On the Complexity of Counting the Hilbert Basis of a Linear Diophantine System

Miki Hermann[1], Laurent Juban[1], and Phokion G. Kolaitis[2]

[1] LORIA (CNRS and Université Henri Poincaré Nancy 1), BP 239, 54506 Vandœuvre-lès-Nancy, France. {hermann,juban}@loria.fr
[2] Computer Science Department, University of California, Santa Cruz, CA 95064, U.S.A., kolaitis@cse.ucsc.edu

Abstract. We investigate the computational complexity of counting the Hilbert basis of a homogeneous system of linear Diophantine equations. We establish lower and upper bounds on the complexity of this problem by showing that counting the Hilbert basis is #P-hard and belongs to the class #NP. Moreover, we investigate the complexity of variants obtained by restricting the number of occurrences of the variables in the system.

1 Introduction and Summary of Results

The Hilbert basis of a homogeneous system of linear Diophantine equations over the non-negative integers is the set of all non-zero vectors that are *minimal* solutions with respect to the pointwise order. This set forms indeed a basis of the space of solutions of the system, that is, every solution can be written as a positive linear combination of vectors from the Hilbert basis, and no member of the Hilbert basis can be expressed as a positive linear combination of other members. Moreover, this basis is essentially unique. The concept of a Hilbert basis was studied as early as the second half of the 19th century by Gordan [Gor73] and Hilbert [Hil90]. Since that time, it has received considerable attention in linear algebra and integer programming (see Schrijver [Sch86]).

Computing the Hilbert basis of a homogeneous system of linear Diophantine equations over non-negative integers has turned out to be one of the key problems in automated deduction. Its importance in this area emerged through the work of Stickel [Sti75,Sti81], who designed the first algorithm for unification in the presence of associative-commutative (AC) function symbols. Stickel showed that the minimal complete set of unifiers of a simultaneous elementary AC-unification problem can be obtained from the Hilbert basis of an associated homogeneous system of linear Diophantine equations over non-negative integers. Indeed, the minimal complete set of AC-unifiers is the set of all *compatible* subsets of the Hilbert basis of that system, where compatible in this context means that every variable can be instantiated by a non-zero linear combination of the members of the compatible subset. Other AC-unification algorithms were presented afterwards, including algorithms by Fages [Fag87], Herold and Siekmann [HS87], Boudet [Bou93], and Boudet, Contejean, and Devie [BCD90]. Although these

AC-unification algorithms differ from each other in several aspects, they all rely on computing the Hilbert basis of the associated homogeneous system.

Following the publication of Stickel's algorithm [Sti75], researchers became interested in algorithms for computing the Hilbert basis. Huet [Hue78] described a conceptually simple algorithm for a single equation. Clausen and Fortenbacher presented in [CF89] a more sophisticated algorithm for a single equation based on automata theory. These papers assume that the computation of the Hilbert basis of a system can be reduced to successive computations of the Hilbert basis of single equations, interlaced with substitutions of the result into the rest of the system. Aggregation is a different method for transforming a system of equations into a single equation with the same space of solutions [EE85]. Both methods entail an exponential blow-up during the transformation. Several researchers, including Contejean and Devie [CD94], Lankford [Lan89], Domenjoud [Dom91a,Dom91b], have also developed direct algorithms for computing the Hilbert basis of systems with an arbitrary number of equations.

Every algorithm for computing the Hilbert basis of a system can also be used to count at the same time the number of elements of the Hilbert basis. Lankford [Lan89] derived an exponential lower bound on the cardinality of the Hilbert basis of a particularly challenging homogeneous Diophantine equation. In the present paper, we investigate the problem of counting the Hilbert basis of a system from a complexity-theoretic perspective. Our results imply that this counting problem is highly intractable and, thus, they shed light on the inherent complexity of algorithms for computing the Hilbert basis. According to Lankford [Lan89], "a complete description of the complexity of non-negative basis algorithms seems to be a difficult open problem".

The computational complexity theory of counting problems was developed by Valiant [Val79a,Val79b], who introduced and studied the class #P of functions that count the number of accepting paths of nondeterministic polynomial-time Turing machines, as well as the larger class #NP of functions that count the number of accepting paths of nondeterministic polynomial-time Turing machines with access to NP oracles. Valiant [Val79a,Val79b] demonstrated that these classes possess natural complete problems under suitable polynomial-time reductions. In particular, there is a large variety of #P-complete problems that arise in graph theory, logic, algebra, and combinatorics. Furthermore, #P-complete problems are encountered in the context of counting the number of minimal complete matchers modulo an equational theory, as shown in [HK95a,HK95b]. It should be noted that #P-complete and #P-hard problems are considered to be highly intractable, since, as established by Toda [Tod89], they dominate in an exact technical sense all problems in the polynomial hierarchy PH (the bottom level of which is NP).

We show that the problem of counting the Hilbert basis is #P-hard and is a member of the class #NP. These two results provide reasonably tight lower and upper bounds on the complexity of counting the Hilbert basis, even though they do not decisively pin down its exact complexity. We also analyze variants of this counting problem for restricted systems of homogeneous linear Diophantine equations; to this effect, we show that if every variable has at most two occurrences in the system, then the problem of counting the Hilbert basis is a member of the class #P. Finally, we establish that it is a #P-complete problem to count the number of compatible subsets of a given set of linearly independent and pairwise incomparable vectors. This result quantifies in a precise manner the inherent complexity of an important step in most AC-unification algorithms.

2 Counting Problems and Computational Complexity

This section contains the definitions of basic concepts, as well as some background material on counting problems and computational complexity. We assume some familiarity with the fundamentals of associative-commutative unification and computational complexity. Additional material on these topics can be found in the survey article [JK91] and in the monographs [BN98,Pap94,Sch86].

A homogeneous linear Diophantine system over non-negative integers is a system of equations $S\colon Ax = 0$, where $A = (a_i^j)_k^n$ is a $k \times n$ integer matrix and $x = (x_1, \ldots, x_n)$ is a vector of variables ranging over non-negative integers. We say that a solution s of S is *nontrivial* if it is different from the all-zero solution $(0, \ldots, 0)$. We say that a solution $s = (s_1, \ldots, s_n)$ of S is *smaller* than a solution $s' = (s'_1, \ldots, s'_n)$, and write $s < s'$, if $s \neq s'$ and, for all $i = 1, \ldots, n$, it is the case that $s_i \leq s'_i$. The relation $<$ is called the *pointwise order* on solutions. A solution s is *minimal* if it is nontrivial and there is no smaller nontrivial solution s'', that is, $s'' < s$ is false for every nontrivial solution s'' of S. The i-th coordinate s_i of a solution s, corresponding to the variable x_i, is alternatively denoted by $s(x_i)$.

The *Hilbert basis* $H(S)$ of the system S is the set of all minimal solutions of S. This set is indeed a *basis* for the space of nontrivial solutions of S, which means that no minimal solution can be expressed as a positive linear combination of the other minimal solutions, whereas every nontrivial solution can be expressed as a positive linear combination of minimal solutions. The Hilbert basis $H(S)$ is finite and it is the unique basis of the space of nontrivial solutions of S.

