Subtractive Reductions and Complete Problems for Counting Complexity Classes*

Arnaud Durand
LACL
Dept. of Computer Science
Université Paris 12
94010 Créteil, France.
durand@univ-paris12.fr

Miki Hermann
LORIA (CNRS, UMR 7503)
BP 239
54506 Vandœuvre-lès-Nancy
France.
hermann@loria.fr

Phokion G. Kolaitis†
Computer Science Department
University of California
Santa Cruz, CA 95064, U.S.A.
kolaitis@cse.ucsc.edu

Abstract

We introduce and investigate a new type of reductions between counting problems, which we call *subtractive reductions*. We show that the main counting complexity classes #P, #NP, as well as all higher counting complexity classes $\#\cdot\Pi_k\mathrm{P}$, $k \geq 2$, are closed under subtractive reductions. We then pursue problems that are complete for these classes via subtractive reductions. We focus on the class #NP (which is the same as the class #·coNP) and show that it contains natural complete problems via subtractive reductions, such as the problem of counting the minimal models of a Boolean formula in conjunctive normal form and the problem of counting the cardinality of the set of minimal solutions of a homogeneous system of linear Diophantine inequalities.

1 Introduction and Summary of Results

Decision problems ask whether a "solution" exists, whereas counting problems ask how many different "solutions" exist. Valiant [Val79a, Val79b] developed a computational complexity theory of counting problems by introducing the class #P of functions that count the number of accepting paths of nondeterministic polynomial-time Turing machines; thus, #P captures counting problems whose underlying decision problem (is there a "solution"?) is in NP. Moreover, Valiant demonstrated that #P contains a wealth of complete problems, that is, there are problems in #P such that every problem in #P can be reduced to them via a suitable polynomial-time Turing reduction. Clearly, a counting problem is at least as hard as its underlying decision problem. Valiant's seminal discovery was that there can be a dramatic gap in inherent computational complexity

*A preliminary version of this paper appeared in *Proceedings 25th International Symposium on Mathematical Foundations of Computer Science* (MFCS 2000), M. Nielsen and B. Rovan, editors, Bratislava (Slovakia), *Lecture Notes in Computer Science*, vol. 1893, Springer-Verlag, August 2000, pp. 323-332.

†Research partially supported by NSF Grant CCR-9732041.

between a counting problem and its underlying decision problem. Specifically, Valiant [Val79a] showed that there are #P-complete problems whose underlying decision problem is solvable in polynomial time. The first problem to exhibit this "easy-to-decide, but hard-to-count" behavior was #PERFECT MATCHINGS, which is the problem of counting the number of perfect matchings in a given bipartite graph. Indeed, Valiant [Val79a] showed that #PERFECT MATCHINGS is #P-complete via polynomial-time 1-Turing reductions, that is, Turing reductions that only allow a single call to an oracle. Subsequent research in this area revealed an abundance of other natural #P-complete problems possessing these properties [Val79b, PB83, Lin86].

In addition to introducing #P, Valiant [Val79a] also developed a machine-based framework for introducing higher counting complexity classes. In this framework, the first class beyond #P is the class #NP of functions that count the number of accepting paths of polynomial-time nondeterministic Turing machines with access to NP oracles. More recently, Hemaspaandra and Vollmer [HV95] developed a predicate-based framework for introducing higher counting complexity classes, which subsumes Valiant's framework and makes it possible to introduce other counting classes that draw finer distinctions. In particular, Valiant's class #NP coincides with the class $\#\cdot$coNP of the Hemaspaandra-Vollmer framework. Wagner [Wag86b, Wag86a] also considered counting problems.

There is an extensive literature on the structural properties of higher counting complexity classes. As regards complete problems for these higher counting complexity classes, the state of affairs is rather complicated. Toda and Watanabe [TW92] showed if a problem is #P-hard via polynomial-time 1-Turing reductions, then it is also $\#\cdot$coNP-hard and $\#\cdot\Pi_k$P-hard, for each $k \geq 2$, where $\#\cdot\Pi_k$P is the counting version of the class Π_kP at the k-th level of the polynomial hierarchy PH. This surprising result yields an abundance of problems that are complete for these higher counting classes; for instance, #PERFECT MATCHINGS is such a problem. At the same time, it strongly suggests that #P, $\#\cdot$coNP, and all other higher counting classes are not closed under polynomial-time 1-Turing reductions. In turn, this means that problems like #PERFECT MATCHINGS do not capture the inherent complexity of the higher counting complexity classes. Needless to say that these classes are closed under *parsimonious* reductions, i.e., polynomial-time reductions that preserve the number of solutions. The parsimonious reductions, however, also preserve the complexity of the underlying decision problem; thus, they cannot be used to discover the existence of problems that are complete for the higher counting complexity classes and exhibit an "easy-to-decide, but hard-to-count" behavior.

In this paper, we introduce a new type of reductions between counting problems, which we call *subtractive reductions*, since they make it possible to count the number of solutions by first overcounting them and then carefully subtracting any surplus. We make a case that the subtractive reductions are perfectly tailored for the study of $\#\cdot$coNP and of the higher counting complexity classes $\#\cdot\Pi_k$P, $k \geq 2$. To this effect, we first show that each of these higher counting complexity classes is closed under subtractive reductions. We then focus on the class $\#\cdot$coNP and show that it contains natural complete problems via subtractive reductions, such as the problem of counting the minimal models of a Boolean formula in conjunctive normal form and the problem of counting the cardinality of the set of minimal solutions of a homogeneous system of linear Diophantine inequalities. These two particular counting problems have the added feature that the complexity of their underlying decision problems is lower than Σ_2P-complete, which is the complexity of the decision problem underlying $\#\Pi_1$SAT, the generic $\#\cdot$coNP-complete problem via parsimonious reductions.

2 Counting Problems and Counting Complexity Classes

A *counting problem* is typically presented using a suitable *witness* function which for every input x, returns a set of *witnesses* for x. Formally, a *witness* function is a function $w\colon \Sigma^* \longrightarrow \mathcal{P}^{<\omega}(\Gamma^*)$, where Σ and Γ are two alphabets, and $\mathcal{P}^{<\omega}(\Gamma^*)$ is the collections of all finite subsets of Γ^*. Every such witness function gives rise to the following *counting problem*: given a string $x \in \Sigma^*$, find the cardinality $|w(x)|$ of the *witness* set $w(x)$. In the sequel, we will refer to the function $w \mapsto |w(x)|$ as the *counting function* associated with the above counting problem; moreover, we will identify counting problems with their associated counting functions.

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

#SAT
Input: A Boolean formula φ in conjunctive normal form.
Output: Number of truth assignments that satisfy φ.

