Information and Computation 237 (2014) 56–94

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Computational completeness of equations over sets of natural numbers ☆

Artur Jeż [a,b,*], Alexander Okhotin [c]

[a] Max-Planck-Institut für Informatik, Campus E1 4, D-66123, Saarbrücken, Germany
[b] Institute of Computer Science, University of Wrocław, 50-383 Wrocław, Poland
[c] Department of Mathematics and Statistics, University of Turku, Turku FI-20014, Finland

ARTICLE INFO

Article history:
Received 21 July 2012
Received in revised form 21 February 2014
Available online 14 May 2014

Keywords:
Language equations
Unary languages
Computability

ABSTRACT

Systems of finitely many equations of the form $\varphi(X_1,\ldots,X_n)=\psi(X_1,\ldots,X_n)$ are considered, in which the unknowns X_i are sets of natural numbers, while the expressions φ,ψ may contain singleton constants and the operations of union and pairwise addition $S+T=\{m+n \mid m\in S,\ n\in T\}$. It is shown that the family of sets representable by unique (least, greatest) solutions of such systems is exactly the family of recursive (r.e., co-r.e., respectively) sets of numbers. Basic decision problems for these systems are located in the arithmetical hierarchy. The same results are established for equations with addition and intersection.

1. Introduction

Consider equations, in which the variables assume values of sets of natural numbers, and the left- and right-hand sides use Boolean operations and element-wise addition of sets defined as $S+T=\{m+n \mid m\in S,\ n\in T\}$. The simplest example of such an equation is $X=(X+X)\cup\{2\}$, with the set of all positive even numbers as the least solution. On the one hand, such equations constitute a basic mathematical object, which is closely related to *integer expressions*, introduced in the seminal paper by Stockmeyer and Meyer [31] and later systematically studied by McKenzie and Wagner [21], alongside the more general *integer circuits*; equations over sets of numbers can be regarded as circuits with circular references. On the other hand, they can be regarded as *language equations* over a one-symbol alphabet, with the addition operation on sets of natural numbers representing concatenation of such languages.

Language equations are equations with formal languages as unknowns. They frequently appear in theoretical computer science, wherever any relationship between sets of strings is expressed. For instance, Ginsburg and Rice [8] used systems of language equations to define the semantics of formal grammars, Baader and Narendran [3] applied another kind of language equations to represent unification in description logics, while Martens et al. [19] represented some questions on XML schema design by language equations of yet another form, and used this representation to establish their computational complexity. A theoretical investigation of language equations was initiated in the pioneering monographs by Conway [6] and by Leiss [18]. In the recent decade, this subject became an active area of research, with the state-of-the-art as of 2007 presented in a survey by Kunc [16].

☆ The extended abstract of this paper was presented at the 35th International Colloquium on Automata, Languages and Programming (ICALP 2008) held in Reykjavík, Iceland on July 6–13, 2008.

* Corresponding author.
E-mail addresses: aje@cs.uni.wroc.pl (A. Jeż), alexander.okhotin@utu.fi (A. Okhotin).

http://dx.doi.org/10.1016/j.ic.2014.05.001
0890-5401/

Connections between language equations and computability have long been noted. Undecidability of the solution existence problem for language equations with concatenation and Boolean operations was established by Parikh et al. [29], in relation with the research on process logic. The same undecidability was independently shown by Charatonik [4], who regarded language equations as a variant of set constraints [1,5]. Further connections to computability were established by Okhotin [23,25,26], who determined that, for every alphabet Σ containing at least two symbols, the family of sets representable by unique solutions of language equations over Σ is exactly the family of recursive languages over Σ; languages represented by least solutions of such equations are exactly the recursively enumerable (r.e.) sets, whereas greatest solutions of language equations define the complements of the recursively enumerable sets (co-r.e.). Shortly thereafter, Kunc [15] constructed a language equation of the form $XL = LX$, where $L \subseteq \{a, b\}^*$ is a finite constant language, with a computationally universal greatest solution, thus answering a problem raised by Conway [6]. The main ideas of the general computational completeness results for language equations are summarized in Section 2 of this paper.

The cited results essentially use languages over alphabets containing at least two symbols, as they rely on representing structured information in strings and on preserving this information when strings are concatenated. For a one-symbol alphabet $\Sigma = \{a\}$, strings degenerate to natural numbers, and concatenation means simply adding these numbers. The existing methods for proving computational universality in language equations are certainly not suited for expressing anything in this case.

Until recently, the seemingly trivial case of language equations over a one-symbol alphabet $\Sigma = \{a\}$ received little attention. Systems of the form

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

with *union and concatenation* represent context-free grammars, and their solutions over a unary alphabet are well-known to be regular [8]. Constructing any equation with regular constants and a non-regular unique solution is already not a trivial task. The first example of such an equation using the operations of *concatenation and complementation* was presented by Leiss [17]: that was an individual example, without a systematic method for producing any further examples. A few years ago, Jeż [10] constructed a system $(*)$ using *concatenation, union and intersection*, which has a non-regular solution, and this example had a clear explanation in terms of base-4 representation of numbers: concatenating and intersecting several unknown sets ultimately allowed defining the set of powers of four. Using this method, Jeż [10] proved that every set of numbers with their base-k representation recognized by a finite automaton over the alphabet of base-k digits—known as an *automatic set* in some of the literature [2]—can be defined by such a system of equations. This method was further developed by Jeż and Okhotin [11], who showed that equations $(*)$ with concatenation, union and intersection can simulate a simple kind of cellular automaton [7] recognizing positional representations of numbers. The membership of a number in a solution can be tested in exponential time, and it was subsequently shown that a particular EXPTIME-complete set of numbers can be defined by a system of this form [12]. This result was improved soon thereafter: already a resolved equation $X = \varphi(X)$ with one variable can define such an EXPTIME-complete set [13].

These recent advances suggest the task of understanding the exact limits of the expressive power of equations over sets of numbers (or language equations over a unary alphabet) of the general form

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

Unexpectedly, this paper establishes computational completeness of these systems. To be precise, for systems $(**)$, in which the expressions φ_j, ψ_j on both sides use only addition and union, as well as singleton constants, it is proved that a set is representable as a component of a unique solution of such a system if and only if this set is recursive. The same result is obtained for systems with addition and intersection. Similar characterizations are obtained for least and greatest solutions (by r.e. and co-r.e. sets, respectively). This characterizes the notion of computability by extremely simple arithmetical equations. The proof method can be described as a new kind of arithmetization of Turing machines, which uses addition of natural numbers as the only arithmetical operation, applied to sets of numbers element-wise.

This paper begins with establishing the claimed computational completeness result in its weaker version, using systems $(**)$ with union, intersection and addition. The general line of the computational completeness argument presented in this paper models upon the existing computational completeness results for language equations [23], which are summarized in Section 2. The existing argument for language equations is by first defining the *language of computation histories* of a Turing machine [9] by a system of equations, and then using a special inequality that *extracts* the language recognized by the machine from the language of its computation histories. These equations naturally use concatenation of languages, where strings containing structured information are concatenated to each other. Re-implementing the same steps of the construction by adding numbers instead of concatenating strings requires more effort. In the new construction, each computation history is represented by a number, the set of all computation histories is a set of numbers, and the task is to extract the set of numbers accepted by the machine out of this set of computation histories.

The known methods of manipulating data in equations over sets of numbers [10,11] are explained in Section 3: these methods use equations with union, intersection and addition. Those equations are used as main building blocks in the new construction, given in the next Section 4, where the computational completeness arguments for language equations are *remade* for equations over sets of numbers with *union, intersection and addition*.

These results are then improved in Section 5 to use systems (∗∗) with either only union or only intersection (referred to as *one Boolean operation* in the following), along with the addition. No examples of their non-triviality have been known up to date. The first examples of such systems defining non-regular sets, which are constructed in this paper, require elaborate constructions. The arguments proceed by re-implementing all core constructions from Section 3, and then all computational completeness constructions from Section 4, using these more restricted equations. In most cases, this is achieved by *translating* existing equations to use only one Boolean operation by a uniform transformation.

Finally, decision problems for equations over sets of numbers are considered in Section 6. It is proved that, like in the case of language equations [23,26], testing existence of a solution of a given system (∗∗) is undecidable (complete for Π_1^0 in the arithmetical hierarchy), testing solution uniqueness is Π_2^0-complete, while testing whether a system has finitely many solutions is Σ_3^0-complete.

2. Language equations and their computational completeness

Let Σ be a finite alphabet. The set of finite strings over Σ is denoted by Σ^*. The operation of writing two strings $u, v \in \Sigma^*$ one after another, as $u \cdot v = uv$, is called *concatenation*. There is an opposite operation of erasing a prefix, denoted by $u^{-1}(uv) = v$ and called *left quotient*, as well as of erasing a suffix, *right quotient*, $(uv)v^{-1} = u$. The length of a string $w = a_1 \dots a_n$ is the number of symbols in it, denoted by $|w| = n$. The unique string of length 0 is denoted by ε.

Any set of strings $L \subseteq \Sigma^*$ is called a *(formal) language*. The usual operations on languages are Boolean operations on sets and the *concatenation of languages*, defined as $KL = K \cdot L = \{uv \mid u \in K,\ v \in L\}$ for any two languages K and L. The *n-th power of concatenation* is $L^n = L \cdot \ldots \cdot L$, and, in particular, $L^0 = \{\varepsilon\}$. This paper also uses the *Kleene star* operation on languages, $L^* = \bigcup_{n \geqslant 0} L^n$, and its positive variant, $L^+ = \bigcup_{n \geqslant 1} L^n$.

Consider systems of equations of the form

$$\varphi_j(X_1, \dots, X_n) = \psi_j(X_1, \dots, X_n) \quad (1 \leqslant j \leqslant m),$$

where the unknowns $X_1, \dots, X_n$ are languages over Σ, and φ_j, ψ_j are expressions using some language-theoretic operations and some constant languages over Σ. When a system has a unique solution, it can be regarded as a definition of this solution. However, a system is, in general, not guaranteed to have a solution, and when it has any, the solution need not be unique. Multiple solutions of a system are partially ordered by componentwise inclusion as $(L_1, \dots, L_n) \sqsubseteq (L'_1, \dots, L'_n)$ if $L_i \subseteq L'_i$ for each i. Then, *least* and *greatest* solutions with respect to this ordering are also naturally considered.

The typical operations used in equations are Boolean operations and concatenation. Among them, the operations of union, intersection and concatenation are monotone with respect to the above ordering, where an n-ary operation φ is called *monotone*, if $(L_1, \dots, L_n) \sqsubseteq (L'_1, \dots, L'_n)$ implies $\varphi(L_1, \dots, L_n) \subseteq \varphi(L'_1, \dots, L'_n)$. A composition of monotone operations is monotone. Complementation is not monotone.

Another general property of operations is continuity. A sequence of languages $\{L_i\}_{i \geqslant 0}$ is *convergent*, if every string w either belongs to almost all elements L_i, or only to finitely many of them. In such a case, $\lim_{i \to \infty} L_i = \{w \mid w \text{ is in infinitely many } L_i\}$. This definition extends to sequences of vectors of languages and their limits. Now, an operation φ is *continuous*, if, for every convergent sequence of vectors $\{(L_1^{(i)}, \dots, L_n^{(i)})\}_{i=1}^{\infty}$,

$$\lim_{i \to \infty} \varphi\big(L_1^{(i)}, \dots, L_n^{(i)}\big) = \varphi\Big(\lim_{i \to \infty} \big(L_1^{(i)}, \dots, L_n^{(i)}\big)\Big).$$

Concatenation and all Boolean operations are continuous.

The following computational completeness result is known:

Theorem A *(Okhotin [23,25]). Let Σ be a finite alphabet and let $\varphi_j(X_1, \dots, X_n) = \psi_j(X_1, \dots, X_n)$ be a system of equations, where the unknowns X_i are languages over Σ, while φ_j and ψ_j are expressions using union, intersection and concatenation, as well as singleton constants. Assume that the system has a unique (least, greatest) solution $(L_1, \dots, L_n)$. Then each component L_i is recursive (r.e., co-r.e., respectively). Conversely, for every recursive (r.e., co-r.e.) language $L \subseteq \Sigma^*$ (with $|\Sigma| \geqslant 2$) there exists such a system (∗∗) with the unique (least, greatest, respectively) solution $(L, \dots)$. The transformation is effective in both directions.*

The upper bounds in Theorem A are proved by constructing Turing machines that test the membership of given strings in the solution of the desired type. These tests rely on the continuity of the operations used in the equations. As demonstrated in another paper by the authors [14], using the non-continuous operation of quotient to language equations, defined as $K \cdot L^{-1} = \{u \mid \exists v :\ uv \in KL,\ v \in L\}$, allows representing all hyper-arithmetical sets by unique solutions. As the hyper-arithmetical sets properly contain the whole arithmetical hierarchy, in which the recursive sets form the bottom level, those equations are strictly more powerful than allowed by Theorem A.

As this paper considers a much more restricted family of equations than those in Theorem A, the first part of the theorem will apply as it is, whereas the lower bound proofs will have to be entirely remade. The proof of the second part of Theorem A will serve as a model for the arguments in the case of equations over sets of numbers. The following sketch of this proof is useful for understanding the constructions presented later in this paper.

The main underlying device used for constructing such a system is the *language of computation histories* of a Turing machine, defined and used by Hartmanis [9] to give systematical undecidability proofs for context-free grammars. In short, for every Turing machine T over an input alphabet Σ, one can construct an alphabet Γ and an encoding of computations $C_T\colon \Sigma^* \to \Gamma^*$, so that for every string w accepted by T, the string $C_T(w)$ lists the configurations of T at each step of its accepting computation on w, and the language

$$\mathrm{VALC}(T) = \left\{ w \natural C_T(w) \;\middle|\; C_T(w) \text{ is an accepting computation} \right\},$$

where $\natural \notin \Sigma \cup \Gamma$, is an intersection of two linear context-free languages. Since equations as in Theorem A can directly simulate context-free grammars and are equipped with intersection, for every Turing machine it is easy to construct a system in variables $(X_1, \ldots, X_n)$ with a unique solution $(L_1, \ldots, L_n)$, so that $L_1 = \mathrm{VALC}(T)$.

It remains to "extract" $L(T)$ out of $\mathrm{VALC}(T)$ using a language equation. Let Y be a new variable and consider the inequality

$$\mathrm{VALC}(T) \subseteq Y \natural \Gamma^*, \tag{1}$$

which can be formally rewritten as an equation $X_1 \cup Y \natural \Gamma^* = Y \natural \Gamma^*$. This inequality states that for every string $w \in L(T)$, the computation history $w \natural C_T(w)$ should be in $Y \natural \Gamma^*$, that is, w should be in Y. This makes $L(T)$ the least solution of this inequality, and proves the second part of Theorem A with respect to r.e. sets and least solutions. The construction for a co-r.e. set and a greatest solution is established by a dual argument, and these two constructions can then be combined to represent every recursive set [25]. Furthermore, the same construction is used to establish the undecidability of the solution existence and solution uniqueness problems for a given language equation [23,26].

At the first glance, the idea that the above results could possibly hold if the alphabet consists of a single letter sounds odd. However, this is what will be proved in this paper, and, moreover, the general plan of the argument remains essentially the same.

3. Resolved systems with union, intersection and addition

A system of language equations is called *explicit* or *resolved* if it is of the form

$$X_i = \varphi_i(X_1, \ldots, X_n) \quad (1 \leqslant i \leqslant n).$$

As long as the right-hand sides φ_i are monotone, a least and a greatest solution exist by Tarski's set-theoretic fixpoint theorem. If, furthermore, the right-hand sides are continuous, the least solution is given by a componentwise union $\bigsqcup_{i \geqslant 0} \varphi^i(\varnothing, \ldots, \varnothing)$, where $\varphi = (\varphi_1, \ldots, \varphi_n)\colon (2^{\Sigma^*})^n \to (2^{\Sigma^*})^n$ is the right-hand side written of the system as a single function; this process is known as *fixpoint iteration*. Such equations are used to define the context-free grammars [8] (with the union and concatenation operations) and their generalization, the conjunctive grammars [22] (further using the intersection).

Consider formal languages over the alphabet $\Sigma = \{a\}$, known as *unary* or *tally* languages, which are the main topic of this paper. A language $L \subseteq a^*$ can be regarded as a set of natural numbers $\{n \mid a^n \in L\}$. A regular unary language corresponds to an ultimately periodic set of numbers.[1] Concatenation of unary languages turns into element-wise addition of sets of numbers, $S + T = \{m + n \mid m \in S,\ n \in T\}$. Equations over sets of numbers represent a very special subclass of language equations. If intersection is disallowed, such systems are basically context-free grammars over a unary (one-symbol) alphabet, and hence their solutions are ultimately periodic. Equations with both union and intersection are equivalent to conjunctive grammars over a unary alphabet, and the question whether any non-periodic set can be defined by such a system of equations had been open for some years, until answered by the following example.

Example 1 *(Jeż [10]).* 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 $(\{4^n \mid n \geqslant 0\}, \{2 \cdot 4^n \mid n \geqslant 0\}, \{3 \cdot 4^n \mid n \geqslant 0\}, \{6 \cdot 4^n \mid n \geqslant 0\})$.

[1] A set $S \subseteq \mathbb{N}$ is *ultimately periodic* if there exist such numbers $d \geqslant 0$ and $p \geqslant 1$, that for every $n \geqslant d$, the number n is in S if and only if $n + p$ is in S.

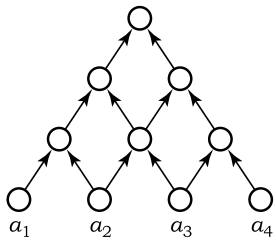

Fig. 1. Computation of a trellis automaton.

To understand this construction, it is useful to consider positional notation of numbers. Let $\Sigma_k = \{0, 1, \ldots, k-1\}$ be digits in base-k notation. For every string of digits $w \in \Sigma_k^*$, let $(w)_k$ be the number defined by this string of digits; the empty string ε denotes zero. Define $(L)_k = \{(w)_k \mid w \in L\}$.

Now the least solution of the above system can be conveniently represented in base-4 notation as $((10^*)_4, (20^*)_4, (30^*)_4, (120^*)_4)$. The equations define the membership of numbers in these four sets inductively. The union with $\{1\}$, $\{2\}$ and $\{3\}$ is the basis of the induction. Consider the first equation. The intersection $(X_1 + X_3) \cap (X_2 + X_2)$ represents each power of four $(10^n)_4$ in two ways: as $(10^{n-1})_4 + (30^{n-1})_4$ and as $(20^{n-1})_4 + (20^{n-1})_4$. Intersections of sums in the three remaining equations similarly express larger elements of the solution in terms of its smaller elements. No other numbers are produced by each of these expressions; for instance, in the first equation,

$$\left((10^*)_4 + (30^*)_4\right) \cap \left((20^*)_4 + (20^*)_4\right) = \left((10^+)_4 \cup (10^*30^*)_4 \cup (30^*10^*)_4\right) \cap \left((10^+)_4 \cup (20^*20^*)_4\right) = (10^+)_4,$$

that is, both sums contain some "garbage", yet the garbage in the sums is disjoint, and is accordingly filtered out by the intersection.

The following generalization of this example is known. Consider any set of natural numbers, which can be recognized by a finite automaton operating on base-k representations of numbers. Such sets are known in the literature as *automatic sets*, and they have been a subject of extensive combinatorial studies presented in a monograph by Allouche and Shallit [2]. As it turns out, every such set is representable by a resolved system of equations over sets of numbers.

Theorem B *(Jeż [10]). For every $k \geqslant 2$ and for every regular language $L \subseteq \Sigma_k^+$ there exists a resolved system of equation over sets of natural numbers, in variables $X, Y_1, \ldots, Y_n$, and using the operations of union, intersection and addition, as well as singleton constants, which has the unique solution $X = (L)_k$ and $Y_i = K_i$ for some $K_i \subseteq \mathbb{N}$. Given a finite automaton recognizing L, the system can be effectively constructed.*

Let the language L be recognized by deterministic finite automaton (DFA) M, which reads an input string backwards, from its right-most (least significant) digit to the left-most one. The equations constructed in the proof of Theorem B use variables $X_{i,j,q}$, where i and j are two base-k digits and q is a state of M. The solution of the constructed system is

$$X_{i,j,q} = \left\{(ijw)_k \;\middle|\; M \text{ reaches the state } q \in Q \text{ upon reading } w\right\}.$$

The equations express larger elements of these sets through their smaller elements as follows. Consider a number of the form $(ijw)_k$ in $X_{i,j,q}$, and let $\ell \in \Sigma_k$ be the first digit of w, so that $w = \ell w'$. When M reads w from right to left, let q' be the state it reaches after reading w', from where it proceeds to the state q by the digit ℓ. The number $(j\ell w')_k$ then belongs to the set $X_{j,\ell,q'}$, and it shall be transformed to the desired number $(ij\ell w')_k$, thus effectively concatenating a new digit i to the left of the base-k representation. This manipulation of leading digits is done using the techniques presented in Example 1: generally, by adding a set $(i0^*)_k$ and then intersecting several such representations to ensure that no garbage gets through. The equation for each variable $X_{i,j,q}$ expresses it as the union of such cases, for all appropriate $q' \in Q$ and $\ell \in \Sigma_k$, according to the transition function of the DFA. The details of the construction can be found in the cited paper [10], as well as in Lemma 12 later in this paper, where the construction will be remade using a different kind of equations.

A further extension of this construction allows representing any set of natural numbers, whose base-k representations are recognized by a *one-way real-time cellular automaton*, also known as a *trellis automaton*. A trellis automaton [7,24], defined as a quintuple $(\Sigma, Q, I, \delta, F)$, processes an input string of length $n \geqslant 1$ using a uniform array of $\frac{n(n+1)}{2}$ nodes, as presented in Fig. 1. Each node computes a value from a fixed finite set Q. The nodes in the bottom row obtain their values directly from the input symbols using a function $I\colon \Sigma \to Q$. The rest of the nodes compute the function $\delta\colon Q \times Q \to Q$ of the values in their predecessors. The string is accepted if and only if the value computed by the topmost node belongs to the set of accepting states $F \subseteq Q$.

Definition 1. A trellis automaton is a quintuple $M = (\Sigma, Q, I, \delta, F)$, in which:

- Σ is the input alphabet,
- Q is a finite non-empty set of states,
- $I\colon \Sigma \to Q$ is a function that sets the initial states,
- $\delta\colon Q \times Q \to Q$ is the transition function, and
- $F \subseteq Q$ is the set of accepting states.

The initial function I is extended to a length-preserving homomorphism $I\colon \Sigma^* \to Q^*$, which maps an input string to the bottom row of states, as in Fig. 1. Given the bottom row of states, the set computed at the top of the triangle is defined by the following extension of the transition function δ to the domain Q^+: let $\delta\colon Q^+ \to Q$ be defined by $\delta(q) = q$ and

$$\delta(q_1, \ldots, q_n) = \delta\big(\delta(q_1, \ldots, q_{n-1}), \delta(q_2, \ldots, q_n)\big).$$

For each state $q \in Q$, denote the set of strings, on which the automaton computes the state q, by $L_M(q) = \{w \mid \delta(I(w)) = q\}$. The language recognized by the automaton is the set of strings on which it computes an accepting state: $L(M) = \bigcup_{q \in F} L_M(q)$.

Trellis automata are known to be equivalent in power to resolved systems of language equations $X_i = \varphi_i(X_1, \ldots, X_n)$, where the right-hand sides use the operations of union, intersection and two-sided linear concatenation of the form $\{u\} \cdot X$ or $X \cdot \{u\}$, where $\{u\}$, with $u \in \Sigma^*$, is a singleton constant language and X may be any variable. These equations correspond to a class of formal grammars known as *linear conjunctive grammars* [22,24]. The family of languages recognized by trellis automata shall hence be called *linear conjunctive*.

In this paper, linear conjunctive languages are used for the same purpose as regular languages are in Theorem B. That is, a set of numbers will be defined by the language of its base-k representations, and when this language is linear conjunctive, one can construct a system of equations over sets of numbers that defines this set by its unique solution.

Theorem C *(Jeż and Okhotin [11]). For every $k \geqslant 2$ and for every linear conjunctive language $L \subseteq \Sigma_k^*$, with $L \cap 0\Sigma_k^* = \varnothing$, there exists a resolved system over sets of natural numbers in variables $X, Y_1, \ldots, Y_n$, with the union, intersection and addition operations and with singleton constants, which has the unique solution $X = (L)_k$ and $Y_i = K_i$ for some $K_i \subseteq \mathbb{N}$. Given a trellis automaton recognizing L, the system can be effectively constructed.*

The constructed system has a variable X_q for each state q of the trellis automaton M. Consider that the state computed by M on a string iwj, with $i, j \in \Sigma_k$ and $w \in \Sigma_k^*$, depends solely on the pair of states computed by M on the strings iw and wj. Ideally, the numbers $(iw)_k$ and $(wj)_k$ would be used to define the number $(iwj)_k$. However, the actual representation is more involved. A string $w \in \Sigma_k^*$ is represented by a set of numbers $(1w1)_k$, $(1w10)_k$, $(1w100)_k$, etc. The reason to have multiple representatives is that there is no apparent uniform way to obtain a number $(wj)_k$, with $w \in \Sigma_k^*$ and $j \in \Sigma_k$, from a number $(w)_k$ using addition. However, a number $(1wj10^\ell)_k$ with trailing zeroes may be obtained from another number $(1w10^{\ell+1})_k$ with one more trailing zero by adding a number of a simple form $((j-1)10^\ell)_k$. Such an addition consumes one trailing zero, and an unbounded supply of these zeroes is thus needed. The leading and ending digits 1 are used to mark the boundary of w: without these 1s, the representations of $0w0$ and w would be the same.

