and the bits of the numerator and the denominator of q. We use zero denominator to represent $\pm\infty$.

An I-vector $u \in \mathbb{Q}^I$ is represented by a two-sorted structure, where the first sort $\bar{I}$ is the index set I and the second sort $\bar{B}$ is a domain $\{0, \ldots, N-1\}$ of bit positions, where N is large enough to encode all the numerators and denominators in the entries of u in binary. The vocabulary L_{vec} of this structure has one unary relation symbol I for $\bar{I}$, one binary relation symbol $\leq$ for the natural linear order on $\bar{B}$, and three binary relation symbols P_s, P_n and P_d, each of type $\bar{I} \times \bar{B}$, that are used to encode the entries of u in the expected way: $P_s(i, 0)$ if and only if $u(i)$ is positive, $P_n(i, j)$ if and only if the j-th bit of the numerator of $u(i)$ is 1, and $P_d(i, j)$ if and only if the j-th bit of the denominator of $u(i)$ is 1.

More generally, if $I_1, \ldots, I_d$ denote index sets that are not necessarily pairwise distinct, then the corresponding tensors $u \in \mathbb{Q}^{I_1 \times \cdots \times I_d}$ are represented by many-sorted structures, with one sort $\bar{I}$ for each index set I for as many different index sets as there are in the list $I_1, \ldots, I_d$, plus one sort $\bar{B}$ for the bit positions. The vocabulary $L_{\mathrm{vec},d}$ of these structures has one unary relation symbol I for each index sort $\bar{I}$, one binary relation symbol $\leq$ for the natural linear order on the bit positions $\bar{B}$, and three $d+1$-ary relation symbols P_s, P_n and P_d,

each of type $\bar{I}_1 \times \cdots \times \bar{I}_d \times \bar{B}$, for encoding the signs and the bits of the numerators and the denominators of the entries of the tensor. Matrices $A \in \mathbb{Q}^{I \times J}$ and square matrices $A \in \mathbb{Q}^{I \times I}$ are special cases of these, and so are indexed sets of vectors $\{u_i : i \in K\} \subseteq \mathbb{Q}^I$ and indexed sets of matrices $\{A_i : i \in K\} \subseteq \mathbb{Q}^{I \times J}$.

3 The Definable Ellipsoid Method

In this section we show that the ellipsoid method can be implemented in FPC for any family of explicitly-bounded convex sets. We begin by defining the problems involved.

3.1 Geometric problems and the ellipsoid method

Let $\mathscr{C}$ be a class of convex sets, each of the form $K \subseteq \mathbb{R}^I$ for some non-empty index set I. We will consider elements of $\mathscr{C}$ as inputs of computational problems, and therefore the class $\mathscr{C}$ comes with an associated encoding scheme. Most usual encoding schemes encode instances of a problem as finite binary strings. In our case, since we want to refer to definability in a logic, the encoding scheme for $\mathscr{C}$ will encode each set K through a finite relational structure. The details are discussed in Subsection 3.2 below.

We assume that the encoding of a set $K \subseteq \mathbb{R}^I$ carries within it enough information to determine the set I. If the encoding also carries information about a rational R satisfying $K \subseteq S(0^I, R)$, then we say that K is *circumscribed*, and we write $(K; I, R)$ to refer to it. We write $(K; n, R)$ whenever $I = [n]$.

The *exact feasibility problem* for $\mathscr{C}$ takes as input the encoding of a set $K \subseteq \mathbb{R}^I$ in $\mathscr{C}$ and asks for a bit $b \in \{0, 1\}$ that is 1 if K is non-empty, and 0 if K is empty. The *weak feasibility problem* for $\mathscr{C}$ takes as input the encoding of a set $K \subseteq \mathbb{R}^I$ in $\mathscr{C}$ and a rational $\epsilon > 0$ and asks for a bit $b \in \{0, 1\}$ and a vector $x \in \mathbb{Q}^I$ such that:

1. $b = 1$ and $x \in S(K, \epsilon)$, or
2. $b = 0$ and $\mathrm{vol}(K) \leq \epsilon$.

The reason why the exact feasibility problem is formulated as a decision problem and does not ask for a feasible point is that K could well be a single point with non-rational components. In the weak feasibility problem this is not an issue because if K is non-empty, then the ball $S(K, \epsilon)$ surely contains a rational point. The *not-so-weak separation problem* for $\mathscr{C}$ takes as input the encoding of a set $K \subseteq \mathbb{R}^I$ in $\mathscr{C}$, a vector $y \in \mathbb{Q}^I$, and a rational $\delta > 0$ and asks as output for a bit $b \in \{0, 1\}$ and a vector $s \in \mathbb{Q}^I$ such that $||s||_\infty = 1$ and:

1. $b = 1$ and $y \in S(K, \delta)$, or
2. $b = 0$ and $\langle s, y \rangle + \delta \geq \sup\{\langle s, x \rangle : x \in K\}$.

The problems carry the adjective *weak* in their name to stress on the fact that in both cases the more natural requirement of membership in K is replaced by the looser requirement of membership in $S(K, \gamma)$ for a given $\gamma > 0$. For the weak separation problem, the additional

qualification *not-so-weak* serves the purpose of distinguishing it from the *weak(er)* version in which condition 2 is replaced by the looser requirement that $b = 0$ and $\langle s, y\rangle + \delta \geq \sup\{\langle s, x\rangle : x \in S(K, -\delta)\}$. It turns out that the main procedure of the ellipsoid method, as stated in the monograph [16] and in Theorem 3.1 below, requires the *not-so-weak* version. Recall that an ellipsoid in $\mathbb{R}^I$ is a set of the form $E(A, a) = \{x \in \mathbb{R}^I : (x-a)^T A(x-a) \leq 1\}$, where $a \in \mathbb{R}^I$ is the center, and A is an $I \times I$ positive definite matrix.

Theorem 3.1 (Theorem 3.2.1 in [16])**.** *There is an oracle polynomial-time algorithm, the central-cut ellipsoid method* CC*, that solves the following problem: Given a rational number $\epsilon > 0$ and a circumscribed closed convex set $(K; n, R)$ given by an oracle that solves the not-so-weak separation problem for K, outputs one of the following: either a vector $x \in S(K, \epsilon)$, or a positive definite matrix $A \in \mathbb{Q}^{n\times n}$ and a vector $a \in \mathbb{Q}^n$ such that $K \subseteq E(A, a)$ and* $\mathrm{vol}(E(A, a)) \leq \epsilon$.

We plan to use the algorithm CC from Theorem 3.1 almost as a black box, except for the four aspects of it listed below. Although they are not stated in Theorem 3.2.1 in [16], inspection of the proof and the definitions in the book shows that they hold:

1. the input to the algorithm is the triple given by ϵ, n and R,
2. the rational numbers ϵ and R are represented in binary,
3. the natural number n is represented in unary (i.e., 2^n is given in binary),
4. the algorithm makes at least one oracle query, and the output is determined by the answer to the last oracle call in the following way: if this last call was (y, δ) and the answer was the pair (b, s), then $\delta \leq \epsilon$ and the output vector x of CC is y itself whenever $b = 1$, and there exists a positive definite matrix A and a vector a so that $K \subseteq E(A, a)$ and $\mathrm{vol}(E(A, a)) \leq \epsilon$ whenever $b = 0$.

The last point implies, in particular, that CC solves the weak feasibility problem for the given K. However, note also that the theorem states a notably stronger claim than the existence of a polynomial-time oracle reduction from the weak feasibility problem for a class $\mathscr{C}$ of sets to the not-so-weak separation problem for the same class $\mathscr{C}$ of sets: indeed, CC solves the feasibility problem for K by making oracle calls to the separation problem for *the same* K.

3.2 Definability of ellipsoid

We encode sets in $\mathscr{C}$ as finite relational structures in an isomorphism-invariant way. Such encodings we call *representations*. We define this formally.

Let us first specify what it means for two sets $P \subseteq \mathbb{R}^I$ and $Q \subseteq \mathbb{R}^J$ to be isomorphic, where I and J are two non-empty index sets. For a function $\sigma : I \to J$ and a J-vector v, we denote by $[v]^{-\sigma}$ the I-vector defined by $[v]^{-\sigma}(i) = v(\sigma(i))$ for every $i \in I$. For sets of J-vectors, such as Q, we define $[Q]^{-\sigma} = \{[v]^{-\sigma} : v \in Q\}$. We say that P and Q are *isomorphic*, denoted $P \cong Q$, if there is a bijection $\sigma : I \to J$ such that $P = [Q]^{-\sigma}$. Now we can define

representations of classes of sets. A *representation* of the class $\mathscr{C}$ of sets is a surjective partial map r from the class of finite L-structures onto $\mathscr{C}$, where L is a finite vocabulary with at least one unary relation symbol I, that satisfies the following conditions:

1. for every two $\mathbb{A}, \mathbb{B} \in \mathrm{Dom}(r)$, if $\mathbb{A} \cong \mathbb{B}$, then $r(\mathbb{A}) \cong r(\mathbb{B})$,
2. for every $\mathbb{A} \in \mathrm{Dom}(r)$ it holds that $r(\mathbb{A}) \subseteq \mathbb{R}^I$ where $I = I(\mathbb{A})$.

A *circumscribed representation* of $\mathscr{C}$ is a surjective partial map r from the class of finite L-structures onto $\mathscr{C}$, where L is a finite vocabulary containing at least one unary relation symbol I as well as a copy of the vocabulary $L_{\mathbb{Q}}$, that satisfies the following conditions:

1. for every two $\mathbb{A}, \mathbb{B} \in \mathrm{Dom}(r)$, if $\mathbb{A} \cong \mathbb{B}$, then $r(\mathbb{A}) \cong r(\mathbb{B})$,
2. for every $\mathbb{A} \in \mathrm{Dom}(r)$ it holds that $r(\mathbb{A}) \subseteq \mathbb{R}^I$ where $I = I(\mathbb{A})$,
3. for every $\mathbb{A} \in \mathrm{Dom}(r)$ it holds that $r(\mathbb{A}) \subseteq S(0^I, R)$ where R is the rational number represented by the $L_{\mathbb{Q}}$-reduct of $\mathbb{A}$.

Note that a circumscribed representation of $\mathscr{C}$ exists only if every K in $\mathscr{C}$ is bounded. For a given representation r of $\mathscr{C}$, any of the existing preimages $\mathbb{A} \in r^{-1}(K)$ of a set $K \in \mathscr{C}$ is called a *representation* of K. If L is the vocabulary of the representation, then we say that $\mathscr{C}$ is represented in vocabulary L. If $\mathscr{C}$ has a representation in some vocabulary L, then we say that $\mathscr{C}$ is a *represented class of sets*, and if it has a circumscribed representation, then we say that it is a *represented class of circumscribed sets*.

If $\mathscr{C}$ is a represented class of convex sets, L is the vocabulary of the representation, and Φ is a class of logical formulas, then we say that the weak feasibility problem for $\mathscr{C}$ is Φ-definable if there exists a Φ-interpretation that, given as input a representation of a set K in $\mathscr{C}$ and a rational $\epsilon > 0$ as a structure over $L \,\dot\cup\, L_{\mathbb{Q}}$, produces a structure over $L_{\mathbb{B}} \,\dot\cup\, L_{\mathrm{vec}}$ representing a valid output. It is required in addition that the represented $K \subseteq \mathbb{R}^I$ from the input and the vector $x \in \mathbb{Q}^I$ from the output share the same sort $\bar{I}$ with the same relation symbol I interpreted by the same set. Similarly, for the not-so-weak separation problem, the input is a structure over $L \,\dot\cup\, L_{\mathbb{Q}} \,\dot\cup\, L_{\mathrm{vec}}$ and the output is a structure over $L_{\mathbb{B}} \,\dot\cup\, L_{\mathrm{vec}}$. Again, the represented $K \subseteq \mathbb{R}^I$ and the vector $y \in \mathbb{Q}^I$ from the input, and the vector $s \in \mathbb{Q}^I$ from the output, share the same sort $\bar{I}$ with the same relation symbol I interpreted by the same set.

The following is the main result of this section.

Theorem 3.2. *Let $\mathscr{C}$ be a represented class of circumscribed closed convex sets. If the not-so-weak separation problem for $\mathscr{C}$ is* FPC*-definable, then the weak feasibility problem for $\mathscr{C}$ is also* FPC*-definable.*

Although all the main ideas of the proof in Subsection 3.4 below were already present in the works [2] and [11], we present a detailed proof for completeness as then the key new insights become clearer.

At an intuitive level, the main difficulty for simulating the ellipsoid method within a logic is that one needs to make sure that the execution of the algorithm stays *canonical*; i.e.,

invariant under the isomorphisms of the input structure. The principal device to achieve this is the following clever idea from [2]: instead of running the ellipsoid method directly over the given set $K \subseteq \mathbb{R}^I$, the algorithm is run over certain *folded* versions $[K]^\sigma \subseteq \mathbb{R}^{\sigma(I)}$ of K, where $\sigma(I)$ is an *ordered subset* of I. If the execution of the ellipsoid algorithm does not detect the difference between K and the folded $[K]^\sigma$, then an appropriately defined *unfolding* of the solution for $[K]^\sigma$ will give the right solution for K. If, on the contrary, the ellipsoid detects the difference in the form of a vector $u \in \mathbb{Q}^I$ whose folding $[u]^\sigma$ does not unfold appropriately, then the knowledge of u is exploited to *refine* the current folding into a strictly larger ordered $\sigma'(I) \subseteq I$, and the execution is rebooted with the new $[K]^{\sigma'} \subseteq \mathbb{R}^{\sigma'(I)}$. After no more than $|I|$ refinements the folding will be indistinguishable from K, and the execution will be correct.

The crux of the argument that makes this procedure FPC-definable is that the ellipsoid algorithm is always operating over an *ordered* set $\sigma(I)$. In particular, the algorithm stays canonical, and the polynomially many steps of its execution are expressible in fixed-point logic FP by the Immerman-Vardi Theorem [18, 32]. Indeed, the counting ability of FPC is required only during the folding/unfolding/refining steps.

Our formalization of these ideas will bring in two key insights that were not present in earlier works. The first one is the observation that it is possible to simulate the oracle queries to the folded $[K]^\sigma$ by oracle queries to the original K, and that this works for any closed convex set K. Furthermore, it is possible to transfer an appropriately chosen termination condition on the volume of the folded $[K]^\sigma$ to the required termination condition on the volume of the original K. The observation that the volume condition commutes with the folding operations on arbitrary K's did not appear in the earlier work on LPs [2], nor on SDPs [11]. For LPs, the ellipsoid method on $[K]^\sigma$ is typically combined with a rounding procedure to actually solve the exact feasibility problem. Therefore, the termination condition in that case is just exact feasibility, or plain emptiness. For SDPs, the volume-based termination condition is not analyzed in [11], since the results there hold under the additional assumption of full-dimensionality and the ellipsoid method always outputs a vector in $S(K, \epsilon)$. The claim that the folding operations do have a mild effect on the volume of arbitrary K's is the subject of Lemma 3.4 below.

The second key insight that our formalization brings in is the observation that the two *precision* parameters of an input for an arbitrary K, i.e., the *large* radius $R > 0$ in the circumscribing assumption, and the *small* margin guarantee $\epsilon > 0$ in the weak feasibility goal, do not interfere with the requirement that the algorithm behaves in an isomorphism-invariant way. Again, this observation was not clearly analyzed in the previous work on LPs, nor on SDPs. As explained in the introduction, it will be crucial for us to be able to handle arbitrarily large $R > 0$, and arbitrarily small $\epsilon > 0$, to get the results of Section 4.

Before we move on to the actual proof of Theorem 3.2, we discuss the required material for the method of foldings.