Valiant [Val79a] showed that #SAT is #P-*complete* via *parsimonious* reductions, that is, every counting problem in #P can be reduced to #SAT via a polynomial-time reduction that preserves the cardinalities of the witness sets. Moreover, the same holds true for the counting versions of many other NP-complete problems. Valiant's seminal discovery, however, was the existence of a plethora of problems that exhibit an "easy-to-decide, but hard-to-count" behavior. More precisely, if a counting problem is described via a witness function w, then the *underlying decision problem for w* asks: given a string x, is $w(x) \neq \emptyset$? Valiant [Val79a, Val79b] showed that there are #P-complete problems such that their underlying decision problems is solvable in polynomial time. The first important problem shown to possess these properties was #PERFECT MATCHINGS, which is the problem of counting the number of perfect matchings in a bipartite graph. Clearly, unless P = NP, #PERFECT MATCHINGS (and any other problem exhibiting the easy-to-decide, but hard-to-count behavior) cannot be #P-complete under parsimonious reductions. As it turns out, #PERFECT MATCHINGS is #P-complete via *polynomial-time 1-Turing reductions*, which are a restricted form of Turing reductions allowing a single query to an oracle. More precisely, a counting problem v is *polynomial-time 1-Turing reducible* to a counting problem w, if there is a deterministic Turing machine M that computes $|v(x)|$ in polynomial time by making a single call to an oracle that computes $|w(y)|$. Note that parsimonious reductions constitute 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 $|w(y)|$ is queried once and no computation is performed after the oracle's answer is received.

In addition to initiating the study of #P, Valiant [Val79a, Val79b] developed a framework for introducing higher counting complexity classes. 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. Note that, since there is no difference between querying the oracle or its complement, $\#\mathcal{C} = \#\mathrm{co}\mathcal{C}$ holds for every complexity class $\mathcal{C}$. In particular, we have that $\#\mathrm{NP} = \#\mathrm{coNP}$; more generally, $\#\Sigma_k\mathrm{P} = \#\Pi_k\mathrm{P}$, for every $k \geq 1$, where $\Sigma_k\mathrm{P}$ is the k-th level of the polynomial hierarchy PH and $\Pi_k\mathrm{P} = \mathrm{co}\Sigma_k\mathrm{P}$ (recall that $\Sigma_1\mathrm{P} = \mathrm{NP}$ and $\Pi_1\mathrm{P} = \mathrm{coNP}$).

More recently, researchers have introduced higher complexity counting classes using a predicate-based framework that focuses on the complexity of membership in the witness sets. Specifically, if $\mathcal{C}$ is a complexity class of decision problems, then Hemaspaandra and Vollmer [HV95] define $\#\cdot\mathcal{C}$ to be the class of all counting problems whose witness function w satisfies the following conditions:

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

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

What is the relationship between counting complexity classes in these two different frameworks? First, it is easy to verify that $\#\mathrm{P} = \#\cdot\mathrm{P}$. As regards higher counting complexity classes, the precise relationship is provided by Toda's result [Tod91], which asserts that

$$\#\cdot\Sigma_k\mathrm{P} \subseteq \#\Sigma_k\mathrm{P} = \#\cdot\mathrm{P}^{\Sigma_k\mathrm{P}} = \#\cdot\Pi_k\mathrm{P},$$

for every $k \geq 1$ (see also [HV95]). In particular, $\#\cdot\mathrm{NP} \subseteq \#\mathrm{NP} = \#\cdot\mathrm{P}^{\mathrm{NP}} = \#\cdot\mathrm{coNP}$. This result shows that the predicate-based framework not only subsumes the machine-based framework, but also makes it possible to make finer distinctions between counting complexity classes that were absent in the machine-based framework. Indeed, for each $k \geq 1$, Valiant's class $\#\Sigma_k\mathrm{P}$ (which is the same as $\#\Pi_k\mathrm{P}$) coincides with $\#\cdot\Pi_k\mathrm{P}$. Moreover, the class $\#\cdot\Pi_k\mathrm{P}$ appears to be different and, hence, larger than $\#\cdot\Sigma_k\mathrm{P}$. In particular, results by Köbler, Schöning, and Torán [KST89] imply that $\#\cdot\mathrm{NP} = \#\cdot\mathrm{coNP}$ if and only if $\mathrm{NP} = \mathrm{coNP}$.

In general, what makes a complexity class interesting is the existence of natural problems that are complete for the class. As mentioned earlier, #P is a particularly interesting complexity class because it contains natural complete problems, such as #PERFECT MATCHINGS, whose underlying decision problem is solvable in polynomial time. Do the higher counting complexity classes $\#\cdot\Pi_k\mathrm{P}$ (and $\#\cdot\Sigma_k\mathrm{P}$) contain natural complete problems and, if so, do some of these problems have an easier underlying decision problem than others? We begin exploring these questions by considering counting problems based on quantified Boolean formulas with a bounded number of quantifier alternations. In what follows, k is a fixed positive integer.

$\#\mathbf{\Pi_k}$SAT
Input: A formula $\varphi(y_1, \ldots, y_n) = \forall x_1 \exists x_2 \cdots Q_k x_k\ \psi(x_1, \ldots, x_k, y_1, \ldots, y_n)$, where ψ is a Boolean formula, each x_i is a tuple of variables, and each y_j is a variable.
Output: Number of truth assignment to the variables $y_1, \ldots, y_n$ that satisfy φ.

The counting problem $\#\Sigma_k$SAT is defined in a similar manner using formulas of the form $\exists x_1 \forall x_2 \cdots Q_k x_k\ \ \psi(x_1, \ldots, x_k, y_1, \ldots, y_n)$, where ψ is a Boolean formula, each x_i is a tuple of variables, and each y_j is a variable. The next result seems to be part of the folklore, although we have not been able to locate a specific reference to it. It can also be derived from results of Wrathall [Wra76].

Theorem 2.1 $\#\Pi_k$SAT *is* $\#\cdot\Pi_k\mathrm{P}$*-complete via parsimonious reductions. In addition, if k is odd (even), then the problem remains* $\#\cdot\Pi_k\mathrm{P}$*-complete when restricted to inputs in which the quantifier-free part is a Boolean formula in disjunctive normal form (respectively, in conjunctive normal form). Similarly,* $\#\Sigma_k$SAT *is* $\#\cdot\Sigma_k\mathrm{P}$*-complete via parsimonious reductions.*

Note that the decision problem underlying $\#\Pi_k$SAT is Σ_{k+1}SAT, which is the prototypical $\Sigma_{k+1}\mathrm{P}$-complete problem. Thus, the question becomes: are there any natural $\#\cdot\Pi_k\mathrm{P}$-complete problems such that their underlying decision problem is of lower computational complexity (i.e., lower than $\Sigma_{k+1}\mathrm{P}$-complete)? Clearly, unless $\Sigma_{k+1}\mathrm{P}$ collapses to a lower complexity class, no such problem can be $\#\cdot\Pi_k$P-complete via parsimonious reductions, which means that a broader class of reductions has to be considered. To this effect, Toda and Watanabe [TW92] proved the following surprising and quite significant result: if a counting problem is #P-hard via polynomial-time 1-Turing reductions, then it is also $\#\cdot\Pi_k$P-complete via the same reductions, for every $k \geq 1$. Consequently, #PERFECT MATCHINGS is $\#\cdot\Pi_k$P-complete via polynomial-time 1-Turing reductions. At first sight, Toda and Watanabe's theorem [TW92] can be interpreted as providing an abundance of $\#\cdot\Pi_k$P-complete problems such that their underlying decision problem is of low complexity. A moment's reflection, however, reveals that this theorem provides strong evidence that #P, #·coNP, and all other higher counting complexity $\#\cdot\Pi_k$P, $k \geq 2$, are *not* closed under polynomial-time 1-Turing reduction. Moreover, it implies that polynomial-time 1-Turing reductions cannot help us discover complete problems that embody the inherent difficulty of each counting complexity classes $\#\cdot\Pi_k$P, $k \geq 1$, and allow us to draw meaningful distinctions between these classes. Consequently, the challenge is to discover a different class of reductions that have the following two crucial properties: (1) each class $\#\cdot\Pi_k$P, $k \geq 1$, is closed under these reductions; (2) each class $\#\cdot\Pi_k$P, $k \geq 1$, contains natural problems that are complete for the class via these reductions. In what follows, we take the first steps towards confronting this challenge.