The actual equations constructed in Theorem C essentially use the large arsenal of constants provided by Theorem B. For example, the idealized addition $(1w10^{\ell+1})_k + ((j-1)10^\ell)_k = (1wj10^\ell)_k$ mentioned in the previous paragraph is in fact implemented as

$$\big(\big[\big(X_q + \big(10^*\big)_k\big) \cap \big(1\Sigma_k^* 20^*\big)_k\big] + \big((j-2)10^*\big)_k\big) \cap \big(1\Sigma_k^* j10^*\big)_k, \quad \text{for } j \geqslant 3.$$

Here, adding a constant set $(10^*)_k$ produces a number $(1w20^{\ell+1})_k$, as well as infinitely many other numbers, which are filtered out by the intersection with another constant $(1\Sigma_k^* 20^*)_k$. The subsequent addition of $((j-2)10^*)_k$ yields the desired number $(1wj10^\ell)_k$, and an intersection with the fourth constant removes all undesired sums. All four constants have regular base-k notation. Again, the detailed construction is given in the cited paper [11], and reconsidered in Lemma 13 in the following.

An important example of a set representable according to Theorem C is the numerical version of the set of computation histories of a given Turing machine, presented in Section 2. First, the symbols needed to define the standard language of computations of a Turing machine are interpreted as digits. Then, every string from this language represents a certain number, and this number encodes the computation history of the machine on one of the strings it accepts. Since the standard language of computations can be recognized by a trellis automaton, by Theorem C, there is a resolved system of equations that defines the set of their numerical representations.

In the next section, such a set of numbers will be used for the same purpose as the standard language VALC in the computational completeness proofs for language equations [23,25,26].

4. Unresolved systems with union, intersection and addition

Consider systems of equations of the form

$$\varphi_j(X_1,\ldots,X_n)=\psi_j(X_1,\ldots,X_n)\quad(1\leqslant j\leqslant m),$$

where the unknowns X_i are sets of natural numbers and the expressions φ_j, ψ_j on both sides may use union, intersection and addition operations, as well as singleton constants. The ultimate result of this paper is the computational completeness of such systems using only union *or* only intersection. The proof of that result is quite involved, but its most important ideas are already present in the proof of a weaker result, with the constructed systems using both union and intersection. This weaker result shall be established first.

Theorem 1. *The family of sets of natural numbers representable by unique (least, greatest) solutions of systems of equations of the form $\varphi_i(X_1,\ldots,X_n)=\psi_i(X_1,\ldots,X_n)$ with union, intersection and addition operations, and with the only constant $\{1\}$, is exactly the family of recursive sets (r.e. sets, co-r.e. sets, respectively). The transformation is effective in both directions.*

These solutions are recursive (r.e., co-r.e., respectively), because so are the solutions of language equations with union, intersection and concatenation, as asserted by Theorem A. So the task is to take any recursive (r.e., co-r.e.) set of numbers, given by a Turing machine recognizing it, and to construct a system of equations representing this set by a solution of the corresponding kind.

The construction for Theorem 1, presented in the rest of this section, is based upon an arithmetization of Turing machines, which proceeds in several stages. Given a (deterministic) Turing machine T, first, consider the set of its computation histories: for every number n accepted by the machine, the computation history $C_T(n)$ is a string over a fixed alphabet that lists all consecutive configurations in the accepted computation on n. These strings can be considered as strings of digits, which in turn represent natural numbers, so that every computation history is regarded as a number. The encoding of computation histories shall use base-6 positional notation, where 6 is the smallest base, for which the details of the constructions below could be conveniently implemented. Thus, $C_T(n)$ is defined as a string over the alphabet $\Sigma_6=\{0,1,2,3,4,5\}$, and the exact form of $C_T(n)$ shall be defined so that the language $\{C_T(n)\mid n\in L(T)\}\subseteq\Sigma_6^*$ of all such histories is recognized by a trellis automaton (the actual form of this language, given in Definition 3 below, includes a few extra details). Then, by Theorem C, the set of numerical representations of these strings of digits will be defined by a resolved system of equations over sets of numbers (see Lemma 2 below for details).

It remains to extract the numbers accepted by T out of its computational histories on these numbers. In the case of formal languages over multiple-symbol alphabets, this was done by a language equation (1) involving *concatenation of strings*. In order to do the same for computation histories represented as numbers, one should *concatenate their base-k representation*. In general, no such operation is available. However, for strings of digits of a certain specific form, such as those in Definitions 2 and 3, one can simulate their concatenation using addition of sets of numbers, by using the property established in Lemma 1. Using this form of concatenation, one can re-implement the language equation (1) that extracts the language of a Turing machine out of the set of its computation histories—now using numbers and their positional notation. This allows defining the set of numbers accepted by T as the least solution of a system of equations using union, intersection and addition, which will be done in Lemma 3. The subsequent Lemmata 4 and 5 establish the corresponding results for the representation of co-r.e. sets of numbers by greatest solutions, and of recursive sets by unique solutions.

The constructions leading to the proof of Theorem 1 begin with defining the encoding of computation histories as strings of digits, which will later allow representing them using trellis automata, as well as simulating their concatenation through the addition of numbers.

Definition 2. Let T be a Turing machine operating on natural numbers given to it in a base-6 notation. Let V with $\Sigma_6\subseteq V$ be its tape alphabet, let Q be its set of states, and define $\Gamma=V\cup Q\cup\{\sharp\}$. Let $L(T)\subseteq\mathbb{N}$ be the set of numbers accepted by T.

For every number $n\in L(T)$, let ℓ be the number of steps in the accepting computation of T on n. Denote the instantaneous description of T after i steps of computation on n by a string $ID_i=\alpha qa\beta\subseteq V^*QVV^*$, where T is in the state $q\in Q$ scanning the symbol $a\in\Gamma$, and the tape contains $\alpha a\beta$. Define the computation history of T on n as

$$\widetilde{C}_T(n)=(ID_0)^R\cdot\sharp\cdot(ID_1)^R\cdot\sharp\cdot\ldots\cdot\sharp\cdot(ID_{\ell-1})^R\cdot\sharp\cdot(ID_\ell)^R\cdot\sharp\cdot ID_\ell\cdot\sharp\cdot\ldots\cdot\sharp\cdot ID_1\cdot\sharp\cdot ID_0,$$

where α^R denotes the reversal of a string α.

Next, consider any code $h\colon\Gamma^*\to\Sigma_6^*$, under which every codeword is in $\{30,300\}^+$. Define $C_T(n)=h(\widetilde{C}_T(n))300$.

Now the set of accepting computations of a Turing machine is represented as the following six sets of numbers.

Definition 3. Let T be a Turing machine recognizing numbers given in base-6 notation. For every $i\in\{1,2,3,4,5\}$, the set of valid accepting computations of T on numbers $n\geqslant 6$ with their base-6 notation beginning with the digit i is

$$\mathrm{VALC}_i(T)=\left\{\left(C_T\left((iw)_6\right)1w\right)_6\;\middle|\;w\in\Sigma_6^+,\ (iw)_6\in L(T)\right\}.$$

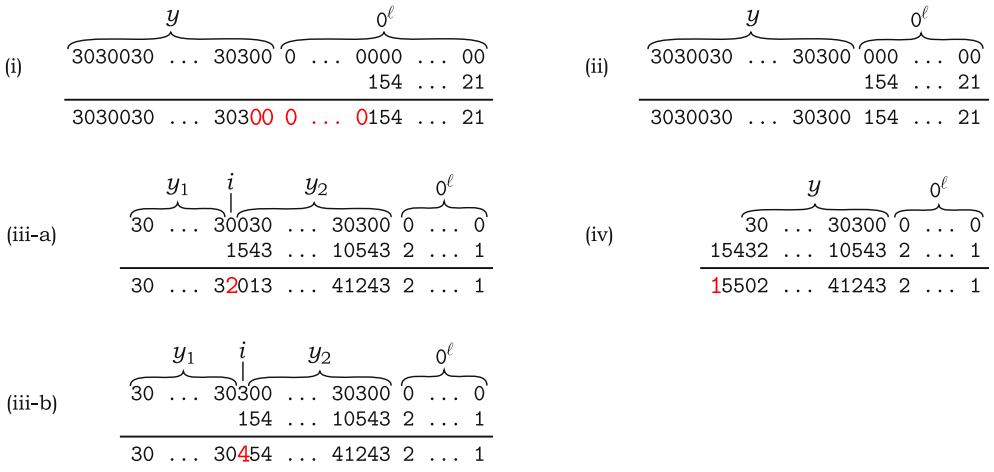

Fig. 2. Five cases of addition $(y0^{\ell})_6 + (1v)_6$ in Lemma 1.

All accepting computations of T on numbers $n \in \{0,1,2,3,4,5\}$, if any, are represented by the following finite set of numbers:

$$\mathrm{VALC}_0(T) = \left\{ \left(C_T(n)\right)_6 + n \;\middle|\; n \in \{0,1,2,3,4,5\} \text{ and } n \in L(T) \right\}.$$

For example, under this encoding, the accepting computation on a number $n = (543210)_6$ will be represented by a number $(30300300\ldots30300143210)_6 \in \mathrm{VALC}_5(T)$, where the whole sequence of the Turing machine's configurations is encoded by blocks of digits 30 and 300 used as bits, the digit 1 separates the computation history from the input number, and the lowest digits 43210 represent n with its leading digit 5 cut off.

The computation of T on an input number $n = (iw)_6 \in L(T)$ is encoded as the number $(x1w)_6$, with $x = C_T((iw)_6)$, and this number is obviously representable as a sum of $(1w)_6$ and an appropriate number in $(\{30,300\}^*3000^*)_6$. The crucial property of this encoding is that the converse statement holds as well: that is, whenever the sum of a number $(1w)_6$ and any number in $(\{30,300\}^*3000^*)_6$ is of the form $(x1u)_6$, with $x \in \{30,300\}^*300$ and $u \in \Sigma_6^*$, the string u must be equal to w. The following lemma rules out the hypothetical possibility that the number $(x1u)_6$ could be obtained in any other way.

Lemma 1. *Let $S \subseteq (1\Sigma_6^+)_6$. Then, for all strings $x \in \{30,300\}^*300$ and $u \in \Sigma_6^+$, if $(x1u)_6 \in (\{30,300\}^*3000^*)_6 + S$, then $(1u)_6 \in S$.*

Proof. Let $(x1u)_6 = (y0^{\ell})_6 + (1v)_6$, where $y \in \{30,300\}^*300$, $\ell \geqslant 0$ and $(1v)_6 \in S$. The goal is to show that $u = v$, $x = y$ and $|1v| = \ell$, that is, the only way to obtain $(x1u)_6$ is by adding $(x0^{|u|+1})_6$ to $(1u)_6$, and no other pairs of numbers from $(\{30,300\}^*3000^*)_6$ and from S could be summed to fake this number.

The proof is by the analysis of all cases of addition. In each "improper" case it will be shown that the base-6 notation of $(y0^{\ell})_6 + (1v)_6$ has a different structure than $x1u$. Depending on the number of digits in $|1v|$, consider the following cases, illustrated in Fig. 2:

i. If $|1v| < \ell$, then $(y0^{\ell})_6 + (1v)_6 = (y0^{\ell-|1v|}1v)_6$, which is a number with a base-6 notation containing at least three consecutive zeroes to the left of the leftmost digit 1, as shown in Fig. 2(i). Since $(x1u)_6$ has only two zeroes to the left of the leftmost 1, it follows that $(y0^{\ell})_6 + (1v)_6 \neq (x1u)_6$, which makes this case impossible.

ii. Let $|1v| = \ell$. This is the case of addition done as intended, illustrated in Fig. 2(ii). Then $(y0^{\ell})_6 + (1v)_6 = (y1v)_6$, and thus $(y1v)_6 = (x1u)_6$. The leftmost instance of 1 in $(y1v)_6$ and in $(x1u)_6$ is at the first position of $1v$ and $1u$, respectively. Therefore, $y = x$ and $1v = 1u$.

iii. Suppose that $\ell < |1v| \leqslant |y| + \ell$. Then the leading 1 from $1v$ is at the same position as some digit of y in $(y0^{\ell})_6$. Let $y = y_1 i y_2$, where $|y_2| + \ell = |v|$. The digit i is either 0 or 3.

   (a) If $i = 0$, as in Fig. 2(iii-a), then y_1 ends with 3 or 30. The sum $(y_1 i y_2 0^{\ell})_6 + (1v)_6$ is thus of the form $(y_1 i' z)_6$, where $i' \in \{1,2\}$ (a digit 2 can appear due to a possible carry from the earlier position), and the prefix $y_1 i'$ is in $\{30,300\}^*\{31,32,301,302\}$. On the other hand, in $(x1u)_6$, the leftmost occurrence of digits other than 0 or 3 must be of the form 3001.

   (b) If $i = 3$, as in Fig. 2(iii-b), then the sum $(y_1 i y_2 0^{\ell})_6 + (1v)_6$ is of the form $(y_1 i' z)_6$, where $|z| = |v|$ and $i' \in \{4,5\}$ (again, 5 can appear due to a possible carry). Consider the leftmost digits of the numbers $(y_1 i' z)_6$ and $(x1u)_6$ different from 0 and 3. For $(x1u)_6$ it is 1, while for $(y_1 i' z)_6$ it is 4 or 5, and thus these numbers cannot be the same.

In both cases it follows that $(y_1 i y_2 0^\ell)_6 + (1v)_6$ and $(x1u)_6$ must be different, and the case is impossible.

iv. Consider the last possibility, $|1v| > |y| + \ell$. Then the leading digit of $(y0^\ell)_6 + (1v)_6$ is 1 or 2 (due to a possible carry), as shown in Fig. 2(iv). As the leading digits are different, $(y0^\ell)_6 + (1v)_6 \neq (x1u)_6$, which rules out this case.

It has thus been established that $y = x$ and $1v = 1u$ is the only possible case, which proves the claim. □

In order to use the set of numbers $\mathrm{VALC}_i(T)$ in equations, one should first express it using a separate system of equations with union, intersection and addition. This is achieved in the lemma below, using a trellis automaton recognizing base-6 notation of numbers in $\mathrm{VALC}_i(T)$.

Lemma 2. *For every Turing machine T operating on numbers given in base-6 notation, there exists and can be effectively constructed a system of equations*

$$\varphi_j(Y_0, \ldots, Y_5, X_1, \ldots, X_m) = \psi_j(Y_0, \ldots, Y_5, X_1, \ldots, X_m)$$

over sets of natural numbers, using union, intersection, addition and constant {1}*, which has a unique solution* $Y_0 = \mathrm{VALC}_0(T)$, …, $Y_5 = \mathrm{VALC}_5(T)$, $X_1 = S_1, \ldots, X_m = S_m$*, for some sets* $S_1, \ldots, S_m \subseteq \mathbb{N}$.

Proof. For each $i \in \{1, \ldots, 5\}$, it is claimed that the language $\{C_T(n)1w \mid n \in (i\Sigma_6^+)_6,\ n \in L(T)\} \subseteq \Sigma_6^*$ of base-6 representations of the numbers in $\mathrm{VALC}_i(T)$ is linear conjunctive.

First, consider the language $L_i = \{\widetilde{C}_T(n) \mid n \in (i\Sigma_6^+)_6,\ n \in L(T)\}$ of computation histories, defined over the alphabet of symbols used by the Turing machine. As defined in this paper, L_i is only marginally different from another form of the language of computation histories considered in the literature [28, Lemma 3], and hence is representable as an intersection of two linear context-free languages in exactly the same way as the known language. Hence, it is linear conjunctive.

The homomorphic image of L_i over the alphabet of base-6 digits, as per Definition 2, is $h(L_i) = \{C_T(n) \mid n \in (i\Sigma_6^+)_6,\ n \in L(T)\} \subseteq \{30, 300\}^*$. This language is linear conjunctive, by the known closure of the linear conjunctive languages under codes [27].

The last step is appending the base-6 notation of the number, with its leading digit truncated, to the end of the computation history, leading to the desired language

$$L_i' = \left\{ C_T(n)1w \;\middle|\; n \in (i\Sigma_6^+)_6,\ n \in L(T) \right\} \subseteq \{30, 300\}^* 1 \Sigma_6^+.$$

Given a trellis automaton recognizing $h(L_i)$, it is an exercise to modify it to recognize L_i'. To be precise, one would have to match the suffix $1w$ to the initial tape contents of the Turing machine, which is written in the beginning of $C_T(n)$, in reverse and encoded by h, as $h((ID_0)^R) = h(w^R i q_0)$, where q_0 is the machine's initial state. As this is a variant of the language of palindromes, a trellis automaton is perfectly capable of recognizing it.

The set of numbers $(L_i')_6$ corresponding to this language is exactly $\mathrm{VALC}_i(T)$, and, by Theorem C, it is defined by a system of equations of the desired form. All steps of this transformation are effective, and thus the resulting system of language equations is effectively produced from a given Turing machine T.

It remains to consider the last set $\mathrm{VALC}_0(T) = \{(C_T(n))_6 + n \mid n \in \{0, 1, 2, 3, 4, 5\} \text{ and } n \in L(T)\}$, which contains up to 6 numbers: the computation histories of T on any numbers $n \in \{0, \ldots, 5\}$, provided that they are accepted. Though this is a finite set, representing it effectively requires the same approach as for $\mathrm{VALC}_i(T)$ with $i > 0$. For each number $n \in \{0, \ldots, 5\}$, since the string of digits $C_T(n)$ ends with 0 and n is less than 6, the base-6 notation of $(C_T(n))_6 + n$ is obtained from $C_T(n)$ by deleting the trailing 0 and appending n as a single digit. Now a trellis automaton recognizing valid computation histories can be modified to accept exactly these strings of digits. Then an application of Theorem C yields the system of equations that defines the corresponding set of numbers $\mathrm{VALC}_0(T)$. □

Later in Section 5, the same result will be proved in a stronger form (Lemma 2*) using equations with addition and only one of the two Boolean operations: either only union, or only intersection.

The next task is to use these sets of numbers as constants, in order to construct equations defining the set $L(T)$ of numbers accepted by the Turing machine. The first case to be established is the case of least solutions and r.e. sets.

Lemma 3. *For every Turing machine T recognizing a set $L(T) \subseteq \mathbb{N}$, there exists a system of equations of the form*

$$\varphi_j(Y, X_1, \ldots, X_m) = \psi_j(Y, X_1, \ldots, X_m)$$

with union, intersection, addition and constant {1}*, which has the set of solutions*

$$\left\{ \left(S, f_1(S), \ldots, f_m(S)\right) \;\middle|\; L(T) \subseteq S \right\},$$

where $f_1, \ldots, f_m \colon 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ *are some monotone functions on sets of numbers, which depend on T. In particular, there is a least solution with $Y = L(T)$. Given T, the system can be effectively constructed.*

Each function f_i represents the dependence of the X_i-component of a solution on the Y-component. The assumption that the functions $f_1, \ldots, f_m$ are monotone is needed to guarantee that the system has a least solution. Otherwise, if some f_i is non-monotone, then, for some $S' \supset S_0$, it could happen that $f_i(S') \subset f_i(S_0)$, so that the vectors $(S', f_1(S'), \ldots, f_m(S'))$ and $(S_0, f_1(S_0), \ldots, f_m(S_0))$ are incomparable.

Proof. The proof is by constructing a system in variables $(Y, Y_1, \ldots, Y_5, Y_0, X_7, \ldots, X_m)$, where the number m shall be determined below, and the set of solutions of this system is defined by the following conditions (2a)–(2c), which ensure that the statement of the lemma is fulfilled. The first condition is that every number from 0 to 5 accepted by the Turing machine must therefore be in Y_0; at the same time, no numbers other than $\{0, 1, 2, 3, 4, 5\}$ may be in Y_0, which puts Y_0 between the following bounds:

$$L(T) \cap \{0, 1, 2, 3, 4, 5\} \subseteq Y_0 \subseteq \{0, 1, 2, 3, 4, 5\}. \tag{2a}$$

For each number greater than 5 accepted by T, let $(iw)_6$ be its base-6 notation, where $i \in \{1, \ldots, 5\}$ is the leading digit. Then the corresponding number $(1w)_6$, with the leading digit replaced with 1, should belong to the variable Y_i; at the same time, Y_i is restricted to contain only numbers with the base-6 notation beginning with 1:

$$\left\{(1w)_6 \;\middle|\; w \in \Sigma_6^+,\ (iw)_6 \in L(T)\right\} \subseteq Y_i \subseteq \left(1\Sigma_6^+\right)_6 \quad (1 \leqslant i \leqslant 5). \tag{2b}$$

The variable Y represents the union of all sets encoded in $Y_0, Y_1, \ldots, Y_5$, with their leading digits converted back to normal:

$$Y = Y_0 \cup \bigcup_{i=1}^{5} \left\{(iw)_6 \;\middle|\; (1w)_6 \in Y_i\right\}. \tag{2c}$$

The remaining variables are

$$X_j = K_j \quad (7 \leqslant j \leqslant m) \tag{2d}$$

where the sets $K_7, \ldots, K_m \subseteq \mathbb{N}$ are some constants needed for the construction to work. These constants and the equations needed to specify them will be implicitly obtained in the proof. The constructed system will use inequalities of the form $\varphi \subseteq \psi$, which can be equivalently rewritten as equations $\varphi \cup \psi = \psi$ or $\varphi \cap \psi = \varphi$.

For each leading digit $i \in \{1, 2, 3, 4, 5\}$, consider the set $\mathrm{VALC}_i(T)$, as in Definition 3, which can be represented by equations according to Lemma 2. Define the following two inequalities that set the bounds on the variable Y_i:

$$Y_i \subseteq \left(1\Sigma_6^+\right)_6, \tag{3a}$$

$$\mathrm{VALC}_i(T) \subseteq \left(\{30, 300\}^*3000^*\right)_6 + Y_i. \tag{3b}$$

Both constants in the right-hand sides are given by regular languages of base-6 representations, and therefore can be defined by equations according to Theorem B. It is claimed that this system is equivalent to the condition (2b).

Let (2b) hold for Y_i. Then (3a) immediately follows. To check (3b), consider any number $(C_T((iw)_6)1w)_6 \in \mathrm{VALC}_i(T)$. Since this number represents the accepting computation of T on $(iw)_6$, this implies that $(iw)_6 \in L(T)$, and hence $(1w)_6 \in Y_i$ by (2b). Then $(C_T((iw)_6)1w)_6 \in (\{30, 300\}^*3000^{|1w|})_6 + (1w)_6 \subseteq (\{30, 300\}^*3000)_6 + Y_i$, which proves the inclusion (3b).

Conversely, assuming (3), it has to be proved that for every number $(iw)_6 \in L(T)$, where $w \in \Sigma_6^+$, the modified number $(1w)_6$ is in Y_i. Since $(iw)_6 \in L(T)$, there exists an accepting computation of T represented by the number $(C_T((iw)_6)1w)_6 \in \mathrm{VALC}_i(T)$. Hence, $(C_T((iw)_6)1w)_6 \in (\{30, 300\}^*3000^*)_6 + Y_i$ due to the inclusion (3b), and therefore $(1w)_6 \in Y_i$ by Lemma 1.

Define one more variable Y_0, which shall take care of the six numbers $\{0, 1, \ldots, 5\}$. With the knowledge of which of these numbers is accepted by T, the desired condition (2a) could be expressed by explicitly listing the accepted numbers. However, as it is undecidable whether T accepts any of these numbers, in order to make the construction effective, one should follow the same approach with computation histories. Define the equations

$$Y_0 \subseteq \{0, 1, 2, 3, 4, 5\}, \tag{4a}$$

$$\mathrm{VALC}_0(T) \subseteq \left(\{30, 300\}^*300\right)_6 + Y_0. \tag{4b}$$

The claim is that these inequalities (4) hold if and only if the condition (2a) is satisfied.

Assume (2a) and consider any number $(C_T(n))_6 + n \in \mathrm{VALC}_0(T)$, where $n \in \{0, 1, 2, 3, 4, 5\}$. Then n is accepted by T, and, by (2a), $n \in Y_0$. Hence, $(C_T(w))_6 + n \in (\{30, 300\}^*300)_6 + n \subseteq (\{30, 300\}^*300)_6 + Y_0$, which proves (4b).

The converse claim is that (4) implies that every number $n \in L(T) \cap \{0, 1, 2, 3, 4, 5\}$ must be in Y_0. For any such number n, the corresponding computation $(C_T(n))_6 + n \in \mathrm{VALC}_0(T)$ ends with the digit n, because $C_T(n)$ ends with a zero. By the inequality (4b), $(C_T(n))_6 + n$ is in Y_0, and hence Y_0 must contain a number ending with the digit n. Since every number in Y_0 is one-digit by (4a), this can only be n, that is, $n \in Y_0$.

Putting the above six systems together, the variables $Y_0, Y_1, \ldots, Y_5$ define all the desired numbers, yet the numbers in Y_2, Y_3, Y_4, Y_5 have their leading digits replaced with 1. In order to change them back according to condition (2c), add a new variable Y with the following equation:

$$Y = Y_0 \cup Y_1 \cup \bigcup_{\substack{i\in\{2,3,4,5\}\\ i'\in\Sigma_6}} \left(\left[\left(Y_i \cap \left(1i'\Sigma_6^*\right)_6\right) + \left((i-1)0^*\right)_6\right] \cap \left(ii'\Sigma_6^*\right)_6\right). \tag{5}$$

The variables Y_0 and Y_1 are used as they are. Each variable Y_i, with $i \geqslant 2$, is used in six subexpressions of the form $[(Y_i \cap (1i'\Sigma_6^*)_6) + ((i-1)0^*)_6] \cap (ii'\Sigma_6^*)_6$, corresponding to all possible second digits $i' \in \Sigma_k$. The purpose of this subexpression is to replace the leading digit 1 with the desired leading digit i in all numbers from Y_i that have the two leading digits $1i'$. The transformation is carried out by the usual method of first adding a constant set, and then filtering out sums of an undesired form using an intersection with another constant set. The equation (5) is actually borrowed from the authors' previous paper [11, Lem. 7], where it was proved equivalent to $Y = Y_0 \cup \{(iw)_6 \mid (1w)_6 \in Y_i\}$, that is, to (2c). Hence, a proof is omitted here.

Note that this is the only equation in this proof that uses explicit union or intersection. It will be shown later, in Lemmata 15–16, that this equation can be equivalently represented using only one Boolean operation. This will subsequently allow establishing a stronger form of Lemma 3.

