Equations over sets of natural numbers with addition only

Artur Jeż[a,1], Alexander Okhotin[b]

[a] *Institute of Computer Science, University of Wrocław, Poland*
[b] *Department of Mathematics, University of Turku, Finland; Academy of Finland*

Abstract

Systems of equations of the form $X = Y + Z$ and $X = C$ are considered, in which the unknowns are sets of integers, "+" denotes pairwise sum of sets $S + T = \{m + n \mid m \in S,\ n \in T\}$, and C is an ultimately periodic constant. When restricted to sets of natural numbers, such equations can be equally seen as language equations over a one-letter alphabet with concatenation and regular constants, and it is shown that such systems are computationally universal, in the sense that for every recursive (r.e., co-r.e.) set $S \subseteq \mathbb{N}$ there exists a system with a unique (least, greatest) solution containing a component T with $S = \{n | 16n+13 \in T\}$. This implies undecidability of basic properties of these equations: solution existence is Π^0_1-complete, solution uniqueness is Π^0_2-complete, and finiteness of the set of solutions is Σ^0_3-complete. For systems over sets of all integers, both positive and negative, there is a similar construction of a system with a unique solution $S = \{n \mid 16n \in T\}$ representing any hyper-arithmetical set $S \subseteq \mathbb{N}$. Testing solution existence for such systems is Σ^1_1-complete.

Keywords: Language equations, unary languages, concatenation, computability

1. Introduction

Language equations are equations of the form $\varphi(X_1, \ldots, X_n) = \psi(X_1, \ldots, X_n)$, in which the unknowns X_i are formal languages, while the expressions φ, ψ use language-theoretic operations, such as concatenation, Kleene star and Boolean operations, as well as constant languages. It is well-known that systems of the *resolved form*

$$\begin{cases} X_1 &= \varphi_1(X_1, \ldots, X_n) \\ & \vdots \\ X_n &= \varphi_n(X_1, \ldots, X_n) \end{cases} \tag{*}$$

with union, concatenation and singleton constants define the semantics of the context-free grammars [2]. If intersection is also allowed, such equations characterize an extension of the context-free grammars known as *conjunctive grammars* [13], which have an increased expressive power and are at the same time notable for preserving efficient parsing algorithms [14, 18].

The expressive power of language equations of the general form

$$\begin{cases} \varphi_1(X_1, \ldots, X_n) &= \psi_1(X_1, \ldots, X_n) \\ & \vdots \\ \varphi_1(X_1, \ldots, X_n) &= \psi_1(X_1, \ldots, X_n) \end{cases} \tag{**}$$

was determined by Okhotin [17, 16], who proved that a language is representable by a unique solution of a system with concatenation, Boolean operations and singleton constants if and only if this language is

☆This work has been presented at the STACS 2009 conference held in Freiburg, Germany on February 26–28, 2009. Research supported by MNiSW grant number N N206 49/2638, 2010–2012, and by the Academy of Finland under grants 118540 and 134860.

recursive. Further characterisation of recursively enumerable (r.e.) and co-r.e. sets was given in terms of *least* and *greatest* (with respect to component-wise inclusion) solutions of such systems [17]. The same expressive power is attained using concatenation with constants and union [16]. It was subsequently discovered that language equations can be computationally universal even without any Boolean operations: Kunc [9] constructed a finite language $L \subseteq \{a, b\}^*$, for which the greatest solution of a language equation $LX = XL$ is Π_1^0-hard (that is, hard for co-r.e. sets). This paper establishes a similar result in the seemingly trivial case of a one-letter alphabet.

Unary languages, defined over an alphabet $\{a\}$, form an important special class of formal languages. It is well-known that context-free grammars over this alphabet generate only regular languages [2]. The first example of a language equation over a unary alphabet with a non-regular unique solution was constructed by Leiss [11]: this was an equation $X = \varphi(X)$ with φ containing concatenation, complementation and constant $\{a\}$. The question of whether conjunctive grammars (in other words, systems of language equations with union, intersection and concatenation) can generate any non-regular languages had been a long-standing open problem [13], until Jeż [3] constructed a conjunctive grammar generating $\{a^{4^n} \mid n \geqslant 0\}$. The ideas of this example were used by Jeż and Okhotin [4] to establish some general results on the expressive power of these equations, as well as the EXPTIME-completeness of their solutions [5]. For systems of the general form (**) using concatenation and union, it has recently been shown by the authors [6] that they are computationally complete; on the higher level, this result can be considered a remake of the proof of the computational completeness of language equations [17], but encoding that proof using only sets of numbers required much more difficult constructions.

As unary languages can be regarded as sets of natural numbers, unary language equations are naturally viewed as *equations over sets of numbers.* Concatenation of languages accordingly turns into *addition* of sets

$$S + T = \{m + n \mid m \in S,\ n \in T\},$$

an operation that has been a subject of much study in number theory and combinatorics [23]. Computational complexity of expressions and circuits over sets of natural numbers with addition and different sets of Boolean operations has first been investigated by Stockmeyer and Meyer [22] and then extensively studied by McKenzie and Wagner [12]. A similar study for expressions and circuits over sets of both positive and negative integers was done by Travers [24].

As compared to these circuits, equations over sets of numbers are a more general formalism, which can express circular dependencies. The expressive power of equations over sets of natural numbers with addition and Boolean operations was determined in the aforementioned work on language equations over a unary alphabet [3, 4, 5, 6]: they are computationally complete. Equations over sets of integers were recently investigated by the authors [8], and proved to define exactly the *hyper-arithmetical sets*, which is a class situated at the bottom of the analytical hierarchy and properly containing the sets representable in first-order Peano arithmetic.

This paper is concerned with equations over sets of numbers that use *only addition and no Boolean operations*, both in the case of sets of natural numbers and sets of integers as unknowns. The first to be considered is the case of natural numbers and systems of equations of the form

$$X_{i_1} + \ldots + X_{i_k} + C = X_{j_1} + \ldots + X_{j_\ell} + D$$

in variables $(X_1, \ldots, X_n)$, where $C, D \subseteq \mathbb{N}$ are ultimately periodic constants. In terms of language equations over $\{a\}$, these are equations

$$X_{i_1} \ldots X_{i_k} K = X_{j_1} \ldots X_{j_\ell} L,$$

with regular constants $K, L \subseteq a^*$. This is the ultimately simplest case of language equations, and at the first glance it seems out of question that such equations could have any non-trivial unique solutions (and considering least or greatest solutions makes little difference). Probably for that reason no one has ever proclaimed their expressive power to be an open problem. However, as proved in this paper, these equations can have not only non-periodic unique solutions, but in fact are computationally universal. Furthermore, their main decision problems are as hard as similar problems for language equations over multiple-letter alphabets and using all Boolean operations [17, 16].

The new results are directly based on the authors' recent proof of the computational completeness of equations over sets of numbers with addition and union [6], though it is established using completely different methods. The idea is to take an arbitrary system using addition and union and *encode* it in another system using addition only. The solutions of the two systems will not be identical, but there will be a bijection between solutions based upon an encoding of sets of numbers.

This encoding of sets, defined in Section 3, is an injection $\sigma : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$, which represents every number n of the encoded set as the number $16n+13$ in the encoding. The given encoding has two key properties. First of all, its form can be *checked* by an equation, which is satisfied exactly by those sets of natural numbers that are valid encodings of some sets; such an equation is constructed in Section 3. Second, the sum of any two valid encodings $\sigma(S)$ and $\sigma(T)$ encodes *both the sum and the union* of the encoded sets of numbers S and T, and furthermore, adding a certain constant to such a sum of encodings produces a set that encodes only the sum $S + T$ of the original sets, while adding another constant allows representing only their union $S \cup T$. In overall, as shown in Section 4, the sum and the union of any two sets is represented by their encodings.

Finally, on the basis of this encoding, in Section 5 it is demonstrated how an arbitrary system of equations over sets of natural numbers with union and addition can be simulated using addition only. Each variable X_i of the original system will be represented in the new system by a variable X'_i, and the solutions of the new system will be of the form $X'_i = \sigma(S_i)$ for all variables X'_i, where $X_i = S_i$ is a solution of the original system.

The paper continues with a similar investigation of *equations over sets of integers* in Section 6. The method described above is applied to encode every system with union and addition into a system with addition only. The encoding σ is redefined as $\sigma : \mathbb{Z} \to \mathbb{Z}$, with the same property that a number $n \in \mathbb{Z}$ is in S if and only if $16n + 13 \in \sigma(S)$. This leads to representing a set $S \subseteq \mathbb{Z}$ by the set $\sigma(S)$. The general structure of the encoding and the associated equations are preserved and its correctness is established by a similar argument.

All constants in both constructions are ultimately periodic; some of them are finite and some are infinite. The last question is whether infinite constants are necessary to specify any non-periodic sets, and an affirmative answer is given in Section 7.