It is well known that Hilbert bases can be used to compute minimal complete sets of AC-unifiers. Indeed, let $AX \doteq_{\mathrm{AC}} A'X'$ be a simultaneous elementary AC-unification problem, where A and A' are matrices over non-negative integers, $X = (X_1, \ldots, X_j)$ and $X' = (X_{j+1}, \ldots, X_n)$ are not necessarily disjunctive vectors of formal variables, and $+$ is the unique AC-symbol. With this AC-unification problem, associate the homogeneous linear Diophantine system $S\colon (A - A')x = 0$, where the arithmetic variable x_i corresponds to the formal variable X_i for $i = 1, \ldots, n$. Consider the Hilbert basis $H(S)$ of the system S over the variables $x_1, \ldots, x_n$. Let $\{\alpha_1, \ldots, \alpha_m\}$ be a subset of $H(S)$ and $v = (v_1, \ldots, v_m)$ be a vector of new variables. For each $i = 1, \ldots, n$, assign the linear expression $\alpha_1^i v_1 + \cdots + \alpha_m^i v_m$ to the variable x_i, where α_j^i is the i-th coordinate of the vector α_j. We say that $\{\alpha_1, \ldots, \alpha_m\}$ is a *compatible subset* of $H(S)$ if, for each variable x_i, there exists a vector α_j such that $\alpha_j^i \neq 0$, that is, the variable x_i is not assigned the value 0. The minimal complete set of unifiers of the AC-unification problem $AX \doteq_{\mathrm{AC}} A'X'$ turns out to be the set of all compatible subsets of $H(S)$

of the system S above, where $x_i \mapsto \alpha_1^i v_1 + \cdots + \alpha_m^i v_m$ is the substitution of the variable x_i, when $\{\alpha_1, \ldots, \alpha_m\}$ is the chosen compatible subset.

In this paper, we are mainly concerned with the computational complexity of finding the cardinality of the Hilbert basis $H(S)$ of a given homogeneous linear Diophantine system $S\colon Ax = 0$, that is, counting the number of minimal solutions of S.

Valiant [Val79a,Val79b] was the first to investigate the computational complexity of counting problems. To this effect, he introduced the class #P of functions that count the number of accepting paths of nondeterministic polynomial-time Turing machines. The prototypical problem in this class is #SAT, which is the counting version of Boolean satisfiability.

#SAT
Input: Set V of Boolean variables and Boolean formula ϕ over V in conjunctive normal form.
Output: The number of truth assignments for the variables in V that satisfy ϕ.

In addition to initiating the study of #P, Valiant [Val79a,Val79b] developed a machine-based framework for introducing higher classes of counting problems. Specifically, for every complexity class $\mathcal{C}$ of decision problems, he defined $\#\mathcal{C}$ to be the union $\bigcup_{A \in \mathcal{C}} (\#\mathrm{P})^A$, where $(\#\mathrm{P})^A$ is the collection of all functions that count the accepting paths of nondeterministic polynomial-time Turing machines having A as their oracle. Thus, in this framework, #NP is the class of functions that count the number of accepting paths of $\mathrm{NP}^{\mathrm{NP}}$ machines, that is, nondeterministic polynomial-time Turing machines that have access to NP oracles. More recently, however, researchers have introduced complexity classes of counting problems using the framework of *witness functions* and *witness sets*. In this framework, a counting problem is viewed as a *witness function* w such that if x is an input, then $w(x)$ is a set of succinct certificates (witnesses) for x, and the goal is to compute the cardinality $|w(x)|$ of this *witness set* $w(x)$. Different classes of counting problems can then be obtained by considering the computational complexity of deciding membership in the witness set. Specifically, if C is a complexity class of decision problems, then Hemaspaandra and Vollmer [HV95] define $\# \cdot C$ to be the class of all witness functions w that satisfy the following conditions:

(1) There is a polynomial p such that for every x and every $y \in w(x)$, we have that $|y| \leq p(|x|)$, where $|x|$ is the size of x and $|y|$ is the size of y;

(2) The decision problem "given x and y, is $y \in w(x)$?" is in C.

It is easy to verify that $\#\mathrm{P} = \# \cdot \mathrm{P}$, that is, Valiant's class #P coincides with the class of witness functions for which membership in the witness set can be tested in polynomial time. For complexity classes C beyond P, however, the corresponding classes $\#C$ and $\# \cdot C$ may differ (see [HV95]); in general, finer-grained complexity classes can be obtained using the witness-based framework. For our purposes here, we are interested in a result from Toda's Ph.D. thesis [Tod91] that reveals a rather surprising connection between the two different approaches.

Theorem 1 (Toda). $\#\mathrm{NP} = \# \cdot \mathrm{coNP}$.

Thus, the above result asserts that Valiant's class #NP coincides with the class of witness functions for which membership in the witness set is in coNP. A proof of this theorem can also be found in [HV95].

It is clear that #P is contained in #NP. Moreover, #P contains the class FP of counting functions that can be computed in deterministic polynomial time. Therefore, we have the inclusions: FP $\subseteq$ #P $\subseteq$ #NP.

As a general principle, what makes a complexity class C interesting is the existence of natural *complete* problems for C, that is, members of C such that every problem in C can be reduced to them via a suitable reduction. As regards classes of counting problems, these reductions are polynomial-time reductions that allow us to efficiently compute the number of solutions of one problem from the number of solutions of another. Let v and w be two witness functions with domain Π^* and Σ^* respectively, where Π and Σ are finite alphabets. The counting problem v is *polynomial-time Turing reducible* to the counting problem w if there exists a polynomial-time deterministic Turing machine M that computes v by making calls to an oracle for w. Restricted notions of polynomial-time Turing reductions between counting problems have also been considered. In particular, a *polynomial-time 1-Turing reduction* is a polynomial-time Turing reduction in which the Turing machine M is allowed to make at most one call to the oracle for w [Val79a,TW92]. *Parsimonious reductions* constitute the most restricted notion of reducibility. These are the special case of polynomial-time 1-Turing reductions in which $v = w \circ g$, for some polynomial-time computable total function g. In other words, the oracle for w is queried once and no computation is performed after the answer of the oracle is received. Thus, parsimonious reductions preserve the number of solutions between counting problems.

Let $\#\mathcal{C}$ be a counting complexity class, such as #P or #NP. A counting problem w is *$\#\mathcal{C}$-hard* if, for each counting problem v in $\#\mathcal{C}$, there is a polynomial-time Turing reduction from v to w. If, in addition, w is a member of $\#\mathcal{C}$, then we say that w is *$\#\mathcal{C}$-complete*. If restricted notions of reductions are considered, then analogous concepts of hardness and completeness can be defined. For example, it is easy to show that #SAT is #P-complete under parsimonious reductions.

It should be pointed out that #P-hard problems are considered to be truly intractable. As a matter of fact, establishing #P-hardness implies in a precise technical sense that the problem at hand is substantially more intractable than an NP-complete problem (see [Joh90, page 109]). Valiant's [Val79a,Val79b] seminal discovery was that there are #P-complete counting problems such that, unlike SAT, their underlying decision problem is solvable in polynomial time. In fact, there are #P-complete problems having a trivial decision problem. The following two counting problems manifest these phenomena.