3.3 Folding operations

Let I and J be non-empty index sets. Let $\sigma : I \to J$ be an onto map. The *almost-folding* $(u)^\sigma$ and the *normalized almost-folding* $(u)^\sigma_{\mathrm{n}}$ of an I-vector u are the J-vectors defined by

$$(u)^\sigma(j) := \sum_{i \in \sigma^{-1}(j)} u(i) \qquad \text{and} \qquad (u)^\sigma_{\mathrm{n}} := \frac{(u)^\sigma}{\|(u)^\sigma\|_\infty} \tag{3.1}$$

for every $j \in J$, with the understanding that if $\|(u)^\sigma\|_\infty = 0$, then $(u)^\sigma_{\mathrm{n}}$ is defined as the zero vector. The *folding* $[u]^\sigma$ of an I-vector u and the *unfolding* $[v]^{-\sigma}$ of a J-vector v are the vectors defined by

$$[u]^\sigma(j) := \frac{1}{|\sigma^{-1}(j)|} \sum_{i \in \sigma^{-1}(j)} u(i) \qquad \text{and} \qquad [v]^{-\sigma}(i) := v(\sigma(i)) \tag{3.2}$$

for every $j \in J$ and every $i \in I$, respectively. For sets $K \subseteq \mathbb{R}^I$ and $L \subseteq \mathbb{R}^J$, define $[K]^\sigma := \{[u]^\sigma : u \in K\}$ and $[L]^{-\sigma} := \{[v]^{-\sigma} : v \in L\}$. Observe that the notation $[v]^{-\sigma}$ and $[L]^{-\sigma}$ agrees with the one we introduced earlier when we defined representations. The map σ is said to *respect* a vector $u \in \mathbb{R}^I$ if $u_i = u_{i'}$ whenever $\sigma(i) = \sigma(i')$ for every $i, i' \in I$. The following lemma collects a few important properties of foldings. See Propositions 17 and 18 in [11] in which properties 4 and 7 from the lemma are also proved for all sets but stated only for convex sets. A small difference is that our statement of 7 is written in terms of the normalized almost folding operation defined above which is what is actually needed in the uses of the lemma.

Lemma 3.3. *Let $\sigma : I \to J$ be an onto map, let u and v be I-vectors, and let K be a set of I-vectors. Then the following hold:*

1. $[au + bv]^\sigma = a[u]^\sigma + b[v]^\sigma$ *for every* $a, b \in \mathbb{R}$,
2. $\|[u]^\sigma\|_2 \le \|u\|_2$,
3. $K \subseteq S(0^I, R)$ *implies* $[K]^\sigma \subseteq S(0^J, R)$,
4. $u \in S(K, \delta)$ *implies* $[u]^\sigma \in S([K]^\sigma, \delta)$,
5. *if K is convex, then $[K]^\sigma$ is convex, and*
6. *if K is bounded and closed, then $[K]^\sigma$ is bounded and closed,*
7. *if $\delta > 0$ and σ respects u, and $\|u\|_\infty = 1$ and $\langle u, v\rangle + \delta \ge \sup\{\langle u, x\rangle : x \in K\}$, then $\|(u)^\sigma_{\mathrm{n}}\|_\infty = 1$ and $\langle (u)^\sigma_{\mathrm{n}}, [v]^\sigma\rangle + \delta \ge \sup\{\langle (u)^\sigma_{\mathrm{n}}, x\rangle : x \in [K]^\sigma\}$.*

Proof. Property 1 is straightforward by definition. Property 2 follows from the inequality $(x_1 + \cdots + x_d)^2 \le (x_1^2 + \cdots + x_d^2)d$, which is the special case of the Cauchy-Schwartz inequality $|\langle x, y\rangle| \le \|x\|_2 \|y\|_2$ where y is the d-dimensional all-ones vector. Property 3 is an immediate consequence of 2. Property 4 follows from 1 and 2: if $\|u - x\|_2 \le \delta$, then $\|[u]^\sigma - [x]^\sigma\|_2 = \|[u - x]^\sigma\|_2 \le \|u - x\|_2 \le \delta$. Property 5 follows from the fact that

the map $u \mapsto [u]^\sigma$ is linear. Property 6 follows from the fact that a continuous image of a compact set is compact: indeed the map $u \mapsto [u]^\sigma$ is continuous, and a subset of Euclidean space is compact if and only if it is closed and bounded. Property 7 follows from the straightforward fact that whenever σ respects u, we have $\langle (u)^\sigma, [y]^\sigma \rangle = \langle u, y \rangle$ and $\|(u)^\sigma\|_\infty \geq \|u\|_\infty$. Indeed, $\sup\{\langle (u)^\sigma_{\mathrm{n}}, x \rangle : x \in [K]^\sigma\} = \sup\{\langle (u)^\sigma_{\mathrm{n}}, [x]^\sigma \rangle : x \in K\}$, and for every $x \in K$ we have $\langle (u)^\sigma_{\mathrm{n}}, [x]^\sigma \rangle - \langle (u)^\sigma_{\mathrm{n}}, [v]^\sigma \rangle = \langle (u)^\sigma_{\mathrm{n}}, [x - v]^\sigma \rangle$. Now either $\langle (u)^\sigma_{\mathrm{n}}, [x - v]^\sigma \rangle \leq 0 < \delta$, or $\langle (u)^\sigma_{\mathrm{n}}, [x - v]^\sigma \rangle > 0$ and since $\|(u)^\sigma\|_\infty \geq \|u\|_\infty = 1$, we have $\langle (u)^\sigma_{\mathrm{n}}, [x - v]^\sigma \rangle = \langle (u)^\sigma, [x - v]^\sigma \rangle / \|(u)^\sigma\|_\infty \leq \langle (u)^\sigma, [x - v]^\sigma \rangle = \langle u, x - v \rangle = \langle u, x \rangle - \langle u, v \rangle \leq \delta$. □

There is one further important property of foldings that we need for correctness of the FPC-interpretation that we are about to define. Let us extend the definition of the set $E(A, a) = \{x \in \mathbb{R}^J : (x - a)^T A (x - a) \leq 1\}$ to arbitrary positive semidefinite matrices A. It should be noted that if A is positive semidefinite but not positive definite, then at least one of the semi-axes of $E(A, a)$ is infinite and hence the set is unbounded. In this case we call $E(A, a)$ an *unbounded ellipsoid*. If the simulation of the run of CC is executed until the end over a folded $[K]^\sigma$ and the output bit is 0, then the algorithm certifies that $[K]^\sigma$ is contained in an ellipsoid of a small volume (see point 3 immediately following the statement of Theorem 3.1). To ensure that the volume of K itself is small we use the following lemma.

Lemma 3.4. *Let $K \subseteq \mathbb{R}^I$ be a set, let $\sigma : I \to J$ be an onto map, and let $R \in \mathbb{R}^{J \times I}$ and $L \in \mathbb{R}^{I \times J}$ be the matrices that define the linear maps $u \mapsto [u]^\sigma$ and $v \mapsto [v]^{-\sigma}$, respectively. If there is a positive definite matrix $A \in \mathbb{R}^{J \times J}$ and a vector $a \in \mathbb{R}^J$ such that $[K]^\sigma \subseteq E(A, a)$, then $K \subseteq E(R^T A R, La)$. Moreover, for every $\epsilon > 0$ and $r > 0$, if $\mathrm{vol}(E(A, a)) \leq \epsilon$, then $\mathrm{vol}(E(R^T A R, La) \cap S(0^I, r)) \leq 2^n r^{n-1} n k \epsilon^{1/k}$, where $n = |I|$ and $k = |J|$.*

Proof. Assume that $[K]^\sigma \subseteq E(A, a)$, where $A = B^T B$ is positive definite. Take a point $x \in K$. We want to show that x is in $E((BR)^T (BR), La)$. We have:

$$\|BR(x - La)\|_2^2 = \|B(Rx - RLa)\|_2^2 = \|B(Rx - a)\|_2^2 \leq 1, \tag{3.3}$$

with the first equality following from the linearity of R, the second equality following from the easily verified fact that $[[a]^{-\sigma}]^\sigma = a$, and the inequality following from the fact that $x \in K$ and hence $Rx = [x]^\sigma$ belongs to $[K]^\sigma \subseteq E(A, a) = E(B^T B, a)$.

For the second part of the proof, observe that the matrix $R^T A R = (BR)^T (BR)$ is positive semidefinite. Let $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$ be the eigenvalues of $R^T A R$, let $V = \{u_1, \ldots, u_n\}$ be an orthonormal basis of corresponding eigenvectors, and let $(b_1, \ldots, b_n)$ be the coordinates of La with respect to the basis V. The axes of symmetry of the (possibly unbounded) ellipsoid $E(R^T A R, La)$ correspond to the vectors in V. As we show below, $\lambda_1 > 0$ and therefore the shortest axis of $E(R^T A R, La)$ has a finite length $2(1/\lambda_1)^{1/2}$. It follows that $E(R^T A R, La)$ is contained in the set of points whose coordinates, with respect to the basis V, are given by $[b_1 - (1/\lambda_1)^{1/2}, b_1 + (1/\lambda_1)^{1/2}] \times \mathbb{R}^{n-1}$. Since the r-ball $S(0, r)$ is inscribed in the n-dimensional hypercube $[-r, r]^n$, where the coordinates are again given with respect to the basis V, this implies that $E(R^T A R, La) \cap S(0^I, r)$ is contained in $[b_1 - (1/\lambda_1)^{1/2}, b_1 + (1/\lambda_1)^{1/2}] \times [-r, r]^{n-1}$. Hence,

$$\mathrm{vol}(E(R^T A R, La) \cap S(0^I, r)) \leq 2(1/\lambda_1)^{1/2} (2r)^{n-1} = 2^n r^{n-1} (1/\lambda_1)^{1/2}. \tag{3.4}$$

We will finish the proof by showing that $\mathrm{vol}(E(A,a)) \leq \epsilon$ implies $(1/\lambda_1)^{1/2} \leq nk\epsilon^{1/k}$, and in particular $\lambda_1 > 0$.

Let $\mu_1 \geq \cdots \geq \mu_k > 0$ be the eigenvalues of the matrix A. We have

$$\mathrm{vol}(E(A,a)) = V_k(1/\mu_1)^{1/2} \cdots (1/\mu_k)^{1/2} \geq V_k(1/\mu_1)^{k/2}, \tag{3.5}$$

where V_k denotes the volume of a 1-ball in the k-dimensional real vector space (for the volume of an ellipsoid see, e.g., [16]). Therefore, if $\mathrm{vol}(E(A,a)) \leq \epsilon$, then $\mu_1 \geq (V_k/\epsilon)^{2/k} > k^{-2}(1/\epsilon)^{2/k}$, where the last inequality follows from the fact that $V_k > k^{-k}$. Now, let $y \in \mathbb{R}^J$ be an eigenvector of A corresponding to the eigenvalue μ_1, and let $x = Ly$. Note that $x^T x \leq ny^T y$. Hence,

$$x^T R^T A R x = y^T A y = \mu_1 y^T y \geq (\mu_1/n) x^T x. \tag{3.6}$$

Since $y \neq 0$ also $x \neq 0$, and the Rayleigh quotient principle implies that $\lambda_1 \geq \mu_1/n > 0$. Hence $\lambda_1 \geq k^{-2}(1/\epsilon)^{2/k}/n$, which gives $(1/\lambda_1)^{1/2} \leq n^{1/2}k\epsilon^{1/k} \leq nk\epsilon^{1/k}$. □

From now on, all maps $\sigma : I \to J$ will be onto and have $J = [k]$ for some positive integer k. Such maps define a preorder $\leq_\sigma$ on I with exactly k equivalence classes which is defined by $i \leq_\sigma i'$ if and only if $\sigma(i) \leq \sigma(i')$. A second map $\sigma' : I \to [k']$ is a *refinement* of σ if $\sigma'(i) \leq \sigma'(i')$ implies $\sigma(i) \leq \sigma(i')$. The refinement is *proper* if there exist $i, i' \in I$ such that $\sigma'(i) < \sigma'(i')$ and $\sigma(i) = \sigma(i')$. Recall that $\sigma : I \to [k]$ *respects* a vector $v \in \mathbb{R}^I$ if $v(i) = v(i')$ whenever $\sigma(i) = \sigma(i')$. Since any bijective map respects any vector, observe that if σ does not respect v, then there exists at least one proper refinement of σ that does respect v. We aim for a canonical such refinement, that we denote σ^v, and that is definable in FPC. We define it as follows.

Fix an onto map $\sigma : I \to [k]$ and a vector $v \in \mathbb{R}^I$. Define:

$$\begin{array}{lcll} n(j) & := & |\{v(\ell) : \sigma(\ell) = j\}| & \text{for } j \in [k], \\ m(i) & := & |\{v(\ell) : \sigma(\ell) = \sigma(i),\ v(\ell) \leq v(i)\}| & \text{for } i \in I, \\ \sigma'(i) & := & n(1) + \cdots + n(\sigma(i) - 1) + m(i) & \text{for } i \in I, \\ k' & := & n(1) + \cdots + n(k). & \end{array}$$

In words, $n(j)$ is the number of distinct v-values in the j-th equivalence class of $\leq_\sigma$, and $m(i)$ is the number of distinct v-values in the equivalence class of i that are no bigger than the v-value $v(i)$ of i. The map $\sigma' : I \to [k']$ is our σ^v. Note that if σ respects v, then $\sigma^v = \sigma$. On the other hand:

Fact 3.5. *If σ does not respect v, then σ^v is onto and a proper refinement of σ that respects v.*

Although not strictly needed, it is useful to note that σ^v is a coarsest refinement of σ that respects v. The final lemma before we proceed to the proof of Theorem 3.2 collects a few computation tasks about foldings that are FPC-definable:

Lemma 3.6. *The following operations have* FPC*-interpretations:*

1. *given a set I, output the 0 vector 0^I and the constant 1 map $\sigma : I \to [1]$,*

2. *given* $u \in \mathbb{Q}^I$ *and onto* $\sigma : I \to [k]$, *output* $(u)^\sigma_{\mathrm{n}}$,
3. *given* $u \in \mathbb{Q}^k$ *and onto* $\sigma : I \to [k]$, *output* $[u]^{-\sigma}$,
4. *given* $u \in \mathbb{Q}^I$ *and onto* $\sigma : I \to [k]$, *output* 1 *if* σ *respects* u, *else output* 0,
5. *given* $u \in \mathbb{Q}^I$ *and* $\sigma : I \to [k]$, *output* $\sigma^u : I \to [k']$.

Proof. All five cases are straightforward given the ability of FPC to perform the basic arithmetic of rational numbers, compute sums of sets of rationals indexed by definable sets, and compute cardinalities of definable sets. □

3.4 Proof of Theorem 3.2

Let Ψ be an FPC-interpretation that witnesses that the not-so-weak separation problem for $\mathscr{C}$ is FPC-definable. We start by showing that there is an FPC-interpretation Ψ' that either simulates the not-so-weak separation oracle for $[K]^\sigma$ or outputs a vector not respected by σ. More precisely, Ψ' takes as input a representation of a set $K \subseteq \mathbb{R}^I$ in $\mathscr{C}$, an onto mapping $\sigma : I \to [k]$ where k is an integer that satisfies $1 \leq k \leq |I|$, a vector $y \in \mathbb{Q}^k$, and a rational $\delta > 0$ and outputs an integer $b \in \{-1, 0, 1\}$ and a vector $s \in \mathbb{Q}^I$ such that $\|s\|_\infty = 1$ and:

1. $b = 1$ and σ respects s and $[y]^{-\sigma} \in S(K, \delta)$ and $y \in S([K]^\sigma, \delta)$, or
2. $b = 0$ and σ respects s and $\langle (s)^\sigma_{\mathrm{n}}, y\rangle + \delta \geq \sup\{\langle (s)^\sigma_{\mathrm{n}}, x\rangle : x \in [K]^\sigma\}$, or
3. $b = -1$ and σ does not respect s.

Concretely, let Ψ' be the interpretation that does the following:

01. given a representation of $K \subseteq \mathbb{R}^I$ in $\mathscr{C}$, $\sigma : I \to [k]$, $y \in \mathbb{Q}^k$, and $\delta \in \mathbb{Q}$,
02. compute $y^- := [y]^{-\sigma}$ and $(b, s) := \Psi(K; y^-, \delta)$,
03. if σ respects s, output the same (b, s),
04. if σ does not respect s, output $(-1, s)$.

The claim that Ψ' is FPC-definable follows from points 3 and 4 in Lemma 3.6. The claim that Ψ' satisfies the required conditions follows from the correctness of Ψ, together with the fact that $[[y]^{-\sigma}]^\sigma = y$, and Properties 4 and 7 in Lemma 3.3. For later use, let us note that if the given $\sigma : I \to [k]$ is a bijection, then the third type of output $b = -1$ cannot occur.