3 Subtractive Reductions

Researchers in structural complexity theory have extensively investigated various *closure properties* of #P and of certain other counting complexity classes (see [HO92, OH93]). For instance, it is well known and easy to prove that #P is closed under both addition and multiplication.[1] In turn, this has motivated researchers to introduce reductions that take advantage of closure properties. Indeed, Saluja, Subrahmanyam and Thakur [SST95] and Sharell [Sha98] used the closure of #P under addition and multiplication to introduce approximation-preserving reductions between counting problems. In particular, Sharell's [Sha98] *PL-reductions* involve positive linear combinations that approximate the desired value from below. Unfortunately, these reductions do not seem to be suited for our purposes. Instead, we adopt a different approach and introduce the class of *subtractive reductions* that first overcount and then subtract any surplus items. It should be emphasized that defining such reductions is a delicate matter, since many counting complexity classes, including #P, do *not* appear to be closed under subtraction. Specifically, Ogiwara and Hemachandra [OH93] have shown that #P is closed under subtraction if and only if the class PP of problems solvable in probabilistic polynomial time coincides with the class UP of problems solvable by an unambiguous Turing machine in polynomial time, which is considered an unlikely eventuality.

Before defining the class of subtractive reductions, we need to introduce certain auxiliary concepts and establish notation.

Let Σ, Γ be two alphabets and let $R \subseteq \Sigma^* \times \Gamma^*$ be a binary relation between strings such that, for each $x \in \Sigma^*$, the set $R(x) = \{y \in \Gamma^* \mid R(x,y)\}$ is finite. We write $\#\cdot R$ to denote the following counting problem: given a string $x \in \Sigma^*$, find the cardinality $|R(x)|$ of the witness set $R(x)$ associated with x. It is easy to see that every counting problem is of the form $\#\cdot R$ for some R.

Definition 3.1 Let Σ, Γ be two alphabets and let $\#\cdot A$ and $\#\cdot B$ be two counting problems determined by the binary relations A and B between strings from Σ and Γ.

- We say that the counting problem $\#\cdot A$ reduces to the counting problem $\#\cdot B$ via a **strong**

[1] Apparently, K. Regan was the first to observe this closure property of #P, see [HO92].

subtractive reduction, and write $\#\cdot A \leq_{ssr} \#\cdot B$, if there exist two polynomial-time computable functions f and g such that for every string $x \in \Sigma^*$:

1. $B(f(x)) \subseteq B(g(x))$;
2. $|A(x)| = |B(g(x))| - |B(f(x))|$.

- We say that the counting problem $\#\cdot A$ reduces to the counting problem $\#\cdot B$ via a **subtractive reduction**, and write $\#\cdot A \leq_{sr} \#\cdot B$, if there exists a positive integer n and a sequence of counting problems $\#\cdot A_1, \ldots, \#\cdot A_n$ such that $\#\cdot A = \#\cdot A_1$, $\#\cdot B = \#\cdot A_n$, and $\#\cdot A_i$ reduces to $\#\cdot A_{i+1}$ via a strong subtractive reduction, for each $i = 1, \ldots, n-1$.

Note that in the above definition strong subtractive reductions and subtractive reductions are defined between counting problems determined by binary relations on strings. If we consider counting problems C and D given via counting functions, then we say that C is reducible to D via a (strong) subtractive reduction if there are binary relations A and B on strings such that $C = \#\cdot A$, $D = \#\cdot B$, and $\#\cdot A$ reduces to $\#\cdot B$ via a (strong) subtractive reduction.

Clearly, parsimonious reductions constitute a special case of subtractive reductions. In general, the composition of two strong subtractive reductions need not be a strong subtractive reduction. In contrast, subtractive reductions do not suffer from this drawback. The following proposition is easily proved by induction on the length of the sequence of strong subtractive reductions.

Proposition 3.2 *Reducibility via subtractive reductions is a transitive relation. In other words, if* $\#\cdot A \leq_{sr} \#\cdot B$ *and* $\#\cdot B \leq_{sr} \#\cdot C$, *then* $\#\cdot A \leq_{sr} \#\cdot C$.

The reader familiar with the preliminary version of this paper in the Proceedings of MFCS 2000 will notice that the above Definition 3.1 of subtractive reduction is different from the definition of "subtractive reduction" presented in the Proceedings of MFCS 2000, even though both definitions contain strong subtractive reductions as a special case. Klaus W. Wagner and Heribert Vollmer discovered that our earlier definition of "subtractive reduction" was flawed in the sense that, using that earlier definition, it was impossible to show that "subtractive reductions" compose and thus Proposition 3.2 could not be established.

Next we state and prove the main result of this section; it asserts that Valiant's counting complexity classes are closed under subtractive reductions.

Theorem 3.3 $\#\mathrm{P}$ *and all higher counting complexity class* $\#\cdot\Pi_k\mathrm{P} = \#\Sigma_k\mathrm{P}$, $k \geq 1$, *are closed under subtractive reductions.*

Proof: Let k be a fixed positive integer. In what follows, we prove that the class $\#\cdot\Pi_k\mathrm{P}$ is closed under strong subtractive reductions. The result will follow by induction. Recall that Toda [Tod91] showed that $\#\cdot\Pi_k\mathrm{P} = \#\Sigma_k\mathrm{P} = \#\cdot\mathrm{P}^{\Sigma_k\mathrm{P}}$.

Let $\#\cdot A$ and $\#\cdot B$ be two counting problems such that $\#\cdot B \in \#\cdot\Pi_k\mathrm{P}$ and $\#\cdot A$ reduces to $\#\cdot B$ via a strong subtractive reduction. We will show that $\#\cdot A$ belongs to $\#\cdot\Pi_k\mathrm{P}$ by constructing a predicate A' in $\mathrm{P}^{\Sigma_k\mathrm{P}}$ such that for each string x

$$|A'(x)| \;=\; |B(g(x))| - |B(f(x))| \;=\; |A(x)|\,,$$

where f and g are the polynomial-time computable function in the subtractive reduction of $\#\cdot A$ to $\#\cdot B$. Let $*$ be a delimiter symbol not in the alphabets of the counting problems $\#\cdot A$ and $\#\cdot B$. The predicate A' consists of all pairs (x, y') of strings x and y' such that y' is of the form $f(x) * g(x) * y$ with $(g(x), y) \in B$ and $(f(x), y) \notin B$. Thus, a pair (x, y') belongs to A' if and only if (x, y') is accepted by the following algorithm:

1. extract $f(x)$, $g(x)$, and y from y';

2. check that $(g(x), y)$ belongs to B;

3. check that $(f(x), y)$ does not belong to B.