2. Equations over sets of natural numbers

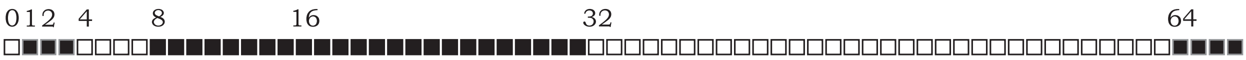
Throughout this paper, the set of natural numbers $\mathbb{N} = \{0, 1, 2, \dots\}$ is assumed to contain zero. A set of numbers $S \subseteq \mathbb{N}$ is *ultimately periodic* if there exist numbers $d \geqslant 0$ and $p \geqslant 1$, such that $n \in S$ if and only if $n + p \in S$ for every $n \geqslant d$. Otherwise, S is *non-periodic*. Note that S is ultimately periodic if and only if the corresponding language $L = \{a^n \mid n \in S\} \subseteq a^*$ is regular.

For every two subsets of natural numbers $S, T \subseteq \mathbb{N}$, their *sum* is the set $\{m + n \mid m \in S,\ n \in T\}$. Other typical operations on sets are the Boolean operations, such as union, intersection and complementation. Using complementation and addition, the first example of an equation with a non-periodic unique solution was constructed:

Example 1 (Leiss [11])**.** *For every expression φ, denote $2\varphi = \varphi + \varphi$. Then the unique solution of the equation*

$$X = 2\Big(\overline{2\big(\overline{2\overline{X}}\big)}\Big) + \{1\}$$

is $\{n \mid \exists i \geqslant 0 :\ 2^{3i} \leqslant n < 2^{3i+2}\} = \{n \mid$ base-8 notation of n begins with `1`*,* `2` *or* `3`*$\}$.*



Expressive power of this family of equations is still quite limited [19, 20], with some simple languages being non-representable.

The second example of a non-periodic solution of equations over sets of natural numbers was constructed by Jeż [3] as a conjunctive grammar [13] generating the language $\{a^{4^n} \mid n \geqslant 0\}$. In terms of equations it is stated as follows:

Example 2 (Jeż [3]). *The least solution of the system*

$$\begin{cases} X_1 &= \big((X_1{+}X_3) \cap (X_2{+}X_2)\big) \cup \{1\} \\ X_2 &= \big((X_1{+}X_1) \cap (X_2{+}X_6)\big) \cup \{2\} \\ X_3 &= \big((X_1{+}X_2) \cap (X_6{+}X_6)\big) \cup \{3\} \\ X_6 &= (X_1{+}X_2) \cap (X_3{+}X_3) \end{cases}$$

is $X_1 = \{4^n \mid n \geqslant 0\}$, $X_2 = \{2 \cdot 4^n \mid n \geqslant 0\}$, $X_3 = \{3 \cdot 4^n \mid n \geqslant 0\}$, $X_6 = \{6 \cdot 4^n \mid n \geqslant 0\}$. *The system has other solutions as well, but the given one is the least with respect to component-wise inclusion.*

0 16 32 48 64
□■□□■□□□□□□□□□□□■□□□■□□□□

The idea behind this example is to manipulate positional notations of natural numbers, and this idea was subsequently used to establish the following general result on the expressive power of such equations. The statement refers to the family of *linear conjunctive languages* [13], which is known to be equivalent to one-way real-time cellular automata [15], and which properly contains the Boolean closure of linear context-free languages.

Proposition 1 (Jeż, Okhotin [4]). *For every* $k \geqslant 2$ *and for every linear conjunctive language* $L \subseteq \{0, 1, \ldots, k-1\}^+$ *there exists a resolved system of equations*

$$\begin{cases} X_1 &= \varphi_1(X_1, \ldots, X_n) \\ &\vdots \\ X_n &= \varphi_n(X_1, \ldots, X_n), \end{cases}$$

with φ_i *using singleton constants and the operations of union, intersection and addition, which has a unique solution with* $X_1 = \{n \mid$ *the base-*k *notation of* n *is in* $L\}$.

On the basis of this result, it was shown that systems of equations of the general form $\varphi = \psi$ with the same operations are computationally complete, and it is sufficient to use only one of the two Boolean operations to attain computational completeness. Recursive sets are represented by unique solutions, while using least and greatest solutions (with respect to component-wise inclusion) allows representing recursively enumerable (r.e.) and co-recursively enumerable (co-r.e.) sets as their components.

Theorem 1 (Jeż, Okhotin [6]). *For every recursive (r.e., co-r.e.) set* $S \subseteq \mathbb{N}$ *there exists an unresolved system*

$$\begin{cases} \varphi_1(X_1, \ldots, X_n) &= \psi_1(X_1, \ldots, X_n) \\ &\vdots \\ \varphi_m(X_1, \ldots, X_n) &= \psi_m(X_1, \ldots, X_n), \end{cases}$$

with φ_j, ψ_j *using singleton constant* $\{1\}$ *and the operations of union and addition, which has a unique (least, greatest, respectively) solution with* $X_1 = S$.

Exactly the same results hold for unresolved systems with intersection, addition and singleton constants [6], though they will not be used in this paper. A matching upper bound on the complexity of solutions is known from the more general case of language equations [17].

The goal is now to take any system of equations with union and addition, such as those constructed in Theorem 1, and to simulate it by another system using addition only. The solutions of the new system will *encode* the solutions of the original system as described in the next section.

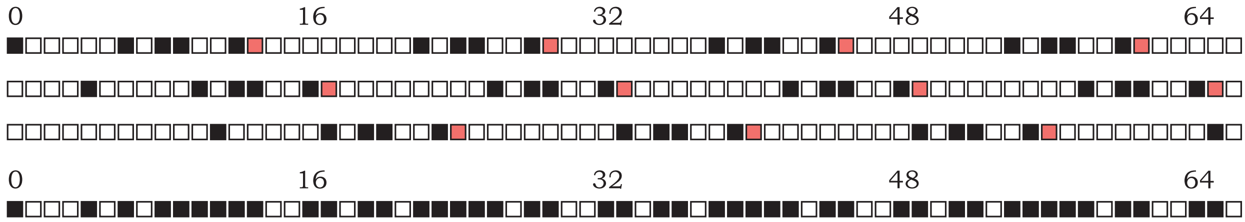


Figure 1: The addition of $\sigma(T) + \{0, 4, 11\}$. The following rows represent $\sigma(T)$, $\sigma(T) + \{4\}$, $\sigma(T) + \{11\}$ and finally $\sigma(T) + \{0, 4, 11\}$. It can be seen that the last sum has only black and white cells.

3. Encoding of sets

The constructed system with addition operates with sets of a specific form. These sets have a certain fixed periodic structure, which yields predictable results when adding sets to each other. Inscribed within this structure is a single set of natural numbers, and sums of such encoded sets effectively produce both union and sums of the encoded sets. Let $\sigma : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ be a function mapping a set $S \subseteq \mathbb{N}$ to its encoding $\sigma(S)$.

An arbitrary set of numbers $T \subseteq \mathbb{N}$ will be represented by another set $S \subseteq \mathbb{N}$, which contains a number $16n + 13$ if and only if n is in T. The membership of numbers i with $i \not\equiv 13 \pmod{16}$ in S does not depend on T and will be defined below. Since many constructions in the following will be done modulo 16, the following notation shall be adopted:

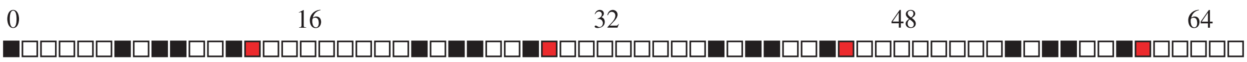
Definition 1. *Let $S \subseteq \mathbb{N}$. For each $i \in \{0, 1, \dots, 15\}$, $\textsc{track}_i(S) = \{n \mid 16n + i \in S\}$ is the set on the i-th* track *of S. Moreover, for each set $T \subseteq \mathbb{N}$, let $\tau_i(T) = \{16n + i \mid n \in T\}$ denote the set with T on the i-th track and the rest of the tracks empty.*

A set S is said to have an empty track *i if $\textsc{track}_i(S) = \varnothing$, and a* full track *$i$ if $\textsc{track}_i(S) = \mathbb{N}$.*

In these terms, it can be said that a set T shall be encoded in the 13-th track of a set S. The rest of the tracks of S contain technical information needed for the below constructions to work: track 0 contains a singleton $\{0\}$, tracks 6, 8, 9 and 12 are full and the rest of the tracks are empty.

Definition 2. *For every set $T \subseteq \mathbb{N}$, its* encoding *is the set*

$$S = \sigma(T) = \{0\} \cup \tau_6(\mathbb{N}) \cup \tau_8(\mathbb{N}) \cup \tau_9(\mathbb{N}) \cup \tau_{12}(\mathbb{N}) \cup \tau_{13}(T).$$



The first property of the encoding announced in the introduction is that there exists an equation with the set of all valid encodings as its set of solutions. Such an equation will now be constructed.