The final step of the construction is to express the constants with regular base-6 notation used in the above systems through constant $\{1\}$, which can be done by Theorem B and Lemma 2. The variables needed to define these languages are denoted by $X_7, \ldots, X_n$, and the equations for these variables have a unique solution $X_j = K_j$ for all j.

This proves that the set of solutions of the system (3)–(5) is exactly as given by conditions (2a)–(2c). The conditions can be equivalently reformulated to match the statement of the lemma by presenting the lower bounds on $Y_0, Y_1, \ldots, Y_5$ as lower bounds on Y

$$L(T) \subseteq Y,$$

and by expressing all other variables as functions of Y as follows:

$$\begin{aligned} Y_0 &= Y \cap \{0,1,2,3,4,5\},\\ Y_i &= \left\{(1w)_6 \;\middle|\; w \in \Sigma_6^+,\ (iw)_6 \in Y\right\},\\ X_j &= K_j. \end{aligned}$$

Then, in particular, there is a least solution in this set, with $Y = L(T)$, $Y_0 = L(T) \cap \{0,1,2,3,4,5\}$, $Y_i = \{(1w)_6 \mid w \in \Sigma_6^+, (iw)_6 \in L(T)\}$ and $X_j = K_j$. □

The representation of co-recursively enumerable sets by greatest solutions is dual to the case of least solutions and is established by an analogous argument, as shown below.

Denote the complements of the languages $\mathrm{VALC}_i(T)$ $(0 \leqslant i \leqslant 5)$ by $\mathrm{INVALC}_i(T)$. Base-6 notations of numbers in these sets are recognized by trellis automata, due to the closure of trellis automata under complementation. Therefore, analogously to Lemma 2, the sets $\mathrm{INVALC}_i(T)$ are representable by equations.

Lemma 4. *For every Turing machine T recognizing a recursively enumerable set of numbers $L(T) \subseteq \mathbb{N}$, there exists a system of equations of the form*

$$\varphi_j(Z, X_1, \ldots, X_m) = \psi_j(Z, X_1, \ldots, X_m)$$

with union, intersection and addition, and constant $\{1\}$, which has the set of solutions

$$\left\{\left(S, f_1(S), \ldots, f_m(S)\right) \;\middle|\; S \subseteq \overline{L(T)}\right\},$$

where $f_1, \ldots, f_m : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ are some monotone functions on sets of numbers, which depend on T. In particular, there is a greatest solution with $Z = \overline{L(T)}$. Given the Turing machine, the system can be effectively constructed.

The functions $f_1, \ldots, f_m$ have to be monotone for the system to have a greatest solution. Otherwise, as explained along with Lemma 3, there could be incomparable maximal solutions.

Proof. The system uses the variables $Z, Z_1, \ldots, Z_5, Z_0, X_7, \ldots, X_m$, and its set of solutions will be characterized by the following conditions, cf. conditions (2a)–(2d) in the proof of Lemma 3. First, the variable Z_0 may only contain numbers from 0 to 5, and if such a number is in Z_0, then it is *not accepted* by T:

$$Z_0 \subseteq \overline{L(T)} \cap \{0,1,2,3,4,5\}. \tag{6a}$$

Each variable Z_i contains only numbers of the form $(1w)_6$, with $w \in \Sigma_6^+$, and if such a number belongs to Z_i, then the number $(iw)_6$, with the leading digit modified, is *not accepted* by T:

$$Z_i \subseteq \left\{(1w)_6 \;\middle|\; w \in \Sigma_6^+,\ (iw)_6 \notin L(T)\right\} \quad (1 \leqslant i \leqslant 5). \tag{6b}$$

The variable Z represents the union of the sets encoded in $Z_0, Z_1, \ldots, Z_5$:

$$Z = Z_0 \cup \bigcup_{i=1}^{5} \left\{(iw)_6 \;\middle|\; (1w)_6 \in Z_i\right\}. \tag{6c}$$

The remaining variables define some fixed sets:

$$X_j = K_j \quad (7 \leqslant j \leqslant m), \tag{6d}$$

where the number m and the sets $K_7, \ldots, K_m$ will be determined below.

The equations defining the value of each Z_i, with $1 \leqslant i \leqslant 5$, are as follows:

$$Z_i \subseteq \left(1\Sigma_6^+\right)_6, \tag{7a}$$

$$\left(\{30, 300\}^*3000^*\right)_6 + Z_i \subseteq \mathrm{INVALC}_i(T). \tag{7b}$$

It is claimed that (7) holds if and only if the condition (6b) is satisfied.

If Z_i satisfies (6b), then (7a) follows immediately, and in order to verify (7b), one has to consider any number not in $\mathrm{INVALC}_i(T)$ and show that it is not in $(\{30, 300\}^*3000^*)_6 + Z_i$ either. By definition, a number is not in $\mathrm{INVALC}_i(T)$ if and only if it is in $\mathrm{VALC}_i(T)$, and every number in $\mathrm{VALC}_i(T)$ represents the computation history of T on some number $n = (iw)_6 \in L(T)$, and accordingly is of the form $(C_T(n)1w)_6 \in \mathrm{VALC}_i(T)$, with $C_T((iw)_6) \in \{30, 300\}^*300$. Suppose that $(C_T((iw)_6)1w)_6 \in (\{30, 300\}^*3000^*)_6 + Z_i$. Then, by Lemma 1, $(1w)_6 \in Z_i$, and hence $(iw)_6 \notin L(T)$ by (6b), which yields a contradiction.

The converse is established as follows. Assuming (7), suppose that (6b) is not satisfied, that is, there exists a number $n = (1w)_6 \in Z_i$, with $i \in \{1, 2, 3, 4, 5\}$ and $w \in \Sigma_6^+$, for which $(iw)_6 \in L(T)$. Then $(C_T(n)1w)_6 \in (\{30, 300\}^*3000^*)_6 + Z_i \subseteq \mathrm{INVALC}_i(T)$ by (7b). However, $(C_T(n)1w)_6$ is in $\mathrm{VALC}_i(T)$ and thus cannot be in $\mathrm{INVALC}_i(T)$. The contradiction obtained proves this case.

Next, define the following equations for the variable Z_0:

$$Z_0 \subseteq \{0, 1, 2, 3, 4, 5\}, \tag{8a}$$

$$\left(\{30, 300\}^*300\right)_6 + Z_0 \subseteq \mathrm{INVALC}_0(T). \tag{8b}$$

Again, the claim is that these equations are equivalent to (6a).

Let Z_0 be a subset of $\{0, 1, 2, 3, 4, 5\} \setminus L(T)$, as stated in (6a). This immediately implies (8a). Consider any number not in $\mathrm{INVALC}_0(T)$; proving that it is not in $(\{30, 300\}^*300)_6 + Z_0$ will establish (8b). A number not in $\mathrm{INVALC}_0(T)$ must be in $\mathrm{VALC}_0(T)$, so let $C_T(n) + n \in \mathrm{VALC}_0(T)$ for any $n \in \{0, 1, 2, 3, 4, 5\}$, and suppose, for the sake of a contradiction, that $C_T(n)+n \in (\{30, 300\}^*300)_6 + Z_0$. By (8a), the number n is represented by a single digit, and thus the last digit of $C_T(n)+n$ is n. For a number ending with n to be in $(\{30, 300\}^*300)_6 + Z_0$, the number n must be in Z_0. Therefore, by (6a), $n \notin L(T)$, which contradicts the accepting computation $C_T(n)$.

Conversely, assume (8) and suppose that there exists a number $n \in \{0, 1, 2, 3, 4, 5\}$, which is at the same time in $L(T)$ and in Z_0. Then there exists an accepting computation $C_T(n) + n \in \mathrm{VALC}_0(T)$, that is, $C_T(n) + n \notin \mathrm{INVALC}_0(T)$. However, $C_T(n) + n \in (\{30, 300\}^*300)_6 + Z_0$, because $C_T(n) \in (\{30, 300\}^*300)_6$ and $n \in Z_0$ by assumption, which contradicts (8b). The contradiction obtained proves that no such n exists, which establishes (6a).

The equation for Z is the same as the equation for Y in Lemma 3:

$$Z = Z_0 \cup Z_1 \cup \bigcup_{\substack{i \in \{2,3,4,5\} \\ i' \in \Sigma_6}} \left(\left[\left(Z_i \cap \left(1i'\Sigma_6^*\right)_6\right) + \left((i-1)0^*\right)_6\right] \cap \left(ii'\Sigma_6^*\right)_6\right). \tag{9}$$

As in Lemma 3, this equation is equivalent to (6c). Again, this is the only equation using union and intersection, which will be replaced by simpler equations later in Lemmata 15–16.

To conclude the construction, linear conjunctive constants in the system of equations (7)–(9) are expressed according to Theorem C, using extra variables $X_7, \ldots, X_n$. This is the system of the form promised in this lemma, and its set of solutions is described by the above conditions (6a)–(6d). These conditions can be reformulated as follows:

$$Z \subseteq \overline{L(T)},$$

$$Z_0 = Z \cap \{0, 1, 2, 3, 4, 5\},$$

$$Z_i = \{(1w)_6 \mid w \in \Sigma_6^+,\ (iw)_6 \in Z\},$$
$$X_j = K_j.$$

Then the greatest of these solutions is $Z_0 = \overline{L(T)} \cap \{0,1,2,3,4,5\}$, $Z_i = \{(1w)_6 \mid w \in \Sigma_6^+,\ (iw)_6 \in \overline{L(T)}\}$, $Z = \overline{L(T)}$. □

Finally, the result on representing recursive languages by unique solutions can be established by combining the constructions in Lemmata 3 and 4 as follows.

Lemma 5. *For every Turing machine T that halts on every input and recognizes a recursive set of numbers $S \subseteq \mathbb{N}$, there exists a system of equations of the form $\varphi_i(Y, Z, X_1, \ldots, X_n) = \psi_i(Y, Z, X_1, \ldots, X_n)$, with union, intersection, addition and constant $\{1\}$, such that its unique solution is $Y = Z = S$, $X_i = K_i$, where $K_1, \ldots, K_n$ are some sets. Given T, the system can be effectively constructed; if the given machine does not halt on some input, then the result of the construction is undefined.*

Proof. Define a second Turing machine T' that makes the same transitions as T, but accepts whenever T rejects, and rejects whenever T accepts. This machine T' halts on every input and recognizes the set $\overline{S}$. Then, Lemma 3 is applied to T and Lemma 4 is applied to T'. Consider both systems of language equations given by these lemmata, let Y be the variable from Lemma 3, let Z be the variable from Lemma 4, and let $X_1, \ldots, X_n$ be the rest of the variables in these systems put together. Each of these variables functionally depends on Y or on Z, and hence the set of solutions of the system obtained can be represented as

$$\{(Y, Z, f_1(Y,Z), \ldots, f_n(Y,Z)) \mid S \subseteq Y \text{ and } Z \subseteq \overline{\overline{S}}\},$$

or, equivalently, as

$$\{(Y, Z, f_1(Y,Z), \ldots, f_n(Y,Z)) \mid Z \subseteq S \subseteq Y\}.$$

All that matters here is the functional dependence on Y and Z; the monotonicity of this dependence, established in Lemmata 3 and 4, is not relevant in this argument.

Adding one more equation

$$Y = Z$$

to the system collapses the bounds $Z \subseteq S \subseteq Y$ to an equality $Z = S = Y$, and the resulting system has the unique solution

$$(S, S, f_1(S,S), \ldots, f_n(S,S)),$$

which completes the proof. □

The proof of Theorem 1 now proceeds as follows. The sets representable by unique, least and greatest solutions of equations over sets of numbers with union, intersection and addition are known to be recursive, r.e. and co-r.e., respectively, in the more general case of language equations, presented in Theorem A. Every r.e. set is defined by a least solution of some system of equations according to Lemma 3, every co-r.e. set can be defined by a greatest solution by Lemma 4, and Lemma 5 asserts that every recursive set is definable by a unique solution, Q.E.D.

5. Unresolved systems with one Boolean operation and addition

All results so far have been established for equations with addition, union and intersection. In fact, the same results hold for equations using addition and only one of these Boolean operations (that is, either *only union* or *only intersection*), which are thereby already computationally complete. In this section, all constructions in Sections 3–4 are re-implemented, leading to the following stronger result.

Theorem 1*. *The family of sets of natural numbers representable by unique (least, greatest) solutions of systems of equations of the form $\varphi_i(X_1, \ldots, X_n) = \psi_i(X_1, \ldots, X_n)$, with union and addition operations and constant $\{1\}$, is exactly the family of recursive sets (r.e., co-r.e., respectively). The transformation is effective in both directions.*

The same result holds for systems with intersection and addition.

While the proof of the earlier similar result (Theorem 1) for equations with union, intersection and addition could rely on the previously known expressive power of those systems (Theorems B and C), no such results are known for systems using only one Boolean operation. In this section, the expressive power of such systems is gradually built from scratch.

The first step is simple: one has to learn how to represent finite and co-finite sets using only constant $\{1\}$, which is achieved by a simple direct construction in Section 5.1. Next, the plan is to construct equations defining all sets with a regular base-k representation, and then all sets with a linear conjunctive base-k representation. However, rather than

reinventing the whole construction, Section 5.2 presents a systematic transformation of the existing constructions using union and intersection (Theorems B and C) into the new constructions employing a single Boolean operation (Theorems 2 and 3, respectively). Generally, the transformation works by replacing an intersection $Z = X \cap Y$ with inclusions $Z \subseteq X$ and $Z \subseteq Y$. This allows Z to be any subset of $X \cap Y$; to prevent Z from having too few elements, the transformation introduces constraints of the form "the union of several such variables Z is a constant set", where the constant is represented separately. Though this approach would not work in every case, it is applicable to the systems of equations used in Theorems B and C.

The transformation in Section 5.2 shall be used three times. At the first time, in Section 5.3, all available constants are finite and co-finite sets, and they are used to define every set of the form $(ij0^*)_k$, with $i, j \in \{0, \ldots, k-1\}$ and $i \neq 0$, by translating a known resolved system with union and intersection defining such a set [10]. The second use of this transformation, also in Section 5.3, employs the recently obtained sets of the form $(ij0^*)_k$ as constant sets, to translate the equations from Theorem B defining every set with a regular base-k notation. Finally, in Section 5.4, constant sets with a regular base-k notation are used to define every set with linear conjunctive base-k notation by translating the equations in Theorem C.

In the end, the proof of Theorem 1* follows the general line of that of Theorem 1, using the updated results on representing constant languages. Furthermore, the constructions of Lemmata 3–4 have to undergo some changes, with a single equation re-implemented using only one Boolean operation; this final step of the construction is presented in Section 5.5.

5.1. Representing finite and co-finite sets

The first result on the expressive power of equations with one Boolean operation asserts the representability of all finite and co-finite sets of numbers, using only constant $\{1\}$. For equations with union and addition, this is an obvious fact, because all finite and co-finite sets are ultimately periodic, and hence can be represented by simulating a finite automaton. Equations with intersection and addition require a new construction given below.

Lemma 6. *Every finite or co-finite subset of $\mathbb{N}$ is representable by a unique solution of a resolved system with union and addition using constant $\{1\}$, as well as by a unique solution of an unresolved system with intersection, addition and constant $\{1\}$. The system can be effectively constructed.*

Proof. First, every singleton constant $\{n\}$ with $n \geqslant 0$ can be defined using constant $\{1\}$ as follows: for $n > 1$, the set $\{n\}$ equals the sum $\{1\} + \{1\} + \ldots + \{1\}$, and $\{0\}$ is the unique solution of the equation $\{1\} + X = \{1\}$.

In the case of union, every ultimately periodic set $S \subseteq \mathbb{N}$ with period p beginning from n_0 is represented by the following resolved system, where Y defines all multiples of p, and X uses them to express S.

$$\begin{cases} X = \bigcup\limits_{\substack{n<n_0 \\ n\in S}} \{n\} \cup \bigcup\limits_{\substack{n_0\leqslant n<n_0+p \\ n\in S}} \big(\{n\} + Y\big) \\ Y = \big(\{p\} + Y\big) \cup \{0\} \end{cases}$$

Since all finite and co-finite sets are ultimately periodic with period $p = 1$, this proves the lemma in the case of union.

Consider the case of intersection, where the use of unresolved equations becomes essential. These equations shall be presented as inequalities $\varphi \subseteq \psi$, which can be reformulated as equations $\varphi = \varphi \cap \psi$. Let $K = \{n_1, n_2, \ldots, n_m\}$, with $0 \leqslant n_1 < \ldots < n_m$, be any finite set of numbers. First define the following equations for a variable X:

$$X + \{1\} \subseteq X \tag{10a}$$

$$\{n_m + 1\} \subseteq X \tag{10b}$$

$$\{n_m\} \cap X = \varnothing \tag{10c}$$

Here the first inequality (10a) ensures that the solution is of the form $\{n \mid n \geqslant k\}$ for some k (or is empty), the second inequality (10b) states that $n_m + 1$ is in X, while the equality (10c) ensures that n_m is not in X. Thus, the unique solution of these equations is $X = \{n \mid n > n_m\}$. Using this variable, define three more equations for a new variable Y:

$$X \cap Y = \varnothing \tag{10d}$$

$$\{n_i\} \subseteq Y, \qquad \text{for } i \in \{1, 2, \ldots, m\} \tag{10e}$$

$$\{n\} \cap Y = \varnothing, \quad \text{for each } n < n_m \text{ with } n \notin K \tag{10f}$$

By the first equation (10d), Y must be a subset of $\{0, \ldots, n_m\}$. The remaining equations fix the membership of every number between 0 and n_m in Y: it should be in Y if and only if it is in K. Hence, the unique solution is $Y = K$. Finally, define one more variable Z, with the following equations:

$$X \subseteq Z \tag{10g}$$

$$\{n_i\} \cap Z = \varnothing, \quad \text{for } i = 1, 2, \ldots, m \tag{10h}$$

$$\{n\} \subseteq Z, \qquad \text{for } n < n_m,\ n \notin \{n_1, \ldots, n_m\} \tag{10i}$$

The equation (10g) states that every number greater than n_m must be in Z. The next two equations, similarly to the equations for Y, define, for each number not exceeding n_m, that it should be in Z if and only if it is not in K. Altogether these equations specify $Z = \mathbb{N} \setminus K$, which completes the proof. □

This result basically allows using finite and co-finite constants in all constructions in the rest of this section, under the assumption that these constants are actually expressed in separate variables, using such equations as in Lemma 6.

5.2. Two general translation lemmata

Consider resolved systems of equations over sets of numbers, as in Section 3. They are of the form

$$X_i = \varphi_i(X_1, \ldots, X_n) \quad (1 \leqslant i \leqslant n),$$

where each right-hand side φ_i may contain union, intersection and addition, as well as singleton constants. This subsection defines a syntactical transformation of resolved equations of a certain special form defined below into unresolved equations using only one Boolean operation: that is, either union or intersection. Under those special conditions, the other Boolean operation shall be replaced by set inclusion.

For instance, when a resolved system with union and intersection is transformed into a system using only union, equations of the form $Z = X \cap Y$ should be eliminated. Such an equation is rewritten as two inequalities $Z \subseteq X$ and $Z \subseteq Y$. However, this is not sufficient, because then Z may be any subset of $X \cap Y$, and it can be a proper subset. To ensure that nothing is missing from Z, another constraint is added: let the variable Z in question be one of several variables $Z_1, \ldots, Z_k$, and assume that the union of all Z_i in the intended solution is a set $C \subseteq \mathbb{N}$ of an easily representable kind (such as the set of all numbers), and, furthermore, all sets Z_i are disjoint. Then, adding an extra equation of the form $\bigcup_i Z_i = C$, in particular, forces Z to be exactly $X \cap Y$.

Let $\mathcal{C}$ be an arbitrary class of constant sets of numbers, which are of an easy form, in the sense that they can be represented as a unique solution of a separate system of equations. For instance, $\mathcal{C}$ can be the class of all finite and co-finite sets (representable as in Lemma 6), and later on, as the expressive power of unresolved equations with one Boolean operations is further developed, one can let $\mathcal{C}$ be the class of all sets with regular base-k representation, etc.

The first pre-requisite of the transformation is that the variables of the original resolved system are grouped into subsets of variables as follows.

Definition 4. Let $\mathcal{C}$ be a class of constant sets. Let $X_i = \varphi_i(X_1, \ldots, X_n)$ be a resolved system of equations with union, intersection, addition and constants from $\mathcal{C}$, and let $(S_1, \ldots, S_n)$ be its least solution. Then a *group* is any such subset of variables $\{X_i\}_{i \in I}$, that the sets $\{S_i\}_{i \in I}$ are pairwise disjoint and their union is in $\mathcal{C}$.

The proposed transformation requires that the set of variables of the given system of resolved equations is *covered* by such groups, in the sense that every variable belongs to some group. The groups need not be disjoint, that is to say, a variable may belong to multiple groups at the same time.

In order for this transformation to work, the system has to satisfy several further conditions. The equation for each variable may be either a union of sums $X = \alpha_1 \cup \ldots \cup \alpha_\ell$, where every α_i is a sum of variables and constants from $\mathcal{C}$, or an intersection of sums $X = (\alpha_1 \cap \ldots \cap \alpha_\ell) \cup C$, followed by a union with a constant set $C \in \mathcal{C}$. These two types of equations shall be called *union-equations* and *intersection-equations*, respectively.

Another requirement is that the given system does not contain such equations as $X = X \cup \varphi$ or $X = X \cap \varphi$, according to which the membership of a number in a variable depends on the membership of the same number in the same variable. Similar dependencies via longer chains of equations are also prohibited.

Definition 5. A resolved system of equations is said to have a *chain dependency* of X on Y, if there is such a sequence of variables $X_0 = X, X_1, \ldots, X_{\ell-1}, X_\ell = Y$, that the equation defining each X_i with $i \in \{0, \ldots, \ell-1\}$ is of the form $X_i = X_{i+1} \cap \varphi$ or $X_i = X_{i+1} \cup \varphi$, where φ is an arbitrary expression.

A dependency of a variable on itself is called a *cyclic chain dependency*, and the transformation requires that the system has no such cycles.

Besides cyclic chain dependencies, another possible source of a self-dependency is any occurrence of zero in the solution: indeed, a sum $X + Y$, with $0 \in X$, induces a chain dependency, as in Definition 5. In order to rule out this problem both in the original system and in the system to be constructed, it is sufficient to require that no constant set contains the zero.

With all requirements explained, the transformation can now be defined, first for the case when the intersection is eliminated.

Lemma 7. *Let $\mathcal{C}$ be a class of constant sets, in which every constant contains only positive integers. Let $X_i = \varphi_i(X_1, \ldots, X_n)$ be a resolved system of equations with union, intersection and addition and with constants from $\mathcal{C}$, where the equation for each variable X_i is of one of the following two forms.*

$$X_i = \bigcup_j \alpha_{i,j} \qquad \textit{(union-equation)}$$

$$X_i = \bigcap_j \alpha_{i,j} \cup C_i \qquad \textit{(intersection-equation)}$$

Here each $\alpha_{i,j}$ is a sum $A_1 + \ldots + A_k$, with $k \geqslant 1$ and with each A_t being a constant or a variable, and $C_i \in \mathcal{C}$ in the second equation is a constant set. Let $(S_1, \ldots, S_n)$ be the least solution of the system. Assume that each variable belongs to some group, in the sense of Definition 4. Furthermore, assume that there are no cyclic chain dependencies in the system.

Then there exists an unresolved system using union and addition, with the same variables and with constants from $\mathcal{C}$, which has the unique solution $(S_1, \ldots, S_n)$. Given the original system, the groups of variables and the constants representing the union of each group, the resulting unresolved system can be effectively constructed.

Proof. Such a system is obtained by preserving all union-equations as they are, rewriting all intersection-equations with inequalities, and finally expressing the union of every group by an extra equation.

Each union-equation

$$X_i = \bigcup_{j=1}^{\ell} \alpha_{i,j}, \tag{11a}$$

where $\alpha_{i,j}$ is a sum of variables and constants, is included in the new system as it is. Every intersection-equation $X_i = \bigcap_{j=1}^{\ell} \alpha_{i,j} \cup C_i$, in which every $\alpha_{i,j}$ is a sum of constants and variables and C_i is a constant, is replaced in the new system by the following collection of inequalities:

$$X_i \subseteq \alpha_{i,j} \cup C_i \quad \big(\text{for all } j \in \{1, \ldots, \ell\}\big). \tag{11b}$$

For each group of variables $\{X_i\}_{i \in I}$, which has a union $\bigcup_{i \in I} S_i = \widehat{C}_I$, the following equation is added:

$$\bigcup_{i \in I} X_i = \widehat{C}_I. \tag{11c}$$

Clearly, the least solution $(S_1, \ldots, S_n)$ of the original resolved system is a solution of the new system. It remains to prove that no other solutions exist.

First, observe that no solution of the constructed system has a zero in any of its components. Indeed, by the equation (11c), each component of the solution is a subset of some constant $\widehat{C}_I$, and all constants are 0-free. Thus, $0 \notin S_i$ for all $i \in \{1, \ldots, n\}$.

Assume, for the sake of contradiction, that there exists another solution $(S'_1, \ldots, S'_n)$. Then, for some variable X_i, there is a number m in the symmetric difference $S_i \mathbin{\Delta} S'_i$. Such a number is called *wrong* or *wrong for* X_i. In particular, if $m \in S'_i \setminus S_i$, then m is said to be an *extra number* for X_i, and if $m \in S_i \setminus S'_i$, then m is a *missing number* for X_i. Each wrong number is greater than zero, because neither S_i nor S'_i contain the zero.

Let $m > 0$ be the smallest wrong number. Then, if this number is obtained from a non-trivial sum of variables and constants under the substitution of either of these solutions, then it is also obtained under the substitution of the other:

Claim 7.1. *If m is the smallest wrong number and $\alpha = A_1 + \ldots + A_k$, where $k \geqslant 2$ and all A_j are variables and constants, then $m \in \alpha(S_1, \ldots, S_n)$ if and only if $m \in \alpha(S'_1, \ldots, S'_n)$.*

Proof. If $m \in \alpha(S_1, \ldots, S_n)$, then $m = m_1 + \ldots + m_k$, with $m_j \in A_j(S_1, \ldots, S_n)$. As all sets $A_j(S_1, \ldots, S_n)$ are 0-free, each number m_j must be positive. Furthermore, each of them must be less than m, because $k \geqslant 2$. Since m is the smallest wrong number, none of $m_1, \ldots, m_k$ is wrong for its respective variable, and hence $m_j \in A_j(S'_1, \ldots, S'_n)$, which implies that $m \in \alpha(S'_1, \ldots, S'_n)$.

