On the Containment Problem for Linear Sets

Hans U. Simon
Department of Mathematics, Ruhr-University Bochum, Germany
hans.simon@rub.de

Abstract

It is well known that the containment problem (as well as the equivalence problem) for semilinear sets is log-complete in Π_2^p (where hardness even holds in dimension 1). It had been shown quite recently that already the containment problem for multi-dimensional linear sets is log-complete in Π_2^p (where hardness even holds for a unary encoding of the numerical input parameters). In this paper, we show that already the containment problem for 1-dimensional linear sets (with binary encoding of the numerical input parameters) is log-hard (and therefore also log-complete) in Π_2^p. However, combining both restrictions (dimension 1 and unary encoding), the problem becomes solvable in polynomial time.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases Polynomial Hierarchy, Completeness, Containment Problem, Linear Sets

Digital Object Identifier 10.4230/LIPIcs.STACS.2018.55

Acknowledgements I want to thank the reviewers of this paper for their valuable suggestions. Many thanks go to Dmitry Chistikov and Christoph Haase who pointed my attention to [2], a paper that (without mentioning this explicitly) yields the log-hardness of the containment problem for linear sets of variable dimension.

1 Introduction

The containment problem for a family of sets consists in finding an answer to the following question: given two sets of the family, is the first one a subset of the second one?

It had been shown in a very early stage of complexity theory that the containment and the equivalence problem for semilinear sets are log-complete in Π_2^p (the second level of the polynomial hierarchy) [4]. This early investigation had been motivated by the fact that, first, the equivalence problem for contextfree languages is recursively undecidable and, second, the commutative images of contextfree languages happen to be semilinear sets according to Parikh's theorem [5]. Showing inequivalence of the commutative images of two given contextfree languages would therefore demonstrate their inequivalence.

Linear sets are the basic building blocks of semilinear sets. (The latter are finite unions of linear sets.) Moreover, 1-dimensional linear sets are the central object of research in the study of numerical semigroups [6]. It was shown quite recently that the containment problem for linear sets of variable dimension is log-complete in Π_2^p, where hardness even holds when numbers are encoded in unary [2]. In this paper, we extend the latter result as follows:

1. The containment problem for 1-dimensional linear sets (with a binary encoding of numbers) is log-hard (and therefore also log-complete) in Π_2^p.
2. On the other hand, the containment problem for 1-dimensional linear sets with a unary encoding of numbers becomes is solvable in polynomial time.

Moreover, in order to prove these results, we show the following:

35th Symposium on Theoretical Aspects of Computer Science (STACS 2018).
Editors: Rolf Niedermeier and Brigitte Vallée; Article No. 55; pp. 55:1–55:12
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- The containment problem for so-called simple unary $(+,\cup)$-expressions[1] is log-hard in Π_2^p.
- The containment problem for linear sets is still log-hard in Π_2^p under a relatively strong promise. See Sections 2.5 and 3 for details.

These results might be of independent interest.

As for semilinear sets, the containment and the inequivalence problem have the same inherent complexity: both are log-complete in Π_2^p. We briefly note that the situation is different for linear sets. The equivalence problem for linear sets is easily shown to be computationally equivalent to the word problem for linear sets, and the latter is easily shown to be NP-complete. Hence, for linear sets, verifying containment is much harder than verifying equivalence.

This paper is structured as follows. In Section 2 we present the basic definitions and notations, and we mention some facts. Our main results are stated and proved in Section 3. One of these proofs is however postponed to the final Section 4 because it is a suitable modification of a similar proof of Stockmeyer (and is given for the sake of completeness). In the final Section 5, an open problem is mentioned.

2 Definitions, Notations and Facts

We assume familiarity with basic concepts from complexity theory (e.g., logspace reductions, log-hardness or log-completeness, polynomial hierarchy etc.). The complexity classes of the polynomial hierarchy will be denoted, as usual, by Σ_k^p and Π_k^p for $k = 0, 1, 2, \ldots$. We will mainly deal with the class Π_2^p on the second level of the hierarchy.

In Section 2.1, we briefly call into mind the definition of true quantified Boolean formulas which give rise to a hierarchy of problems with one log-complete problem at every level of the polynomial hierarchy. Section 2.2 contains the basic definitions that we need in connection with integer expressions. In Section 2.3, we briefly remind the reader to the definition of linear and semilinear sets. Some well known results on the inherent complexity of the containment problem for integer expressions resp. for semilinear sets are mentioned in Section 2.4. Section 2.5 briefly calls into mind the notion of promise problems.

2.1 Quantified Boolean Formulas

▶ **Definition 1** ([7])**.** Let $X_1, X_2, \ldots, X_k$ with $X_i = \{x_{i1}, x_{i2}, \ldots\}$ be disjoint collections of Boolean variables. Let $f(X_1, \ldots, X_k)$ denote any Boolean formula over (finitely many of) the variables from $X_1 \cup \ldots \cup X_k$. Let $Q_k = \exists$ if $k \geq 1$ is odd and $Q_k = \forall$ if $k \geq 1$ is even. The notation "$\exists X_i : \ldots$" means "there exists an assignment of the variables in X_i such that $\ldots$". The analogous remark applies to the notation "$\forall X_i : \ldots$". Given these notations, we define

$$\mathcal{B}_k = \{f(X_1, \ldots, X_k) :\ (\exists X_1, \forall X_2, \ldots, Q_k X_k : f(X_1, \ldots, X_k) = 1\}\ .$$

The set consisting of Boolean formulas $f(X_1, \ldots, X_k)$ outside of $\mathcal{B}_k$ is denoted as $\overline{\mathcal{B}_k}$. The subproblem of $\mathcal{B}_k$ (resp. of $\overline{\mathcal{B}_k}$) with f being a formula in conjunctive normal form is denoted as $\mathcal{B}_k^{CNF}$ (resp. as $\overline{\mathcal{B}_k}^{CNF}$). The corresponding subproblems with f being a formula in disjunctive normal form are denoted as $\mathcal{B}_k^{DNF}$ and $\overline{\mathcal{B}_k}^{DNF}$, respectively.