Lemma 1. *A set $S \subseteq \mathbb{N}$ satisfies an equation*

$$S + \{0, 4, 11\} = \bigcup_{\substack{i \in \{0,4,6,8,9,\\ 10,12,13\}}} \tau_i(\mathbb{N}) \cup \bigcup_{i \in \{1,3,7\}} \tau_i(\mathbb{N} + 1) \cup \{11\}$$

if and only if $S = \sigma(T)$ for some $T \subseteq \mathbb{N}$.

Proof. $\Leftarrow$ Let S be any set that satisfies the equation. Then the sum $S + \{0, 4, 11\}$ has empty tracks 2, 5, 14 and 15:

$$\textsc{track}_2(S+\{0, 4, 11\}) = \textsc{track}_5(S+\{0, 4, 11\}) = \textsc{track}_{14}(S+\{0, 4, 11\}) = \textsc{track}_{15}(S+\{0, 4, 11\}) = \varnothing$$

For this condition to hold, S must have many empty tracks as well. To be precise, each track t with any of t, $t+4$ or $t+11 \pmod{16}$ being in $\{2,5,14,15\}$ must be an empty track in S. Calculating such set of tracks, $\{2,5,14,15\} - \{0,4,11\} \pmod{16} = \{1,2,3,4,5,7,10,11,14,15\}$ are the numbers of tracks that must be empty in S.

Similar considerations apply to track 11, as $\textsc{track}_{11}(S+\{0,4,11\}) = \{0\}$. For every track t with $t = 11$, $t+4 = 11$ or $t+11 = 11 \pmod{16}$, the t-th track of S must either be an empty track or contain singleton zero: $\textsc{track}_t(S) = \{0\}$. The latter must hold for at least one such t. Let us calculate all such tracks t: these are tracks with numbers $\{11\} - \{0,4,11\} \pmod{16} = \{0,7,11\}$. Since tracks number 7 and 11 are already known to be empty, it follows that $\textsc{track}_0(S) = \{0\}$.

In order to prove that S is a valid encoding of some set, it remains to show that tracks number $6,8,9,12$ in S are full. Consider first that $\textsc{track}_3(S+\{0,4,11\}) = \mathbb{N}+1$. Let us calculate the track numbers t, for which there exists $t' \in \{0,4,11\}$ with $(t+t') \pmod{16} = 3$: these are $\{3\} - \{0,4,11\} \pmod{16} = \{3,8,15\}$. Since tracks $3,15$ are known to be empty

$$\begin{aligned}\mathbb{N}+1 = \textsc{track}_3(S+\{0,4,11\}) = \textsc{track}_3(S) \cup (\textsc{track}_{15}(S)+1) \cup (\textsc{track}_8(S)+1) = \\ = \varnothing \cup \varnothing \cup (\textsc{track}_8(S)+1) = \textsc{track}_8(S)+1,\end{aligned}$$

and thus track 8 of S is full. The analogous argument is used to prove that tracks $12,9,6$ are full. Consider $\textsc{track}_7(S+\{0,4,11\}) = \mathbb{N}+1$. Then $\{7\} - \{0,4,11\} \pmod{16} = \{7,3,12\}$. Since it is already known that tracks 3 and 7 are empty, the track 12 is full:

$$\begin{aligned}\mathbb{N}+1 = \textsc{track}_7(S+\{0,4,11\}) = \textsc{track}_7(S) \cup \textsc{track}_3(S) \cup (\textsc{track}_{12}(S)+1) = \\ = \varnothing \cup \varnothing \cup (\textsc{track}_{12}(S)+1) = \textsc{track}_{12}(S)+1.\end{aligned}$$

In the same way consider $\textsc{track}_9(S+\{0,4,11\}) = \mathbb{N}$. Then $\{9\} - \{0,4,11\} \pmod{16} = \{9,5,14\}$ and tracks $5,14$ are empty, thus track 9 is full:

$$\begin{aligned}\mathbb{N} = \textsc{track}_9(S+\{0,4,11\}) = \textsc{track}_9(S) \cup \textsc{track}_5(S) \cup (\textsc{track}_{14}(S)+1) = \\ = \textsc{track}_9(S) \cup \varnothing \cup \varnothing = \textsc{track}_9(S).\end{aligned}$$

Now let us inspect $\textsc{track}_{10}(S+\{0,4,11\})$. Then $\{10\} - \{0,4,11\} \pmod{16} = \{10,6,15\}$. Since the tracks $10,15$ are empty, the 6-th track is full:

$$\begin{aligned}\mathbb{N} = \textsc{track}_{10}(S+\{0,4,11\}) = \textsc{track}_{10}(S) \cup \textsc{track}_6(S) \cup (\textsc{track}_{15}(S)+1) = \\ = \varnothing \cup \textsc{track}_6(S) \cup \varnothing = \textsc{track}_6(S).\end{aligned}$$

Thus it has been proved that $S = \sigma(T)$ for $T = \textsc{track}_{13}(S)$.

$\Leftarrow$ It remains to show the converse, that is, that if $S = \sigma(T)$ for some $T \subseteq \mathbb{N}$, then

$$S+\{0,4,11\} = \bigcup_{\substack{i\in\{0,4,6,8,9,\\10,12,13\}}} \tau_i(\mathbb{N}) \cup \bigcup_{i\in\{1,3,7\}} \tau_i(\mathbb{N}+1) \cup \{11\}.$$

Consider the sum $\sigma(T)+\{0,4,11\} = (\sigma(T)+\{0\}) \cup (\sigma(T)+\{4\}) \cup (\sigma(T)+\{11\})$, where each term in the union is split into tracks, as illustrated in Table 1. Since $S = \bigcup_{i=0}^{15} \tau_i(\textsc{track}_i(S))$,

$$\sigma T+\{0,4,11\} = \Big(\bigcup_i \tau_i(\textsc{track}_i(\sigma T))+0\Big) \cup \Big(\bigcup_i \tau_i(\textsc{track}_i(\sigma T))+4\Big) \cup \Big(\bigcup_i \tau_i(\textsc{track}_i(\sigma T))+11\Big),$$

and Table 1 presents the form of each term in this union. Each column represents the tracks of a particular set, that is $\sigma(T)$, $\sigma(T)+0$, $\sigma(T)+4$, $\sigma(T)+11$ and finally $\sigma(T)+\{0,4,11\}$. The i-th row gives the set encoded on the i-th track of the set represented in this column. Whenever a given track is empty, the respective cell in the table is empty as well.

tracks	$\sigma(T)$	$\sigma(T)+\{0\}$	$\sigma(T)+\{4\}$	$\sigma(T)+\{11\}$	$\sigma(T)+\{0,4,11\}$
0	$\{0\}$	$\{0\}$	$\mathbb{N}$		
1			$T+1$	$\mathbb{N}+1$	$\mathbb{N}+1$
2					
3			$\mathbb{N}+1$		$\mathbb{N}$
4			$\{0\}$	$\mathbb{N}+1$	$\mathbb{N}$
5					
6	$\mathbb{N}$	$\mathbb{N}$			$\mathbb{N}$
7				$\mathbb{N}+1$	$\mathbb{N}+1$
8	$\mathbb{N}$	$\mathbb{N}$		$T+1$	$\mathbb{N}$
9	$\mathbb{N}$	$\mathbb{N}$			$\mathbb{N}$
10			$\mathbb{N}$		$\mathbb{N}$
11				$\{0\}$	$\{0\}$
12	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$		$\mathbb{N}$
13	T	T	$\mathbb{N}$		$\mathbb{N}$
14					
15					

Table 1: Tracks in the sum $\sigma(T)+\{0,4,11\}$. Empty cells represent empty tracks.