Step 1 can be carried out in polynomial time. The test in Step 2 is in Π_kP, therefore also in $\mathrm{P}^{\Sigma_k\mathrm{P}}$. The test in Step 3 is in Σ_kP, hence it can be done in $\mathrm{P}^{\Sigma_k\mathrm{P}}$. Consequently, the predicate A' is in $\mathrm{P}^{\Sigma_k\mathrm{P}}$. Moreover, it is clear that $|A(x)| = |A'(x)|$, for every string x. It follows that the counting problem $\#\cdot A$ is in $\#\cdot\mathrm{P}^{\Sigma_k\mathrm{P}} = \#\cdot\Pi_k\mathrm{P}$.

The closure of $\#P$ under subtractive reductions is established using a similar argument. □

In view of the preceding Theorem 3.3, it is natural to ask whether the classes $\#\cdot\Sigma_k\mathrm{P}$, $k \geq 1$, introduced by Hemaspaandra and Vollmer [HV95], are also closed under subtractive reductions. We now provide evidence to the effect that *no* class $\#\cdot\Sigma_k\mathrm{P}$ is closed under subtractive reductions. For this, we observe that $\#\Pi_k$SAT, the generic complete problem for $\#\cdot\Pi_k\mathrm{P}$, can easily be reduced to $\#\Sigma_k$SAT, the generic complete problem for $\#\cdot\Sigma_k\mathrm{P}$, via a strong subtractive reduction. Consequently, if $\#\cdot\Sigma_k\mathrm{P}$ were closed under subtractive reductions, then $\#\cdot\Pi_k\mathrm{P}$ would collapse to $\#\cdot\Sigma_k\mathrm{P}$, which is generally considered as highly unlikely.

Let $\varphi(y_1, \ldots, y_n)$ be any Π_k-formula $\forall x_1 \exists x_2 \cdots Q_k x_k\ \phi(x_1, \ldots, x_k, y_1, \ldots, y_n)$. Let $\bar{\varphi}(y_1, \ldots, y_n)$ be the Σ_k formula that is equivalent to $\neg\varphi$ and is obtained from φ by propagating the negation symbol through the quantifiers and applying de Morgan laws to the quantifier-free part of φ. Let $\psi(y_1, \ldots, y_n)$ be the tautology $y_1 \vee \neg y_1 \vee y_2 \vee \neg y_2 \vee \cdots \vee y_n \vee \neg y_n$. It is obvious that every satisfying truth assignment of $\bar{\varphi}$ is a satisfying truth assignment of ψ and that $|sat(\varphi)| = |sat(\psi)| - |sat(\bar{\varphi})|$ hold, where $sat(\varphi)$ denotes the satisfying truth assignments of φ (and similarly for ψ and $\bar{\varphi}$). Consequently, the polynomial-time computable functions $f(\varphi) = \bar{\varphi}$ and $g(\varphi) = \psi$ constitute a strong subtractive reduction of $\#\Pi_k$SAT to $\#\Sigma_k$SAT.

Observe that the preceding argument can also be applied to a Boolean formula φ in conjunctive normal form (i.e., assume $k = 0$) to produce a subtractive reduction of #SAT to #DNF, where #DNF is the following counting problem.

#DNF
Input: A Boolean formula θ in disjunctive normal form.
Output: Number of truth assignments that satisfy θ.

Consequently, we obtain the following result concerning #P-completeness via subtractive reductions.

Proposition 3.4 *#DNF is #P-complete via subtractive reductions.*

Observe that #DNF cannot be #P-complete via parsimonious reductions, since its underlying decision problem is easily solvable in polynomial time. As stated earlier, #PERFECT MATCHINGS is #P-complete via polynomial-time 1-Turing reductions. It is an interesting open problem to determine whether #PERFECT MATCHINGS is also #P-complete via subtractive reductions.

4 Alternative Definitions of Subtractive Reductions

Subtractive reductions, as introduced in the previous section, have the following three desirable properties: reducibility via subtractive relations is a transitive relation; each class $\#\cdot\Pi_k\mathrm{P}$ is closed under subtractive reductions; each class $\#\cdot\Pi_k\mathrm{P}$ contains natural counting problems that are $\#\cdot\Pi_k\mathrm{P}$-complete via subtractive reductions. As it turns out, the concept of "reduction by subtraction" can also be introduced in several different ways while preserving the above three properties. This section is devoted to the presentation of two such alternative definitions of the notion of "subtractive reduction". These three different definitions of subtractive reductions do not appear to be equivalent; it remains an open problem to delineate the exact relationship between these concepts. Note, however, that all completeness results presented in this paper remain true under any one of the three different definitions of subtractive reductions.

The first alternative is to deal directly with the underlying witness set within the reduction. This leads to the following modification of the definition of the strong subtractive reduction.

Definition 4.1 Let Σ, Γ be two alphabets and let A and B be two binary relations between strings from Σ and Γ. We say that the counting problem $\#\cdot A$ reduces to the counting problem $\#\cdot B$ via a **strong subtractive reduction**, and write $\#\cdot A \leq_{ssr} \#\cdot B$, if there exist two polynomial-time computable functions f and g, and a polynomial-time computable injection $h\colon A \longrightarrow B$, such that for every string $x \in \Sigma^*$:

- $B(f(x)) \subseteq B(g(x))$;
- $h(A(x)) = B(g(x)) \setminus B(f(x))$.

Compared with Definition 3.1, this new definition of a strong subtractive reduction prefers the witness set structure to the cardinality equation. Of course, Definition 4.1 implies the equality $|A(x)| = |B(g(x))| - |B(f(x))|$, what makes Definition 3.1 a special case of Definition 4.1.

Again, the subtractive reduction is defined, as previously, by a transitive closure of strong subtractive reductions. The drawback of this definition is that the notion is given in two stages: first a basic reduction relation is defined, upon which we apply the transitive closure to get the actually desired reduction. One can get rid of this feature by introducing multisets in the definition.

We first recall some basic notions of multisets. Let D be a non-empty set. Intuitively, a *multiset* on D is a collection of elements of D in which elements may have multiple occurrences. More formally, a *multiset* M on D can be viewed as a function $M\colon D \longrightarrow \mathbb{N}$ that assigns to each element $x \in D$ the number $M(x)$ of the occurrences of x in M. The multisets on D can be equipped with the operations of *union* and *difference* as follows.

Let A and B be two multisets on D. The *union* of A and B is the multiset $A \oplus B$ such that $(A \oplus B)(x) = A(x) + B(x)$ for every $x \in D$. The *difference* of A and B is the multiset $A \ominus B$ such that $(A \ominus B)(x) = \max(A(x) - B(x), 0)$ for every $x \in D$. We say that A is *contained* in B, and write $A \subseteq B$, if $A(x) \leq B(x)$ for every $x \in D$. Note that if $B \subseteq A$, then $(A \ominus B)(x) = A(x) - B(x)$ holds for all $x \in D$. Hence, whenever multiset difference is taking place between two multisets such that one is contained in the other, then the multiset operations can be replaced by the ordinary arithmetic operations. Finally, if $A_1, \ldots, A_n$ are multisets, then we write $\bigoplus_{i=1}^n A_i$ to denote the union $A_1 \oplus \cdots \oplus A_n$.