[1] a variant of a problem that has originally been analyzed by Stockmeyer [7]

▶ **Theorem 2** ([7]). *For any $k \geq 1$, $\mathcal{B}_k$ is log-complete in Σ_k^p. The same is true for $\mathcal{B}_k^{CNF}$ if k is odd and for $\mathcal{B}_k^{DNF}$ if k is even.*

▶ **Corollary 3.** *For any $k \geq 1$, $\overline{\mathcal{B}_k}$ is log-complete in Π_k^p. This even holds for the set $\overline{\mathcal{B}_k}^{CNF}$ if $k \geq 1$ is odd and for the set $\overline{\mathcal{B}_k}^{DNF}$ if $k \geq 1$ is even.*

▶ **Example 4.** The set $\overline{\mathcal{B}_2}^{DNF}$, which coincides with the set of all Boolean DNF-formulas $f(X_1, X_2)$ satisfying

$$\forall X_1, \exists X_2 : f(X_1, X_2) = 0 \ .$$

is log-complete in Π_2^p.

2.2 Integer Expressions

▶ **Definition 5.** Let $m \geq 1$ be a positive integer. The set $\mathcal{E}_m$ of *m-dimensional unary integer expressions*, simply called *unary integer expressions* if m is clear from context, is the smallest set with the following properties:

1. $\{0,1\}^m \subseteq \mathcal{E}_m$. The tuples $(b_1, \ldots, b_m) \in \{0,1\}^m$ are called *atomic expressions.*
2. For any $E_1, E_2 \in \mathcal{E}_m$: $(E_1 \cup E_2), (E_1 + E_2) \in \mathcal{E}_m$.

Every expression $E \in \mathcal{E}_m$ represents a set $L(E) \subseteq \mathbb{N}_0^m$ that is defined in the obvious manner.

We briefly note that the classical definition of integer expressions in [7] is different from ours: there the expressions define subsets of $\mathbb{N}_0$, and an atomic expression is a binary representation of a single number in $\mathbb{N}_0$. In other words, the classical definition deals with 1-dimensional binary expressions whereas we deal with multi-dimensional unary expressions.

Since "$\cup$" is an associative operation, we may simply write $(E_1 \cup E_2 \cup E_3 \cup \ldots \cup E_s)$ instead of $(\ldots((E_1 \cup E_2) \cup E_3) \cup \ldots \cup E_s)$. The analogous remark applies to the operation "$+$".

▶ **Definition 6.** An expression $E \in \mathcal{E}_m$ is said to be a $(+,\cup)$-*expression* if it is a sum of unions of atomic expressions. A $(+,\cup)$-expression is called *simple* if every union in the sum is the union of precisely two (not necessarily different) atomic expressions.

▶ **Example 7.** The string

$$E = ((1,1,0) \cup (0,0,0)) + ((1,0,0) \cup (1,0,0)) + ((1,1,1) \cup (0,0,0))$$

is a simple unary $(+,\cup)$-expression. It represents the set

$$L(E) = \{(1,0,0), (2,1,0), (2,1,1), (3,2,1)\} \ .$$

2.3 Linear and Semilinear Sets

▶ **Definition 8.** The set $L(\mathbf{c}, P) \subseteq \mathbb{N}_0^m$ induced by $\mathbf{c} \in \mathbb{N}_0^m$ and a finite set $P = \{\mathbf{p_1}, \ldots, \mathbf{p_k}\} \subset \mathbb{N}_0^m$ is defined as

$$L(\mathbf{c}, P) = \mathbf{c} + \langle P \rangle \ \text{ where } \ \langle P \rangle = \left\{ \sum_{i=1}^{k} a_i \cdot \mathbf{p_i} : a_i \in \mathbb{N}_0 \right\} \ .$$

The elements in P are called *periods* and $\mathbf{c}$ is called the *constant vector* of $L(\mathbf{c}, P)$. A subset L of $\mathbb{N}_0^m$ is called *linear* if $L = L(\mathbf{c}, P)$ for some $\mathbf{c} \in \mathbb{N}_0^m$ and some finite set $P \subset \mathbb{N}_0^m$. A *semilinear set* in $\mathbb{N}_0^m$ is a finite union of linear sets in $\mathbb{N}_0^m$.

2.4 Containment Problems

As mentioned already in the introduction, the *containment problem* for a family of sets consists in finding an answer to the following question: given two sets of the family, is the first one a subset of the second one? We will be mainly concerned with the containment problem for integer expressions and with the containment problem for linear and semilinear sets. We will assume that the dimension m of sets in $\mathbb{N}_0^m$ is part of the input unless we explicitly talk about an m-dimensional problem for some fixed constant m. The following is known:

1. The containment problem for 1-dimensional binary integer expressions is log-complete in Π_2^p [7].
2. The containment problem for semilinear sets is log-complete in Π_2^p [4]. The log-hardness in Π_2^p even holds either when numbers are encoded in unary or when the dimension is fixed to 1.
3. The containment problem for linear sets is log-complete in Π_2^p [2]. The log-hardness in Π_2^p even holds when numbers are encoded in unary.

The first two hardness results are shown by means of a logspace reduction from $\overline{\mathcal{B}_2}^{DNF}$ to the respective containment problem. A suitable modification of Stockmeyer's reduction from $\overline{\mathcal{B}_2}^{DNF}$ to the containment problem for 1-dimensional binary integer expressions leads to the following result:

▶ **Theorem 9.** *The containment problem for simple unary* $(+,\cup)$*-expressions is* log*-complete in* Π_2^p.

The proof of this theorem will be given in Section 4.

Notation for Vectors: The j-th component of a vector $\mathbf{x}$ is denoted as x_j or, occasionally, as $\mathbf{x}[j]$. The latter notation is used, for instance, if there is a sequence of vectors, say $\mathbf{x_1},\ldots,\mathbf{x_n}$. The j-th component of $\mathbf{x_i}$ is then denoted as $\mathbf{x_i}[j]$ (as opposed to $x_{i,j}$ or $(x_i)_j$). Throughout the paper, we use $\mathbf{a^m}$ (with $a \in \mathbb{N}_0$) as a short notation for $(a,\ldots,a) \in \mathbb{N}_0^m$. For instance $\mathbf{1^m}$ denotes the all-ones vector in $\mathbb{N}_0^m$. The vector with value 1 in the i-th component and zeros in the remaining $m-1$ components is denoted as $\mathbf{e_i^m}$.

2.5 Promise Problems

A decision problem (without promise) is a problem with "yes"- and "no"-instances. A *promise problem* is a decision problem augmented by a promise that the input instances passed to an algorithm satisfy a certain condition. An algorithm needs to solve the promise problem only on the input instances that satisfy this condition. It may output anything on the remaining instances. Hence a promise problem has besides the "yes"- and the "no"-instances a third kind of instances: the ones that violate the promised condition. Decision problem can be viewed as promise problems with an empty promise. Reductions between promise problems should map "yes"-instances (resp. "no"-instances) of the first problem to "yes"-instances (resp. "no"-instances) of the second problem.