Next we show how to use Ψ' in order to implement, in FPC, the algorithm CC from Theorem 3.1. Consider the following variant CC′ of CC:

01. given a rational $\epsilon > 0$ and a representation of a set $K \subseteq \mathbb{R}^I$ in $\mathscr{C}$,
02. compute R satisfying $K \subseteq S(0^I, R)$ from the representation of K,
03. let $n := |I|$ and $k := 1$, and let $\sigma : I \to [1]$ be the constant 1 map,
04. start a run of CC on input (γ, k, R) where $\gamma := \min\{(\epsilon/(2^n R^{n-1} nk))^k, \epsilon\}$,
05. given an oracle query (y, δ), replace it by $(b, s) := \Psi'(K; \sigma, y, \delta)$,
06. if σ respects s, then

07. compute $(s)^\sigma_{\mathrm{n}}$ and take the pair $(b, (s)^\sigma_{\mathrm{n}})$ as an output to the query (y, δ),
08. if the run of CC makes a new query (y, δ), go back to step 05,
09. if the run of CC makes no more queries, go to step 13,
10. else
11. compute $\sigma^s : I \to [k']$, the canonical refinement of σ that respects s,
12. reboot the run of CC with $\sigma := \sigma^s$ and $k := k'$ and go back to step 04,
13. let (b, s) be the output of Ψ' for the last oracle call (y, δ),
14. output $(b, [y]^{-\sigma})$.

As discussed above, CC$'$ simulates the ellipsoid method over folded versions $[K]^\sigma$ of K. By Properties 3, 5 and 6 in Lemma 3.3 each such folded version is a circumscribed closed convex set. A key aspect of CC that makes this algorithm well defined is that the only knowledge about the targeted set $[K]^\sigma$ that it needs for steps 04, 05, 08 and 09 are its dimension k, its bounding radius R, and correct answers to the earlier queries to the not-so-weak separation oracle for $[K]^\sigma$ itself. In particular, the algorithm does not need the folded $[K]^\sigma$ to belong to the class $\mathscr{C}$. Indeed, as long as all s vectors are respected by a particular σ, Properties 4 and 7 in Lemma 3.3 guarantee that the answers in step 07 stay consistent with the assumption that the circumscribed closed convex set given by the oracle *is* $[K]^\sigma$. As soon as an s vector that is not respected by σ is found, the map σ is refined, and the run of CC is rebooted with the new k and γ for the new σ (and the same R).

No later than after $|I|$ refinements of σ, the simulation of CC will be executed until the end. This happens at the latest when σ becomes the totally refined map: at that point σ is a bijection that respects every s. Whenever a run of the simulation is completed, the algorithm reaches step 13 with a pair (b, s) and a σ that respects s. We use this to show that CC$'$ solves the weak feasibility problem for K, and that it can be implemented in FPC.

The claim that CC$'$ solves the weak feasibility problem for any K in $\mathscr{C}$ is proved as follows. Let (b, s) be the output of Ψ' for the last oracle call (y, δ) of the execution of CC. As noted above, $\sigma : I \to [k]$ respects s and hence $b \in \{0, 1\}$ by Property 3 in the description of Ψ'. If $b = 1$, then $[y]^{-\sigma} \in S(K, \delta)$ by Property 1 in the description of Ψ', and $S(K, \delta) \subseteq S(K, \epsilon)$ because $\delta \leq \gamma \leq \epsilon$. This shows that $(b, [y]^{-\sigma})$ is a correct output for the weak feasibility problem for ϵ and K in case $b = 1$. In case $b = 0$ we have $[K]^\sigma \subseteq E(A, a)$ for a positive definite matrix A and a vector a, with $\mathrm{vol}(E(A, a)) \leq \gamma \leq (\epsilon/(2^n R^{n-1} nk))^k$, by point 4 immediately following the statement of Theorem 3.1. Since $K \subseteq S(0, R)$, by Lemma 3.4 this means that the volume of K is at most ϵ and the answer $b = 0$ is a correct output.

For the implementation in FPC, we note that CC$'$ is a relational WHILE algorithm that halts after at most $|I|$ iterations all whose steps can be computed by FPC-interpretations without quotients. Step 01 is the description of the input. Step 02 follows from the fact that K has a circumscribed representation: just take the $L_{\mathbb{Q}}$-reduct of the representation of K, where $L_{\mathbb{Q}}$ is the copy of the vocabulary that is used for representing the rational radius R. Step 03 follows from point 1 in Lemma 3.6. Step 04 follows from the Immerman-Vardi Theorem on the fact that the representation of $[k]$ is an ordered structure and the computation of CC in between oracle calls runs in polynomial time. Step 05 follows from the fact that Ψ' is FPC-definable. Step 06 is just a control statement. Step 07 follows

from point 2 in Lemma 3.6. Step 08 follows, again, from the Immerman-Vardi Theorem on the fact that the representation of $[k]$ is an ordered structure and the computation of CC in between oracle calls runs in polynomial time. Step 09 follows from the same reason as Step 08. Step 10 is a control statement. Step 11 follows from point 5 in Lemma 3.6. Steps 12 and 13 are just control statements. Step 14 follows from point 3 in Lemma 3.6.

This completes the proof of Theorem 3.2, and this section.

4 Feasibility of SDPs

In this section we use Theorem 3.2 to show that the exact feasibility of semidefinite programs is definable in $\mathrm{C}^{\omega}_{\infty\omega}$.

4.1 Semidefinite sets

The *semidefinite set* $K_{A,b} \subseteq \mathbb{R}^I$ is defined by the constraints

$$\langle A_i, X\rangle \leq b_i \ \text{ for } i \in M \ \text{ and } \ X \succeq 0, \tag{4.1}$$

where $A \in \mathbb{R}^{M\times(J\times J)}$ is an indexed set of $J \times J$ matrices, $b \in \mathbb{R}^M$ is an indexed set of reals, X is a $J \times J$ symmetric matrix of formal variables $x_{ij} = x_{ji} = x_{\{i,j\}}$ for $i, j \in J$, and $I = \{\{i, j\} : i, j \in J\}$ is the set of variable indices. A *circumscribed semidefinite set* is a pair $(K_{A,b} \subseteq \mathbb{R}^I, R)$, where $K_{A,b} \subseteq \mathbb{R}^I$ is a semidefinite set as defined above and R is a rational satisfying $K_{A,b} \subseteq S(0^I, R)$.

When A and b have rational coefficients, the semidefinite set $K_{A,b} \subseteq \mathbb{R}^I$ is represented by a four-sorted structure, with one sort $\bar{I}$ for the set I of indices of variables, two sorts $\bar{J}$ and $\bar{M}$ for the index sets J and M, and one sort $\bar{B}$ for a domain $\{0, \ldots, N-1\}$ of bit positions that is large enough to encode all the numbers in binary. The vocabulary L_{SDP} includes the following relation symbols:

1. three unary symbols I, J and M, for $\bar{I}$, $\bar{J}$ and $\bar{M}$, respectively,
2. one ternary symbol P of type $\bar{I} \times \bar{J} \times \bar{J}$,
3. one binary symbol $\leq$ for the natural linear order on $\bar{B}$,
4. three 4-ary symbols $P_{A,s}, P_{A,n}, P_{A,d}$,
5. three binary symbols $P_{b,s}, P_{b,n}, P_{b,d}$.

The relation that interprets P encodes the two indices of each variable. The relations that interpret $P_{A,s}, P_{A,n}, P_{A,d}$ encode the signs and the bits of the numerators and the denominators of the entries of the matrices in $\{A_i : i \in M\}$. The relations that interpret $P_{b,s}, P_{b,n}, P_{b,d}$ encode the signs and the bits of the numerators and the denominators of the rationals in $\{b_i : i \in M\}$. The representation of the circumscribed semidefinite set $(K_{A,b} \in \mathbb{R}^I, R)$ is a structure over the vocabulary $L_{\mathrm{SDP}} \dot{\cup}\, L_{\mathbb{Q}}$ whose L_{SDP}-reduct is the representation of $K_{A,b} \in \mathbb{R}^I$, and whose $L_{\mathbb{Q}}$-reduct is the representation of R.

The class of semidefinite sets together with the representation defined above form a represented class of sets, which we denote by $\mathscr{C}_{\text{SDP}}$. Similarly, the class of circumscribed semidefinite sets form a represented class of circumscribed sets denoted $\mathscr{C}^C_{\text{SDP}}$.

In [11] Dawar and Wang show the FPC-definability of the weak optimization problem for $\mathscr{C}^C_{\text{SDP}}$ with the additional assumption of the input SDP being non-empty.

Theorem 4.1 ([11])**.** *There exists an* FPC*-interpretation that takes as input a non-empty circumscribed semidefinite set* $(K_{A,b} \subseteq \mathbb{R}^I, R)$*, a vector* $c \in \mathbb{R}^I$ *and a rational* $\delta > 0$*, and outputs a vector* $x \in \mathbb{R}^I$ *such that* $x \in S(K_{A,b}, \delta)$ *and* $\langle c, x\rangle + \delta \geq \sup\{\langle c, y\rangle : y \in K_{A,b}\}$*.*

In order to do so they prove Theorem 3.2 for the special case of full-dimensional semidefinite sets and note that weak optimization reduces to weak feasibility by adding the cost vector as a constraint. They also observe that a non-empty input set can be made full-dimensional by considering an ϵ-perturbation of the constraints, for an appropriately chosen $\epsilon > 0$. Finally, they propose an FPC-interpretation for the not-so-weak separation oracle. We work out the details of a variant of their oracle construction that takes care of a missing step in their proof. This fix was first published in the preliminary report of this paper [4] and the conference version of this paper [5]. The revised version of Wang's PhD thesis [33] included the same fix.

As a consequence of Theorem 3.2 and the FPC-definability of the not-so-weak separation oracle we get the following:

Theorem 4.2. *The weak feasibility problem for circumscribed semidefinite sets is definable in* FPC*.*

4.2 Separation oracle

We show that the not-so-weak separation problem is FPC-definable for the class $\mathscr{C}_{\text{SDP}}$ of all semidefinite sets. This clearly implies the FPC-definability of the not-so-weak separation problem for $\mathscr{C}^C_{\text{SDP}}$, which is what is needed for the proof of Theorem 4.2. We begin with a few definitions and lemmas. In particular, since one of the steps of the separation procedure is a reduction to a family of LPs, we specify an encoding of an LP as a finite relational structure.

The *polytope* $K_{u,v} \subseteq \mathbb{R}^I$ is defined by a system of linear inequalities:

$$\langle u_i, x\rangle \leq v_i \quad \text{for } i \in M, \tag{4.2}$$

where x is an I-vector of variables, $u \in \mathbb{R}^{M\times I}$ is an indexed set of I-vectors, and $v \in \mathbb{R}^M$ is an indexed set of reals. If the entries of the vectors $\{u_i : i \in M\}$ and v are rational numbers, then the polytope $K_{u,v} \subseteq \mathbb{R}^I$ is represented by a three-sorted structure, with two sorts $\bar{I}$ and $\bar{M}$ for the index sets I and M, and one sort $\bar{B}$ for a domain $\{0, \ldots, N-1\}$ of bit positions that is large enough to encode all the numbers in binary. The vocabulary L_{LP} includes the following relation symbols:

1. two unary symbols I and M, for $\bar{I}$ and $\bar{M}$, respectively,

2. one binary symbol $\leq$ for the natural linear order on $\bar{B}$,
3. three ternary symbols $P_{u,s}, P_{u,n}, P_{u,d}$,
4. three binary symbols $P_{v,s}, P_{v,n}, P_{v,d}$.

The relations that interpret the symbols in point 3 encode the signs and the bits of the numerators and the denominators of the entries of the vectors in $\{u_i : i \in M\}$. The relations that interpret the symbols in point 4 encode those of the rationals in $\{v_i : i \in M\}$.

Linear programs of the form:

$$(P) \quad : \quad \inf{}_x \langle c, x\rangle \ \text{ s.t. } \langle u_i, x\rangle \leq v_i \ \text{ for } i \in M, \tag{4.3}$$

where x, u and v are as specified above and c is an I-vector, are represented similarly as polytopes. The vocabulary L_{optLP} contains three additional binary symbols $P_{c,s}, P_{c,n}, P_{c,d}$ that encode the vector c.

Theorem 4.3 ([2])**.** *There exists an* FPC*-interpretation that takes as input a linear program $P : \inf_x \langle c, x\rangle$ s.t. $\langle u_i, x\rangle \leq v_i$ for $i \in M$, and outputs an integer $b \in \{-1, 0, 1\}$, a vector $s \in \mathbb{Q}^I$ and a rational r, such that:*

1. *$b = 1$ and P is feasible but unbounded below, or*
2. *$b = 0$ and P has an optimal feasible solution of value r, and s is one, or*
3. *$b = -1$ and P is infeasible.*

We also need the following lemma from [11] showing that the smallest eigenvalue of a given symmetric matrix can be approximated in FPC:

Lemma 4.4 ([11])**.** *There exists an* FPC*-interpretation that takes as input a symmetric matrix $A \in \mathbb{Q}^{I\times I}$ and a rational $\delta > 0$ and outputs a rational $\hat{\lambda}$, such that $\hat{\lambda}$ is the approximate value of the smallest eigenvalue λ of A up to precision δ, i.e., $|\hat{\lambda} - \lambda| \leq \delta$.*

We are now ready to show the following:

Proposition 4.5. *The not-so-weak separation problem for semidefinite sets is definable in* FPC*.*

Proof. If $K_{A,b} \subseteq \mathbb{R}^I$ is a non-empty semidefinite set and $Y \in \mathbb{R}^{J\times J}$ is a symmetric matrix outside $K_{A,b}$, then either Y violates at least one of the linear inequalities that describe $K_{A,b}$, or fails to be positive semidefinite. In the former case, we get a separating hyperplane by taking the normal of the violated inequality, and a canonical one by taking the sum of all of them, as in [2]. In the latter case, the smallest eigenvalue λ of Y is negative, and if v is an eigenvector of this eigenvalue, then vv^T is a valid separating hyperplane (after normalization). Such an eigenvector would be found if we were able to find an optimal solution to the optimization problem

$$\inf{}_y \|(Y - \lambda \mathrm{I})y\|_1 \ \text{ s.t. } \ \|y\|_\infty = 1. \tag{4.4}$$

Unfortunately, this optimization problem cannot be easily phrased into an LP because the constraint $\|y\|_\infty = 1$ cannot be expressed by linear inequalities. Here is where we differ from [11]: first we relax the constraint $\|y\|_\infty = 1$ to $\|y\|_\infty \leq 1$, but then we add the condition that some component y_l is 1, and we do this for each $l \in J$ separately. Thus, for each $l \in J$, let $P(Y, \lambda, l)$ be the following optimization problem:

$$\inf_y \|(Y - \lambda \mathrm{I})y\|_1 \quad \text{s.t.} \quad \|y\|_\infty \leq 1,\ y_l = 1. \tag{4.5}$$

This we can formulate as an LP. The problem $P(Y, \lambda, l)$ may be feasible for some $l \in J$ and infeasible for some other $l \in J$, but at least one is guaranteed to be feasible. We take a solution for each feasible one and add them together to produce a canonical separating hyperplane. All this would be an accurate description of what our separation oracle does if we could compute λ exactly, but unfortunately only an approximation $\hat{\lambda}$ is available. Still, if the approximation is good enough, using $\hat{\lambda}$ in place of λ in the $P(Y, \lambda, l)$'s will do the job. We provide the details.

Let Ψ be the interpretation that takes as input a symmetric matrix $Y \in \mathbb{Q}^{J \times J}$, a rational $\delta > 0$, and a representation of $K_{A,b} \subseteq \mathbb{R}^I$ in $\mathscr{C}_{\mathrm{SDP}}$, where $A \in \mathbb{Q}^{M \times (J \times J)}$ and $b \in \mathbb{Q}^M$, and does the following:

01. given Y, δ, and $K_{A,b} \subseteq \mathbb{R}^I$ as specified,
02. compute $L := \{i \in M \colon \langle A_i, Y \rangle > b_i\}$,
03. if $|L| \neq 0$, then
04. compute $D := \|\sum_{i \in L} A_i\|_\infty$,
05. if $D \neq 0$, compute $S := \sum_{i \in L} A_i / D$, and output $(0, S)$,
06. if $D = 0$, output $(0, \mathrm{I})$,
07. else
08. compute $n := |J|$,
09. compute $\hat{\lambda}$, the smallest eigenvalue of Y up to precision $\delta/2n^2$,
10. if $\hat{\lambda} > \delta/2n^2$, output $(1, \mathrm{I})$,
11. else
12. compute $T := \{l \in J : P(Y, \hat{\lambda}, l)$ is feasible with optimum $\leq \delta/2n\}$,
13. compute $v := \{v_l \in \mathbb{Q}^J : l \in T$ and v_l is optimal for $P(Y, \hat{\lambda}, l)\}$,
14. compute $D := \|\sum_{l \in T} v_l v_l^T\|_\infty$ and $S := -\sum_{l \in T} v_l v_l^T / D$,
15. output $(0, S)$.