According to the table, the values of the set T are reflected in three tracks of the sum $\sigma(T)+\{0,4,11\}$: in tracks 13, 1 and 8 (in the last two cases, with offset 1). However, at the same time the sum contains full tracks 8 and 13, as well as $\mathbb{N}+1$ in track 1, and the contributions of T to the sum are subsumed by these numbers, as $\tau_{13}(T) \subseteq \tau_{13}(\mathbb{N})$, $\tau_1(T+1) \subseteq \tau_1(\mathbb{N}+1)$ and $\tau_8(T+1) \subseteq \tau_8(\mathbb{N})$. Therefore, the value of the expression does not depend on T. Taking the union of all entries of Table 1 proves that $\sigma(T)+\{0,4,11\}$ equals

$$\bigcup_{\substack{i\in\{0,4,6,8,9,\\10,12,13\}}} \tau_i(\mathbb{N}) \cup \bigcup_{i\in\{1,3,7\}} \tau_i(\mathbb{N}+1) \cup \{11\},$$

as stated in the lemma. $\square$

4. Simulating operations

The goal of this section is to establish the second property of the encoding σ, that is, that a sum of encodings of two sets and a fixed constant set effectively encodes the union of these two sets, while the addition of a different fixed constant set allows encoding the sum of the two original sets. This property is formally stated in the following lemma, along with the actual constant sets:

Lemma 2. *For all sets $X, Y, Z \subseteq \mathbb{N}$,*

$$\sigma(Y)+\sigma(Z)+\{0,1\} = \sigma(X)+\sigma(\{0\})+\{0,1\} \quad \textit{if and only if} \quad Y+Z=X$$

and

$$\sigma(Y)+\sigma(Z)+\{0,2\} = \sigma(X)+\sigma(X)+\{0,2\} \quad \textit{if and only if} \quad Y\cup Z=X.$$

Proof. The goal is to show that for all $Y, Z \subseteq \mathbb{N}$, the sum

$$\sigma(Y)+\sigma(Z)+\{0,1\}$$

encodes the set $Y+Z+1$ on one of its tracks, while the contents of all other tracks do not depend on Y or on Z. Similarly, the sum

$$\sigma(Y)+\sigma(Z)+\{0,2\}$$

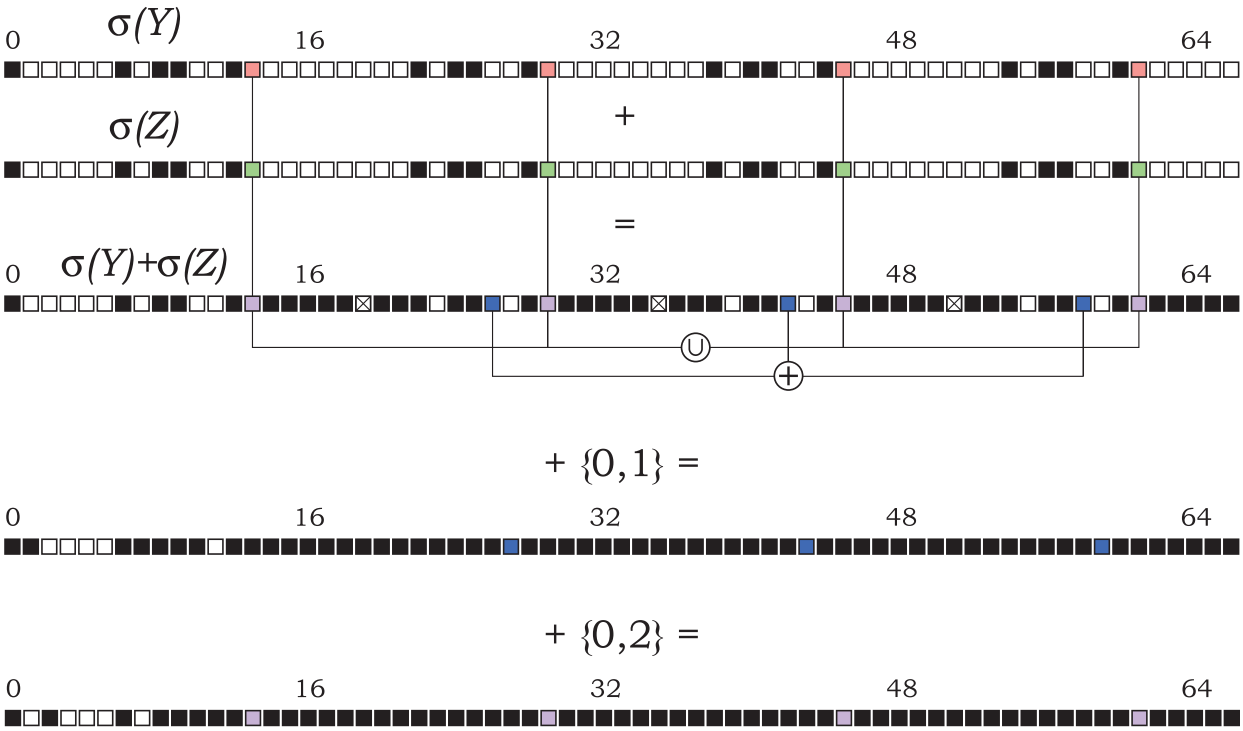


Figure 2: The sum and union tracks in $\sigma(Y)+\sigma(Z)$. Isolating them by adding $\{0,1\}$ and $\{0,2\}$.

has a track that encodes $Y \cup Z$, while the rest of its tracks also do not depend on Y and Z.

The common part of both of the above sums is $\sigma(Y)+\sigma(Z)$, so let us calculate it first. Since

$$\sigma(Y) = \{0\} \cup \tau_6(\mathbb{N}) \cup \tau_8(\mathbb{N}) \cup \tau_9(\mathbb{N}) \cup \tau_{12}(\mathbb{N}) \cup \tau_{13}(Y) \quad \text{and}$$
$$\sigma(Z) = \{0\} \cup \tau_6(\mathbb{N}) \cup \tau_8(\mathbb{N}) \cup \tau_9(\mathbb{N}) \cup \tau_{12}(\mathbb{N}) \cup \tau_{13}(Z),$$

by the distributivity of union over addition, the sum $\sigma(Y)+\sigma(Z)$ is a union of 36 nonempty terms, each being a sum of two individual tracks. Every such sum is contained in a single track as well, and Table 2 gives a case inspection of the form of all these terms. Each of its six rows corresponds to one of the nonempty tracks of $\sigma(Y)$, while its six columns refer to the nonempty tracks in $\sigma(Z)$. Then the cell gives the sum of these tracks, in the form of the track number and track contents: that is, for a row representing $\textsc{track}_i(\sigma(Y))$ and for a column representing $\textsc{track}_j(\sigma(Z))$, the cell (i,j) represents the set $\textsc{track}_i(\sigma(Y)) + \textsc{track}_j(\sigma(Z))$, which is bound to be on the track $i+j \pmod{16}$. For example, the sum of track 8 of $\sigma(Y)$ and track 9 of $\sigma(Z)$ falls onto track $1 = 8+9 \pmod{16}$ and equals

$$\tau_8(\mathbb{N}) + \tau_9(\mathbb{N}) = \{8+9+16(m+n) \mid m,n \geqslant 0\} = \{1+16n \mid n \geqslant 1\} = \tau_1(\mathbb{N}+1),$$

while adding track 13 of $\sigma(Y)$ to track 13 of $\sigma(Z)$ results in

$$\tau_{13}(Y) + \tau_{13}(Z) = \{26+16(m+n) \mid m \in Y,\ n \in Z\} = \tau_{10}(Y+Z+1),$$

which is reflected in the table. Each question mark denotes a track with unspecified contents. Though these contents can be calculated, they are actually irrelevant, because they do not influence the value of the subsequent sums $\sigma(Y)+\sigma(Z)+\{0,1\}$ and $\sigma(Y)+\sigma(Z)+\{0,2\}$. What is important is that none of these tracks contains 0.

The value of each i-th track of $\sigma(Y)+\sigma(Z)$ is obtained as the union of all sums in Table 2 that belong to the i-th track. The final values of these tracks are presented in the corresponding column of Table 3.

	$0\colon \{0\}$	$6\colon \mathbb{N}$	$8\colon \mathbb{N}$	$9\colon \mathbb{N}$	$12\colon \mathbb{N}$	$13\colon Z$
$0\colon \{0\}$	$0\colon \{0\}$	$6\colon \mathbb{N}$	$8\colon \mathbb{N}$	$9\colon \mathbb{N}$	$12\colon \mathbb{N}$	$13\colon Z$
$6\colon \mathbb{N}$	$6\colon \mathbb{N}$	$12\colon \mathbb{N}$	$14\colon \mathbb{N}$	$15\colon \mathbb{N}$	$2\colon \mathbb{N}+1$	$3\colon ?$
$8\colon \mathbb{N}$	$8\colon \mathbb{N}$	$14\colon \mathbb{N}$	$0\colon \mathbb{N}+1$	$1\colon \mathbb{N}+1$	$4\colon \mathbb{N}+1$	$5\colon ?$
$9\colon \mathbb{N}$	$9\colon \mathbb{N}$	$15\colon \mathbb{N}$	$1\colon \mathbb{N}+1$	$2\colon \mathbb{N}+1$	$5\colon \mathbb{N}+1$	$6\colon ?$
$12\colon \mathbb{N}$	$12\colon \mathbb{N}$	$2\colon \mathbb{N}+1$	$4\colon \mathbb{N}+1$	$5\colon \mathbb{N}+1$	$8\colon \mathbb{N}+1$	$9\colon ?$
$13\colon Y$	$13\colon Y$	$3\colon ?$	$5\colon ?$	$6\colon ?$	$9\colon ?$	$10\colon Y+Z+1$

Table 2: Tracks in the sum $\sigma(Y)+\sigma(Z)$. Question marks denote subsets of $\mathbb{N}+1$ that depend on Y or Z and whose actual values are unimportant.

