Artur Jeż

Gramatyki koniunkcyjne i układy równań nad zbiorami liczb naturalnych

Rozprawa doktorska napisana pod kierunkiem
prof. Krzysztofa Lorysia, Uniwersytet Wrocławski
oraz
dr. hab. Alexandra Okhotina, University of Turku

Instytut Informatyki
Uniwersytet Wrocławski
2010

Artur Jeż

Conjunctive Grammars and Equations over Sets of Natural Numbers

Ph.D. Thesis
Supervisors:
prof. Krzysztof Loryś, University of Wrocław
and
dr. docent Alexander Okhotin, University of Turku

Institute of Computer Science
University of Wrocław
2010

Contents

Streszczenie

W niniejszej rozprawie rozważamy układy równań postaci $\psi(\vec{X}) = \varphi(\vec{X})$ nad zbiorami liczb naturalnych. Równania te mogą używać operacji sumy mnogościowej, przecięcia oraz dodawania. Są one równoważne układom równań języków formalnych nad alfabetem jednoliterowym z operacjami sumy mnogościowej, przecięcia oraz konkatenacji.

Ważnym przypadkiem szczególnym są układy równań w postaci rozwiązanej, tj. $\vec{X} = \varphi(\vec{X})$. Odpowiadające im układy równań języków formalnych w postaci rozwiązanej są z kolei równoważne gramatykom koniunkcyjnym. Pokażemy, że układy równań w postaci rozwiązanej mogą mieć najmniejsze rozwiązania, które nie są ciągami arytmetycznymi. Równoważnie, gramatyki koniunkcyjne generują nieregularne języki nad alfabetem jednoliterowym, w przeciwieństwie do gramatyk bezkontekstowych. Dowód tego twierdzenia jest konstruktywny: dla zadanego zbioru S konstruujemy układ równań taki, że S jest jego najmniejszym rozwiązaniem, o ile tylko notacje pozycyjne liczb z S są rozpoznawane przez pewien specjalny automat komórkowy.

W przypadku ogólnym dowodzimy, że klasa rozwiązań jedynych (najmniejszych, największych) równań nad zbiorami liczb naturalnych pokrywa się z klasą zbiorów rekurencyjnych (odpowiednio: rekurencyjnie przeliczalnych, ko-rekurencyjnie przeliczalnych). Wynik ten zachodzi nawet dla równań używających tylko operacji sumy mnogościowej i dodawania lub tylko przecięcia i dodawania. Twierdzenie to jest uogólnieniem znanego wyniku dla układów równań języków formalnych nad alfabetami wieloliterowymi. Następnie zajmujemy się układami, w których jedyną dozwoloną operacją jest dodawanie. W ich przypadku odpowiednie klasy rozwiązań są obliczeniowo zupełne, tj. klasa rozwiązań jedynych (najmniejszych, największych) zawiera kodowanie wszystkich zbiorów rekurencyjnych (odpowiednio: rekurencyjnie przeliczalnych, ko-rekurencyjnie przeliczalnych).

Ponadto zajmujemy się złożonością obliczeniową problemów decyzyjnych dla rozważanych układów równań. Pokazaliśmy, że problem przynależności do najmniejszego rozwiązania układu równań w postaci rozwiązanej jest EXPTIME-zupełny. Inne problemy decyzyjne dla obu formalizmów są w większości nierozstrzygalne, nawet w przypadku ograniczonym do jednej zmiennej i jednego równania.

Abstract

Systems of equations $\psi(\vec{X}) = \varphi(\vec{X})$ over sets of natural numbers with union, intersection and addition allowed, are studied in this thesis. Such systems can be equally viewed as systems of language equations over a single-letter alphabet and operations of union, intersection and concatenation.

The first to be considered is the subclass of systems of equations over sets of numbers of the resolved form $\vec{X} = \varphi(\vec{X})$. Their counterparts among the language equations are the resolved systems of language equations over a single-letter alphabet, which can be also seen as a conjunctive grammar over a single-letter alphabet. It is shown that the resolved systems of equations over sets of natural numbers can have non-ultimately periodic sets as the least solutions. Equivalently, conjunctive grammars over a single-letter alphabet can generate non-regular languages, as opposed to context-free grammars. To this end, an explicit construction of a resolved system with a given set of numbers as the least solution is presented, provided that base-k positional notations of numbers from this set are recognised by a certain type of a real-time cellular automaton.

In the general case of systems of equations, it is shown that the class of unique (least, greatest) solutions of such systems coincides with the class of recursive (recursively enumerable, co-recursively enumerable, respectively) sets. This result holds even when only union and addition (or only intersection and addition) are allowed in the system. This generalises the known result for systems of language equations over a multiple-letter alphabet. Systems with addition as the only allowed operation are also considered, and it is shown that the obtained class of sets is computationally universal, in the sense that their unique (least, greatest) solutions can represent encodings of all recursive (recursively enumerable, co-recursively enumerable, respectively) sets.

The computational complexity of decision problems for both formalisms is investigated. It is shown that the membership problem for the resolved systems of equations is EXPTIME-hard. Many other decision problems for both types of systems are proved to be undecidable, and their exact undecidability level is settled. Most of these results hold even when the systems are restricted to the use of one equation with one variable.

Acknowledgments

I would like to thank my co-advisors: Krzysiek Loryś, for his idea to investigate conjunctive grammars, encouragement when it seemed to be pointless and all the discussion about the problem; Alexander Okhotin, for all the time we spent writing, thinking and reading and for good scientific ideas in general.

My parents and brother, for their patience and support.

Ania, for her understanding in the crucial moment.

Friends, for their help in making leisure-time not work-time.

Polish Ministry of Science and Higher Education, for their kind and generous support under grant MNiSW N N206 259035 2008–2010.

Chapter 1

Introduction

Systems of equations over sets of natural numbers are among the simplest and most intuitive mathematical formalisms. Each equation is of the form

$$\varphi(X_1, \dots, X_n) = \psi(X_1, \dots, X_n) \ ,$$

where expressions φ and ψ may use Boolean operations and arithmetical operations, applied pairwise to elements of sets, for example:

$$A + B = \{a + b \mid a \in A,\ b \in B\} \ .$$

In this thesis, equations with only union and intersection as the allowed Boolean operations and with addition as the only arithmetical operation are studied.

Consider the following example of an equation with an unknown $X \subseteq \mathbb{N}$, with the operations of addition and union, and with two singleton constant languages.

$$X = (X + \{2\}) \cup \{0\} \ .$$

This equation defines even numbers inductively, by stating that 0 is an even number, and that an even number plus two is an even number as well. Formally, the equation has a unique solution, which is the set of all even numbers.

Another example of a univariate equation

$$X + \{1\} = (X + X) \cup \{2\}$$

no longer defines an explicit induction, as none of its sides is X (in the following, such equations shall be called *unresolved*, vs. *resolved* for equations of the previous type). It has multiple solutions, among them the solutions $X = \{1\}$ and $X = \{1, 2, \dots\}$. Furthermore, every solution of this equation must contain 1 and be contained in $\{1, 2, \dots\}$, which makes these two solutions the *least* and the *greatest solutions* of this equation.

As already shown by the above examples, an equation over sets of numbers may have a unique solution or multiple solutions, or sometimes no solutions at all, as in the example $X = X + \{1\}$. There is a common outlook on an equation as a *formalism for defining its solutions.* An equation with a unique solution may be seen as a definition of this solution, and in case of multiple solutions, least or greatest solutions (provided that they exist, as in the second example above) may be similarly considered.

All the examples considered so far define ultimately periodic sets, and in fact, every ultimately periodic set can be easily defined by a unique solution of an equation with finite constants. However, when one tries to write any such system of equations, its solution somehow tends to be ultimately periodic, and there seems not to be an easy way to avoid this. Prior to this thesis, only one example of a non-periodic set defined by an equation over sets of numbers has been known, and no general method of constructing any further examples.

In this thesis a surprising result is shown: the class of unique solutions of equations of the general form is exactly the class of recursive sets, that is, the sets with an effectively decidable membership. For equations in the resolved form $X = \varphi(X)$, the computational power of their solutions is limited, yet still much exceeds the class of ultimately periodic sets. Such results belong to the domain of the theory of computations, and the methods used to establish them come from formal language theory.

1.1 Formal languages

The idea of a formal language as a set accepted by some sort of an abstract machine goes back to Turing. While in general Turing Machines recognise exactly the recursive sets, one can limit their computational power in many ways and thus obtain simpler abstract automata corresponding to different models of computation with restricted resources.

The desire to process natural languages led Chomsky [9] to define formal grammars. This is where the notations of grammars and derivations were coined. The idea that abstract constructions of formal languages are means to generate languages was stressed. In this approach, basic grammatical terms, such as a noun or a verb, as well as more complex ones, such as a noun phrase or a verb phrase, are denoted by some intermediate symbols used in the rewriting, usually called *non-terminal symbols.* In this way the syntax defined by a grammar is strongly connected with the semantics of the studied object.

A parallel approach is motivated by programming languages and parser design, where the language constructs have to be precisely defined so that they can be used to swift and unambiguous text parsing. For example, the ALGOL 60 committee devised, independently from Chomsky, the notion of a context-free grammar and used it in the specification of ALGOL programming language [2, 3]. Since this type of study is very practically oriented, it introduced the idea of efficiency, properties related to parsing and subtle trade-offs between expressive power and complexity of decision problems, mainly the membership problem.

These independent sources defined the most popular formalisms in formal language theory: pushdown automata and context-free grammars, which capture the two intuitive properties of formal languages: the acceptance by some machine and the derivation, which is a top-down way of producing a parse tree. Nevertheless, these are not the only formalisms known. The language equations are an alternative approach. They achieve a great simplicity and mathematical soundness of the semantics as well as robustness at the slight expense of the intuition behind the formalism.

1.2 Language equations

Language equations are equations with formal languages as unknowns, their most general form can be given as

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

where each φ_i, ψ_i consist of variables, constants from some given class, and operations on languages, such as concatenation, union, intersection, complementation, and possibly others. While language equations are an old formalism, studied theoretically already in the early research of Conway [10], in the recent years they have attracted new attention; this resulted in some unexpected results, including solving old open questions. This renewed interest was summarised by Kunc in his recent survey [30].

Language equations have several useful properties. Their semantics is intuitively clear and can be mathematically well formalised. One can trim or extend them in many ways to obtain different classes of languages or to adapt them to a particular application. Such changes do not affect the way the semantics is given. Moreover, grammars and automata can be translated to systems of language equations. While this does not introduce any theoretical or practical novelty on its own, it often eases the proofs and formalisation and provides an elegant common generalisation.

Despite all the motivation, language equations are far away from being fully, or even partially, understood. The quest for their better understanding is crucial. In particular, even a partial progress in a limited simple subcase is important, as the gained experience might eventually help in dealing with more general cases.

1.2.1 General language equations

When no restriction on the form of language equations is imposed, they possess great computational power. It was observed quite early, that the solution existence problem is undecidable, as shown by Charatonik [8]. Later it was determined by Okhotin [47, 41, 42] that the family of sets representable by unique (least, greatest) solutions of such equations is exactly the family of recursive languages (recursively enumerable, co-recursively enumerable, respectively). To understand the formulation of this theorem, a partial order on solutions of a fixed system of equations is needed. In the general case, a solution is a vector of languages $(L_1, \ldots, L_n)$, and it is greater than another solution $(K_1, \ldots, K_n)$, if $K_i \subseteq L_i$ for each i-th coordinates. The least (greatest) solution is the one that is less (greater, respectively) than any other solution. Though such extremal solutions do not always exist, when they do, then have a computationally complete expressive power:

Theorem 1.1 (Okhotin [47, 41]). *Consider a system of language equations using union, intersection, concatenation and recursive constants that has a unique (least, greatest) solution $(L_1, \ldots, L_n)$. Then each component L_i is recursive (recursively enumerable, co-recursively enumerable, respectively).*

Conversely, for every recursive (recursively enumerable, co-recursively enumerable) language $L \subseteq \Sigma^*$ (with $|\Sigma| \geq 2$) there exists a system using concatenation, union and singleton constants (concatenation, intersection and singleton constants) with the unique (least, greatest, respectively) solution $(L, \ldots)$.

This shows that language equations are computational devices equivalent to Turing Machines and other established notions.

All hardness results shown in Theorem 1.1 use an explicit construction of a system of language equations, with the language of computation histories of a given Turing Machine as a unique solution. Then one can extract the starting tape contents of a Turing Machine out of its computation history. It should be stressed that both the representation of the computation history and the extraction of the tape contents essentially use a multiple-letter alphabet.

Among the interesting results obtained for language equations, the one of Kunc [29] should be mentioned, as it settled a question that remained open for decades. In his book, Conway [10] asked, whether all greatest solutions of equations of the form $XL = LX$ for regular L are context free. For many years even a non-regular solution could not be found, which actually raised a conjecture, that all such equations have regular greatest solutions. Contrary to that, Kunc [29] constructed an equation $XL = LX$ with a finite L, which has a co-recursively enumerable hard greatest solution.

1.2.2 Resolved language equations

Among all language equations a particularly useful subclass consists of *resolved language equations*, i.e., the one of the form:

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

Such systems possess several nice properties. First of all, if the expressions φ_i are monotone, the least solution exists, which is assured by the Tarski's Fixpoint Theorem [52]. Moreover, only slight restrictions on the form of φ_i's are needed to guarantee that the least solution is in fact the unique solution not containing ε (i.e., the empty word) in any of its coordinate.

Their second key property was discovered by Ginsburg and Rice [13], who gave a semantics of context-free grammars by using resolved language equations with concatenation and union as the only operations. In fact this semantics is a simple (syntactic) transformation between context-free grammars and resolved language equations using union and concatenation. Such a transformation supports the claim that there is a strong connection between resolved language equations and grammars.

This connection was further explored by A. Okhotin, who defined *conjunctive grammars*. They can be viewed as an extension of context-free grammars by a possibility of an unrestricted intersection in the body of every rule. Informally speaking, a string is generated by a conjunction of rules if and only if it is produced by every conjunct.

The particular importance of conjunctive grammars among other extensions of context-free grammars lies not in the possibility of the intersection itself, but in the semantics. It can be given not only in terms of string rewriting, but also in terms of language equations. In fact this is the (unique) semantics generalising the one of Ginsburg and Rice [13] to the language equations using union, intersection and concatenation. And this is the major justification both for conjunctive grammars and for resolved systems of language equations.

It should be noted that the conjunctive grammars inherit almost all good parsing properties of context-free grammars, while having much larger expressive power, stretching beyond the finite intersection of context-free languages.

1.3 Systems of equations over sets of numbers

Consider language equations over a single letter alphabet. By a trivial isomorphism $a^n \longleftrightarrow n$, unary languages can be regarded as sets of natural

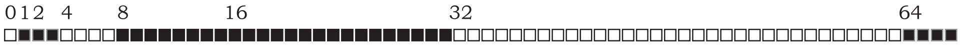

Figure 1.1: The Leiss construction depicted graphically. The black boxes represent the numbers from the unique solution, while the white ones represent the numbers outside the solution.

numbers, and language equations over a single-letter alphabet become *equations over sets of numbers*. Concatenation of languages accordingly turns into *addition* (or *sum*) of sets

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

Such equations constitute a basic mathematical object on its own, the study of which can be traced up to the seminal paper of Stockmeyer and Meyer [51], who studied *integer expressions*, i.e., regular expressions (though with complementation) over unary languages. Their work was continued by Yang [56], who studied *integer circuits*, that is expressions which may share the subexpressions; recursive calls are still not allowed though. A very systematic and thorough study of complexity of such problems was conducted by McKenzie and Wagner [34]. As such expressions (and circuits) define only finite sets, they were studied mainly with complexity of their decision problems in mind, while their expressive power was not investigated, as there was virtually nothing to investigate.

Equations over sets of numbers are a more general formalism, and its expressive power seems to be related to the allowed Boolean operations. For example, it is relatively easy to see that for resolved equations using union and addition, only ultimately periodic sets can be obtained (or, in terms of formal languages, that unary context-free languages are regular [13]).

Constructing any system of equations, even an unresolved one, with a non-regular unique solution has proved to be a non-trivial task; the first example of such an equation using the operations of concatenation and complementation was presented by Leiss [31], in fact this was a single resolved equation.

Example 1.1 (Leiss [31])**.** For every expression φ, denote $2\varphi = \varphi + \varphi$. Then the unique solution of the equation

$$X = 2\left(\overline{2\left(\overline{2\overline{X}}\right)}\right) + \{1\}$$

is $\{n \mid \exists i \geq 0 : \ 2^{3i} \leq n < 2^{3i+2}\} =$

$$= \{n \mid \text{base-8 notation of } n \text{ starts with } \mathtt{1},\ \mathtt{2} \text{ or } \mathtt{3}\} \ .$$

Knowing this result, and seeing the expressive power of conjunctive grammars, Wotschke [55] asked the question, whether unary conjunctive grammars generate non-regular languages, or equivalently, whether there exists a non-periodic least solution of resolved system of equations over sets of numbers using union, intersection and addition.

This question remained open for some time, in fact it was considered by A. Okhotin [44] as one of the most important problem in the field. In particular, a negative answer was conjectured.

Contrary to that, I found an example of such a system, with the least solution $\{4^n \mid n \in \mathbb{N}\}$. This thesis is a result of a study started by this single simple result.

It should be noted, that all the previous result by Stockmeyer and Meyer, Yang, McKenzie and Wagner can be rewritten as resolved systems of equations over sets of numbers and apparently no work was carried for general systems. This may look strange, as in case of language equations over general alphabet their true expressive power was found only for unresolved equations, see Theorem 1.1. It seems natural and very tempting to ask, what is the complexity of general systems of equations over sets of numbers? A recursive upper bound for unique solution can be easily deduced from Theorem 1.1, but it is hard to believe that this level of complexity can be achieved. It is shown that such a lower bound in fact holds. This raises even more questions—for example, can any non-trivial set be obtained with equations with addition only? The result of Kunc [29] demonstrates that this is true for language equation with concatenation only, but this construction is very complicated and seems impossible to replicate. Still, some non trivial partial results are provided in the later chapters.

1.4 Outline of the results

This thesis deals with various aspects of equations over sets of numbers. Both resolved and general systems of equations are considered, restricted to usage of monotone Boolean operations, i.e., union and intersection. The former systems may be equally interpreted as unary conjunctive grammars. Results concerning expressive power, computational complexity as well as descriptional complexity of such systems are provided.

Firstly, the expressive power of resolved systems of equations is investigated, as the first results in whole area emerged there. While matching bounds are not provided, still the lower bound obtained is quite unexpected and its application is crucial for the later results to hold. Being more precise, for a set S with language L_S of base-k positional notations of numbers from S, an effective construction of resolved system of equations with least solution S is supplied, assuming L belongs to a certain class of formal languages. This class has an automaton characterisation: it is recognised by a certain subclass of real-time cellular automata. It is relatively rich, as it is closed under all Boolean operations and properly contains linear context-free grammars.

The first example of a non-regular least solution and representation of sets with regular positional notation is the author's independent work [19], while the general representation of sets with positional notation recognised by real time cellular automata is a joint work with the advisor, Alexander Okhotin [24]

Secondly, the interest is switched to the expressive power of general systems. Matching lower and upper bound are provided, they are the same as the bounds for general language equations from Theorem 1.1, i.e., the class of unique (least, greatest) solutions of such systems of equations over sets of natural numbers coincides with the class of recursive (recursively enumerable, co-recursively enumerable, respectively) sets. This part of the thesis is based on author's joint work with the advisor, Alexander Okhotin [21, 22].

Then the descriptional complexity of such systems is investigated. That is, the attempts are made to understand how limiting the use of operations and variables affects the complexity of the solutions. Contrary to the intuition, these systems are as hard as the general ones already when limited to one variable and one equation. The general equations are also equally hard when allowed to use addition only. This part of the thesis is based on author's joint work with the advisor, Alexander Okhotin [23, 25].

Finally the computational complexity of the decision problems for these formalisms is studied. As soon as their expressive power is understood well, the complexity of most common decision problems for both resolved and unresolved equations is easily settled. Among surprising result, the membership problem for the resolved systems is shown to be EXPTIME-complete, while most of other decision problems for both resolved and unresolved systems, such as emptiness, equivalence, finiteness etc., are shown to be undecidable. This part of the thesis is based on author's joint work with the advisor, Alexander Okhotin [20, 21, 22, 24].

Statement of originality

The results presented in this thesis are original and are either author's own work or joint work with the advisor, Alexander Okhotin. All of them were either accepted at refereed conferences; or printed or accepted for printing in refereed journals.

Results by other authors are properly cited.

Chapter 2

Basic terminology and notation

In this chapter, the basic terminology and the general facts about conjunctive grammars, language equations and equations over sets of natural numbers are stated.

2.1 General notation conventions

2.1.1 Formal languages

The standard notation for formal languages is used, i.e., Σ denotes a finite *alphabet*, its elements are called *letters*. A *word* over Σ is a finite sequence of letters from Σ, the empty word is denoted by ε. Any set of words over Σ is called a *language*. The usual set-theoretic operations, such as union, intersection and complementation, are applied to languages with the standard meaning. *Concatenation* is denoted by $\cdot$ and defined as $L \cdot L' = \{ww' \mid w \in L, w' \in L'\}$. For a fixed language L, its concatenation with itself is shortened to: $L^0 = \{\varepsilon\}$ and $L^k = L^{k-1} \cdot L$. The *Kleene star* denotes the set of words obtained by a finite concatenation of words from a given language L: $L^* = \bigcup_{i \geq 0} L^i$. An operation inverse to concatenation of a word is the *quotient* with a word

$$Lw^{-1} = \{w' \mid w'w \in L\}, \quad w^{-1}L = \{w' \mid ww' \in L\} \ .$$

It is extended to quotient with a language in a standard way.

$$LT^{-1} = \{w' \mid w'w \in L, w \in T\}, \quad T^{-1}L = \{w' \mid ww' \in L, w \in T\} \ .$$

Single-word languages $\{w\}$ are abbreviated to w, whenever there is no danger of confusion. By w^r the word w written in reverse is denoted.