#PERFECT MATCHINGS [Val79b]

Input: Bipartite graph G with $2n$ nodes.

Output: The number of *perfect matchings* of G, i.e., sets of n edges such that no two edges share a common node.

#POSITIVE 2SAT [Val79b]

Input: Set V of Boolean variables and Boolean formula ϕ over V in conjunctive normal form, such that each clause of ϕ consists of exactly two positive literals.
Output: The number of truth assignments for the variables in V that satisfy ϕ.

Since the decision problem for positive 2CNF formulas is trivial, #POSITIVE 2SAT cannot be #P-complete under parsimonious reductions. Similarly, since there is a polynomial-time algorithm that tells whether a perfect matching exists, #PERFECT MATCHINGS cannot be #P-complete under parsimonious reductions, unless P = NP. It is known, however, that #PERFECT MATCHINGS is #P-complete under polynomial-time 1-Turing reductions (see [Pap94,Zan91]).

Both #PERFECT MATCHINGS and #POSITIVE 2SAT will be of use to us in establishing lower bounds for the complexity of the counting problems that we will study in the sequel. It is now time to formally introduce these problems.

#HILBERT
Input: A system of homogeneous linear Diophantine equations $S \colon Ax = 0$ over non-negative integers.
Output: The cardinality of the Hilbert basis $H(S)$ of S.

We mentioned earlier that Hilbert bases are used in computing minimal complete sets of unifiers of elementary AC-unification problems. As an intermediate step in this computation, one has to produce the set of all compatible subsets of a Hilbert basis $H(S)$ of a homogeneous linear Diophantine system $S \colon Ax = 0$ with n variables, where a set $\{\alpha_1, \ldots \alpha_m\}$ of n-dimensional vectors is *compatible* if for every $i \leq n$ there is a $j \leq m$ such that $\alpha_j^i \neq 0$. Note also that the members of a Hilbert basis are pairwise incomparable in the pointwise order and linearly independent with respect to linear combinations with nonnegative coefficients. This motivates the following counting problem.

#COMPATIBLE SUBSETS
Input: A set T of vectors of non-negative integers that are pairwise incomparable in the pointwise order and linearly independent with respect to linear combinations with nonnegative coefficients.
Output: The cardinality of the set of all compatible subsets of T.

3 The Complexity of Counting the Hilbert Basis

In this section, we obtain upper and lower bounds for the computational complexity of counting the Hilbert basis. Before stating and proving the main results, we address the issue of how the inputs are encoded.

If the input of a decision problem involves integers, then the complexity of that problem may depend on whether these integers are encoded in binary or in unary. For instance, KNAPSACK is NP-complete when the coefficients of the input are given in binary, but is solvable in polynomial time, when these coefficients are given in unary. In contrast, 3-PARTITION remains NP-complete, even if the coefficients of the input are given in unary (see [GJ79]). Note that typically the inputs of an AC-unification problem are given in unary, that is, each monomial ax is encoded by a bits (instead of $\log(a)$ bits), because ax represents a occurrences of the variable x and the inputs are terms over the alphabet of variables and function symbols. Since #HILBERT originates in elementary AC-unification, it would be natural to assume that the coefficients of linear Diophantine systems are written in unary. In linear algebra and integer programming, however, coefficients of linear systems are usually given in binary. At it turns out, however, the complexity of #HILBERT does not depend on whether the coefficients are encoded in binary or in unary, because there is a polynomial-time reduction that reduces #HILBERT in binary to #HILBERT in unary. Specifically, each system $S\colon Ax = 0$ can be transformed in polynomial time to a system $S'\colon A'x' = 0$ such that the coefficients of the matrix A' are 0, 1 or -1, there is a one-to-one mapping from the variables x to the variables x', and there is a one-to-one and onto mapping between the Hilbert bases of S and S'. For every monomial ax with $|a| > 2$ occurring in the system $S\colon Ax = 0$, let $p = \lceil \log_2 |a| \rceil$ and let $|a| = \sum_{i=0}^{p} a_i 2^i$ be the binary expansion of the absolute value of the coefficient a. Introduce p new variables $z_0, \ldots, z_{p-1}$, and add the equations $z_0 = x$ and $z_i = 2z_{i-1}$ for each $i = 1, \ldots, p-1$. Furthermore, introduce a new variable v and add the equation $v = a_0 z_0 + \cdots + a_p z_p$. If $a > 0$, then replace the monomial ax by the variable v; otherwise, replace ax by $-v$. By applying this transformation repeatedly, we obtain a system with coefficients from $\{-2, -1, 0, 1, 2\}$. Now, replace each monomial $2x$ (respectively, $-2x$) by the expression $v_1 + v_2$ (respectively, $-v_1 - v_2$), and add the equation $v_1 = v_2$ for the new variables v_1 and v_2. This completes the transformation of S to a system S' with the desired properties. Moreover, the entire transformation can be carried out in polynomial time, since for each monomial ax with $|a| > 2$ we add $\mathcal{O}(\log_2 |a|)$ new variables and equations, i.e., polynomially many in the size of the system S in binary.

Theorem 2. *The counting problem* #HILBERT *is in the class* #NP.

Proof. We first show that #HILBERT belongs to the class $\# \cdot$coNP, and then use Theorem 1 to conclude that it is in #NP.

Suppose we are given a homogeneous linear Diophantine system $S\colon Ax = 0$, where A is a $k \times n$ integer matrix. As the witness set for S, we take the set of all minimal solutions of S. We have to verify that the size of each minimal solution is polynomially bounded in the size of S, and that testing for minimality is in coNP. Papadimitriou [Pap81] proved that INTEGER PROGRAMMING is in NP by establishing that if a system of Diophantine equations has a nonnegative integer solution, then it has a "short" nonnegative integer solution. Several different researchers, including Domenjoud [Dom91a], Lambert [Lam87], and Pottier [Pot91] have shown that Papadimitriou's [Pap81] argument can be adapted to yield the stronger result that every minimal solution is "short" (see also [BN98]). Specifically, if $(s_1, \ldots, s_n)$ is an arbitrary minimal solution of $S\colon Ax = 0$, then $s_i \le n(ka)^{2k+1}$ holds for all $1 \le i \le n$, where a is the maximum of the coefficients of the matrix A. It follows that the size of every minimal solution is at most $n \log n + n(2k+1)(\log a + \log k)$, which implies that it is bounded by a polynomial in the size of the system S.

Finally, we have to show that the following decision problem is in coNP: given a system S as above and a vector s, is s a minimal solution of S? Since testing that s is a solution of S can be checked in polynomial time, we may as well assume that the given vector s is a solution of S. The complement of the aforementioned decision problem is in NP: to show that s is *not* a minimal solution, simply guess a nontrivial solution s' that is smaller than s in the pointwise order. $\square$

For the lower bound, we show that #HILBERT is #P-hard. For this, we produce a parsimonious reduction from #PERFECT MATCHINGS to #HILBERT.