	$\sigma(Y)$	$\sigma(Z)$	$\sigma(Y)+\sigma(Z)$	$\sigma(Y)+\sigma(Z)+\{0,1\}$	$\sigma(Y)+\sigma(Z)+\{0,2\}$
0	$\{0\}$	$\{0\}$	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$
1			$\mathbb{N}+1$	$\mathbb{N}$	$\mathbb{N}+1$
2			$\mathbb{N}+1$	$\mathbb{N}+1$	$\mathbb{N}$
3			$?$	$\mathbb{N}+1$	$\mathbb{N}+1$
4			$\mathbb{N}+1$	$\mathbb{N}+1$	$\mathbb{N}+1$
5			$\mathbb{N}+1$	$\mathbb{N}+1$	$\mathbb{N}+1$
6	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$
7				$\mathbb{N}$	$\mathbb{N}+1$
8	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$
9	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$
10			$Y+Z+1$	$\mathbb{N}$	$\mathbb{N}$
11				$Y+Z+1$	$\mathbb{N}$
12	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$
13	Y	Z	$Y\cup Z$	$\mathbb{N}$	$Y\cup Z$
14			$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$
15			$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$

Table 3: Tracks in the sums of $\sigma(Y)+\sigma(Z)$ with constants. Empty cells represent empty tracks.

Now the contents of the tracks in $\sigma(Y) + \sigma(Z) + \{0,1\}$ can be completely described. The calculations are given in Table 3, and the result is that, for all Y and Z,

$$\begin{aligned} \textsc{track}_{11}(\sigma(Y) + \sigma(Z) + \{0,1\}) &= Y + Z + 1, && \\ \textsc{track}_i(\sigma(Y) + \sigma(Z) + \{0,1\}) &= \mathbb{N} + 1, && \text{for } i \in \{2,3,4,5\}, \\ \textsc{track}_i(\sigma(Y) + \sigma(Z) + \{0,1\}) &= \mathbb{N}, && \text{for all other } i. \end{aligned}$$

It easily follows that

$$X = Y + Z$$

if and only if

$$\sigma(X) + \sigma(\{0\}) + \{0,1\} = \sigma(Y) + \sigma(Z) + \{0,1\},$$

as $X = X + \{0\}$.

For the set $\sigma(Y) + \sigma(Z) + \{0,2\}$, in the same way, for all Y and Z,

$$\begin{aligned} \textsc{track}_{13}(\sigma(Y) + \sigma(Z) + \{0,2\}) &= Y \cup Z, && \\ \textsc{track}_j(\sigma(Y) + \sigma(Z) + \{0,2\}) &= \mathbb{N} + 1, && \text{for } j \in \{1,3,4,5,7\}, \\ \textsc{track}_j(\sigma(Y) + \sigma(Z) + \{0,2\}) &= \mathbb{N}, && \text{for all other } j, \end{aligned}$$

and therefore, for all X, Y, Z,

$$X = Y \cup Z$$

if and only if

$$\sigma(X) + \sigma(X) + \{0,2\} = \sigma(Y) + \sigma(Z) + \{0,2\},$$

since $X = X \cup X$.

Both claims of the lemma follow. $\square$

5. Simulating a system over sets of natural numbers

Using the encoding defined above, it is now possible to represent a system with union and addition by a system with addition only. Since Lemma 2 on the simulation of individual operations is applicable only to equations of a simple form, the first task is to convert a given system to such a form:

Lemma 3. *For every system of equations over sets of natural numbers in variables $(X_1, \ldots, X_n)$ using union, addition and constants from a class $\mathcal{C}$ there exists a system in variables $(X_1, \ldots, X_n, X_{n+1}, \ldots, X_{n+m})$ with all equations of the form $X_i = X_j + X_k$, $X_i = X_j \cup X_k$ or $X_i = C$ with $C \in \mathcal{C}$, such that the set of solutions of this system is*

$$\big\{(S_1, \ldots, S_n, f_1(S_1, \ldots, S_n), \ldots, f_m(S_1, \ldots, S_n)) \mid (S_1, \ldots, S_n) \text{ is a solution of the original system}\big\},$$

for some monotone functions $f_1, \ldots, f_m$.

The construction is by a straightforward decomposition of equations, with new variables representing subexpressions of the sides of the original equations. Once the equations are thus transformed, the system can be encoded as follows.

Lemma 4. *For every system of equations over sets of natural numbers in variables $(X_1, \ldots, X_n)$ and with all equations of the form $X = Y + Z$, $X = Y \cup Z$ or $X = C$, there exists a system in variables $(X'_1, \ldots, X'_n)$, using only addition and constants $\{0,1\}$, $\{0,2\}$, $\{0,4,11\}$, $\sigma(\{0\})$ and $\sigma(C)$ with C used in the original system and the ultimately periodic constant from Lemma 1, such that $X'_i = S'_i$ is a solution of the latter system if and only if the former system has a solution $X_i = S_i$ with $S'_i = \sigma(S_i)$.*

Proof. The proof is by transforming this system according to Lemmata 1 and 2. First, the new system contains the following equation for each variable X':

$$X' + \{0,4,11\} = \bigcup_{\substack{i\in\{0,4,6,8,9,\\10,12,13\}}} \tau_i(\mathbb{N}) \cup \bigcup_{i\in\{1,3,7\}} \tau_i(\mathbb{N}+1) \cup \{11\}. \tag{1}$$

Next, for each equation $X = Y + Z$ in the original system, there is a corresponding equation

$$X' + \sigma(\{0\}) + \{0,1\} = Y' + Z' + \{0,1\} \tag{2}$$

in the new system. Similarly, for each equation of the form $X = Y \cup Z$, the new system contains an equation

$$X' + X' + \{0,2\} = Y' + Z' + \{0,2\}. \tag{3}$$

Finally, every equation $X = C$ in the original system is represented in the new system by the following equation:

$$X' = \sigma(C). \tag{4}$$

By Lemma 1, (1) ensures that each solution $(S'_1,\ldots,S'_n)$ of the constructed system satisfies $S'_i = \sigma(S_i)$ for some sets $S_i \subseteq \mathbb{N}$. It is claimed that $(S'_1,\ldots,S'_n)$ satisfies each equation of the original system if and only if $(S_1,\ldots,S_n)$ satisfies the corresponding equation (2)–(4) of the constructed system. Consider each pair of corresponding equations:

- Consider an equation $X = Y + Z$ from the original system. Then there is a corresponding equation (2), and, by Lemma 2, $(S_1,\ldots,S_n)$ satisfies the original equation if and only if $(S'_1,\ldots,S'_n)$ satisfies the corresponding equation (2).
- Similarly, by Lemma 2, an equation of the form $X = Y \cup Z$ is satisfied by $(S_1,\ldots,S_n)$ if and only if $(S'_1,\ldots,S'_n)$ satisfies (3).
- For each equation of the form $X_i = C$ it is claimed that a set S_i satisfies it if and only if $\sigma(S'_i)$ satisfies the corresponding equation (4). Indeed, $\sigma(S_i) = \sigma(C)$ if and only if $\textsc{track}_{13}(\sigma(S_i)) = \textsc{track}_{13}(\sigma(C))$, and since $\textsc{track}_{13}(\sigma(S_i)) = S'_i$ and $\textsc{track}_{13}(\sigma(C)) = C$, this is equivalent to $S_i = C$.

This shows that $(S_1,\ldots,S_n)$ satisfies the original system if and only if $(S'_1,\ldots,S'_n)$ satisfies the constructed system, which proves the correctness of the construction. $\square$

Note that σ is a bijection between the sets of solutions of the two systems. Then, in particular, if the original system has a unique solution, then the constructed system has a unique solution as well, which encodes the solution of the original system.

Furthermore, it is important that the encoding σ respects inclusion, that is, if $X \subseteq Y$, then $\sigma(X) \subseteq \sigma(Y)$. Consider the partial order on solutions of a system, defined as $(S_1,\ldots,S_n) \preccurlyeq (S'_1,\ldots,S'_n)$ if $S_i \subseteq S'_i$ for all i. Now if one solution of the original system is less than another, then the corresponding solutions of the constructed system maintain this relation. Therefore, if the original system has a least (greatest) solution with respect to this partial order, then so does the new one, and its least (greatest) solution is the image of the least (greatest) solution of the original system.

These observations allow applying Lemmata 3 and 4 to encode each system in Theorem 1 within a system using addition only.

Theorem 2. *For every recursive (r.e., co-r.e.) set $T \subseteq \mathbb{N}$ there exists a system of equations*

$$\begin{cases} \varphi_1(X_1,\ldots,X_n) &=& \psi_1(X_1,\ldots,X_n) \\ & \vdots & \\ \varphi_m(X_1,\ldots,X_n) &=& \psi_m(X_1,\ldots,X_n) \end{cases}$$

with φ_j,ψ_j using the operation of addition and ultimately periodic constants, which has a unique (least, greatest, respectively) solution with $X_1 = S$, where $T = \{n \mid 16n+13 \in S\}$.