2.1.2 Natural numbers

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

By default, systems of equations are assumed to be over sets of numbers, and use operations $\cap$, $\cup$ and $+$. The allowed constants are usually the singletons, unless stated otherwise.

To simplify the notation, the following precedence of operations is assumed: addition has the highest precedence, followed by the intersection and with the union having the least precedence. For example:

$$Y \cup A + B \cap X$$

should be parsed as

$$Y \cup \big((A + B) \cap X\big) \ .$$

A *system of equations* is a system of the form

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

where the unknowns X_i are subsets of $\mathbb{N}$, while φ_j and ψ_j are expressions using union, intersection and addition, as well as singleton constants. A *solution* of such a system is a vector $(S_1, \ldots, S_n)$, such that substituting $X_i = S_i$ for $i = 1, \ldots, n$ in the system turns each equation into an equality.

A solution $(S_1, \ldots, S_n)$ is the *least solution* (*greatest*, *unique*) if for any solution $(S'_1, \ldots, S'_n)$ of the system satisfies $S_i \subseteq S'_i$ ($S_i \supseteq S'_i$, $S_i = S'_i$, respectively) for each $i = 1, \ldots, n$. Classes of least, greatest and unique solutions are of a particular interest, since they are already known to have good properties, as demonstrated by Theorem 1.1 or Kunc's results [29].

2.2 Semantics of conjunctive grammars

As already mentioned, informally speaking, the conjunctive grammars extend the context-free grammars by allowing an operation of unrestricted intersection in the body of each production. Presenting it more formally:

Definition 2.1 (Okhotin [36])**.** A conjunctive grammar is a quadruple $G = (\Sigma, N, P, S)$, in which Σ and N are disjoint finite nonempty sets of terminal and nonterminal symbols respectively; P is a finite set of grammar rules, each of the form

$$A \to \alpha_1 \& \ldots \& \alpha_n \quad (\text{with } A \in N,\ n \geq 1 \text{ and } \alpha_1, \ldots, \alpha_n \in (\Sigma \cup N)^*) \quad (2.1)$$

and $S \in N$ is a nonterminal designated as the start symbol.

A rule (2.1) 'states' that if a string is generated by each α_i, then it is generated by A. This semantics can be formalised using term rewriting, which generalises Chomsky's string rewriting [9].

Definition 2.2 (Okhotin [36])**.** Given a conjunctive grammar G, consider terms over concatenation and conjunction with symbols from $\Sigma \cup N$ as atomic terms. The relation $\Longrightarrow$ of immediate derivability on the set of terms is defined as follows:

- Using a rule $A \to \alpha_1 \& \ldots \& \alpha_n$, a subterm $A \in N$ of any term $\varphi(A)$ can be rewritten as $\varphi(A) \Longrightarrow \varphi(\alpha_1 \& \ldots \& \alpha_n)$.
- A conjunction of several identical strings can be rewritten by one such string: $\varphi(w \& \ldots \& w) \Longrightarrow \varphi(w)$, for every $w \in \Sigma^*$.

The language generated by a term φ is $L_G(\varphi) = \{w \mid w \in \Sigma^*,\ \varphi \Longrightarrow^* w\}$. The language generated by the grammar is $L(G) = L_G(S) = \{w \mid w \in \Sigma^*,\ S \Longrightarrow^* w\}$.

There are advantages in such a semantics. Firstly, it is easy to understand for a human. Secondly, it refers to the well-established grammar derivation defined by Chomsky [9]. Unfortunately, it has some serious drawbacks as well. Firstly, this particular variant of rewriting may seem to be chosen in an arbitrary way, as opposed to the intuitiveness of the context-free grammars. Secondly, while term rewriting is good for understanding and in showing the production of a word, it is very cumbersome in proving properties of the grammars, in particular that there is no derivation. For these reasons, another semantics is employed—it is given with the help of language equations. With each non-terminal, a language variable is associated, and the grammar is represented as a system of language equations. The semantics is given by the least solution of this system. The approach of giving semantics of grammars by language equations was started by Ginsburg and Rice [13], who presented an alternative semantics for context-free grammars.

Definition 2.3 (Okhotin [44])**.** Let $G = (\Sigma, N, P, S)$ be a conjunctive grammar. The *associated system of language equations* is the following system in variables N:

$$A = \bigcup_{A \to \alpha_1 \& \ldots \& \alpha_n \in P} \ \bigcap_{i=1}^{n} \alpha_i \quad , \text{ for all } A \in N \ .$$

Let $(\ldots, L_A, \ldots)$ be its least solution and denote $L_G(A) := L_A$ for each $A \in N$. Define $L(G) := L_G(S)$.

It should be noted, that this semantics gives a clear idea why conjunctive grammars should be viewed as a natural extension of context-free grammars: they are the only extension that generalises the semantics based upon language equations. Moreover, the two semantics do not differ, i.e., it is relatively easy to prove that the language of the grammar is the same under both semantics.

The other reason for supporting the semantics given by language equations is that it can be generalised even to grammars with negation [39]. This topic, however, is not covered in this thesis at all.

As this thesis deals with equations over subsets of natural numbers, they can be equally seen as language equations over a single-letter alphabet. The latter approach is quite cumbersome in general, but in some specific cases unary conjunctive grammars are used, if a presentation of the result using this formalism is a little easier.

2.3 Solutions of systems of equations

There is a natural partial order on the sets of solution of a given system. A solution $(S_1, \dots, S_n)$ is greater than $(S'_1, \dots, S'_n)$, denoted as $(S_1, \dots, S_n) \supseteq (S_1, \dots, S_n)$ if for each $i = 1, \dots, n$ it holds that $S_i \supseteq S'_i$. Note that this order justifies the previously used names of least and greatest solutions, as a solution $(S_1, \dots, S_n)$ is the least solution according to this order if for any solution $(S'_1, \dots, S'_n)$ of the system it holds that

$$(S_1, \dots, S_n) \subseteq (S_1, \dots, S_n) \ .$$

It is the greatest solution according to this order if for any solution $(S'_1, \dots, S'_n)$ it holds that

$$(S_1, \dots, S_n) \supseteq (S_1, \dots, S_n) \ .$$

2.3.1 Least solutions of resolved systems

With every resolved system of equations of the form

$$X = \varphi(X) \ ,$$

where X is a vector of k variables, an operator $\varphi : \left[2^{\mathbb{N}}\right]^k \mapsto \left[2^{\mathbb{N}}\right]^k$ can be associated. Then every solution is a fixpoint of such an operator and vice-versa, in particular, the least fixpoint is the least solution of the system. While for an arbitrary operator φ this is just a rephrasing, for some special types of operators this is quite meaningful.

An operator is monotone, if

$$A \subseteq A' \Longrightarrow \varphi(A) \subseteq \varphi(A') \ ,$$

and $\cup$-*continuous*, if for a sequence $\{A_n\}_{n\geq 1}$ with $A_n \subseteq A_{n+1}$ it holds that

$$\bigcup_n \varphi(A_n) = \varphi(\bigcup_n A_n) \enspace .$$

It can be routinely verified that if an operator is a composition of monotone ($\cup$-continuous) operators, it is itself monotone ($\cup$-continuous).

By well-known Tarski's Fixpoint Theorem [52], each monotone operator has the least fixpoint. Still, this does not give any construction of this operator. An older Kleene's Fixpoint Theorem [28] can be used instead. It deals with monotone and $\cup$-continuous operators and provides a constructive version of the Tarski's Fixpoint Theorem:

Theorem 2.1. *The least fixpoint of a monotone and $\cup$-continuous operator is given by*

$$\bigcup_{i=0}^{\infty} \varphi^i(\varnothing, \dots, \varnothing) \enspace . \tag{2.2}$$

In particular, it is easily seen that for $\left[2^{\mathbb{N}}\right]^k$, each operation $\cup$, $\cap$ and $+$ is monotone and $\cup$-continuous. In fact, the generalisation for languages is true as well, i.e., $\cup$, $\cap$ and $\cdot$, are monotone and $\cup$-continuous. Thus the least solution for resolved systems of equations is given by (2.2).

It should be noted, that the formula given by Kleene (2.2) in fact resembles the derivation. That is, a number is in set on the j^{th} coordinate of $\varphi^i(\varnothing, \dots, \varnothing)$ if and only if it can be derived from a proper terminal using a derivation with a derivation tree of height at most i.

2.3.2 Solutions from natural numbers and positive natural numbers

In many proofs in the following chapters, it is shown that a certain vector of sets is a solution of a given resolved system of equations. Usually this is not the end of the proof, as it remains to show that this solution is the least solution of the system. It turns out, that for a relatively large class of resolved systems there is a unique solution in positive numbers, which is also the least solution in all natural numbers. Informally speaking, the key property is the lack of 0 on the right-hand sides of the equations.

Definition 2.4. Consider expressions with $\cup$, $\cap$ and $+$. Such an expression is called *positive* if it is of the following inductively defined form:

1. a constant set not containing 0 is a positive expression;

2. a sum $\varphi + \psi$ of any two expressions, in which no constant set contains 0, is a positive expression;

3. a union or intersection of two positive expressions is a positive expression.

For a system of equations with resolved positive expressions on the right-hand sides, there is always a unique positive solution:

Lemma 2.1. *Let $X_i = \varphi_i(X_1, \dots, X_n)$ be a system of equations over natural numbers, in which every φ_i is a positive expression. Then it has a unique solution in sets of positive integers.*

This solution is the least solution in sets of natural numbers

The class of positive systems is a variant of the notion of a *proper system* of language equations defined by Autebert et al. [1]. Lemma 2.1 is proved by a straightforward use of the same methods. Such a proof can be easily imagined using the derivation approach: for grammars with which such systems are associated, an ε can be derived either as concatenation of ε's or from a constant; both are not possible.

It should be noted that not much is lost by considering only proper systems, as even for conjunctive grammars one can transform each grammar to an equivalent grammar in *binary normal form* [36], which is a variant of the Chomsky normal form. The systems associated with unary conjunctive grammars in the binary normal form constitute a subclass of proper systems.

Definition 2.5 (Okhotin [36])**.** A conjunctive grammar $G = (\Sigma, N, P, S)$ such that $\varepsilon \notin L(G)$ is said to be in the *binary normal form*, if each rule in P is of the form

$$\begin{aligned} &A \to B_1C_1\& \dots \& B_mC_m, \quad \text{where } m \geq 1; A, B_i, C_i \in N \\ &A \to a, \quad \text{where } A \in N, a \in \Sigma\ . \end{aligned}$$

Theorem 2.2 (Okhotin [36])**.** *For each conjunctive grammar $G = (\Sigma, N, P, S)$ with $\varepsilon \notin L(G)$ there exists and can be effectively constructed a conjunctive grammar $G_0 = (\Sigma, N_0, P_0, S_0)$ in the binary normal form, such that $L(G) = L(G_0)$.*

2.4 Constants and solutions

In the thesis, the majority of lower bounds are shown for systems of equations using only singleton constants. Nevertheless, the actual constructions use a large number of complicated constants, which are represented separately as a solution of some systems of equations using only singleton constants. Such an approach looks natural and intuitively correct, but still it shall be stated formally for completeness.

Also, in the transformations of equations it is usually assumed that the equations to be transformed are of a simple form, while in reality long and complicated equations are constructed. It should be obvious that introducing new variables and splitting the equations solves the problem; again, for completeness this is stated explicitly.

Nevertheless, as all the propositions given here are in fact much more meta-theorems than theorems, they are not going to be addressed directly, but used implicitly instead.

The first two propositions state that values of the unique (least, greatest) solutions may be substituted for the corresponding variables and vice-versa.

Proposition 2.1. *Let a system*

$$\varphi_t(X_1, \ldots, X_m, Y_1, \ldots, Y_n) = \psi_t(X_1, \ldots, X_m, Y_1, \ldots, Y_n)$$

have a unique (least, greatest) solution $X_i = S_i$, $Y_j = T_j$. *Then the system*

$$\varphi_t(S_1, \ldots, S_m, Y_1, \ldots, Y_n) = \psi_t(S_1, \ldots, S_m, Y_1, \ldots, Y_n)$$

in variables $\{Y_1, \ldots, Y_n\}$ *has the unique (least, greatest, respectively) solution* $Y_j = T_j$.

Proposition 2.2. *Let a system*

$$\varphi_t(S_1, \ldots, S_m, Y_1, \ldots, Y_n) = \psi_t(S_1, \ldots, S_m, Y_1, \ldots, Y_n)$$

have a unique (least, greatest) solution $Y_j = T_j$ *and a system*

$$\varphi'_\ell(X_1, \ldots, X_m) = \psi'_\ell(X_1, \ldots, X_m)$$

have a unique (least, greatest) solution $X_i = S_i$. *Then the system*

$$\begin{aligned} \varphi_t(X_1, \ldots, X_m, Y_1, \ldots, Y_n) &= \psi_t(X_1, \ldots, X_m, Y_1, \ldots, Y_n) \ , \\ \varphi'_\ell(X_1, \ldots, X_m) &= \psi'_\ell(X_1, \ldots, X_m) \end{aligned}$$

has the unique (least, greatest, respectively) solution $X_i = S_i$ *and* $Y_j = T_j$.

The other transformation is a decomposition of complex sides of the equation by introducing extra variables:

Proposition 2.3. *Let* $(S_1, \ldots, S_m)$ *be the unique (least, greatest) solution of a system of equations in variables* $(X_1, \ldots, X_m)$ *using union, intersection and addition, and let*

$$\varphi(X_1, \ldots, X_m; \psi(X_1, \ldots, X_m)) = \eta(X_1, \ldots, X_m)$$

be one of its equations. Then a system with a new variable Y, *a new equation* $Y = \psi(X_1, \ldots, X_m)$, *and with the above equation replaced by*

$$\varphi(X_1, \ldots, X_m; Y) = \eta(X_1, \ldots, X_m)$$

has the unique (least, greatest, respectively) solution

$$(S_1, \ldots, S_m; \psi(S_1, \ldots, S_m)) \ .$$

2.5 Representing numbers in positional notation

2.5.1 From strings to numbers

While all the equations defined in this thesis deal with numbers as they are, the principal outlook on these numbers considers their base-k positional notation, for a suitable k. For example, the very first non-trivial example of a least solution of a resolved system of equations defines the set $\{4^n \mid n \in \mathbb{N}\}$, i.e., the set of all numbers that, written in base-4 notation, consist of $\mathtt{1}$ followed by some zeroes. For this reason, a special notation for sets of natural numbers is employed. Fix a base $k \geq 2$ and define the alphabet $\Sigma_k = \{\mathtt{0}, \mathtt{1}, \mathtt{2}, \ldots, k-1\}$ of k-ary digits. Note that a special font is used to distinguish them from numbers. Nevertheless, a standard order on numbers is sometimes applied to them to shorten the notation.

Every string $w = d_{n-1} \ldots d_1 d_0$ with $d_i \in \Sigma_k$ represents the following number:

$$(w)_k = (d_{n-1} \ldots d_1 d_0)_k = \sum_{i=0}^{n-1} d_i \cdot k^i \ .$$

In particular, the empty string ε denotes the number 0. Accordingly, every language $L \subseteq \Sigma_k^*$ defines a certain set of numbers:

$$(L)_k = \left\{ \ (w)_k \ \middle| \ w \in L \ \right\} \ .$$

Note that every number has infinitely many notations with different number of leading zeroes. In some cases it is required that no notations have leading zeroes, that is, $L \subseteq \Sigma_k^* \setminus \mathtt{0}\Sigma_k^*$. There is a bijection between such strings (languages) and non-negative integers (sets thereof).

To simplify the notation and case analysis, the base-k numerical values of the words outside Σ_k are considered, i.e., also the ones using digits greater than $k-1$ and smaller than $\mathtt{0}$. They do not denote proper numbers and are included only to streamline the analysis.

2.5.2 Symbolic addition and subtraction

The technical proofs in the thesis employ symbolic addition and subtraction. They are symbolic in the sense that they are applied to strings and not to actual numbers. The symbolic addition of w' is defined as $w'' = w \boxplus w'$, where w'' is the unique string such that $(w)_k + (w')_k = (w'')_k$ and $|w| = |w''|$. Note that the second condition implicitly states that $(w')_k < k^{|w|} - (w)_k$, i.e., that the result fits in the w's number of digits. This operation is never applied to words not fulfilling this condition. For example, in decimal notation, $\mathtt{0099} \boxplus \mathtt{1} = \mathtt{0100}$.

The symbolic subtraction is defined for a pair of words as $w'' = w \boxminus w'$, where $(w'')_k = (w)_k - (w')_k$ and $|w''| = |w|$. This operation is applied only to w, w' such that $(w)_k \geq (w')_k$, i.e., that the result has meaning. One can write, e.g., $((u \boxminus v)\mathtt{0}^*)_k$ for the set of all numbers with their k-ary notation beginning with the fixed digits determined by the given difference, followed with any number of zeroes.

Symbolic addition and subtraction are extended to languages:

$$L \boxplus w' = \{w \boxplus w' \mid w \in L,\ (w')_k + (w')_k < k^{|w|}\} \ ,$$
$$L \boxminus w' = \{w \boxminus w' \mid w \in L,\ (w)_k \geq (w')_k\} \ ,$$

they are both extended only to addition (subtraction) of a single word, as only such operation is ever used in this thesis. Those operations obviously preserve regularity, hence they can be used inside regular expressions for sets of positional notations, and the sets thus defined remain regular.

2.6 Distributivity

The majority of equations constructed in this thesis use expressions with a particular property: they are *distributive over infinite union*, in the sense that the value of an expression for a set S can be evaluated by substituting all singletons $\{n\} \subseteq S$ into the expression, and then taking the union of all results. This allows treating such an expression in a simplified manner—whenever showing something about it, it is enough to calculate the value of this expression on a single number from the considered set. In fact, while the properties of the first equations given early in this thesis will be shown by an exhaustive case analysis, the later equations will be devised so that the majority of expressions are in fact (partial) functions on numbers.

The following sufficient condition of distributivity covers all distributive expressions developed in this thesis.

Lemma 2.2. *Let $\varphi(X)$ be an expression defined as a composition of the following operations:*

- *the variable X;*
- *constant sets;*
- *union;*
- *intersection with a constant set;*
- *addition of a constant set.*

Then φ is distributive over infinite union, that is, $\varphi(X) = \bigcup_{n \in X} \varphi(\{n\})$.

Proof. Induction on the structure of φ.

Basis. If $\varphi(X) = X$ or $\varphi(X) = C \subseteq \mathbb{N}$, the statement trivially holds.

Induction step I. Let $\varphi(X) = \psi(X) \cup \xi(X)$. By the induction hypothesis, $\psi(X) = \bigcup_{n \in X} \psi(\{n\})$ and $\xi(X) = \bigcup_{n \in X} \xi(\{n\})$. Therefore

$$\begin{aligned}
\varphi(X) &= \psi(X) \cup \xi(X) \\
&= \bigcup_{n \in X} \psi(\{n\}) \cup \bigcup_{n \in X} \xi(\{n\}) \\
&= \bigcup_{n \in X} \big(\psi(\{n\}) \cup \xi(\{n\})\big) \\
&= \bigcup_{n \in X} \varphi(\{n\}) \ .
\end{aligned}$$

Induction step II. If $\varphi(X) = \psi(X) \cap C$ for some $C \subseteq \mathbb{N}$, then, by the induction hypothesis, $\psi(X) = \bigcup_{n \in X} \psi(\{n\})$. Since union and intersection are distributive,

$$\begin{aligned}
\varphi(X) &= \psi(X) \cap C \\
&= \left(\bigcup_{n \in X} \psi(\{n\})\right) \cap C \\
&= \bigcup_{n \in X} \big(\psi(\{n\}) \cap C\big) \\
&= \bigcup_{n \in X} \varphi(\{n\}) \ .
\end{aligned}$$

Induction step III. The case of $\varphi(X) = \psi(X) + C$ is handled similarly to the previous case, using the distributivity of union and concatenation. $\square$

Part I

Resolved equations over sets of natural numbers

Chapter 3

Sets with regular positional notation

It was already found by Leiss [31](Example 1.1) that equations over sets of natural numbers can have non-periodic sets as the unique solutions. Unfortunately, the construction of this example was rather cryptic and gave no idea behind it. Moreover it seemed to base on the structure of the obtained set. In particular there was no generalisation of this example to any larger class of sets. Last not least, it essentially used complementation.

After hearing about conjunctive grammars, D. Wotschke asked A. Okhotin [55] the question of the expressive power of conjunctive grammars over a single letter alphabet. This question was suppose to settle, whether conjunctive grammars possess more expressive power than context-free grammars even in the simplest case of a single-letter alphabet, as context free grammars over such an alphabet generate only regular languages. Due to problems with generalising Leiss example and failed attempts to find any other approach, A. Okhotin conjectured that only regular languages can be generated by unary conjunctive grammars [44], similarly to the case of context-free grammars.

In this chapter the contrary claim is shown: a construction of a resolved system of equations with non-periodic least solution is presented. While this example presents a relatively simple set, the approach and understanding devised as a by-product is much more important. In particular, this construction can be generalised to sets of the form $(L)_k$ for regular L. Moreover, this construction is the cornerstone of the rest of this thesis.

3.1 Toy example

The first example of unary conjunctive grammar generating non-regular language actually comes from attempts of proving that each such language is regular. It was an intention to show that a language of the form $\{a^{2^n} | n \geq 0\}$

cannot be generated by a conjunctive grammar. At the first glance there is no obvious reason to forbid a production $A \to A \cdot A$ to be intersected with some other (specially designed) set, so that only concatenation of two words of the same length could pass through the following intersection; this would yield $L(A) = \{a^{2^n} \mid n \in \mathbb{N}\}$.