The proof in the other direction is identical. □

Among all pairs (m, X_i), where m is the smallest wrong number and it is wrong for X_i, choose such a pair, that m is an extra number for X_i, and if it is not possible, then a pair with m being a missing number for X_i. Such a choice of X_i implies that if m is a missing number for X_i, then m is not an extra number for any variable. Then it is claimed that m must be wrong for another variable $X_{i'}$, with a chain dependency of X_i on $X_{i'}$. Furthermore, if m was an extra number for X_i, then it is extra for $X_{i'}$.

Assume that X_i has a union-equation $X_i = \bigcup_{j=1}^{\ell} \alpha_{i,j}$ in the original system, which is preserved in the new system. Hence, $S_i = \bigcup_{j=1}^{\ell} \alpha_{i,j}(S_1, \ldots, S_n)$. If m is an extra number, then $m \in \alpha_{i,j}(S'_1, \ldots, S'_n) \setminus \alpha_{i,j}(S_1, \ldots, S_n)$ for some j; similarly, if m is a missing number, then there is $\alpha_{i,j}$, such that $m \in \alpha_{i,j}(S_1, \ldots, S_n) \setminus \alpha_{i,j}(S'_1, \ldots, S'_n)$. Clearly, in neither case $\alpha_{i,j}$ is a constant. By Claim 7.1, in both cases, $\alpha_{i,j}$ cannot be a non-trivial sum of variables and constants. Hence, $\alpha_{i,j}$ is a single variable, say $X_{i'}$. Then there is a chain dependency of X_i on $X_{i'}$, and m is wrong for $X_{i'}$. Furthermore, if m is an extra (missing) number for X_i, it is an extra (missing, respectively) number for $X_{i'}$ as well.

Let the equation for X_i in the original system be an intersection-equation $X_i = \bigcap_{j=1}^{\ell} \alpha_{i,j} \cup C_i$, and consider any group of variables containing X_i, that is, a subset $I \subseteq \{1, \ldots, n\}$ with $i \in I$ and $\bigcup_{j \in I} S_j = \widehat{C}_I$. Suppose that m is a missing number for X_i. By the equation (11c), $\bigcup_{j \in I} S_j = \widehat{C}_I$, and hence $m \in \widehat{C}_I$. On the other hand, $\bigcup_{j \in I} S'_j = \widehat{C}_I$ by the same equation, and hence m must be in $S'_{i'}$ for some $i' \in I$. Since m is missing for X_i, it is not an element of S'_i, and so $i \neq i'$. As the sets S_i and $S_{i'}$ are disjoint, $m \in S_i$ cannot be an element of $S_{i'}$, and hence m is an extra number for $X_{i'}$. This is a contradiction: m is an extra number for $X_{i'}$, and m was assumed not to be an extra number for any variable.

Let again X_i be defined by an intersection-equation $X_i = \bigcap_{j=1}^{\ell} \alpha_{i,j} \cup C_i$, and assume that m is an extra number. Because m is not in S_i, it cannot be in C_i. In the new system, there are inclusions $X_i \subseteq \alpha_{i,j} \cup C_i$ for all $j \in \{1, \ldots, \ell\}$. Then, since $m \in S'_i$, it is an element of $\alpha_{i,j}(S'_1, \ldots, S'_n)$ for all j. On the other hand, $m \notin S_i = \bigcap_{j=1}^{\ell} \alpha_{i,j}(S_1, \ldots, S_n) \cup C_i$, and so there is a particular value of j, for which $m \notin \alpha_{i,j}(S_1, \ldots, S_n)$. Hence, $m \in \alpha_{i,j}(S'_1, \ldots, S'_n) \setminus \alpha_{i,j}(S_1, \ldots, S_n)$. Clearly, $\alpha_{i,j}$ cannot be a constant; assuming that it is a non-trivial sum would derive a contradiction by Claim 7.1. Therefore, $\alpha_{i,j}$ is a variable $X_{i'}$. Then there is a chain dependency of X_i on $X_{i'}$ and m is an extra number for $X_{i'}$.

Now the same argument applies to the pair $(m, X_{i'})$, and in this way an infinite sequence of variables with a chain dependency on their successors is obtained. Eventually, some variable is repeated, and this is a contradiction, as there are no cyclic chain dependencies in the system. □

A similar construction produces equations with intersection instead of union. The next lemma is very similar in spirit and proof technique to Lemma 7, but the details of the construction are different, and therefore it is proved separately.

Lemma 8. *Under the assumptions of Lemma 7, there exists and can be effectively constructed an unresolved system with intersection and addition and with constants from $\mathcal{C}$, which has a unique solution that coincides with the least solution of the given system.*

Proof. The new system is obtained by the following transformation. For every union-equation $X_i = \bigcup_{j=1}^{\ell} \alpha_{i,j}$ in the original system, where each $\alpha_{i,j}$ is a sum of constants and variables, the new system contains the following inequalities:

$$\alpha_{i,j} \subseteq X_i, \quad \text{for each } j \in \{1, \ldots, \ell\}. \tag{12a}$$

For each intersection-equation $X_i = \bigcap_{j=1}^{\ell} \alpha_{i,j} \cup C_i$, where C_i is a constant and $\alpha_{i,j}$ is a sum of constants and variables, one should eliminate the union with the constant. This is done by replacing it with the following two inequalities:

$$C_i \subseteq X_i, \tag{12b}$$

$$\bigcap_{j=1}^{\ell} \alpha_{i,j} \subseteq X_i. \tag{12c}$$

Finally, for every group of variables $\{X_i\}_{i \in I}$ with a union $\bigcup_{i \in I} S_i = \widehat{C}_I$, the following equations are added:

$$X_i \cap X_j = \varnothing, \quad \text{for each } i, j \in I \text{ with } i \neq j, \tag{12d}$$

$$X_i \subseteq \widehat{C}_I, \quad \text{for each } i \in I. \tag{12e}$$

Clearly, the solution $(S_1, \ldots, S_n)$ of the former system is a solution of the constructed system. It should be proved that no other solutions exist.

Like in Lemma 7, all solutions of the constructed system are 0-free, because every variable X_i is bounded by a constant set by the inequality (12e), and each constant $\widehat{C}_I$ is 0-free. Also, since the assumptions of the lemma are the same as in Lemma 7, Claim 7.1 holds.

Suppose that there is another solution $(S'_1, \ldots, S'_n)$; as proved above, none of S'_i contains the zero. Define wrong numbers, missing numbers and extra numbers as in the proof of Lemma 7. Let m be the smallest wrong number, and let i be the number of any component, for which m is in the symmetric difference $S_i \,\Delta\, S'_i$. Since neither S_i nor S'_i contain the zero, m must be positive. Among all such pairs (m, X_i) that m is the smallest wrong number and it is wrong for X_i, choose one in which m is a missing number, if there is any such pair. If there are none, then choose a pair (m, X_i) where m is an extra number for X_i. The choice of m and X_i implies that if m is an extra number for X_i, then m is not a missing number for any variable. As in the proof of Lemma 7, the idea is to show that there is another variable $X_{i'}$, with a chain dependency of X_i on $X_{i'}$, for which, if m is missing (extra) number for $X_{i'}$, then it is a missing number (extra number, respectively) for $X_{i'}$.

Suppose first that m is an extra number, that is, $m \in S'_i \setminus S_i$. Let $I \subseteq \{1,\dots,n\}$ with $i \in I$ be any group containing X_i, and consider the corresponding inequality (12e). Since $(S'_1,\dots,S'_n)$ satisfies this inequality, $m \in \widehat{C}_I$. On the other hand, by the definition of a group, $\bigcup_{j\in I} S_j = \widehat{C}_I$, and hence there exists such $i' \in I$, that $m \in S_{i'}$. Since m is in $S_{i'}$ but not in S_i, it follows that $i \neq i'$. At the same time, $S'_i \cap S'_{i'} = \varnothing$ by (12d), and hence the number m is not in $S'_{i'}$. Thus, m is a missing number for $X_{i'}$, which contradicts the assumption that m is not a missing number for any variable.

Assume now that m is a missing number and the equation for X_i in the original resolved system is a union-equation $X_i = \bigcup_{j=1}^{\ell} \alpha_{i,j}$. By the construction, there are equations $\alpha_{i,j} \subseteq X_i$ for each $j \in \{1,\dots,\ell\}$ in the new system, and hence $m \notin \alpha_{i,j}(S'_1,\dots,S'_n)$ for each $j \in \{1,\dots,\ell\}$. On the other hand, $m \in S_i = \bigcup_{j=1}^{\ell} \alpha_{i,j}(S_1,\dots,S_n)$, and thus, $m \in \alpha_{i,j}(S_1,\dots,S_n)$ for some $j \in \{1,\dots,\ell\}$. Altogether, $m \in \alpha_{i,j}(S_1,\dots,S_n) \setminus \alpha_{i,j}(S'_1,\dots,S'_n)$. By Claim 7.1, $\alpha_{i,j}$ cannot be a non-trivial sum. Clearly, it cannot be a constant either, and hence it must be a variable, that is, $\alpha_{i,j} = X_{i'}$ for some i'. Then there is a chain dependency of X_i on $X_{i'}$, and m is a missing number for $X'_{i'}$.

Suppose now that m is a missing number and X_i has an intersection-equation $X_i = \bigcap_{j=1}^{\ell} \alpha_{i,j} \cup C_i$ in the original system. The construction guarantees that there are equations $\bigcap_{j=1}^{\ell} \alpha_{i,j} \subseteq X_i$ and $C_i \subseteq X_i$ in the new system. Then $m \notin \bigcap_{j=1}^{\ell} \alpha_{i,j}(S'_1,\dots,S'_n)$ and $m \notin C_i$. On the other hand, as $m \in S_i$, it holds that $m \in \bigcap_{j=1}^{\ell} \alpha_{i,j}(S_1,\dots,S_n) \cup C_i$. Since $m \notin C_i$ by previous observation, $m \in \bigcap_{j=1}^{\ell} \alpha_{i,j}(S_1,\dots,S_n)$, hence $m \in \alpha_{i,j}(S_1,\dots,S_n)$ for every $j \in \{1,\dots,\ell\}$. As $m \notin \bigcap_{j=1}^{\ell} \alpha_{i,j}(S'_1,\dots,S'_n)$, there exists such $j \in \{1,\dots,\ell\}$, that $m \in \alpha_{i,j}(S_1,\dots,S_n) \setminus \alpha_{i,j}(S'_1,\dots,S'_n)$. Similarly to the analysis in the previous case, $\alpha_{i,j}$ cannot be a constant and cannot be a non-trivial sum. Hence, $\alpha_{i,j} = X_{i'}$ for some variable $X_{i'}$, that is, there is a chain dependency of X_i on $X_{i'}$ and m is a missing number for $X_{i'}$.

And so, for every number m missing for a variable X_i, it is possible to find another such variable $X_{i'}$, that m is missing for it as well, and there is a chain dependency of X_i on $X_{i'}$. Since there are only finitely many variables, this forms a cyclic chain dependency. As the system has no cyclic chain dependencies by assumption, a contradiction is obtained. □

The next task is to apply Lemmata 7 and 8 to resolved systems constructed in the proofs of Theorems B and C. For the lemmata to be applicable, the equations given by Jeż [10] and by Jeż and Okhotin [11] need to be decomposed into smaller parts and slightly changed. Then the variables can be grouped, as required by the lemmata.

5.3. Sets with a regular base-k notation

Using the lemmata from the previous section, the resolved equations of Jeż [10] and by Jeż and Okhotin [11] will now be converted to unresolved equations with addition and either only union or only intersection. The first task is to reformulate them, so that Lemmata 7 and 8 become applicable.

The following known properties of equations over sets of numbers will be used in the constructions. The first of these properties is motivated by the fact that most of the later constructions require the base k to be large enough. In order to extend those representations to small values of k, the next lemma asserts that those sets are equally representable in bases k^2, k^3, etc.

Lemma 9 *([11, Lem. 3]). Let $S \subseteq \mathbb{N}$ be a set of numbers, let k and k^m (with $k \geqslant 2$, $m \geqslant 2$) be two bases of positional notation. Then the language $L \subseteq \Sigma_k^* \setminus 0\Sigma_k^*$ of base-k representations of numbers in S is regular (linear conjunctive) if and only if the language $L' \subseteq \Sigma_{k^m}^* \setminus 0\Sigma_{k^m}^*$ of their base-k^m representations is regular (linear conjunctive, respectively).*

Indeed, a base-k^m digit is nothing but a group of m base-k digits, and a finite automaton and a trellis automaton processing one of these representations can be transformed to process the other.

The next technical lemma presents a class of expressions over sets of numbers that are *distributive over infinite union*, so that one can evaluate such expressions on a set $S \subseteq \mathbb{N}$ by evaluating it on all singletons $\{n\}$ with $n \in S$, and then taking the union of the results.

Lemma 10 *([11, Lem. 4]). Let $\varphi(X)$ be an expression defined as a composition of (i) the variable X; (ii) constant sets; (iii) union; (iv) intersection with a constant set; (v) addition of a constant set. Then φ is distributive over arbitrary union, that is, $\varphi(X) = \bigcup_{n\in X} \varphi(\{n\})$.*

For any two digits $i, j \in \Sigma_k$, of which i is non-zero, consider a set of natural numbers with base-k representation $ij0^*$; for example, $(120^*)_4 = \{6 \cdot 4^n \mid n \geqslant 0\}$. It is known that every such set is representable by a resolved system with union, intersection and addition [10]. This result will now be reconstructed to use only one Boolean operation, at the expense of turning the resolved equations into unresolved ones. The new construction is based upon a slightly modified version of equations from the original paper [10].

Lemma 11. *For every $k \geqslant 9$, there exists and can be effectively constructed an unresolved system with union (intersection), addition and constant $\{1\}$, which has a unique solution, and among the components of this solution, there are all sets of the form*

$$\left(ij0^*\right)_k, \quad \text{for all } i, j \in \Sigma_k \text{ with } i \neq 0.$$

Proof. It is known that there exists a resolved system of equations with union, intersection and addition representing all sets $S_{i,j} = (ij0^*)_k$ by variables $X_{i,j}$, which are recursively expressed through each other [10, Thm. 14]. This previously known system is presented below.

$$X_{i,j} = \begin{cases} \left(\bigcap_{n=1}^{2}(X_{k-n,0} + X_{j+n,0})\right) \cup (1j)_k, & \text{for } i = 1 \text{ and } j \in \{0,1,2\}, \\ \left(\bigcap_{n=1}^{2}(X_{i-1,k-n} + X_{j+n,0})\right) \cup (ij)_k, & \text{for } i \geqslant 2 \text{ and } j \in \{0,1,2\}, \\ \left((X_{i,0} + X_{j,0}) \cap \bigcap_{n=1}^{2}(X_{i,j-n} + X_{n,0})\right) \cup (ij)_k, & \text{for } i \geqslant 1 \text{ and } j \geqslant 3. \end{cases}$$

Its least solution was proved to be $X_{i,j} = (ij0^*)_k$. In general, the system works very similarly to Example 1 (the powers of four): every set of the form $(ij0^*)_k$ is represented in two different ways, each containing some junk, and once these two representations are intersected, all the junk is cancelled. For example, in the first case of the above equations, the intersection is

$$\begin{aligned} &\left[\left((k-1)0^+\right)_k + \left((j+1)0^+\right)_k\right] \cap \left[\left((k-2)0^+\right)_k + \left((j+2)0^+\right)_k\right] \\ &\quad = \left[\left(1j0^+\right)_k \cup \left((k-1)0^*(j+1)0^+\right)_k \cup \left((j+1)0^*(k-1)0^+\right)_k\right] \\ &\qquad \cap \left[\left(1j0^+\right)_k \cup \left((k-2)0^*(j+2)0^+\right)_k \cup \left((j+2)0^*(k-2)0^+\right)_k\right] \\ &\quad = \left(1j0^+\right)_k, \end{aligned}$$

and thus the entire equation defines $X_{1j} = (1j0^+)_k$. A full correctness proof for this construction can be found in the literature [10, Thm. 14].

The above system defining the sets $(ij0^*)_k$ is not sufficient for the present paper, since these sets cannot be grouped to match the conditions of Lemmata 7 and 8. To remedy this, a construction representing the complementary sets $\widetilde{S}_{i,j} = (ij(\Sigma_k^* \setminus 0^*))_k$ should also be given. Then, all the "two-digit sets" put together, that is, both $S_{i,j}$ and $\widetilde{S}_{i,j}$, for all leading digits i, j, will be pairwise disjoint and their union will be co-finite, making the translation lemmata applicable. Co-finiteness of the union is important, because co-finite sets are so far the only infinite sets proved to be representable by equations with intersection and addition.

The system of equations defining these complementary sets in turn requires further sets: these are the "three-digit sets" of the form $(ij\ell0^*)_k$, where i, j, ℓ are any three leading digits, followed by trailing zeroes. The system defining these "three-digit sets" is constructed similarly to the above system for $X_{i,j}$, and builds upon the variables $X_{i,j}$ defined in that system.

$$X_{i,j,\ell} = \begin{cases} \bigcap_{n=1}^{4}(X_{k-n,0} + X_{j+n,\ell}), & \text{for } i = 1,\ j \leqslant 3,\ \ell \in \Sigma_k, \\ \bigcap_{n=1}^{4}(X_{i-1,j+n} + X_{k-n,\ell}), & \text{for } i \geqslant 2,\ j \leqslant 3,\ \ell \in \Sigma_k, \\ \bigcap_{n=0}^{3}(X_{i,n} + X_{j-n,\ell}), & \text{for } i \geqslant 1,\ j \geqslant 4,\ \ell \in \Sigma_k. \end{cases}$$

The proof that its least solution is $X_{i,j,\ell} = (ij\ell0^*)_k$ can be carried out using the same methods as in the original paper [10, Thm. 14].

The construction for the sets $\widetilde{S}_{i,j}$ can also be deduced from the previous work [10, Lem. 17]. Consider that $\Sigma_k^* \setminus 0^*$ is a regular language recognized by a finite automaton reading the string of digits from the right to the left. The automaton has two states, q_0 and q_1; it remains in its initial state q_0 while all digits encountered so far are zeroes, and once any non-zero digit is read, it enters the (accepting) state q_1 and remains there. Applying the known construction [10, Lem. 17] to this automaton gives a system in variables $Y_{i,j,q}$, where $i, j \in \Sigma_k$, $i \neq 0$ and $q \in \{q_0, q_1\}$. The unique solution in sets of positive integers of that system is $Y_{i,j,q_0} = S_{i,j}$ and $Y_{i,j,q_1} = \widetilde{S}_{i,j}$. Note that the variables Y_{i,j,q_0} have the same values as $X_{i,j}$ from the earlier constructed system. Thus, Y_{i,j,q_0} may be replaced by $X_{i,j}$, and furthermore, Y_{i,j,q_1} shall be referred to simply as $Y_{i,j}$.

Besides the complementary sets for the "two-digit" variables $X_{i,j}$, the "three-digit" variables $X_{i,j,\ell}$ defining the sets $(ij\ell(\Sigma_k^* \setminus 0^*))_k$ also require the complementary sets of their own, $(ij\ell(\Sigma_k^* \setminus 0^*))_k$, which will be defined by the variables $Y_{i,j,\ell}$. The equations defining both $Y_{i,j}$ and $Y_{i,j,\ell}$ are written below; as compared to the original paper [10, Lem. 17], they are slightly reformulated to match Lemmata 7 and 8.

$$Y_{i,j} = \bigcup_{\ell \neq 0} X_{i,j,\ell} \cup \bigcup_{\ell \in \Sigma_k} Y_{i,j,\ell}, \quad \text{for } i \geqslant 1,\ j \in \Sigma_k,$$

$$Y_{i,j,\ell} = \begin{cases} \bigcap_{n=1}^{4}(X_{k-n,0} + Y_{j+n,\ell}), & \text{for } i = 1,\ j \leqslant 3,\ \ell \in \Sigma_k, \\ \bigcap_{n=1}^{4}(X_{i-1,j+n} + Y_{k-n,\ell}), & \text{for } i \geqslant 2,\ j \leqslant 3,\ \ell \in \Sigma_k, \\ \bigcap_{n=0}^{3}(X_{i,n} + Y_{j-n,\ell}), & \text{for } i \geqslant 1,\ j \geqslant 4,\ \ell \in \Sigma_k. \end{cases}$$

As already said, the unique solution of these equations is:

$$Y_{i,j} = \left(ij\left(\Sigma_k^* \setminus 0^*\right)\right)_k,$$
$$Y_{i,j,\ell} = \left(ij\ell\left(\Sigma_k^* \setminus 0^*\right)\right)_k.$$

The equations for $Y_{i,j,\ell}$ work in a similar way to equations in Example 1: each number $(ij\ell w)_k \in (ij\ell(\Sigma_k^* \setminus 0^*))_k$ is represented in four different ways as sums of numbers from appropriate sets. The intersection of those sums guarantees that the garbage is filtered out. The equation for $Y_{i,j}$ expresses the set of all numbers $(ijw)_k \in (ij(\Sigma_k^* \setminus 0^*))_k$ as the union of the following two possibilities: either w is in $\ell 0^*$, in which case the desired number is in $X_{i,j,\ell}$, or w contains a non-zero digit after its first digit ℓ, and thus $(ijw)_k$ is found in $Y_{i,j,\ell}$.

It remains to show that these equations satisfy the assumptions of Lemmata 7 and 8, with the variables separated into the following two groups, one comprised of all "two-digit" variables, and the other of the "three-digit" variables:

$$\mathcal{G}_2 = \{X_{i,j} \mid i, j \in \Sigma_k,\ i \neq 0\} \cup \{Y_{i,j} \mid i, j \in \Sigma_k,\ i \neq 0\},$$
$$\mathcal{G}_3 = \{X_{i,j,\ell} \mid i, j, \ell \in \Sigma_k,\ i \neq 0\} \cup \{Y_{i,j,\ell} \mid i, j, \ell \in \Sigma_k,\ i \neq 0\}.$$

The unions of the corresponding sets in the solution for the group $\mathcal{G}_2$ is $\{n \mid n \geqslant k\}$, and for $\mathcal{G}_3$ it is $\{n \mid n \geqslant k^2\}$; both are co-finite sets. In each group, all the components are pairwise disjoint, separated by the leading digits. The only chain dependencies are those of variables $Y_{i,j}$ on some $X_{i,j,\ell}$ and $Y_{i,j,\ell}$; hence, there are no cyclic chain dependencies. And therefore, by Lemma 7 and Lemma 8, there exist unresolved systems with union (intersection), addition and finite and co-finite constants, whose unique solution has the requested components. Co-finite and finite constants are eliminated by expressing them according to Lemma 6. □

Now the construction of Theorem B can be remade for unresolved equations using only one Boolean operation.

Lemma 12. *For every $k \geqslant 9$ and for every deterministic finite automaton $M = (\Sigma_k, Q, q_0, \delta, F)$, there exists an unresolved system of equations using union (intersection), addition and constant $\{1\}$, in which some of the components of the unique solution are*

$$S_{i,j,q} := \left\{(ijw)_k \;\middle|\; w \in \Sigma_k^*,\ \delta(q_0, w^R) = q\right\}, \quad \text{for } i, j \in \Sigma_k,\ i \neq 0,\ q \in Q.$$

Given k and M, the system can be effectively constructed.

Recalling a standard definition, a DFA $M = (\Sigma, Q, q_0, \delta, F)$ has an input alphabet Σ, a set of states Q, an initial state $q_0 \in Q$, a complete transition function $\delta\colon Q \times \Sigma \to Q$ that defines the state entered upon reading a given symbol in a given state, and a set of accepting states $F \subseteq Q$. For a state $q \in Q$ and a string $x \in \Sigma^*$, denote the state reached by M from q after reading x by $\delta(q, x)$. The language defined by the automaton is $L(M) = \{x \in \Sigma^* \mid \delta(q_0, x) \in F\}$.

Note that the automaton in Lemma 12 processes the reversal w^R of the string of digits w, that is, reads w from right to left.

Proof of Lemma 12. Consider the following resolved system of language equations [10, Lem. 17], which uses constants of the form $(ij0^*)_k$.

$$X_{i,j,q} = \begin{cases} \bigcup\limits_{\substack{\ell \in \Sigma_k,\ q' \in Q\colon \\ \delta(q',\ell)=q}} X_{i,j,\ell,q'} & \text{for } i \geqslant 1,\ j \in \Sigma_k,\ q \neq q_0, \\ \bigcup\limits_{\substack{\ell \in \Sigma_k,\ q' \in Q\colon \\ \delta(q',\ell)=q}} X_{i,j,\ell,q'} \cup (ij)_k, & \text{for } i \geqslant 1,\ j \in \Sigma_k,\ q = q_0. \end{cases}$$

$$X_{i,j,\ell,q} = \begin{cases} \bigcap_{n=1}^{4} \left((k-n)00^*)_k + X_{j+n,\ell,q}\right), & \text{for } i = 1,\ j \leqslant 3, \\ \bigcap_{n=1}^{4} \left(((i-1)(j+n)0^*)_k + X_{k-n,\ell,q}\right), & \text{for } i \geqslant 2,\ j \leqslant 3, \\ \bigcap_{n=0}^{3} \left((in0^*)_k + X_{j-n,\ell,q}\right), & \text{for } i \geqslant 1,\ j \geqslant 4. \end{cases}$$

For these equations, it was proved that their unique solution in sets of positive integers is

$$X_{i,j,q} = \left\{(ijw)_k \;\middle|\; \delta(q_0, w^R) = q\right\},$$
$$X_{i,j,\ell,q} = \left\{(ij\ell w)_k \;\middle|\; \delta(q_0, w^R) = q\right\}.$$

The idea behind these equations for the variables $X_{i,j,\ell,q}$ is similar to the one for equations for the variables $Y_{i,j,\ell}$ in Lemma 11. Each number from the set $\{(ij\ell w)_k \mid \delta(q_0, w^R) = q\}$ is represented in four different ways as a sum of numbers from appropriate sets. Those representations are intersected, and so the numbers of any other form are filtered out. The simulation of the automaton is encoded in the equations for $X_{i,j,q}$. Consider any number $(ijw)_k$ that should be in $X_{i,j,q}$; then the automaton is in the state q after reading w^R. There are two natural cases, with w is empty or not. If w is an empty string, then the considered number is simply $(ij)_k$, and it is explicitly listed on the right-hand side for $X_{i,j,q}$. If w is not empty, then it can be written as $\ell w'$. Let q' be the state reached by the automaton after reading w'. Then the corresponding number $(ij\ell w')_k$ is in $X_{i,j,\ell,q'}$. Each number $(ijw)_k$ falls into one of the considered cases, which yields the equation for $X_{i,j,q}$.