Let us show that Ψ is FPC-definable and satisfies the required conditions.

Step 01 is the description of the input. Steps 07 and 11 are control steps. FPC-definability of Steps 02, 03, 04, 05, 06, 08, 10, 14 and 15 follows from the ability of FPC to perform the basic arithmetic of rational numbers, compare rational numbers, and compute cardinalities of definable sets. Step 09 follows from Lemma 4.4. Below we first argue that the output of Ψ is always correct and finally that the Steps 12 and 13 are FPC-definable.

Suppose that $L = \{i \in M \colon \langle A_i, Y \rangle > b_i\} \neq \emptyset$ and let us prove that the output in Steps 05

and 06 is correct. If $\sum_{i\in L} A_i$ is the zero matrix, then we have that

$$\sum_{i\in L} b_i < \sum_{i\in L} \langle A_i, Y\rangle = \langle \sum_{i\in L} A_i, Y\rangle = 0. \tag{4.6}$$

Therefore, the feasibility region $K_{A,b}$ is empty. Indeed, every $X \in K_{A,b}$ satisfies

$$0 > \sum_{i\in L} b_i \geq \sum_{i\in L} \langle A_i, X\rangle = \langle \sum_{i\in L} A_i, X\rangle = 0, \tag{4.7}$$

which is a contradiction. Hence, for any matrix whose L_∞-norm is 1, in particular for the identity matrix I, the output $(0, \mathrm{I})$ is correct.

If $\sum_{i\in L} A_i$ is not the zero matrix, let $D = \|\sum_{i\in L} A_i\|_\infty$ and $S = \sum_{i\in L} A_i/D$. Then for every $X \in K_{A,b}$ we have that

$$\langle S, X\rangle = \langle \sum_{i\in L} \frac{A_i}{D}, X\rangle = \frac{1}{D}\sum_{i\in L}\langle A_i, X\rangle \leq \frac{1}{D}\sum_{i\in L} b_i < \frac{1}{D}\sum_{i\in L}\langle A_i, Y\rangle = \langle S, Y\rangle. \tag{4.8}$$

Moreover, the matrix S has L_∞-norm 1. So the output is correct.

Suppose that $L = \{i \in M\colon \langle A_i, Y\rangle > b_i\} = \emptyset$, $n = |J|$ and $\hat{\lambda} > \delta/2n^2$, and let us argue that the output in Step 10 is correct. Observe that, for every $i \in M$, the matrix Y satisfies $\langle A_i, Y\rangle \leq b_i$, and its smallest eigenvalue λ is positive, which means that the matrix Y is positive semidefinite. Hence, Y is in the feasibility region $K_{A,b}$ and the output is correct.

Finally, let us assume that $\hat{\lambda} \leq \delta/2n^2$. In this case, for every $l \in J$, the FPC interpretation needs to compute the optimal value and an optimal solution of the optimization problem $P(Y, \hat{\lambda}, l)$. To show that this is possible, we define an essentially equivalent linear program $P'(l)$ and use Theorem 4.3 to conclude.

To perform Steps 12 and 13 the FPC interpretation takes, for each $l \in J$, the linear program $P'(l)$ with variables $\{x_i : i \in J\} \cup \{y_i : i \in J\}$, defined by:

$$\begin{array}{lll} \inf_{x,y} & \sum_{i\in J} x_i & \\ \text{s.t.} & -x_i \leq (Yy - \hat{\lambda}y)_i \leq x_i, & \text{for every } i \in J \\ & -1 \leq y_i \leq 1, & \text{for every } i \in J \\ & y_l = 1, & \end{array}$$

where y is the vector $\{y_i : i \in J\}$. In the following, since Y and $\hat{\lambda}$ are fixed, let us write $P(l)$ instead of $P(Y, \hat{\lambda}, l)$.

Claim 4.6. *The program $P(l)$ is feasible if and only if the program $P'(l)$ is feasible and the optimal values of $P(l)$ and $P'(l)$ are the same. Moreover, if a vector $\{x_i : i \in J\}\cup\{y_i : i \in J\}$ is an optimal solution to $P'(l)$, then the vector $\{y_i : i \in J\}$ is an optimal solution to $P(l)$.*

Proof. Suppose that the feasibility region of $P(l)$ is non-empty. For every vector $y = \{y_i : i \in J\}$ in the feasibility region of $P(l)$, the vector $\{x_i : i \in J\} \cup \{y_i : i \in J\}$, where $x_i = |(Yy - \hat{\lambda}y)_i|$, belongs to the feasibility region of $P'(l)$ and its value $\sum_{i\in J} x_i = \|(Y - \hat{\lambda}I)y\|_1$

is the same as the value of $\{y_i : i \in J\}$ for $P(l)$. Therefore, the feasibility region of $P'(l)$ is non-empty and the optimal value opt' of $P'(l)$ is smaller or equal to the optimal value opt of $P(l)$.

Suppose that the feasibility region of $P'(l)$ is non-empty, and take an optimal solution $\{x_i : i \in J\} \cup \{y_i : i \in J\}$ for $P'(l)$. It holds that $\|y\|_\infty = 1$ and $y_l = 1$, so the vector y is in the feasibility region of $P(l)$. Therefore, the feasibility region of $P(l)$ is non-empty, and $opt \leq \|(Y - \hat{\lambda}I)y\|_1$. Moreover, for every $i \in J$, we have that $|(Yy - \hat{\lambda}y)_i| \leq x_i$ so $\|(Y - \hat{\lambda}I)y\|_1 \leq \sum_{i \in J} x_i = opt'$. On the other hand we know that $opt' \leq opt$. To summarize

$$opt \leq \|(Y - \hat{\lambda}I)y\|_1 \leq \sum_{i \in J} x_i = opt' \leq opt. \tag{4.9}$$

Hence, the vector y is an optimal solution for $P(l)$ and the optimal values are the same. □

To perform Steps 12 and 13 the FPC interpretation computes, for every $l \in J$, an optimal solution and the optimal value of the optimization problem $P(l)$, by computing an optimal solution and the optimal value of the linear program $P'(l)$ via Theorem 4.3, and projecting the output to the variables $\{y_i : i \in J\}$.

We now show that the set T, defined in Step 12, is non-empty, and that $\|\sum_{l \in T} v_l v_l^T\|_\infty \neq 0$. It follows that the output matrix S in Step 14 is well defined.

Claim 4.7. *$T \neq \emptyset$.*

Proof. Let v be an eigenvector of Y with the smallest eigenvalue λ, and let $\|v\|_\infty = 1$. We have the following

$$\begin{aligned} \|(Y - \hat{\lambda}I)v\|_1 &= \|(Y - \lambda I)v - (\hat{\lambda} - \lambda)Iv\|_1 \leq \\ &\leq \|(Y - \lambda I)v\|_1 + \|(\hat{\lambda} - \lambda)Iv\|_1 = \\ &= \|(\hat{\lambda} - \lambda)Iv\|_1 \leq \frac{\delta}{2n^2} n \|v\|_\infty = \frac{\delta}{2n}. \end{aligned} \tag{4.10}$$

If there exists $l \in J$ such that $v_l = 1$, then $v \in P(l)$ and $T \neq \emptyset$. Otherwise, there exists $l \in J$ such that $v_l = -1$. Then $-v \in P(l)$ and we are done as well. □

Claim 4.8. *$1 \leq \|\sum_{l \in T} v_l v_l^T\|_\infty \leq |T|$.*

Proof. Observe that for every $l \in J$ all the main diagonal entries of the matrix $v_l v_l^T$ are squares and since $\|v_l\|_\infty = 1$, at least one of those entries is equal 1. Therefore,

$$\|\sum_{l \in T} v_l v_l^T\|_\infty \geq 1, \tag{4.11}$$

and on the other hand,

$$\|\sum_{l \in T} v_l v_l^T\|_\infty \leq \sum_{l \in T} \|v_l v_l^T\|_\infty = |T|. \tag{4.12}$$

□

Finally, let us show that the output $(0, S)$ in Step 15 is correct.

Claim 4.9. *For every $l \in T$, let v_l be the optimal solution of $P(l)$. Then for every $X \in K_{A,b}$,*

$$\langle -v_l v_l^T, Y\rangle + \frac{\delta}{n} \geq \langle -v_l v_l^T, X\rangle. \tag{4.13}$$

Proof. Take $X \in K_{A,b}$. Since the matrix X is positive semidefinite, $\langle -v_l v_l^T, X\rangle = -v_l^T X v_l \leq 0$. We will show that $\langle -v_l v_l^T, Y\rangle + \delta/n \geq 0$. It holds that

$$\begin{aligned}\langle -v_l v_l^T, Y\rangle &= -v_l^T Y v_l = -v_l^T(\hat{\lambda} I + (Y - \hat{\lambda} I))v_l = \\ &= -\hat{\lambda} v_l^T v_l - v_l^T(Y - \hat{\lambda} I)v_l \geq -\hat{\lambda} v_l^T v_l - |v_l^T(Y - \hat{\lambda} I)v_l| \geq \\ &\geq -\hat{\lambda} v_l^T v_l - \|v_l\|_\infty \|(Y - \hat{\lambda} I)v_l\|_1 \geq -\hat{\lambda} v_l^T v_l - \frac{\delta}{2n}.\end{aligned} \tag{4.14}$$

It follows that

$$\langle -v_l v_l^T, Y\rangle + \frac{\delta}{n} \geq -\hat{\lambda} v_l^T v_l + \frac{\delta}{2n}. \tag{4.15}$$

Now if $\hat{\lambda} \leq 0$, then $-\hat{\lambda} v_l^T v_l + \delta/2n = -\hat{\lambda}\|v_l\|_2^2 + \delta/2n \geq \delta/2n > 0$. Otherwise $0 < \hat{\lambda} \leq \delta/2n^2$, and

$$\hat{\lambda} v_l^T v_l \leq \frac{\delta}{2n^2}\|v_l\|_2^2 \leq \frac{\delta}{2n^2}(\sqrt{n}\|v_l\|_\infty)^2 = \frac{\delta}{2n^2} n = \frac{\delta}{2n}. \tag{4.16}$$

Hence, $-\hat{\lambda} v_l^T v_l + \delta/2n \geq -\delta/2n + \delta/2n = 0$. □

We finish the proof by showing that for every $X \in K_{A,b}$,

$$\langle S, Y\rangle + \delta \geq \langle S, X\rangle. \tag{4.17}$$

Let X be any matrix in $K_{A,b}$. From now on, let $D = \|\sum_{l\in T} v_l v_l^T\|_\infty$. Recall from Claim 4.8 that $1 \leq D \leq |T|$. It holds that

$$\begin{aligned}\langle S, Y\rangle &= \langle -\sum_{l\in T} \frac{v_l v_l^T}{D}, Y\rangle = \frac{1}{D}\sum_{l\in T}\langle -v_l v_l^T, Y\rangle \geq \frac{1}{D}\sum_{l\in T}(\langle -v_l v_l^T, X\rangle - \frac{\delta}{n}) = \\ &= \langle -\sum_{l\in T} \frac{v_l v_l^T}{D}, X\rangle - \frac{|T|}{D}\frac{\delta}{n} = \langle S, X\rangle - \frac{|T|}{n}\frac{\delta}{D} \geq \langle S, X\rangle - \delta,\end{aligned} \tag{4.18}$$

where the first inequality follows from (4.13) in Claim 4.9 and the last inequality follows from the fact that $|T| \leq n$ and $D \geq 1$. □

4.3 Exact feasibility

We use Theorem 4.2 to prove the main result of this section:

Theorem 4.10. *The exact feasibility problem for semidefinite sets is definable in $\mathrm{C}^{\omega}_{\infty\omega}$.*

We begin the proof by relating the problem of exact feasibility to the subject of Theorem 4.2, i.e., the weak feasibility problem for circumscribed semidefinite sets.

For any $R > 0$ and any semidefinite set $K_{A,b}$, the *R-restriction* of $K_{A,b}$ is the set of all those points in $K_{A,b}$ whose L_∞-norm is bounded by R, i.e., it is the semidefinite set given by:

$$\begin{array}{lcll} \langle A_i, X\rangle & \leq & b_i & \text{for } i \in M, \\ X_{\{i,j\}} & \leq & R & \text{for } i,j \in J, \\ -X_{\{i,j\}} & \leq & R & \text{for } i,j \in J, \\ X \succeq 0. & & & \end{array}$$

For any $\epsilon > 0$ and any semidefinite set $K_{A,b}$, the *ϵ-relaxation* of $K_{A,b}$ is the semidefinite set given by:

$$\begin{array}{lcll} \langle A_i, X\rangle & \leq & b_i + \epsilon & \text{for } i \in M \\ X \succeq 0. & & & \end{array}$$

Since an R-restriction of a semidefinite set is a semidefinite set itself, it makes sense to talk about its ϵ-relaxation. The question of emptiness for ϵ-relaxations of R-restrictions of semidefinite sets is closely linked to the exact feasibility problem under consideration. Recall the Cantor Intersection Theorem: If $K_1 \supseteq K_2 \supseteq \cdots$ is a decreasing nested sequence of non-empty compact subsets of $\mathbb{R}^n$, then the intersection $\bigcap_{i\geq 1} K_i$ is non-empty. We use it for the following lemma.

Lemma 4.11. *A semidefinite set $K_{A,b}$ is non-empty if and only if there exists a positive rational R such that for every positive rational ϵ it holds that the ϵ-relaxation of the R-restriction of $K_{A,b}$ is non-empty.*

Proof. Assume that $K_{A,b}$ is non-empty and let x be a point in it. Let R be a rational bigger than $\|x\|_\infty$. Then x is also in the R-restriction of $K_{A,b}$, and therefore in the ϵ-relaxation of the R-restriction of $K_{A,b}$ for every positive rational ϵ.

Assume now that R is a positive rational such that the ϵ-relaxation of the R-restriction of $K_{A,b}$ is non-empty for every positive rational ϵ. For each positive integer m, let K_m be the $1/m$-relaxation of the R-restriction of $K_{A,b}$. Each K_m is closed and bounded, hence compact. Moreover $K_1 \supseteq K_2 \supseteq \cdots$, i.e., the sets K_m form a decreasing nested sequence of non-empty subsets of $\mathbb{R}^I$. It therefore follows from the Cantor Intersection Theorem that $\bigcap_{m\geq 1} K_m$ is non-empty. The claim follows from the observation that $\bigcap_{m\geq 1} K_m$ is indeed the R-restriction of $K_{A,b}$. □

It follows from Theorem 4.2 that the emptiness problem for ϵ-relaxations of R-restrictions of semidefinite sets is definable in FPC in the following sense.

Proposition 4.12. *There exists a formula ψ of* FPC *such that if $\mathbb{A}$ is a structure over $L_{\mathrm{SDP}} \dot\cup L_{\mathbb{Q}} \dot\cup L_{\mathbb{Q}}$, representing a semidefinite set $K_{A,b} \subseteq \mathbb{R}^I$ and two positive rational numbers R and ϵ, then:*

1. *if* $\mathbb{A} \models \psi$*, then the* ϵ*-relaxation of the* R*-restriction of* $K_{A,b}$ *is non-empty,*
2. *if* $\mathbb{A} \not\models \psi$*, then the* R*-restriction of* $K_{A,b}$ *is empty.*

Proof. Let Φ be an FPC-interpretation that witnesses that the weak feasibility problem for the class of circumscribed semidefinite sets is FPC-definable. The formula ψ takes as input the representation of a semidefinite set $K_{A,b} \subseteq \mathbb{R}^I$, a rational $\epsilon > 0$ and a rational $R > 0$, and does the following:

01. given $K_{A,b} \subseteq \mathbb{R}^I$, ϵ and R as specified,
02. compute $k := |I|$,
03. compute $R' := \lceil \sqrt{k(R+\epsilon)^2} \rceil$,
04. compute a representation of K, the ϵ-relaxation of the R-restriction of $K_{A,b}$,
05. compute $m := \max\{\|A_i\|_2 : i \in M\} \cup \{1\}$,
06. compute $\delta = \epsilon^k/(k!(2km)^k)$,
07. compute $(b, x) := \Phi((K, R'), \delta)$,
08. if $b = 1$ output $\top$,
09. if $b = 0$ output $\bot$.

This procedure is clearly FPC-definable. In order to prove correctness we will need the following lemma.