So some attempts to construct such a grammar, despite the relatively strong belief that it is not possible, were made. It should be stressed that they were made in order to understand what prevents them from existence. In particular, one attempt was to create grammar generating languages $\{a^{i \cdot 2^n} \mid n \geq 0\}$ for various i in parallel, i.e., by different non-terminals. This approach was a little cumbersome, so power base was replaced with 3 and eventually with 4. After some unsuccessful attempts, the following example of grammar (presented here in a form of a resolved system of equations) was found.

Theorem 3.1. *The least solution of the system*

$$\begin{cases} X_1 &= X_2{+}X_2 \ \cap \ X_1{+}X_3 \ \cup \ \{\mathtt{1}\} \\ X_2 &= X_{12}{+}X_2 \ \cap \ X_1{+}X_1 \ \cup \ \{\mathtt{2}\} \\ X_3 &= X_{12}{+}X_{12} \ \cap \ X_1{+}X_2 \ \cup \ \{\mathtt{3}\} \\ X_{12} &= X_3{+}X_3 \ \cap \ X_1{+}X_2 \ , \end{cases} \tag{3.1}$$

is

$$((\mathtt{10}^*)_4, (\mathtt{20}^*)_4, (\mathtt{30}^*)_4, (\mathtt{120}^*)_4) \ .$$

This is the unique solution of this system in sets of positive integers.

It is enough to show that these sets are a solution. The proof follows by a simple principle: each sum of two sets consists of numbers with a few non-zero digits (in base-4 notation). Intersection of two such sum filters out numbers with unwanted base-4 notation.

proof of Theorem 3.1. To see that this vector is indeed a solution, let us substitute the values $X_i = (i\mathtt{0}^*)_4$ into the equation for X_i. Then the first sum can be transformed as follows:

$$\begin{aligned} X_2 + X_2 &= (\mathtt{20}^*)_4 + (\mathtt{20}^*)_4 \\ &= (\mathtt{10}^+)_4 \cup (\mathtt{20}^*\mathtt{20}^*)_4 \ , \end{aligned}$$

and similarly the second sum equals

$$\begin{aligned} X_1 + X_3 &= (\mathtt{10}^*)_4 + (\mathtt{30}^*)_4 \\ &= (\mathtt{10}^+)_4 \cup (\mathtt{10}^*\mathtt{30}^*)_4 \cup (\mathtt{30}^*\mathtt{10}^*)_4 \ . \end{aligned}$$

Then their intersection is

$$(X_2 + X_2) \cap (X_1 + X_3) = (\mathtt{10}^+)_4 \ ,$$

and after taking union with $\{1\}$, exactly $(\texttt{10}^*)_4 = X_1$ is obtained.

Similar calculations are done for other equations:

$$\begin{aligned} X_{12} + X_2 &= (\texttt{120}^*)_4 + (\texttt{20}^*)_4 \\ &= (\texttt{20}^*\texttt{120}^*)_4 \cup (\texttt{320}^*)_4 \cup (\texttt{20}^+)_4 \cup (\texttt{120}^*\texttt{20}^*)_4 \ , \\ X_1 + X_1 &= (\texttt{10}^*)_4 + (\texttt{10}^*)_4 \\ &= (\texttt{10}^*\texttt{10}^*)_4 \cup (\texttt{20}^*)_4 \ . \end{aligned}$$

Thus

$$\begin{aligned} X_{12} + X_2 \cap X_1 + X_1 \cup \{2\} &= (\texttt{20}^+)_4 \cup \{2\} \\ &= (\texttt{20}^*)_4 \ . \end{aligned}$$

For the third equation similar calculations follow:

$$\begin{aligned} X_{12} + X_{12} &= (\texttt{120}^*)_4 + (\texttt{120}^*)_4 \\ &= (\texttt{120}^*\texttt{120}^*)_4 \cup (\texttt{1320}^*)_4 \cup (\texttt{30}^+)_4 \ , \\ X_1 + X_2 &= (\texttt{10}^*)_4 + (\texttt{20}^*)_4 \\ &= (\texttt{20}^*\texttt{10}^*)_4 \cup (\texttt{10}^*\texttt{20}^*)_4 \cup (\texttt{30}^*)_4 \ . \end{aligned}$$

And hence

$$\begin{aligned} X_{12} + X_{12} \cap X_1 + X_2 \cup \{3\} &= (\texttt{30}^+)_4 \cup \{3\} \\ &= (\texttt{30}^*)_4 \ . \end{aligned}$$

Checking the last equation

$$\begin{aligned} X_3 + X_3 &= (\texttt{30}^*)_4 + (\texttt{30}^*)_4 \\ &= (\texttt{30}^*\texttt{30}^*)_4 \cup (\texttt{120}^*)_4 \ , \\ X_1 + X_2 &= (\texttt{10}^*)_4 + (\texttt{20}^*)_4 \\ &= (\texttt{20}^*\texttt{10}^*)_4 \cup (\texttt{10}^*\texttt{20}^*)_4 \cup (\texttt{30}^*)_4 \ . \end{aligned}$$

After the intersection it follows that

$$X_3 + X_3 \cap X_1 + X_2 = (\texttt{120}^*)_4 \ .$$

It remains to show that the given solution is the unique solution of this system in sets of positive numbers and the least solution in the sets of natural numbers. This follows from the general form of the right-hand sides of this system, in which all occurrences of variables are in sums of two variables, and no constant set contains 0. Thus Lemma 2.1 can be applied and the desired properties hold. □

3.2 Regular notation

The construction presented in Section 3.1 effectively manipulates the leading digits of the numbers in order to construct numbers of a required form. It is natural to ask, how far this method can be extended, i.e., what sets can be constructed in this way. An easy generalisation of this example gives construction of sets of the form $(ij\mathtt{0}^*)_k$, for 'big enough' $k > 1$ and $i, j \in \Sigma_k$, $i \neq 0$.

Then this approach is generalised to a much larger case of sets of the form $(L)_k$ for a regular set L (with some conditions on L imposed). This construction uses the sets of natural numbers of the form $(ij\mathtt{0}^*)_k$, treating them like constants, since it is already know how to construct system with such a solution.

It is technically much easier to deal with larger values of k, as for small k there is only a small set of digits and hence it is harder to filter out unwanted results in intersection. In the following mainly 'big enough' k are considered, that is the proofs focus on $k \geq 9$. Still, this is not restricting, as for $k < 9$ the results can be inferred from constructions for $k' > k$. To this end it is shown that for a given set S the regularity of the base-k notation of its element and base-k' notation are related when k' is a power of k.

Lemma 3.1. *Let $S \subseteq \mathbb{N}$ be a set of numbers, let k and k^m (with $k \geq 2$ and $m \geq 2$) be two bases of positional notation. Then the language $L \subseteq \Sigma_k^* \setminus \mathtt{0}\Sigma_k^*$ such that $(L)_k = S$ is regular if and only if the language $L' \subseteq \Sigma_{k^m}^* \setminus \mathtt{0}\Sigma_{k^m}^*$ such that $(L')_{k^m} = S$ is regular.*

Proof. Define a block code $h : \Sigma_{k^m}^* \to \Sigma_k^*$ as follows: for any k-ary digits $d_0, \ldots, d_{m-1} \in \Sigma_k$, let

$$h\Big(\sum_{i=0}^{m-1} d_i \cdot k^i\Big) = d_{m-1} \ldots d_1 d_0 \enspace ,$$

where $\sum_{i=0}^{m-1} d_i \cdot k^i$ is a single digit in base-k^m notation. Then, clearly, $(w)_{k^m} = (h(w))_k$ for every $w \in \Sigma_{k^m}^*$, and $(L)_{k^m} = (h(L))_k$ for every $L \subseteq \Sigma_{k^m}$.

$\Leftarrow$ Let $S = (L')_{k^m}$, consider the language $\widehat{L} = h(L') \subseteq \Sigma_k^*$. Since regular languages are closed under homomorphisms, $\widehat{L}$ is regular provided that L' is regular. The only remaining problem is that there may be strings starting with $\mathtt{0}$ in $\widehat{L}$. Define $L = \big((\mathtt{0}^*)^{-1} \cdot \widehat{L}\big) \cap \big(\Sigma_k^* \setminus \mathtt{0}\Sigma_k^*\big)$. These operations remove the leading $\mathtt{0}$'es from words in $\widehat{L}$. As all the used operations (effectively) preserve regularity, L is regular and satisfies $(L)_k = (\widehat{L})_k = S$.

$\Rightarrow$ Conversely, let $S = (L)_k$ and construct a modified version of L by appending a limited number of leading zeroes:

$$\widetilde{L} = \big(\{\varepsilon, \mathtt{0}, \mathtt{0}^2, \ldots, \mathtt{0}^{m-1}\} \cdot L\big) \cap \big((\Sigma_k)^m\big)^* \enspace .$$

Obviously, it is possible to append some number of zeroes to every string in L, so that the number of digits in the result is divisible by m. Hence every string in L has a representative in $\widetilde{L}$ with the same numerical value, and $S = (\widetilde{L})_k$. The language $\widetilde{L}$ is regular, as it is obtained as a composition of operations (effectively) preserving regularity.

Consider the language $L' = h^{-1}(\widetilde{L}) \subseteq \Sigma_{k^m}^*$, which is regular by the closure of regular languages under inverse homomorphisms. Every element of $\widetilde{L}$ is represented by an element of L' with the same numerical value, and therefore $S = (L')_{k^m}$. ☐

3.2.1 Sets of numbers with two leading digits fixed

Theorem 3.2. *Let $k \geq 9$ be a natural number. The system*

$$X_{1,j} = \bigcap_{n=1}^{2} X_{k-n,0} + X_{j+n,0} \cup (\mathtt{1}j)_k \quad , \textit{for } j = 0, 1, 2 \ , \tag{3.2}$$

$$X_{i,j} = \bigcap_{n=1}^{2} X_{i-1,k-n} + X_{j+n,0} \cup (ij)_k \qquad , \textit{for } j = 0, 1, 2; \ i > 1 \ , \tag{3.3}$$

$$X_{i,j} = \bigcap_{n=1}^{2} X_{i,j-n} + X_{n,0} \cap X_{i,0} + X_{j,0} \cup (ij)_k \quad , \textit{for } j > 2; i > 1 \ . \tag{3.4}$$

has a unique solution

$$X_{ij} = (ij\mathtt{0}^*)_k, \textit{ for } i > 0, j \geq 0$$

in the set of positive natural numbers.

This is the least solution of this system in the natural numbers.

Proof. It is first checked that $X_{i,j} = (ij\mathtt{0}^*)_k$ is a solution of this system. Then it is argued that it is in fact a unique solution in positive natural numbers and the least solution in the natural numbers.

Consider the first addition in the equations, i.e., for $j = 0, 1, 2$ and $n = 1, 2$:

$$\begin{aligned} X_{k-n,0} + X_{j+n,0} &= ((k-n)\mathtt{0}^+)_4 + ((j+n)\mathtt{0}^+)_4 \\ &= ((k-n)\mathtt{0}^*(j+n)\mathtt{0}^+)_4 \cup ((j+n)\mathtt{0}^*(k-n)\mathtt{0}^+)_4 \cup (1j\mathtt{0}^+)_4 \ . \end{aligned}$$

Now consider $X_{k-1,0} + X_{j+1,0} \cap X_{k-2,0} + X_{j+2,0}$:

$$\begin{aligned} &\Big(X_{k-1,0} + X_{j+1,0}\Big) \cap \Big(X_{k-2,0} + X_{j+2,0}\Big) \\ &\qquad = \Big(((k-1)\mathtt{0}^*(j+1)\mathtt{0}^+)_4 \cup ((j+1)\mathtt{0}^*(k-1)\mathtt{0}^+)_4 \cup (1j\mathtt{0}^+)_4\Big) \\ &\qquad \cap \Big(((k-2)\mathtt{0}^*(j+2)\mathtt{0}^+)_4 \cup ((j+2)\mathtt{0}^*(k-2)\mathtt{0}^+)_4 \cup (1j\mathtt{0}^+)_4\Big) \ . \end{aligned}$$

By comparing the leading digits of the numbers in the sets in question and by using the fact that $1 \le j+1 < j+2 < k-2 < k-1$:

$$\begin{aligned} X_{k-1,0} + X_{j+1,0} \cap X_{k-2,0} + X_{j+2,0} \\ &= \big(((j+1)\mathtt{0}^*(k-1)\mathtt{0}^+)_4 \cup (\mathtt{1}j\mathtt{0}^+)_4\big) \cap (\mathtt{1}j\mathtt{0}^+)_4 \\ &= (\mathtt{1}j\mathtt{0}^+)_4 \ . \end{aligned}$$

Taking the union with $(\mathtt{1}j)_k$ one obtains that

$$\begin{aligned} X_{k-1,0} + & X_{j+1,0} \cap X_{k-2,0} + X_{j+2,0} \cup (\mathtt{1}j)_k \\ &= ((k-1)\mathtt{0}^+)_k + ((j+1)\mathtt{0}^+)_k \cap ((k-2)\mathtt{0}^+)_k + ((j+2)\mathtt{0}^+)_k \\ &\quad \cup (\mathtt{1}j)_k \\ &= (\mathtt{1}j\mathtt{0}^*)_k \ . \end{aligned}$$

Similar calculations can be done for other equations. Consider the second equation—let $j = 0, 1, 2$ and $i > 1$. Then for $n = 1, 2$:

$$\begin{aligned} X_{i-1,k-n} & + X_{j+n,0} \\ &= ((i-1)(k-n)\mathtt{0}^*)_k + ((j+n)\mathtt{0}^+)_k \\ &= ((j+n)\mathtt{0}^*(i-1)(k-n)\mathtt{0}^*)_k \cup ((i-1)(k-n)\mathtt{0}^*(j+n)\mathtt{0}^+)_k \\ &\cup ((i+j+n-1)(k-n)\mathtt{0}^*)_k \cup (\mathtt{1}(i+j+n-1-k)(k-n)\mathtt{0}^*)_k \\ &\cup (ij\mathtt{0}^+)_k \ . \end{aligned}$$

Consider

$$X_{i-1,k-1} + X_{j+1,0} \cap X_{i-1,k-2} + X_{j+2,0} \ .$$

The last non-zero digits of the numbers from this intersection are compared. The numbers from $X_{i-1,k-1}+X_{j+1,0}$ are either in $(ij\mathtt{0}^+)_k$ or have $j+1$ or $k-1$ as last non-zero digit. On the other hand, numbers from $X_{i-1,k-2} + X_{j+2,0}$ belong to $(ij\mathtt{0}^+)_k$ or have either $j+2$ or $k-2$ as the last non-zero digit. Since

$$j+1 < j+2 < k-2 < k-1 \ ,$$

the numbers in $X_{i-1,k-1} + X_{j+1,0} \cap X_{i-1,k-2} + X_{j+2,0}$ are from $(ij\mathtt{0}^+)_k$. Therefore

$$X_{i-1,k-1} + X_{j+1,0} \cap X_{i-1,k-2} + X_{j+2,0} = (ij\mathtt{0}^+)_k \ .$$

Taking the union with $(ij)_k$ one obtains that

$$\begin{aligned} \bigcap_{n=1}^{2} X_{i-1,k-n} + X_{j+n,0} \cup (ij)_k \\ &= \bigcap_{n=1}^{2} ((i-1)(k-n)\mathtt{0}^*)_k + ((j+n)\mathtt{00}^*)_k \cup (ij)_k \\ &= (ij\mathtt{0}^+)_k \cup (ij)_k \\ &= (ij\mathtt{0}^*)_k \ . \end{aligned}$$

In the same way the last equation is treated: let $j > 2$. First the value of the auxiliary addition $X_{i,0} + X_{j,0}$ is calculated:

$$\begin{aligned} X_{i,0} + X_{j,0} &= (i\mathtt{0}^+)_k + (j\mathtt{0}^+)_k \\ &= (i\mathtt{0}^*j\mathtt{0}^+)_k \cup ((i+j)\mathtt{0}^+)_k \cup (\mathtt{1}(i+j-k)\mathtt{0}^+)_k \cup (j\mathtt{0}^*i\mathtt{0}^+)_k \ . \end{aligned}$$

Note that each number in this sum has at most two non-zero digits. This property is used this to filter the numbers in other sums.

Consider $n = 1, 2$ and i such that $i + n \geq k$ (note that this implies $i + j > k$ as well)

$$\begin{aligned} &\Big(X_{i,j-n} + X_{n,0}\Big) \cap \Big(X_{i,0} + X_{j,0}\Big) \\ &\qquad =(i(j-n)\mathtt{0}^*)_k + (n\mathtt{0}^+)_k \cap X_{i,0} + X_{j,0} \\ &\qquad =\Big((n\mathtt{0}^*i(j-n)\mathtt{0}^*)_k \cup (\mathtt{1}(i+n-k)(j-n)\mathtt{0}^*)_k \\ &\qquad\quad \cup (ij\mathtt{0}^+)_k \cup (i(j-n)\mathtt{0}^*n\mathtt{0}^+)_k\Big) \cap \Big(X_{i,0} + X_{j,0}\Big) \\ &\qquad =\Big((\mathtt{1}(i+n-k)(j-n)\mathtt{0}^*)_k \cup (ij\mathtt{0}^+)_k\Big) \cap \big(X_{i,0} + X_{j,0}\big) \ . \end{aligned}$$

with the equality following from the fact that each number in $X_{i,0} + X_{j,0}$ has at most two non-zero digits (the removed sets consist of numbers with at least three non-zero digits). Moreover, if $i + n - k > 0$ also the set $(\mathtt{1}(i+n-k)(j-n)\mathtt{0}^*)_k$ consists of numbers with three non-zero digit, so it is filtered out and the result is

$$(X_{i,j-n} + X_{n,0}) \cap (X_{i,0} + X_{j,0}) = (ij\mathtt{0}^+)_k \ .$$

So consider the case $i = k - n$. Then

$$\begin{aligned} &\Big(X_{i,j-n} + X_{n,0}\Big) \cap \Big(X_{i,0} + X_{j,0}\Big) \\ &\quad =\Big((\mathtt{10}(j-n)\mathtt{0}^*)_k \cup (ij\mathtt{0}^+)_k\Big) \\ &\quad\quad \cap \Big((i\mathtt{0}^*j\mathtt{0}^+)_k \cup (\mathtt{1}(j-n)\mathtt{0}^+)_k \cup (j\mathtt{0}^*i\mathtt{0}^+)_k\Big) \\ &\quad =\Big((\mathtt{10}(j-n)\mathtt{0}^*)_k \cap \big[(i\mathtt{0}^*j\mathtt{0}^+)_k \cup (\mathtt{1}(j-n)\mathtt{0}^+)_k \cup (j\mathtt{0}^*i\mathtt{0}^+)_k\big]\Big) \\ &\quad\quad \cup (ij\mathtt{0}^+)_k \\ &\quad =(ij\mathtt{0}^+)_k \ , \end{aligned}$$

as $i, j > 2$ and $j - n > 0$. Therefore if $i = k - 1$ or $i = k - 2$ then

$$(X_{i,j-1} + X_{1,0}) \cap (X_{i,j-2} + X_{2,0}) \cap (X_{i,0} + X_{j,0}) = (ij\mathtt{0}^+)_k \ .$$

So consider $i < k - 2$.

$$
\begin{aligned}
&(X_{i,j-n} + X_{n,0}) \cap (X_{i,0} + X_{j,0}) \\
=&\Big((i(j-n)\mathtt{0}^*)_k + (n\mathtt{0}^+)_k \Big) \cap \Big(X_{i,0} + X_{j,0} \Big) \\
=&\Big((n\mathtt{0}^* i(j-n)\mathtt{0}^*)_k \cup ((i+n)(j-n)\mathtt{0}^*)_k \cup (ij\mathtt{0}^+)_k \cup (i(j-n)\mathtt{0}^* n\mathtt{0}^+)_k \Big) \\
&\cap \Big(X_{i,0} + X_{j,0} \Big) \\
=&\Big(((i+n)(j-n)\mathtt{0}^*)_k \cup (ij\mathtt{0}^+)_k \Big) \cap \Big(X_{i,0} + X_{j,0} \Big) \ .
\end{aligned}
$$

The sets removed from the first subexpression consisted of numbers with at least three non-zero digits, while $X_{i,0} + X_{j,0}$ consists of numbers with at most two non-zero digits.

When these terms are intersected for $n = 1, 2$

$$
\begin{aligned}
X_{i,j-1} + X_{1,0} \cap X_{i,j-2} + X_{2,0} \cap X_{i,0} + X_{j,0} \\
=&\Big(((i+1)(j-1)\mathtt{0}^*)_k \cup (ij\mathtt{0}^+)_k \Big) \\
&\cap \Big(((i+2)(j-2)\mathtt{0}^*)_k \cup (ij\mathtt{0}^+)_k \Big) \cap \Big(X_{i,0} + X_{j,0} \Big) \\
=&(ij\mathtt{0}^+)_k \cap \Big(X_{i,0} + X_{j,0} \Big) \\
=&(ij\mathtt{0}^+)_k \ ,
\end{aligned}
$$

since the first digit of the number from the first term is $i + 1$ and $i + 2$ for those from the second term. Taking the union with $(ij)_k$ one obtains that for $j > 2$:

$$
\begin{aligned}
&\bigcap_{n=1}^{2} X_{i,j-n} + X_{n,0} \cap X_{i,0} + X_{j,0} \cup (ij)_k \\
=&\bigcap_{n=1}^{2} (i(j-2)\mathtt{0}^*)_k + (2\mathtt{0}^+)_k \cap (i\mathtt{0}^+)_k + (j\mathtt{0}^+)_k \cup (ij)_k \\
=&(ij\mathtt{0}^+)_k \cup (ij)_k \\
=&(ij\mathtt{0}^*)_k \ .
\end{aligned}
$$

Hence this is a solution of this set of equations.

By Lemma 2.1 this system has a unique solution in the positive natural numbers, as each variable appears on the right-hand side only in a sum with other variable and all constants are 0-free. By the same lemma this is the least solution in the natural numbers. ☐

3.2.2 Any regular language

While the construction of all sets of the form $(ij\mathtt{0}^*)_k$ gives a large class of examples of non-periodic sets that are the least solutions of resolved systems, they do not constitute an interesting or particularly useful class of sets on their own. On the other hand, the approach itself seems to be promising and not exploited to the limits.