In order to obtain an unresolved system of equations with the same unique solution, the plan is to apply Lemmata 7 and 8 to the above system. To this end, the variables of the system have to be grouped. Again, there will be two groups,

$$\mathcal{G}_2 = \{X_{i,j,q} \mid i, j \in \Sigma_k,\ i \neq 0,\ q \in Q\} \quad \text{and}$$
$$\mathcal{G}_3 = \{X_{i,j,\ell,q} \mid i, j, \ell \in \Sigma_k,\ i \neq 0,\ q \in Q\}.$$

The union of the solutions in $\mathcal{G}_2$ is $\{n \mid n \geqslant k\}$, and $\{n \mid n \geqslant k^2\}$ for $\mathcal{G}_3$. There are chain dependencies of $X_{i,j,q}$ on $X_{i,j,\ell,q'}$, but none of them are cyclic. The solutions for the variables in each group are disjoint, because M is deterministic and hence enters a unique state upon each string.

The resulting system uses two co-finite constants obtained as unions of the groups, as well as constants of the form $(ij0^*)_k$. The former are expressed as in Lemma 6, while the latter are replaced by references to equations from Lemma 11. □

The simulation of a finite automaton M operating on base-k representations of numbers defined in Lemma 12 does not express the exact set $(L(M))_k$, as it appends extra leading digits ij to every number and does not take the accepting states into account. From this construction, one can infer the following final form of the result on the representation of any set with regular base-k notation, similar to the one of Theorem B.

Theorem 2. *For every $k \geqslant 2$ and for every regular language $L \subseteq \Sigma_k^* \setminus 0\Sigma_k^*$, there exists and can be effectively constructed an unresolved system with union (intersection), addition and constant $\{1\}$, which has a unique solution, with $(L)_k$ as one of its components.*

Proof. First consider the case of $2 \leqslant k < 9$. Then, by Lemma 9, there exists a regular language $L' \subseteq \Sigma_{k'}^*$ for $k' = k^4 > 9$, such that $(L')_{k'} = (L)_k$. Hence, it is sufficient to establish the theorem for $k \geqslant 9$.

Let $M = (\Sigma_k, Q, q_0, \delta, F)$ be a deterministic finite automaton recognizing L^R. By Lemma 12, there exists an unresolved system of the required form, in which every variable $X_{i,j,q}$ in the unique solution equals $\{(ijw)_k \mid \delta(q_0, w^R) = q\}$. Then the set $(L)_k$ can be obtained as the following union:

$$(L)_k = \underbrace{\big((L)_k \cap \{n \mid n < k\}\big)}_{\text{finite constant}} \cup \bigcup_{\substack{i,j,q:\\ \delta(q,ji)\in F}} \underbrace{\big\{(ijw)_k \,\big|\, \delta\big(q_0, w^R\big) = q\big\}}_{X_{i,j,q}}. \tag{13}$$

In the case of unresolved equations with union, the equality (13) can be directly expressed by introducing a new variable Y and adding the following equation:

$$Y = C \cup \bigcup_{\substack{i,j,q:\\ \delta(q,ji)\in F}} X_{i,j,q},$$

where the finite constant $C = (L)_k \cap \{n \mid n < k\}$ is expressed according to Lemma 6.

For the case of intersection, consider that the sets $\{(ijw)_k \mid \delta(q_0, w^R) = q\}$, along with the finite set $\{n \mid n < k\}$, form a partition of $\mathbb{N}$ (because, for every string of digits w, the transition function $\delta(q_0, w^R)$ leads to some state). Then a new variable Y is added, and its intersection with every element of this partition is expressed:

$$Y \cap \{n \mid n < k\} = (L)_k \cap \{n \mid n < k\},$$
$$Y \cap X_{i,j,q} = \varnothing, \quad \text{for } (i,j,q) \text{ with } \delta(q, ji) \notin F,$$
$$Y \cap X_{i,j,q} = X_{i,j,q}, \quad \text{for } (i,j,q) \text{ with } \delta(q, ji) \in F.$$

Because these equalities state the membership of *every* natural number in Y, this representation is equivalent to (13), and hence the system has a unique solution with $Y = (L)_k$. Both finite constants are again replaced according to Lemma 6. □

5.4. Sets with linear conjunctive base-k notation

The next task is to remake another key construction of a system of equations using only one Boolean operation. As stated in Theorem C, for every trellis automaton M, with $L(M) \subseteq \Sigma_k^+ \setminus 0\Sigma_k^*$, there exists a resolved system of equations over sets of natural numbers with $(L(M))_k$ as one of the components of its least solution. This construction essentially uses *both* union and intersection, and now the goal is again to refine the known construction [11], so that Lemmata 7 and 8 could be applied to it.

This construction essentially uses the operations of symbolic addition and subtraction of 1 on base-k representations of numbers. For every base $k \geqslant 2$ and for every string of digits $w \in \Sigma_k^* \setminus (k-1)^*$, the string $w' = w \boxplus 1$ is defined as the unique string with $|w| = |w'|$ and $(w')_k = (w)_k + 1$. Similarly, for every $w \in \Sigma_k^* \setminus 0^*$, define $w' = w \boxminus 1$ as the unique string with $|w| = |w'|$ and $(w')_k = (w)_k - 1$.

For example, in decimal representation, $0099 \boxplus 1 = 0100$ and $0100 \boxminus 1 = 0099$. This notation shall never be used for strings on which it is undefined, such as $999 \boxplus 1$ and $000 \boxminus 1$. This notation is extended to languages element-wise:

$$L \boxplus 1 = \left\{ w \boxplus 1 \;\middle|\; w \in L \setminus (k-1)^* \right\},$$
$$L \boxminus 1 = \left\{ w \boxminus 1 \;\middle|\; w \in L \setminus 0^* \right\}.$$

This operation obviously preserves regularity, and hence it can be used inside regular expressions defining languages of base-k strings of digits, and the sets thus defined remain regular.

The original construction of a resolved system simulating a trellis automaton went in three stages. First, a set $(1(L(M) \boxminus 1)10^*)_k$ was represented for every trellis automaton M [11, Lem. 5]. Note that each string of digits accepted by $L(M)$ is decremented and then delimited by a pair of digits 1 and a tail of trailing zeroes: for instance, if M accepts 555, then the resulting set contains all numbers $(15541)_k$, $(155410)_k$, $(1554100)_k$, etc. At the next stage, the decrementation, the last digit 1 and the trailing zeroes were eliminated, and thus the set $(1 \cdot L(M))_k$ was represented for any M [11, Lem. 6]. Finally, the desired system of equations defining the set $(L(M))_k$ was obtained [11, Lem. 7], where M is any trellis automaton. This composition of the argument will be followed in the proof below, and each part of the known construction will be carefully remade to allow applying Lemmata 7 and 8.

Lemma 13. *For every base $k \geqslant 4$ and for every trellis automaton $M = (\Sigma_k, Q, I, \delta, F)$ over an alphabet $\Sigma_k = \{0, \ldots, k-1\}$, there exists and can be effectively constructed an unresolved system of equations over sets of natural numbers using union and addition (or intersection and addition) and constant $\{1\}$, such that, for each state $q \in Q$, the unique solution of this system contains a component*

$$\left(1\left(\left(L_M(q) \setminus 0^*\right) \boxminus 1\right)10^*\right)_k = \left\{ \left(1w10^\ell\right)_k \;\middle|\; \ell \geqslant 0,\ w \notin (k-1)^*,\ w \boxplus 1 \in L_M(q) \right\}.$$

Proof. Consider the known resolved system of equations with union, intersection and addition that defines the sets of numbers $S_q := (1((L_M(q) \setminus 0^*) \boxminus 1)10^*)_k$ [11, Lem. 5]. It uses variables X_q for all $q \in Q$, and contains the equations

$$X_q = R_q \cup \bigcup_{\substack{p,r:\delta(p,r)=q \\ i,j \in \Sigma_k}} \left(\lambda_i(X_r) \cap \rho_j(X_p)\right), \quad \text{for all } q \in Q,$$

where

$$R_q = \left\{ \left(1(w \boxminus 1)10^*\right)_k \;\middle|\; w \in 0^*(\Sigma_k \setminus 0) \cup (\Sigma_k \setminus 0)0^*,\ w \in L_M(q) \right\}$$
$$\kappa_{i'}(X) = \left(\left(X \cap \left(1i'\Sigma_k^*10^*\right)_k\right) + \left(10^*\right)_k\right) \cap \left(2i'\Sigma_k^*\right)_k, \quad \text{for all } i' \in \Sigma_k$$
$$\lambda_i(X) = \begin{cases} \bigcup_{i' \in \Sigma_k} \left(\kappa_{i'}(X) + ((k+i-2)0^*)_k\right) \cap (1i\Sigma_k^*)_k, & \text{for } i \in \{0,1\} \\ \bigcup_{i' \in \Sigma_k} \left(\kappa_{i'}(X) + (1(i-2)0^*)_k\right) \cap (1i\Sigma_k^*)_k, & \text{for } i \geqslant 2 \end{cases}$$
$$\pi_{j'}(X) = \left(\left(X \cap \left(1\Sigma_k^*j'10^*\right)_k\right) + \left(10^*\right)_k\right) \cap \left(1\Sigma_k^*j'20^*\right)_k, \quad \text{for all } j' \in \Sigma_k$$
$$\rho_j(X) = \begin{cases} \bigcup_{j' \in \Sigma_k} \left(\pi_{j'}(X) + ((k+j-2)10^*)_k\right) \cap (1\Sigma_k^*j10^*)_k, & \text{for } j \in \{0,1\} \\ \bigcup_{j' \in \Sigma_k} \left(\pi_{j'}(X) + (1(j-2)10^*)_k\right) \cap (1\Sigma_k^*j10^*)_k, & \text{for } 2 \leqslant j \leqslant k-2 \\ \bigcup_{j' \in \Sigma_k} \left(\pi_{j'}(X) + ((k-3)10^*)_k\right) \cap (1\Sigma_k^*(k-1)10^*)_k, & \text{for } j = k-1 \end{cases}$$

A string of digits w is represented as the numbers $(1(w \boxminus 1)10^\ell)_k$, where $\ell \geqslant 0$. The purpose of the expression λ_i is to transform this representation into a similar representation for the string iw, that is, $(1(iw \boxminus 1)10^\ell)_k$. This can be regarded as concatenating a digit i, done on the numerical representation, and it is achieved by a series of additions and intersections, similarly to the previous constructions. The goal of the expression ρ_j is symmetrical: it appends a digit j to w, that is, transforms the representation $(1(w \boxminus 1)10^\ell)_k$ into $(1(wj \boxminus 1)10^{\ell-1})_k$. Note that in the latter transformation, one of the trailing zeroes is consumed. The equation for X_q implements the logic of the trellis automaton, in which the state computed on iwj depends on states computed on iw and wj. Some strings w of a simple form are not covered by this representation; they are grouped in the set R_q, whose base-k representation is regular. More details can be found in the authors' earlier work [11].

The correctness statement for this construction reads that the least solution of the system is $X_q = S_q$ [11, Main Claim]. Also, note that all constants used in the system have regular base-k notation.

Another important property of this system is that the sets S_q corresponding to different states $q \in Q$ are pairwise disjoint, and their union over all $q \in Q$ is a set with a regular base-k representation. Let us establish a general statement useful for proving such results.

Claim 13.1. *Let $x \in \Sigma_k^+ \setminus 0\Sigma_k^*$ and $y \in \Sigma_k^+ \setminus 0^*$ be any strings of digits, let $K_1, \ldots, K_m \subseteq \Sigma_k^+$ be any pairwise disjoint languages. Let $S_1, \ldots, S_m$ be sets of numbers defined by*

$$S_t = \left\{ \left(xuy0^\ell\right)_k \mid \ell \geqslant 0,\ u \in K_t \right\}, \quad \text{for each } t \in \{1, \ldots, m\}.$$

Then these sets are pairwise disjoint and their union is

$$\bigcup_{t=1}^{m} S_t = \left(x\left(\bigcup_{t=1}^{m} K_t\right)y0^*\right)_k.$$

Proof. Consider any two sets S_t and $S_{t'}$, with $t \neq t'$, and suppose there is a number n belonging to both sets. Then $n = (xuy0^\ell)_k$ for some $u \in K_t$, and $n = (xu'y0^{\ell'})_k$ with $u' \in K_{t'}$. Since y contains a non-zero digit, the length of the tail of zeroes in n is independent of u and u', and therefore $\ell = \ell'$. Then u and u' must be the same string, which is impossible, because $K_t \cap K_{t'} = \varnothing$ by assumption. This proves that $S_t \cap S_{t'} = \varnothing$.

The union of these sets is

$$\bigcup_t S_t = \bigcup_t \left(xK_t y0^*\right)_k = \left(x\left(\bigcup_t K_t\right)y0^*\right)_k,$$

as stated. □

Now Claim 13.1 can be applied to the particular case of the sets S_q to obtain the following result.

Claim 13.2. *The sets of numbers S_q, for different states $q \in Q$, are pairwise disjoint, and their union is*

$$\bigcup_{q \in Q} S_q = \left(1\left(\Sigma_k^+ \setminus (k-1)^*\right)10^*\right)_k.$$

Proof. Define $K_q = (L_M(q) \setminus 0^*) \boxminus 1$; these sets are pairwise disjoint and their union for all $q \in Q$ is $\Sigma_k^* \setminus (k-1)^*$, because every string $w \in \Sigma_k^+$ belongs to some $L_M(q)$. The rest is given by Claim 13.1 with $x = y = 1$. □

Though the values of the variables X_q as they are already satisfy Lemma 7 and Lemma 8, the right-hand sides of the above equations are not of the required simple form. Now the goal is to transform the system, splitting the existing equations into smaller parts and introducing new variables, so that it satisfies the assumptions of the lemmata.

The first step is to construct equations of the required form that implement the expressions λ and ρ applied to all variables X_q. Each occurrence of $\lambda_i(X_q)$ will be replaced by a new variable $Z^\lambda_{i,q}$. By definition, $\lambda_i(X_q)$ is a union of k subexpressions corresponding to different digits $i' \in \Sigma_k$; each of these subexpressions shall be defined in a separate variable $Y^\lambda_{i,i',q}$. Similarly, $\kappa_{i'}(X_q)$ is to be replaced by a variable $W^\lambda_{i',q}$, and the inner subexpression of $\kappa_{i'}(X_q)$ is extracted to a variable $U^\lambda_{i',q}$. The new variables are defined by the following resolved equations.

$$U^\lambda_{i',q} = X_q \cap \left(1i'\Sigma_k^*10^*\right)_k \tag{14}$$

$$W^\lambda_{i',q} = \left(U^\lambda_{i',q} + \left(10^*\right)_k\right) \cap \left(2i'\Sigma_k^*\right)_k \tag{15}$$

$$Y^\lambda_{i,i',q} = \begin{cases} \left(W^\lambda_{i',q} + ((k+i-2)0^*)_k\right) \cap (1i\Sigma_k^*)_k, & \text{for } i \in \{0,1\} \\ \left(W^\lambda_{i',q} + (1(i-2)0^*)_k\right) \cap (1i\Sigma_k^*)_k, & \text{for } i \geqslant 2 \end{cases} \tag{16}$$

$$Z^\lambda_{i,q} = \bigcup_{i'} Y^\lambda_{i,i',q} \tag{17}$$

Since the equation for $Z^\lambda_{i,q}$ represents the expression $\lambda_i(X_q)$ broken into pieces, the "old" variables $\{X_q\}$ have the same values in the least solution of the new system as in the least solution of the old system. The newly introduced variables are arranged into the following four groups:

$$\mathcal{U}^\lambda = \left\{U^\lambda_{i',q} \mid i' \in \Sigma_k,\ q \in Q\right\},$$
$$\mathcal{W}^\lambda = \left\{W^\lambda_{i',q} \mid i' \in \Sigma_k,\ q \in Q\right\},$$
$$\mathcal{Y}^\lambda = \left\{Y^\lambda_{i,i',q} \mid i, i' \in \Sigma_k,\ q \in Q\right\},$$
$$\mathcal{Z}^\lambda = \left\{Z^\lambda_{i,q} \mid i \in \Sigma_k,\ q \in Q\right\}.$$

Let us calculate the values of these variables in the least solution. For every variable V, let $\mu(V)$ be the set corresponding to V in the least solution of the new system of equations.

Claim 13.3. *The values of the variables from $\mathcal{U}^{\lambda}$ in the least solution are pairwise disjoint, and their union is $(1(\Sigma_k^+ \setminus (k-1)^*)10^*)_k$.*

Proof. It is to be shown that for any two distinct pairs (i'_1, q_1) and (i'_2, q_2), the sets $\mu(U^{\lambda}_{i'_1,q_1})$ and $\mu(U^{\lambda}_{i'_2,q_2})$ are disjoint, and that

$$\bigcup_{i' \in \Sigma_k,\ q \in Q} \mu\big(U^{\lambda}_{i',q}\big) = \big(1\big(\Sigma_k^+ \setminus (k-1)^*\big)10^*\big)_k.$$

It is already known [11, Eq. (3)] that

$$\mu\big(U^{\lambda}_{i',q}\big) = \big\{\big(1i'w10^{\ell}\big)_k \mid \ell \geqslant 0,\ w \in \Sigma_k^*,\ i'w \notin (k-1)^*,\ i'w \boxplus 1 \in L_M(q)\big\}.$$

These sets are obtained from the languages $K^{\lambda}_{i',q} = ((L_M(q) \setminus 0^*) \boxminus 1) \cap i'\Sigma_k^*$, with $i' \in \Sigma_k$ and $q \in Q$, as in the statement of Claim 13.1 for $x = y = 1$. To see that the languages $K^{\lambda}_{i',q}$ are pairwise disjoint, consider $K^{\lambda}_{i'_1,q_1}$ and $K^{\lambda}_{i'_2,q_2}$: if $i'_1 \neq i'_2$, then the strings in these languages begin with different digits, and if $q_1 \neq q_2$, then $K^{\lambda}_{i'_1,q_1} \subseteq L_M(q_1) \boxminus 1$ and $K^{\lambda}_{i'_2,q_2} \subseteq L_M(q_2) \boxminus 1$. In both cases, $K^{\lambda}_{i'_1,q_1} \cap K^{\lambda}_{i'_2,q_2} = \varnothing$.

Therefore, Claim 13.1 with $x = y = 1$ asserts that $\mu(U^{\lambda}_{i',q})$ are pairwise disjoint and the union of this group of sets can be calculated as follows. First, consider the union of all languages $K^{\lambda}_{i',q}$.

$$\bigcup_{i',q} K^{\lambda}_{i',q} = \bigcup_{i',q} \big((L_M(q) \setminus 0^*) \boxminus 1\big) \cap i'\Sigma_k^* = \left(\bigcup_q L_M(q) \setminus 0^*\right) \boxminus 1 = \big(\Sigma_k^* \setminus 0^*\big) \boxminus 1 = \Sigma_k^+ \setminus (k-1)^*$$

Then, as stated in Claim 13.1,

$$\bigcup_{i',q} \mu\big(U^{\lambda}_{i',q}\big) = \big(1\big(\textstyle\bigcup_{i',q} K^{\lambda}_{i',q}\big)10^*\big)_k = \big(1\big(\Sigma_k^+ \setminus (k-1)^*\big)10^*\big)_k,$$

which completes the proof. □

Similar statements shall now be proved for the other three groups of variables.

Claim 13.4. *The values of the variables from the group $\mathcal{W}^{\lambda}$ in the least solution are pairwise disjoint, and their union equals $(2(\Sigma_k^+ \setminus (k-1)^*)10^*)_k$.*

Proof. It is known [11, Eq. (4)] that

$$\mu\big(W^{\lambda}_{i',q}\big) = \big\{\big(2i'w10^{\ell}\big)_k \mid \ell \geqslant 0,\ w \in \Sigma_k^*,\ i'w \notin (k-1)^*,\ i'w \boxplus 1 \in L_M(q)\big\}.$$

These sets are almost the same as the values of $U^{\lambda}_{i',q}$ in Claim 13.3: the only difference is that each $W^{\lambda}_{i',q}$ has leading digit 2, instead of 1 in $U^{\lambda}_{i',q}$. Thus, the proof is the same as in Claim 13.3. □

Claim 13.5. *The values of the variables from $\mathcal{Y}^{\lambda}$ in the least solution are pairwise disjoint, and their union is $\bigcup_{i,i',q} \mu(Y^{\lambda}_{i,i',q}) = (1\Sigma_k(\Sigma_k^+ \setminus (k-1)^*)10^*)_k$.*

Proof. It was shown in the earlier paper [11, Eqs. (5, 6)] that

$$\mu\big(Y^{\lambda}_{i,i',q}\big) = \big\{\big(1ii'w10^{\ell}\big)_k \mid \ell \geqslant 0,\ w \in \Sigma_k^*,\ i'w \notin (k-1)^*,\ i'w \boxplus 1 \in L_M(q)\big\}.$$

Those sets are obtained from the languages $K^{\lambda}_{i,i',q} = (i(L_M(q) \setminus 0^*) \boxminus 1) \cap ii'\Sigma_k^*$, as in Claim 13.1 with $x = 1$ and $y = 1$. To see that the sets $K^{\lambda}_{i,i',q}$ are pairwise disjoint, let $(i_1, i'_1, q_1) \neq (i_2, i'_2, q_2)$: if $i_1 \neq i_2$ or $i'_1 \neq i'_2$, then $K^{\lambda}_{i_1,i'_1,q_1}$ and $K^{\lambda}_{i_2,i'_2,q_2}$ are distinguished by their leading digits, while for $q_1 \neq q_2$, this is implied by the fact that $L_M(q_1)$ and $L_M(q_2)$ are disjoint.

In order to calculate the union of all sets in this group by Claim 13.1, consider first the union of all languages $K^{\lambda}_{i,i',q}$.

$$\bigcup_{i,i',q} K^{\lambda}_{i,i',q} = \bigcup_{i,i',q} \big(i\big(L_M(q) \setminus 0^*\big) \boxminus 1\big) \cap ii'\Sigma_k^* = \bigcup_i i\left(\bigcup_{i',q} \big((L_M(q) \setminus 0^*) \boxminus 1\big) \cap i'\Sigma_k^*\right)$$

For the union over i' and q in the last expression, it was already established in the proof of Claim 13.3 that it is equal to $\Sigma_k^+ \setminus (k-1)^*$, and hence the expression can be further transformed as follows.

$$\bigcup_{i,i',q} K^{\lambda}_{i,i',q} = \bigcup_i i\big(\Sigma_k^+ \setminus (k-1)^*\big) = \Sigma_k\big(\Sigma_k^+ \setminus (k-1)^*\big)$$

Then, according to Claim 13.1,

$$\bigcup_{i,i',q} \mu(Y^{\lambda}_{i,i',q}) = \left(1\left(\bigcup_{i,i',q} K^{\lambda}_{i,i',q}\right)10^*\right)_k = \left(1\Sigma_k\left(\Sigma_k^+ \setminus (k-1)^*\right)10^*\right)_k. \qquad \square$$

Claim 13.6. *The values of the variables from the group $\mathcal{Z}^{\lambda}$ in the least solution are pairwise disjoint, and their union is $(1\Sigma_k(\Sigma_k^+ \setminus (k-1)^*)10^*)_k$.*

Proof. The equation (17) defines $Z^{\lambda}_{i,q}$ as the union of $Y^{\lambda}_{i,i',q}$ for all i', and the union of values of the latter variables is known from Claim 13.5. Then the value of $Z^{\lambda}_{i,q}$ is calculated as follows:

$$\bigcup_{i,q} \mu(Z^{\lambda}_{i,q}) = \bigcup_{i,i',q} \mu(Y^{\lambda}_{i,i',q}) = \left(1\Sigma_k\left(\Sigma_k^+ \setminus (k-1)^*\right)10^*\right)_k.$$

The sets $\mu(Z^{\lambda}_{i,q})$ are pairwise disjoint as unions of different pairwise disjoint sets. $\square$

The equations for ρ shall now undergo a similar reconstruction. Every expression $\rho_j(X_q)$ is replaced with a new variable $Z^{\rho}_{j,q}$. According to the original expression, $Z^{\rho}_{j,q}$ is defined as a union of k different sets corresponding to different digits $j' \in \Sigma_k$; every such set is denoted by $Y^{\rho}_{j,j',q}$. Furthermore, each $\pi_{j'}(X_q)$ is replaced by $W^{\rho}_{j',q}$. Each variable $U^{\rho}_{j',q}$ denotes an intermediate intersection of X_q with a constant. The new variables are defined by the following equations:

$$U^{\rho}_{j',q} = X_q \cap \left(1\Sigma_k^* j' 10^*\right)_k \tag{18}$$

$$W^{\rho}_{j',q} = \left(U^{\rho}_{j',q} + \left(10^*\right)_k\right) \cap \left(1\Sigma_k^* j' 20^*\right)_k \tag{19}$$

$$Y^{\rho}_{j,j',q} = \begin{cases} \left(W^{\rho}_{j',q} + ((k+j-2)10^*)_k\right) \cap (1\Sigma_k^* j 10^*)_k, & \text{for } j \in \{0,1\} \\ \left(W^{\rho}_{j',q} + (1(j-2)10^*)_k\right) \cap (1\Sigma_k^* j 10^*)_k, & \text{for } 2 \leqslant j \leqslant k-2 \\ \left(W^{\rho}_{j',q} + ((k-3)10^*)_k\right) \cap (1\Sigma_k^*(k-1)10^*)_k, & \text{for } j = k-1 \end{cases} \tag{20}$$

$$Z^{\rho}_{j,q} = \bigcup_{j'} Y^{\rho}_{j,j',q} \tag{21}$$

These variables are grouped as follows:

$$\mathcal{U}^{\rho} = \left\{U^{\rho}_{j',q} \mid j' \in \Sigma_k,\ q \in Q\right\},$$
$$\mathcal{W}^{\rho} = \left\{W^{\rho}_{j',q} \mid j' \in \Sigma_k,\ q \in Q\right\},$$
$$\mathcal{Y}^{\rho} = \left\{Y^{\rho}_{j,j',q} \mid j, j' \in \Sigma_k,\ q \in Q\right\},$$
$$\mathcal{Z}^{\rho} = \left\{Z^{\rho}_{j,q} \mid j \in \Sigma_k,\ q \in Q\right\}.$$

As in the case of λ, the values of the variables in each group are pairwise disjoint, and the union of each group is a set with a regular base-k representation.