3 Main Results

The first result in this section will be concerned with the containment problem for linear sets when the latter is viewed as the following promise problem.

Instance: dimension m, finite sets $P, Q \subset \mathbb{N}_0^m$, vectors $\mathbf{c},\mathbf{d} \in \mathbb{N}_0^m$ and $s \in \{1,\ldots,|P|\}$.

Question: $L(\mathbf{c},P) \subseteq L(\mathbf{d},Q)$?

Promise: Let $P = \{\mathbf{p_1},\ldots,\mathbf{p_k}\}$ and let $K_s = \left\{a \in \{0,1\}^k \,\middle|\, \sum_{i=1}^k a_i = s\right\}$. With this notation, the following holds:

$$\forall a \in \mathbb{N}_0^k \setminus K_s : \sum_{i=1}^{k} a_i \cdot \mathbf{p_i} \in L(\mathbf{d},Q) \ . \tag{1}$$

In other words: we make the promise that the inclusion $L(\mathbf{c},P) \subseteq L(\mathbf{d},Q)$ can possibly fail only on linear combinations of $\mathbf{p_1},\ldots,\mathbf{p_k}$ with coefficient vectors taken from K_s.

In [2], it was shown that the containment problem for linear sets is log-hard in Π_2^p. We strengthen this result by showing that even the corresponding promise problem exhibits this kind of hardness. This slightly stronger result will later help us to prove the hardness of the containment problem for 1-dimensional linear sets.

▶ **Theorem 10.** *The containment problem for linear sets is* log*-hard in Π_2^p even under the promise (1) and even when numbers are encoded in unary.*

Proof. We will describe a logspace reduction from the containment problem for simple unary $(+,\cup)$-expressions to the containment problem for linear sets. An instance of the former problem is of the form

$$E = \sum_{i=1}^{s}(\mathbf{B_{i1}} \cup \mathbf{B_{i2}}) \ \text{ and } \ E' = \sum_{i=1}^{s'}(\mathbf{B'_{i1}} \cup \mathbf{B'_{i2}}) \tag{2}$$

where $\mathbf{B_{i1}},\mathbf{B_{i2}},\mathbf{B'_{i1}},\mathbf{B'_{i2}} \in \{0,1\}^m$. Note that we may set $s' = s$ because we could add sum-terms of the form $(\mathbf{0^m} \cup \mathbf{0^m})$ to the expression which has fewer terms. Our goal is to design $(2m+2s)$-dimensional linear sets $\mathbf{c} + \langle P\rangle$ and $\langle P' \cup P''\rangle$ such that

$$L(E) \subseteq L(E') \Leftrightarrow \mathbf{c} + \langle P\rangle \subseteq \langle P' \cup P''\rangle \ . \tag{3}$$

Intuitively, we should think of vectors from $\mathbb{N}_0^{2m+2s}$ as being decomposed into four sections of dimension m,s,s,m, respectively. The first section is called the "base section"; the latter three are called "control sections". The constant vector $\mathbf{c}$ and the periods in $P = \{\mathbf{p_{ij}} : i \in [s], j \in [2]\}$ are chosen as follows:

$$\mathbf{c} = (\mathbf{0^m},\mathbf{2^s},\mathbf{1^s},\mathbf{1^m}) \ \text{ and } \ \mathbf{p_{ij}} = (\mathbf{B_{ij}},\mathbf{e_i^s},\mathbf{0^s},\mathbf{0^m}) \ . \tag{4}$$

Note that the base section of the periods in P contains the atomic sub-expressions of E. The vectors in $\mathbb{N}_0^{2m+2s}$ having $(\mathbf{3^s},\mathbf{1^s},\mathbf{1^m})$ in their control sections are said to be "essential". It is evident that

$$L(E) \times \{3\}^s \times \{1\}^s \times \{1\}^m = (\mathbf{c} + \langle P\rangle) \cap (\mathbb{N}_0^m \times \{3\}^s \times \{1\}^s \times \{1\}^m) \ .$$

In other words: the set of base sections of the essential vectors in $\mathbf{c} + \langle P\rangle$ coincides with $L(E)$. The periods in $P' = \{\mathbf{p'_{ij}} : i \in [s], j \in [2]\}$ are similarly defined as the periods in P:

$$\mathbf{p'_{ij}} = \begin{cases} (\mathbf{B'_{ij}}, 3\cdot\mathbf{e_i^s},\mathbf{e_i^s},\mathbf{0^m}) & \text{if } i \in [s-1] \\ (\mathbf{B'_{sj}}, 3\cdot\mathbf{e_s^s},\mathbf{e_i^s},\mathbf{1^m}) & \text{if } i = s \end{cases} \ .$$

Clearly,

$$L(E') \times \{3\}^s \times \{1\}^s \times \{1\}^m = \langle P'\rangle \cap (\mathbb{N}_0^m \times \{3\}^s \times \{1\}^s \times \{1\}^m) \ .$$

Note that $L(E) \subseteq L(E')$ iff any essential vector in $\mathbf{c} + \langle P\rangle$ is contained in $\langle P'\rangle$. In order to get the desired equivalence (3), we will design P'' such that the following holds:

Claim 1: Any inessential vector from $\mathbf{c} + \langle P \rangle$ is contained in $\langle P'' \rangle$.

Claim 2: Any essential vector in $\mathbf{c} + \langle P \rangle$ is contained in $\langle P' \cup P'' \rangle$ only if it is already contained in $\langle P' \rangle$.

It is evident that (3) is valid if P'' can be defined in accordance with the two above claims. Let $n = 1 + \max\{x_i : \mathbf{x} \in L(E), i \in [m]\}$, i.e., $n-1$ is the largest number that occurs in a component of some vector in $L(E)$. We now set $P'' = P''_1 \cup P''_2$ where

$$\begin{aligned} P''_1 &= \{(\mathbf{0^m}, 2 \cdot \mathbf{e_i^s}, \mathbf{1^s}, \mathbf{0^m}), (\mathbf{0^m}, 2 \cdot \mathbf{e_i^s}, \mathbf{0^s}, \mathbf{0^m}), (\mathbf{0^m}, 3 \cdot \mathbf{e_i^s}, \mathbf{0^s}, \mathbf{0^m}) : i \in [s]\} \ , \\ P''_2 &= \{(r \cdot \mathbf{e_i^m}, \mathbf{0^s}, \mathbf{0^s}, \mathbf{e_i^m}), (n \cdot \mathbf{e_i^m}, \mathbf{0^s}, \mathbf{0^s}, \mathbf{0^m}) : i \in [m], r \in \{0, 1, \ldots, n-1\}\} \ . \end{aligned}$$

The proof of the theorem can now be accomplished by showing that the above two claims are valid for our definition of P'' (and by adding some easy observations).