In order to construct a larger class of sets the idea behind the construction has to be changed a bit. Until now, in the sets $X_{i,j} = (ij\mathtt{0}^*)_k$ the digits i, j were crucial, while the strings of 0'es after them were just a gadget of the construction. For the new sets it works the other way around—they are of the form $X_{i,j,L} = (\{ijw \mid w \in L\})_k$ for some language L. So now i, j are just technical gadgets that are used in the construction to manipulate the leading digits, while the important information is stored in the rest of the digits.

The construction uses several different variables, which represent different sets L and different two leading digits. In order to develop the details, a class of formal languages, from which L comes, has to be chosen. One should imagine that for a fixed ℓ and varying i, j variables $X_{i,j,\ell}$ correspond to one language L_ℓ. Moreover, the technique, if properly generalised, allows manipulating a small amount of leading digits, being more precise, it allows expanding the numbers by a fixed digit. So the intuition is that languages $\{L_\ell\}_\ell$ should express themselves using other languages from this group of languages and left-concatenation with a digit. This is almost the formal definition of the class of regular languages, which have just become the candidate.

Consider an arbitrary regular language $L \subseteq \Sigma_k^+ \setminus 0\Sigma_k^*$. Let $M = \langle \Sigma_k, Q, \delta, F, q_0 \rangle$ be a (deterministic) automaton recognizing L^r. A resolved system of equations with the least solution $X_{i,j,q} = \{(ijw)_k \mid \delta(q_0, w^r) = q\}$, for $i, j \in \Sigma_k$, $i > 0$ and $q \in Q$ will be constructed. Then constructing a resolved equation with the least solution $(L^r)_k$ easily follows. The reversal of the word in the definition of $X_{i,j,q}$ is for technical reasons—while the automata read the word from the left to the right, the constructed systems expand the numbers by digits to the left, i.e., by the digit read first by the automaton.

Constructing such a system using only mentioned variables seems to be cumbersome. In order to simplify the manipulation of the leading digits constants of the form $(ij\mathtt{0}^*)_k$ for $i, j \in \Sigma_k$, $i > 0$ are used. By Theorem 3.2 a separate system with those sets as the unique solution in positive numbers and least in natural numbers (at the same time) can be constructed. For the ease of presentation, constants $(ij\mathtt{0}^*)_k$ are used instead of separate variables with their own subsystems.

Since the construction expands the numbers from each set by one digit it is useful to introduce auxiliary variables also for this intermediate sets. At a proper moment a union of them is taken so that their third leading digit ceases to matter. They are represented by variables $X_{i,j,\ell,q}$ with the intended solution $\{(ij\ell w)_k \mid \delta(q_0, w^r) = q\}$.

Consider the system:

$$X_{i,j,\ell,q} = \bigcap_{n=0}^{3} (in\mathtt{0}^*)_k + X_{j-n,\ell,q'} \cup (\{ij \mid q_0 = q\})_k, \qquad \text{for } j > 3,\ i \neq 0\ , \tag{3.5}$$

$$X_{i,j,q,\ell} = \bigcap_{n=1}^{4} ((i-1)(j+n)\mathtt{0}^*)_k + X_{k-n,\ell,q'} \cup (\{ij \mid q_0 = q\})_k, \qquad \text{for } j < 4,\ i \neq 0, 1\ , \tag{3.6}$$

$$X_{1,j,\ell,q} = \bigcap_{n=1}^{4} ((k-n)\mathtt{00}^*)_k + X_{j+n,\ell,q'} \cup (\{\mathtt{1}j : q_0 = q\})_k, \qquad \text{for } j < 4\ , \tag{3.7}$$

$$X_{i,j,q} = \bigcup_{\substack{(\ell,q'):\\ \delta(q',\ell)=q}} X_{i,j,\ell,q'}, \qquad \text{for } j \in \Sigma_k, i > 1\ , \tag{3.8}$$

$$S = (L \cap \Sigma_k) \cup \bigcup_{\substack{q,i,j:\\ \delta(q,ji)\in F}} X_{i,j,q}\ . \tag{3.9}$$

It is shown that $X_{i,j,q} = \{(ijw)_k \mid \delta(q_0, w^r) = q\}$ and $X_{i,j,\ell,q} = \{(ij\ell w)_k \mid \delta(q_0, w^r) = q\}$ for $i, j, \ell \in \Sigma_k$, $i \neq 0$ and $q \in Q$ is the least solution of (3.5)–(3.8) for large enough k. Then $S = (L^r)_k$ easily follows. Since the right-hand sides do not contain constants that include 0 as an element it easily follows that all the expressions on the right-hand sides are proper and so Lemma 2.1 is applicable to this system. Therefore the natural way of proving this claim is by showing that the specified sets in fact are a solution.

To this end the value of the right-hand sides are inspected, under substitution $X_{i,j,q} = \{(ijw)_k | \delta(q_0, w^r) = q\}$ and $X_{i,j,\ell,q} = \{(ij\ell w)_k | \delta(q_0, w^r) = q\}$.

Lemma 3.2. *For $k \geq 9$, $j > 3$ and $i \neq 0$*

$$\bigcap_{n=0}^{3} (in\mathtt{0}^*)_k + X_{j-n,\ell,q} = X_{i,j,\ell,q}\ ,$$

assuming $X_{i',j',q'} = \{(i'j'w)_k \mid \delta(q_0, w^r) = q'\}$ and $X_{i',j',\ell,q'} = \{(i'j'\ell w)_k \mid \delta(q_0, w^r) = q'\}$ for each i', j', $\ell \in \Sigma_k$, $i' \neq 0$ and $q' \in Q$.

Proof. The proof follows the same principle as the proof of Theorem 3.2, though this time it is a little more complicated, as sets with many unspecified digits are added and intersected. Since the numbers from one of the sets in the addition have specified only two leading digits while their other digits may be arbitrary, the natural attempt to prove the claim of the lemma is to consider only the leading digits of the numbers from the intersection.

Fix $n \in \{0, 1, 2, 3\}$ and consider numbers from the set

$$(in\mathtt{0}^*)_k + (\{(j-n)\ell w \mid \delta(q_0, w^r) = q\})_k \ .$$

Each number p from this sum can be represented as a sum of two numbers p', p'', such that p' has two leading digits i, n and p'' has leading digits $j-n$, ℓ. We inspect the digits of their sum $p' + p''$. This depends of relative positions of leading digits in them. If i and j are on the same position, then p is in the set:

$$\begin{aligned} A_{i,j,\ell,n} =& ((i+j-n)(\ell+n)\Sigma_k^*)_k \cup ((i+j-n+1)(\ell+n-k)\Sigma_k^*)_k \\ & \cup (\mathtt{1}(i+j-n-k)(\ell+n)\Sigma_k^*)_k \\ & \cup (\mathtt{1}(i+j-n+1-k)(\ell+n-k)\Sigma_k^*)_k \ . \end{aligned}$$

The listed sets describe the possible situations: whether there is a carry from position of $\ell + n$ or not and whether there is a carry from position of $i + j - n$.

If the position of i is exactly one greater than the position of $j - n$ then $p \in X_{i,j,\ell,q}$.

If i is on the position smaller than $j - n$ then p is in the set

$$\begin{aligned} B_{i,j,\ell,n} =& ((j-n)\{\ell, \ell+1, \ell+i, \ell+i+1\}\Sigma_k^*)_k \\ & \cup ((j-n+1)\{\ell+1-k, \ell+i-k, \ell+i+1-k\}\Sigma_k^*)_k \\ & \cup (\mathtt{1}(j-n+1-k)\{\ell+1-k, \ell+i-k, \ell+i+1-k\}\Sigma_k^*)_k \ . \end{aligned}$$

If the position of i is greater by more than 1 than the position of $j - n$ then p is in the set

$$(in\Sigma_k^*)_k \ .$$

As the number p was chosen arbitrarily, it can be concluded that

$$\bigcap_{n=0}^{3} (in\mathtt{0}^*)_k + X_{j-n,\ell,q} \subseteq X_{i,j,\ell,q} \cup \bigcap_{n=0}^{3} (A_{i,j,\ell,n} \cup B_{i,j,\ell,n} \cup (in\Sigma_k^*)_k)$$

and it should be shown that the intersection on the right-hand side is empty. To this end the distributivity of union and intersection is applied and then intersections of atomic expressions are inspected. Consider first $A_{i,j,\ell,n}$ and $A_{i,j,\ell,n'}$ for $n \neq n'$. It is shown that their intersection is empty.

Claim 3.1. *$A_{i,j,\ell,n} \cap A_{i,j,\ell,n'} = \varnothing$ for $n \neq n'$.*

Proof. Suppose that there is a number in the intersection. The first possibility is that it comes from one of the two first subexpressions for $A_{i,j,\ell,n}$ and from one of the two last subexpression for $A_{i,j,\ell,n'}$, i.e., that the following intersection is non-empty

$$\Big(((i+j-n)(\ell+n)\Sigma_k^*)_k \cup ((i+j-n+1)(\ell+n-k)\Sigma_k^*)_k\Big)\cap$$
$$\Big((\mathtt{1}(i+j-n'-k)(\ell+n')\Sigma_k^*)_k \cup (\mathtt{1}(i+j-n'+1-k)(\ell+n'-k)\Sigma_k^*)_k\Big) \ .$$

Then $i+j-n'+1 \geq k$ and so $i+j \geq k-1$. Also the leading digit of any number from sets in this intersection is $\mathtt{1}$, since both the subexpressions taken from $A_{i,j,\ell,n'}$ consist of sets of numbers with $\mathtt{1}$ as a leading digit. Hence $i+j-n = 1$ or $i+j+1-n = 1$. This allows estimating $i+j \leq 1+n \leq 4$, contradiction.

The same argument, by symmetry, applies to intersection of last two subexpressions of $A_{i,j,\ell,n}$ and the first two of $A_{i,j,\ell,n'}$, i.e.,

$$\Big((\mathtt{1}(i+j-n-k)(\ell+n)\Sigma_k^*)_k \cup (\mathtt{1}(i+j-n+1-k)(\ell+n-k)\Sigma_k^*)_k\Big)$$
$$\cap\Big(((i+j-n')(\ell+n')\Sigma_k^*)_k \cup ((i+j-n'+1)(\ell+n'-k)\Sigma_k^*)_k\Big)$$

is empty.

So suppose that there is a number that belongs to the

$$((i+j-n)(\ell+n)\Sigma_k^*)_k \cap ((i+j-n')(\ell+n')\Sigma_k^*)_k \ .$$

But they clearly describe the sets of numbers with different leading digits, hence the intersection is empty. Similarly for the other subexpressions:

$$((i+j-n+1)(\ell+n-k)\Sigma_k^*)_k \cap ((i+j-n'+1)(\ell+n'-k)\Sigma_k^*)_k = \varnothing \ ,$$
$$(\mathtt{1}(i+j-n-k)(\ell+n)\Sigma_k^*)_k \cap (\mathtt{1}(i+j-n'-k)(\ell+n')\Sigma_k^*)_k = \varnothing \ ,$$
$$(\mathtt{1}(i+j-n+1-k)(\ell+n-k)\Sigma_k^*)_k$$
$$\cap(\mathtt{1}(i+j-n'+1-k)(\ell+n'-k)\Sigma_k^*)_k = \varnothing \ .$$

with the only exception, that in the second and third cases, second leading digits are considered.

Consider now

$$((i+j-n)(\ell+n)\Sigma_k^*)_k \cap ((i+j-n'+1)(\ell+n'-k)\Sigma_k^*)_k \ .$$

If the intersection is non-empty, then $\ell+n = \ell+n'-k$, which forces $n'-n = k$, which cannot be satisfied. Similarly

$$((i+j-n+1)(\ell+n-k)\Sigma_k^*)_k \cap ((i+j-n')(\ell+n')\Sigma_k^*)_k = \varnothing \ .$$

So look now at the the last cases:

$$(\mathtt{1}(i+j-n-k)(\ell+n)\Sigma_k^*)_k \cap (\mathtt{1}(i+j-n'-k+1)(\ell+n'-k)\Sigma_k^*)_k \ .$$

As in the previous case, if the intersection is non-empty, then $\ell+n = \ell+n'-k$, which cannot happen. By symmetry, also

$$(\mathtt{1}(i+j-n-k+1)(\ell+n-k)\Sigma_k^*)_k \cap (\mathtt{1}(i+j-n'-k)(\ell+n')\Sigma_k^*)_k = \varnothing \ .$$

□

In a similar fashion we deal with $C_{i,j,\ell,n}$. We show that for n, n', n'' pairwise distinct the intersection $B_{i,j,\ell,n} \cap B_{i,j,\ell,n'} \cap B_{i,j,\ell,n''}$ is empty.

Claim 3.2. *For n, n', n'' pairwise different $B_{i,j,\ell,n} \cap B_{i,j,\ell,n'} \cap B_{i,j,\ell,n''} = \varnothing$ for each $j > 0$ and $i \neq 0$.*

Proof. Each of the considered sets consists of three subexpressions. Suppose that the intersection of any two corresponding subexpressions is empty, i.e., consider $n_1 \neq n_2$ and:

$$((j-n_1)\{\ell,\ell+1,\ell+i,\ell+i+1\}\Sigma_k^*)_k \cap ((j-n_2)\{\ell,\ell+1,\ell+i,\ell+i+1\}\Sigma_k^*)_k \ .$$

Those sets consist of numbers with different leading digit, so their intersection is empty. Similarly for other subexpressions, let again $n_1 \neq n_2$:

$$\begin{aligned} &((j-n_1+1)\{\ell+1-k,\ell+i-k,\ell+i+1-k\}\Sigma_k^*)_k \\ \cap&((j-n_2+1)\{\ell+1-k,\ell+i-k,\ell+i+1-k\}\Sigma_k^*)_k \end{aligned}$$

is empty as it consists of the sets of numbers with different leading digit: $j-n_1+1 \neq j-n_2+1$. The last case follows by the same principle:

$$\begin{aligned} &(\mathtt{1}(j-n_1+1-k)\{\ell+1-k,\ell+i-k,\ell+i+1-k\}\Sigma_k^*)_k \\ \cap&(\mathtt{1}(j-n_2+1-k)\{\ell+1-k,\ell+i-k,\ell+i+1-k\}\Sigma_k^*)_k = \varnothing \ , \end{aligned}$$

this time though we focus on second leading digit.

So if the intersection $B_{i,j,\ell,n} \cap B_{i,j,\ell,n'} \cap B_{i,j,\ell,n''}$ is non-empty, the intersection of three subexpressions non corresponding to each other is non-empty. So consider

$$\begin{aligned} &((j-n)\{\ell,\ell+1,\ell+i,\ell+i+1\}\Sigma_k^*)_k \\ \cap&((j-n'+1)\{\ell+1-k,\ell+i-k,\ell+i+1-k\}\Sigma_k^*)_k \\ \cap&(\mathtt{1}(j-n''+1-k)\{\ell+1-k,\ell+i-k,\ell+i+1-k\}\Sigma_k^*)_k \ . \end{aligned}$$

Then $j-n = j-n'+1 = \mathtt{1}$, in particular $j \leq 3$, as $n' \leq 3$. On the other hand $j-n''+1 \geq k$, so $j \geq k-1$, contradiction. □

While it is already known that the intersection of three different $B_{i,j,\ell,n}$'s is empty, in the later proof some more specific knowledge on the non-empty intersection of two such sets is needed.

Claim 3.3. *Suppose that $B_{i,j,\ell,n} \cap B_{i,j,\ell,n'} \neq \varnothing$ for $n < n'$. Then $n = n' - 1$, $j - n + 1 < k$, $i = k - 1$ and*

$$B_{i,j,\ell,n} \cap B_{i,j,\ell,n'} \subseteq ((j-n)\ell\Sigma_k^*)_k \ .$$

Proof. Assume for the sake of contradiction that $j - n + 1 \geq k$. Then

$$B_{i,j,\ell,n} = (\mathtt{1}(j-n+1-k)\{\ell+1-k, \ell+i-k, \ell+i+1-k\}\Sigma_k^*)_k \ .$$

as all other subexpression are empty. On the other hand, $j - n' + 1 < k$ (since $j < k$) and thus

$$\begin{aligned} B_{i,j,\ell,n'} =& ((j-n')\{\ell, \ell+1, \ell+i, \ell+i+1\}\Sigma_k^*)_k \\ &\cup ((j-n'+1)\{\ell+1-k, \ell+i-k, \ell+i+1-k\}\Sigma_k^*)_k \ . \end{aligned}$$

so the leading digits of numbers from those sets are different: by the a contrario assumption $j \geq k + n - 1 \geq k - 1$ and thus $j - n' \geq k - 1 - n' \geq k - 4 > 1$. The obtained contradiction shows that $j - n + 1 < k$. So assume now, for the sake of contradiction, that $n < n' - 1$. Since $j - n + 1 < k$:

$$\begin{aligned} B_{i,j,\ell,n} =& ((j-n)\{\ell, \ell+1, \ell+i, \ell+i+1\}\Sigma_k^*)_k \\ &\cup ((j-n+1)\{\ell+1-k, \ell+i-k, \ell+i+1-k\}\Sigma_k^*)_k \ , \\ B_{i,j,\ell,n'} =& ((j-n')\{\ell, \ell+1, \ell+i, \ell+i+1\}\Sigma_k^*)_k \\ &\cup ((j-n'+1)\{\ell+1-k, \ell+i-k, \ell+i+1-k\}\Sigma_k^*)_k \end{aligned}$$

and their intersection is empty, as numbers in $B_{i,j,\ell,n}$ start with digit at least $j - n$, while numbers in $B_{i,j,\ell,n'}$ start with digit at most $j - n' + 1 < j - n - 1 + 1 = j - n$. Using this we conclude that if $B_{i,j,n} \cap B_{i,j,n-1}$ is non-empty then $n' = n + 1$

$$\begin{aligned} B_{i,j,\ell,n} \cap B_{i,j,\ell,n+1} =& ((j-n)\{\ell, \ell+1, \ell+i, \ell+i+1\}\Sigma_k^*)_k \\ &\cap ((j-n)\{\ell+1-k, \ell+i-k, \ell+i+1-k\}\Sigma_k^*)_k \ . \end{aligned}$$

Let us inspect the second leading digits of numbers from this intersection. As

$$\ell+1-k < \ell+i-k < \ell+i+1-k \leq \ell < \ell+1 < \ell+i < \ell+i+1$$

it can be concluded that $\ell + i + 1 - k = \ell$ which yields $i = k - 1$. Knowing this it can be conclude that

$$B_{i,j,\ell,n} \cap B_{i,j,\ell,n+1} = ((j-n)\ell\Sigma_k^*)_k \ ,$$

which establishes the claim. □

The last intersection of unwanted terms is inspected.

Claim 3.4. *For $n \neq n'$*

$$(in\Sigma_k^*)_k \cap (in'\Sigma_k^*)_k = \varnothing \ .$$

Proof. The first set consists only of numbers with second leading digit n while the second consist of numbers with second leading digit $n' \neq n$. $\square$

Going back to

$$\bigcap_{n=0}^{3} \Big(A_{i,j,\ell,n} \cup B_{i,j,\ell,n} \cup (in\Sigma_k^*)_k \Big) \ ,$$

assume that the intersection is non-empty. As intersection of $A_{i,j,\ell,n} \cap A_{i,j,\ell,n'}$ is empty for $n \neq n'$ by Claim 3.1; so is $(in\Sigma_k^*)_k \cap (in'\Sigma_k^*)_k$ by Claim 3.4; and $B_{i,j,\ell,n} \cap B_{i,j,\ell,n'} \cap B_{i,j,\ell,n''}$ by Claim 3.2, it can be inferred that there are pairwise distinct n, n', n'', n''' such that the intersection

$$A_{i,j,\ell,n} \cap B_{i,j,\ell,n'} \cap B_{i,j,\ell,n''} \cap (in'''\Sigma_k^*)_k$$

is non-empty. By Claim 3.2 this implies that $B_{i,j,\ell,n'} \cap B_{i,j,\ell,n''} \subseteq ((j - n')\ell\Sigma_k^*)_k$ and $i = k-1$. So it is enough to consider the intersection

$$A_{k-1,j,\ell,n} \cap ((j-n')\ell\Sigma_k^*)_k \cap ((k-1)n'''\Sigma_k^*)_k$$

and in particular conclude that $j - n' = k - 1$, which implies that $n' = 0$ and $j = k-1$. Recall the definition of $A_{k-1,k-1,\ell,n}$:

$$\begin{aligned} A_{k-1,k-1,\ell,n} =& (((k-1)+(k-1)-n)(\ell+n)\Sigma_k^*)_k \\ & \cup (((k-1)+(k-1)-n+1)(\ell+n-k)\Sigma_k^*)_k \\ & \cup (\mathtt{1}((k-1)+(k-1)-n-k)(\ell+n)\Sigma_k^*)_k \\ & \cup (\mathtt{1}((k-1)+(k-1)-n+1-k)(\ell+n-k)\Sigma_k^*)_k \ . \end{aligned}$$

Since $2k-2-n > k$, only the last two subexpressions define proper numbers and therefore this can be simplified to

$$A_{k-1,k-1,\ell,n} = (\mathtt{1}(k-n-2)(\ell+n)\Sigma_k^*)_k \cup (\mathtt{1}(k-n-1)(\ell+n-k)\Sigma_k^*)_k \ .$$

But then the intersection of this set with $(in'''\Sigma_k^*)_k = ((k-1)n'''\Sigma_k^*)_k$ is empty.