Claim 13.7. *The values of the variables in the group $\mathcal{U}^{\rho}$ in the least solution are pairwise disjoint, and their union is $\bigcup_{j',q} \mu(U^{\rho}_{j',q}) = (1(\Sigma_k^+ \setminus (k-1)^*)10^*)_k$.*

Proof. It was proved [11, Eq. (8)] that

$$\mu(U^{\rho}_{j',q}) = \left\{\left(1wj'10^{\ell}\right)_k \mid \ell \geqslant 0,\ w \in \Sigma_k^*,\ wj' \notin (k-1)^*,\ wj' \boxplus 1 \in L_M(q)\right\}.$$

These sets can be obtained from the languages $K^{\rho}_{j',q} = ((L_M(q) \setminus 0^*) \boxminus 1) \cap \Sigma_k^* j'$, as in Claim 13.1 with $x = 1$ and $y = 1$. To see that the languages $K^{\rho}_{j'_1,q_1}$ and $K^{\rho}_{j'_2,q_2}$ are disjoint, let $(j'_1, q_1) \neq (j'_2, q_2)$ and consider two cases. If $j'_1 \neq j'_2$, then their last digits are different. For $q_1 \neq q_2$, the languages $K^{\rho}_{j'_1,q_1}$ and $K^{\rho}_{j'_2,q_2}$ are subsets of $(L_M(q_1) \setminus 0^*) \boxminus 1$ and $(L_M(q_2) \setminus 0^*) \boxminus 1$, respectively, and the latter two languages are disjoint. Therefore, by Claim 13.1, $\mu(U^{\rho}_{j'_1,q_1}) \cap \mu(U^{\rho}_{j'_2,q_2}) = \varnothing$ for $(j'_1, q_1) \neq (j'_2, q_2)$. The union of these sets is calculated by the usual method, beginning with the union of the languages $K^{\rho}_{j',q}$.

$$\bigcup_{j',q} K^{\rho}_{j',q} = \bigcup_{j',q} \left(\left(L_M(q) \setminus 0^*\right) \boxminus 1\right) \cap \Sigma_k^* j' = \bigcup_{q} \left(\left(L_M(q) \setminus 0^*\right) \boxminus 1\right) \cap \Sigma_k^+ = \left(\Sigma_k^* \setminus 0^*\right) \boxminus 1 = \Sigma_k^+ \setminus (k-1)^*$$

Then by Claim 13.1,

$$\bigcup_{j',q}\mu\big(U^{\rho}_{j',q}\big)=\left(1\left(\bigcup_{j',q}K^{\rho}_{j',q}\right)10^*\right)_k=\big(1\big(\Sigma_k^+\setminus(k-1)^*\big)10^*\big)_k,$$

as desired. □

Claim 13.8. *The values of the variables in $\mathcal{W}^{\rho}$ in the least solution are disjoint, and $\bigcup_{j',q}\mu(W^{\rho}_{j',q})=(1(\Sigma_k^+\setminus(k-1)^*)20^*)_k$.*

Proof. It was proved [11, Eq. (9)] that

$$\mu\big(W^{\rho}_{j',q}\big)=\big\{\big(1wj'20^{\ell}\big)_k\;\big|\;\ell\geqslant 0,\ w\in\Sigma_k^*,\ wj'\notin(k-1)^*,\ wj'\boxplus 1\in L_M(q)\big\}.$$

These sets are different from the values of $U^{\rho}_{j',q}$ only in the last non-zero digit (2 and 1, respectively), and the proof is the same as in Claim 13.7. □

Claim 13.9. *The variables in the group $\mathcal{Y}^{\rho}$ have pairwise disjoint values in the least solution, and the union in the group equals $\bigcup_{j,j',q}\mu(Y^{\rho}_{j,j',q})=(1((\Sigma_k^*\setminus 0^*)\Sigma_k\boxminus 1)10^*)_k$.*

Proof. Each variable $Y^{\rho}_{j,j',q}$, with $j,j'\in\Sigma_k$ and $q\in Q$, is known to have the following value in the least solution [11, Eqs. (10, 11, 12)].

$$\mu\big(Y^{\rho}_{j,j',q}\big)=\big\{\big(1\big(w'j'\boxplus 1\big)j10^{\ell-1}\big)_k\;\big|\;\ell\geqslant 1,\ w'\in\Sigma_k^*,\ w'j'\notin(k-1)^*,\ w'j'\boxplus 1\in L_M(q)\big\},\qquad\text{for all } j\neq k-1$$
$$\mu\big(Y^{\rho}_{k-1,j',q}\big)=\big\{\big(1w'j'(k-1)10^{\ell-1}\big)_k\;\big|\;\ell\geqslant 1,\ w'\in\Sigma_k^*,\ w'j'\notin(k-1)^*,\ w'j'\boxplus 1\in L_M(q)\big\},\qquad\text{for } j=k-1$$

The sets $\mu(Y^{\rho}_{j,j',q})$ are obtained from the languages

$$K^{\rho}_{j,j',q}=\big(L_M(q)\cap\big(\big(\Sigma_k^*j'\setminus(k-1)^*\big)\boxplus 1\big)\big)j,\qquad\text{for } j\neq k-1,$$
$$K^{\rho}_{k-1,j',q}=\big(\big(L_M(q)\setminus 0^*\big)\boxminus 1\big)(k-1)\cap\Sigma_k^*j'(k-1),\qquad\text{for } j=k-1,$$

as in Claim 13.1, with $x=1$ and $y=1$.

To see that all these languages are pairwise disjoint, let $(j_1,j_1',q_1)\neq(j_2,j_2',q_2)$. Then, if $j_1\neq j_2$, the strings end with different digits. Assume that $j_1=j_2=j$, and first consider the case when $j<k-1$. For $q_1\neq q_2$, it holds that $K^{\rho}_{j,j_1',q_1}\subseteq L_M(q_1)$ and $K^{\rho}_{j,j_1',q_1}\subseteq L_M(q_2)$, and the supersets are disjoint. If $j_1'\neq j_2'$, then the strings from these languages differ in the second-last digit, which are $(j_1'+1)\bmod k$ and $(j_2'+1)\bmod k$, respectively. In the remaining case of $j_1=j_2=k-1$, if $q_1\neq q_2$, then the sets K^{ρ}_{k-1,j_1',q_1} and K^{ρ}_{k-1,j_1',q_1} are subsets of the sets $(L_M(q_1)\setminus 0^*)\boxminus 1$ and $(L_M(q_2)\setminus 0^*)\boxminus 1$, which are disjoint; and if $j_1'\neq j_2'$, then the sets are distinguished by the second leading digit.

In order to calculate the union of all these languages, first fix $j<k-1$ and consider the union of $K^{\rho}_{j,j',q}$ for all $j'\in\Sigma_k$ and $q\in Q$.

$$\bigcup_{j',q}K^{\rho}_{j,j',q}=\bigcup_{j',q}\big(L_M(q)\cap\big(\big(\Sigma_k^*j'\setminus(k-1)^*\big)\boxplus 1\big)\big)j=\bigcup_{j'}\big(\big(\Sigma_k^*j'\setminus(k-1)^*\big)\boxplus 1\big)j$$
$$=\big(\big(\Sigma_k^+\setminus(k-1)^*\big)\boxplus 1\big)j=\big(\Sigma_k^+\setminus 0^*\big)j=\big(\Sigma_k^+\setminus 0^*\big)(j+1)\boxminus 1$$

Next, for $j=k-1$, the union of $K^{\rho}_{j,j',q}$ over all j' and q is calculated similarly to the union in Claim 13.7.

$$\bigcup_{j',q}K^{\rho}_{k-1,j',q}=\big(\Sigma_k^+\setminus(k-1)^*\big)(k-1)=\big(\Sigma_k^+\setminus 0^*\big)0\boxminus 1$$

Then the union of all languages $K^{\rho}_{j,j',q}$ is determined as follows.

$$\bigcup_{j,j',q}K^{\rho}_{j,j',q}=\bigcup_{j\neq k-1}\big(\big(\Sigma_k^+\setminus 0^*\big)(j+1)\boxminus 1\big)\cup\big(\big(\Sigma_k^+\setminus 0^*\big)0\boxminus 1\big)=\big(\Sigma_k^+\setminus 0^*\big)\Sigma_k\boxminus 1$$

Finally, by Claim 13.1, the sets $\mu(Y^{\rho}_{j,j',q})$ are pairwise disjoint, and their union is

$$\bigcup_{j,j',q}\mu\big(Y^{\rho}_{j,j',q}\big)=\big(1\big(\big(\Sigma_k^+\setminus 0^*\big)\Sigma_k\boxminus 1\big)10^*\big)_k.\qquad\square$$

Claim 13.10. *The sets represented by the variables in the group $\mathcal{Z}^\rho$ are disjoint, and their union equals $\bigcup_{j,q} \mu(Z^\rho_{j,q}) = (1((\Sigma_k^* \setminus 0^*)\Sigma_k \boxminus 1)10^*)_k$.*

Proof. Each variable $Z^\rho_{j,q}$ is defined by the equation (21) as the union of $Y^\rho_{j,j',q}$ for all j'. Then

$$\bigcup_{j,q} \mu(Z^\rho_{j,q}) = \bigcup_{j,j',q} \mu(Y^\rho_{j,j',q}) = \big(1\big(\big(\Sigma_k^* \setminus 0^*\big)\Sigma_k \boxminus 1\big)10^*\big)_k,$$

where the second equality is given by Claim 13.9. The latter claim also states that the sets $\mu(Y^\rho_{j,j',q})$ are pairwise disjoint, and hence so are the sets $\mu(Z^\rho_{j,q})$, as they are unions of different pairwise disjoint sets. □

Thus the expressions $\lambda_i(X_q)$ and $\rho_j(X_q)$ have been transcribed in equations of the form satisfying the assumptions of Lemma 7 and Lemma 8. It remains to transform the equation defining X_q to the same form. The original equation [11] was

$$X_q = R_q \cup \bigcup_{\substack{p,r:\ \delta(p,r)=q \\ i,j\in\Sigma_k}} \lambda_i(X_r) \cap \rho_j(X_p).$$

The subexpression corresponding to each quadruple (i,r,j,p) shall be represented by a new variable $X_{i,r,j,p}$ with the equation

$$X_{i,r,j,p} = Z^\lambda_{i,q''} \cap Z^\rho_{j,q'}, \tag{22}$$

while the equation for X_q is accordingly replaced by

$$X_q = R_q \cup \bigcup_{\substack{p,r:\ \delta(p,r)=q \\ i,j\in\Sigma_k}} X_{i,r,j,p}. \tag{23}$$

The variables are divided into two groups, $\mathcal{X}$ and $\mathcal{X}_4$, with

$$\mathcal{X}_4 = \{X_{i,r,j,p} \mid i,j \in \Sigma_k,\ p,r \in Q\},$$
$$\mathcal{X} = \{X_q \mid q \in Q\}.$$

It remains to show the required properties of the variables in each of these two groups.

Claim 13.11. *The values of the sets from $\mathcal{X}_4$ in the least solution are pairwise disjoint, and the union of all these sets is $\bigcup_{i,r,j,p} \mu(X_{i,r,j,p}) = (1((\Sigma_k(\Sigma_k^+ \setminus 0^*) \cap (\Sigma_k^+ \setminus 0^*)\Sigma_k) \boxminus 1)10^*)_k$.*

Proof. It is to be proved that for any two distinct quadruples (i_1,r_1,j_1,p_1) and (i_2,r_2,j_2,p_2), the sets $\mu(X_{i_1,r_1,j_1,p_1})$ and $\mu(X_{i_2,r_2,j_1,p_2})$ are disjoint. According to the equation (22), $\mu(X_{i_1,r_1,j_1,p_1}) = \mu(Z^\lambda_{i_1,r_1}) \cap \mu(Z^\rho_{j_1,p_1})$ and $\mu(X_{i_2,r_2,j_2,p_2}) = \mu(Z^\lambda_{i_2,r_2}) \cap \mu(Z^\rho_{j_2,p_2})$.

Observe that if (i_1,r_1,j_1,p_1) and (i_2,r_2,j_2,p_2) are distinct, then either $(i_1,r_1) \neq (i_2,r_2)$ or $(j_1,p_1) \neq (j_2,p_2)$. In the former case, by Claim 13.6, $\mu(Z^\lambda_{i_1,r_1}) \cap \mu(Z^\lambda_{i_2,r_2}) = \varnothing$, and therefore the intersection $\mu(X_{i_1,r_1,j_1,p_1}) \cap \mu(X_{i_2,r_2,j_2,p_2})$ is empty as well. Similarly, in the latter case, by Claim 13.10, $\mu(Z^\rho_{j_1,p_1}) \cap \mu(Z^\rho_{j_2,p_2}) = \varnothing$ and consequently $\mu(X_{i_1,r_1,j_1,p_1}) \cap \mu(X_{i_2,r_2,j_2,p_2}) = \varnothing$.

By the equation (22), the union of all these sets is

$$\bigcup_{i,r,j,p} \mu(X_{i,r,j,p}) = \bigcup_{i,r,j,p} \mu(Z^\lambda_{i,r}) \cap \mu(Z^\rho_{j,p}) = \Big(\bigcup_{i,r} \mu(Z^\lambda_{i,r})\Big) \cap \Big(\bigcup_{j,p} \mu(Z^\rho_{j,p})\Big).$$

The values of both unions were already determined in Claim 13.6 and in Claim 13.10, as follows.

$$\bigcup_{i,r} \mu(Z^\lambda_{i,r}) = \big(1K_\lambda 10^*\big)_k, \quad \text{where } K_\lambda = \Sigma_k\big(\Sigma_k^+ \setminus (k-1)^*\big)$$

$$\bigcup_{j,p} \mu(Z^\rho_{j,p}) = \big(1K_\rho 10^*\big)_k. \quad \text{where } K_\rho = \big(\Sigma_k^+ \setminus 0^*\big)\Sigma_k \boxminus 1$$

By arguments similar to those in Claim 13.1, $(1K_\lambda 10^*)_k \cap (1K_\rho 10^*)_k = (1(K_\lambda \cap K_\rho)10^*)_k$. In order to calculate the intersection of these two languages, it is useful to reformulate K_λ as $\Sigma_k(\Sigma_k^+ \setminus 0^*) \boxminus 1$. Then

$$K_\lambda \cap K_\rho = \big[\Sigma_k\big(\Sigma_k^+ \setminus 0^*\big) \boxminus 1\big] \cap \big[\big(\Sigma_k^+ \setminus 0^*\big)\Sigma_k \boxminus 1\big] = \big[\Sigma_k\big(\Sigma_k^+ \setminus 0^*\big) \cap \big(\Sigma_k^+ \setminus 0^*\big)\Sigma_k\big] \boxminus 1,$$

and the union of all sets $\mu(X_{i,r,j,p})$ is

$$\Big(\bigcup_{i,r} \mu\big(Z^\lambda_{i,r}\big)\Big) \cap \Big(\bigcup_{j,p} \mu\big(Z^\rho_{j,p}\big)\Big) = \big(1(K_\lambda \cap K_\rho)10^*\big)_k = \big(1\big(\big(\Sigma_k\big(\Sigma_k^+ \setminus 0^*\big) \cap \big(\Sigma_k^+ \setminus 0^*\big)\Sigma_k\big) \boxminus 1\big)10^*\big)_k,$$

which concludes the proof. □

Since the new equations represent the subexpressions of the original system, the values of the common variables X_q, with $q \in Q$, in the least solution remain the same, that is, $\mu(X_q) = S_q$. Moreover, Claim 13.2 asserts that the sets S_q are pairwise disjoint and that their union is a set with a regular base-k representation. Thus, the only thing remaining to be checked in order to apply Lemmata 7 and 8 is that there are no cyclic chain dependencies in the given system.

Claim 13.12. *There are no cyclic chain dependencies in the equations (14)–(23).*

Proof. The constructed system contains the following chain dependencies:

- there may be a chain dependency of $U^\lambda_{i',q}$ on X_q, or of $U^\rho_{j',q}$ on X_q,
- of X_q on (some) $X_{i,r,j,p}$,
- of $X_{i,r,j,p}$ on $Z^\lambda_{i,r}$ and on $Z^\rho_{i,p}$,
- of $Z^\lambda_{i,q}$ on $Y^\lambda_{i,i',q}$, or of $Z^\rho_{j,q}$ on $Y^\rho_{j,j',q}$.

In this way, the groups of variables can be stratified into the following five levels:

$$\mathcal{U}^\lambda \cup \mathcal{U}^\rho, \quad \mathcal{X}, \quad \mathcal{X}_4, \quad \mathcal{Z}^\lambda \cup \mathcal{Z}^\rho, \quad \mathcal{Y}^\lambda \cup \mathcal{Y}^\rho,$$

with $\mathcal{U}^\lambda \cup \mathcal{U}^\rho$ forming the bottom level and $\mathcal{Y}^\lambda \cup \mathcal{Y}^\rho$ forming the top one; the variables from $\mathcal{W}^\lambda$ and $\mathcal{W}^\rho$ are not on this list, because they are not involved in any chain dependencies. Then, according to the above list of dependencies, a variable from a lower level may have a chain dependency only on a variable from a higher level, but not the other way around. Therefore, there are no chain dependencies in the system. □

According to the above claims, there exists a resolved system of equations satisfying the assumption of Lemmata 7 and 8, such that one of the components in its least solution is

$$\big(1\big(L_M(q) \boxminus 1\big)10^*\big)_k = \big\{\big(1w10^\ell\big)_k \,\big|\, \ell \geqslant 0,\ w \notin (k-1)^*,\ w \boxplus 1 \in L_M(q)\big\}.$$

Then, by Lemma 7, there exists an unresolved system with union and addition, which has the same unique solution. Similarly, Lemma 8 asserts the existence of another unresolved system with intersection and addition, again with the same unique solution. Finally, using Theorem 2, regular constants used in these systems are replaced by the singleton constant $\{1\}$. This completes the proof of Lemma 13. □

The next task is to represent all sets of the form $(1L_M(q))_k$, where M is a trellis automaton and q is its state. Similarly to Lemma 13, this will be done by transforming an existing construction [11, Lem. 6] in order to apply Lemmata 7 and 8.

Lemma 14. *For every base $k \geqslant 4$ and for every trellis automaton M over the alphabet Σ_k, there exists and can be effectively constructed an unresolved system of equations over sets of numbers using the operations of union (or intersection) and addition, as well as constant $\{1\}$, such that its unique solution contains a component $(1L(M))_k$.*

Proof. The argument will use a simple auxiliary claim, similar to Claim 13.1 in the proof of Lemma 13.

Claim 14.1. *Let $x \in \Sigma_k^+ \setminus 0\Sigma_k^*$ and $y \in \Sigma_k^*$ be strings of digits, let $K_1, \ldots, K_m \subseteq \Sigma_k^+$ be any pairwise disjoint languages, and let $S_1, \ldots, S_m \subseteq \mathbb{N}$ be sets of numbers defined by*

$$S_t = \big\{(xuy)_k \,\big|\, u \in K_t\big\}.$$

Then these sets are pairwise disjoint and their union is

$$\bigcup_{t=1}^m S_t = \big(x\big(\bigcup\nolimits_{t=1}^m K_t\big)y\big)_k.$$

The proof is omitted; a stronger statement will be proved in the following as Claim 3.1.

Consider the given trellis automaton M over Σ_k, let Q by its set of states. The proof of Lemma 14 proceeds in three steps. At the first step, the goal is to define the sets $(1(((L_M(q)\cdot j^{-1})\setminus 0^*)\boxminus 1)10^*)_k$, for all states $q\in Q$ and for all digits $j\in\Sigma_k$; this is done using multiple constants obtained from Lemma 13. At the second step, the earlier defined sets are used as constants in the definition of the sets $(1\cdot L_M(q))_k$, for all states $q\in Q$. These sets are in turn used as constants at the final third step, when the desired set $(1L(M))_k$ is defined.

At **the first step**, for every state q and for every digit $j\in\Sigma_k$, construct a trellis automaton $M_{q,j}$ recognizing the language $L_M(q)\cdot j^{-1}=\{w\in\Sigma_k^*\mid wj\in L_M(q)\}$ using the known transformation [24]. Throughout this first step of the argument, both q and j are fixed, and all constructions revolve around this language and the sets of numbers derived from it.

Let $Q_{q,j}$ be the set of states of the automaton $M_{q,j}$. By Lemma 13, there is a system of equations using addition and either union or intersection, which contains a variable $Y_{q,j,p}$ for each state p of $M_{q,j}$, and has a unique solution, with

$$Y_{q,j,p}=\big(1\big(\big(L_{M_{q,j}}(p)\setminus 0^*\big)\boxminus 1\big)10^*\big)_k.$$

Consider a single system of equations obtained by combining these systems for $Y_{q,j,p}$ over all p. The first goal is to extend it with extra variable $Y_{q,j}$ so that it has $Y_{q,j}=(1((L(M_{q,j})\setminus 0^*)\boxminus 1)10^*)_k$ in its unique solution.

When union and addition are allowed, the construction is immediate: let $F_{q,j}$ is the set of accepting states of $M_{q,j}$, and define variable $Y_{q,j}$ by the equation

$$Y_{q,j}=\bigcup_{p\in F_{q,j}}Y_{q,j,p}. \tag{24}$$

If the allowed operations are intersection and addition, the plan is to fix the value of $Y_{q,j}$ modulo each set $Y_{q,j,p}$, with $p\in Q_{q,j}$, as well as modulo the set of all numbers not in $Y_{q,j,p}$ for any p. This is done in the following system of equations.

$$Y_{q,j}\cap Y_{q,j,p}=\varnothing, \qquad \text{for } p\notin F_{q,j} \tag{25a}$$

$$Y_{q,j}\cap Y_{q,j,p}=Y_{q,j,p}, \qquad \text{for } p\in F_{q,j} \tag{25b}$$

$$Y_{q,j}\cap\big[\mathbb{N}\setminus\big(1\big(\big(\Sigma_k^+\setminus 0^*\big)\boxminus 1\big)10^*\big)_k\big]=\varnothing \tag{25c}$$

To see that the equations (25) are equivalent to (24), it is sufficient to demonstrate that the sets $Y_{q,j,p}$ are pairwise disjoint, and their union is exactly the set complemented in the equation (25c). In terms of Claim 13.1, the sets $Y_{q,j,p}$, for all p, are obtained from the languages $(L_{M_{q,j}}(p)\setminus 0^*)\boxminus 1$. By the definition of a trellis automaton, these languages are pairwise disjoint and their union is $(\Sigma_k^+\setminus 0^*)\boxminus 1$. This allows applying Claim 13.1 to these languages, for all p, and it asserts that the corresponding sets $Y_{q,j,p}$ are also pairwise disjoint, and their union is exactly $(1((\Sigma_k^+\setminus 0^*)\boxminus 1)10^*)_k$. Thus, the equations (25a), (25b) fix the value of $Y_{q,j}$ on all numbers in $(1((\Sigma_k^+\setminus 0^*)\boxminus 1)10^*)_k$, whereas the last equation (25c) ensures that no other numbers are in $Y_{q,j}$.

This proves that, for each state $q\in Q$ and for each digit $j\in\Sigma_k$, the variable $Y_{q,j}$ defines the set

$$Y_{q,j}=\big(1\big(\big(L(M_{q,j})\setminus 0^*\big)\boxminus 1\big)10^*\big)_k=\big(1\big(\big(\big(L_M(q)\cdot j^{-1}\big)\setminus 0^*\big)\boxminus 1\big)10^*\big)_k.$$

At **the second step** of the proof, these sets are used as constants in the following equation defining the desired set $(1\cdot L_M(q))_k$, which comes from the authors' earlier paper [11, Lem. 6].

$$Z_q=C_q\cup\bigcup_{j=0}^{k-1}\big(\big(Y_{q,j}\cap\big(1\Sigma_k^*1\big)_k\big)+\big(1j\boxminus 1\big)_k\big)$$

The equation uses a constant $C_q=(1L_M(q))_k\cap(10^*\Sigma_k)_k$, with a regular base-$k$ representation, which takes care of all strings of digits of a simple form not handled by the main formula. These constants are similar to the constants R_q in Lemma 13.