Let $G = (U, V, E)$ be a bipartite graph with $2n$ nodes, where $|U| = |V| = n$. For every edge $e_i^j = (u_i, v_j) \in E$, introduce a variable x_i^j. In addition, introduce a variable y. For every node u_i in U, form the equation $x_i^{i_1} + \cdots + x_i^{i_l} = y$, where $e_i^{i_1}, \ldots, e_i^{i_l}$ is the list of all edges adjacent to u_i. Similarly, for each $v_j \in V$, form the equation $x_{j_1}^j + \cdots + x_{j_m}^j = y$, where $e_{j_1}^j, \ldots, e_{j_m}^j$ is the list of all edges adjacent to the node v_j. Let S be the homogeneous linear Diophantine system consisting of the above equations for all nodes $u_i \in U$ and $v_j \in V$. Thus, S consists of $2n$ equations; note that the left hand side of each equation contains occurrences of the variables x_i^j, whereas the right-hand of each equation is always y. Note also that each variable x_i^j occurs exactly twice in the system: once in the equation for node u_i and once in the equation for node v_j. We illustrate this construction with an example.

Example 3. Let $G = (\{u_1, u_2, u_3\}, \{v_1, v_2, v_3\}, E)$ be the bipartite graph having (u_1, v_1), (u_2, v_2), (u_2, v_3), (u_3, v_2), and (u_3, v_3) as edges. Then the system S consists of the following equations in the variables $x_1^1, x_2^2, x_2^3, x_3^2, x_3^3, y$:

$$\begin{array}{llll} u_1\colon & x_1^1 = y & v_1\colon & x_1^1 = y \\ u_2\colon & x_2^2 + x_2^3 = y & v_2\colon & x_2^2 + x_3^2 = y \\ u_3\colon & x_3^2 + x_3^3 = y & v_3\colon & x_2^3 + x_3^3 = y \end{array}$$

The Hilbert basis of the preceding system S is $H(S) = \{110011,\ 101101\}$. Note that the bipartite graph G has two perfect matchings and that the Hilbert basis $H(S)$ has two vectors. It turns out that this is not accidental. Indeed, in a sequence of three lemmas we establish that there exists a one-to-one correspondence between the perfect matchings of an arbitrary bipartite graph G and the minimal solutions of the associated system S.

The first lemma is proved using the well-known Hall's theorem.

Theorem 4 (Hall). *Let $G = (U, V, E)$ be a bipartite graph, where $|U| = |V|$. Then G has a perfect matching if and only if for every subset $A \subseteq U$, we have that $|A| \leq |R(A)|$, where $R(A) = \{v \in V \mid \exists u \in A, (u, v) \in E\}$.*

Lemma 5. *If a bipartite graph G has no perfect matching, then the only solution of S is the trivial all-zero solution $y = 0$, $x_i^j = 0$, where $1 \leq i, j \leq n$.*

Proof. If G has no perfect matching, then, by Hall's theorem, there exists a set of nodes $A \subseteq U$ such that $|A| > |R(A)|$ holds. Let E_A be the set of edges between the nodes of A and of $R(A)$ in the graph G. Let X_A be the variables corresponding to the edges E_A. By summing up the equations for the nodes

$u_i \in A$, we obtain the equation $\sum_{x_i^j \in X_A} x_i^j = |A|\, y$. Similarly, by summing up the equations for the nodes $v_j \in R(A)$, we obtain the relation $\sum_{x_i^j \in X_A} x_i^j \leq |R(A)|\, y$, since the edges E_A are a subset of the edges incident to $R(A)$. The left-hand sides of these two expressions are equal, therefore we can derive the relation $|A|\, y \leq |R(A)|\, y$. Since $|A| > |R(A)|$ holds, the only solution of the last inequation is $y = 0$. The right-hand side of each equation in S is equal to 0, therefore each variable x_i^j must also be 0. ☐

The proof of the next lemma is quite straightforward.

Lemma 6. *For every perfect matching M in G, there exists a solution of the system S such that $y = 1$, $x_i^j \in \{0, 1\}$, and $x_i^j = 1$ if and only if $e_i^j \in M$. Conversely, for every solution of S with $y = 1$ there exists a perfect matching in the bipartite graph G. Moreover, every solution of S with $y = 1$ is minimal.*

The last lemma implies that all minimal solutions of S must have $y = 1$.

Lemma 7. *If a solution of S is such that $y \geq 2$, then it is not minimal.*

Proof. Let $(s_1, \ldots, s_n, t)$ be a solution of $S(x_1, \ldots, x_n, y)$ such that $t \geq 2$. Transform the system S to a system S^* by eliminating all variables x_i for which $s_i = 0$. We obtain the system $S^*(x_1^*, \ldots, x_m^*, y)$ with the solution $(s_1^*, \ldots, s_m^*, t)$, where $s_i^* > 0$ for each $i = 1, \ldots, m$. Note that the system S^* corresponds to a bipartite graph $G^* = (U, V, E^*)$ that is a subgraph of G.

Since $t > 0$, Lemma 5 implies that there exists a perfect matching in G^*. From Lemma 6, it follows that there exists a solution $(s'_1, \ldots, s'_m, 1)$ of S^* such that $s'_i \in \{0, 1\}$. Clearly, $(s_1^*, \ldots, s_m^*, t) > (s'_1, \ldots, s'_m, 1)$ in the pointwise order. Therefore, also $(s_1, \ldots, s_n, t) > (s''_1, \ldots, s''_n, 1)$ holds, where $s''_i = s'_i$ if $s_i = s_i^*$ and $s''_i = 0$ if $s_i = 0$. Moreover, $(s''_1, \ldots, s''_n, 1)$ is a solution of S. Consequently, $(s_1, \ldots, s_n, t)$ is not a minimal solution of S. ☐

Lemmas 5,6, and 7 show that there is a reduction from #PERFECT MATCHINGS to #HILBERT. Thus, we have just completed the proof of the following theorem.

Theorem 8. *The counting problem #HILBERT is #P-hard.*

The preceding Theorems 2 and 8 yield upper and lower bounds for the complexity of counting the Hilbert basis. We now discuss some of the difficulties that arise if one attempts to narrow the gap between #P-hardness and membership in #NP. An inspection of the proof of Theorem 2 reveals that #HILBERT would be in #P, if testing a solution for minimality were solvable in polynomial time. Durand, Hermann, and Juban [DHJ99], however, have shown that it is a coNP-complete problem to tell whether a given solution of a homogeneous linear Diophantine system is minimal for a homogeneous linear Diophantine system. Thus, assuming P $\neq$ NP, to prove that #HILBERT is in #P would require one to come up with a very different set of witnesses for #HILBERT and show that membership in that witness set is in polynomial time. We believe that this is not possible and conjecture that #HILBERT is *not* in #P. As regards the lower bound, we showed that #PERFECT MATCHINGS has a parsimonious reduction to #HILBERT. Since, as mentioned earlier, #PERFECT MATCHINGS is #P-hard under polynomial-time 1-Turing reductions (see [Pap94,Zan91]), it follows that #HILBERT is also #P-hard under polynomial-time 1-Turing reductions. In a breakthrough paper [TW92], however, Toda and Watanabe proved that if a counting problem is #P-hard under polynomial-time 1-Turing reductions, then it is also $\#C$-hard under such reductions for every class C in the polynomial hierarchy PH. In particular, #HILBERT is #NP-hard under polynomial-time 1-Turing reductions and, hence, #NP-complete under such reductions. There is, however, something quite unsatisfactory with this conclusion, because Toda and Watanabe's result provides strong evidence that #P, #NP and other higher counting complexity classes are not closed under polynomial-time 1-Turing reductions. In turn, this raises the intriguing question whether more restricted reductions exist under which #NP is closed and HILBERT is #NP-hard. To appreciate this question, we note that it is not known whether restricted polynomial-time reductions exist under which #P is closed and #PERFECT MATCHINGS is #P-hard.