Lemma 4.13. *Let* $A \in \mathbb{R}^{M\times(J\times J)}$*,* $b \in \mathbb{R}^M$*,* $I = \{\{i,j\} : i,j \in J\}$*,* $k = |I|$*, and* $m = \max\{\|A_i\|_2 : i \in M\} \cup \{1\}$*. For any* $\epsilon > 0$*, if the semidefinite set* $K_{A,b} \in \mathbb{R}^I$ *is non-empty, then its* ϵ*-relaxation has volume greater than* $\delta = \epsilon^k/(k!(2km)^k)$*.*

Proof. Take $\epsilon_1 = \epsilon/2km$. Let Y be an element of $K_{A,b}$. We will show that the ϵ_1-ball around $Y + \epsilon_1\mathrm{I}$ is included in the ϵ-relaxation of $K_{A,b}$. It will follow that the volume of the ϵ-relaxation of $K_{A,b}$ is at least $\epsilon_1^k V_k$, where V_k is the volume of a 1-ball around Y in the k-dimensional real vector space. Since $V_k > 1/k!$ this finishes the proof.

Suppose that $T \in S(Y + \epsilon_1\mathrm{I}, \epsilon_1)$. This means that $T = Y + \epsilon_1\mathrm{I} + Z$, where $\|Z\|_2 \leq \epsilon_1$. We start by showing that T is positive semidefinite. Let v be a vector whose L_2-norm is 1. It holds that

$$\begin{aligned} v^T T v = v^T(Y + \epsilon_1\mathrm{I} + Z)v = v^T Y v + \epsilon_1 v^T \mathrm{I} v + v^T Z v \geq \\ \geq 0 + \epsilon_1\|v\|_2^2 + \langle vv^T, Z\rangle \geq \epsilon_1 - |\langle vv^T, Z\rangle| \geq \\ \geq \epsilon_1 - \|vv^T\|_2\|Z\|_2 = \epsilon_1 - \|v\|_2^2\|Z\|_2 \geq \epsilon_1 - \epsilon_1 = 0. \end{aligned} \tag{4.19}$$

Moreover, for every $i \in M$, we have

$$\begin{aligned}
\langle A_i, T\rangle - b_i &= \langle A_i, Y\rangle + \langle A_i, \epsilon_1 \mathrm{I}\rangle + \langle A_i, Z\rangle - b_i \leq \\
&\leq \langle A_i, \epsilon_1 \mathrm{I}\rangle + \langle A_i, Z\rangle \leq |\langle A_i, \epsilon_1 \mathrm{I}\rangle| + |\langle A_i, Z\rangle| \leq \\
&\leq \epsilon_1 \|A_i\|_2 \|\mathrm{I}\|_2 + \|A_i\|_2 \|Z\|_2 \leq \\
&\leq \epsilon_1 \|A_i\|_2 \sqrt{k} + \|A_i\|_2 \epsilon_1 = \\
&= \frac{\epsilon \|A_i\|_2 \sqrt{k}}{2km} + \frac{\|A_i\|_2 \epsilon}{2km} \leq \\
&\leq \frac{\epsilon}{2\sqrt{k}} + \frac{\epsilon}{2k} \leq \epsilon,
\end{aligned} \tag{4.20}$$

where the first inequality follows from the fact that Y is an element of $K_{A,b}$ hence, for every $i \in M$, it satisfies $\langle A_i, Y\rangle \leq b_i$, the third inequality follows from the Cauchy-Schwartz inequality $|\langle x, y\rangle| \leq \|x\|_2 \|y\|_2$, the second to last inequality follows from the fact that $m = \max\{\|A_i\|_2 : i \in M\} \cup \{1\}$, and the last inequality follows from the fact that $k = |I| \geq 1$. □

We are now ready to conclude the proof. Observe that the L_∞-norm of any point that belongs to the ϵ-relaxation of the R-restriction of a semidefinite set is bounded by $R + \epsilon$, therefore the pair (K, R') computed in Steps 03 and 04 is a representation of a circumscribed semidefinite set. Let (b, x) be the pair computed in Step 07.

If $b = 1$, then there exists a point in $S(K, \delta)$, which in particular means that K is non-empty, so the output in Step 08 is correct. If $b = 0$, then we know that the volume of K is at most δ. The inequalities that define K have the form $\langle A_i, X\rangle \leq b_i + \epsilon$ for $i \in M$, and $X_{\{i,j\}} \leq R + \epsilon$ or $-X_{\{i,j\}} \leq R + \epsilon$ for $i, j \in J$. The maximum 2-norm of the normals of these inequalities and 1 is $m = \max\{\|A_i\|_2 : i \in M\} \cup \{1\}$, so Lemma 4.13 applies. This means that K is empty, and the output in Step 09 is correct. □

To finish the proof of Theorem 4.10 we show a technical lemma that may sound a bit surprising at first: it sounds as if it was stating that $\mathrm{C}^k_{\infty\omega}$-definability is closed under second-order quantification over unbounded domains, which cannot be true. However, on closer look, the lemma states this *only* if the vocabularies of the quantified and the body parts of the formula are totally disjoint. In particular, this means that the domains of the sorts in the quantified and body parts of the formula stay unrelated *except* through the counting mechanism of $\mathrm{C}^k_{\infty\omega}$.

Note for the record that if L and K are two many-sorted vocabularies with disjoint sorts, then obviously the vocabulary $L \cup K$ does not contain any relation symbol whose type mixes the sorts of L and K. If $\mathscr{A}$ is a class of $L \cup K$-structures and $\mathscr{B}$ is a class of K-structures, we use the notation $\exists\mathscr{B} \cdot \mathscr{A}$ to denote the class of all finite L-structures $\mathbb{A}$ for which there exists a structure $\mathbb{B} \in \mathscr{B}$ such that $\mathbb{A} \dot\cup \mathbb{B} \in \mathscr{A}$. In words, this is the set of finite structures that can be disjointly extended and expanded by a structure in $\mathscr{B}$ to a structure in $\mathscr{A}$. Similarly, we use $\forall\mathscr{B} \cdot \mathscr{A}$ to denote the class of all finite L-structures $\mathbb{A}$ such that for all structures $\mathbb{B} \in \mathscr{B}$ we have that $\mathbb{A} \dot\cup \mathbb{B} \in \mathscr{A}$. In words, this is the set of finite structures all whose disjoint extensions and expansions by a structure in $\mathscr{B}$ are in $\mathscr{A}$.

Lemma 4.14. *Let L and K be many-sorted vocabularies with disjoint sorts, let $\mathscr{A}$ be a class of finite $L \cup K$-structures, and let $\mathscr{B}$ be a class of finite K-structures. If $\mathscr{A}$ is $\mathrm{C}^k_{\infty\omega}$-definable, then the classes of L-structures $\exists\mathscr{B} \cdot \mathscr{A}$ and $\forall\mathscr{B} \cdot \mathscr{A}$ are also $\mathrm{C}^k_{\infty\omega}$-definable.*

Proof. The proof is a simple *Booleanization* trick to replace the finite quantifiers $\exists^{\geq i}$ over the sorts in K by finite propositional formulas, followed by replacing $\exists\mathscr{B}$ and $\forall\mathscr{B}$ by infinite disjunctions and conjunctions, respectively, indexed by the structures in $\mathscr{B}$. We provide the details. Let ϕ be a formula of the many-sorted vocabulary $L \cup K$ with all variables of the L-sorts among $x_1, \ldots, x_k$, and all variables of the K-sorts among $y_1, \ldots, y_k$. Note that since L and K have disjoint sorts, all the atomic subformulas of ϕ have all its variables among $x_1, \ldots, x_k$ or all its variables among $y_1, \ldots, y_k$. In other words, there are no atomic subformulas with mixed x-y variables. For every finite K-structure $\mathbb{B}$ with domain B and every $b = (b_1, \ldots, b_k) \in B^k$, let $\phi(\mathbb{B}, b)$ be the *Booleanization* of ϕ with respect to the atomic interpretation of K given by $\mathbb{B}$, the domain of quantification B for the variables of the K-sorts, and the free-variable substitution $x := b$. Formally, using the notation $[E]$ for the truth value of the statement E, the formula $\phi' = \phi(\mathbb{B}, b)$ is defined inductively:

1. if $\phi = R(x_{i_1}, \ldots, x_{i_\ell})$ with $R \in L \cup \{=\}$, define $\phi' = \phi$,
2. if $\phi = R(y_{i_1}, \ldots, y_{i_\ell})$ with $R \in K \cup \{=\}$, define $\phi' = [(b_{i_1}, \ldots, b_{i_\ell}) \in R(\mathbb{B})]$,
3. if $\phi = \neg\theta$, define $\phi' = \neg\theta(\mathbb{B}, b)$,
4. if $\phi = \bigwedge_i \theta_i$, define $\phi' = \bigwedge_i \theta_i(\mathbb{B}, b)$,
5. if $\phi = \exists^{\geq t} x_i(\theta)$, define $\phi' = \exists^{\geq t} x_i(\theta(\mathbb{B}, b))$,
6. if $\phi = \exists^{\geq t} y_i(\theta)$, define

$$\phi' = \bigvee_{c \in B^t} \Big(\bigwedge_{\substack{j,j' \in [t] \\ j \neq j'}} [c_j \neq c_{j'}] \wedge \bigwedge_{j \in [t]} \theta(\mathbb{B}, b[i/c_j]) \Big). \tag{4.21}$$

Since there are no atomic subformulas with mixed x-y variables, the definition covers all cases. The construction of $\phi(\mathbb{B}, b)$ was designed so that for every finite $(L \cup K)$-structure $\mathbb{C}$ with L- and K-reducts $\mathbb{A}$ and $\mathbb{B}$ with domains A and B, respectively, every $a \in A^k$ and every $b \in B^k$, it holds that $\mathbb{C} \models \phi[a, b]$ if and only if $\mathbb{A} \models \phi(\mathbb{B}, b)[a]$. Now, if ϕ is an $(L \cup K)$-sentence, define $\phi(\mathbb{B}) := \bigvee_{b \in B^k} \phi(\mathbb{B}, b)$ and $\phi^\exists := \bigvee_{\mathbb{B} \in \mathscr{B}} \phi(\mathbb{B})$. It follows from the definitions that $\phi^\exists$ defines $\exists\mathscr{B} \cdot \mathscr{A}$. Similarly, defining $\phi^\forall := \bigwedge_{\mathbb{B} \in \mathscr{B}} \phi(\mathbb{B})$ works for $\forall\mathscr{B} \cdot \mathscr{A}$. □

We put everything together in the proof of Theorem 4.10.

Proof of Theorem 4.10. Let ψ be the $L_{\mathrm{SDP}} \dot\cup L_{\mathbb{Q}} \dot\cup L_{\mathbb{Q}}$-formula of FPC defined in Proposition 4.12. Let l be the number of variables in ψ. By the translation from l-variable FPC to $\mathrm{C}^{2l}_{\infty\omega}$ (see Section 2), there exists an $L_{\mathrm{SDP}} \dot\cup L_{\mathbb{Q}} \dot\cup L_{\mathbb{Q}}$-formula τ of $\mathrm{C}^{2l}_{\infty\omega}$ defining the same class $\mathscr{A}$ of finite structures. The vocabulary of $\mathscr{A}$ is a disjoint union of L_{SDP} and two copies of $L_{\mathbb{Q}}$. Hence, all three of those vocabularies have disjoint sorts. Let $\mathscr{B}_R$ be the class of finite structures which are representations of positive rational numbers over the first copy of $L_{\mathbb{Q}}$,

and let $\mathscr{B}_\epsilon$ be the class of finite structures which are representations of positive rational numbers over the second copy of $L_{\mathbb{Q}}$. By Lemma 4.14 the class $\forall\mathscr{B}_\epsilon\cdot\mathscr{A}$, and hence $\exists\mathscr{B}_R\cdot\forall\mathscr{B}_\epsilon\cdot\mathscr{A}$, is also $\mathrm{C}^{2l}_{\infty\omega}$-definable. Let ϕ be the L_{SDP}-formula of $\mathrm{C}^{2l}_{\infty\omega}$ defining this last class. Lemma 4.11 implies that ϕ defines the exact feasibility problem for semidefinite sets. □

5 Sums-of-Squares Proofs and the Lasserre Hierarchy

In this section we develop the descriptive complexity of the problem of deciding the existence of degree-bounded SOS proofs. Along the way we discuss the relationship between the Lasserre hierarchy of SDP relaxations and SOS refutations, and how 0/1-valued variables ensure that they satisfy strong duality. These results will be used in the next section. The strong duality property will also imply that SOS refutations exist if and only if *approximate* SOS refutations exist, for a notion of approximate SOS refutation that we introduce. This will be used to complement the descriptive complexity results for SOS proofs by getting a stronger upper bound in the case of refutations.

5.1 Descriptive Complexity of SOS Proofs

We begin with a few definitions. Let $x_1,\dots,x_n$ be a set of variables. In the following whenever we talk about polynomials or monomials we mean polynomials and monomials over the set of variables $x_1,\dots,x_n$ and real or rational coefficients. By B_n we denote the set containing the following polynomials:

$$1,\quad x_i,\quad 1-x_i,\quad x_i^2-x_i,\quad x_i-x_i^2,\qquad \text{for every } i\in[n]. \tag{5.1}$$

We refer to the inequalities $p\geq 0$ for $p\in B_n$ as *Boolean axioms*. A polynomial s is a *sum of squares of polynomials* if it has the form $s=\sum_{i\in[l]} r_i^2$, for some polynomials $r_1,\dots,r_l$. For a set of polynomials Q and a polynomial p, a *Sums-of-Squares (SOS) proof* of $p\geq 0$ from Q is an indexed set of polynomials $\{s_q : q\in\bar{Q}\}$ that satisfy an identity

$$\sum_{q\in\bar{Q}} q s_q = p, \tag{5.2}$$

where, $\bar{Q}=Q\cup B_n$ and for every $q\in\bar{Q}$, the polynomial s_q is a sum of squares of polynomials. The degree of the proof is defined as $\max\{\deg(qs_q) : q\in\bar{Q}\}$, where, for a polynomial p, the notation $\deg(p)$ denotes the degree of p.

One should think about the set of polynomials Q as representing a system of polynomial inequalities $\{q\geq 0 : q\in Q\}$. The identity (5.2) implies that any 0/1-solution to this system satisfies also the inequality $p\geq 0$. Therefore, if $p=-1$, a proof certifies that the system $\{q\geq 0 : q\in Q\}$ has no 0/1-solutions. This is why we call it a *refutation* of Q. The definition of SOS as a proof system is sometimes attributed to Grigoriev and Vorobyov [15]. We note that, in the case of refutations, our definition is the special case their Positivstellensatz [15, Definition 2] in which all non-trivial products of q_j's have 0

multipliers. Unlike theirs, our proof system includes the Boolean axioms by default, thus ensuring completeness even for proofs of polynomial inequalities over 0/1-valued variables.

We consider the problem of deciding the existence of SOS proofs and refutations of a fixed degree $2d$ for a set of polynomials given as input. The first easy observation is that the proof-existence problem can be reduced to the exact feasibility problem for semidefinite sets, and the reduction can be done in FPC. Then we ask whether the exactness condition in the feasibility problem for semidefinite sets can be relaxed, and we achieve this for refutations. In other words:

1. Proof-existence reduces in FPC to exact feasibility for semidefinite sets.
2. Refutation-existence reduces in FPC to weak feasibility for semidefinite sets.

We note that, in both cases, the semidefinite sets in the outcome of this reduction are not circumscribed. Roughly, this is because their elements are vectors encoding sequences of coefficients of sum of squares polynomials that form a valid proof. By the results in [22], the bit-complexity of these coefficients may not be polynomially bounded, not even for proofs over Boolean variables [27], nor for refutations over Boolean variables [17].

As stated, point 1. above is almost a reformulation of the problem. In order to prove point 2. we need to develop a notion of approximate refutation, and combine it with a strong duality theorem that characterizes the existence of SOS refutations in terms of so-called *pseudoexpectations* [6]. We note that the strong duality theorem that we need relies on the assumption that the Boolean axioms are included in the definition of SOS proof.

Finally, we combine these FPC reductions with the results of the previous section in order to get the following:

Corollary 5.1. *For every fixed positive integer d, the problems of deciding the existence of SOS proofs of degree $2d$, and SOS refutations of degree $2d$, are $\mathrm{C}^{\omega}_{\infty\omega}$-definable. Moreover, there exists a constant c, independent of d, such that the defining formulas are in $\mathrm{C}^{cd}_{\infty\omega}$.*

As usual with descriptive complexity results like these, we need to fix some encoding of the input as finite relational structures. In this case the inputs are indexed sets of polynomials, where each polynomial is an indexed set of monomials and coefficients. The exact choice of encoding is not very essential, but we propose one for concreteness.