This equation is modified to match Lemmata 7 and 8 by introducing new variables $V_{q,j}$ and rewriting it as follows.

$$V_{q,j}=Y_{q,j}\cap\big(1\Sigma_k^*1\big)_k$$

$$Z_q=C_q\cup\bigcup_{j=0}^{k-1}\big(V_{q,j}+(1j\boxminus 1)_k\big)$$

These equations express the set $(1\cdot L_M(q))_k$ in two steps. First, an intersection with the set $(1\Sigma_k^*1)_k$ removes from each $Y_{q,j}$ all encodings that have at least one trailing zero, so that $V_{q,j}=(1(((L_M(q)\cdot j^{-1})\setminus 0^*)\boxminus 1)1)_k$. The sum with $(1j\boxminus 1)_k$ replaces the right delimiting 1 with j, and at the same time adds 1 to the encoded set, thus representing the set $(1((L_M(q)\cdot j^{-1})\setminus 0^*)j)_k$, which equals $(1(L_M(q)\cap(\Sigma_k^*\setminus 0^*)j))_k$. The subsequent union over $j\in\Sigma_k$ yields the set

$(1(L_M(q) \cap (\Sigma_k^* \setminus 0^*)\Sigma_k))_k$: informally speaking, this is *almost* the set $(1 \cdot L_M(q))_k$, and the missing numbers are provided in the constant $C_q = (1(L_M(q) \cap 0^*\Sigma_k))_k$.

Now the system of all equations for Z_q and $V_{q,j}$, for $q \in Q$ and $j \in \Sigma_k$, satisfies the assumptions of Lemma 7 and Lemma 8. The required grouping of variables is

$$\mathcal{Z} = \{Z_q \mid q \in Q\}, \qquad \mathcal{V}_j = \{V_{q,j} \mid q \in Q\}, \quad \text{for } j \in \Sigma_k.$$

It remains be proved that the sets in each group form a disjoint partition of a certain set with a regular notation.

Claim 14.2. *For each $j \in \Sigma_k$, the values of the variables from $\mathcal{V}_j$ in the least solution are pairwise disjoint and their union is $\bigcup_q \mu(V_{q,j}) = (1(\Sigma_k^* \setminus (k-1)^*)1)_k$.*

Proof. It needs to be shown that for every $j \in \Sigma_k$ and for all $q_1 \neq q_2$, the sets $\mu(V_{q_1,j})$ and $\mu(V_{q_2,j})$ are disjoint. The value of $V_{q,j}$ is determined from its equation as follows.

$$\begin{aligned}\mu(V_{q,j}) = \mu(Y_{q,j}) \cap \left(1\Sigma_k^*1\right)_k &= \left(1\left(\left(\left(L_M(q) \cdot j^{-1}\right) \setminus 0^*\right) \boxminus 1\right)10^*\right)_k \cap \left(1\Sigma_k^*1\right)_k \\ &= \left(1\left(\left(\left(L_M(q) \cdot j^{-1}\right) \setminus 0^*\right) \boxminus 1\right)1\right)_k\end{aligned}$$

For any fixed digit j, the sets $\mu(V_{q,j})$ over all q satisfy the assumption of Claim 14.1 with $K_q = ((L_M(q) \cdot j^{-1}) \setminus 0^*) \boxminus 1$ and $x = y = 1$. For $q_1 \neq q_2$, the languages $L_M(q_1)$ and $L_M(q_2)$ are disjoint, and hence the sets $K_{q_1} = ((L_M(q_1) \cdot j^{-1}) \setminus 0^*) \boxminus 1$ and $K_{q_2} = ((L_M(q_2) \cdot j^{-1}) \setminus 0^*) \boxminus 1$ are disjoint as well. Then $\mu(V_{q_1,j}) \cap \mu(V_{q_2,j}) = \varnothing$ by Claim 14.1.

Finally,

$$\bigcup_q \mu(V_{q,j}) = \left(1\left(\bigcup_q K_q\right)1\right)_k = \left(1\left(\Sigma_k^* \setminus (k-1)^*\right)1\right)_k,$$

since every string outside $(k-1)^*$ belongs to some K_q. □

Claim 14.3. *The values of the variables from $\mathcal{Z}$ in the least solution are pairwise disjoint and their union is $\bigcup_q \mu(Z_q) = (1\Sigma_k^+)_k$.*

Proof. Since the new system is the same as the previously known one [11, Lem. 6], with some intermediate terms named as variables, its least solution remains the same as in the original system, that is, $\mu(Z_q) = (1L_M(q))_k$. Then $\mu(Z_q)$ satisfies the assumption of Claim 14.1 with $K'_q = L_M(q)$, $x = 1$ and $y = \varepsilon$. Clearly, the languages $\{K'_q\}$ are pairwise disjoint, as trellis automata are deterministic. Also, each non-empty string belongs to some K'_q, hence $\bigcup_{q \in Q} K'_q = \Sigma_k^+$. Therefore, by Claim 14.1, $\mu(Z_{q_1}) \cap \mu(Z_{q_2}) = \varnothing$ for $q_1 \neq q_2$ and

$$\bigcup_q \mu(Z_q) = \left(1\left(\bigcup_{q \in Q} K'_q\right)\right)_k = \left(1\Sigma_k^+\right)_k,$$

as claimed. □

Lemma 7 and Lemma 8 require that there are no cyclic chain dependencies in the constructed system. As the only chain dependencies are those of Z_q on (some) variables $V_{q,j}$, there are no cycles among them. Therefore, the new system satisfies the assumptions of Lemmata 7 and 8, and accordingly, there exists an unresolved system using addition and either union or intersection, which has a unique solution with $(1 \cdot L_M(q))_k$ as one of its components. The system uses constants with regular base-k representation, which can be eliminated using Theorem 2, and constants $Y_{q,j}$, which are expressed in (24) for the case of union and addition, and in (25a)–(25c) using intersection and addition.

As the last **third step** of the argument, it is left to write an equation for the variable Z that defines the set $(1 \cdot L(M))_k$. If union is allowed, this is done simply by taking a union over all accepting states:

$$Z = \bigcup_{q \in F} Z_q,$$

where F is the set of accepting states of the trellis automaton. If the only allowed Boolean operation is intersection, then the following equations express that $Z \cap Z_q$ should be empty for $q \notin F$ and equal to Z_q for $q \in F$.

$$\begin{aligned} &Z \cap Z_q = \varnothing, && \text{for } q \notin F \\ &Z \cap Z_q = Z_q, && \text{for } q \in F \\ &Z \cap \left(\mathbb{N} \setminus \left(1\Sigma_6^+\right)_k\right) = \varnothing \end{aligned}$$

Since the values of Z_q form a partition of $(1\Sigma_6^+)_k$, and the last equation handles all remaining numbers, this establishes that $Z = \bigcup_{q \in F} Z_q = (1L(M))_k$, as in the similar argument in the first step of this proof. □

The final step of the known construction for simulating trellis automata by equations over sets of numbers [11] was to define the set $(L)_k$, for any given linear conjunctive language $L \subseteq \Sigma_k^*$, with the least assumption on its form: namely, that it contains no strings beginning with a zero. This step will now be similarly replicated using unresolved systems, leading to the following new version of Theorem C for unresolved systems using only one Boolean operation.

Theorem 3. *For every $k \geqslant 4$ and for every trellis automaton M over Σ_k, which satisfies $L(M) \cap 0\Sigma_k^* = \varnothing$, there exists and can be effectively constructed such an unresolved system of equations over sets of numbers using the operations of union (or intersection) and addition, as well as constant $\{1\}$, that one of the components of its unique solution is $(L(M))_k$.*

Proof. The following slightly elaborated version of Claim 14.1 will be used in the proof:

Claim 3.1. *Let $x \in \Sigma_k^+ \setminus 0\Sigma_k^*$ and $y, z \in \Sigma_k^*$ be strings of digits, let $K_1, \ldots, K_m \subseteq \Sigma_k^+$ be any pairwise disjoint languages, and let $S_1, \ldots, S_m \subseteq \mathbb{N}$ be sets of numbers defined by*

$$S_t = \left\{ \left(x\left(z^{-1}u\right)y\right)_k \mid u \in K_t \right\}.$$

Then these sets are pairwise disjoint and their union is

$$\bigcup_{t=1}^{m} S_t = \left(x\left(z^{-1}\bigcup_{t=1}^{m} K_t\right)y\right)_k.$$

Proof. Let S_t and $S_{t'}$ be any two sets with $t \neq t'$, and suppose there is a number n belonging to both of them. Then $n = (x(z^{-1}u)y)_k$ for some $u \in K_t$, and $n = (x(z^{-1}u')y)_k$ for some $u' \in K_{t'}$. Since z is a prefix of both u and u', let $u = zv$ and $u' = zv'$. Then $n = (xvy)_k$ and $n = (xv'y)_k$, and therefore $v' = v$, which in turn implies that $u = u'$. This is a contradiction, as K_t and $K_{t'}$ are disjoint, and it proves that $S_t \cap S_{t'} = \varnothing$.

The union of these sets is

$$\bigcup_t S_t = \bigcup_t \left(x\left(z^{-1}K_t\right)y\right)_k = \left(x\left(z^{-1}\bigcup_t K_t\right)y\right)_k,$$

as desired. □

Let Q be the set of states of M. For every digit $i \in \Sigma_k$ and for every state $q \in Q$, consider the language $i^{-1}L_M(q) \subseteq \Sigma_k^*$. By Lemma 14, there is a system of equations defining the set of numbers $(1(i^{-1}L_M(q)))_k$; let $Z_{i,q}$ be the variable that defines this set. The goal is to use these sets as constants to define the sets $(L_M(q) \setminus 0\Sigma_k^*)_k$, for all $q \in Q$.

Let $q \in Q$, and consider the following resolved equation that defines a new variable T_q [11, Lem. 7]:

$$T_q = \left(L_M(q) \cap \left(\Sigma_k \setminus \{0\}\right)\right)_k \cup Z_{1,q} \cup \bigcup_{i \in \Sigma_k \setminus \{0,1\}} \tau_i(Z_{i,q}), \quad \text{where}$$

$$\tau_i(X) = \bigcup_{i' \in \Sigma_k} \left(\left[\left(X \cap \left(1i'\Sigma_k^*\right)_k\right) + \left((i-1)0^*\right)_k\right] \cap \left(ii'\Sigma_k^*\right)_k\right) \quad (\text{for } i \neq 0, 1).$$

The purpose of the expression $\tau_i(X)$ is to transform each number $(1w)_k$ in X to the number $(iw)_k$:

$$\tau_i\left(\{n\}\right) = \begin{cases} \{(iw)_k\}, & \text{if } n = (1w)_k \text{ for some } w \in \Sigma_k^+, \\ \varnothing, & \text{otherwise.} \end{cases}$$

This expression is intended to be applied as $\tau_i(Z_{i,q})$, thus transforming the set $(1(i^{-1}L_M(q)))_k$ to the set $(i(i^{-1}L_M(q)))_k = (L_M(q) \cap i\Sigma_k^+)_k$. The subsequent union over i yields the set $(L_M(q) \cap (\Sigma_k \setminus \{0,1\})\Sigma_k^+)_k$, and the rest of the equation for T_q handles the numbers beginning with 1 and one-digit numbers. This shows that the above equation for T_q has a least solution, with

$$\mu(T_q) = \left(L_M(q) \setminus 0\Sigma_k^*\right)_k$$

(see the cited paper [11, Lem. 7] for a complete calculation).

In order to fit the requirements of Lemmata 7 and 8, the equation for each T_q is transformed by decomposing the subexpression $\tau_i(Z_{i,q})_q$. The variable $U_{i,i',q}$ denotes the initial intersection of $Z_{i,q}$ with a set $(1i'\Sigma_k^*)_k$. Then the whole expression $\tau_i(Z_{i,q})$ is represented as a union (over i') of subexpressions denoted by $W_{i,i',q}$.

$$\begin{aligned}
U_{i,i',q} &= Z_{i,q} \cap \left(1i'\Sigma_k^*\right)_k && \text{for } i \geqslant 2,\ j \in \Sigma_k,\ q \in Q \\
W_{i,i',q} &= \left[U_{i,i',q} + \left((i-1)0^*\right)_k\right] \cap \left(ii'\Sigma_k^*\right)_k && \text{for } i \geqslant 2,\ j \in \Sigma_k,\ q \in Q \\
T_q &= \left(L_M(q) \cap \left(\Sigma_k \setminus \{0\}\right)\right)_k \cup Z_{1,q} \cup \bigcup_{i \geqslant 2,\ i' \in \Sigma_k} W_{i,i',q} && \text{for } q \in Q
\end{aligned}$$

Let the set of variables be split into the following $k-1$ groups.

$$\mathcal{U}_i = \left\{ U_{i,i',q} \mid i' \in \Sigma_k,\ q \in Q \right\}, \quad \text{for } 2 \leqslant i < k$$
$$\mathcal{W} = \left\{ W_{i,i',q} \mid i \in \{2,\ldots,k-1\},\ i' \in \Sigma_k,\ q \in Q \right\}$$
$$\mathcal{T} = \{ T_q \mid q \in Q \}$$

As in the previous proofs, it is claimed that for each of these groups, the union of values of its variables in the least solution is a set with a regular base-k representation, and that these values are pairwise disjoint. For each variable X in this system, denote by $\mu(X)$ the value of this variable in the least solution of the new system of equations; this is the same notation as in the proof of Lemma 13.

First, fix $i \geqslant 2$ and consider the group $\mathcal{U}_i$. It is known from the previous work [11, Eq. (16)] that

$$\mu(U_{i,i',q}) = \left\{ \left(1 i' w\right)_k \mid w \in \Sigma_k^*,\ ii'w \in L_M(q) \right\}.$$

The sets $\{\mu(U_{i,i',q})\}$, for all $i' \in \Sigma_k$ and $q \in Q$, satisfy the assumption of Claim 3.1 with $K_{i',q} = ii'\Sigma_k^* \cap L_M(q)$, $x = 1$, $y = \varepsilon$ and $z = i$. In order to use the claim, it remains to check that the intersection of any two distinct sets $K_{i'_1,q_1}$, $K_{i'_2,q_2}$ from this group is empty. Let $(i'_1,q_1) \neq (i'_2,q_2)$, and consider that if $i'_1 \neq i'_2$, then $ii'_1\Sigma_k^* \cap ii'_2\Sigma_k^* = \varnothing$ due to a different second digit, and if $q_1 \neq q_2$, then $L_M(q_1) \cap L_M(q_2) = \varnothing$, because trellis automata are deterministic. Therefore, Claim 3.1 asserts that the sets $\mu(U_{i,i'_1,q_1})$ are pairwise disjoint, and furthermore,

$$\bigcup_{i',q} \mu(U_{i,i',q}) = \left(1\left(i^{-1} \bigcup_{i',q} K_{i',q}\right)\right)_k = \left(1\left(i^{-1} i \Sigma_k^+\right)\right)_k = \left(1\Sigma_k^+\right)_k \quad \text{for each } i \geqslant 2.$$

Consider now the group of variables $\mathcal{W}$. It is also known [11, Eq. (17)] that

$$\mu(W_{i,i',q}) = \left\{ \left(ii' w\right)_k \mid w \in \Sigma_k^*,\ ii'w \in L_M(q) \right\} = \left(L_M(q)\right)_k \cap \left(ii'\Sigma_k^*\right)_k.$$

These sets are based on the languages $L_M(q) \cap ii'\Sigma_k^*$ of their base-k representations. The proof of their pairwise disjointness and the calculation of their union goes through Claim 3.1 as in the previous case. Consider any two variables W_{i_1,i'_1,q_1} and W_{i_2,i'_2,q_2} from $\mathcal{W}$, with $(i_1,i'_1,q_1) \neq (i_2,i'_2,q_2)$. The corresponding base-k languages are disjoint: if $q_1 \neq q_2$, then $L_M(q_1) \cap L_M(q_2) = \varnothing$, and if $(i_1,i'_1) \neq (i_2,i'_2)$, then $i_1 i'_1 \Sigma_k^* \cap i_2 i'_2 \Sigma_k^* = \varnothing$. In both cases $\mu(W_{i_1,i'_1,q_1}) \cap \mu(W_{i_2,i'_2,q_2}) = \varnothing$. The union of variables from the group $\mathcal{W}$ in the least solution equals

$$\bigcup_{i \geqslant 2,\ i' \geqslant 0,\ q} \mu(W_{i,i',q}) = \bigcup_{i \geqslant 2,\ i' \geqslant 0,\ q} \left(L_M(q)\right)_k \cap \left(ii'\Sigma_k^*\right)_k = \left(\left(\Sigma_k \setminus \{0,1\}\right)\Sigma_k^+\right)_k.$$

Lastly, consider the group of variables $\mathcal{T}$. As the new equations for T_q represent subexpressions of the original resolved equation, it follows that $\mu(T_q) = (L_M(q) \setminus 0\Sigma_k^*)_k$. Thus, for every two distinct states $q_1 \neq q_2$,

$$\mu(T_{q_1}) \cap \mu(T_{q_2}) \subseteq \left(L_M(q_1)\right)_k \cap \left(L_M(q_2)\right)_k = \varnothing,$$

while the union of all these sets is

$$\bigcup_q \mu(T_q) = \bigcup_q \left(L_M(q) \setminus 0\Sigma_k^*\right)_k = \left(\Sigma_k^+ \setminus 0\Sigma_k^*\right)_k = \mathbb{N} \setminus \{0\}.$$

The only chain dependencies in the constructed system are those of T_q on $W_{i,i',q}$. Hence, there are no cyclic chain dependencies, and the system obtained satisfies the assumptions of Lemma 7 and Lemma 8, with constants $Z_{i,q}$ and regular constants.

Therefore, there exists an unresolved system of the required form, with one of the components of its unique solution equal to $(L(M))_k$. This system uses constants $Z_{i,q}$ and regular constants. The former are expressed using Lemma 14 and the latter by Theorem 2. □

5.5. Computational completeness

The known method for proving computational completeness in language equations, explained in Section 2, proceeds by first defining the language of computation histories of a Turing machine, and then extracting the language recognized by the machine out of it. In Section 4, the numerical version of this argument was presented, in which the computation history is represented by a number, and the numbers accepted by machine are extracted out of the numbers representing computations. The set of numbers representing computations was obtained from the (linear conjunctive) language of their base-k representations using Theorem C, which produces equations with both union and intersection. In the present Section 5, the constructions of Theorem C have been re-implemented in the above Theorem 3 using equations with only one Boolean operation. Employing this theorem instead of Theorem C immediately establishes Lemma 2 in the following strengthened form, in which the equations may use *either* only union *or* only intersection.

Lemma 2*. *For every Turing machine T recognizing natural numbers there exists a system of equations*

$$\varphi_j(Y_0, \ldots, Y_5, X_1, \ldots, X_m) = \psi_j(Y_0, \ldots, Y_5, X_1, \ldots, X_m)$$

over sets of natural numbers using union and addition, such that its unique solution is $Y_0 = \mathrm{VALC}_0(T), \ldots, Y_5 = \mathrm{VALC}_5(T)$, $X_1 = S_1, \ldots, X_m = S_m$, for some sets $S_1, \ldots, S_m \subseteq \mathbb{N}$.

The same holds for systems using intersection and addition.

In either case, given T, the desired system can be effectively constructed.

Once the language of computation histories of a Turing machine is expressed, the final step of the argument is to extract the set of numbers accepted by the machine out of it. This was already done in Lemma 3 and in Lemma 4 using systems of equations with both union and intersection. In the rest of this section, the constructions in Lemmata 3 and 4 shall be refined to use either only union, or only intersection, which will yield the following improved versions of those results.

Lemma 3*. *For every Turing machine T recognizing a set of numbers $S_0 \subseteq \mathbb{N}$, there exists a system of equations of the form*

$$\varphi_j(Y, X_1, \ldots, X_m) = \psi_j(Y, X_1, \ldots, X_m)$$

with union and addition (or equally with intersection and addition) and constant $\{1\}$, which has the set of solutions

$$\big\{\big(S, f_1(S), \ldots, f_m(S)\big) \;\big|\; S_0 \subseteq S\big\},$$

where $f_1, \ldots, f_m \colon 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ are some monotone functions on sets of numbers, which depend on T. In particular, there is a least solution with $Y = S_0$.

Given T, either system can be effectively constructed.

Lemma 4*. *For every Turing machine T recognizing a set $S_0 \subseteq \mathbb{N}$, there exists a system of equations of the form*

$$\varphi_j(Z, X_1, \ldots, X_m) = \psi_j(Z, X_1, \ldots, X_m)$$

with union and addition (or with intersection and addition) and constant $\{1\}$, which has the set of solutions

$$\big\{\big(S, f_1(S), \ldots, f_m(S)\big) \;\big|\; S \subseteq \overline{S_0}\big\},$$

where $f_1, \ldots, f_m \colon 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ are monotone functions on sets of numbers, which depend on T. In particular, there is a greatest solution with $Z = \overline{S_0}$.

Given the Turing machine, one can effectively construct both systems.

The systems constructed in the proofs of Lemmata 3 and 4 consist mostly of inclusions of the form $\varphi \subseteq \psi$, which can be equally simulated by union ($\varphi \cup \psi = \psi$) and by intersection ($\varphi \cap \psi = \varphi$). The only exceptions are the equations (5) in Lemma 3 and (9) in Lemma 4, which use both intersection and union. Since those equations are identical, it is sufficient to rephrase a single equation (5). Its reformulation using addition and intersection is immediate.

Lemma 15. *Let $Y_i \subseteq (1\Sigma_6^+)_6$ for $1 \leqslant i \leqslant 5$, and let $Y_0 \subseteq \{0, 1, 2, 3, 4, 5\}$. Then, for every set $Y \subseteq \mathbb{N}$,*

$$Y = Y_0 \cup Y_1 \cup \bigcup_{\substack{i \in \{2,3,4,5\} \\ j \in \Sigma_6}} \Big(\big((Y_i \cap \big(1j\Sigma_6^*\big)_6) + \big((i-1)0^*\big)_6\big) \cap \big(ij\Sigma_6^*\big)_6\Big) \tag{26}$$

if and only if

$$Y_0 = Y \cap \{0, 1, 2, 3, 4, 5\}, \tag{27a}$$

$$Y_1 = Y \cap \big(1\Sigma_6^+\big)_6, \tag{27b}$$

$$Y \cap \big(ij\Sigma_6^*\big)_6 = \big((Y_i \cap \big(1j\Sigma_6^*\big)_6) + \big((i-1)0^*\big)_6\big) \cap \big(ij\Sigma_6^*\big)_6, \quad \text{for all } i, j \in \Sigma_6 \text{ with } i \neq 0, 1. \tag{27c}$$

The equation (26) is the same as (5) in the proof of Lemma 3, as well as (9) in Lemma 4.

Proof. The equation (26) represents Y as a disjoint union of sets of the following form: first, a subset of $\{0, \ldots, 5\}$; then, a subset of $(1\Sigma_6^+)_6$; and finally, for all $i, j \in \Sigma_6$ with $i \notin \{0, 1\}$, a subset of $(ij\Sigma_6^*)_6$. Note that the base supersets $\{0, \ldots, 5\}$, $(1\Sigma_6^+)_6$ and $(ij\Sigma_6^*)_6$ are disjoint, as witnessed by their leading digits. The new equations (27) state essentially the same using intersection with the base superset of each of these sets. The proof in each direction is by substitution.

⊜ Assume that (26) holds. Then, intersecting both sides of (26) with $\{0,\dots,5\}$, $(1\Sigma_6^+)_6$ and $(ij\Sigma_6^*)_6$ for all $i,j\in\Sigma_k$ with $i\notin\{0,1\}$ leads to the equations (27a), (27b) and (27c), respectively.

⊜ Conversely, assume that the sets Y_0, Y_1, …, Y_5 and Y satisfy (27). Substituting these equalities into the right-hand side of the equation (26) yields the following calculations:

$$\begin{aligned}&\big(Y\cap\{0,\dots,5\}\big)\cup\big(Y\cap\big(1\Sigma_6^+\big)_6\big)\cup\bigcup_{\substack{i\in\{2,3,4,5\}\\ j\in\Sigma_6}}\big(Y\cap\big(ij\Sigma_6^*\big)_6\big)\\ &=Y\cap\left(\{0,\dots,5\}\cup\big(1\Sigma_6^+\big)_6\cup\bigcup_{\substack{i\in\{2,3,4,5\}\\ j\in\Sigma_6}}\big(ij\Sigma_6^*\big)_6\right)=Y.\qquad\square\end{aligned}$$

An analogous result for addition and union requires introducing new variables, which makes its statement more complicated. The solutions of the simulating system are in a one-to-one correspondence with the solutions of the original system, and the new variables functionally depend on the original ones.

Lemma 16. *Consider the following system of equations in variables* $\{Y,Y_0,\dots,Y_5\}$.

$$Y=Y_0\cup Y_1\cup\bigcup_{\substack{i\in\{2,3,4,5\}\\ j\in\Sigma_6}}\Big(\big(\big(Y_i\cap\big(1j\Sigma_6^*\big)_6\big)+\big((i-1)0^*\big)_6\big)\cap\big(ij\Sigma_6^*\big)_6\Big)\tag{28a}$$

$$Y_0\subseteq\{0,1,2,3,4,5\}\tag{28b}$$

$$Y_i\subseteq\big(1\Sigma_6^+\big)_6,\quad\text{for } 1\leqslant i\leqslant 5\tag{28c}$$

There exists another system of equations in variables $\{Y,Y_0,\dots,Y_5\}\cup\{F_{i,j},G_{i,j},H_{i,j}\mid 2\leqslant i\leqslant 5,\ 0\leqslant j\leqslant 5\}$ *using the operations of union and addition, which has the set of solutions*

$$\big\{\big(Y=S,\ Y_i=S_i,\ F_{i,j}=f_{i,j}(S_i),\ G_{i,j}=g_{i,j}(S_i),\ H_{i,j}=h_{i,j}(S_i)\big)\ \big|\ (Y=S,\ Y_i=S_i)\ \textit{satisfies (28)}\big\},$$

where $f_{i,j},g_{i,j},h_{i,j}$ *are monotone functions, each mapping a set of numbers to a set of numbers.*

The monotonicity of the functions $f_{i,j}$, $g_{i,j}$ and $h_{i,j}$ is necessary to use the resulting system in the contexts where least or greatest solutions are being sought, such as in Lemmata 3* and 4*.

Proof. The system begins with the equations simulating the inner subexpressions in the equation (28a), where each variable Y_i is intersected with all sets $(1j\Sigma_6^*)_6$. The following equations are defined for each $i\in\{2,\dots,5\}$:

$$\bigcup_{j=0}^{5}F_{i,j}=Y_i,\tag{29a}$$

$$F_{i,j}\subseteq\big(1j\Sigma_6^*\big)_6,\quad\text{for all } j\in\Sigma_6.\tag{29b}$$

The first claim is that these equations bind each new variable $F_{i,j}$ to the variable Y_i.