Proof of Claim 1: Let $\mathbf{x} \in \mathbf{c} + \langle P \rangle$ be inessential. An inspection of (4) reveals that there must exist an index $i_0 \in [s]$ such that the i_0-th component of the first control section of $\mathbf{x}$ has a value that differs from 3. Since already the constant vector $\mathbf{c}$ makes a contribution of 2 in this control section, the possible values for x_{m+i_0} are $2, 4, 5, 6, \ldots$. In order to cast $\mathbf{x}$ as a member of $\langle P'' \rangle$, we first pick the vector $\mathbf{u} = (\mathbf{0^m}, 2 \cdot \mathbf{e^s_{i_0}}, \mathbf{1^s}, \mathbf{0^m})$. Note that $\mathbf{u} \le \mathbf{x}$ and $\mathbf{u}$ already coincides with $\mathbf{x}$ in the second control section. Adding to $\mathbf{u}$ properly chosen multiples of vectors of the form $(\mathbf{0^m}, 2 \cdot \mathbf{e_i^s}, \mathbf{0^s}, \mathbf{0^m})$ or $(\mathbf{0^m}, 3 \cdot \mathbf{e_i^s}, \mathbf{0^s}, \mathbf{0^m})$, we obtain a vector $\mathbf{v} \le \mathbf{x}$ that coincides with $\mathbf{x}$ also in the first control section. Consider now the entries of $\mathbf{v}$ and $\mathbf{x}$ in the base section. For any $i \in [m]$, consider the decomposition $x_i - v_i = q_i n + r_i$ with $q_i \ge 0$ and $0 \le r_i \le n-1$. Adding to v the vector

$$\sum_{i=1}^{m} (q_i \cdot (n \cdot \mathbf{e_i^m}, \mathbf{0^s}, \mathbf{0^s}, \mathbf{0^m}) + (r_i \cdot \mathbf{e_i^m}, \mathbf{0^s}, \mathbf{0^s}, \mathbf{e_i^m})) \ ,$$

we obtain a vector that coincides with $\mathbf{x}$ (since, by now, it also coincides with $\mathbf{x}$ in the base section and in the third control section).

Proof of Claim 2: Let $\mathbf{x} \in \mathbf{c} + \langle P \rangle$ be essential and suppose that $\mathbf{x} \in \langle P' \cup P'' \rangle$. A representation of $\mathbf{x}$ as a member of $\langle P' \cup P'' \rangle$ cannot make use of a vector of the form $(\mathbf{0^m}, 2 \cdot \mathbf{e_i^s}, \mathbf{1^s}, \mathbf{0^m})$ because there is no way to extend the value 2 in the i-th component of the first control section to 3 (since any period in $P' \cup P''$ adds either 0 or a value greater than 1 to this component). Given that we do not employ these vectors, it follows that any representation of $\mathbf{x}$ as a member of $\langle P' \cup P'' \rangle$ must be of the form $\mathbf{x} = \mathbf{x}' + \mathbf{x}''$ for some essential vector $\mathbf{x}' \in \langle P' \rangle$ and some vector $\mathbf{x}'' \in \langle P' \cup P'' \rangle$ (because, without employing an essential vector from $\langle P' \rangle$, we wouldn't get $\mathbf{1^s}$ into the second control section). Since $\mathbf{x}'$ is essential, it will already contribute $(\mathbf{3^s}, \mathbf{1^s}, \mathbf{1^m})$ to the three control sections. It follows that $\mathbf{x}'' = \mathbf{0^{2m+2s}}$ because adding any period from $P' \cup P''$ to $\mathbf{x}'$ will destroy the pattern $(\mathbf{3^s}, \mathbf{1^s}, \mathbf{1^m})$ in the control sections or will induce a component of value at least n in the base section (which is larger than any component of $\mathbf{x}$ in the base section). It follows that $\mathbf{x} = \mathbf{x}' \in \langle P' \rangle$.

It can be shown by standard arguments that the transformation $(E, E') \mapsto (\mathbf{c}, P, P', P'')$ is logspace-computable (even when numbers are encoded in unary). Finally observe that the above definition of essential vectors implies that every essential vector from $\mathbf{c} + \langle P \rangle$ employs a coefficient vector from $\{0,1\}^{|P|}$ with precisely s ones. Since any inessential vector from $\mathbf{c} + \langle P \rangle$ also belongs to $\langle P' \rangle \subseteq \langle P' \cup P'' \rangle$, the promised condition (1) is satisfied (with $P' \cup P''$ at the place of Q). This concludes the proof. ◀

We will show in the sequel that the containment problem for 1-dimensional linear sets (with numerical input parameters given in binary representation) is log-hard in Π^p_2. To this end, we will make use of the following result on the aggregation of diophantine equations:

▶ **Lemma 11** ([3]). *Let*

$$\sum_{j=1}^{r} a_{1j}x_j = b_1 \quad \textit{and} \quad \sum_{j=1}^{r} a_{2j}x_j = b_2 \tag{5}$$

be a system of two linear diophantine equations where a_{1j}, a_{2j} are non-negative integers and b_1, b_2 are strictly positive integers. Let t_1, t_2 be positive integers satisfying the following conditions:

1. *t_1 and t_2 are relatively prime.*
2. *t_1 does not divide b_2 and t_2 does not divide b_1.*
3. *$t_1 > b_2 - a_2$ and $t_2 > b_1 - a_1$ where a_i denotes the smallest nonzero coefficient in $\{a_{i1}, \ldots, a_{ir}\}$.*

Then, restricting x_j to non-negative integers, the solution set of (5) is the same as the solution set of

$$t_1 \cdot \sum_{j=1}^{r} a_{1j}x_j + t_2 \cdot \sum_{j=1}^{r} a_{2j}x_j = t_1 \cdot b_1 + t_2 \cdot b_2 \ .$$

Note that

$$t_1 = 1 + \max\{b_1, b_2\} \ \text{ and } t_2 = 1 + t_1 \tag{6}$$

is among the choices for t_1, t_2 such that the three conditions mentioned in Theorem 11 are satisfied.