Definition 4.2 Let Σ, Γ be two alphabets and let A and B be two binary relations between strings from Σ and Γ. We say that the counting problem $\#\cdot A$ reduces to the counting problem $\#\cdot B$ via a **multiset subtractive reduction**, and write $\#\cdot A \leq_{ms} \#\cdot B$, if there exist a positive integer n, polynomial-time computable functions f_i and g_i, $i = 1, \ldots, n$, and polynomial time computable bijection h, such that for every string $x \in \Sigma^*$:

- $\bigoplus_{i=1}^n h(B(f_i(x))) \subseteq \bigoplus_{i=1}^n h(B(g_i(x)))$;
- $A(x) = \bigoplus_{i=1}^n h(B(g_i(x))) \ominus \bigoplus_{i=1}^n h(B(f_i(x)))$.

Multiset subtractive reductions compose well without any additional explicit transitivity requirement. For proving this result, we need the following basic properties of multisets whose proof is left to the reader.

Lemma 4.3 *Let A_i, B_i, for $i = 1, \ldots, n$, A, B, C, and D be multisets.*

1. *If $B_i \subseteq A_i$ for each i, then*

$$\bigoplus_{i=1}^{n}(A_i \ominus B_i) \quad = \quad (\bigoplus_{i=1}^{n} A_i) \ominus (\bigoplus_{i=1}^{n} B_i).$$

2. *If $B \subseteq A$, $D \subseteq C$, and $C \ominus D \subseteq A \ominus B$ then*

$$(A \ominus B) \ominus (C \ominus D) \quad = \quad (A \oplus D) \ominus (B \oplus C).$$

We are able now to prove that a composition of two multiset subtractive reductions produces another multiset subtractive reduction.

Theorem 4.4 *Reducibility via subtractive reductions is a transitive relation, that is, if $\#{\cdot}A \leq_{ms} \#{\cdot}B$ and $\#{\cdot}B \leq_{ms} \#{\cdot}C$, then $\#{\cdot}A \leq_{ms} \#{\cdot}C$.*

Proof: Suppose that $\#{\cdot}A$ reduces to $\#{\cdot}B$ via a multiset subtractive reduction with the functions f_i^1, g_i^1 and h^1. Suppose also that $\#{\cdot}B$ reduces to $\#{\cdot}C$ via a multiset subtractive reduction with the functions f_j^2, g_j^2 and h^2. We prove that there exists a multiset subtractive reduction from $\#{\cdot}A$ to $\#{\cdot}C$ with the functions f_k, g_k and h.

Let

$$M \quad = \quad \bigoplus_i h^1(B(g_i^1(x))) \ominus \bigoplus_i h^1(B(f_i^1(x)))$$

i.e., $|M| = |A(x)|$. Since there is a subtractive reduction from $\#{\cdot}B$ to $\#{\cdot}C$, the following equation holds for the witness set $B(g_i^1(x))$ (similarly for $B(f_i^1(x))$):

$$B(g_i^1(x)) \quad = \quad \bigoplus_j h^2(C(g_j^2.g_i^1(x))) \ominus \bigoplus_j h^2(C(f_j^2.g_i^1(x)))$$

Then the multiset M is equal to

$$\begin{aligned} &\bigoplus_i h^1(\bigoplus_j h^2(C(g_j^2.g_i^1(x))) \ominus \bigoplus_j h^2(C(f_j^2.g_i^1(x)))) \\ &\quad \ominus \quad \bigoplus_i h^1(\bigoplus_j h^2(C(g_j^2.f_i^1(x))) \ominus \bigoplus_j h^2(C(f_j^2.f_i^1(x)))). \end{aligned}$$

Function h^2 is a bijection and $\bigoplus_j h^2(C(g_j^2.g_i^1(x))) \ominus \bigoplus_j h^2(C(f_j^2.g_i^1(x)))$ is a set. Then the function h^1 can be pushed inside the multiset sum, still preserving the inclusions. Then the multiset M is equal to

$$\bigoplus_i (\bigoplus_j h^1.h^2(C(g_j^2.g_i^1(x))) \ominus \bigoplus_j h^1.h^2(C(f_j^2.g_i^1(x))))$$
$$\ominus \quad \bigoplus_i (\bigoplus_j h^1.h^2(C(g_j^2.f_i^1(x))) \ominus \bigoplus_j h^1.h^2(C(f_j^2.f_i^1(x)))).$$

Since the corresponding inclusions are satisfied, following property 1 of Lemma 4.3, the previous multiset is equal to

$$\bigoplus_i \bigoplus_j h^1.h^2(C(g_j^2.g_i^1(x))) \ominus \bigoplus_i \bigoplus_j h^1.h^2(C(f_j^2.g_i^1(x))))$$
$$\ominus \quad \bigoplus_i \bigoplus_j h^1.h^2(C(g_j^2.f_i^1(x))) \ominus \bigoplus_i \bigoplus_j h^1.h^2(C(f_j^2.f_i^1(x)))).$$

Following property 2 of Lemma 4.3, the latter multiset is equal to

$$\bigoplus_i \bigoplus_j (h^1.h^2(C(g_j^2.g_i^1(x))) \oplus h^1.h^2(C(f_j^2.f_i^1(x)))$$
$$\ominus \quad \bigoplus_i \bigoplus_j (h^1.h^2(C(f_j^2.g_i^1(x))) \oplus h^1.h^2(C(f_j^2.f_i^1(x))).$$

Hence, we choose the functions $g_j^2(g_i^1(x))$ and $f_j^2(f_i^1(x))$ for $g_k(x)$, whereas the functions $f_j^2(g_i^1(x))$ and $g_j^2(f_i^1(x))$ become the functions $f_k(x)$. Finally, we take $h^1.h^2$ for the function h. □

The closure of Valiant's counting classes under multiset subtractive reductions can be obtained by a straightforward modification of the proof of Theorem 3.3.

5 #·coNP-complete Problems via Subtractive Reductions

Many important counting problems are known to be #P-complete via polynomial-time 1-Turing reductions and have the property that their underlying decision problem is solvable in polynomial time [Val79a, Val79b, PB83, Lin86]. The current state of knowledge, however, is very different for the higher counting complexity classes $\#\cdot\Pi_k$P and $\#\cdot\Sigma_k$P, $k \geq 1$. We do know that they possess generic complete problem, such as $\#\Sigma_k$SAT and $\#\Pi_k$SAT, that are complete for these classes via parsimonious reductions, but have inherently high computational complexity (see Proposition 2.1). We also know that every counting problem that is #P-complete via polynomial-time 1-Turing reductions is also complete for these classes under the same reductions [TW92]. Up to this point, however, it is not known if these higher counting complexity classes contain any problems that have the following two properties: (1) they are complete for the class via reductions under which the class is closed; (2) their underlying decision problems has complexity lower than that of the generic complete problem for the class.

In this section, we focus on the class #·coNP and establish that it contains certain natural counting problems that possess the above two properties. Recall that #·coNP is the first higher counting complexity class that arises in Valiant's framework, since #·coNP = #NP. Moreover, it is quite robust, since, as shown by Toda [Tod91], #·coNP = #NP = $\#\cdot\mathrm{P}^{\mathrm{NP}}$.