Let I be an index set for variables and let $\{x_i : i \in I\}$ be a set of formal variables. A *monomial* is a product of variables. For $\alpha = (\alpha_i : i \in I) \in \mathbb{N}^I$, we use the notation x^α to denote the monomial that has *degree* α_i on variable x_i. We write $|\alpha|$ for the degree $\sum_{i\in I} \alpha_i$ of the monomial x^α. A polynomial is a finite linear combination of monomials, i.e., a formal expression of the form $\sum_\alpha c_\alpha x^\alpha$ in which all but finitely many of the coefficients c_α are zero. A polynomial p with rational coefficients is represented by a three-sorted structure, with a sort $\bar{I}$ for the index set I, a second sort $\bar{M}$ for the finite set of monomials that have non-zero coefficients in p, and a third sort $\bar{B}$ for a domain $\{0, \ldots, N-1\}$ of bit positions, where N is large enough to encode all the coefficients of p and all the degrees of its monomials in binary. The vocabulary of this structure has one unary relation symbol I for $\bar{I}$, one binary relation symbol $\leq$ for the natural linear order on $\bar{B}$, three binary relations symbols P_s, P_n, and P_d

of type $\bar{M} \times \bar{B}$ that encode, for each monomial, the sign, the bits of the numerator, and the bits of the denominator of its coefficient, respectively, and a ternary relation symbol D of type $\bar{M} \times \bar{I} \times \bar{B}$ that encodes, for each monomial and each variable, the bits of the degree of this variable in the monomial.

Let J be an index set for polynomials and let $\{p_j : j \in J\}$ be a set of polynomials on the variables $\{x_i : i \in I\}$. Such a set is represented by a four-sorted structure, with a sort $\bar{J}$ for the index set J, and the three sorts $\bar{I}, \bar{M}, \bar{B}$ of the previous paragraph. The vocabulary for this structure has one unary relation symbol J for $\bar{J}$, one binary relation symbol $\leq$ for the natural linear order on $\bar{B}$, three ternary relation symbols P_s, P_n, and P_d of type $\bar{J} \times \bar{M} \times \bar{B}$ that encode, for each $j \in J$, the coefficients of the monomials in p_j, and a four-ary relation symbol of type $\bar{J} \times \bar{M} \times \bar{I} \times \bar{B}$ that encodes, for each $j \in J$, the degrees of the variables in the monomials in p_j.

5.2 The Lasserre hierarchy

There is a sense in which sums-of-squares proofs can be seen as the *dual solutions* in a hierarchy of semidefinite programming relaxations of an associated optimization problem. This correspondence will be used explicitly in Subsection 5.4. Some of the concepts we introduce now will also be useful in the next Subsection 5.3.

We adopt the setting in [19]. For a set of polynomials $\{q_0, q_1, \ldots, q_k\}$, we denote the following polynomial optimization problem by $\mathrm{POP}(q_0; \{q_1, \ldots, q_k\})$:

$$\text{(POP)} \quad : \quad \inf_x q_0(x) \ \text{ s.t. } \ q_i(x) \geq 0 \ \text{ for } i \in [k]. \tag{5.3}$$

Take a positive integer d. Recall that we use the notation x^α, where $\alpha = (\alpha_i : i \in [n]) \in \mathbb{N}^n$, to denote the monomial that has degree α_i on variable x_i. We identify the monomial x^α with its vector of degrees α. By M_d we denote the matrix indexed by monomials of degree at most d defined by $(M_d)_{\alpha,\beta} = x^{\alpha+\beta}$. For every monomial x^α, we introduce a variable y_α and by $M_d(y)$ we denote the corresponding matrix of variables, defined by $(M_d(y))_{\alpha,\beta} = y_{\alpha+\beta}$. More generally, for any polynomial $q = \sum_\gamma c_\gamma x^\gamma$, the matrix $M_{q,d}$, indexed by monomials of degree at most d, is defined by $M_{q,d} = qM_d$, i.e., $(M_{q,d})_{\alpha,\beta} = qx^{\alpha+\beta}$. The corresponding matrix of variables $M_{q,d}(y)$ is defined by $(M_{q,d}(y))_{\alpha,\beta} = \sum_\gamma c_\gamma y_{\alpha+\beta+\gamma}$. Observe that the entries of the matrix $M_{q,d}$ are polynomials of degree at most $2d + \deg(q)$, while the entries of the matrix $M_{q,d}(y)$ are the corresponding linear combinations of variables. Note also that $M_{1,d} = M_d$ and $M_{1,d}(y) = M_d(y)$. For every variable y_α, consider the coefficients of y_α in the matrix $M_{q,d}(y)$. Those coefficients form a matrix which we denote by $A_{q,d,\alpha}$. Formally, for $|\alpha| \leq 2d + \deg(q)$, the matrices $A_{q,d,\alpha}$ are defined as the real matrices satisfying $M_{q,d}(y) = \sum_\alpha y_\alpha A_{q,d,\alpha}$ or equivalently $M_{q,d} = \sum_\alpha x^\alpha A_{q,d,\alpha}$. Finally, for any polynomial q, by d_q we denote the biggest integer satisfying $2d_q + \deg(q) \leq 2d$.

Let Q be a set of polynomials and let $q_0 = \sum_\alpha x^\alpha$ be a polynomial. For any positive integer d, the *level-d Lasserre SDP relaxation* of the polynomial optimization problem $\mathrm{POP}(q_0; Q)$ is the pair of semidefinite programs (P_d, D_d), where P_d is the *primal* semidef-

inite program:

$$\begin{array}{ll} \inf_y & \sum_\alpha a_\alpha y_\alpha \\ \text{s.t.} & y_\emptyset = 1 \\ & M_{q,d_q}(y) \succeq 0 \quad \text{for } q \in Q \end{array} \tag{5.4}$$

and D_d is the *dual* semidefinite program:

$$\begin{array}{ll} \sup_{z,Z} & z \\ \text{s.t.} & \sum_{q\in Q}\langle A_{q,d_q,\emptyset}, Z_q\rangle = a_\emptyset - z \\ & \sum_{q\in Q}\langle A_{q,d_q,\alpha}, Z_q\rangle = a_\alpha \qquad \text{for } 1 \le |\alpha| \le 2d \\ & Z_q \succeq 0 \text{ for } q \in Q \end{array} \tag{5.5}$$

Weak SDP duality implies that the optimal value of P_d is always greater or equal than the optimal value of D_d. The main theorem in [19] establishes a condition which guarantees strong duality for primal and dual SDP problems in the Lasserre hierarchy.

Theorem 5.2 ([19])**.** *If* $\mathrm{POP}(q_0; Q)$ *is a polynomial optimization problem where one of the inequalities describing the feasibility region is* $R^2 - \sum_{i\in[n]} x_i^2 \ge 0$*, then for every positive integer* d*, the optimal values of* P_d *and* D_d *are equal.*

Strong duality for primal and dual problems implies, in particular, that P_d is infeasible if and only if D_d is unbounded above and, analogously, P_d is unbounded below if and only if D_d is infeasible.

The polynomial optimization problem $\mathrm{POP}(q_0; Q)$ is called *encircled* if a polynomial $R^2 - \sum_{i\in[n]} x_i^2$ can be obtained as a non-negative linear combination of polynomials from Q of degree at most 2. The following lemma implies strong duality for primal and dual SDP problems in the Lasserre hierarchy for encircled polynomial optimization problems.

Lemma 5.3. *Let* Q *be a set of polynomials and let* $p = \sum_{q\in Q} c_q q$ *be a non-negative linear combination of polynomials from* Q*, such that* $\deg(p) = \max\{\deg(q) : c_q > 0\}$*. For some polynomial* q_0*, let* (P_d, D_d) *and* (P'_d, D'_d) *be the level-*d *Lasserre SDP relaxations of* $\mathrm{POP}(q_0; Q)$ *and* $\mathrm{POP}(q_0; Q \cup \{p\})$*, respectively. The optimal values of* P_d *and* P'_d*, as well as the optimal values of* D_d *and* D'_d *are equal.*

Proof. Let $q_0 = \sum_\alpha a_\alpha x^\alpha$ and let d be some positive integer.

The primal P'_d is the following semidefinite program:

$$\begin{array}{ll} \inf_y & \sum_\alpha a_\alpha y_\alpha \\ \text{s.t.} & y_\emptyset = 1 \\ & M_{q,d_q}(y) \succeq 0 \quad \text{for } q \in Q \\ & M_{p,d_p}(y) \succeq 0 \end{array} \tag{5.6}$$

Let $P = \{q \in Q : c_q > 0\}$. Note that since $\deg(p) = \max\{\deg(q) : q \in P\}$, for every $q \in P$, we have $d_p \le d_q$. For each $q \in P$, by $M'_{q,d_q}(y)$ let us denote the principal submatrix of $M_{q,d_q}(y)$ obtained by removing the rows and columns indexed by monomials of degree greater than d_p. Observe that $M_{p,d_p}(y) = \sum_{q\in P} c_q M'_{q,d_q}(y)$. Since the constraints $\{M_{q,d_q}(y) \succeq$

$0 : q \in P\}$ imply the constraint $M_{p,d_p}(y) = \sum_{q\in P} c_q M'_{q,d_q}(y) \succeq 0$, the feasibility regions, and therefore also the optimal values, of P_d and P'_d are the same.

The dual D'_d is the following semidefinite program:

$$\begin{array}{lll} \sup_{z,Z} & z & \\ \text{s.t.} & \sum_{q\in Q}\langle A_{q,d_q,\emptyset}, Z_q\rangle + \langle A_{p,d_p,\emptyset}, Z_p\rangle = a_\emptyset - z & \\ & \sum_{q\in Q}\langle A_{q,d_q,\alpha}, Z_q\rangle + \langle A_{p,d_p,\alpha}, Z_p\rangle = a_\alpha & \text{for } 1 \le |\alpha| \le 2d \\ & Z_q \succeq 0 & \text{for } q \in Q \\ & Z_p \succeq 0 & \end{array} \tag{5.7}$$

Any solution to the program D_d can be extended to a solution to the program D'_d with the same optimal value by taking Z_p to be the zero matrix. On the other hand, any solution $(z, \{Z_q\}_{q\in Q}, Z_p)$ to the program D'_d gives rise to a solution $(\tilde{z}, \{\tilde{Z}_q\}_{q\in Q})$ to the program D_d with the same optimal value by setting $\tilde{z} := z$ and $\tilde{Z}_q := Z_q + c_q Z_p$ for each $q \in P$, and $\tilde{Z}_q := Z_q$ for each $q \in Q \setminus P$. This follows from the fact that $A_{p,d_p,\alpha} = \sum_{q\in P} c_q A_{q,d_q,\alpha}$. □

5.3 SOS proofs as semidefinite sets

Fix a set of polynomials Q and a further polynomial $p = \sum_\alpha a_\alpha x^\alpha$ such that $\deg(p) \le 2d$. Our goal now is to describe degree-$2d$ SOS proofs of the polynomial inequality $p \ge 0$ from Q as points in a semidefinite set $K_d(Q, p)$ that we are about to define. Recall that a degree-$2d$ SOS proof of $p \ge 0$ from Q is an indexed set of polynomials $\{s_q : q \in \bar{Q}\}$ that satisfy an identity $\sum_{q\in\bar{Q}} q s_q = p$ where $\bar{Q} = Q \cup B_n$ and for every $q \in \bar{Q}$, the polynomial s_q is a sum of squares of polynomials and has degree at most $2d_q$. A polynomial s of degree at most $2t$ is a sum of squares if and only if there exists a positive semidefinite matrix Z indexed by monomials of degree at most t such that $s = \langle M_t, Z\rangle$. Therefore, there exists a degree-$2d$ SOS proof of the polynomial inequality $p \ge 0$ from Q if and only if, for every $q \in \bar{Q}$, there exists a positive semidefinite matrix Z_q indexed by monomials of degree at most d_q such that

$$\sum_{q\in\bar{Q}} q\langle M_{d_q}, Z_q\rangle = \sum_\alpha a_\alpha x^\alpha. \tag{5.8}$$

Let us have a closer look at the expression $\sum_{q\in\bar{Q}} q\langle M_{d_q}, Z_q\rangle$ on the left-hand side of the above identity. It can be rewritten in terms of the matrices introduced at the beginning of Subsection 5.2:

$$\begin{aligned} \sum_{q\in\bar{Q}} q\langle M_{d_q}, Z_q\rangle = \sum_{q\in\bar{Q}} \langle M_{q,d_q}, Z_q\rangle = \sum_{q\in\bar{Q}} \langle \sum_\alpha x^\alpha A_{q,d_q,\alpha}, Z_q\rangle = \\ = \sum_\alpha x^\alpha \sum_{q\in\bar{Q}} \langle A_{q,d_q,\alpha}, Z_q\rangle. \end{aligned} \tag{5.9}$$

Hence, there exists a degree-$2d$ SOS proof of $p \ge 0$ from Q if, and only if, there exists a set of positive semidefinite matrices $\{Z_q : q \in \bar{Q}\}$ such that for every $q \in \bar{Q}$ the matrix Z_q is indexed

by monomials of degree at most d_q and for all $|\alpha| \leq 2d$ it holds $\sum_{q\in\bar{Q}} \langle A_{q,d_q,\alpha}, Z_q\rangle = a_\alpha$, which, in turn, can be expressed as non-emptiness of the semidefinite set $K_d(Q,p) \subseteq \mathbb{R}^{I_d}$ given by:

$$\sum_{q\in\bar{Q}} \langle A_{q,d_q,\alpha}, Z_q\rangle = a_\alpha \quad \text{for } |\alpha| \leq 2d \quad \text{and} \quad X \succeq 0, \tag{5.10}$$

where $J_d = \{(q, x^\alpha) : q \in \bar{Q}, |\alpha| \leq d_q\}$ is a set of indices, X is a $J_d \times J_d$ symmetric matrix of formal variables, $I_d = \{\{(q, x^\alpha), (q', x^{\alpha'})\} : (q, x^\alpha), (q', x^{\alpha'}) \in J_d\}$ is a set of variable indices, and for every $q \in \bar{Q}$, the matrix Z_q is the principal submatrix of X corresponding to the rows and columns indexed by $\{(q, x^\alpha) : |\alpha| \leq d_q\}$.

Indeed, from every feasible point $X \in K_d(Q,p)$ we get a set of positive semidefinite matrices $\{Z_q : q \in \bar{Q}\}$ satisfying the identity (5.8) by setting Z_q be the principal submatrix of X corresponding to the rows and columns indexed by $\{(q, x^\alpha) : |\alpha| \leq d_q\}$. On the other hand, any set of positive semidefinite matrices $\{Z_q : q \in \bar{Q}\}$ satisfying the identity (5.8) can be extended to a point in $K_d(Q,p)$ by setting all remaining variables to 0.

The representation of the semidefinite set $K_d(Q,p)$ can be easily obtained from the representation of the set of polynomials Q and the polynomial p by means of FPC-interpretations:

Fact 5.4. *For every fixed positive integer d, there is an* FPC*-interpretation that takes a set of polynomials Q and a polynomial p as input and outputs a representation of the semidefinite set $K_d(Q,p)$. Moreover, there exists a constant c, independent of d, such that the formulas in the* FPC *interpretation have at most cd variables.*

Therefore, as a consequence of Theorem 4.10 we obtain Corollary 5.1.

Proof of Corollary 5.1. Let us fix a positive integer d and let Φ be the FPC-interpretation from Fact 5.4. We compose Φ with the $\mathrm{C}^\omega_{\infty\omega}$-sentence from Theorem 4.10 that decides the exact feasibility of semidefinite sets. The resulting sentence ψ decides the existence of an SOS proof of degree $2d$. It is a sentence of $\mathrm{C}^k_{\infty\omega}$, where $k = cd$, for an integer c that is independent of d. A $\mathrm{C}^\omega_{\infty\omega}$-sentence deciding the existence of an SOS refutation of degree $2d$ is obtained analogously by starting with an FPC-interpretation which takes as input a set of polynomials Q and outputs the semidefinite set $K_d(Q,-1)$. □

5.4 SOS refutations

We will now relate the existence of SOS refutations of a set of polynomials Q to the primal and dual problems in the Lasserre hierarchy for the polynomial optimization problem $\mathrm{POP}(0;\bar{Q})$. Then we will introduce the concept of ϵ-approximate SOS refutation and use the primal-dual correspondence to show that, for small enough $\epsilon > 0$, the existence of SOS refutations is equivalent to the existence of ϵ-approximate ones. It will follow from this that the problem of deciding the existence of SOS refutations of a fixed degree reduces, by means of FPC-interpretations, to the weak feasibility problem for semidefinite sets.