From Lemma 11, the following result can be derived:

▶ **Lemma 12.** *Let $\mathbf{c} \in \mathbb{N}_0^m$, $A \in \mathbb{N}_0^{m\times r}$ and $A' \in \mathbb{N}_0^{m\times r'}$. Let $A_1, \ldots, A_m$ and $A'_1, \ldots, A'_m$ denote the row vectors of A and A', respectively. Let $s \geq 1$ and*

$$K_s = \{\mathbf{x} \in \{0,1\}^r : \sum_{i=1}^{r} x_i = s\} \ .$$

Suppose that the following holds:

$$\forall \mathbf{x} \in \mathbb{N}_0^r \setminus K_s, \exists \mathbf{y} \in \mathbb{N}_0^{r'} : \mathbf{c} + A\mathbf{x} = A'\mathbf{y} \ . \tag{7}$$

Then there exist $t_1^*, \ldots, t_m^* \in \mathbb{N}$ such that

$$(\forall \mathbf{x} \in \mathbb{N}_0^r, \exists \mathbf{y} \in \mathbb{N}_0^{r'} : \mathbf{c} + A\mathbf{x} = A'\mathbf{y}) \Leftrightarrow \\ \left(\forall \mathbf{x} \in \mathbb{N}_0^r, \exists \mathbf{y} \in \mathbb{N}_0^{r'} : \sum_{j=1}^{m} t_j^* c_j + \sum_{j=1}^{m} t_j^* A_j \mathbf{x} = \sum_{j=1}^{m} t_j^* A'_j \mathbf{y}\right) \ . \tag{8}$$

Moreover, the aggregation coefficients $t_1^*, \ldots, t_m^*$ are logspace-computable from c, A, A'.

Proof. A solution for a system of m diophantine equations is always a solution for a single equation that represents an aggregation of the m given equations (regardless of how the aggregation coefficients $t_1^*, \ldots, t_m^*$ are chosen). Hence the equivalence (8) certainly holds for every $\mathbf{x} \in \mathbb{N}_0^r \setminus K_s$ and the direction "⇒" certainly holds for every $\mathbf{x} \in K_s$. Therefore, we need to verify only that there exist $t_1^*, \ldots, t_m^* \in \mathbb{N}$ such that the following implication is valid:

$$(\exists \mathbf{x} \in K_s, \forall \mathbf{y} \in \mathbb{N}_0^{r'} : \mathbf{c} + A\mathbf{x} \neq A'\mathbf{y}) \Rightarrow \\ \left(\exists \mathbf{x} \in K_s, \forall \mathbf{y} \in \mathbb{N}_0^{r'} : \sum_{j=1}^{m} t_j^* c_j + \sum_{j=1}^{m} t_j^* A_j \mathbf{x} \neq \sum_{j=1}^{m} t_j^* A'_j \mathbf{y}\right) \ . \tag{9}$$

It is evident that (9) follows from (7) if A is the all-zeros matrix. We assume therefore in the sequel that A has at least one entry in $\mathbb{N}$. Clearly $\mathbf{c} + A\mathbf{x} = A'\mathbf{y}$ can be written in the form

$$\left[\begin{array}{cc} -A & A' \end{array} \right] \left(\begin{array}{c} \mathbf{x} \\ \mathbf{y} \end{array} \right) = \mathbf{c} \ .$$

Moreover, for $\mathbf{x} \in K_s$ and any $u > 0$, it can be written as follows:

$$\left[\begin{array}{cc} (uJ - A) & A' \end{array} \right] \left(\begin{array}{c} \mathbf{x} \\ \mathbf{y} \end{array} \right) = \mathbf{c} + u \cdot s \cdot \mathbf{1^m} \ . \tag{10}$$

Here J denotes the $m \times r$ all-ones matrix. Setting u equal to the largest absolute value of an entry in the matrix $-A$, the matrix $uJ - A$ has non-negative entries. Note that $u \geq 1$ since A has at least one entry in $\mathbb{N}$. Hence $\mathbf{c} + us\mathbf{1^m} \in \mathbb{N}^m$ so that we may bring Lemma 11 into play. Actually, we will apply this lemma iteratively in stages. In the first stage, we decompose the m diophantine equations in (10) into $m/2$ pairs, and we aggregate every pair into a single equation (by virtue of Lemma 11). After Stage 1, we are left with $m/2$ diophantine equations. Iterating this procedure for a total of $\lceil \log(m) \rceil$ stages, we finally arrive at a single diophantine equation whose solution space in $\mathbb{N}_0^{r+r'}$ coincides with the solution space for (10) in $\mathbb{N}_0^{r+r'}$. Moreover, for all $(\mathbf{x}, \mathbf{y}) \in K_s \times \mathbb{N}_0^{r'}$, it even coincides with the solution space for $\mathbf{c} + A\mathbf{x} = A'\mathbf{y}$. Hence the implication (9) is valid, as desired.
Since, in any individual application of Lemma 11, the coefficients t_1, t_2 can be chosen according to (6), the final aggregation coefficients $t_1^*, \ldots, t_m^*$ are easy to compute and, in fact, logspace computable from $\mathbf{c}, A, A'$ if all details are filled in properly. ◀

We are ready now for the next result:

▶ **Theorem 13.** *The containment problem for* 1*-dimensional linear sets is* log*-hard in* Π_2^p*.*

Proof. We reuse the notations from the proof of Theorem 10. Within that proof, we described a transformation $(E, E') \mapsto (\mathbf{c}, P, P', P'')$ which maps an instance of the containment problem for simple unary $(+, \cup)$-expressions into an instance of the containment problem for linear sets such that the latter satisfies the promised condition $(1)^2$ and such that the equivalence (3) is valid. Let d denote denote the dimension of the linear sets $\mathbf{c} + \langle P \rangle$ and $\langle P' \cup P'' \rangle$. Moreover let $r = |P|$ and $r' = |P' \cup P''|$. Let A be the $(d \times r)$-matrix with the periods from P as column vectors. Similarly, let A' be the $(d \times r')$- matrix with periods from $P' \cup P''$ as column vectors. It follows immediately from (3) that $L(E) \subseteq L(E')$ iff

$$\forall \mathbf{x} \in \mathbb{N}_0^r, \exists \mathbf{y} \in \mathbb{N}_0^{r'} : \mathbf{c} + A\mathbf{x} = A'\mathbf{y} \ .$$