Hence it is concluded that

$$\bigcap_{n=0}^{3} \Big(A_{i,j,\ell,n} \cup B_{i,j,\ell,n} \cup (in\Sigma_k^*)_k \Big) = \varnothing$$

and thus

$$\bigcap_{n=0}^{3} (in\mathtt{0}^*)_k + X_{j-n,\ell,q} \subseteq X_{i,j,\ell,q} \ .$$

What is left to show is that

$$X_{i,j,\ell,q} \subseteq \bigcap_{n=0}^{3} (in\mathtt{0}^*)_k + X_{j-n,\ell,q} \ .$$

So take any $p = (ij\ell w)_k \in X_{i,j,\ell,q}$, i.e., such that $\delta(q_0, w^r) = q$. Fix $n \in \{0, 1, 2, 3\}$. Then $((j-n)\ell w)_k \in X_{j-n,\ell,q}$ by assumption of the lemma and $(in\mathtt{0}^{|w|+1})_k \in (in\mathtt{0}^*)_k$. Hence

$$p = (ij\ell w)_k = ((j-n)\ell w)_k + (in\mathtt{0}^{|w|+1})_k \in X_{j-n,\ell,q} + (in\mathtt{0}^*)_k \ .$$

Taking the intersection over possible n and noting that p was chosen arbitrarily:

$$X_{i,j,\ell,q} \subseteq \bigcap_{n=0}^{3} (in\mathtt{0}^*)_k + X_{j-n,\ell,q} \subseteq X_{i,j,\ell,q} \ ,$$

which ends the proof. □

The techniques and approach used to show Lemma 3.2 are now going to be used to show analogous results concerning (3.6) and (3.7) .

Lemma 3.3. *For $k \geq 9$, $j < 4$ and $i \neq 0, 1$*

$$\bigcap_{n=1}^{4} ((i-1)(j+n)\mathtt{0}^*)_k + X_{k-n,\ell,q} = X_{i,j,\ell,q} \ ,$$

under the assumption that $X_{i',j',q'} = \{(i'j'w)_k \mid \delta(q_0, w^r) = q'\}$ and $X_{i',j',\ell,q'} = \{(i'j'\ell w)_k \mid \delta(q_0, w^r) = q'\}$ for each i', j', $\ell \in \Sigma_k$, $i' \neq 0$ and $q' \in Q$.

Proof. Fix $n \in \{1, 2, 3, 4\}$ and consider a number p from the set

$$((i-1)(j+n)\mathtt{0}^*)_k + (\{(k-n)\ell w \mid \delta(q_0, w^r) = q\})_k \ .$$

Each such a number number is a sum of two numbers p' and p'', each from one summand. Number p' has leading digits $i-1$ and $j+n$ while p'' has $k-n$ and ℓ. Consider the possible leading digits of $p' + p''$. Depending on the relative position of the digits $i-1$ and $k-n$, p is in one of the specified sets: if $i-1$ is on position greater by at least two than $k-n$ number p satisfies

$$p \in ((i-1)(j+n)\Sigma_k^*)_k \ .$$

When $i-1$ is on position exactly one greater than $k-n$ then:

$$p \in (\{ij\ell w \mid \delta(q_0, w^r) = q\})_k = X_{i,j,\ell,q} \ .$$

When $i-1$ is on the same position as $k-n$, number p is in the set

$$\begin{aligned}C_{i,j,\ell,n} =&\big((k+i-1-n)(\ell+j+n)\Sigma_k^*\big)_k \cup \big((k+i-n)(\ell+j+n-k)\Sigma_k^*\big)_k \\ &\cup \big(\mathtt{1}(i-1-n)(\ell+j+n)\Sigma_k^*\big)_k \cup \big(\mathtt{1}(i-n)(\ell+j+n-k)\Sigma_k^*\big)_k \ ,\end{aligned}$$

with various subsets corresponding to the cases, when there is a carry from position of sum $(i-1)+(k-n)$ or not and when there is a carry from position of $\ell+(j+n)$ or not.

The last possibility is that $i-1$ is on position strictly smaller than $k-n$. Then p is in set

$$\begin{aligned}D_{i,j,\ell,n} =&\big((k-n)\{\ell,\ell+1,\ell+i-1,\ell+i\}\Sigma_k^*\big)_k \\ &\cup \big((k-n+1)\{\ell+1-k,\ell+i-1-k,\ell+i-k\}\Sigma_k^*\big)_k \\ &\cup \big(\mathtt{1}(-n+1)\{\ell+1-k,\ell+i-1-k,\ell+i-k\}\Sigma_k^*\big)_k \ .\end{aligned}$$

Then

$$\begin{aligned}&\bigcap_{n=1}^{4}((i-1)(j+n)\mathtt{0}^*)_k + X_{k-n,\ell,q} \subseteq \\ &\qquad \subseteq \bigcap_{n=1}^{4}\Big(\big((i-1)(j+n)\Sigma_k^*\big)_k \cup C_{i,j,\ell,n} \cup D_{i,j,\ell,n}\Big) \cup X_{i,j,\ell,q}\end{aligned}$$

and our goal is to show that the intersection on the right-hand side is empty. For the sake of contradiction, assume that it is not. Then there is $n \neq n'$ such that one of the following is not empty:

$$\big((i-1)(j+n)\Sigma_k^*\big)_k \cap \big((i-1)(j+n')\Sigma_k^*\big)_k, \quad C_{i,j,\ell,n} \cap C_{i,j,\ell,n'}, \quad D_{i,j,\ell,n} \cap D_{i,j,\ell,n'} \ .$$

We arrive to contradiction by showing, that in fact all of those intersections are empty.

Claim 3.5. *For $n \neq n'$*

$$\big((i-1)(j+n)\Sigma_k^*\big)_k \cap \big((i-1)(j+n')\Sigma_k^*\big)_k = \varnothing \ .$$

Proof. This is clearly empty, as those sets consist of numbers with different second leading digit. ∎

Similarly consider $C_{i,j,\ell,n} \cap C_{i,j,\ell,n'}$, without loss of generality it may be assumed that $n > n'$

Claim 3.6. *For $n > n'$*

$$C_{i,j,\ell,n} \cap C_{i,j,\ell,n'} = \varnothing \ .$$

Proof. By taking into account that $i > 1$, $j < 4$, $k \geq 9$ and $1 \leq n < n' \leq 4$ it can be deduced that

$$1 < k + i - 1 - n < k + i - n \leq k + i - 1 - n' < k + i - n' \ .$$

Then using the distributivity of union and intersection and comparing the leading digits of numbers from $C_{i,j,\ell,n}$ and $C_{i,j,\ell,n'}$ it can be observed that $C_{i,j,\ell,n} \cap C_{i,j,\ell,n'}$ is a union of

$$\begin{aligned}&\big[((k+i-1-n)(\ell+j+n)\Sigma_k^*)_k \cup ((k+i-n)(\ell+j+n-k)\Sigma_k^*)_k\big]\\ \cap&\big[((k+i-1-n')(\ell+j+n')\Sigma_k^*)_k \cup ((k+i-n')(\ell+j+n'-k)\Sigma_k^*)_k\big]\end{aligned}$$

and

$$\begin{aligned}&\big[(\mathtt{1}(i-1-n)(\ell+j+n)\Sigma_k^*)_k \cup (\mathtt{1}(i-n)(\ell+j+n-k)\Sigma_k^*)_k\big]\\ \cap&\big[(\mathtt{1}(i-1-n')(\ell+j+n')\Sigma_k^*)_k \cup (\mathtt{1}(i-n')(\ell+j+n'-k)\Sigma_k^*)_k\big] \ .\end{aligned}$$

The former can be shown to be empty by comparing the second leading digits from sets taking part in the intersection, as

$$\ell + j + n > \ell + j + n' > \ell + j + n - k > \ell + j + n' - k \ ,$$

i.e., the sets taking part in the intersection consist of numbers with different second leading digits.

The latter is also empty, which can be shown by the similar argument applied to the third leading digits, which are the same as the second leading digits in the former case. Thus

$$C_{i,j,\ell,n} \cap C_{i,j,\ell,n'} = \varnothing \ ,$$

which establishes the claim. ∎

Similar argument is given for intersection of $D_{i,j,\ell,n} \cap D_{i,j,\ell,n'}$ for $n < n'$.

Claim 3.7. *For $n < n'$ it holds that*

$$D_{i,j,\ell,n} \cap D_{i,j,\ell,n'} \ .$$

Proof. Note, that since $n' > n > 0$, $-n' + 1 < 0$ and therefore it does not define a proper digit. Thus

$$\begin{aligned}D_{i,j,\ell,n'} \cap D_{i,j,\ell,n} =&\Big(((k-n')\{\ell,\ell+1,\ell+i-1,\ell+i\}\Sigma_k^*)_k\\ &\cup((k-n'+1)\{\ell+1-k,\ell+i-1-k,\ell+i-k\}\Sigma_k^*)_k\Big)\\ \cap&\Big(((k-n)\{\ell,\ell+1,\ell+i-1,\ell+i\}\Sigma_k^*)_k\\ &\cup((k-n+1)\{\ell+1-k,\ell+i-1-k,\ell+i-k\}\Sigma_k^*)_k\\ &\cup(\mathtt{1}(-n+1)\{\ell+1-k,\ell+i-1-k,\ell+i-k\}\Sigma_k^*)_k\Big) \ .\end{aligned}$$

This expression is transformed using the distributivity of intersection and union. Then the values of the leading digits of the numbers in the intersected sets can be compared. As

$$1 < k - n' < k - n' + 1 \leq k - n < k - n + 1$$

it can be concluded that

$$\begin{aligned} D_{i,j,\ell,n'} \cap D_{i,j,\ell,n} =& ((k-n'+1)\{\ell+1-k, \ell+i-1-k, \ell+i-k\}\Sigma_k^*)_k \\ & \cap ((k-n)\{\ell, \ell+1, \ell+i-1, \ell+i\}\Sigma_k^*)_k \ . \end{aligned}$$

But this intersection is empty, as the second leading digits of numbers from these sets satisfy

$$\ell + 1 - k \leq \ell + i - 1 - k < \ell + i - k < \ell < \ell + 1 \leq \ell + i - 1 < \ell + i \ ,$$

and so they are different. Thus

$$D_{i,j,\ell,n} \cap D_{i,j,\ell,n'} = \varnothing \ ,$$

which shows the claim. □

Concluding, it was shown in Claims 3.5, 3.6 and 3.7 that for $n \neq n'$

$$\begin{aligned} ((i-1)(j+n)\Sigma^*)_k \cap ((i-1)(j+n')\Sigma^*)_k &= C_{i,j,\ell,n} \cap C_{i,j,\ell,n'} \\ &= D_{i,j,\ell,n} \cap D_{i,j,\ell,n'} \\ &= \varnothing \ . \end{aligned}$$

and therefore

$$\begin{aligned} &\bigcap_{n=1}^{4} ((i-1)(j+n)\mathtt{0}^*)_k + X_{k-n,\ell,q} \\ &\qquad \subseteq \bigcap_{n=1}^{4} (((i-1)(j+n)\Sigma^*)_k \cup C_{i,j,\ell,n} \cup D_{i,j,\ell,n}) \cup X_{i,j,\ell,q} \\ &\qquad = X_{i,j,\ell,q} \ . \end{aligned}$$

What is left to show is that

$$X_{i,j,\ell,q} \subseteq \bigcap_{n=1}^{4} ((i-1)(j+n)\mathtt{0}^*)_k + X_{k-n,\ell,q} \ .$$

Fix $n \in \{1,2,3,4\}$ and consider $p \in X_{i,j,\ell,q}$. There is w such that $p = (ij\ell w)_k$, by lemma assumption, where $\delta(q_0, w^r) = q$. Then $p' = ((k-n)\ell w)_k \in X_{k-n,\ell,q}$, also by lemma assumption, and $p'' = ((i-1)(j+n)\mathtt{0}^{|w|+1})_k \in ((i-1)(j+n)\mathtt{0}^*)_k$. Hence

$$p = p' + p'' \in X_{k-n,\ell,q} + ((i-1)(j+n)\mathtt{0}^*)_k \ .$$

Therefore

$$X_{i,j,\ell,q} \subseteq \bigcap_{n=1}^{4} ((i-1)(j+n)\mathtt{0}^*)_k + X_{k-n,\ell,q} \subseteq X_{i,j,\ell,q} \ ,$$

which concludes the proof. □

The proof for (3.7) follows the same pattern.

Lemma 3.4. *For $k \geq 9$, $i, j \in \Sigma_k$, $i \neq 0$, $j < 4$*

$$\bigcap_{n=1}^{4} ((k-n)\mathtt{0}^+)_k + X_{j+n,\ell,q} = X_{1,j,\ell,q} \ ,$$

assuming $X_{i',j',q'} = \{(i'j'w)_k \mid \delta(q_0, w^r) = q'\}$ and $X_{i',j',\ell,q'} = \{(i'j'\ell w)_k \mid \delta(q_0, w^r) = q'\}$ for each i', j', $\ell \in \Sigma_k$, $i > 0$ and $q' \in Q$.

Proof. Fix $n \in \{1, 2, 3, 4\}$ and consider any two numbers $p' \in ((k-n)\mathtt{0}^+)_k$ and $p'' \in X_{j+n,\ell,q'} = (\{(j+n)\ell w | \delta(q_0, w^r) = q'\})_k$. Then the sum $p = p' + p''$ has a form depending on the position of digit $k-n$ in p' and $j+n$ in p''.

If $k-n$ is on greater position, then p is in the set

$$((k-n)\Sigma_k^*)_k \ .$$

If $k-n$ is on exactly the same position as $j+n$ then p is in the set

$$(\{\mathtt{1}j\ell w \mid \delta(q_0, w^r) = q\})_k = X_{1,j,\ell,q} \ .$$

If $k-n$ is on position smaller than $j+n$ then the result is in set

$$E_{j,n,\ell} = ((j+n)\{\ell, \ell+1, \ell+k-n\}\Sigma_k^*)_k \cup ((j+n+1)\{\ell+1-k, \ell-n\}\Sigma_k^*)_k \ ,$$

with the cases representing whether $k-n$ is on position exactly one smaller than $j+n$ or not and whether there was some carry or not.

Then

$$((k-n)\mathtt{0}\mathtt{0}^*)_k + X_{j+n,\ell,q'} \subseteq X_{1,j,\ell,q} \cup \bigcap_{n=1}^{4} \Big(((k-n)\Sigma_k^*)_k \cup E_{j,n,\ell} \Big)$$

and the goal is to show that $\bigcap_{n=1}^{4}(((k-n)\Sigma_k^*)_k \cup E_{j,n,\ell}) = \varnothing$. Using distributivity this can be reduced to showing that for $n < n'$ both

$$((k-n)\Sigma_k^*)_k \cap ((k-n')\Sigma_k^*)_k \quad \text{and } E_{j,n,\ell} \cap E_{j,n',\ell}$$

are empty.

Consider $((i-1)(j+n)\Sigma_k)_k \cap ((i-1)(j+n')\Sigma_k)_k$. Then it is clearly empty, as it describes the intersection of two sets consisting of numbers with different leading digits.

Consider $E_{j,n,\ell} \cap E_{j,n',\ell}$ for $n < n'$.

$$\begin{aligned} E_{j,n,\ell} \cap E_{j,n',\ell} =& \Big(((j+n)\{\ell, \ell+1, \ell+k-n\}\Sigma_k^*)_k \\ & \cup ((j+n+1)\{\ell+1-k, \ell-n\}\Sigma_k^*)_k \Big) \\ & \cap \Big(((j+n')\{\ell, \ell+1, \ell+k-n'\}\Sigma_k^*)_k \\ & \cup ((j+n'+1)\{\ell+1-k, \ell-n'\}\Sigma_k^*)_k \Big) \ . \end{aligned}$$

Consider the first digit of the numbers from the sets involved in the intersection. As

$$j+n < j+n+1 \leq j+n' < j+n'+1$$

the considered intersection is equal to

$$\begin{aligned} E_{j,n,\ell} \cap E_{j,n',\ell} =& ((j+n+1)\{\ell+1-k, \ell-n\}\Sigma_k^*)_k \\ & \cap ((j+n')\{\ell, \ell+1, \ell+k-n'\}\Sigma_k^*)_k \ . \end{aligned}$$

But then, as

$$\ell+1-k < \ell-n < \ell < \ell+1 < \ell+k-n'$$

the second leading digits in numbers from those sets are different and hence the intersection is empty, i.e.,

$$E_{j,n,\ell} \cap E_{j,n',\ell} = \varnothing \ .$$

Consequently

$$\begin{aligned} ((k-n)\mathtt{0}\mathtt{0}^*)_k + X_{j+n,\ell,q'} \subseteq& X_{1,j,\ell,q} \cup \bigcap_{n=1}^{4} (((k-n)\Sigma_k^*)_k \cup E_{j,n}) \\ =& X_{1,j,\ell,q} \ . \end{aligned}$$

What is left to be shown is that

$$X_{1,j,\ell,q} \subseteq ((k-n)\mathtt{0}\mathtt{0}^*)_k + X_{j+n,\ell,q} \ .$$

So consider any $p \in X_{1,j,\ell,q}$. Fix $n \in \{1,2,3,4\}$. By assumption of the lemma, $p = (\mathtt{1}j\ell w)_k$, where $\delta(q_0, w^r) = q$. By the same assumption, $p' = ((k-n)\mathtt{0}^{|w|+1})_k \in ((k-n)\mathtt{0}\mathtt{0}^*)_k$ and $p'' = ((j+n)\ell w)_k \in X_{j+n,\ell,q}$. Then

$$p = p' + p'' \in ((k-n)\mathtt{0}\mathtt{0}^*)_k + X_{j+n,\ell,q} \ .$$

Since n was chosen arbitrarily, an intersection over its possible values can be taken. As also p was chosen in an arbitrary fashion among elements of $X_{1,j,\ell,q}$, it can be concluded that

$$X_{1,j,\ell,q} \subseteq \bigcap_{n=1}^{4} ((k-n)\mathtt{0}\mathtt{0}^*)_k + X_{j+n,\ell,q} \subseteq X_{1,j,\ell,q} \ ,$$

which completes the proof. $\square$

When basic equations (3.5)–(3.7) have been validated, it is time to validate (3.8) and (3.9). Since they are defined uniquely with respect to $X_{i,j,\ell,q} = \{(ij\ell w)_k \mid \delta(q_0, w^r) = q\}$ (the former) and $X_{i,j,q} = \{(ijw)_k \mid \delta(q_0, w^r) = q\}$ (the latter), the proofs follow by a simple calculation.

Lemma 3.5. *For $k \geq 9$ sets $X_{i,j,q} = \{(ijw)_k \mid \delta(q_0, w^r) = q\}$ and $X_{i,j,\ell,q} = \{(ij\ell w)_k \mid \delta(q_0, w^r) = q\}$ for $i, j, \ell \in \Sigma_k$, $i \neq 0$ and $q \in Q$ are the unique solution in positive natural numbers of* (3.5)*–*(3.8)*.*

This is also the least solution in the natural numbers.

Proof. Firstly it is shown that these sets are in fact a solution. Consider (3.8). Then by Lemmata 3.2–3.4 substituting $\{((j-n)\ell w)_k \mid \delta(q_0, w^r) = q'\}$ for $X_{j-n,\ell,q'}$ on the right-hand side of (3.5)–(3.7) yields $X_{i',j',\ell,q'} = \{(i'j'\ell w)_k \mid \delta(q_0, w^r) = q'\}$. Then substituting these values to the right-hand side of (3.8)

$$\begin{aligned}
X_{i,j,q} = & \bigcup_{\substack{(\ell,q'):\\ \delta(q',\ell)=q}} X_{i,j,\ell,q'} \cup (\{ij \mid q_0 = q\})_k \\
= & \bigcup_{\substack{(\ell,q'):\\ \delta(q',\ell)=q}} (\{ij\ell w \mid \delta(q_0, w^r) = q'\})_k \cup (\{ij \mid q_0 = q\})_k \\
= & (\{ij\ell w \mid \delta(q_0, (w\ell)^r) = q\})_k \cup (\{ij \mid q_0 = q\})_k \\
= & (\{ijw \mid \delta(q_0, w^r) = q\}, |w| > 0)_k \cup (\{ijw \mid \delta(q_0, w^r) = q\}, |w| = 0)_k \\
= & (\{ijw \mid \delta(q_0, w^r) = q\})_k \ .
\end{aligned}$$

Hence it was shown that

$$\begin{aligned}
X_{i,j,q} &= \{(ijw)_k \mid \delta(q_0, w^r) = q\}, \quad \text{for } i, j \in \Sigma_k, i > 0, q \in Q \ , \\
X_{i,j,\ell,q} &= \{(ij\ell w)_k \mid \delta(q_0, w^r) = q\}, \quad \text{for } i, j \in \Sigma_k, i > 0, q \in Q \ .
\end{aligned}$$

is a solution of the system. As the system is proper, thus Lemma 2.1 is applicable and so this is the unique solution in the sets of positive natural numbers as well as the least solution in the sets of natural numbers. ☐

After establishing the technical lemmata, the general representation theorem can be stated and easily proved.

Theorem 3.3. *For every natural $k > 1$ and every regular language $L \subseteq \Sigma_k^+ \setminus 0\Sigma^*$ the set of numbers $(L)_k$ can be represented as a unique solution of a resolved system of equations.*

This solution is the least solution of this system in the natural numbers.

Proof. Suppose first that $k \geq 9$. Consider the system (3.5)–(3.9).

By Lemma 3.5, subsystem (3.5)–(3.8) has a unique solution in positive numbers, which is also the least solution in natural numbers, $X_{i,j,q} =$

$\{(ijw)_k \mid \delta(q_0, w^r) = q\}$, $X_{i,j,\ell,q} = \{(ij\ell w)_k \mid \delta(q_0, w^r) = q\}$ for $i, j, \ell \in \Sigma_k$, $i \neq 0$ and $q \in Q$. Then, as (3.9) defines S uniquely with respect to $X_{i,j,q}$ the whole system has a unique solution in the positive numbers, which is also the least solution in natural numbers.