To be precise, the construction requires finite constants $\{0,1\}$, $\{0,2\}$, $\{0,4,11\}$, an infinite constant $\sigma(\{1\})$ encoding the set used in Theorem 1, another infinite constant $\sigma(\{0\})$ required by Lemma 2, and one more infinite constant from Lemma 1.

Note that T is computationally reducible to S via the "$16n + 13$" transduction, hence the following statements:

Corollary 1. *For every recursive set $T \subseteq \mathbb{N}$ there exists a system of equations over sets of natural numbers using addition and ultimately periodic constants that has a unique solution, which is computationally as hard as T.*

Corollary 2. *There exists a system of equations over sets of natural numbers using addition and ultimately periodic constants which has a least (greatest) solution with its first component being r.e.-complete (co-r.e.-complete, respectively).*

Finally, the decision problems for these systems of equations turn out to be as hard as in the case of union and addition:

Theorem 3. *The problem of testing whether a system of equations over sets of natural numbers using addition and ultimately periodic constants has a solution is Π_1^0-complete. The problem of whether it has a unique, least or greatest solution is Π_2^0-complete. The problem of whether it has finitely many solutions is Σ_3-complete.*

The above results equally apply to *language equations* over a one-letter alphabet with concatenation as the only allowed operation and with regular constants.

Corollary 3. *For every recursive (r.e., co-r.e.) language $L \subseteq a^*$ there exists a system of language equations*

$$\begin{cases} \varphi_1(X_1, \ldots, X_n) & = & \psi_1(X_1, \ldots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \ldots, X_n) & = & \psi_m(X_1, \ldots, X_n) \end{cases}$$

with φ_j, ψ_j using the operation of concatenation and regular constants over a one-letter alphabet, which has a unique (least, greatest, respectively) solution with $X_1 = L'$, where $L = \{a^n \mid a^{16n+13} \in L'\}$.

Testing whether a system of language equations of this form has a solution is Π_1^0-complete; testing whether it has a unique, least or greatest solution is Π_2^0-complete; testing for finitely many solutions is Σ_3^0-complete.

6. Equations over sets of integers

The purpose of this section is to obtain a similar result for equations over sets of integers: namely, that equations with union and addition can be simulated by equations with addition only, using a certain simple encoding.

Turning to the expressive power of equations over sets of integers with union and addition, the sets representable by their unique solutions are exactly the *hyper-arithmetical sets* [21]. These sets can be defined in terms of the analytical hierarchy as follows. Let Σ_1^1 denote the class of sets definable by existential second-order arithmetical formulae, that is, sets of the form

$$\{n \in \mathbb{Z} \mid \exists X \subseteq \mathbb{Z} : \varphi(X, n)\},$$

where φ is a first-order formula containing subexpressions $x \in X$. The sets in Π_1^1 are defined similarly, with only universal second-order quantification. The intersection of these classes, $\Delta_1^1 = \Sigma_1^1 \cap \Pi_1^1$, is the class of hyper-arithmetical sets. It has the following representation by equations over sets of integers, similar to the representation of recursive sets by equations over sets of natural numbers given in Theorem 1:

Theorem 4 (Jeż, Okhotin [8]). *For every hyper-arithmetical set $S \subseteq \mathbb{Z}$ there is a system of equations over sets of integers using union, addition, and the constants $\{1\}$, $\mathbb{N}$ and $-\mathbb{N}$, which has a unique solution $(S, \ldots)$.*

And conversely, if a set is given by a unique solution of a system of equations over sets of integers using any operations representable in first-order Peano arithmetic, then this set is hyper-arithmetical.

Systems produced by Theorem 4 shall now be encoded into systems using addition only along the same lines as in Sections 3–5. Sets of integers are subjected to generally the same transformation as in the case of natural numbers: for every set $T \subseteq \mathbb{Z}$, its *encoding* as the set

$$S = \sigma(T) = \{0\} \cup \tau_6(\mathbb{Z}) \cup \tau_8(\mathbb{Z}) \cup \tau_9(\mathbb{Z}) \cup \tau_{12}(\mathbb{Z}) \cup \tau_{13}(T),$$

where the τ-notation for tracks is obviously extended from the natural numbers to integers and the period 16 is maintained.

The first result on this encoding is that the condition of a set X being an encoding of any set can be specified by an equation of the form $X + C = D$.

Lemma 5 (cf. Lemma 1). *A set $S \subseteq \mathbb{Z}$ satisfies an equation*

$$S + \{0, 4, 11\} = \bigcup_{\substack{i \in \{0,1,3,4,6,7,\\ 8,9,10,12,13\}}} \tau_i(\mathbb{Z}) \cup \{11\}$$

if and only if $S = \sigma(T)$ for some $T \subseteq \mathbb{Z}$.

Sketch of the proof. $\Leftarrow$ Let S be any set satisfying the equation. Similarly as in Lemma 1, first the empty tracks of S can be identified, using the property that $S + \{0, 4, 11\}$ has empty tracks 2, 5, 14 and 15. By the same argument as in Lemma 1, S has empty tracks $\{2, 5, 14, 15\} - \{0, 4, 11\} \pmod{16} = \{1, 2, 3, 4, 5, 7, 10, 11, 14, 15\}$.

Then a similar analysis is applied to track 11 of $S + \{0, 4, 11\}$: every track t of S with $t + t' = 11$ for some $t' \in \{0, 4, 11\}$ has to be either empty or encode $\{0\}$, and the latter should hold for at least one such t. This yields that track 0 of S encodes $\{0\}$.

As the last step, it is shown that S has the appropriate full tracks. To this end, full tracks $\{3, 7, 9, 10\}$ of $S + \{0, 4, 11\}$ are investigated. Each such track t is the union of the tracks $t - \{0, 4, 11\}$ of S. It turns out that in each case two of these tracks are already known to be empty, and it is thus concluded that tracks $\{6, 8, 9, 12\}$ of S are full.

$\Rightarrow$ To show that $S = \sigma(T)$ for some $T \subseteq \mathbb{Z}$ satisfies the equation

$$S + \{0, 4, 11\} = \bigcup_{\substack{i \in \{0,1,3,4,6,7,\\ 8,9,10,12,13\}}} \tau_i(\mathbb{Z}) \cup \{11\},$$

the set $\sigma(T)$ is represented as the union $\sigma(T) = \bigcup_{i=0}^{15} \tau_i(\textsc{track}_i(\sigma(T)))$, and this representation is transferred to $\sigma(T) + \{0, 4, 11\}$:

$$\sigma(T) + \{0, 4, 11\} = \Big(\bigcup_i \tau_i(\textsc{track}_i(\sigma(T))) + 0\Big) \cup \Big(\bigcup_i \tau_i(\textsc{track}_i(\sigma(T))) + 4\Big) \cup \Big(\bigcup_i \tau_i(\textsc{track}_i(\sigma(T))) + 11\Big).$$

Calculations as in Lemma 1 are made, with the result given in Table 4. The last column of this table contains the result of this whole calculation, and it can be seen that each copy of T in the union is overwritten by some full track. $\square$

Now, assuming that the given system of equations with union and addition is decomposed to have all equations of the form $X = U + V$, $X = U \cup V$ or $X = \textit{const}$, these equations can be simulated in a new system as follows:

tracks	$\sigma(T)$	$\sigma(T)+\{0\}$	$\sigma(T)+\{4\}$	$\sigma(T)+\{11\}$	$\sigma(T)+\{0,4,11\}$
0	$\{0\}$	$\{0\}$	$\mathbb{Z}$		
1			$T+1$	$\mathbb{Z}+1$	$\mathbb{Z}+1$
2					
3			$\mathbb{Z}+1$		$\mathbb{Z}$
4			$\{0\}$	$\mathbb{Z}+1$	$\mathbb{Z}$
5					
6	$\mathbb{Z}$	$\mathbb{Z}$			$\mathbb{Z}$
7				$\mathbb{Z}+1$	$\mathbb{Z}+1$
8	$\mathbb{Z}$	$\mathbb{Z}$		$T+1$	$\mathbb{Z}$
9	$\mathbb{Z}$	$\mathbb{Z}$			$\mathbb{Z}$
10			$\mathbb{Z}$		$\mathbb{Z}$
11				$\{0\}$	$\{0\}$
12	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$		$\mathbb{Z}$
13	T	T	$\mathbb{Z}$		$\mathbb{Z}$
14					
15					

Table 4: Tracks in the sum $\sigma(T)+\{0,4,11\}$. Empty cells represent empty tracks.