Circumscription is a well-developed formalism of common-sense reasoning introduced by McCarthy [McC80] and extensively studied by the artificial intelligence community. The key idea behind circumscription is that one is interested in the *minimal models* of formulas, since they are the ones that have as few "exceptions" as possible and, therefore, embody common sense. In the context of Boolean logic, circumscription amounts to the study of satisfying assignments of Boolean formulas that are *minimal* with respect to the *pointwise partial order* on truth assignments. More precisely, if $s = (s_1, \ldots, s_n)$ and $s' = (s'_1, \ldots, s'_n)$ are two elements of $\{0,1\}^n$, then we write $s < s'$ to denote that $s \neq s'$ and $s_i \leq s'_i$ holds for every $i \leq n$. Let $\varphi(x_1, \ldots, x_n)$ be a Boolean formula having $x_1, \ldots, x_n$ as its variables and let $s \in \{0,1\}^n$ be a truth assignment. We say that s is a *minimal model of* φ if s is a satisfying truth assignment of φ and there is no satisfying truth assignment s' of φ such that $s < s'$. This concept gives rise to the following natural counting problem.

#CIRCUMSCRIPTION
Input: A Boolean formula $\varphi(x_1, \ldots, x_n)$ in conjunctive normal form.
Output: Number of minimal models of $\varphi(x_1, \ldots, x_n)$.

The underlying decision problem for #CIRCUMSCRIPTION is NP-complete, since a Boolean formula has a minimal model if and only if it is satisfiable. Thus, it has lower complexity than Σ_2P-complete, which is the complexity of the underlying decision problem for $\#\Pi_1$SAT, the generic problem for #·coNP.

Theorem 5.1 #CIRCUMSCRIPTION *is #·coNP-complete via subtractive reductions.*

Proof: It is clear that the problem belongs to #·coNP, since testing whether a given truth assignment is a minimal model of a given formula is in coNP (actually, this decision problem is coNP-complete [Cad92]).

For the lower bound, we construct a strong subtractive reduction of $\#\Pi_1$SAT to #CIRCUMSCRIPTION. In what follows, we write $A(F)$ to denote the set of all satisfying assignments of a Π_1-formula F; we also write $B(\psi)$ to denote the set of all minimal models of a Boolean formula ψ. Let $F(x) = \forall y\ \phi(x, y)$ be a Π_1-formula, where $\phi(x, y)$ is a Boolean formula in disjunctive normal form, and $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_m)$ are tuples of Boolean variables. Let $x' = (x'_1, \ldots, x'_n)$ be a tuple of new Boolean variables, let z be a single new Boolean variable, let $P(x, x')$ be the formula $(x_1 \equiv \neg x'_1) \wedge \cdots \wedge (x_n \equiv \neg x'_n)$, let $Q(y)$ be the formula $y_1 \wedge \cdots \wedge y_m$, and, finally, let $F'(x, x', y, z)$ be the formula

$$P(x, x') \wedge (z \rightarrow Q(y)) \wedge (\phi(x, y) \rightarrow z).$$

There is a polynomial-time computable function g such that, given a Π_1-formula F as above, it returns as value a Boolean formula $g(F)$ in conjunctive normal form that is logically equivalent to the formula $F'(x, x', y, z)$ (this is so, because $\phi(x, y)$ is in disjunctive normal form). Now let $F''(x, x', y, z)$ be the formula $F'(x, x', y, z) \wedge (z \rightarrow \neg Q(y))$ and let f be a polynomial-time computable function such that, given a Π_1-formula F as above, it returns as value a Boolean formula $f(F)$ in conjunctive normal form that is logically equivalent to the formula $F''(x, x', y, z)$.

We will show in a sequence of four claims that every minimal model of F'' is a minimal model of F' and that there is a bijection between the minimal models of F and the set difference of the minimal models of F' and F''.

Claim 1: (x, x', y, z) is a model of F' if and only if either $P(x, x') = 1$ and $Q(y) = 1$ and $z = 1$, or $P(x, x') = 1$ and $z = 0$ and $\phi(x, y) = 0$.

This is obvious from the definition of F', since $z = 1$ implies $Q(y) = 1$.

Claim 2: (x, x', y, z) is a minimal model of F' if and only if either $\phi(x, y) = 1$ for all y and $P(x, x') = 1$ and $Q(y) = 1$ and $z = 1$, or $P(x, x') = 1$ and $z = 0$ and $\phi(x, y) = 0$ and there is no y' such that $y' < y$ and $\phi(x, y') = 0$.

Consider the models $(x, x', 1, \ldots, 1, 1)$. Assume that $(x, x', 1, \ldots, 1, 1)$ is a minimal model of F'. Then for every y we must have that $\phi(x, y) = 1$, since otherwise $(x, x', y, 0)$ would be a model of F' smaller than $(x, x', 1, \ldots, 1, 1)$. Assume that x is such that $\forall y\ \phi(x, y) = 1$. Then $(x, x', 1, \ldots, 1, 1)$ is a minimal model of F', since the only way to have a smaller model would be to have one of the form $(x, x', y, 0)$ with $\phi(x, y) = 0$, which contradicts the hypothesis on x. Now, consider models of the form $(x, x', y, 0)$. From Claim 1 it follows that such a model is minimal if and only if there is no $y' < y$ such that $\phi(x, y') = 0$.

Claim 3: (x, x', y, z) is a model of F'' if and only if $P(x, x') = 1$ and $z = 0$ and $\phi(x, y) = 0$.

This follows easily from the definition of F''.

Claim 4: (x, x', y, z) is a minimal model of F'' if and only if $P(x, x') = 1$ and $z = 0$ and $\phi(x, y) = 0$ and there is no y' such that $y' < y$ and $\phi(x, y') = 0$.

This follows from the definition of F'' and Claim 3.

From Claims 1 to 4, it follows that the set difference of the minimal models of F' and F'' is equal to the set $\{(x, x', 1, \ldots, 1, 1) \mid \forall y\ \phi(x, y) \wedge P(x, x')\}$. Note that this set has the same cardinality as the set of satisfying assignments of the formula F, since the variables x' are functionally dependent on the variables x through the formula $P(x, x')$. Hence, we have that $|A(F)| = |B(F')| - |B(F'')|$, which establishes that the polynomial-time computable functions f and g constitute a strong subtractive reduction of $\#\Pi_1$SAT to #CIRCUMSCRIPTION. □

The following result is an immediate consequence of Theorems 3.3 and 5.1.

Corollary 5.2 *#·coNP = #P if and only if #CIRCUMSCRIPTION is in #P.*

We now move from counting problems in Boolean logic to counting problems in integer linear programming. A *system of linear Diophantine inequalities over the non-negative integers* is a system of the form $S\colon Ax \leq b$, where A is an integer matrix, b is an integer vector, and we are interested in the non-negative integer solutions of this system. If b is the zero-vector $(0, \ldots, 0)$, then we say that the system is *homogeneous.* A non-negative integer solution s of S is *minimal* if there is no non-negative solution s' of S such that $s' < s$ in the pointwise partial order on integer vectors. It is well known that the set of all minimal solutions plays an important role in analyzing the space of all non-negative integer solutions of linear Diophantine systems (see Schrijver [Sch86]). Clearly, every homogeneous system has $(0, \ldots, 0)$ as a trivial minimal solution. Here, we are interested in counting the number of non-trivial minimal solutions of homogeneous systems.

#HOMOGENEOUS MINIMAL SOLUTION
Input: A homogeneous system $S\colon Ax \leq 0$ of linear Diophantine inequalities.
Output: Number of non-trivial minimal solutions of S.