For any set of polynomials Q, the polynomial optimization problem $\mathrm{POP}(0;\bar{Q})$, characterizing the existence of 0/1-solutions to the system of polynomial inequalities $\{q \geq 0 : q \in Q\}$,

will be denoted by SOL(Q):

$$(\mathrm{SOL}(Q)) \quad : \quad \inf{}_x \ 0 \ \text{ s.t. } \ q(x) \geq 0 \ \text{ for } q \in \bar{Q}. \tag{5.11}$$

Indeed, the optimization problem SOL(Q) is feasible if and only if the system of polynomial inequalities $\{q \geq 0 : q \in Q\}$ has a 0/1-solution if and only if the optimal value of SOL(Q) is 0. Otherwise, the optimal value of SOL(Q) is $+\infty$. Although we care only about the feasibility of SOL(Q), we define it as an optimization problem, since we want to analyze its Lasserre SDP relaxations.

For a positive integer d, by $(P_d(Q), D_d(Q))$ we denote the level-d Lasserre SDP relaxation of the polynomial optimization problem SOL(Q), i.e., $P_d(Q)$ is the semidefinite program:

$$\begin{array}{ll} \inf{}_y & 0 \\ \text{s.t.} & y_\emptyset = 1 \\ & M_{q,d_q}(y) \succeq 0 \quad \text{for } q \in \bar{Q} \end{array} \tag{5.12}$$

and $D_d(Q)$ is the semidefinite program:

$$\begin{array}{lll} \sup{}_{z,Z} & z & \\ \text{s.t.} & \sum_{q \in \bar{Q}} \langle A_{q,d_q,\emptyset}, Z_q \rangle = -z & \\ & \sum_{q \in \bar{Q}} \langle A_{q,d_q,\alpha}, Z_q \rangle = 0 & \text{for } 1 \leq |\alpha| \leq 2d \\ & Z_q \succeq 0 & \text{for } q \in \bar{Q} \end{array} \tag{5.13}$$

Observe that degree-$2d$ SOS refutations of Q correspond precisely to the feasible solutions to $D_d(Q)$ with value 1 (see (5.8) and (5.9)). The following lemma summarizes the relationship between degree-$2d$ SOS refutations of Q and solutions to the program $D_d(Q)$. The second equivalence follows from the fact that by multiplying a solution to $D_d(Q)$ with value v by any $c \geq 0$ we obtain another solution with value cv.

Lemma 5.5. *There exists an* SOS *refutation of Q of degree $2d$ if and only if $D_d(Q)$ has a solution with value 1 if and only if the optimal value of $D_d(Q)$ is $+\infty$.*

For a system of polynomials Q, a *pseudoexpectation for Q of degree $2d$* is a linear mapping F from the set of polynomials of degree at most $2d$ over the set of variables $x_1, \ldots, x_n$ to the reals such that $F(1) = 1$, and for every $q \in \bar{Q}$ and every sum of squares polynomial s of degree at most $2d_q$, we have $F(qs) \geq 0$.

A linear mapping from the set of polynomials of degree at most $2d$ to the reals is uniquely defined by its restriction to monomials. Therefore, there is a natural one-to-one correspondence between linear functions from the set of polynomials of degree at most $2d$ to the reals and assignments to the set of variables $\{y_\alpha : |\alpha| \leq 2d\}$ of the program $P_d(Q)$, given by $G(y_\alpha) = F(x^\alpha)$. We recall the known fact that an assignment G to the variables of $P_d(Q)$ is a feasible solution if and only if F is a pseudoexpectation of degree $2d$.

Lemma 5.6. *There exists a degree-$2d$ pseudoexpectation for Q if and only if the program $P_d(Q)$ is feasible.*

Proof. Let F be a linear function from the set of polynomials of degree at most $2d$ to the reals and let G be the corresponding assignment to the variables of $P_d(Q)$. The statement of the lemma follows by showing that for every $q \in \bar{Q}$, the matrix $M_{q,d_q}(G(y))$ is positive semidefinite if and only if for every sum of squares polynomial s of degree at most $2d_q$, we have $F(qs) \geq 0$.

Let us take some $q \in \bar{Q}$. Observe that for every matrix Z indexed by monomials of degree at most d_q, we have

$$\langle M_{q,d_q}(G(y)), Z\rangle = \langle F(M_{q,d_q}), Z\rangle) = F(\langle qM_{d_q}, Z\rangle) = F(q\langle M_{d_q}, Z\rangle). \tag{5.14}$$

The matrix $M_{q,d_q}(G(y))$ is positive semidefinite if and only if for every positive semidefinite matrix Z indexed by monomials of degree at most d_q, it holds that $\langle M_{q,d_q}(G(y)), Z\rangle = F(q\langle M_{d_q}, Z\rangle) \geq 0$ if and only if $F(qs) \geq 0$ for every sum of squares polynomial s of degree at most $2d_q$. The last equivalence follows from the fact that a polynomial s of degree at most $2t$ is a sum of squares if and only if there exists a positive semidefinite matrix Z indexed by monomials of degree at most t such that $s = \langle M_t, Z\rangle$. □

Note that by summing the inequalities $1 - x_1 \geq 0, \ldots, 1 - x_n \geq 0$, together with the inequalities $x_1 - x_1^2 \geq 0, \ldots, x_n - x_n^2 \geq 0$, we get the inequality $n - \sum_{i\in[n]} x^2 \geq 0$, which witnesses the fact that the problem SOL(Q) is encircled. By Lemma 5.3 and Theorem 5.2 it follows that for the problem SOL(Q) there is no duality gap between primal and dual SDP problems in the Lasserre hierarchy. In particular, the optimal value of $D_d(Q)$ is $+\infty$ if and only if $P_d(Q)$ is infeasible. Now, recall from Lemma 5.5 that the optimal value of $D_d(Q)$ is $+\infty$ if and only if there exists an SOS refutation of Q of degree $2d$, and from Lemma 5.6 that the program $P_d(Q)$ is infeasible if and only if there is no pseudoexpectation for Q of degree $2d$. Hence, we obtain the following:

Corollary 5.7. *There exists an* SOS *refutation of Q of degree $2d$ if and only if there is no pseudoexpectation for Q of degree $2d$.*

For any $\epsilon > 0$, an *ϵ-approximate* degree-$2d$ SOS refutation of a set of polynomials Q is an indexed set of polynomials $\{s_q : q \in \bar{Q}\}$ that satisfy an identity

$$\sum_{q\in\bar{Q}} qs_q = \sum_{\alpha} a_\alpha x^\alpha, \tag{5.15}$$

where for every $q \in \bar{Q}$, the polynomial s_q is a sum of squares, for each x^α of degree at least 1 we have $|a_\alpha| \leq \epsilon$, and $|1 + a_\emptyset| \leq \epsilon$. In the same way as the degree-$2d$ SOS refutations correspond to the points in the semidefinite set $K_d(Q, -1)$, the ϵ-approximate degree-$2d$ SOS refutations correspond to the points in the ϵ-relaxation of $K_d(Q, -1)$.

In what follows, suppose that Q has no degree-$2d$ SOS refutation. By Corollary 5.7, there exists a degree-$2d$ pseudoexpectation. This in turn, as we will show now, precludes even the existence of ϵ-approximate refutations, for small enough ϵ. The key is the following lemma, which says that in the presence of Boolean axioms the absolute values of a pseudoexpectation on the set of monomials are bounded by 1.

Lemma 5.8. *If F is a degree-$2d$ pseudoexpectation for Q, then $0 \leq F(m) \leq 1$ for every monomial m of degree at most d, and $-1 \leq F(m) \leq 1$ for every monomial m of degree at most $2d$.*

Proof. Consider a monomial m written as a product of powers of distinct variables. The *multilinearization* $\bar{m}$ of m is the monomial obtained from m by replacing each x^c with $c \geq 2$ that appears in this product by x. For instance, the multilinearization of x^2y^3z is the monomial xyz.

First we show that if m is a monomial of degree at most $2d$, then $F(\bar{m}) = F(m)$. We do this by showing that $F(x^2m) = F(xm)$ for every variable x and every monomial m of degree at most $2d-2$. Fix such a monomial m and let r and s be monomials of degree at most $d-1$ such that $m = rs$. Note that $m = p^2 - q^2$ where $p = (r+s)/2$ and $q = (r-s)/2$, and both p^2 and q^2 have degree at most $2d-2$. It holds that

$$\begin{aligned} F((x^2-x)m) &= F((x^2-x)(p^2-q^2)) \\ &= F((x^2-x)p^2) + F((x-x^2)q^2) \\ &\geq 0, \end{aligned} \tag{5.16}$$

$$\begin{aligned} F((x^2-x)m) &= F((x^2-x)(p^2-q^2)) \\ &= -F((x^2-x)q^2) - F((x-x^2)p^2) \\ &\leq 0, \end{aligned} \tag{5.17}$$

where the last inequalities in (5.16) and (5.17) follow from the fact that the polynomials x^2-x and $x-x^2$ are Boolean axioms so they belong to $\bar{Q}$ and $d_{x^2-x} = d_{x-x^2} = 2d-2$. Hence, by the definition of a pseudoexpectation all the values $F((x^2-x)p^2)$, $F((x-x^2)q^2)$, $F((x^2-x)q^2)$ and $F((x-x^2)p^2)$ are non-negative.

This shows that $F((x^2-x)m) = 0$ and hence $F(x^2m) = F(xm)$.

Now we show that $0 \leq F(m) \leq 1$ for every monomial m of degree at most d. By the previous paragraph we have $F(m) = F(m^2)$, and $F(m^2) \geq 0$ because m^2 is a square of degree at most $2d$. The other inequality will be shown by induction on the degree. For the empty monomial 1 we have $F(1) = 1$. Now let m be a monomial of degree at most $d-1$ such that $F(m) \leq 1$ and let x be a variable. It holds that $F(m) - F(xm) = F((1-x)m) = F((1-x)m^2) \geq 0$, and hence $F(xm) \leq F(m) \leq 1$.

Finally, let m be a monomial of degree at most $2d$ and let r and s be monomials of degree at most d such that $m = rs$. We have $F(r^2) + 2F(rs) + F(s^2) = F((r+s)^2) \geq 0$. Therefore, $2F(rs) \geq -F(r^2) - F(s^2) \geq -2$, so $F(m) \geq -1$. Similarly $F(r^2) - 2F(rs) + F(s^2) = F((r-s)^2) \geq 0$. Therefore, $2F(rs) \leq F(r^2) + F(s^2) \leq 2$, so $F(m) \leq 1$. □

Let

$$\epsilon_{n,d} = \frac{1}{3}\binom{n+2d}{2d}^{-1}. \tag{5.18}$$

Note that $1/(3\epsilon_{n,d})$ is the number of monomials of degree $2d$ over the set of n variables. We are now ready to show that the existence of a degree-$2d$ SOS refutation of a system of

polynomial inequalities with n variables is equivalent to the existence of an $\epsilon_{n,d}$-approximate such refutation.

Proposition 5.9. *There exists an* SOS *refutation of Q of degree $2d$ if and only if there exists an $\epsilon_{n,d}$-approximate* SOS *refutation of Q of degree $2d$, where n is the number of variables in Q.*

Proof. If Q has an SOS refutation of degree $2d$, then clearly it has an $\epsilon_{n,d}$-approximate refutation of degree $2d$.

Now assume that Q has no SOS refutation of degree $2d$. Therefore, by Corollary 5.7 there exists a pseudoexpectation of degree $2d$. Let us denote it by F. Suppose that Q has an $\epsilon_{n,d}$-approximate SOS refutation of degree $2d$, i.e., there exists a set of sum of squares polynomials $\{s_q : q \in \bar{Q}\}$ such that

$$\sum_{q\in\bar{Q}} qs_q = \sum_{\alpha} a_\alpha x^\alpha, \tag{5.19}$$

where for each x^α of degree at least 1, we have $|a_\alpha| \leq \epsilon_{n,d}$, and $|1 + a_\emptyset| \leq \epsilon_{n,d}$.

Now, observe that $F\big(\sum_{q\in\bar{Q}} qs_q\big) = \sum_{q\in\bar{Q}} F(qs_q) \geq 0$, while

$$F(\sum_{\alpha} a_\alpha x^\alpha) = a_\emptyset + \sum_{\alpha\neq\emptyset} a_\alpha F(x^\alpha) \leq -1 + \epsilon_{n,d} + \binom{n+2d}{2d}\epsilon_{n,d} \leq -\frac{1}{3}. \tag{5.20}$$

This contradiction finishes the proof. □

An ϵ-relaxation of a convex set K is either empty, which clearly implies the emptiness of the set K itself, or it has volume greater than δ (see Lemma 4.13), where δ can be easily computed by means of FPC-interpretations from the representation of K and ϵ. We therefore get the following:

Corollary 5.10. *For every positive integer d, there is an* FPC*-definable reduction from the problem of deciding the existence of* SOS *refutations of degree $2d$, to the weak feasibility problem for semidefinite sets.*

Proof. The reduction is an FPC-interpretation which takes a set of polynomials Q with n variables as input and outputs the $\epsilon_{n,d}$-relaxation of $K_d(Q, -1)$ and a rational $\delta > 0$, such that either the $\epsilon_{n,d}$-relaxation of $K_d(Q, -1)$ is empty, or it has volume greater than δ. □

6 Graph Isomorphism

We formulate the isomorphism problem for graphs G and H as a system $\mathrm{ISO}(G, H)$ of quadratic polynomial equations with 0/1-valued variables. Let U and V denote the sets of vertices of G and H, respectively, assumed to be disjoint. The atomic type of a tuple of points in a relational structure is the complete description of the equalities and the relations

that the points in the tuple satisfy. In the special case of graphs, these are the equalities and the edge and non-edge relationships between the vertices in the tuple. For $u_1, u_2 \in U$, we write $\mathrm{tp}_G(u_1, u_2)$ for the atomic type of (u_1, u_2) in G. Similarly, for $v_1, v_2 \in V$, we write $\mathrm{tp}_H(v_1, v_2)$ for the atomic type of (v_1, v_2) in H. The system of equations has one 0/1-valued variable $x_{u,v}$ for each pair of vertices $u \in U$ and $v \in V$; the intended meaning of $x_{u,v} = 1$ is that the vertex u is mapped to v by a fixed isomorphism. The set of equations of $\mathrm{ISO}(G, H)$ is the following:

$$\begin{array}{ll} \sum_{v \in V} x_{u,v} - 1 = 0 & \text{for } u \in U, \\ \sum_{u \in U} x_{u,v} - 1 = 0 & \text{for } v \in V, \\ x_{u_1,v_1} x_{u_2,v_2} = 0 & \text{for } u_1, u_2 \in U, v_1, v_2 \in V \text{ s.t. } \mathrm{tp}_G(u_1, u_2) \neq \mathrm{tp}_H(v_1, v_2). \end{array}$$

When necessary, we think of the equations $q = 0$ from $\mathrm{ISO}(G, H)$ as pairs of inequalities $q \geq 0$ and $-q \geq 0$. It is straightforward to check that the relational structure that represents $\mathrm{ISO}(G, H)$ can be produced from G and H by an FPC-interpretation. As a structure, the pair of graphs (G, H) is given by two sorts $\bar{U}$ and $\bar{V}$ for U and V, and two binary relations E and F of types $\bar{U} \times \bar{U}$ and $\bar{V} \times \bar{V}$ for the sets of edges of G and H, respectively. For sets of polynomial equations and inequalities we use the representation described in Section 5.

Fact 6.1. *There is an* FPC*-interpretation that takes a pair of graphs* (G, H) *as input and outputs the set of equations* $\mathrm{ISO}(G, H)$.