4 Restricted Systems of Diophantine Equations

To understand further the sources of intractability of #HILBERT, in this section we examine the complexity of counting the Hilbert basis for systems of homogeneous linear Diophantine equations that obey certain structural restrictions. As regards systems of equations, a natural restriction is to impose bounds on the number of occurrences of the variables in the system. In fact, this type of restriction has been studied in the context of elementary AC-matching and in many cases has helped in delineating the boundary between tractability and intractability [BKN87,HK95b]. In what follows, we investigate the impact of this structural restriction on the complexity of #HILBERT.

For every positive integer m, let #HILBERT(m) be the restriction of #HILBERT to systems of equations such that every variable has at most m distinct occurrences in the system, where a monomial ax accounts for a occurrences of the variable x. As seen earlier, each system $S\colon Ax = 0$ can be transformed to an essentially equivalent system $S'\colon A'x' = 0$, where the coefficients of the integer matrix A' are from $\{-1, 0, 1\}$. It is conceivable, however, that S' contains variables that have more than three occurrences. Let x be a such a variable with $m > 3$ occurrences. Introduce m new variables $u_1, \ldots, u_m$. Replace the i-th occurrence of the variable x by u_i and obtain a new system S'' by adding the equations $u_1 = u_2$, $u_2 = u_3$, $\ldots$, $u_m = u_1$. It is clear that each variable in S'' has at most three occurrences and that the coefficients of the system are from $\{-1, 0, 1\}$. It is also clear that there is a one-to-one and onto correspondence between the solutions of the systems S and S''. Furthermore, only a polynomial number of new equations is added. Thus, there exists a parsimonious reduction from the general problem #HILBERT to the restricted problem #HILBERT(3).

In view of the above, we focus our attention on the computational complexity of #HILBERT(1) and #HILBERT(2). Lincoln and Christian [LC89] pointed out that the Hilbert basis of a homogeneous linear Diophantine equation of the form $x_1 + \cdots + x_k = y_1 + \cdots + y_n$ consists of kn solutions. Specifically, it consists of all vectors satisfying the condition that there exist two indices i and j such that $x_i = 1, x_l = 0$ for all $l \neq i$, and $y_j = 1, y_m = 0$ for all $m \neq j$. Since there are k independent choices for i, and n independent choices for j, we have a total of kn minimal solutions. Suppose now we are given a system of homogeneous linear Diophantine equations in which each variable occurs exactly once. Thus, each variable occurs in exactly one equation and every equation of the system is of the form $x_1+\cdots+x_k = y_1+\cdots+y_n$. Consequently, the cardinality of the Hilbert basis of such a system can be computed in polynomial time by first finding the number of minimal solutions for each equation separately, and then add these numbers. This argument shows that the counting problem #HILBERT(1) is in the class FP of functions computable in deterministic polynomial time.

It remains to consider the computational complexity of #HILBERT(2), that is, counting the Hilbert basis of systems of homogeneous linear Diophantine equations in which each variable has at most two occurrences. Our main result in this section is that #HILBERT(2) is in the class #P and, thus, appears to have lower complexity than #HILBERT(3). The proof of this theorem is based on the following two lemmas.

Lemma 9. *Let S be a homogeneous linear Diophantine system over non-negative integers such that each variable has at most two occurrences. Then there exists a homogeneous linear Diophantine system S', called the reduced form of S, that is equivalent to S and has the following properties:*

- *Each variable has at most two occurrences;*
- *Each equation is of type $x_1 + x_2 = x_3$ (type A) or of type $x_1 = x_2$ (type B);*
- *Each variable having exactly two occurrences appears in two distinct equations, once in an equation of type A and once in an equation of type B.*

Proof. (*Hint*) Let $S\colon Ax = 0$ be a homogeneous linear Diophantine system with at most two occurrences of each variable. Eliminate from S all variables that can only take 0 as value, e.g., the equation $5x_1 + 3x_3 + x_5 = 0$ forces $x_1 = x_3 = x_5 = 0$. Eliminate all trivial equations $0 = 0$. Move the variables with negative coefficients to the right-hand side of the equations. Replace each monomial of the form $2x$ by x_1+x_2, where x_1 and x_2 are new variables, and add the equation $x_1 = x_2$. Now, every variable occurs with coefficient 1. After these transformations, each equation has the form $\alpha x = \alpha' x$, where α, α' are vectors over $\{0, 1\}$. Split each equation of the form $\alpha x = \alpha' x$ into $\alpha x = x'$, $\alpha' x = x''$, and $x' = x''$, where x', x'' are new variables. Split each equation of the form $x_1+x_2+\cdots+x_n = x_{n+1}$, where $n > 2$, into the three equations $x_1 + x'_1 = x_{n+1}$, $x_2 + \cdots x_n = x''_1$, and $x'_1 = x''_1$. Repeat this last step until there are no more equations with more than two variables on the left-hand sides. Finally, if there are two equations of the type $x_1 = x_2$ and $x_2 = x_3$, then contract these equations to a single equation $x_1 = x_3$ and delete the variable x_2. □

We saw earlier that the size of every minimal solution of an arbitrary homogeneous linear Diophantine equations is bounded by a polynomial in the size of the system. The second lemma shows that this bound can be dramatically improved, if every variable has at most two occurrences in the system.

Lemma 10. *Let S be a homogeneous linear Diophantine system over non-negative integers such that every variable has at most two occurrences. Then every minimal solution $(s_1, \ldots, s_n)$ of S has the property that the value of each coefficient s_i is at most 2.*

Proof. (*Hint*) The main idea of the proof is to transform the system S and a given solution s into a graph having the following property: each cycle in the graph represents a new solution s' such that s' is either the trivial all-zero solution or it is a solution that is pointwise smaller than or equal to the original solution s. After this, using properties of the graph, it can be shown that each variable with value 3 can be decremented by 1 or 2, and the result is still a solution of the original system. This is established via an analysis of 16 cases, the details of which can be found at `http://www.loria.fr/~juban/16cases.ps.gz`. It follows that in every minimal solution each variable must have value at most 2 [Jub98]. □

The previous two lemmas make it possible to obtain the following result, whose proof can be found in the Appendix.