The value of S in this solution can be easily calculated by substituting values obtained in Lemma 3.5 to (3.9)

$$\begin{aligned}
S &= (L \cap \Sigma_k)_k \cup \bigcup_{\substack{q,i,j:\\ \delta(q,ji) \in F}} X_{i,j,q} \\
&= (L \cap \Sigma_k)_k \cup \bigcup_{\substack{q,i,j:\\ \delta(q,ji) \in F}} \{(ijw)_k \mid \delta(q_0, w^r) = q\} \\
&= (L \cap \Sigma_k)_k \cup \{(w)_k \mid \delta(q_0, w^r) \in F, |w| \geq 2\} \\
&= (L \cap \Sigma_k)_k \cup \{(w)_k \mid w \in L, |w| \geq 2\} \\
&= (L)_k \ .
\end{aligned}$$

What is left to do is the removal of the constants of the form $(ij\mathtt{0}^*)_k$, which are present in the system (3.5)–(3.7). By Theorem 3.2, there is a resolved system such that all sets of the form $(ij\mathtt{0}^*)_k$ are the coordinates of its least solution, which is also the unique solution in positive numbers. Replacing the constants of this form in (3.5)–(3.7) with variables and adding a subsystem from Theorem 3.2 ends the proof for $k \geq 9$.

Consider now $1 < k < 9$ and a language $L \subseteq \Sigma_k^+ \setminus \mathtt{0}\Sigma_k^*$. Let $k' = k^4 > 9$. By Lemma 3.1 the language $L' \subseteq \Sigma_{k'}^+ \setminus \mathtt{0}\Sigma_{k'}^*$, such that $(L')_{k'} = (L)_k$, is regular, as L is regular. And since $k' > 9$, it was already shown in this proof that there is a system such that $(L')_{k'} = (L)_k$ is a component of its unique solution in positive numbers, which is also the least solution in natural numbers. $\square$

Chapter 4

Sets with trellis positional notation

It seems that the methods developed in Chapter 3, at least in the way they have been used so far, were exploited to the limit—they allow modifying only the leading digit and the information about the positional notation is kept in a finite amount of 'memory' (here—variable name). This naturally corresponds to regular languages. Still, the obtained class of sets is not very large. It has an exponentially bounded growth rate, the decision problems are likely to be decidable, etc. A natural attempt to extend this class is to slightly enrich the method used—since the leading digits can be manipulated (or even less formally—the notations of numbers are expanded to the left by a given digit), maybe the same can be done with the ending digit? This may lead to a larger class of numbers. Unfortunately, it looks that the previous techniques are not applicable here.

In order to enforce this approach, a different representation of a word is needed: in the previous construction, a single word w was represented by a number $(ijw)_k$, in particular one word was represented by one number. Now a word w is represented by the numbers $(\mathtt{1}w\mathtt{1000}\ldots\mathtt{0})_k$, i.e., by the numbers with their base-k notation consisting of $\mathtt{1}w\mathtt{1}$ followed by a series of $\mathtt{0}$'es. The two additional $\mathtt{1}$'s are 'sentinels', as they mark the beginning and the end of the word w—note that w may contain both leading and ending $\mathtt{0}$'es. Then going from w to, say, $w\mathtt{3}$ is just changing the $\mathtt{10}$ in the end into $\mathtt{31}$. Moreover, since it is not known in advance, how many ending $\mathtt{0}$'es will be consumed, w is represented as $(\mathtt{1}w\mathtt{10}^*)_k$, i.e., one word is represented by an infinite set of numbers.

This intuition requires a lot of work to be transformed into a formally sound construction. Before that one needs to understand what type of computational device can be simulated in this way. On a high level, it is an automaton with a finite state control, since the state of the computation is uniquely determined by the variable name. Concerning the transition function, the state calculated on a word awb depends on the states calculated on aw and on wb. This corresponds to the idea that the notation of the number is expanded both to the left and the right.

Such a class of automata is well-known in the literature—these are *trellis automata*[11, 12, 18]. The class of *trellis languages* (i.e., those recognised by such automata) includes: arbitrarily dense and arbitrarily sparse languages, languages of computation histories of Turing Machines, linear context-free languages; it is closed under all Boolean operations, etc. This makes it an interesting class of languages in general. Quite surprisingly, the class of trellis languages coincides with the class of linear conjunctive languages [40], which are a strict subclass of conjunctive languages.

The rest of this chapter is dedicated to formalising the concept of trellis automata and simulating its action on notations of numbers by resolved systems of equations over sets of numbers. Such equations are used in the later chapters as a building block for systems of unresolved equations. Moreover, they are the source of all undecidability results for unary conjunctive grammars.

It should be pointed out, that after representing sets of numbers with a regular positional notation, the task of constructing equations with specified least solutions becomes substantially easier, as using constants with a regular positional notation is allowed. It is particularly useful to be able to intersect the sums of two sets with a set with a specified regular notation. Generally speaking, this allows modifying a single digit in one of the input sets.

4.1 Definition of trellis automata

Trellis automata, also known as one-way real-time cellular automata, were studied by Culik, Gruska and Salomaa [11, 12], Ibarra and Kim [18], and others. We explain this concept generally following Culik et al. [11].

A trellis automaton (TA) processes an input string of length $n \geq 1$ using a uniform triangular array of $\frac{n(n+1)}{2}$ processor nodes, as presented in the Fig. 4.1. 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 $I : \Sigma \to Q$. The rest of the nodes compute the function $\delta : Q \times Q \to Q$ 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 $F \subseteq Q$. This is formalised in the below definition.

Definition 4.1. 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,

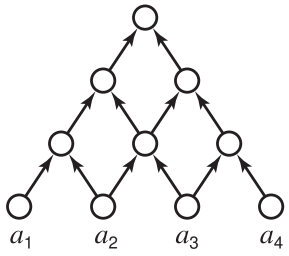

Figure 4.1: A schematic example of a trellis automaton.

- $I : \Sigma \to Q$ is a function that sets the initial states,
- $\delta : Q \times Q \to Q$ is the transition function, and
- $F \subseteq Q$ is the set of final states.

Extend δ to a function $\delta : Q^+ \to Q$ by $\delta(q) = q$ and

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

while I is extended to a homomorphism $I : \Sigma^* \to Q^*$.

Let $L_M(q) = \{w \mid \delta(I(w)) = q\}$ and define $L(M) = \bigcup_{q \in F} L_M(q)$.

The family of languages recognised by trellis automata is known to be closed under all Boolean operations. On the other hand, it is not closed under concatenation [53], using similar methods it can be shown that it is not closed under Kleene star [40]. This family is not closed under homomorphisms in general, but it is closed under block codes [12]. It is also not closed under quotient with regular languages; however, it is closed under quotient with singletons, that is, whenever a language $L \in \Sigma^*$ is recognised by a trellis automaton M, then for every $u, v \in \Sigma^*$, the languages $u^{-1}L$ and Lv^{-1} are also recognised by some trellis automata. These trellis automata can be effectively computed from M and u, v.

Trellis automata over a unary alphabet recognise only regular languages [40]. Together with the above closure property, this implies the following simple result, which will be used later on:

Lemma 4.1. *Let L be a trellis language over an alphabet Σ, let $u, v \in \Sigma^*$ and $a \in \Sigma$. Then the language $L \cap ua^*v$ is regular.*

Proof. Let $K = L \cap ua^*v$. Then the language

$$\widetilde{K} = u^{-1} \cdot K \cdot v^{-1} = \{w \mid uwv \in K\}$$

is a trellis language by the closure of this family under quotient with singletons. Since $\widetilde{K}$ is a unary trellis language, it is regular. Then $K = u\widetilde{K}v$ is regular as well. ☐

Lemma 3.1 assured that for regular positional base-k notations one can consider only 'big enough' languages, as the regularity of base-k notations of a fixed set S was equivalent to regularity of base-k^m notations of S. This property is shared by trellis languages, the proof of this fact is similar as the one of Lemma 3.1, nevertheless it is given for completeness.

Lemma 4.2. *Let $S \subseteq \mathbb{N}$ be a set of numbers, let k and k^m (with $k \geq 2$ and $m \geq 2$) be two bases of positional notation. Then the language $L \subseteq \Sigma_k^* \setminus \mathtt{0}\Sigma_k^*$ such that $(L)_k = S$ is a trellis language if and only if the language $L' \subseteq \Sigma_{k^m}^* \setminus \mathtt{0}\Sigma_{k^m}^*$ such that $(L')_{k^m} = S$ is a trellis language.*

Proof. Consider the same block code as in Lemma 3.1: $h : \Sigma_{k^m}^* \to \Sigma_k^*$ given for any k-ary digits $d_0, \ldots, d_{m-1} \in \Sigma_k$ by

$$h\Big(\sum_{i=0}^{m-1} d_i \cdot k^i\Big) = d_{m-1} \ldots d_1 d_0 \enspace ,$$

where $\sum_{i=0}^{m-1} d_i \cdot k^i$ is a single digit in base-k^m notation.

It was already noted that $(w)_{k^m} = (h(w))_k$ for every $w \in \Sigma_{k^m}$, and $(L)_{k^m} = (h(L))_k$ for every $L \subseteq \Sigma_{k^m}$.

Now the proof follows the same lines as the proof of Lemma 3.1.

$\Leftarrow$ Let $S = (L')_{k^m}$ and define $\widehat{L} = h(L') \subseteq \Sigma_k^*$. A theorem by Culik et al. [12] assures that trellis languages are (effectively) closed under block codes, and h is a block code, so $\widehat{L}$ is (effectively) a trellis language. Moreover, $S = (\widehat{L})_k$. The problem remaining is that some strings in $\widehat{L}$ have leading zeroes. The number of such leading zeroes is at most $m - 1$. Let

$$L = \big(\{\varepsilon, \mathtt{0}, \mathtt{0}^2, \ldots, \mathtt{0}^{m-1}\}^{-1} \cdot \widehat{L}\big) \setminus \mathtt{0}\Sigma_k^* \enspace .$$

Every string in $\widehat{L}$ has a representative in L with all leading zeroes removed. By the closure properties of trellis languages, this is a trellis language, moreover $S = (L)_k$.

$\Rightarrow$ Conversely, let $S = (L)_k$. Construct $\widetilde{L}$ as in Lemma 3.1, i.e., by appending a limited number of leading zeroes to L:

$$\widetilde{L} = \big(\{\varepsilon, \mathtt{0}, \mathtt{0}^2, \ldots, \mathtt{0}^{m-1}\} \cdot L\big) \cap \big((\Sigma_k)^m\big)^* \enspace .$$

Every string in L has a (unique) representative in $\widetilde{L}$ with the same numerical value, and $S = (\widetilde{L})_k$. By closure properties of trellis languages, $\widetilde{L}$ is a trellis language.

Consider the language $L' = h^{-1}(\widetilde{L}) \subseteq \Sigma_{k^m}^*$. It is known that trellis languages are closed under inverse homomorphisms [12, 18], hence L' is a trellis language. Then $S = (L')_{k^m}$, as every element of $\widetilde{L}$ has its counterpart with the same numerical value in L' . ∎

It is quite surprising that trellis automata are connected with conjunctive grammars. They are equivalent to linear conjunctive grammars, whose definition was modelled on linear CFG. The computation equivalence of linear conjunctive grammars and trellis automata is stated as follows:

Theorem 4.1 (Okhotin [40]). *A language $L \subseteq \Sigma^+$ is defined by a linear conjunctive grammar if and only if L is recognised by a trellis automaton. These representations can be effectively transformed into each other.*

4.2 A representation of trellis automata

In this section the informal approach presented at the beginning of this chapter is transformed into a solid construction with provable properties. That is, similarly to Theorem 3.3 which dealt with sets of numbers with regular positional notation, it is asserted that every set of numbers with positional notation recognised by trellis automaton is represented by a least solution of a resolved system of equations.

Theorem 4.2. *For every $k \geq 2$ and for every trellis automaton M over $\Sigma_k = \{\mathtt{0}, \ldots, k-1\}$, such that $L(M) \cap \mathtt{0}\Sigma_k^* = \varnothing$, there exists and can be effectively constructed a resolved system of equations over sets of numbers using the operations $\cup$, $\cap$ and $+$ and singleton constants, such that its least solution contains a component*

$$(L(M))_k = \{n \mid k\text{-ary notation of } n \text{ is in } L(M)\} \ .$$

This is the unique solution in sets of positive natural numbers.

The proof of the theorem is given in the rest of this chapter. The core of the argument is the following simulation of the computation of a trellis automaton by a system of equations. Once this lemma is shown, the rest of the argument used to prove Theorem 4.2 easily follows.

Lemma 4.3. *For every $k \geq 5$ and for every trellis automaton M over Σ_k, there exists and can be effectively constructed a system using constant sets with regular base-k notation, such that one of the components of the unique solution in positive numbers of this system is*

$$(\mathtt{1}(L(M) \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k = \{(\mathtt{1}w\mathtt{1}\mathtt{0}^\ell)_k \mid \ell \geq 0, w \notin (k-1)^*, w \boxplus \mathtt{1} \in L(M)\} \ .$$

This solution is also the least solution in natural numbers.

A brief explanation how the informal intuition was changed into this specific encoding should be given before. When devising the representation for word w several possible obstacles should be taken taken into account:

- while we $L(M) \cap \mathtt{0}\Sigma_k^* = \varnothing$ can be assumed, and hence in the end $w \in L(M)$ does not start with $\mathtt{0}$, its substring can start with $\mathtt{0}$. Thus a way of marking the number of leading $\mathtt{0}$'es is needed;
- similar question arises for w's ending $\mathtt{0}$'es—they should be distinguishable from $\mathtt{0}$'es added in the end.

The natural approach is to add *sentinels* on both side of the word, i.e., to represent w as $(\mathtt{1}w\mathtt{1}\mathtt{0}^*)_k$. This solves the two mentioned problems. Still it creates another: resolved equations are 'monotone' in the sense that if $(w)_k \in \varphi(\{(w')_k\})$ then usually $(w)_k \geq (w')_k$. The current encoding does not guarantee such property—the right sentinel is a problem here: trying to extend w by $\mathtt{0}$ to the right causes the drop of value of representation—$(\mathtt{1}w\mathtt{1}\mathtt{0}^\ell)_k > (\mathtt{1}w\mathtt{0}\mathtt{1}\mathtt{0}^{\ell-1})_k$. This is answered by representing w not by $\mathtt{1}w\mathtt{1}\mathtt{0}^*$ directly, but by $\mathtt{1}(w \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*$ instead. However, in this way $w \in \mathtt{0}^*$ cannot be represented at all and thus $w \in \Sigma_k\mathtt{0}^* \cup \mathtt{0}^*\Sigma_k$ cannot be constructed by referencing to other constructed sets. Therefore such w have to be treated separately.

proof of Lemma 4.3. For a given trellis automaton $M = (\Sigma_k, Q, I, \delta, F)$, define a system of equations over sets of numbers with the set of variables X_q for $q \in Q$, and with an additional variable Y. It will be proved that the least solution of that system is $X_q = S_q$, $Y = S$, where

$$\begin{aligned} S_q &= (\mathtt{1}((L_M(q) \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k \\ &= \{(\mathtt{1}w\mathtt{1}\mathtt{0}^\ell)_k \mid \ell \geq 0, w \notin (k-1)^*, w \boxplus \mathtt{1} \in L_M(q)\} \ , \\ S &= (\mathtt{1}((L(M) \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k \\ &= \{(\mathtt{1}w\mathtt{1}\mathtt{0}^\ell)_k \mid \ell \geq 0, w \notin (k-1)^*, w \boxplus \mathtt{1} \in L(M)\} \ . \end{aligned}$$

The equations are rather complex and will be constructed and explained in stages.

As already mentioned, every string $w \in \Sigma^+ \setminus \mathtt{0}^*$ is represented by numbers of the form $(\mathtt{1}(w \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^\ell)_k$ with $\ell \geq 0$. The main idea of the construction is to represent numbers corresponding to a string $iwj \in L_M(q)$, with $i, j \in \Sigma_k$ and $w \in \Sigma_k^*$, through numbers corresponding to the strings iw and wj. Consider that iwj belongs to $L_M(q)$ if and only if there are states q', q'' with $\delta(q', q'') = q$, $iw \in L_M(q')$ and $wj \in L_M(q'')$. In terms of the encodings, the number $(\mathtt{1}(iwj \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^\ell)_k$ should belong to X_q if and only if there are states q', q'' with $\delta(q', q'') = q$, $(\mathtt{1}(iw \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^{\ell+1})_k \in X_{q'}$ and $(\mathtt{1}(wj \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^\ell)_k \in X_{q''}$.

To this end, two expressions: λ_i and ρ_j for $i, j \in \Sigma_k$ depending on the variables X_q are devised. The purpose of λ_i is to take a number of the form $(\mathtt{1}(wj \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^\ell)_k$ and append the digit i to the left of the encoded string obtaining a number $(\mathtt{1}(iwj \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^\ell)_k$. Similarly, ρ_j starts with a number $(\mathtt{1}(iw \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^{\ell+1})_k$ and appends j to the right of the string, also obtaining $(\mathtt{1}(iwj \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^\ell)_k$; note that one of the zeroes in the tail has to be consumed. This is implemented by adding digits at some specific positions, so that selected digits in the original number (at the left and at the right of the encoding, respectively, whence the letters λ and ρ come from) could be modified in the resulting number, while the rest of the digits remain the same. This is to be done by adding a number of the form $((j-i)\mathtt{0}^\ell)_k$, where ℓ is the position in which digit i is to be replaced by digit $j > i$.

For convenience, the inner subexpressions of λ_i are denoted by $\kappa_{i'}$ and the inner subexpressions of ρ_j are $\pi_{j'}$.

$$\kappa_{i'}(X) = \big(X \cap (\mathtt{1}i'\Sigma_k^*\mathtt{1}\mathtt{0}^*)_k\big) + (\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{2}i'\Sigma_k^*)_k, \quad \text{for all } i' \in \Sigma_k \ ,$$

$$\lambda_i(X) = \bigcup_{i' \in \Sigma_k} \big(\kappa_{i'}(X) + ((k+i-2)\mathtt{0}^*)_k \cap (\mathtt{1}i\Sigma_k^*)_k\big), \quad \text{for } i = 0, 1 \ ,$$

$$\lambda_i(X) = \bigcup_{i' \in \Sigma_k} \big(\kappa_{i'}(X) + (\mathtt{1}(i-2)\mathtt{0}^*)_k \cap (\mathtt{1}i\Sigma_k^*)_k\big), \quad \text{for } i \geq 2 \ ,$$

$$\pi_{j'}(X) = \big(X \cap (\mathtt{1}\Sigma_k^*j'\mathtt{1}\mathtt{0}^*)_k\big) + (\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^*j'\mathtt{2}\mathtt{0}^*)_k, \quad \text{for all } j' \in \Sigma_k \ ,$$

$$\rho_j(X) = \bigcup_{j' \in \Sigma_k} \big(\pi_{j'}(X) + ((k+j-2)\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^*j\mathtt{1}\mathtt{0}^*)_k\big), \quad \text{for } j = 0, 1 \ ,$$

$$\rho_j(X) = \bigcup_{j' \in \Sigma_k} \big(\pi_{j'}(X) + (\mathtt{1}(j-2)\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^*j\mathtt{1}\mathtt{0}^*)_k\big), \quad \text{for } 2 \leq j \leq k-2 \ ,$$

$$\rho_{k-1}(X) = \bigcup_{j' \in \Sigma_k} \big(\pi_{j'}(X) + ((k-3)\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^*(k-1)\mathtt{1}\mathtt{0}^*)_k\big) \ .$$

Define also the following constant sets $R_q \subseteq \mathbb{N}$ for every $q \in Q$:

$$R_q = \{(\mathtt{1}(w \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k \mid w \in \mathtt{0}^*(\Sigma_k \setminus \mathtt{0}) \cup (\Sigma_k \setminus \mathtt{0})\mathtt{0}^*,\ w \in L_M(q)\} \ .$$

Constant R_q represents the numbers in S_q that cannot be obtained by a reference to $S_{q'}$ and $S_{q''}$, since this would involve representing $w \boxminus \mathtt{1}$, where $w \in \mathtt{0}^*$. By Lemma 4.1, the language of positional notations of these numbers is regular, so R_q are allowed constants.

Using the functions λ_i and ρ_j, as well as the constant sets R_q, the resolved system of equations can be succinctly represented in the following form:

$$\begin{cases} X_q = R_q \cup \bigcup\limits_{\substack{q',q'' \,:\, \delta(q',q'')=q \\ i,j\in\Sigma_k}} \lambda_i(X_{q''}) \cap \rho_j(X_{q'}), \quad \text{for all } q \in Q \ , \\ Y = \bigcup_{q\in F} X_q \ . \end{cases}$$

Intuition behind this equations is as follows: the sets R_q contain the starting part of S_q representing elements of $L_M(q)$ of a very simple form: either a non-zero digit followed by zeroes, or a sequence of zeroes ending with one nonzero digit. All other numbers are constructed by simulating the transition of M: a word w belongs to $L_M(q)$ if and only if there exists two states $q', q'' \in Q$ such that $\delta(q', q'') = q$ and letters $i, j \in \Sigma_k$ such that $w = iw'j$ and $iw' \in L_M(q')$ and $w'j \in L_M(q'')$.

The overall goal is to prove the following statement:

Main Claim. *The unique solution of the system in sets of non-negative integers is $X_q = S_q$ (for all $q \in Q$), $Y = L$.*

This is the least solution in natural numbers.

Consider the system for the variables X_q, for all q. By Lemma 2.1, the form of this system ensures the uniqueness of its solution in non-negative integers. The equation for Y is just a union of some of these variables and it cannot yield any extra solutions. It remains to substitute the vector $(\ldots, S_q, \ldots, S)$ into the system and verify that all equations hold true.

The first to be calculated is the value of each λ_i on each S_q, beginning with its subexpression $\kappa_{i'}$.