	$0:\{0\}$	$6:\mathbb{Z}$	$8:\mathbb{Z}$	$9:\mathbb{Z}$	$12:\mathbb{Z}$	$13:V$
$0:\{0\}$	$0:\{0\}$	$6:\mathbb{Z}$	$8:\mathbb{Z}$	$9:\mathbb{Z}$	$12:\mathbb{Z}$	$13:V$
$6:\mathbb{Z}$	$6:\mathbb{Z}$	$12:\mathbb{Z}$	$14:\mathbb{Z}$	$15:\mathbb{Z}$	$2:\mathbb{Z}$	$3:?$
$8:\mathbb{Z}$	$8:\mathbb{Z}$	$14:\mathbb{Z}$	$0:\mathbb{Z}$	$1:\mathbb{Z}$	$4:\mathbb{Z}$	$5:?$
$9:\mathbb{Z}$	$9:\mathbb{Z}$	$15:\mathbb{Z}$	$1:\mathbb{Z}$	$2:\mathbb{Z}$	$5:\mathbb{Z}$	$6:?$
$12:\mathbb{Z}$	$12:\mathbb{Z}$	$2:\mathbb{Z}$	$4:\mathbb{Z}$	$5:\mathbb{Z}$	$8:\mathbb{Z}$	$9:?$
$13:U$	$13:U$	$3:?$	$5:?$	$6:?$	$9:?$	$10:(U+V)+1$

Table 5: Tracks in the sum $\sigma(U)+\sigma(V)$. Question marks denote sets that depend on U or V and whose actual values are unimportant.

Lemma 6 (cf. Lemma 2)**.** *For all sets $X,U,V \subseteq \mathbb{Z}$,*

$$\sigma(U)+\sigma(V)+\{0,1\} = \sigma(X)+\sigma(\{0\})+\{0,1\} \quad \textit{if and only if} \quad U+V=X$$
$$\sigma(U)+\sigma(V)+\{0,2\} = \sigma(X)+\sigma(X)+\{0,2\} \quad \textit{if and only if} \quad U\cup V=X.$$

Sketch of the proof. As in Lemma 2, it is shown that for all $U,V\subseteq\mathbb{Z}$, the sum $\sigma(U)+\sigma(V)$ encodes the sets $U+V+1$ and $U\cup V$ on some of its tracks. Both of those encodings can be uniquely recovered by adding constants $\{0,1\}$ and $\{0,2\}$, respectively.

First, $\sigma(U)+\sigma(V)$ is calculated. Using

$$\sigma(U) = \{0\}\cup\tau_6(\mathbb{Z})\cup\tau_8(\mathbb{Z})\cup\tau_9(\mathbb{Z})\cup\tau_{12}(\mathbb{Z})\cup\tau_{13}(U) \quad \text{and}$$
$$\sigma(V) = \{0\}\cup\tau_6(\mathbb{Z})\cup\tau_8(\mathbb{Z})\cup\tau_9(\mathbb{Z})\cup\tau_{12}(\mathbb{Z})\cup\tau_{13}(V),$$

this sum can be represented as a union of 36 non-empty terms, given in Table 5. Question marks denote tracks with unspecified contents. The value of each track in $\sigma(U)+\sigma(V)$ is given in the corresponding column of Table 6.

Knowing the sets encoded on the tracks of $\sigma(U)+\sigma(V)$ allows calculating the sets encoded on tracks of $\sigma(U)+\sigma(V)+\{0,1\}$, which is done in Table 6 as well. The result is that for all U and V,

$$\textsc{track}_{11}(\sigma(U)+\sigma(V)+\{0,1\}) = U+V+1,$$
$$\textsc{track}_i(\sigma(U)+\sigma(V)+\{0,1\}) = \mathbb{Z}, \qquad \text{for } i\neq 11.$$

	$\sigma(U)$	$\sigma(V)$	$\sigma(U)+\sigma(V)$	$\sigma(U)+\sigma(V)+\{0,1\}$	$\sigma(U)+\sigma(V)+\{0,2\}$
0	$\{0\}$	$\{0\}$	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$
1			$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$
2			$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$
3			?	$\mathbb{Z}$	$\mathbb{Z}$
4			$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$
5			$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$
6	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$
7				$\mathbb{Z}$	$\mathbb{Z}$
8	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$
9	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$
10			$U+V+1$	$\mathbb{Z}$	$\mathbb{Z}$
11				$U+V+1$	$\mathbb{Z}$
12	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$
13	U	V	$U \cup V$	$\mathbb{Z}$	$U \cup V$
14			$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$
15			$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$

Table 6: Tracks in the sums of $\sigma(U) + \sigma(V)$ with constants.

Thus

$$X = U + V$$

holds if and only if

$$\sigma(X) + \sigma(\{0\}) + \{0,1\} = \sigma(U) + \sigma(V) + \{0,1\}.$$

For the set $\sigma(U) + \sigma(V) + \{0,2\}$, similarly,

$$\begin{aligned} \text{TRACK}_{13}(\sigma(U) + \sigma(V) + \{0,2\}) &= U \cup V, \\ \text{TRACK}_{j}(\sigma(U) + \sigma(V) + \{0,2\}) &= \mathbb{Z}, \qquad \text{for } j \neq 13 \end{aligned}$$

and therefore for all X, U, V,

$$X = U \cup V$$

if and only if

$$\sigma(X) + \sigma(X) + \{0,2\} = \sigma(U) + \sigma(V) + \{0,2\}.$$

□

Using these two lemmata, one can simulate any system with addition and union by a system with addition only. Taking systems representing different hyper-arithmetical sets, the following result on the expressive power of systems with addition can be established:

Theorem 5. *For every hyper-arithmetical set $S \subseteq \mathbb{Z}$ there exists a system of equations over sets of integers using the operation of addition and ultimately periodic constants, which has a unique solution with $X_1 = T$, where $S = \{n \mid 16n \in T\}$.*

The constants used in the construction are $\{0,1\}$, $\{0,2\}$, $\sigma(\{0\})$, $\sigma(\{1\})$, $\sigma(\mathbb{N})$, $\sigma(-\mathbb{N})$ and the constant in Lemma 5.

Sketch of a proof. A system of equations with union and addition representing S exists by Theorem 4. This system is first decomposed to have all equations of the form $X = U + V$, $X = U \cup V$ or $X = C$. For every variable X of this system, the new system has a variable X' with an equation as in Lemma 5. Next, according to Lemma 6, the equations $U + V = X$, $U \cup V = X$ or $X = C$ are transformed to equations

$U' + V' + \{0, 1\} = X' + \sigma(\{0\}) + \{0, 1\}$, $U' + V' + \{0, 2\} = X' + X' + \{0, 2\}$ and $X' = \sigma(C)$, respectively, and the resulting system should have a unique solution with $X' = \sigma(X)$. Thus the constructed system represents the set $\sigma(S)$, and adding an extra equation $X_1 = X + \{-13\}$ yields the set $T = \sigma(S) - 13$ with the desired properties. □

7. Systems with finite constants

Both constructions above essentially use infinite ultimately periodic constants. It will now be shown that the use of such constants is necessary, and systems using only addition and *finite* constants cannot specify any non-trivial infinite sets, neither in the case of sets of natural of numbers, nor in the case of sets of integers.

This is done by demonstrating that every solution $(\dots, S, \dots)$ of such a system can be *pruned* in the sense that each of its infinite components can be replaced by an empty set and the resulting vector remains a solution.

Lemma 7. *Consider a system of equations over sets of natural numbers or over sets of integers, in variables $(\dots, X_j, \dots, Y_i, \dots)$, using addition and only finite constants. If the system has a solution $(\dots, F_j, \dots, S_i, \dots)$, where each F_j is finite and each S_i infinite, then $(\dots, F_j, \dots, \varnothing, \dots)$ is a solution as well.*

Proof. Let $(\dots, F_j, \dots, S_i, \dots)$ be a solution and consider its substitution into each equation:

$$\varphi(\dots, X_j, \dots, Y_i, \dots) = \psi(\dots, X_j, \dots, Y_i, \dots).$$

If both sides equal $\varnothing$ under this substitution, then another substitution $X_j = F_j$, $Y_i = \varnothing$ produces $\varnothing$ on both sides as well.

If both sides produce a finite set, this means that neither φ nor ψ refer to any variables Y_i. Therefore, the substitution of $X_j = F_j$, $Y_i = \varnothing$ produces the same value of both sides.

Finally, assume that the substitution yields an infinite set. As there are no infinite constants and all X_j have finite values, this means that each side contains some Y-variable. Hence, under the substitution $X_j = F_j$, $Y_i = \varnothing$, both sides evaluate to $\varnothing$. □

In a similar way, infinite components of a solution can be augmented to co-finite sets. This time the result differs for sets of natural numbers and for sets of integers.

Consider first the case of natural numbers. For every nonempty set $S \subseteq \mathbb{N}$, its *upward closure* $S + \mathbb{N}$ is always co-finite.