Theorem 11. *The counting problem #HILBERT(2) is in the class #P.*

It is an interesting open problem to determine whether this upper bound is tight or it can be lowered even further. We conjecture that #HILBERT(2) is actually a #P-complete problem.

5 The Complexity of Counting the Compatible Subsets

Stickel's algorithm [Sti75] for simultaneous elementary AC-unification proceeds by first finding the Hilbert basis of the associated homogeneous linear Diophantine system, and then producing the set of all compatible subsets of that basis. To gain insight into the inherent complexity of this algorithm, we examine the computational complexity of counting the number of compatible subsets of a given set T of linearly independent and pairwise incomparable vectors of non-negative integers. We identify the complexity of this problem by showing that #COMPATIBLE SUBSETS is #P-complete and, thus, highly intractable. Before embarking on the proof of this result, we remind the reader that, when we speak about linear combinations, we always restrict ourselves to linear combinations with non-negative coefficients. By the same token, when we speak about linearly independent vectors, we mean independence with respect to linear combinations with nonnegative coefficients.

Theorem 12. *The counting problem #COMPATIBLE SUBSETS is #P-complete.*

Proof. Since membership in #P is quite obvious, we focus on the #P-hardness of this problem. This will be achieved by showing that #POSITIVE 2SAT has a polynomial-time 1-Turing reduction to #COMPATIBLE SUBSETS. Let $X = \{x_1, \ldots, x_k\}$ be the set of variables and let $C = \{c_1, \ldots c_m\}$ be the set of clauses of a positive 2CNF formula φ. Consider the $k \times m$ matrix $A = (a_i^j)_k^m$ such that $a_i^j = 1$, if the variable x_i appears in the clause c_j, and $a_i^j = 0$, otherwise. Without loss of generality, we can assume that no two rows of the matrix A are equal. Indeed, assume that x and y are two variables such that the corresponding rows of A are equal. If each of the variables x and y has at least two occurrences in φ, say in the clauses c_i and c_j, then we must have $c_i = c_j = x \vee y$, which is impossible, since C consists of different clauses. If each of the variables x and y has only one occurrence in φ, say in the clause c, then we can delete the clause c from the set C and consider instead the positive 2CNF formula ψ with clauses $C \setminus \{c\}$. Note that the number of satisfying assignments of φ is equal to three times the number of satisfying assignments of ψ. The reason for this is that the variables x and y do not occur in ψ, whereas the clause $c = x \vee y$ has three satisfying assignments.

Our goal is to transform A into another matrix A^* such that the rows of A^* form an instance of #COMPATIBLE SUBSETS. For this, we augment the matrix A with an additional column vector $\beta = (\beta_1, \ldots, \beta_k)$ such that each β_i is equal to one plus the number of zeros in the i-th row of A. We claim that the resulting matrix A^* has the property that all rows are pairwise incomparable in the pointwise order and that no row can be written as a linear combination with non-negative coefficients of the remaining rows.

Since no two rows of A are equal, it follows that the same property holds for the rows of A^*. Suppose that A^* has two comparable rows $a_i^* < a_j^*$. Then the relation $a_i < a_j$ must hold for the corresponding rows a_i and a_j of A. Consequently, the number of the zero entries of row a_i must be bigger than the number of the zero entries of row a_j. In turn, this implies that $\beta_i > \beta_j$, which contradicts the inequality $a_i^* < a_j^*$. Suppose now that A^* has a row a_i^* that is a linear combination with non-negative coefficients of the remaining rows. This implies that the corresponding row a_i of A is a nonnegative linear combination of the remaining rows in A. In follows that there exists a row a_j of A for some $j \neq i$ that occurs with a positive coefficient in the linear combination producing row a_i. Consequently, the inequality $a_j < a_i$ must hold. As before, this implies that $\beta_j > \beta_i$, hence the rows a_j^* and a_i^* of A^* must be incomparable. This, however, contradicts the assumption that a_i^* is a nonnegative linear combination in which a_j^* occurs with a positive coefficient.

We now claim that there is a one-to-one correspondence between satisfying truth assignments of the given 2CNF formula and compatible subsets of the set of rows of A^*. A truth assignment to the variables of φ can be identified with the set of variables that are assigned value *true*. Consequently, such a truth assignment can be identified with the subset of the rows of A^* whose corresponding variables are assigned value *true*. A truth assignment satisfies the positive 2CNF formula φ if and only if for each clause c_i there exists at least one variable x_j in c_i such that x_j is assigned value *true*. This means that the set of rows identified with the truth assignment is a compatible subset of the set of all rows of A^* (note

Counting Problem	Lower Bound	Upper Bound
#HILBERT	#P-hard	#NP
#HILBERT(3)	#P-hard	#NP
#HILBERT(2)	???	#P
#HILBERT(1)	—	FP

Table 1. The complexity of counting the Hilbert basis

that the last row of A^* consists entirely of positive entries). Vice versa, every compatible subset of A^* gives rise to exactly one satisfying assignment of φ. Thus, we have produced a polynomial-time 1-Turing reduction of #POSITIVE 2SAT to #COMPATIBLE SUBSETS. □

6 Concluding Remarks

In this paper, we initiated a study of the computational complexity of counting the Hilbert basis of a homogeneous system of linear Diophantine equations. We established that #HILBERT is in #NP and also that it is #P-hard under polynomial-time 1-Turing reductions. Admittedly, these results yield only upper and lower bounds on the complexity of this problem. Nonetheless, they appear to be the first results on the complexity of counting the Hilbert basis. Furthermore, we argued that, in view of Toda and Watanabe's work [TW92], it appears unlikely that the bounds obtained here can be made tighter without significant advances in computational complexity.

We also examined restricted variants of #HILBERT obtained by taking into account the number of occurrences of the variables in the system. In particular, we showed that if each variable has at most two occurrences, then the problem of counting the Hilbert basis is a member of the class #P (see also Table 1). Finally, we established that it is a #P-complete problem to count the number of compatible subsets of a given set of pairwise incomparable and linearly independent set of vectors.

References

[BCD90] A. Boudet, E. Contejean, and H. Devie. A new AC unification algorithm with a new algorithm for solving Diophantine equations. In *Proceedings 5th IEEE Symposium on Logic in Computer Science (LICS'90), Philadelphia (Pennsylvania, USA)*, pages 289–299, June 1990.

[BKN87] D. Benanav, D. Kapur, and P. Narendran. Complexity of matching problems. *Journal of Symbolic Computation*, 3:203–216, 1987.

[BN98] F. Baader and T. Nipkow. *Term rewriting and all that.* Cambridge University Press, 1998.

[Bou93] A. Boudet. Competing for the AC-unification race. *Journal of Automated Reasoning*, 11(2):185–212, 1993.

[CD94] E. Contejean and H. Devie. An efficient incremental algorithm for solving systems of linear Diophantine equations. *Information and Computation*, 113(1):143–172, 1994.

[CF89] M. Clausen and A. Fortenbacher. Efficient solution of linear Diophantine equations. *Journal of Symbolic Computation*, 8(1-2):201–216, 1989.