Note that the underlying decision problem of #HOMOGENEOUS MINIMAL SOLUTION amounts to whether a given homogeneous system of linear Diophantine inequalities has a non-negative integer solution other than the trivial solution $(0, \ldots, 0)$. It is easy to show that this problem is solvable in polynomial time, since it can be reduced to LINEAR PROGRAMMING. In contrast, counting the number of non-trivial minimal solutions turns out to be a hard problem. More precisely, #HOMOGENEOUS MINIMAL SOLUTION appears to be #·coNP-complete via subtractive reductions. As stepping stones towards proving that result, we will introduce and use two other technical counting problems.

#SATISFIABLE CIRCUMSCRIPTION
Input: A satisfiable Boolean formula $\varphi(x_1, \ldots, x_n)$ in conjunctive normal form.
Output: Number of minimal models of $\varphi(x_1, \ldots, x_n)$.

Proposition 5.3 *The counting problem* #SATISFIABLE CIRCUMSCRIPTION *is* $\#\cdot\mathrm{coNP}$*-complete via subtractive reductions.*

Proof: Deciding membership in the witness sets for this problem is in P^{NP}, because deciding satisfiability of a Boolean formula φ is in NP and deciding minimality of a model of φ is in coNP. Hence, #SATISFIABLE CIRCUMSCRIPTION belongs to $\#\cdot\mathrm{P}^{\mathrm{NP}} = \#\cdot\mathrm{coNP}$.

For the lower bound, it is not hard to verify that a strong subtractive reduction of #CIRCUMSCRIPTION to #SATISFIABLE CIRC can be obtained as follows: given a Boolean formula $\phi(x_1, \ldots, x_n)$ in conjunctive normal form the new formula

$$\psi(x_0, x_0', x_1, \ldots, x_n) \quad = \quad ((x_0 \wedge x_1 \wedge \cdots \wedge x_n) \vee (\neg x_0 \wedge \phi(x_1, \ldots, x_n))) \wedge (x_0 \not\equiv x_0').$$

The formula ψ has at least one model, namely $m_0 = (x_0 = 1, x_0' = 0, x_1 = \cdots = x_n = 1)$.

We show that m_0 is minimal for ψ. Suppose that there exists a smaller model m_0'. Then $m_0'(x_0) = 0$ or $m_0'(x_i) = 0$ for some i. If $m_0'(x_0) = 0$ then $m_0'(x_0') = 1$, hence the models m_0 and m_0' are incomparable. If $m_0'(x_i) = 0$ for some i, then $x_0 \wedge x_1 \wedge \cdots \wedge x_n = 0$. Hence, $\neg x_0 \wedge \phi(x_1, \ldots, x_n) = 1$ From this, it follows that $\neg x_0 = 1$, i.e., $m_0'(x_0) = 0$. Once again, this leads to $m_0'(x_0') = 1$ and the two models are incomparable. Since we arrive at a contradiction in both cases, it follows that m_0 is minimal.

Now, we show that $(x_1, \ldots, x_n)$ is a minimal model of ϕ if and only if $m_1 = (0, 1, x_1, \ldots, x_n)$ is a minimal model of ψ, i.e., if $x_0 = 0$ and $x_0' = 1$. Construct the new formula

$$\psi' \quad = \quad \psi \wedge x_0 \wedge x_1 \wedge \cdots \wedge x_n \wedge (x_0 \not\equiv x_0').$$

The formula ψ' has exactly one model, namely m_0. This model is therefore also minimal for ψ'.

Let $A(\phi)$ be the set of minimal solutions of ϕ and $B(\rho)$ be the set of minimal solutions of a satisfiable formula ρ. The inclusion $B(\psi') \subseteq B(\psi)$ holds, since ψ' has only one model m_0 which is also minimal for ψ. It is clear that if $(x_1, \cdots, x_n)$ is a model of ϕ then, $m_1 = (0, 1, x_1, \cdots, x_n)$ satisfies ψ. Moreover, the only model of ψ that does not satisfy ϕ is the unique model of ψ', $m_0 = (x_0 = 1, x_0' = 0, x_1 = \cdots = x_n = 1)$. This implies that the equality $|A(\phi)| = |B(\psi)| - |B(\psi')|$ holds. The formulas ψ and ψ' can be written in conjunctive normal form without exponential explosion. Hence, we have constructed a strong subtractive reduction. □

#SATISFIABLE MINIMAL SOLUTION

Input: A system $S\colon Ax \leq b$ of linear Diophantine inequalities having at least one non-negative integer solution.
Output: Number of minimal solutions of S.

Proposition 5.4 #SATISFIABLE MINIMAL SOLUTION *is* $\#\cdot\mathrm{coNP}$*-complete via subtractive reductions.*

Proof: Deciding membership in the witness sets for this problem is in P^{NP} and, hence, the problem is in $\#\cdot\mathrm{P}^{\mathrm{NP}} = \#\cdot\mathrm{coNP}$. Indeed, testing the system for solvability is in NP, whereas testing a given solution for minimality is in coNP. In both tests, we use the fact that the size of minimal solutions is bounded by a polynomial in the size of the system (see Corollary 17.1b in [Sch86, page 239]).

For the lower bound, observe that the standard reduction of Boolean satisfiability to integer linear programming also constitutes a parsimonious reduction of #SATISFIABLE CIRCUMSCRIPTION to #SATISFIABLE MINIMAL SOLUTION. □

We are able now to prove the main result of this section.

Theorem 5.5 #HOMOGENEOUS MINIMAL SOLUTION *is* #$\cdot$coNP*-complete via subtractive reductions.*

Proof: The problem is in #$\cdot$coNP, because deciding membership in the witness sets is in coNP, using the bounds in the size of minimal solutions (see the proof of Proposition 5.4).

For the lower bound, we exhibit a strong subtractive reduction from #SATISFIABLE MINIMAL SOLUTION. Let $S\colon Ax \leq b$ be a system of linear Diophantine inequalities with at least one non-negative integer solution and such that A is $k \times n$ integer matrix. First construct the system

$$S'\colon \quad Ax - b\bar{y} \leq 0, \quad 2z - t = y, \quad x_i \leq y, \quad x_i \geq y - t,$$

where $\bar{y} = (y, \ldots, y)$ is a vector of length k having the same variable y in each coordinate, and z and t are additional new variables.

Claim 1: The vector $s_0 = (x_1 = x_2 = \cdots = x_n = y = 0, z = 1, t = 2)$ is a minimal solution of S'. This is obviously a solution. The only smaller solution is the trivial all-zero solution.

Claim 2: All nontrivial minimal solutions of S', other than s_0, are of the form

$$(x_1, \ldots, x_n, y = 2k, z = k, t = 0) \quad \text{or} \quad (x_1, \ldots, x_n, y = 2k + 1, z = k + 1, t = 1).$$

Suppose that s is a solution of S' different from s_0. There are two subcases to analyze, namely when y is even or odd.

Let $y = 2k$ with $k \geq 1$. The parametric solutions of the equation $2z - t = y$ are $z = k + i$ and $t = 2i$ for each i. Whenever the inequality $i \geq 1$ holds, the solution s is greater than s_0. Therefore only the solution with $z = k$ and $t = 0$ satisfies also the additional constraint that s must be different from s_0.