Lemma 8. *Consider a system of equations over sets of natural numbers in variables $(\dots, X_j, \dots, Y_i, \dots)$ using addition and only finite constants. If it has a solution $(\dots, F_j, \dots, S_i, \dots)$, where each F_j is finite and each S_i infinite, then $(\dots, F_j, \dots, S_i + \mathbb{N}, \dots)$ is a solution as well.*

Proof. As in the previous lemma, let $(\dots, X_j, \dots, Y_i, \dots)$ be a solution, which is substituted into each equation

$$\varphi(\dots, X_j, \dots, Y_i, \dots) = \psi(\dots, X_j, \dots, Y_i, \dots).$$

If both sides evaluate to $\varnothing$ or to any finite nonempty set, these cases are handled as in Lemma 7.

Assume that the value of both sides under the substitution $X_j = F_j$, $Y_i = S_i$ is an infinite set S. Then both sides must contain occurrences of some Y-variables. Then the substitution $X_j = F_j$, $Y_i = S_i + \mathbb{N}$ produces $S + \mathbb{N}$ on both sides. This completes the proof that $(\dots, F_j, \dots, S_i + \mathbb{N}, \dots)$ is a solution. □

The same argument yields the following similar result for the case of integers.

Lemma 9. *If a system of equations with unknown sets of integers $(\dots, X_j, \dots, Y_i, \dots)$ using addition and only finite constants has a solution $(\dots, F_j, \dots, S_i, \dots)$, where each F_j is finite and each S_i infinite, then $(\dots, F_j, \dots, \mathbb{Z}, \dots)$ is a solution as well.*

Theorem 6. *If a system of equations over sets of natural numbers using addition and finite constants has a least (greatest, unique) solution $(\dots, S_i, \dots)$, then each S_i is finite (finite or co-finite, finite, respectively).*

For a system of equations over sets of integers using addition and finite constants, if it has a least (greatest, unique) solution $(\dots, S_i, \dots)$, then each S_i is finite (finite or $\mathbb{Z}$, finite, respectively).

Since equations with finite constants have so trivial solutions, it is natural to expect their decision problems to be much easier than in Theorem 3. Establishing the exact complexity of these problems is left for future work.

8. The origin of the encoding

The encoding σ of sets of natural numbers, as well as its variant for sets of integers, has been constructed and proved correct, but no comments have yet been given on how it was obtained.

The first to be invented was the general idea of an encoding $\sigma\colon 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ with a period p and a single data track d, satisfying $pn + d \in \sigma(S)$ if and only if $n \in S$, and with the rest of the tracks in $\sigma(S)$ independent of S. It was further assumed that each of the rest of the tracks must be either a singleton, or empty, or full, and the general form of the statements of Lemmata 1 and 2 were postulated. These properties were first expressed in the form that an equation $X + C = D$ should check the correctness of the encoding, the sum $\sigma(S) + \sigma(T)$ should contain tracks for the union and for the sum of S and T, and that adding constants $\{0, e\}$ and $\{0, e'\}$ with $e, e' \geqslant 1$ to this sum should isolate the union track and the sum track, respectively. The constants C and D were also assumed to have a track structure modulo p.

What remained was to find the precise encoding σ, given by the numbers p and d and by the composition of its singleton, empty and full tracks, as well as the tracks structure of C and D and the numbers e and e' for the future Lemmata 1 and 2. This was achieved by an exhaustive search over all possibilities, done by a computer program for increasing p. Such encodings and the constants for the associated equations were successfully found, and the period $p = 16$ used in this paper was simply the smallest number for which the above requirements all held at the same time.

Although the encoding provided in this paper is the smallest one, it is probably not the simplest one, as it compresses all the necessary checks as tightly as possibly. A man-made encoding with more tracks could allow more intuitive arguments. Finding such an encoding and producing better readable arguments is left as an open problem.

9. Conclusion

The study of language equations, surveyed in a recent paper by Kunc [10], has progressed by showing the computational universality of simpler and simpler models [17, 9, 6]. The equations proved universal in this paper are the simplest considered so far: the constructions use systems of equations $X = YZ$ and $X = C$ over an alphabet $\Sigma = \{a\}$, with ultimately periodic constants $C \subseteq a^*$. Little room is left for further improvement, as infinite constants were proved to be essential.

This result, in particular, has implications on the recent research on language equations, in which concatenation is replaced by other operations [1], ranging between the shuffle product used to model concurrency, and some new operations motivated by bio-informatics. Over a one-letter alphabet, most of such operations coincide with concatenation, and thus this paper has shown computational completeness of many previously considered classes of language equations.

An apparent omission in this paper is the lack of any simple example of a system of equations over sets of numbers with addition and ultimately periodic constants, which would represent any non-periodic set. The reason is that the given method of constructing such systems depends on first constructing a system with union and addition, while already for the latter, the smallest known example of representing a non-periodic set requires hundreds of variables [6], and if such an example is further subjected to an encoding into 16 tracks, the result would hardly be intuitive. Constructing a small example of such a system of equations with a non-periodic unique (or least, or greatest) solution would perhaps lead to a better understanding of what makes these simplest of language equations computationally universal.

[1] M. Domaratzki, K. Salomaa, “Decidability of trajectory-based equations”, *Theoretical Computer Science*, 345:2–3 (2005), 304–330.

[2] S. Ginsburg, H. G. Rice, “Two families of languages related to ALGOL”, *Journal of the ACM*, 9 (1962), 350–371.

[3] A. Jeż, “Conjunctive grammars can generate non-regular unary languages”, *International Journal of Foundations of Computer Science*, 19:3 (2008), 597–615.

[4] A. Jeż, A. Okhotin, “Conjunctive grammars over a unary alphabet: undecidability and unbounded growth”, *Theory of Computing Systems*, 46:1 (2010), 27–58.

[5] A. Jeż, A. Okhotin, “Complexity of solutions of equations over sets of natural numbers”, *25th Annual Symposium on Theoretical Aspects of Computer Science* (STACS 2008, Bordeaux, France, 21–23 February, 2008), 373–383.

[6] A. Jeż, A. Okhotin, “On the computational completeness of equations over sets of natural numbers”, *Automata, Languages and Programming* (ICALP 2008, Reykjavík, Iceland, July 6–13, 2008), part II, LNCS 5126, 63–74.

[7] A. Jeż, A. Okhotin, “One-nonterminal conjunctive grammars over a unary alphabet”, *Computer Science in Russia* (CSR 2009, Novosibirsk, Russia, 18–23 August, 2009), LNCS 5675, 191–202.

[8] A. Jeż, A. Okhotin, “On equations over sets of integers”, *STACS 2010* (Nancy, France, 4–6 March, 2010), 477–488.

[9] M. Kunc, “The power of commuting with finite sets of words”, *Theory of Computing Systems*, 40:4 (2007), 521–551.

[10] M. Kunc, “What do we know about language equations?”, *Developments in Language Theory* (DLT 2007, Turku, Finland, July 3–6, 2007), LNCS 4588, 23–27.

[11] E. L. Leiss, “Unrestricted complementation in language equations over a one-letter alphabet”, *Theoretical Computer Science*, 132 (1994), 71–93.

[12] P. McKenzie, K. Wagner, “The complexity of membership problems for circuits over sets of natural numbers”, *Computational Complexity*, 16:3 (2007), 211–244.

[13] A. Okhotin, “Conjunctive grammars”, *Journal of Automata, Languages and Combinatorics*, 6:4 (2001), 519–535.

[14] A. Okhotin, “Boolean grammars”, *Information and Computation*, 194:1 (2004), 19–48.

[15] A. Okhotin, “On the equivalence of linear conjunctive grammars to trellis automata”, *Informatique Théorique et Applications*, 38:1 (2004), 69–88.

[16] A. Okhotin, “Unresolved systems of language equations: expressive power and decision problems”, *Theoretical Computer Science*, 349:3 (2005), 283–308.

[17] A. Okhotin, “Decision problems for language equations”, *Journal of Computer and System Sciences*, 76:3–4 (2010), 251–266.

[18] A. Okhotin, “Fast parsing for Boolean grammars: a generalization of Valiant’s algorithm”, *Developments in Language Theory* (DLT 2010, London, Ontario, Canada, August 17–20, 2010), LNCS 6224, 340–351.

[19] A. Okhotin, O. Yakimova, “On language equations with complementation”, *Developments in Language Theory* (DLT 2006, Santa Barbara, USA, June 26–29, 2006), LNCS 4036, 420–432.

[20] A. Okhotin, O. Yakimova, “Language equations with complementation: Decision problems”, *Theoretical Computer Science*, 376:1–2 (2007), 112–126.

[21] H. Rogers, Jr., *Theory of Recursive Functions and Effective Computability*, McGraw-Hill, 1967.

[22] L. J. Stockmeyer, A. R. Meyer, “Word problems requiring exponential time”, *STOC 1973*, 1–9.

[23] T. Tao, V. Vu, *Additive Combinatorics*, Cambridge University Press, 2006.

[24] S. D. Travers, “The complexity of membership problems for circuits over sets of integers”, *Theoretical Computer Science*, 369:1–3 (2006), 211–229.