[DHJ99] A. Durand, M. Hermann, and L. Juban. On the complexity of recognizing the Hilbert basis of a linear Diophantine system. Research report 99-R-033, LORIA, Nancy, 1999.

[Dom91a] E. Domenjoud. *Outils pour la déduction automatique dans les théories associative-commutatives.* PhD thesis, Université Henri Poincaré, Nancy, France, September 1991.

[Dom91b] E. Domenjoud. Solving systems of linear Diophantine equations: An algebraic approach. In A. Tarlecki, editor, *Proceedings 16th International Symposium on Mathematical Foundations of Computer Science (MFCS'91), Kazimierz Dolny (Poland)*, volume 520 of *Lecture Notes in Computer Science*, pages 141–150. Springer-Verlag, September 1991.

[EE85] A. A. Elimam and S. E. Elmaghraby. On the reduction method for integer linear programs. II. *Discrete Applied Mathematics*, 12(3):241–260, 1985.

[Fag87] F. Fages. Associative commutative unification. *Journal of Symbolic Computation*, 3(3):257–275, 1987.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness.* W.H. Freeman and Co, 1979.

[Gor73] P. Gordan. Ueber die Auflösung linearen Gleichungen mit reellen Coefficienten. *Mathematische Annalen*, 6:23–28, 1873.

[Hil90] D. Hilbert. Ueber die Theorie der algebraischen Formen. *Mathematische Annalen*, 36:473–534, 1890.

[HK95a] M. Hermann and P. G. Kolaitis. The complexity of counting problems in equational matching. *Journal of Symbolic Computation*, 20(3):343–362, 1995.

[HK95b] M. Hermann and P. G. Kolaitis. Computational complexity of simultaneous elementary matching problems. In J. Wiedermann and P. Hájek, editors, *Proceedings 20th International Symposium on Mathematical Foundations of Computer Science (MFCS'95), Prague (Czech Republic)*, volume 969 of *Lecture Notes in Computer Science*, pages 359–370. Springer-Verlag, August 1995.

[HS87] A. Herold and J. H. Siekmann. Unification in Abelian semigroups. *Journal of Automated Reasoning*, 3(3):247–283, 1987.

[Hue78] G. Huet. An algorithm to generate the basis of solutions to homogeneous linear Diophantine equations. *Information Processing Letters*, 7(3):144–147, 1978.

[HV95] L. A. Hemaspaandra and H. Vollmer. The satanic notations: Counting classes beyond #P and other definitional adventures. *SIGACT News*, 26(1):2–13, March 1995.

[JK91] J.-P. Jouannaud and C. Kirchner. Solving equations in abstract algebras: A rule-based survey of unification. In J.-L. Lassez and G. Plotkin, editors, *Computational Logic. Essays in honor of Alan Robinson*, chapter 8, pages 257–321. MIT Press, Cambridge (MA, USA), 1991.

[Joh90] D. S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, chapter 2, pages 67–161. North-Holland, Amsterdam, 1990.

[Jub98] L. Juban. Comptage de l'ensemble des éléments de la base de Hilbert d'un système d'équations diophantiennes linéaires. Technical Report 98-R-066, LORIA, 1998.

[Lam87] J.-L. Lambert. Une borne pour les générateurs des solutions entières positives d'une équation diophantienne linéaire. *Compte-rendus de l'Académie des Sciences de Paris*, 305(1):39–40, 1987.

[Lan89] D. Lankford. Non-negative integer basis algorithms for linear equations with integer coefficients. *Journal of Automated Reasoning*, 5(1):25–35, 1989.

[LC89] P. Lincoln and J. Christian. Adventures in associative-commutative unification. *Journal of Symbolic Computation*, 8(1-2):217–240, 1989.

[Pap81] C. H. Papadimitriou. On the complexity of integer programming. *Journal of the Association for Computing Machinery*, 28(4):765–768, 1981.

[Pap94] C. H. Papadimitriou. *Computational complexity.* Addison-Wesley, 1994.

[Pot91] L. Pottier. Minimal solutions of linear Diophantine systems: bounds and algorithms. In R.V. Book, editor, *Proceedings 4th Conference on Rewriting Techniques and Applications (RTA'91), Como (Italy)*, volume 488 of *Lecture Notes in Computer Science*, pages 162–173. Springer-Verlag, April 1991.

[Sch86] A. Schrijver. *Theory of linear and integer programming.* John Wiley & Sons, 1986.

[Sti75] M. Stickel. A complete unification algorithm for associative-commutative functions. In *Proceedings 4th International Joint Conference on Artificial Intelligence (IJCAI'75), Tbilisi (USSR)*, pages 71–82, 1975.

[Sti81] M. Stickel. A unification algorithm for associative-commutative functions. *Journal of the Association for Computing Machinery*, 28(3):423–434, 1981.

[Tod89] S. Toda. On the computational power of PP and $\oplus$P. In *Proceedings 30th IEEE Symposium on Foundations of Computer Science (FOCS'89), Research Triangle Park (North Carolina, USA)*, pages 514–519, 1989.

[Tod91] S. Toda. *Computational complexity of counting complexity classes.* PhD thesis, Tokyo Institute of Technology, Department of Computer Science, Tokyo, Japan, 1991.

[TW92] S. Toda and O. Watanabe. Polynomial-time 1-Turing reductions from #PH to #P. *Theoretical Computer Science*, 100(1):205–221, 1992.

[Val79a] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[Val79b] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[Zan91] V. Zankó. #P-completeness via many-one reductions. *International Journal of Foundations of Computer Science*, 2(1):77–82, 1991.

Appendix

Proof of Theorem 11: In order to establish that the problem #HILBERT(2) belongs to #P, we need to show that the following conditions hold for systems S of homogeneous linear Diophantine equations in reduced form (we use here Lemma 9): (1) the size of every minimal solution of S is bounded by a fixed polynomial in the size of S; (2) there is a polynomial-time algorithm to test whether, given a vector s and a system S in reduced form, s is a minimal solution of S. The first condition clearly holds (for instance, by Lemma 10). Thus, it remains to show that we can verify in polynomial time that a vector s is a minimal solution of a system S. It is straightforward to verify in polynomial time that the vector s is a solution of S and that it satisfies the bounds stated in Lemma 10. To verify minimality, we give a polynomial-time reduction to the problem of finding an *alternating circuit* or an *augmenting path* in a graph, a problem that is well known to be solvable in polynomial time. Before proceeding with the rest of the proof, we define the necessary graph-theoretic concepts.

Let $G = (V, E)$ be a graph. A *matching* of G is a set $M \subseteq E$ of edges, such that no pair of edges in M share a common vertex. A vertex $v \in V$ is called *matched* by a matching M if there is an edge $e \in M$ adjacent to v. Otherwise, the vertex v is called *exposed* or *free* for M. An *alternating path* p for a matching M is a simple path in G (without repeated vertices), joining two distinct vertices, such that edges from M alternate in p with edges from $E \setminus M$. Similarly, an *alternating circuit* c for M is a simple circuit in G (without repeated vertices except the first and the last one), that alternates edges from M and from $E \setminus M$. An alternating circuit is always of even length. An alternating path for a matching M that joins two exposed vertices is called an *augmenting path*.