Claim 4.1. *For each $q \in Q$ and $i' \in \Sigma_k$,*

$$\kappa_{i'}(S_q) = \{(2i'w10^m)_k \mid m \geq 0,\ i'w \notin (k-1)^*,\ i'w \boxplus 1 \in L_M(q)\} \ .$$

Proof. The inner subexpression $X \cap (1i'\Sigma_k^* 10^*)_k$ clearly has value

$$S_q \cap (1i'\Sigma_k^* 10^*)_k = \{(1i'w10^m)_k \mid m \geq 0,\ i'w \notin (k-1)^*,\ i'w \boxplus 1 \in L_M(q)\} \ . \tag{4.1}$$

The evaluation of an entire expression $\kappa_{i'}$ is first done for singletons $X = \{n\}$, and then Lemma 2.2 is applied to obtain its value on $X = S_q$. In view of the above intersection (4.1), it is sufficient to consider numbers of the form

$$n = (1i'w10^m)_k \ .$$

with $w' \in \Sigma_k^*$, $j' \in \Sigma_k$ and $m \geq 0$. Then any number $n' = (10^\ell)_k$ with $\ell \geq 0$ can be added, and it is required that the base-k notation of the sum $n + n'$ has $2i'$ as its first two digits.

- If the number of digits in n and n' is the same, that is, if $|i'w10^m| = \ell$, then the leading 1s in n and n' are in the same position, and the sum is $n + n' = (2i'w10^m)_k \in (2i'\Sigma_k^*)_k$.

- If n' has more digits than n, then $n + n'$ begins with $\mathtt{1}$, and hence $n + n' \notin (\mathtt{2}i'\Sigma_k^*)_k$.
- If n' has fewer digits than n, then the form of the sum depends on whether there is a carry into the first position. If there is no carry, then the sum of these numbers does not begin with $\mathtt{2}$. Otherwise, if it begins with $\mathtt{20}$ due to a carry, then $i' = k - 1$, but at the same time the second digit of $n + n'$ is $\mathtt{0}$ and hence not i'.

These wrong combinations are filtered out by an intersection with $(\mathtt{2}i'\Sigma_k^*)_k$. Altogether the substitution of $X = \{n\}$ yields

$$\big(\{(\mathtt{1}i'w\mathtt{1}\mathtt{0}^m)_k\} + (\mathtt{1}\mathtt{0}^*)_k\big) \cap (\mathtt{2}i'\Sigma_k^*)_k = \{(\mathtt{2}i'w\mathtt{1}\mathtt{0}^m)_k\} \enspace .$$

Since this expression is a superposition of intersection with a constant set and addition of a constant set, by Lemma 2.2, it is distributive: that is, for any $T \subseteq (\mathtt{1}i'\Sigma_k^*\mathtt{1}\mathtt{0}^*)_k$,

$$\begin{aligned} T + (\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{2}i'\Sigma_k^*)_k &= \bigcup_{n \in T} \big(\{n\} + (\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{2}i'\Sigma_k^*)_k\big) \\ &= \{\mathtt{2}i'w\mathtt{1}\mathtt{0}^m \mid \mathtt{1}i'w\mathtt{1}\mathtt{0}^m \in T\} \enspace . \end{aligned}$$

It remains to substitute the value (4.1) of the inner subexpression for T, obtaining

$$\begin{aligned} \kappa_{i'}(S_q) =& \big(S_q \cap (\mathtt{1}i'\Sigma_k^*\mathtt{1}\mathtt{0}^*)_k\big) + (\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{2}i'\Sigma_k^*)_k \qquad (4.2) \\ =& \big(\{(\mathtt{1}i'w\mathtt{1}\mathtt{0}^m)_k \mid m \geq 0,\ i'w \notin (k-1)^*,\ i'w \boxplus \mathtt{1} \in L_M(q)\} \\ & + (\mathtt{1}\mathtt{0}^*)_k\big) \cap (\mathtt{2}i'\Sigma_k^*)_k \\ =& \{(\mathtt{2}i'w\mathtt{1}\mathtt{0}^m)_k \mid m \geq 0,\ i'w \notin (k-1)^*,\ i'w \boxplus \mathtt{1} \in L_M(q)\} \enspace , \end{aligned}$$

which completes the proof of Claim 4.1. □

Now, knowing the value of $\kappa_{i'}(S_q)$, one can evaluate λ_i on S_q.

Claim 4.2. *For each $i \in \Sigma_k$ and $q \in Q$,*

$$\lambda_i(S_q) = \{(\mathtt{1}(iw \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^m)_k \mid w \in L_M(q) \setminus \mathtt{0}^*,\ m \geq 0\} \enspace .$$

Proof. The proof is again by evaluating each subexpression, which is first done for singletons $X = \{n\}$. Addition followed by intersection used in λ_i follows the same principle as in $\kappa_{i'}$. The set being added has a different form depending on i, and, according to the definition of λ_i, there are two cases.

Consider first the case of $i \in \{\mathtt{0}, \mathtt{1}\}$. By Claim 4.1, every number in $\kappa_{i'}(S_q)$ is of the form

$$n = (\mathtt{2}i'w'\mathtt{1}\mathtt{0}^m)_k \enspace .$$

Then a number $n' = ((k + i - 2)0^\ell)_k$ is added to n, and it is required that the result begins with digits $\mathtt{1}i$. Since n has $\mathtt{2}$ as the leading digit, this digit should be modified to obtain a result of such a form.

- If the number of digits in n and n' is the same, that is, $|i'w'10^m| = \ell$, then the digits $\mathtt{2}$ and $(k+i-2)$ are in the same position, and so the result $n + n' = (\mathtt{1}ii'w'\mathtt{10}^m)_k$ is as intended.
- If n' has more digits than n, then the sum $n+n'$ has the leading digit $k+i-2 \in \{k-2, k-1\}$, which is not $\mathtt{1}$, because $k \geq 4$. Hence, $n + n' \notin (\mathtt{1}i\Sigma_k^*)_k$.
- If n' has fewer digits than n, then the leading digit of their sum is $\mathtt{2}$ or $\mathtt{3}$, and again the sum is not in $(\mathtt{1}i\Sigma_k^*)_k$.

After an intersection, again, only one number remains:

$$\big(\{(\mathtt{2}i'w'\mathtt{10}^m)_k\} + ((k+i-2)\mathtt{0}^*)_k\big) \cap (\mathtt{1}i\Sigma_k^*)_k = \{(\mathtt{1}ii'w'\mathtt{10}^m)_k\} \ .$$

As in the previous case, this subexpression is distributive by Lemma 2.2, that is, its value on any set $T \subseteq (\mathtt{2}i'\Sigma_k^*\mathtt{10}^*)_k$ is obtained from its value on singletons as follows:

$$\begin{aligned} T + ((k+i-2)\mathtt{0}^*)_k \cap (\mathtt{1}i\Sigma_k^*)_k &= \bigcup_{n \in T} \big(\{n\} + ((k+i-2)\mathtt{0}^*)_k \cap (\mathtt{1}i\Sigma_k^*)_k\big) \\ &= \{(\mathtt{1}ii'w'\mathtt{10}^m)_k \mid \mathtt{2}i'w'\mathtt{10}^m \in T\} \ . \end{aligned}$$

The value (4.2) of the nested subexpression $\kappa_{i'}(S_q)$ is known from Claim 4.1. Substituting this value for T, one obtains

$$\begin{aligned} \kappa_{i'}(S_q) &+ ((k+i-2)\mathtt{0}^*)_k \cap (\mathtt{1}i\Sigma_k^*)_k \\ &= \{(\mathtt{2}i'w'\mathtt{10}^m)_k \mid m \geq 0,\ i'w' \notin (k-1)^*,\ i'w' \boxplus \mathtt{1} \in L_M(q)\} \\ &\quad + ((k+i-2)\mathtt{0}^*)_k \cap (\mathtt{1}i\Sigma_k^*)_k \\ &= \{(\mathtt{1}ii'w'\mathtt{10}^m)_k \mid m \geq 0, i'w' \notin (k-1)^*, i'w' \boxplus \mathtt{1} \in L_M(q)\} \ . \end{aligned} \tag{4.3}$$

Now consider the case of $i \geq 2$, where a number $n' \in (\mathtt{1}(i-2)\mathtt{0}^*)_k$ is added to n.

- If n' has exactly one digit more than n, that is, the digits $\mathtt{2}$ and $i-2$ are in the same position, then the sum equals $n + n' = (\mathtt{1}iw\mathtt{10}^m)_k$, as intended.
- If n' has exactly as many digits as n, then there are two subcases:
  - if $i'+i-2 < k$, then the result is $n+n' = (\mathtt{3}(i'+i-2)w\mathtt{10}^m)_k \notin (\mathtt{1}i\Sigma_k^*)_k$;
  - if $i'+i-2 \geq k$ the sum is $n+n' = (\mathtt{4}(i'+i-2-k)w\mathtt{10}^m)_k \notin (\mathtt{1}i\Sigma_k^*)_k$.
- If the number of digits in n' is greater than the number of digits in n plus one, then the result is $n + n' = (\mathtt{1}(i-2)\mathtt{0}^t\mathtt{2}i'w'\mathtt{10}^m)_k$ for some $t \geq 0$, and hence $n + n' \notin (\mathtt{1}i\Sigma_k^*)_k$.

- Finally, if there are fewer digits in n' than in n, then the leading digit in $n + n'$ is $\mathtt{2}$ or $\mathtt{3}$.

Thus, all unintended results are filtered by intersection with $(\mathtt{1}i\Sigma_k^*)_k$, and the result is the same as in the previous case:

$$\big(\{(\mathtt{2}i'w'\mathtt{10}^m)_k\} + (\mathtt{1}(i-2)\mathtt{0}^*)_k\big) \cap (\mathtt{1}i\Sigma_k^*)_k = \{(\mathtt{1}ii'w'\mathtt{10}^m)_k\} \ .$$

Then, using Lemma 2.2 and the value (4.2) of the nested subexpression $\kappa_{i'}(S_q)$, the following is obtained for each $i \geq 2$:

$$\begin{aligned}
\kappa_{i'}(S_q) + &(\mathtt{1}(i-2)\mathtt{0}^*)_k \cap (\mathtt{1}i\Sigma_k^*)_k = \\
=&\{(\mathtt{2}i'w'\mathtt{10}^m)_k \mid m \geq 0,\ i'w' \notin (k-1)^*,\ i'w' \boxplus \mathtt{1} \in L_M(q)\} \\
&+ (\mathtt{1}(i-2)\mathtt{0}^*)_k \cap (\mathtt{1}i\Sigma_k^*)_k \\
=&\{(\mathtt{1}ii'w'\mathtt{10}^m)_k \mid m \geq 0, i'w' \notin (k-1)^*, i'w' \boxplus \mathtt{1} \in L_M(q)\} \ , \qquad (4.4)
\end{aligned}$$

and the result is the same as in the case of $i \in \{\mathtt{0}, \mathtt{1}\}$.

As the whole expression λ_i is defined as the union of subexpressions evaluated above, the set $\lambda_i(S_q)$ equals the union of the values (4.3), (4.4) for all $i' \in \Sigma_k$. Taking a union over i', one obtains:

$$\begin{aligned}
\lambda_i(S_q) &= \bigcup_{i'} \{(\mathtt{1}ii'w\mathtt{10}^m)_k \mid m \geq 0, i'w \notin (k-1)^*, i'w \boxplus \mathtt{1} \in L_M(q)\} \\
&= \{(\mathtt{1}iw\mathtt{10}^m)_k \mid m \geq 0, w \notin (k-1)^*, w \boxplus \mathtt{1} \in L_M(q)\} \\
&= \{(\mathtt{1}iw\mathtt{10}^m)_k \mid m \geq 0, w \in (L_M(q) \setminus \mathtt{0}^*) \boxminus \mathtt{1}\} \\
&= (\mathtt{1}(i(L_M(q) \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{10}^*)_k \ , \qquad (4.5)
\end{aligned}$$

and the claim is proved. □

The expressions ρ_j operate in a way similar to λ_i. While λ_i modifies the most significant digits of numbers $(\mathtt{1}w\mathtt{10}^\ell)_k$, ρ modifies the digits around the last non-zero digit. This is also done by addition in two stages, but instead of intersecting the result with sets of the form $(x\Sigma_k^*\mathtt{10}^*)_k$, where $x \in \Sigma_k^+$ are the intended digits, this time one has to use intersection with $(\mathtt{1}\Sigma^*x\mathtt{0}^*)_k$. The corresponding correctness statement for ρ_j is established using generally the same argument as the previous Claim 4.2. First its inner subexpression $\pi_{j'}$ is evaluated on S_q.

Claim 4.3. *For all $q \in Q$ and $j' \in \Sigma_k$,*

$$\pi_{j'}(S_q) = \{(\mathtt{1}w'j'\mathtt{20}^m)_k \mid m \geq 0, w'j' \notin (k-1)^*, w'j' \boxplus \mathtt{1} \in L_M(q)\} \ .$$

Proof. As in the proof of Claim 4.1, every subexpression of $\pi_{j'}(X)$ will be evaluated on $X = S_q$. The inner subexpression of $\pi_{j'}(S_q)$ is

$$S_q \cap (\mathtt{1}\Sigma_k^*j'\mathtt{10}^*)_k = \{(\mathtt{1}w'j'\mathtt{10}^m)_k \mid m \geq 0, w'j' \notin (k-1)^*, w'j' \boxplus \mathtt{1} \in L_M(q)\} \ . \qquad (4.6)$$

Consider the next subexpression in $\pi_{j'}$, which is $\big(X \cap (\mathtt{1}\Sigma_k^* j'\mathtt{1}\mathtt{0}^*)_k\big) + (\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^* j'\mathtt{2}\mathtt{0}^*)_k$. The first task is to evaluate it for a singleton $X = \{n\}$, where the number is of the form

$$n = (\mathtt{1}w'j'\mathtt{1}\mathtt{0}^m)_k \ .$$

In this subexpression, any number $n' = (\mathtt{1}\mathtt{0}^\ell)_k$ with $\ell \geq 0$ can be added to n, and the results are restricted to be in $(\mathtt{1}\Sigma_k^* j'\mathtt{2}\mathtt{0}^*)_k$, that is, $n + n'$ has $j'\mathtt{2}$ as its last non-zero digits.

- If $m = \ell$, that is, the last digit $\mathtt{1}$ in n is in the same position as the leading digit $\mathtt{1}$ in n', then $n + n' = (\mathtt{1}iw'j'\mathtt{2}\mathtt{0}^m)_k$.
- If these digits are not aligned, then the last non-zero digit in $n + n'$ is $\mathtt{1}$, and hence $n + n' \notin (\mathtt{1}\Sigma_k^* j'\mathtt{2}\mathtt{0}^*)_k$.

Thus it has been shown that

$$\{(\mathtt{1}w'j'\mathtt{1}\mathtt{0}^m)_k\} + \mathtt{1}\mathtt{0}^* \cap (\mathtt{1}\Sigma_k^* j'\mathtt{2}\mathtt{0}^*)_k = \{(\mathtt{1}w'j'\mathtt{2}\mathtt{0}^m)_k\} \ .$$

As in the previous proofs, the considered subexpression is distributive by Lemma 2.2, that is, for any $T \subseteq (\mathtt{1}\Sigma_k^* j'\mathtt{1}\mathtt{0}^*)_k$,

$$\begin{aligned} T + (\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^* j'\mathtt{2}\mathtt{0}^*)_k &= \bigcup_{n \in T} \big(\{n\} + (\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^* j'\mathtt{2}\mathtt{0}^*)_k\big) \\ &= \{\mathtt{1}w'j'\mathtt{2}\mathtt{0}^m \mid \mathtt{1}w'j'\mathtt{1}\mathtt{0}^m \in T\} \ . \end{aligned}$$

Now let T be the value (4.6) of the inner subexpression, which gives

$$\begin{aligned} \pi_{j'}(S_q) &= \big(S_q \cap (\mathtt{1}\Sigma_k^* j'\mathtt{1}\mathtt{0}^*)_k\big) + (\mathtt{1}\mathtt{0}^*)_k \ \cap \ (\mathtt{1}\Sigma_k^* j'\mathtt{2}\mathtt{0}^*)_k \\ &= \{(\mathtt{1}w'j'\mathtt{2}\mathtt{0}^m)_k \mid m \geq 0, w'j' \notin (k-1)^*, w'j' \boxplus \mathtt{1} \in L_M(q)\} \ , \end{aligned} \quad (4.7)$$

and thus proves Claim 4.3. □

Claim 4.4. *For each $j \in \Sigma_k$ and $q \in Q$,*

$$\rho_j(S_q) = \{(\mathtt{1}(w(j+1 \mod k) \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^m)_k \mid w \in L_M(q) \setminus \mathtt{0}^*, \ m \geq 0\} \ .$$

Proof. The evaluation of subexpressions of $\rho_j(X)$ on $X = S_q$ splits into three cases according to the digit j. In each case the innermost subexpression is $\pi_{j'}(X)$, and its value $\pi_{j'}(S_q)$ contains only numbers of the form

$$n = (\mathtt{1}w'j'\mathtt{2}\mathtt{0}^m)_k \ ,$$

with $w'j' \notin (k-1)^*$.

The first case is $j \in \{\mathtt{0}, \mathtt{1}\}$, where ρ_j has a subexpression

$$\pi_{j'}(X) + ((k+j-2)\mathtt{1}\mathtt{0}^*)_k \ \cap \ (\mathtt{1}\Sigma_k^* j\mathtt{1}\mathtt{0}^*)_k \ .$$

Here any number $n' = ((k+j-2)\mathtt{1}\mathtt{0}^\ell)_k$ with $\ell \geq 0$ is added to n, and it is required that the sum $n + n'$ has $j\mathtt{1}$ as its last non-zero digits. There are several subcases depending on the number of ending zeroes in n and in n', which is m and ℓ, respectively:

- If $\ell = m - 1$, that is, the last non-zero digit $\mathtt{2}$ in n is aligned with the leading digit $(k + j - 2)$ of n', and they sum up to j, with a carry to the next digit, j'. Accordingly, the sum of two numbers is $n + n' = (\mathtt{1}(w'j' \boxplus \mathtt{1})j\mathtt{1}\mathtt{0}^{m-1})_k$, as it is intended to be. Note that since $w'j' \notin (k-1)^*$, the string $w'j' \boxplus \mathtt{1}$ is well-defined.

- If $\ell < m - 1$, then the last two non-zero digits of $n + n'$ are $(k + j - 2)\mathtt{1}$. As $k \geq 5$ and $k + j - 2 \neq j$, the last non-zero digits of $n + n'$ are different from $j\mathtt{1}$, and therefore $n + n' \notin (\mathtt{1}\Sigma_k^* j\mathtt{1}\mathtt{0}^*)_k$.

- If $\ell = m$, then the last non-zero digit of $n + n'$ is $\mathtt{3}$ and not $\mathtt{1}$, and the sum is again not in $(\mathtt{1}\Sigma_k^* j\mathtt{1}\mathtt{0}^*)_k$.

- In case of $\ell > m$, the last digit is $\mathtt{2}$ inherited from n, and again the sum is not of the required form.

It follows that all unintended sums are filtered out by intersection, and the result is

$$(\mathtt{1}w'j'\mathtt{2}\mathtt{0}^m)_k + ((k + j - 2)\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^* j\mathtt{1}\mathtt{0}^*)_k = \{(\mathtt{1}(w'j' \boxplus \mathtt{1})j\mathtt{1}\mathtt{0}^{m-1})_k\} ,$$

which, by Lemma 2.2, extends to any set $T \subseteq (\mathtt{1}\Sigma_k^* j'\mathtt{2}\mathtt{0}^*)_k \setminus (\mathtt{1}(k-1)^*\mathtt{2}\mathtt{0}^*)_k$ as follows:

$$T + ((k+j-2)\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^* j\mathtt{1}\mathtt{0}^*)_k = \{(\mathtt{1}(w'j' \boxplus \mathtt{1})j\mathtt{1}\mathtt{0}^{m-1})_k \mid \mathtt{1}w'j'\mathtt{2}\mathtt{0}^m \in T\} .$$

Hence, after the addition and intersection in the subexpression for $j \in \{\mathtt{0}, \mathtt{1}\}$, the intermediate result is

$$\begin{aligned}&\pi_{j'}(S_q) + ((k + j - 2)\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^* j\mathtt{1}\mathtt{0}^*)_k = \\ &= \{(\mathtt{1}(w'j' \boxplus \mathtt{1})j\mathtt{1}\mathtt{0}^{m-1})_k \mid m \geq 1, w'j' \notin (k-1)^*, w'j' \boxplus \mathtt{1} \in L_M(q)\} .\end{aligned} \tag{4.8}$$

Next, consider the case of $j \in \{\mathtt{2}, \ldots, k-\mathtt{2}\}$, when ρ_j has inner subexpression $\pi_{j'}(X) + (\mathtt{1}(j-\mathtt{2})\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^* j\mathtt{1}\mathtt{0}^*)_k$. Here a number $n' = (\mathtt{1}(j-\mathtt{2})\mathtt{1}\mathtt{0}^\ell)_k$ is added to n, and the subsequent intersection requires that $n + n'$ has $j\mathtt{1}$ as its last non-zero digits.

- Again, for $\ell = m - 1$ the sum is $(\mathtt{1}(w'j' \boxplus \mathtt{1})j\mathtt{1}\mathtt{0}^{m-1})_k$, which is of the required form.

- If $\ell < m - 1$, then the last two non-zero digits of $n + n'$ are $(j - 2)\mathtt{1}$. Since $j - 2 \neq j$, the sum is not in $(\mathtt{1}\Sigma_k^* j\mathtt{1}\mathtt{0}^*)_k$.

- If $\ell = m$, then the last non-zero digit of $n + n'$ is $\mathtt{3}$.

- If $\ell > m$, then the last non-zero digit is $\mathtt{2}$ (coming from n).