Note that condition (1), written in matrix notation, translates into (7). According to Lemma 12, there exist $t_1^*, \ldots, t_m^*$ such that the equivalence in (8) is valid. Setting $c_0 = \sum_{j=1}^m t_j^* c_j$, $q_i = \sum_{j=1}^m t_j^* A_{ji}$ for $i = 1, \ldots, r$, and $q_i' = \sum_{j=1}^m t_j^* A'_{ji}$ for $i = 1, \ldots, r'$, $Q = \{q_1, \ldots, q_r\}$ and $Q' = \{q_1', \ldots, q_{r'}'\}$, we obtain a transformation $(E, E') \mapsto (c_0, Q, Q')$, which witnesses that the containment problem for 1-dimensional linear sets is log-hard in Π_2^p. ◀

Combining the restrictions of dimensionality 1 and unary encoding of numbers, the containment problem for linear sets becomes solvable in polynomial time:

▶ **Theorem 14.** *The containment problem for* 1*-dimensional linear sets with a unary encoding of numbers is in P.*

[2] with $P' \cup P''$ at the place of Q

Proof. Consider an input instance given by (the unary encoding of) c, P, c', P' with $c, c' \in \mathbb{N}_0$ and $P, P' \subset \mathbb{N}$. Let g (resp. g') be the greatest common divisor of the periods in P (resp. in P'). We make the following observation:

Claim: The containment $c + \langle P \rangle \subseteq c' + \langle P' \rangle$ is possible only if $c' \leq c$ and if g' is a divisor of g and of $c - c'$.

Given the assertion in the claim, we can accomplish the proof as follows. Setting $c_0 = c - c'$, our original question, "$c + \langle P \rangle \subseteq c' + \langle P' \rangle$?", is equivalent to "$c_0 + \langle P \rangle \subseteq \langle P' \rangle$?". We may now even assume that $g' = 1$ (because, if necessary, we can divide all numerical parameters by g'). If 1 is among the periods of P', then the answer to "$c_0 + \langle P \rangle \subseteq \langle P' \rangle$?" is clearly "yes". Suppose now that $1 \notin P'$. It is well known that $\langle P' \rangle$ contains all but finitely many natural numbers [6]. Let $F(P')$ (called the *Frobenius number* of P') denote the largest number in $\mathbb{N}$ that is not contained in $\langle P' \rangle$. It is well known that $F(P') < (\max(P') - 1) \cdot (\min(P') - 1)$ [1]. The questions "$x \in c_0 + \langle P \rangle$?" and "$x \in \langle P' \rangle$?" can be answered for all $x < (\max(P') - 1) \cdot (\min(P') - 1)$ in the obvious way by dynamic programming. Given the answers to these questions, we can immediately decide whether $c_0 + \langle P \rangle \subseteq \langle P' \rangle$.

All that remains to be done is proving the above claim. Suppose that

$$c + \langle P \rangle \subseteq c' + \langle P' \rangle \ . \tag{11}$$

This obviously implies that $c' \leq c$. It is furthermore obvious that $\langle P \rangle \subseteq g \cdot \mathbb{N}_0$ and $\langle P' \rangle \subseteq g' \cdot \mathbb{N}_0$. Moreover, by the definition of the Frobenius number, $s := g \cdot F\left(\left(\frac{1}{g} \cdot \langle P \rangle\right)\right)$ is the largest multiple of g that does not belong to $\langle P \rangle$. Hence $c + s + g, c + s + 2g \in c + \langle P \rangle$ and, because of (11), there must exist $q_2 > q_1 \geq 1$ such that $c + s + g = c' + q_1 g'$ and $c + s + 2g = c' + q_2 g'$. Now we obtain $g = (q_2 - q_1)g'$ so that g' is a divisor of g. Since $(c - c') + \langle P \rangle \subseteq \langle P' \rangle \subseteq g' \cdot \mathbb{N}_0$ and $\langle P \rangle$ contains only multiples of g' (because it only contains multiples of g), it follows that g' must also be a divisor of $c - c'$, which concludes the proof of the claim and the proof of the theorem. ◀

4 Proof of Theorem 9

It is easy to see that the containment problem for simple unary $(+, \cup)$-expressions is a member of the complexity class Π_2^p. In somewhat more detail, let E and E' be two simple unary expressions of the form (2). Then $L(E) \subseteq L(E')$ iff

$$\forall a \in \{1,2\}^s, \exists a' \in \{1,2\}^{s'} : \sum_{i=1}^{s} \mathbf{B}_{\mathbf{ia_i}} = \sum_{i=1}^{s'} \mathbf{B}'_{\mathbf{ia'_i}} \ .$$

The membership in Π_2^p is now immediate from a well known characterization of Π_2^p due to Wrathall [8]: $L \in \Pi_2^p$ iff there exists a polynomial q and a language $L_0 \in \mathrm{P}$ such that

$$L = \{x | (\forall y_1 \text{ with } |y_1| \leq q(|x|))(\exists y_2 \text{ with } |y_2| \leq q(|x|)) : \langle y_1, y_2, x \rangle \in L_0\} \ .$$

It remains to show that it is log-hard in Π_2^p. To this end, we will design a logspace reduction from $\overline{\mathcal{B}_2}^{DNF}$ to this problem. Let $f(X_1, X_2)$ be an instance of $\overline{\mathcal{B}_2}^{DNF}$ (as described in Example 4). Since f employs only finitely many variables, we may assume that $X_i = \{x_{i1}, \ldots, x_{in}\}$ for $i = 1, 2$ and some $n \geq 1$. As a DNF-formula, f is the disjunction of Boolean monomials, say $f = M_1 \vee \ldots \vee M_m$. We may clearly assume that none of the monomials contains the same variable twice. We will transform $f(X_1, X_2)$ into simple unary $(+, \cup)$-expressions E_1 and E_2 such that

$$(\forall X_1, \exists X_2 : f(X_1, X_2) = 0) \Leftrightarrow (L(E_1) \subseteq L(E_2)) \ . \tag{12}$$