An SOS proof that G and H are not isomorphic is an SOS refutation of $\mathrm{ISO}(G, H)$. A Sherali-Adams (SA) proof that G and H are not isomorphic is an SA proof of the inequality $-1 \geq 0$ from $\mathrm{ISO}(G, H)$, where an SA proof is an identity of the type (5.2) in which the polynomials s_q are not sums-of-squares but sums of extended monomials, i.e., polynomials of the form $\sum_{i \in I} c_i \prod_{j \in J_i} x_j \prod_{k \in K_i} (1 - x_k)$ where each c_i is a positive real, and each J_i and K_i is a subset of indices of variables. A Polynomial Calculus (PC) proof that G and H are not isomorphic is a PC proof of the equation $-1 = 0$ from the system of polynomial equations $\mathrm{ISO}(G, H)$, where by PC we mean the (deductive) proof system for deriving polynomial equations over $\mathbb{R}[x_1, \ldots, x_n]$ by means of the following inference rules: from nothing derive the axiom polynomial equation $x^2 - x = 0$, from the equations $p = 0$ and $q = 0$ derive the equation $p + q = 0$, and from the equation $p = 0$ derive the equations $ap = 0$ and $xp = 0$, where p and q are polynomials, a is a real, and x is a variable. In monomial PC, as defined in [8], the polynomial p in the last rule is required to be either a monomial, or a product of a monomial with one of the polynomials from the set of hypotheses (in our case $\mathrm{ISO}(G, H)$), or a product of a monomial and an axiom polynomial $x^2 - x$.

We rely on the following facts from [3] and [8]:

Theorem 6.2. *Let* G *and* H *be graphs and let* k *be a positive integer. The following are equivalent:*

1. $G \equiv^k H$*, i.e.,* G *and* H *cannot be distinguished by* $\mathrm{C}^k_{\infty\omega}$*-sentences,*
2. *there is no degree-*k SA *proof that* G *and* H *are not isomorphic,*

3. there is no degree-k monomial PC *proof that G and H are not isomorphic.*

To be precise, the main result in [3] is stated for the formulation of the graph isomorphism problem as a system of *linear* equations with 0/1-valued variables. For that encoding, the correspondence between $\equiv^k$-equivalence and the non-existence of degree-k SA proofs is not exact but only a tight sandwich: if there is no degree-k SA proof that G and H are not isomorphic then $G \equiv^k H$, and if $G \equiv^k H$ then there is no degree-$(k-1)$ SA proof that G and H are not isomorphic. However, it follows from the methods in [3] and [8] that, for the quadratic encoding used here, Theorem 6.2 holds as stated. For the collapse result we are about to prove, we use Corollary 5.1 and the implication 2. implies 1. from Theorem 6.2.

Theorem 6.3. *There exists an integer c such that, for all pairs of graphs G and H and all positive integers d, if there is a degree-$2d$* SOS *proof that G and H are not isomorphic, then there is a degree-cd* SA *proof that G and H are not isomorphic.*

Proof. Fix a positive integer d. Let Φ be the FPC-interpretation from Fact 6.1 and compose it with the $\mathrm{C}^{\omega}_{\infty\omega}$-sentence from Corollary 5.1 that decides the existence of SOS refutations of degree $2d$. The resulting sentence ϕ is a sentence of $\mathrm{C}^{k}_{\infty\omega}$, where $k = cd$ for an integer c that is independent of d. The sentence ϕ was designed in such a way that for every pair of graphs G and H it holds that $(G, H) \models \phi$ if and only if there is a degree-$2d$ SOS proof that G and H are not isomorphic. In particular, since there certainly is no degree-$2d$ SOS proof that G is not isomorphic to an isomorphic copy of itself, we have $(G', G) \models \neg\phi$, where G' is an isomorphic copy of G on a disjoint set of vertices. Now assume that there is no degree-k SA proof that G and H are not isomorphic. We get $G \equiv^k H$ by Theorem 6.2, from which it follows that $(G', G) \equiv^k (G, H)$ because $G' \cong G$ and hence $G' \equiv^k G$, and $G \equiv^k H$. Since ϕ is a $\mathrm{C}^{k}_{\infty\omega}$-sentence and $(G', G) \models \neg\phi$ we get $(G, H) \models \neg\phi$. Therefore, by design of ϕ, there is no degree-$2d$ SOS proof that G and H are not isomorphic. □

Next we use the result of Berkholz [7] showing that, for systems of polynomial-equations over 0/1-valued variables, SOS simulates PC.

Theorem 6.4 ([7])**.** *Let Q be a system of polynomial equations with real coefficients over* 0/1*-valued variables and let d be a positive integer. If Q has a* PC *refutation of degree d, then Q has an* SOS *refutation of degree $2d + 1$.*

The discrepancy between the $2d + 1$ in the conclusion of Theorem 6.4 and the $2d$ in the conclusion of Theorem 1.1 from [7] is due to a small difference between our definition of SOS and the variant of SOS used in [7]. We discuss this next. We focus on the case of polynomial equations, which is the subject of the above theorem. We denote the variant by SOS′.

Given a system of polynomial equations $Q = \{q_i = 0 : i \in [k]\}$ over Boolean variables and a polynomial q, an SOS′ proof of $q \geq 0$ from Q is a sequence of polynomials $(g_1, \ldots, g_k, h_1, \ldots, h_n, s_0)$ that satisfy an identity

$$\sum_{i\in[k]} q_i g_i + \sum_{j\in[n]} (x_j^2 - x_j) h_j + s_0 = q, \tag{6.1}$$

where the polynomial s_0 is a sum of squares of polynomials. To be able to compare SOS with SOS$'$ we view each equation $q_i = 0$ in Q as two inequalities $q_i \geq 0$ and $-q_i \geq 0$. Hence, an SOS proof of $q \geq 0$ from Q is a set $\{s_1, \ldots, s_m\}$ of sum of squares polynomials that satisfy the identity $\sum_{j\in[m]} p_j s_j = q$, where, for every $j \in [m]$, the polynomial p_j either is in the set $\{q_1, -q_1, \ldots, q_k, -q_k\}$ or is one of the Boolean axioms listed in (5.1).

Lemma 6.5. *Let Q be a system of polynomial equations over 0/1-valued variables. If $q \geq 0$ has an* SOS$'$ *proof from Q of degree $2d$, then it has an* SOS *proof from Q of degree at most $2d + 1$.*

Proof. For any polynomial p and any monomial m, such that $\deg(pm) \leq 2d$, we will show that the product pm can be written as $pm = ps + (-p)s'$, where s and s' are sums of squares of polynomials and $\deg(ps) = \deg(-ps') \leq 2d + 1$. This last fact implies that the left-hand side of any degree-$2d$ SOS$'$ proof as in (6.1) can be rewritten as follows

$$\sum_{i\in[k]} q_i s_i + \sum_{i\in[k]} (-q_i)s'_i + \sum_{j\in[n]} (x_j^2 - x_j)z_j + \sum_{j\in[n]} (x_j - x_j^2)z'_j + s_0, \tag{6.2}$$

where, in the sequence $(s_1, \ldots, s_k, s'_1, \ldots, s'_k, z_1, \ldots, z_n, z'_1, \ldots, z'_n, s_0)$, all the polynomials are sums of squares. As a result of this rewriting we obtain an SOS proof of $q \geq 0$ from $\{q_1, -q_1, \ldots, q_k, -q_k\}$, and hence an SOS proof of $q \geq 0$ from Q by the convention that we introduced to be able to compare SOS with SOS$'$. Note that the degree of the proof increases by at most 1.

Take any polynomial p and any monomial m, such that $\deg(pm) \leq 2d$. Let r and t be monomials such that $m = rt$ and $|\deg(r) - \deg(t)| \leq 1$. Then we have $m = s - s'$, where $s = ((r + t)/2)^2$ and $s' = ((r - t)/2)^2$. Moreover, $\deg(s) = \deg(s') \leq \deg(m) + 1$. We obtain, $pm = ps + (-p)s'$, where s and s' are sums of squares of polynomials and $\deg(ps) = \deg(-ps') \leq 2d + 1$, which finishes the proof. □

For graphs G and H, let $\mathrm{sos}(G, H)$, $\mathrm{sa}(G, H)$, $\mathrm{monpc}(G, H)$ and $\mathrm{pc}(G, H)$ denote the smallest degrees for which SOS, SA, monomial PC and PC prove that G and H are not isomorphic, respectively, taken as ∞ if the graphs are isomorphic. Combining Theorems 6.4, 6.2, 6.3, we get a full cycle of implications.

Corollary 6.6. *There exists an integer constant c such that, for all pairs of graphs G and H, the following inequalities hold:*

$$\tfrac{1}{2} \cdot (\mathrm{sos}(G, H) - 1) \leq \mathrm{pc}(G, H) \leq \mathrm{monpc}(G, H) \leq \mathrm{sa}(G, H) \leq \tfrac{c}{2} \cdot \mathrm{sos}(G, H). \tag{6.3}$$

Let us now state the collapse for the *primals*. Recall from Subsection 5.2 that the Lasserre SDP relaxation of a polynomial optimization problem is defined to be a primal-dual pair of semidefinite programs. However, it is the primal that is most often referred to as the Lasserre relaxation. This is the terminology we will use now. For a positive integer k, let $\mathrm{LA}_k(G, H)$ denote the level-k Lasserre relaxation of $\mathrm{ISO}(G, H)$, i.e., the primal in the primal-dual SDP-pair $(P_k(\mathrm{ISO}(G, H)), D_k(\mathrm{ISO}(G, H)))$ as defined in Subsection 5.4. By Lemma 5.5 and the

strong duality implied by Lemma 5.3 and Theorem 5.2, for every positive integer d it holds that $\mathrm{LA}_{2d}(G, H)$ is feasible if and only if there is no degree-$2d$ SOS proof that G and H are not isomorphic. Similarly, we write $\mathrm{SA}_k(G, H)$ to denote the primal in the primal-dual LP-pair corresponding to the level-k Sherali-Adams relaxation of $\mathrm{ISO}(G, H)$ as defined in [29, Section 4] for generic systems of polynomial constraints over 0/1-valued variables. We refer to it as the level-k Sherali-Adams relaxation of $\mathrm{ISO}(G, H)$. In this case, strong duality holds by the duality theorem for linear programming, and the dual solutions are degree-k SA refutations of $\mathrm{ISO}(G, H)$. It follows that, for every positive d, the linear program $\mathrm{SA}_d(G, H)$ is feasible if and only if there is no degree-d SA proof that G and H are not isomorphic. Theorem 6.3 gives then the following.

Corollary 6.7. *There exists an integer c such that, for all pairs of graphs G and H and all positive integers d, if the level-$2d$ Lasserre relaxation of $\mathrm{ISO}(G, H)$ is infeasible, then the level-cd Sherali-Adams relaxation of $\mathrm{ISO}(G, H)$ is infeasible.*

As mentioned in the introduction, our proof of Corollary 6.7 is very indirect as it goes through many black boxes. It would be very instructive to find a concrete and direct way of lifting feasible LP-solutions of $\mathrm{SA}_{cd}(G, H)$ to feasible SDP-solutions of $\mathrm{LA}_{2d}(G, H)$. Corollary 6.7 and the fact that its indirect proof is nonetheless constructive imply that such a direct way of lifting does, in principle, exist.

Acknowledgments. We are grateful to Christoph Berkholz, Anuj Dawar, and Wied Pakusa, for useful discussions at an early stage of this work. We are also grateful to Aaron Potechin for pointing out that the ability of the Lasserre hierarchy to capture spectral arguments was relevant for our result. Special thanks go to Moritz Müller for carefully reading and commenting on a preliminary version of this paper. First author partially funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement ERC-2014-CoG 648276 (AUTAR) and MICCIN grant TIN2016-76573-C2-1P (TASSAT3) and AEI grant PID2019-109137GB-C22 (PROOFS). The work of second author on this manuscript is a part of the project BOBR that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 948057). Second author supported also by the French Agence Nationale de la Recherche, QUID project reference ANR-18-CE40-0031. Part of this work was done while the second author was visiting UPC funded by AUTAR.

References

[1] Eric Allender, Peter Bürgisser, Johan Kjeldgaard-Pedersen, and Peter Bro Miltersen. On the complexity of numerical analysis. *SIAM J. Comput.*, 38(5):1987–2006, 2009.

[2] Matthew Anderson, Anuj Dawar, and Bjarki Holm. Solving linear programs without breaking abstractions. *J. ACM*, 62(6):48:1–48:26, 2015.

[3] Albert Atserias and Elitza Maneva. Sherali–Adams relaxations and indistinguishability in counting logics. *SIAM J. Comput.*, 42(1):112–137, 2013.

[4] Albert Atserias and Joanna Ochremiak. Definable ellipsoid method, sums-of-squares proofs, and the isomorphism problem. *CoRR*, abs/1802.02388, 2018.

[5] Albert Atserias and Joanna Ochremiak. Definable ellipsoid method, sums-of-squares proofs, and the isomorphism problem. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, page 66–75, New York, NY, USA, 2018. Association for Computing Machinery.

[6] Boaz Barak, Fernando G. S. L. Brandão, Aram W. Harrow, Jonathan A. Kelner, David Steurer, and Yuan Zhou. Hypercontractivity, sum-of-squares proofs, and their applications. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 307–326. ACM, 2012.

[7] Christoph Berkholz. The Relation between Polynomial Calculus, Sherali-Adams, and Sum-of-Squares Proofs. In *Proceedings of the 35th Symposium on Theoretical Aspects of Computer Science*, pages 11:1–11:14, 2018.

[8] Christoph Berkholz and Martin Grohe. Limitations of algebraic approaches to graph isomorphism testing. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming, Part I*, pages 155–166, 2015.

[9] Andreas Blass, Yuri Gurevich, and Saharon Shelah. On polynomial time computation over unordered structures. *J. Symb. Log.*, 67(3):1093–1125, 2002.

[10] Anuj Dawar, Simone Severini, and Octavio Zapata. Descriptive complexity of graph spectra. *Ann. Pure Appl. Log.*, 170(9):993–1007, 2019.

[11] Anuj Dawar and Pengming Wang. Definability of semidefinite programming and Lasserre lower bounds for CSPs. In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 1–12, 2017.

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

[13] Heinz-Dieter Ebbinghaus, Jörg Flum, and Wolfgang Thomas. *Mathematical logic (2. ed.)*. Undergraduate texts in mathematics. Springer, 1994.

[14] Erich Grädel, Martin Grohe, Benedikt Pago, and Wied Pakusa. A finite-model-theoretic view on propositional proof complexity. *Log. Methods Comput. Sci.*, 15(1), 2019.

[15] Dima Grigoriev and Nicolai Vorobjov. Complexity of null- and positivstellensatz proofs. *Annals of Pure and Applied Logic*, 113(1):153–160, 2001. First St. Petersburg Conference on Days of Logic and Computability.

[16] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1993.

[17] Tuomas Hakoniemi. Monomial size vs. bit-complexity in sums-of-squares and polynomial calculus. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–7. IEEE, 2021.

[18] Neil Immerman. Relational queries computable in polynomial time (extended abstract). In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, page 147–152, New York, NY, USA, 1982. Association for Computing Machinery.

[19] Cédric Josz and Didier Henrion. Strong duality in Lasserre's hierarchy for polynomial optimization. *Optim. Lett.*, 10(1):3–10, 2016.

[20] Jean-Bernard Lasserre. Global optimization with polynomials and the problems of moments. *SIAM J. Optim.*, 11(3):796–817, 2001.

[21] Peter N. Malkin. Sherali–Adams relaxations of graph isomorphism polytopes. *Discrete Optim.*, 12:73–97, 2014.

[22] Ryan O'Donnell. SOS is not obviously automatizable, even approximately. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPIcs*, pages 59:1–59:10. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[23] Ryan O'Donnell and Tselil Schramm. Sherali-adams strikes back. *Theory Comput.*, 17:1–30, 2021.

[24] Ryan O'Donnell, John Wright, Chenggang Wu, and Yuan Zhou. Hardness of robust graph isomorphism, Lasserre gaps, and asymmetry of random graphs. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1659–1677, 2014.

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

[26] Pablo A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, Massachussets Institute of Technology, 2000.

[27] Prasad Raghavendra and Benjamin Weitz. On the bit complexity of sum-of-squares proofs. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 80:1–80:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[28] Walter Rudin. *Real and Complex Analysis, 3rd Ed.* McGraw-Hill, Inc., 1987.

[29] Hanif D. Sherali and Warren P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.*, 3(3):411–430, 1990.

[30] Sergey P. Tarasov and Mikhail N. Vyalyi. Semidefinite programming and arithmetic circuit evaluation. *Discrete Appl. Math.*, 156(11):2070–2078, 2008.

[31] Gottfried Tinhofer. Graph isomorphism and theorems of birkhoff type. *Computing*, 36(4):285–300, 1986.

[32] Moshe Y. Vardi. The complexity of relational query languages (extended abstract). In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, page 137–146, New York, NY, USA, 1982. Association for Computing Machinery.

[33] Pengming Wang. Descriptive complexity of constraint problems. *Doctoral thesis*, 2018.