Given a homogeneous linear Diophantine system S over non-negative integers with each variable occurring at most twice and a solution s, we will construct a graph $G = (V, E)$ and a matching $M \subseteq E$ having the property that the solution s is *not* minimal if and only if there exists an alternating circuit or an augmenting path for M not including all vertices V.

In the first stage, we eliminate from S each variable that takes the value 0 in the solution s, and we construct a system S' in the reduced form (cf. Lemma 9) equivalent to the original system S. For each solution s of S, there exists a solution s' of S' such that s is minimal for S if and only if s' is minimal for S'. The system S' has the property that each variable occurs at most twice, once in a type A equation and once in a type B equation. Moreover, the solution s' has the property that the constraint $1 \leq s'_i \leq 2$ holds for each of its coefficients.

We will restrict further the system S', so that each variable with one occurrence appears in a type A equation. We will see later, during the transformation of the system S' to the graph G, why this restriction is necessary. Before performing this restriction, let us analyze the possible situations for a variable with one occurrence. Assume first, that a variable x with one occurrence appears in a type B equation $x = x'$. We perform a case analysis on the number of occurrences of x'. If the variable x' has one occurrence and the system S' contains also other equations, then the solution s' is *not* minimal, since a vector s^*, where we set $s^*(x) = x^*(x') = 0$ and $s^*(y) = s'(y)$ for the variables y different from both x and x', is a smaller solution of S' than s'. If the variable x' has one occurrence and $x = x'$ is the unique equation in the system S', then s' is minimal if and only if $s'(x) = s'(x') = 1$. Otherwise, if $s'(x) = s'(x') = 2$, then the solution s' is obviously not minimal. If the variable x' has two occurrences, then we can delete the equation $x = x'$ from the system S'. This can be done, since the value of the variable x is fully determined by the value of x' and the variable x' is determined by the remaining system $S \setminus \{x = x'\}$. In the last case, the systems S' and $S' \setminus \{x = x'\}$ have the same number of solutions. Hence, we get a new system $S'' = S' \setminus \{x = x'\}$ and a new solution s'', restricted to the variables of S'', such that s is a minimal solution of the system S if and only if s'' is a minimal solution of the system S''.

We construct a graph $G = (V, E)$ and a matching M form the system S'', such that s is a minimal solution of S if and only if there is no augmenting path and no alternating circuit in G. The vertices V are constructed from the solution $s'' = (s''_1, \ldots, s''_n)$. For each coefficient $s_i = 1$ we add the vertex v_i to V. Similarly, for each coefficient $s_i = 2$ we add the vertices v_i^1 and v_i^2 to V. The edges E correspond to the equations of the system S''. The set of edges E is the union $E_\mathrm{A} \cup E_\mathrm{B}$ of two disjoint sets, corresponding respectively to the type A and the type B equations. For each type A equation $x_i + x_k = x_j$ we add the four edges (x_i, x_j^1), (x_i, x_j^2), (x_k, x_j^1), (x_k, x_j^2) to E_A. For each type B equation $x_i = x_j$, such that $s_i = s_j = 1$, we add the edge (v_i, v_j) to E_B. For each type B equation, such that $s_i = s_j = 2$, we add the two edges (v_i^1, v_j^1), (v_i^2, v_j^2) to E_B. Note that the degree of each vertex $v \in V$ is at most 3 and that the set of edges E_B is a matching M of the graph G. Indeed, each vertex is incident to at most one edge from E_B, since each variable occurs at most once in a type B equation.

An augmenting path p in the graph G for a matching M has the property that the vertices on the extremities of p are exposed. We will show that each augmenting path in G for a matching M allows us to construct a new solution s^* smaller than s. In fact, the augmenting path p corresponds to a traversal of equations, alternating type A and type B equations. Moreover, the vertices on the extremities of p correspond to variables with one occurrence. Thus, we can construct a new solution s^* from s by decrementing by 1 every variable corresponding to a vertex in the path p. The variables x_i with value $s_i = 2$ are represented by two vertices v_i^1 and v_i^2, hence they can be decremented twice. An alternating circuit c of even length in the graph G for a matching M alternates edges from M with edges from $E \setminus M$. Similarly, as for augmenting paths, each alternating circuit of even length in G for a matching M allows us to construct a new solution s^* smaller than s. Once more, we decrement by 1 every variable corresponding to a vertex in the circuit c.

Note that the fact of having deleted the type A equations $x = y$ when one of the variables occurs only once, allows us to find a 1-to-1 correspondence between a new smaller solution of the system and the existence of an augmenting path or an alternating circuit in the corresponding graph. If we did not delete these equations, we would have alternating paths that are neither augmenting paths nor alternating circuits not corresponding to any solution of the system.

After the construction of the graph G with the matching M, we make correspond to each solution s^*, such that $s^* < s''$, an augmenting path or an alternating circuit of even length. Indeed, if $r_i = s''_i - s^*_i$ for each $i = 1, \ldots, n$, then there exists an augmenting path or an alternating circuit containing the vertices corresponding to the non-zero coefficients $r_i \neq 0$. If $r_i = 2$ then both vertices v_i^1 and v_i^2 are in the path or circuit. Otherwise, either v_i is present, when $s_i = 1$, or one of the vertices v_i^1, v_i^2 is present, when $s_i = 2$. We know that finding an augmenting path or an alternating circuit in a graph G for a matching M is a polynomial-time problem. Therefore we can detect the existence of a smaller solution $s^* < s$ in polynomial time. The only drawback is that s^* can be the trivial all-zero solution, when the found augmenting path or alternating circuit contains all vertices of the graph G. We must avoid this situation by deleting some edges from E, constructing a new graph G'. We cannot delete a vertex and all its adjacent edges, since then another vertex, that was matched before, can become exposed. Instead, to assure that a vertex v is *not* included into an augmenting path or an alternating circuit, it is sufficient to delete all edges from $E \setminus M$ that are adjacent to v. In this case, since no edge from the matching M is deleted, the augmenting paths and the alternating circuits of the graph G' are those also in the original graph G. Moreover, it is sufficient to choose only matched vertices v for deleting the adjacent edges from $E \setminus M$, since each exposed vertex is connected by an edge from $E \setminus M$ to a matched vertex. This is a direct consequence of alternating edges from E_{A} and E_{B}.

Let v be a matched vertex in the graph G with the matching M. Construct the graph G_v from G by deleting the edges from $E \setminus M$ adjacent to v. If there exists an augmenting path or an alternating circuit in G_v for M, then the solution s of the system S is *not* minimal. Since there are only polynomially many possibilities to choose a matched vertex in the graph G and an augmenting path or an alternating circuit can be found in polynomial time with respect to the size of the graph, we can check in polynomial time whether a vector s is a minimal solution of a homogeneous linear Diophantine system S in which every variable has at most two occurrences. This completes the proof of the theorem. $\square$