Now, let $y = 2k+1$ and $k \geq 0$. The parametric solutions of the equation $2z - t = y$ are $z = k+i$ and $t = 2i - 1$ for each $i \geq 1$. Once $i \geq 2$ holds, the solution s becomes greater than s_0. Therefore only the solution with $z = k + 1$ and $t = 1$ assures that s is different from s_0.

Claim 3: There exists a minimal solution of S' with $y \geq 3$ and y odd if and only if there are no solutions for $y = 1$ and $y = 2$. If there exists a solution with $y = 1$ or $y = 2$, then there exists also a minimal solution with the same value of y. Suppose that there exists a minimal solution with $y \geq 3$ and $y = 2k + 1$, then $t = 1$. From this. it follows that $x_i \geq 2k$, for each i. We have that $k \geq 1$, since $y \geq 3$, therefore $x_i \geq 2$ holds for each i. From $2z - t = y$, $t = 1$, and $y \geq 3$ follows $z \geq 2$. Let $s_3 = (x_1 \geq 2, \ldots, x_n \geq 2, y \geq 3, z \geq 2, t = 1)$ be a minimal solution of S'. If there is a minimal solution with $y = 1$, it must have the form $s_1 = (x_1 \leq 1, \ldots, x_n \leq 1, y = 1, z = 1, t = 1)$ and s_1 is smaller than s_3, which is a contradiction. If there is a minimal solution with $y = 2$, it must have the form $s_2 = (x_1 \leq 2, \ldots, x_n \leq 2, y = 2, z = 1, t = 0)$ and s_2 is smaller than s_3, which is also a contradiction.

Claim 4: If there exists a minimal solution with y even, then this solution must be equal to the vector $(x_1 = \cdots = x_n = 2 = y, z = 1, t = 0)$. For $y = 2k$ and $t = 0$, we must have $x_1 = \cdots = y = 2k$ and $z = k$ for some $k \geq 1$. Since S' is a homogeneous system, we can divide this solution by k.

We use now the fact that the known minimal model in #SATISFIABLE CIRCUMSCRIPTION and also the known minimal solution of $Ax \leq b$ for #SATISFIABLE MINMAL SOLUTION both have a value $x_i = 0$ for some i. Hence, this solution falsifies the system of equations $x_1 = \cdots = x_n$.

After this, construct the system $S'' = S' \cup \{x_1 = \cdots = x_n = y\}$. Clearly, the system S'' has the minimal solution $s_0 = (x_1 = \cdots = x_n = 0, y = 0, z = 1, t = 2)$ and also another minimal solution $s_2 = (x_1 = \cdots x_n = 2, y = 2, z = 1, t = 0)$ when s_2 is a solution of S'. Therefore the minimal solutions of S'' are included in the minimal solutions of S'.

We know that S' has at least one minimal solution s for $y = 1$, since $S\colon Ax \leq b$ has one solution. Moreover, s is *not* a minimal solution of S''.

Let $A(S)$ be the set of minimal solutions of the system S, and let $B(S')$ and $B(S'')$ be the sets of nontrivial minimal solutions of S' and S'', respectively. From the previous reasoning follows that $B(S'') \subseteq B(S')$ and that $|A(S)| = |B(S')| - |B(S'')|$. This establishes that the polynomial-time computable functions $f(S) = S'$ and $g(S) = S''$ constitute a strong subtractive reduction of #SATISFIABLE MINIMAL SOLUTION to #HOMOGENEOUS MINIMAL SOLUTION. $\square$

Corollary 5.6 $\#\cdot$coNP = #P *if and only if* #HOMOGENEOUS MINIMAL SOLUTION *is in* #P.

To the best of our knowledge, the above result provides the first example of a counting problem whose underlying decision problem is solvable in polynomial time, but the counting problem itself is not in #P, unless higher counting complexity classes collapse to #P.

6 Concluding Remarks

We conclude by recalling Valiant's assertion from his influential paper [Val79b] to the effect that *"The completeness class for* #P *appears to be rivalled only by that for* NP *in relevance to naturally occurring computational problems."* The passage of time and the subsequent research in this area certainly proved this to be the case. We believe that the results reported here suggest that also $\#\cdot$coNP contains complete problems of computational significance. Furthermore, we believe that subtractive reductions are the right tool for investigating $\#\cdot$coNP and identifying other natural problems that are $\#\cdot$coNP-complete via these reductions. The next challenge in this vein is to determine whether #HILBERT is $\#\cdot$coNP-complete via subtractive reductions. #HILBERT is the problem of computing the cardinality of the *Hilbert basis* of a homogeneous system $S\colon Ax = 0$ of linear Diophantine equations, i.e., counting the number of non-trivial minimal solutions of such a system. We note that this counting problem was first studied by Hermann, Juban and Kolaitis [HJK99], where it was shown to be a member of $\#\cdot$coNP and also to be #P-hard under polynomial-time 1-Turing reductions.

Acknowledgements We are truly grateful to Klaus W. Wagner and to Heribert Vollmer for their valuable comments on this work and most importantly for discovering the flaw in the original definition of "subtractive reduction", as explained in the remarks following Proposition 3.2.

References

[Cad92] M. Cadoli. The complexity of model checking for circumscriptive formulae. *Information Processing Letters*, 44(3):113–118, 1992.

[HJK99] M. Hermann, L. Juban, and P. G. Kolaitis. On the complexity of counting the Hilbert basis of a linear Diophantine system. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, *Proceedings 6th International Conference on Logic for Programming and Automated Reasoning (LPAR'99), Tbilisi (Republic of Georgia)*, volume 1705 of *Lecture*

Notes in Computer Science (in Artificial Intelligence), pages 13–32, September 1999. Springer-Verlag.

[HO92] L. A. Hemachandra and M. Ogiwara. Is #P closed under subtraction? *Bulletin of the European Association for Theoretical Computer Science*, 46:107–122, February 1992.

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

[Imm99] N. Immerman. *Descriptive complexity.* Springer-Verlag, 1999.

[KST89] J. Köbler, U. Schöning, and J. Torán. On counting and approximation. *Acta Informatica*, 26(4):363–379, 1989.

[Lin86] N. Linial. Hard enumeration problems in geometry and combinatorics. *SIAM Journal on Algebraic and Discrete Methods*, 7(2):331–335, 1986.

[McC80] J. McCarthy. Circumscription — A form of non-monotonic reasoning. *Artificial Intelligence*, 13(1-2):27–39, 1980.

[OH93] M. Ogiwara and L. A. Hemachandra. A complexity theory for feasible closure properties. *Journal of Computer and System Science*, 46(3):295–325, 1993.

[PB83] J. S. Provan and M. O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12(4):777–788, 1983.

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

[Sha98] A. Sharell. *Descriptive complexity and approximation of counting functions.* PhD thesis, Université Paris 11, Orsay (France), February 1998.

[SST95] S. Saluja, K. V. Subrahmanyam, and M. N. Thakur. Descriptive complexity of #P functions. *Journal of Computer and System Science*, 50(3):493–505, 1995.

[Sto76] L. J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.

[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.

[Wag86a] K. W. Wagner. The complexity of combinatorial problems with succinct input representation. *Acta Informatica*, 23(3):325–356, 1986.

[Wag86b] K. W. Wagner. Some observations on the connection between counting and recursion. *Theoretical Computer Science*, 47(3):131–147, 1986.

[Wra76] C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):23–33, 1976.