Claim 16.1. *For every* $i\in\{2,\dots,5\}$, *the system (29) is equivalent to* $Y_i\subseteq(1\Sigma_6^+)_6$ *and* $F_{i,j}=Y_i\cap(1j\Sigma_6^*)_6$.

Proof. ⊜ The equations (29a) and (29b) directly imply that $Y_i=\bigcup_{j=0}^5F_{i,j}\subseteq\bigcup_{j=0}^5(1j\Sigma_6^*)_6=(1\Sigma_6^+)_6$, as well as that $F_{i,j}$ is contained in $Y_i\cap(1j\Sigma_6^*)_6$. To see that every number $n\in Y_i\cap(1j\Sigma_6^*)_6$ is in $F_{i,j}$, consider that it must belong to one of $F_{i,0},\dots,F_{i,5}$ by (29a), and this must be $F_{i,j}$, because all other $F_{i,j'}$ are disjoint with $(1j\Sigma_6^*)_6$ due to (29b).

⊜ Conversely, if $Y_i\subseteq(1\Sigma_6^+)_6$ and $F_{i,j}=Y_i\cap(1j\Sigma_6^*)_6$, then the equations (29) hold as follows:

$$\bigcup_{j=0}^{5}F_{i,j}=\bigcup_{j=0}^{5}Y_i\cap\big(1j\Sigma_6^*\big)_6=Y_i\cap\bigcup_{j=0}^{5}\big(1j\Sigma_6^*\big)_6=Y_i\cap\big(1\Sigma_6^+\big)_6=Y_i,$$

$$F_{i,j}=Y_i\cap\big(1j\Sigma_6^*\big)_6\subseteq\big(1j\Sigma_6^*\big)_6.\qquad\square$$

Next, the system (29) is extended with further equations that simulate larger subexpressions in the right-hand side of (28a). The subexpression $(F_{i,j}+((i-1)0^*)_6)\cap(ij\Sigma_6^*)_6$, defined by the variable $G_{i,j}$, replaces the leading digit 1 with i in all numbers from $F_{i,j}$. This is done by adding all numbers of the form $((i-1)0^*)_6$, and then intersecting with $(ij\Sigma_6^*)_6$ in

order to filter out all sums of an unintended form. Another variable $H_{i,j}$ defines the complementary set $(F_{i,j}+((i-1)0^*)_6)\cap(\Sigma_6^*\setminus ij\Sigma_6^*)_6$ of *all unintended sums* in that addition. The latter sets do not occur in the original equations, but they are needed for the proposed simulation to work.

For each $i\in\{2,\ldots,5\}$, consider the following system.

$$G_{i,j}\subseteq\left(ij\Sigma_6^*\right)_6, \qquad \text{for all } j\in\Sigma_6 \tag{30a}$$
$$H_{i,j}\subseteq\left(\Sigma_6^*\setminus ij\Sigma_6^*\right)_6, \qquad \text{for all } j\in\Sigma_6 \tag{30b}$$
$$G_{i,j}\cup H_{i,j}=F_{i,j}+\left((i-1)0^*\right)_6, \qquad \text{for all } j\in\Sigma_6 \tag{30c}$$

Together with the earlier equations (29), this system binds $G_{i,j}$ and $H_{i,j}$ to Y_j; this is the next claim in the proof.

Claim 16.2. *For each $i\in\{2,\ldots,5\}$, the system (29)–(30) is equivalent to*

$$\begin{aligned}
&Y_i\subseteq\left(1\Sigma_6^+\right)_6,\\
&F_{i,j}=Y_i\cap\left(1j\Sigma_6^*\right)_6,\\
&G_{i,j}=\left(F_{i,j}+\left((i-1)0^*\right)_6\right)\cap\left(ij\Sigma_6^*\right)_6,\\
&H_{i,j}=\left(F_{i,j}+\left((i-1)0^*\right)_6\right)\cap\left(\Sigma_6^*\setminus ij\Sigma_6^*\right)_6.
\end{aligned}$$

Proof. $\Rightarrow$ From Claim 16.1, it is already known that (29) implies $Y_i\subseteq(1\Sigma_6^+)_6$ and $F_{i,j}=Y_i\cap(1j\Sigma_6^*)_6$. Due to the inequalities (30a) and (30b), $G_{i,j}$ contains only numbers with the leading digits ij, while $H_{i,j}$ contains only numbers that *do not* have these two leading digits. Hence, the equation (30c) asserts that all numbers in the set $F_{i,j}+((i-1)0^*)_6$ beginning with ij are in $G_{i,j}$, while all numbers in that set that do not begin with ij are in $H_{i,j}$:

$$\begin{aligned}
&G_{i,j}=\left(F_{i,j}+\left((i-1)0^*\right)_6\right)\cap\left(ij\Sigma_6^*\right)_6,\\
&H_{i,j}=\left(F_{i,j}+\left((i-1)0^*\right)_6\right)\cap\left(\Sigma_6^*\setminus ij\Sigma_6^*\right)_6,
\end{aligned}$$

as claimed.

$\Leftarrow$ In the other direction, it follows from Claim 16.1 that if Y_i and $F_{i,j}$ satisfy the given system, then they also satisfy (29). It is left to verify (30a)–(30c) under the intended substitution given in the statement of the claim.

$$\begin{aligned}
&G_{i,j}=\left(F_{i,j}+\left((i-1)0^*\right)_6\right)\cap\left(ij\Sigma_6^*\right)_6\subseteq\left(ij\Sigma_6^*\right)_6\\
&H_{i,j}=\left(F_{i,j}+\left((i-1)0^*\right)_6\right)\cap\left(\Sigma_6^*\setminus ij\Sigma_6^*\right)_6\subseteq\left(\Sigma_6^*\setminus ij\Sigma_6^*\right)_6\\
&G_{i,j}\cup H_{i,j}=\left(\left(F_{i,j}+\left((i-1)0^*\right)_6\right)\cap\left(ij\Sigma_6^*\right)_6\right)\cup\left(\left(F_{i,j}+\left((i-1)0^*\right)_6\right)\cap\left(\Sigma_6^*\setminus ij\Sigma_6^*\right)_6\right)\\
&\qquad=\left(F_{i,j}+\left((i-1)0^*\right)_6\right)\cap\left(\left(ij\Sigma_6^*\right)_6\cup\left(\Sigma_6^*\setminus ij\Sigma_6^*\right)_6\right)\\
&\qquad=F_{i,j}+\left((i-1)0^*\right)_6 \qquad \square
\end{aligned}$$

Thus, all subexpressions of the equation (28a) have been expressed in the new variables $F_{i,j}$, $G_{i,j}$ and $H_{i,j}$. Using these variables, the equation (28a) is simulated by the following new equation.

$$Y=Y_0\cup Y_1\cup\bigcup_{\substack{i\in\{2,3,4,5\}\\ j\in\Sigma_6}}G_{i,j} \tag{31a}$$

Finally, the variables Y_0 and Y_1 are bounded by the following inequalities (reused from (28b) and (28c) in the statement from the lemma).

$$Y_0\subseteq\{0,1,2,3,4,5\} \tag{31b}$$
$$Y_1\subseteq\left(1\Sigma_6^+\right)_6 \tag{31c}$$

With the system completed, the statement of the lemma, which, roughly speaking, asserts the equivalence of the original system (28) and the constructed system (29)–(31) is proved as follows.

By Claim 16.2, every solution of the constructed system (29), (30), (31) is of the form $Y=S$, $Y_0=S_0$, …, $Y_5=S_5$, $F_{i,j}=S_i\cap(1j\Sigma_6^*)_6$, $G_{i,j}=(F_{i,j}+((i-1)0^*)_6)\cap(ij\Sigma_6^*)_6$, $H_{i,j}=(F_{i,j}+((i-1)0^*)_6)\cap(\Sigma_6^*\setminus ij\Sigma_6^*)_6$. It remains to show that a substitution of this form satisfies the constructed system if and only if $Y=S$, $Y_0=S_0$, …, $Y_5=S_5$ is a solution of the original system (28).

Substituting the expressions for $G_{i,j}$ into the new equation (31a) forms exactly the original equation (28).

$$S = S_0 \cup S_1 \cup \bigcup_{\substack{i\in\{2,3,4,5\}\\ j\in\Sigma_6}} G_{i,j} = S_0 \cup S_1 \cup \bigcup_{\substack{i\in\{2,3,4,5\}\\ j\in\Sigma_6}} \Big(\big((S_i \cap (1j\Sigma_6^*)_6) + ((i-1)0^*)_6\big) \cap (ij\Sigma_6^*)_6\Big)$$

Hence, the new equation (31a) is equivalent to the original equation (28a). Next, the constructed equation (31b) is exactly the same as the original equation (28b). The last original equation (28c) holds true for $i \geqslant 2$ by Claim 16.2; for $i = 1$, this equation is listed in the new system as (31c).

This shows that the set of solutions of the new system is as stated in the lemma, with the functions $f_{i,j}$, $g_{i,j}$ and $h_{i,j}$ defined by the expressions in Claim 16.2. These functions are monotone as compositions of monotone functions. □

Using equivalent reformulations (29)–(31) of equations (5) and (9), the constructions in the proofs of Lemmata 3 and 4 can be modified to use either union only or intersection only, thus proving those lemmata in their full form.

Proof of Lemma 3*. The desired system of equations is obtained by taking the system (3), (4), (5) from the proof of Lemma 3 and reimplementing the equation (5) according to Lemma 15 or Lemma 16. The correctness is proved separately in the case of union and in the case of intersection.

Union and addition. Consider the original system of equations defined in the proof of Lemma 3. This is a system in variables $\{Y, Y_0, \ldots, Y_5\}$, and it is comprised of the inequalities (3), (4) and the equation (5). The set of solutions of this system is defined by

$$\big\{\big(Y = S,\ Y_i = \psi_i(S)\big) \bigm| L(T) \subseteq S\big\},$$

where each function $\psi_i(Y)$ is monotone.

The new system of equations constructed in this updated argument uses the variables $\{Y, Y_0, \ldots, Y_5\} \cup \{F_{i,j}, G_{i,j}, H_{i,j} \mid 2 \leqslant i \leqslant 5,\ 0 \leqslant j \leqslant 5\}$ and consists of the same inequalities (3), (4) and the new equations (29)–(31) defined in the proof of Lemma 16. Though the latter equations use extra variables $F_{i,j}$, $G_{i,j}$ and $H_{i,j}$, by Lemma 16, their values are bound to Y_i by monotone functions $f_{i,j}$, $g_{i,j}$ and $h_{i,j}$. Lemma 16 states that these equations are equivalent to the desired equation (5) with the extra constraints (28b),(28c). Since these extra constraints are already listed in the original system as (4a) and (3a), respectively, they do not affect the equivalence of the two systems. Thus, the new system has the following set of solutions.

Claim 3*.1. *The set of solutions of the system* (3), (4), (29)–(31) *is*

$$\big\{\big(Y = S,\ Y_i = \psi_i(S),\ F_{i,j} = f_{i,j}\big(\psi_i(S)\big),\ G_{i,j} = g_{i,j}\big(\psi_i(S)\big),\ H_{i,j} = h_{i,j}\big(\psi_i(S)\big)\big) \bigm| L(T) \subseteq S\big\}.$$

All variables Y_i, $F_{i,j}$, $G_{i,j}$, $H_{i,j}$ depend on Y by a monotone function: the monotonicity of ψ_i was established in Lemma 3, the functions $f_{i,j}$, $g_{i,j}$ and $h_{i,j}$ are monotone by Lemma 16, and their compositions are monotone as well. The constant sets are expressed through singleton constants in the same way as in the proof of Lemma 3, only using Lemma 2* instead of Lemma 2 for $\mathrm{VALC}_i(T)$, and Theorem 2 instead of Theorem B for sets with regular base-6 representation. All resulting extra variables also monotonely depend on Y.

Intersection and addition. Construct a system of equations in variables $Y_0, Y_1, \ldots, Y_5$ comprised of the inequalities (3)–(4) from the proof of Lemma 3 and the equations (27) defined in Lemma 15. The equations (3a) and (4a) ensure that the conditions of Lemma 15 are met, and it states that the equations (27) have the same effect as (5). Therefore, the new system has the same set of solutions as the system (3), (4), (5). The constant sets are expressed as in the case of the union, using Lemma 2* and Theorem 2. □

The proof of Lemma 4* proceeds in exactly the same way as the above proof of Lemma 3*, by using the construction in Lemma 4 and by rephrasing the problematic equation (9) according to Lemma 16 and Lemma 15.

This was the last missing element of the proof of Theorem 1*. The entire argument can now be summarized as follows.

Proof of Theorem 1*. Recursively enumerable sets by least solutions. For every recursively enumerable set $S_0 \subseteq \mathbb{N}$, consider the Turing machine T recognizing this set, and the corresponding system of equations given by Lemma 3*. This system has the set of solutions

$$\big\{\big(Y = S,\ Y_1 = f_1(S),\ \ldots,\ Y_m = f_m(S)\big) \bigm| S_0 \subseteq S\big\},$$

where $f_0, \ldots, f_m \colon 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ are monotone functions on sets of numbers, which depend on T. In particular, its least solution is $Y = S_0$ and $Y_i = f_i(S_0)$ for all $i \in \{1, \ldots, m\}$.

Co-recursively enumerable sets by greatest solutions. Similarly, for a co-recursively enumerable set $\overline{S_0}$, take the Turing machine T recognizing its complement S_0. The system from Lemma 4* has the set of solutions

$$\left\{\left(Y = S,\ Y_1 = f_1(S),\ \ldots,\ Y_m = f_m(S)\right) \mid S \subseteq \overline{S_0}\right\},$$

where $f_0, \ldots, f_m \colon 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ are monotone functions on sets of numbers, which depend on T. Then its greatest solution is $Y = \overline{S_0}$, $Y_i = f_i(\overline{S_0})$ for all $i \in \{1, \ldots, m\}$.

Recursive sets by unique solutions. Every recursive set is defined by combining these two constructions in the same way as described in the proof of Lemma 5, which establishes the last assertion of Theorem 1*. □

In the notation of language equations over a unary alphabet, Theorem 1* can be equivalently restated as the following stronger statement of Theorem A, which applies to alphabets of any size.

Corollary 1. *For every finite non-empty alphabet Σ, with $|\Sigma| \geqslant 1$, a language $L_0 \subseteq \Sigma^*$ is representable by a unique (least, greatest) solution of a system of language equations $\varphi_j(X_1, \ldots, X_n) = \psi_j(X_1, \ldots, X_n)$, using singleton constants over Σ and the operations of concatenation and either only union or only intersection, if and only if L_0 is recursive (r.e., co-r.e., respectively).*

6. Decision problems

Consider basic properties of equations, such as existence and uniqueness of solutions. In the more general case of language equations, these and a few other properties are undecidable [23,25,26], and their exact position in the arithmetical hierarchy was determined in the literature.

A set is in the arithmetical hierarchy if, for some $k \geqslant 0$, it can be obtained from a $(k+1)$-ary recursive relation using k quantifiers. The number of quantifiers necessary to represent a set determines its level in the hierarchy. Each level is comprised of two classes, Σ_k^0 or Π_k^0, which correspond to the cases of the first quantifier's being existential or universal. The class of recursive sets forms the bottom of the arithmetical hierarchy, denoted by Σ_0^0 or, equivalently, by Π_0^0. For every $k \geqslant 1$, a set is in Σ_k^0 if it can be represented as

$$\left\{w \mid \exists x_1 \forall x_2 \ldots Q_k x_k\ R(w, x_1, \ldots, x_k)\right\}$$

for some recursive relation R, where $Q_k = \forall$ if k is even and $Q_k = \exists$ if k is odd. Similarly, a set is in Π_k^0 if it admits a representation

$$\left\{w \mid \forall x_1 \exists x_2 \ldots Q_k x_k\ R(w, x_1, \ldots, x_k)\right\}.$$

In particular, Σ_1^0 and Π_1^0 are the classes of recursively enumerable sets and of their complements, respectively. At every k-th level, $\Pi_k^0 = \{L \mid \overline{L} \in \Sigma_k^0\}$. The arithmetical hierarchy is known to be strict: $\Sigma_k^0 \subset \Sigma_{k+1}^0$ and $\Pi_k^0 \subset \Pi_{k+1}^0$ for every $k \geqslant 0$. Furthermore, for every $k \geqslant 1$, the inclusion $\Sigma_k^0 \cup \Pi_k^0 \subset \Sigma_{k+1}^0 \cap \Pi_{k+1}^0$ is proper.

Each class Σ_k^0 and Π_k^0 has complete sets with respect to one-to-one reduction. The halting problem for Turing machines is the standard example of a Σ_1^0-complete problem, while Turing machine universality and Turing machine co-finiteness are Π_2^0-complete and Σ_3^0-complete, respectively. These complete problems were earlier use to determine the hardness of basic decision problems for language equations over multiple-symbol alphabets.

Theorem D. *(See Okhotin [23,25,26].) Given a system of language equations $\varphi_i(X_1, \ldots, X_n) = \psi_i(X_1, \ldots, X_n)$ over an alphabet Σ with $|\Sigma| \geqslant 2$, using the operations of union, intersection and concatenation, as well as singleton constants,*

i. *testing whether it has a solution is Π_1^0-complete [23,25];*
ii. *testing whether it has a unique solution is Π_2^0-complete [23,25];*
iii. *testing whether it has finitely many solutions is Σ_3^0-complete [26].*

These results will now be re-created for equations over sets of numbers, based upon the constructions from the previous section.

Theorem 4. *For systems of equations $\varphi_i(X_1, \ldots, X_n) = \psi_i(X_1, \ldots, X_n)$ over sets of natural numbers using the operations of union, intersection and addition, as well as singleton constants,*

i. *testing whether a given system has a solution is Π_1^0-complete;*
ii. *testing whether a system has a unique solution is Π_2^0-complete;*
iii. *testing whether a system has finitely many solutions is Σ_3^0-complete.*

All problems maintain their hardness if the allowed operations are union and addition, or intersection and addition.

Proof. According to Theorem D, in the more general case of language equations, these three problems are in Π_1^0, in Π_2^0 and in Σ_3^0, respectively.

The Π_1^0-hardness of the solution existence problem is proved by a reduction from the emptiness problem for Turing machines. Let T be a given Turing machine. Consider the system of equations given for T by Lemma 3*. This is a system in variables $(Y, X_1, \ldots, X_m)$, each of its solutions satisfies $L(T) \subseteq Y$, and there is a solution with $Y = L(T)$. Then it is sufficient to add a new equation $Y = \varnothing$, so that the resulting system has a solution if and only if $L(T) = \varnothing$.

The Π_2^0-hardness of the solution uniqueness problem is proved by a reduction from the known Π_2^0-complete Turing machine universality problem, which can be stated as follows: "Given a Turing machine M working on natural numbers, determine whether it accepts every number $n \in \mathbb{N}$". Given T, construct the system of equations as in Lemma 3*. It has a unique solution if and only if the bounds $L(T) \subseteq S \subseteq \mathbb{N}$ are tight, that is, if and only if the Turing machine accepts every number. This completes the reduction.

Finally, consider the problem of testing whether a system has finitely many solutions. To prove its Σ_3^0-hardness, consider the co-finiteness problem for Turing machines, which is stated as "Given a Turing machine T working on natural numbers, determine whether $\mathbb{N} \setminus L(T)$ is finite", which is known to be Σ_3-complete [30, Cor. 14-XVI]. Given T, use Lemma 3* to construct the system of equations with the set of solutions $\{(S, f_1(S), \ldots, f_k(S)) \mid L(T) \subseteq S\}$. This set is finite if and only if $\mathbb{N} \setminus L(T)$ is finite, which completes the reduction. □

7. Conclusion

The equations considered in this paper are a pure mathematical object, and seemingly a rather simple one: constructing any system with a non-periodic solution is a challenging task in itself. Unexpectedly, this object turned out to be equivalent to the notion of effective computability.

The nature of this result bears a certain resemblance to the famous case of Diophantine equations, which were proved to be computationally complete by Matiyasevich, after a series of results on expressing individual sets and relations in these equations [20]. Due to this result, it is known, for instance, that there exists a Diophantine equation, for which the range of admissible values of a certain variable x is exactly the set of primes. Similarly, our Lemma 3 allows constructing a system of equations over sets of natural numbers, which has a unique solution with one of its components being exactly the set of primes.

Among the applications of this result, it settles the expressive power of a natural extension of integer circuits [21], as well as shows that language equations are computationally complete even in the seemingly trivial case of a unary alphabet.

Acknowledgments

The authors are grateful to the anonymous reviewers for pointing out numerous shortcomings of the presentation in the original manuscript.

Research supported by the National Science Centre (NCN) under grant number 2011/01/D/ST6/07164, 2011–2015, and by the Academy of Finland under grants 118540, 134860 and 257857.

This research was commenced during the first author's visit to Turku funded by a grant from the European Science Foundation, as part of the project "Automata: from Mathematics to Applications", Ref. num. 1763.

References

[1] A. Aiken, D. Kozen, M. Vardi, E. Wimmers, The complexity of set constraints, in: Computer Science Logic, CSL 1993, Swansea, United Kingdom, September 13–17, 1993, in: Lect. Notes Comput. Sci., vol. 832, 1994, pp. 1–17.
[2] J.-P. Allouche, J. Shallit, Automatic Sequences: Theory, Applications, Generalizations, Cambridge University Press, 2003.
[3] F. Baader, P. Narendran, Unification of concept terms in description logic, J. Symb. Comput. 31 (3) (2001) 277–305.
[4] W. Charatonik, Set constraints in some equational theories, Inf. Comput. 142 (1) (1998) 40–75.
[5] W. Charatonik, L. Pacholski, Set constraints with projections, J. ACM 57 (4) (2010), article 23.
[6] J.H. Conway, Regular Algebra and Finite Machines, Chapman and Hall, 1971.
[7] K. Culik II, J. Gruska, A. Salomaa, Systolic trellis automata I, Int. J. Comput. Math. 15 (1984) 195–212, Systolic trellis automata II, Int. J. Comput. Math. 16 (1984) 3–22.
[8] S. Ginsburg, H.G. Rice, Two families of languages related to ALGOL, J. ACM 9 (1962) 350–371.
[9] J. Hartmanis, Context-Free Languages and Turing Machine Computations, Proc. Symp. Appl. Math., vol. 19, AMS, 1967, pp. 42–51.
[10] A. Jeż, Conjunctive grammars can generate non-regular unary languages, Int. J. Found. Comput. Sci. 19 (3) (2008) 597–615.
[11] A. Jeż, A. Okhotin, Conjunctive grammars over a unary alphabet: undecidability and unbounded growth, Theory Comput. Syst. 46 (1) (2010) 27–58.
[12] A. Jeż, A. Okhotin, Complexity of equations over sets of natural numbers, Theory Comput. Syst. 48 (2) (2011) 319–342.
[13] A. Jeż, A. Okhotin, One-nonterminal conjunctive grammars over a unary alphabet, Theory Comput. Syst. 49 (2) (2011) 319–342.
[14] A. Jeż, A. Okhotin, Representing hyper-arithmetical sets by equations over sets of integers, Theory Comput. Syst. 51 (2) (2012) 196–228.
[15] M. Kunc, The power of commuting with finite sets of words, Theory Comput. Syst. 40 (4) (2007) 521–551.
[16] M. Kunc, What do we know about language equations?, in: Developments in Language Theory, DLT 2007, Turku, Finland, July 3–6, 2007, in: Lect. Notes Comput. Sci., vol. 4588, 2007, pp. 23–27.
[17] E.L. Leiss, Unrestricted complementation in language equations over a one-letter alphabet, Theor. Comput. Sci. 132 (1994) 71–93.
[18] E.L. Leiss, Language Equations, Springer-Verlag, 1999.
[19] W. Martens, M. Niewerth, T. Schwentick, Schema design for XML repositories: complexity and tractability, in: Symposium on Principles of Database Systems, PODS 2010, Indianapolis, USA, June 6–11, 2010, pp. 239–250.

[20] Yu.V. Matiyasevich, Hilbert's Tenth Problem, MIT Press, 1993.
[21] P. McKenzie, K.W. Wagner, The complexity of membership problems for circuits over sets of natural numbers, Comput. Complex. 16 (2007) 211–244.
[22] A. Okhotin, Conjunctive grammars, J. Autom. Lang. Comb. 6 (4) (2001) 519–535.
[23] A. Okhotin, Decision problems for language equations, J. Comput. Syst. Sci. 76 (3–4) (2010) 251–266.
[24] A. Okhotin, On the equivalence of linear conjunctive grammars to trellis automata, Inform. Théor. Appl. 38 (1) (2004) 69–88.
[25] A. Okhotin, Unresolved systems of language equations: expressive power and decision problems, Theor. Comput. Sci. 349 (3) (2005) 283–308.
[26] A. Okhotin, Strict language inequalities and their decision problems, in: Mathematical Foundations of Computer Science, MFCS 2005, Gdańsk, Poland, August 29–September 2, 2005, in: Lect. Notes Comput. Sci., vol. 3618, 2005, pp. 708–719.
[27] A. Okhotin, Homomorphisms preserving linear conjunctive languages, J. Autom. Lang. Comb. 13 (3–4) (2008) 299–305.
[28] A. Okhotin, Language equations with symmetric difference, Fundam. Inform. 116 (1–4) (2012) 205–222.
[29] R. Parikh, A. Chandra, J. Halpern, A. Meyer, Equations between regular terms and an application to process logic, SIAM J. Comput. 14 (4) (1985) 935–942.
[30] H. Rogers Jr., Theory of Recursive Functions and Effective Computability, 1967.
[31] L.J. Stockmeyer, A.R. Meyer, Word problems requiring exponential time: Preliminary report, in: Proceedings of the 5th Annual ACM Symposium on Theory of Computing, STOC 1973, April 30–May 2, 1973, Austin, Texas, USA, pp. 1–9.