For all $i = 1, \dots, n$ and $j = 1, \dots, m$, let

$$\mathbf{b_{1i}}[j] = \begin{cases} 1 & \text{if } x_{1i} \in M_j \\ 0 & \text{otherwise} \end{cases} ,$$

i.e., the binary vector $\mathbf{b_{1i}} \in \{0,1\}^m$ indicates in which monomials the variable x_{1i} actually occurs. Let $\mathbf{b'_{1i}} \in \{0,1\}^m$ denote the corresponding vector with indicator bits for the occurrences of $\overline{x_{1i}}$ within $M_1, \dots, M_m$. Let the vectors $\overline{\mathbf{b_{1i}}}$ and $\overline{\mathbf{b'_{1i}}}$ be obtained from $\mathbf{b_{1i}}$ and $\mathbf{b'_{1i}}$, respectively, by bitwise negation. Clearly, the bits of these vectors indicate the non-occurrences of x_{1i} resp. $\overline{x_{1i}}$ within $M_1, \dots, M_m$. Let $\mathbf{b_{2i}}, \mathbf{b'_{2i}}, \overline{\mathbf{b_{2i}}}, \overline{\mathbf{b'_{2i}}}$ be the corresponding vectors with indicator bits for the occurrences resp. non-occurrences of the variable x_{2i}. We now define a couple of $(+, \cup)$-expressions:

$$E'_1 = \sum_{i=1}^{n} (\mathbf{1^m} \cup \mathbf{1^m}) \quad \text{and} \quad E_1 = E'_1 + \sum_{i=1}^{n} (\overline{\mathbf{b_{1i}}} \cup \overline{\mathbf{b'_{1i}}})$$

$$E'_2 = \sum_{j=1}^{m} \sum_{i=1}^{2n-1} (\mathbf{e_j^m} \cup \mathbf{0^m}) \quad \text{and} \quad E_2 = E'_2 + \sum_{i=1}^{n} (\mathbf{b_{2i}} \cup \mathbf{b'_{2i}}) \ .$$

The following immediate observations will prove useful:

1. $L(E'_1) = \{n \cdot \mathbf{1^m}\}$ and $L(E'_2) = \{0, \dots, 2n-1\}^m$.
2. $L(E_1) \subseteq \{n, \dots, 2n\}^m$ and $L(E_2) \supseteq \{n, \dots, 2n-1\}^m$.

Note that the only vectors of $L(E_1)$ which might perhaps not belong to $L(E_2)$ are the ones with at least one component of size $2n$. The following definitions take care of these "critical vectors". We say that a partial assignment of the variables in $X_1 \cup X_2$ *annuls* M_j if one of the literals contained in M_j is set to 0. Let $\mathbf{y} \in \{n, \dots, 2n\}^m$. An assignment $A_1 : X_1 \to \{0,1\}$ is said to be an X_1*-assignment of type* $\mathbf{y}$ if the following holds:

$$\forall j = 1, \dots, m : (\mathbf{y}[j] = 2n \Leftrightarrow A_1 \text{ does not annul } M_j) \ .$$

We say that $A_2 : X_2 \to \{0,1\}$ is an X_2*-assignment of type* $\mathbf{y}$ if the following holds:

$$\forall j = 1, \dots, m : (\mathbf{y}[j] = 2n \Rightarrow A_2 \text{ annuls } M_j) \ .$$

The desired equivalence (12) is easy to derive from the following claims:

Claim 1: For every $\mathbf{y} \in L(E_1)$, there exists an X_1-assignment A_1 of type $\mathbf{y}$.

Claim 2: For every $A_1 : X_1 \to \{0,1\}$, there exists $\mathbf{y} \in L(E_1)$ such that A_1 is an X_1-assignment of type $\mathbf{y}$.

Claim 3: For every $\mathbf{y} \in \{n, \dots, 2n\}^m$:

$$\mathbf{y} \in L(E_2) \Leftrightarrow (\exists A_2 : X_2 \to \{0,1\} : A_2 \text{ is an } X_2\text{-assignment of type } \mathbf{y}) \ .$$

Proof of Claim 1: Pick any $\mathbf{y} \in L(E_1)$. It follows that $\mathbf{y}$ is of the form

$$\mathbf{y} = n \cdot \mathbf{1^m} + \sum_{i=1}^{n} \widetilde{\mathbf{b_{1i}}} \ \text{ with } \ \widetilde{\mathbf{b_{1i}}} \in \{\overline{\mathbf{b_{1i}}}, \overline{\mathbf{b'_{1i}}}\} \ . \tag{13}$$

If $\widetilde{\mathbf{b_{1i}}} = \overline{\mathbf{b_{1i}}}$, we set $A_1(x_{1i}) = 0$ else, if $\widetilde{\mathbf{b_{1i}}} = \overline{\mathbf{b'_{1i}}}$, we set $A_1(x_{1i}) = 1$. We claim that A_1 is of type $\mathbf{y}$. This can be seen as follows. Pick any $j \in \{1, \dots, m\}$. An inspection of (13) reveals the following:

- Suppose that $\mathbf{y}[j] = 2n$. It follows that $\widetilde{\mathbf{b_{1i}}}[j] = 1$ for $i = 1, \dots, n$. Hence, if $\widetilde{\mathbf{b_{1i}}} = \overline{\mathbf{b_{1i}}}$, then $A_1(x_{1i}) = 0$, $\overline{\mathbf{b_{1i}}}[j] = 1$ and, therefore, $x_{1i} \notin M_j$. Similarly, if $\widetilde{\mathbf{b_{1i}}} = \overline{\mathbf{b'_{1i}}}$, then $A_1(x_{1i}) = 1$, $\overline{\mathbf{b'_{1i}}}[j] = 1$ and, therefore, $\overline{x_{1i}} \notin M_j$. Since these observations hold for all $i = 1, \dots, n$, we may conclude that A_1 does not annul M_j.

- Suppose that $\mathbf{y}[j] \leq 2n-1$. Then there exists $i \in \{1,\ldots,n\}$ such that $\widetilde{\mathbf{b_{1i}}}[j]=0$. Hence, if $\widetilde{\mathbf{b_{1i}}} = \overline{\mathbf{b_{1i}}}$, then $A_1(x_{1i})=0$, $\overline{\mathbf{b_{1i}}}[j]=0$ and, therefore, $x_{1i} \in M_j$. Similarly, if $\widetilde{\mathbf{b_{1i}}} = \overline{\mathbf{b'_{1i}}}$, then $A_1(x_{1i})=1$, $\overline{\mathbf{b'_{1i}}}[j]=0$ and, therefore, $\overline{x_{1i}} \in M_j$. It follows that A_1 does annul M_j.

The above discussion shows that A_1 is of type $\mathbf{y}$, indeed.

Proof of Claim 2: Given any $A_1 : X_1 \rightarrow \{0,1\}$, we set $\mathbf{y} = n \cdot \mathbf{1^m} + \sum_{i=1}^{n} \widetilde{\mathbf{b_{1i}}}$ where $\widetilde{\mathbf{b_{1i}}} = \overline{\mathbf{b_{1i}}}$ if $A_1(x_{1i})=0$ and, similarly, $\widetilde{\mathbf{b_{1i}}} = \overline{\mathbf{b'_{1i}}}$ if $A_1(x_{1i})=1$. Note that, with this definition of $\mathbf{y}$, A_1 is precisely the X_1-assignment that we had chosen in the proof of Claim 1. As argued in the proof of Claim 1 already, A_1 is of type $\mathbf{y}$.

Proof of Claim 3: Pick any $\mathbf{y} \in \{n,\ldots,2n\}^m$. Suppose first that $\mathbf{y} \in L(E_2)$. It follows that $\mathbf{y}$ is of the form

$$\mathbf{y} = \mathbf{y}' + \sum_{i=1}^{n} \widetilde{\mathbf{b_{2i}}} \ \text{ with } \ \mathbf{y}' \in \{0,\ldots,2n-1\}^m \ \text{ and } \ \widetilde{\mathbf{b_{2i}}} \in \{\mathbf{b_{2i}}, \mathbf{b'_{2i}}\} \ . \tag{14}$$

If $\widetilde{\mathbf{b_{2i}}} = \mathbf{b_{2i}}$, we set $A_2(x_{2i})=0$ else, if $\widetilde{\mathbf{b_{2i}}} = \mathbf{b'_{2i}}$, we set $A_2(x_{2i})=1$. We claim that A_2 is of type $\mathbf{y}$. Consider an index $j \in \{1,\ldots,m\}$ such that $\mathbf{y}[j]=2n$. An inspection of (14) reveals that there exists $i \in \{1,\ldots,n\}$ such that $\widetilde{\mathbf{b_{2i}}}[j]=1$. If $\widetilde{\mathbf{b_{2i}}} = \mathbf{b_{2i}}$, then $A_2(x_{2i})=0$, $\mathbf{b_{2i}}[j]=1$ and, therefore, $x_{2i} \in M_j$. Similarly, if $\widetilde{\mathbf{b_{2i}}} = \mathbf{b'_{2i}}$, then $A_2(x_{2i})=1$, $\mathbf{b'_{2i}}[j]=1$ and, therefore, $\overline{x_{2i}} \in M_j$. In any case, A_2 annuls M_j and we may conclude that A_2 is of type $\mathbf{y}$.

Suppose now that there exists an X_2-assignment A_2 that is of type $\mathbf{y} \in \{n,\ldots,2n\}^m$. We define $\mathbf{y}'' = \sum_{i=1}^{n} \widetilde{\mathbf{b_{2i}}}$ where $\widetilde{\mathbf{b_{2i}}} = \mathbf{b_{2i}}$ if $A_2(x_{2i})=0$ and, similarly, $\widetilde{\mathbf{b_{2i}}} = \mathbf{b'_{2i}}$ if $A_2(x_{2i})=1$. Since A_2 is of type $\mathbf{y}$, it annuls every M_j with $\mathbf{y}[j]=2n$. It follows that, for every $j \in \{1,\ldots,m\}$ with $\mathbf{y}[j]=2n$, there exists $i \in \{1,\ldots,n\}$ such either $x_{2i} \in M_j$ and $A_2(x_{2i})=0$ or $\overline{x_{2i}} \in M_j$ and $A_2(x_{2i})=1$. In both cases, we have that $\widetilde{\mathbf{b_{2i}}}[j]=1$. It follows from this discussion that $\mathbf{y}''[j] \geq 1$ for every j with $\mathbf{y}[j]=2n$. Obviously $\mathbf{y}''[j] \leq n$ for all $j=1,\ldots,m$. Since $L(E'_2) = \{0,\ldots,2n-1\}^m$ and $\mathbf{y} \in \{n,\ldots,2n\}^m$, there exists $\mathbf{y}' \in L(E_2)$ such that $\mathbf{y} = \mathbf{y}' + \mathbf{y}''$. This decomposition of $\mathbf{y}$ shows that $\mathbf{y} \in L(E_2)$.

We are ready now for proving (12). Assume first that the condition on the left hand-side of (12) is valid. Pick any $\mathbf{y} \in L(E_1)$. Pick an X_1-assignment A_1 of type $\mathbf{y}$ (application of Claim 1). It follows that the monomials M_j with $\mathbf{y}[j]=2n$ are not yet annulled by A_1. According to the left hand-side of (12), there must exist an assignment $A_2 : X_2 \rightarrow \{0,1\}$ that annuls them. In other words: A_2 is an X_2-assignment of type $\mathbf{y}$. We may now conclude from Claim 3 that $\mathbf{y} \in L(E_2)$, as desired.

Suppose now that $L(E_1) \subseteq L(E_2)$. Pick any assignment $A_1 : X_1 \rightarrow \{0,1\}$. Pick $\mathbf{y} \in L(E_1)$ such A_1 is an X_1-assignment of type $\mathbf{y}$ (application of Claim 2). It follows that only the monomials M_j with $\mathbf{y}[j]=2n$ are not yet annulled by A_1. Since $\mathbf{y}$, as an element of $L(E_1)$, must satisfy $\mathbf{y} \in \{n,\ldots,2n\}^m$ and must furthermore belong to $L(E_2)$, we may conclude from Claim 3 that there exists an X_2-assignment $A_2 : X_2 \rightarrow \{0,1\}$ of type $\mathbf{y}$. In other words: A_2 annuls all monomials M_j with $\mathbf{y}[j]=2n$. It follows from this discussion that the condition on the left hand-side of (12) is valid, which concludes the proof.

5 Open Problems

In the proof of our hardness results, we made essential use of the fact that $\langle P \rangle$ contains all linear combinations of the periods in P with coefficient vectors from $\mathbb{N}_0^{|P|}$. We would be interested to know whether the computational complexity of the containment problem is still the same when we deal with coefficient vectors from $\mathbb{N}^{|P|}$ (thereby ruling out 0-coefficients).

References

1 Alfred Brauer. On a problem of partitions. *American Journal of Mathematics*, 64(1):299–312, 1942.

2 Dmitry Chistikov, Christoph Haase, and Simon Halfon. Context-free commutative grammars with integer counters and resets. *Theoretical Computer Science*, 2016. In press. Online version: https://doi.org/10.1016/j.tcs.2016.06.017.

3 F. Glover and R. E. D. Woolsey. Aggregating diophantine equations. *Zeitschrift für Operations Research*, 16:1–10, 1972.

4 Thiet-Dung Huynh. The complexity of semilinear sets. *Elektronische Informationsverarbeitung und Kybernetik*, 18(6):291–338, 1982.

5 Rohit J. Parikh. On context-free languages. *Journal of the Association on Computing Machinery*, 13(4):570–581, 1966.

6 José C. Rosales and Pedro A. García-Sánchez. *Numerical Semigroups.* Springer, 2009.

7 Larry J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1977.

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