Therefore, as in the previous case, the only result that passes through the intersection is $(\mathtt{1}(w'j' \boxplus \mathtt{1})j\mathtt{1}\mathtt{0}^{m-1})_k$, and this subexpression evaluates to

$$(\mathtt{1}w'j'\mathtt{2}\mathtt{0}^m)_k + (\mathtt{1}(j-2)\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^* j\mathtt{1}\mathtt{0}^*)_k = \{(\mathtt{1}(w'j' \boxplus \mathtt{1})j\mathtt{1}\mathtt{0}^{m-1})_k\} \ .$$

Hence the value of the inner subexpression for $2 \leq j \leq k-2$ is

$$\begin{aligned} &\pi_{j'}(S_q) + ((j-2)\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^* j\mathtt{1}\mathtt{0}^*)_k = \\ &= \{(\mathtt{1}(w'j' \boxplus \mathtt{1})j\mathtt{1}\mathtt{0}^{m-1})_k \mid m \geq 1, w'j' \notin (k-1)^*, w'j' \boxplus \mathtt{1} \in L_M(q)\} \ , \end{aligned} \tag{4.9}$$

which is the same as for $j \in \{\mathtt{0}, \mathtt{1}\}$.

To conclude this case, as well as the previous one, it has been proved that for all $j \neq k-1$, regardless of the value of j', ρ_j transforms $(\mathtt{1}w'j'\mathtt{1}\mathtt{0}^m)_k$ into $(\mathtt{1}(w'j' \boxplus \mathtt{1})j\mathtt{0}^{m-1})_k$, or equivalently, $(\mathtt{1}w'j'j\mathtt{1}\mathtt{0}^m)_k$ is obtained from $(\mathtt{1}(w'j' \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^{m+1})_k$.

Finally, consider the case of $j = k-1$. Here any number $n' = ((k-3)\mathtt{1}\mathtt{0}^\ell)_k$ with $\ell \geq 0$ can be added to $n = (\mathtt{1}w'j'\mathtt{2}\mathtt{0}^m)_k$, and their sum $n + n'$ is required to have $(k-1)\mathtt{1}$ as its last non-zero digits.

- If $\ell = m-1$, then the sum is $(\mathtt{1}w(k-1)\mathtt{1}\mathtt{0}^{m-1})_k$, which passes the intersection with $(\mathtt{1}\Sigma_k^*(k-1)\mathtt{1}\mathtt{0}^*)_k$.
- If $\ell < m-1$, then the last two digits of $n + n'$ are $(k-3)\mathtt{1}$, and since $k-3 \neq k-1$, this number is not of the required form.
- If $\ell = m$, then the last digit of $n + n'$ is $\mathtt{3}$.
- If $\ell > m$, then the last digit is $\mathtt{2}$ from n.

As all wrong values have been filtered out, the value of the expression is again a singleton:

$$(\mathtt{1}w'j'\mathtt{2}\mathtt{0}^m)_k + ((k-3)\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^*(k-1)\mathtt{1}\mathtt{0}^*)_k = \{(\mathtt{1}w'j'(k-1)\mathtt{1}\mathtt{0}^{m-1})_k\} \ .$$

Thus the subexpression corresponding to $j = k-1$ has the following value:

$$\begin{aligned} &\pi_{j'}(S_q) + ((k-3)\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^*(k-1)\mathtt{1}\mathtt{0}^*)_k = \\ &= \{(\mathtt{1}w'j'(k-1)\mathtt{1}\mathtt{0}^{m-1})_k \mid m \geq 1, w'j' \notin (k-1)^*, w'j' \boxplus \mathtt{1} \in L_M(q)\} \ . \end{aligned} \tag{4.10}$$

Now the value of ρ_{k-1} on S_q is obtained as the union of (4.10) over j':

$$\begin{aligned} &\rho_{k-1}(S_q) \\ &= \bigcup_{j'} \{(\mathtt{1}w'j'(k-1)\mathtt{1}\mathtt{0}^{m-1})_k \mid m \geq 1, w'j' \notin (k-1)^*, w'j' \boxplus \mathtt{1} \in L_M(q)\} \\ &= \{(\mathtt{1}w(k-1)\mathtt{1}\mathtt{0}^{m-1})_k \mid m \geq 1, w \notin (k-1)^*, w \boxplus \mathtt{1} \in L_M(q)\} \\ &= \{(\mathtt{1}(w \boxminus \mathtt{1})(k-1)\mathtt{1}\mathtt{0}^m)_k \mid m \geq 0, w \notin \mathtt{0}^*, w \in L_M(q)\} \ . \end{aligned}$$

Note that $(w \boxminus 1)(k-1) = w0 \boxminus 1$, and so the latter set can be rewritten as

$$\rho_{k-1}(S_q) = \{(1(w0 \boxminus 1)10^m)_k \mid m \geq 0, w \in L_M(q) \setminus 0^*\} \ ,$$

which is of the form stated in the claim.

Similarly, for each $j \neq k-1$, the union of (4.8) and (4.9) over j' gives the value of ρ_j:

$$\begin{aligned}
&\rho_j(S_q) \\
&\quad = \bigcup_{j'} \{(1(w'j' \boxplus 1)j10^{m-1})_k \mid m \geq 1, w'j' \notin (k-1)^*, w'j' \boxplus 1 \in L_M(q)\} \\
&\quad = \{(1(w \boxplus 1)j10^{m-1})_k \mid m \geq 1, w \notin (k-1)^*, w \boxplus 1 \in L_M(q)\} \\
&\quad = \{(1wj10^m)_k \mid m \geq 0, w \notin 0^*, w \in L_M(q)\} \ .
\end{aligned}$$

Here, since $j \neq k-1$, the string wj is equal to $w(j+1) \boxminus 1$, and hence the result equals

$$\{(1(w(j+1) \boxminus 1)10^m)_k \mid m \geq 0, w \in L_M(q) \setminus 0^*\} \ ,$$

which completes the proof of Claim 4.4. ☐

Now the intended solution can be substituted into the system.

Proof of the Main Claim. For every $i, j \in \Sigma_k$ and $q', q'' \in Q$, consider the expression $\lambda_i(S_{q''}) \cap \rho_j(S_{q'})$. The values of $\lambda_i(S_{q''})$ and $\rho_j(S_{q'})$ are known from Claim 4.2 and Claim 4.4, and the form of the expressions can be unified as follows (all sums $j+1$ are modulo k):

$$\begin{aligned}
\lambda_i(S_{q''}) &= \{(1(iw(j+1) \boxminus 1)10^m)_k \mid w(j+1) \in L_M(q'') \setminus 0^*,\ m \geq 0\} \ , \\
\rho_j(S_{q'}) &= \{(1(iw(j+1) \boxminus 1)10^m)_k \mid iw \in L_M(q') \setminus 0^*,\ m \geq 0\} \ .
\end{aligned}$$

The intersection of these sets therefore is

$$\begin{aligned}
\lambda_i(S_{q''}) \cap \rho_j(S_{q'}) = \big\{&(1(iw(j+1) \boxminus 1)10^m)_k \mid \\
&iw \in L_M(q'),\ w(j+1) \in L_M(q''),\ iw \notin 0^*,\ w(j+1) \notin 0^*,\ m \geq 0\big\} \ .
\end{aligned}$$

Fix $q \in Q$. According to the definition of a trellis automaton, $iw(j+1) \in L_M(q)$ if and only if $iw \in L_M(q')$ and $w(j+1) \in L_M(q'')$ for some q' and q'' such that $\delta(q', q'') = q$. Then the union of $\lambda_i(S_{q''}) \cap \rho_j(S_{q'})$ over all such states q' and q'' equals

$$\bigcup_{q', q'' : \delta(q', q'') = q} \lambda_i(S_{q''}) \cap \rho_j(S_{q'}) =$$
$$= \{(1(iw(j{+}1) \boxminus 1)10^m)_k | iw(j{+}1) \in L_M(q), iw \notin 0^*, w(j{+}1) \notin 0^*, m \geq 0\} \ .$$

Taking another union over all digits i and j, the following expression is obtained:

$$\bigcup_{i,j\in\Sigma_k} \bigcup_{q',q'':\delta(q',q'')=q} \lambda_i(S_{q''}) \cap \rho_j(S_{q'}) = \\ = \{(\mathtt{1}(w \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^m)_k \mid w \in L_M(q),\ w \notin \Sigma_k\mathtt{0}^* \cup \mathtt{0}^*\Sigma_k,\ m \geq 0\} \ .$$

The cases of $w \in (\Sigma_k\mathtt{0}^* \cup \mathtt{0}^*\Sigma_k) \setminus \mathtt{0}^*$ are not reflected in the above expression. However, they are included in R_q, and therefore

$$\begin{aligned} R_q \cup \bigcup_{i,j\in\Sigma_k} \bigcup_{q',q'':\delta(q',q'')=q} & \lambda_i(S_{q''}) \cap \rho_j(S_{q'}) \\ &= \{(\mathtt{1}(w \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^m)_k \mid w \in L_M(q),\ w \notin \mathtt{0}^*,\ m \geq 0\} \\ &= \{(\mathtt{1}w\mathtt{1}\mathtt{0}^m)_k \mid w \boxplus 1 \in L_M(q),\ w \notin (k-1)^*,\ m \geq 0\} \\ &= S_q \ , \end{aligned}$$

that is, the equation for X_q turns into an equality.

As the system is proper, the least solution in sets natural numbers is also the unique solution in sets of positive numbers. This concludes the proof of the Main Claim and hence of the entire Lemma 4.3. $\square$

$\square$

The system constructed in Lemma 4.3 represents the set $(\mathtt{1}(L(M) \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k$ for every trellis automaton M. Every number in this set represents an encoding of a string $w \in L(M)$ modified by decrementing w as well as by introducing a pair of sentinel digits $\mathtt{1}$ and a tail of zeroes. This next step towards representing the set $(L(M))_k$ for every M with $L(M) \cap \mathtt{0}\Sigma_k^* = \varnothing$ is the following lemma, in which a string $w \in \Sigma_k$ is encoded as a number $(\mathtt{1}w)_k$, that is, using only one sentinel digit, no zeroes and no decrementation.

Lemma 4.4. *For every $k \geq 5$ and for every trellis automaton M over Σ_k there exists and can be effectively constructed a resolved system of equations using constants with a regular base-k notation, such that one of the components of its least solution is $(\mathtt{1} \cdot L(M))_k$.*

This is the unique solution in sets of positive natural numbers.

Proof. For every $j \in \Sigma_k$ and $q \in Q$, consider the language

$$L_{j,q} = L_M(q) \cdot j^{-1} \setminus \mathtt{0}^* \ .$$

By the closure properties of trellis automata, this language is recognised by a trellis automaton $M_{j,q}$. Then, by Lemma 4.3, there exists a resolved system of equations, such that one of its variables, $Y_{j,q}$, has value

$$Y_{j,q} = \{(\mathtt{1}w\mathtt{1}\mathtt{0}^\ell)_k \mid \ell \geq 0, w \notin (k-1)^*, w \boxplus \mathtt{1} \in L_{j,q}\} \ .$$

in the least solution.

Let us combine these equations for all j into a single system, adding a new equation

$$Z_q = (\mathtt{1}L_M(q) \cap \mathtt{10}^*\Sigma_k)_k \cup \bigcup_{j=0}^{k-1} (Y_{j,q} \cap (\mathtt{1}\Sigma_k^*\mathtt{1})_k) + (\mathtt{1}j \boxminus \mathtt{1})_k \ .$$

The constant set $(\mathtt{1}L_M(q) \cap \mathtt{10}^*\Sigma_k)_k$ has a regular base-k notation by Lemma 4.1. Since the equation for Z_q does not have a self-reference, its value in the least solution can be calculated by substituting the values of $Y_{j,q}$ in the least solution into the equation for Z_q.

First, the inner intersection with $(\mathtt{1}\Sigma_k^*\mathtt{1})_k$ filters out the elements of $Y_{j,q}$ with one or more zeroes in the end:

$$\begin{aligned}\{(\mathtt{1}w\mathtt{10}^\ell)_k \mid \ell \geq 0,\ w \notin (k-1)^*,\ w \boxplus \mathtt{1} \in L_{j,q}\} \cap (\mathtt{1}\Sigma_k^*\mathtt{1})_k = \\ = \{(\mathtt{1}w\mathtt{1})_k \mid w \notin (k-1)^*,\ w \boxplus \mathtt{1} \in L_{j,q}\} \ .\end{aligned}$$

The subsequent addition of a number $(\mathtt{1}j \boxminus \mathtt{1})_k = k + j - 1$ to each number $(\mathtt{1}w\mathtt{1})_k$ changes the lowest digit from $\mathtt{1}$ to j and produces a carry to the second digit, thus changing w to $w \boxplus \mathtt{1}$. Hence this subexpression has the following value:

$$\begin{aligned}
&\{(\mathtt{1}w\mathtt{1})_k \mid w \notin (k-1)^*,\ w \boxplus \mathtt{1} \in L_{j,q}\} + (\mathtt{1}j \boxminus \mathtt{1})_k \\
&= \{(\mathtt{1}w\mathtt{1})_k + (\mathtt{1}j \boxminus \mathtt{1})_k \mid w \notin (k-1)^*,\ w \boxplus \mathtt{1} \in L_{j,q}\} \\
&= \{(\mathtt{1}(w \boxplus 1)j)_k \mid w \notin (k-1)^*, w \boxplus \mathtt{1} \in L_{j,q}\} \\
&= (\mathtt{1}(L_{j,q} \setminus \mathtt{0}^*)j)_k \\
&= (\mathtt{1}(L_M(q) \cdot j^{-1} \setminus \mathtt{0}^*)j)_k \\
&= (\mathtt{1}(L_M(q) \cdot j^{-1})j)_k \setminus (\mathtt{10}^*\Sigma_k)_k \\
&= (\mathtt{1}(L_M(q) \cap \Sigma_k^* j))_k \setminus (\mathtt{10}^*\Sigma_k)_k \ .
\end{aligned} \tag{4.11}$$

It remains to substitute these values of subexpressions into the full expression for Z, obtaining

$$\begin{aligned}
&(\mathtt{1}L_M(q) \cap \mathtt{10}^*\Sigma_k)_k \cup \bigcup_{j=0}^{k-1} \Big((\mathtt{1}(L_M(q) \cap \Sigma_k^* j))_k \setminus (\mathtt{10}^*\Sigma_k)_k\Big) \\
&= (\mathtt{1}L_M(q) \cap \mathtt{10}^*\Sigma_k)_k \cup \Big((\mathtt{1}L_M(q))_k \setminus (\mathtt{10}^*\Sigma_k)_k\Big) \\
&= (\mathtt{1}L_M(q))_k \ ,
\end{aligned} \tag{4.12}$$

which is accordingly the value of Z_q in the least solution.

Now the equation

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

has the least solution $(\mathtt{1}L(M))_k$.

Since the system is proper, this is also the unique solution in positive numbers. □

The last major step of the argument is eliminating the leading digit $\mathtt{1}$ in the representation given by Lemma 4.4.

Lemma 4.5. *For every $k \geq 5$ and for every trellis automaton M over Σ_k with $L(M) \cap \mathtt{0}\Sigma_k^* = \varnothing$ there exists and can be effectively constructed a system of resolved equations using constants with a regular base-k notation, such that one of the components of its least solution is $(L(M))_k$.*

This solution is also the unique solution in positive numbers.

Proof. For every $i \in \Sigma_k \setminus \{\mathtt{0}\}$ and for every $q \in Q$, the language $i^{-1}L_M(q)$ is recognised by a certain trellis automaton. Then, by Lemma 4.4, there is a resolved system of equations over sets of numbers, such that one of its variables, $Z_{i,q}$, represents the set $(\mathtt{1}(i^{-1}L_M(q)))_k$.

These systems are combined into one, and a new variable T_q is added, along with the equation

$$T_q = (L_M(q) \cap (\Sigma_k \setminus \{\mathtt{0}\}))_k \cup Z_{1,q} \cup \bigcup_{i \in \Sigma_k \setminus \{\mathtt{0},\mathtt{1}\}} \tau_i(Z_{i,q}), \quad \text{where} \tag{4.13}$$

$$\tau_i(X) = \bigcup_{i' \in \Sigma_k} \Big((X \cap (\mathtt{1}i'\Sigma_k^*)_k) + ((i-1)\mathtt{0}^*)_k \cap (ii'\Sigma_k^*)_k, \Big) \quad \text{for } i \neq \mathtt{0},\mathtt{1} \ . \tag{4.14}$$

The purpose of the subexpression τ_i is to convert a number $(\mathtt{1}w)_k$ with $w \in \Sigma_k^+$ to a number $(iw)_k$. Then the right-hand side of the equation for T_q should evaluate to $T_q = (L_M(q) \setminus \mathtt{0}\Sigma_k^*)_k$ under the substitution $Z_{i,q} = (\mathtt{1}(i^{-1}L_M(q)))_k$.

The proof starts with evaluating τ_i on each $Z_{i,q}$:

Claim 4.5. *For every $i \in \Sigma_k \setminus \{\mathtt{0},\mathtt{1}\}$ and $q \in Q$,*

$$\tau_i(Z_{i,q}) = (L_M(q) \cap i\Sigma_k^+)_k \ .$$

Proof. For every $i' \in \Sigma_k$, consider the component of the union for i' in τ_i. Clearly, the innermost subexpression has the following value:

$$\begin{aligned} Z_{i,q} \cap (\mathtt{1}i'\Sigma_k^*)_k &= \{(\mathtt{1}w)_k \mid iw \in L_M(q)\} \cap (\mathtt{1}i'\Sigma_k^*)_k \\ &= \{(\mathtt{1}i'w)_k \mid ii'w \in L_M(q)\} \ . \end{aligned} \tag{4.15}$$

The next subexpression involves an addition of $((i-1)\mathtt{0}^*)_k$ followed by intersection with $(ii'\Sigma_k)_k$. This subexpression is first evaluated on a singleton $\{n\}$. The nested intersection with $(\mathtt{1}i'\Sigma_k^*)_k$ leaves only numbers of the form

$$n = (\mathtt{1}i'w)_k \ ,$$

with $w \in \Sigma_k^*$. Afterwards, any number $n' = ((i-1)\mathtt{0}^\ell)_k$ with $\ell \geq 0$ can be added to n, and it is required that the sum $n + n'$ starts with two digits ii'.

- If the number of digits in n and n' is the same, that is, if $|i'w| = \ell$, then the sum is $n + n' = (ii'w)_k$, as intended.

- If n' has more digits than n, then the sum $n + n'$ has the leading digit $i - 1$, and hence it is not in $(ii'\Sigma_k^*)_k$.

- Suppose n' has exactly one digit fewer than n, that is, $|w| = \ell$. Then the second digit i' in n is aligned with the leading digit $i-1$ of n', and the second digit of the sum $n + n'$ equals $i' + (i-1)$ modulo k. Since $i' < i' + i - 1 < i' + k$, it follows that this second digit cannot be i', and therefore $n + n'$ is not in $(ii'\Sigma_k^*)_k$.

- If the number of digits in n' is less by more than one than the number of digits in n, there are two subcases:
  - If the addition of n' to n results in a carry to the second digit, then the second digit of the result is $i' + 1 \pmod{k}$, hence it is different from i'.
  - If there is no carry, then the leading digit of the sum is $\mathtt{1} \neq i$. In both cases $n + n' \notin (ii'\Sigma_k^*)_k$.

This concludes the case study: all wrong combinations are excluded by an intersection, and therefore

$$\{(\mathtt{1}i'w)_k\} + ((i-1)\mathtt{0}^*)_k \,\cap\, (ii'\Sigma_k^*)_k = \{(ii'w)_k\} \ .$$

Hence,

$$\begin{aligned} \big(Z_{i,q} \cap (\mathtt{1}i'\Sigma_k^*)_k\big) + ((i-1)\mathtt{0}^*)_k \cap (ii'\Sigma_k^*)_k &= \{(ii'w)_k \mid (\mathtt{1}i'w)_k \in Z_{i,q}\} \\ &= \{(ii'w)_k \mid ii'w \in L_M(q)\} \\ &= (L_M(q) \cap ii'\Sigma^*)_k \ . \qquad (4.16) \end{aligned}$$

Taking the union over i', one obtains

$$\tau_i(Z_{i,q}) = (L_M(q) \cap i\Sigma^+)_k \ .$$

which completes the proof of the claim. □

Getting back to the proof of Lemma 4.5, now the right-hand side of the equation for each T_q can be evaluated on $Z_{i,q} = (\mathtt{1}(i^{-1}L_M(q)))_k$. This is a union of k expressions, the first of them representing a finite set, the second being the variable $Z_{\mathtt{1},q}$, and the rest are the subexpressions $\tau_i(Z_{i,q})$ with

$i \in \Sigma_k \setminus \{\mathtt{0}, \mathtt{1}\}$, evaluated in Claim 4.5. Altogether, the value of T_q in the least solution of the constructed system is the union of the following sets:

$$(L_M(q) \cap (\Sigma_k \setminus \{\mathtt{0}\}))_k \cup \underbrace{(\mathtt{1}(\mathtt{1}^{-1} L_M(q)))_k}_{(L_M(q) \cap \mathtt{1}\Sigma_k^+)_k} \cup \bigcup_{i=2}^{k-1} (L_M(q) \cap i\Sigma_k^+)_k$$
$$= (L_M(q) \setminus \mathtt{0}\Sigma_k^*)_k \ .$$

Since $L(M) \cap \mathtt{0}\Sigma_k^* = \varnothing$ by assumption, the least solution of the equation

$$T = \bigcup_{q \in F} T_q$$

is $(L(M))_k$.

As the system is proper, this is also the unique solution in the sets of positive numbers. ☐

Now the proof of the theorem can be easily inferred from Lemma 4.5.

Proof of Theorem 4.2. First assume $k \geq 5$. Then Lemma 4.5 gives a system of equations over sets of numbers with the desired least solution. This system uses constants with a regular notation. By Theorem 3.3, each of these constants can be expressed by a separate system of equations using singleton constants. The resulting system satisfies the statement of Theorem 4.2.

It remains to consider the cases of $k = 2, 3, 4$. Define the language L' of base-k^3 notations of numbers whose base-k notation is in $L(M)$. Then, by Lemma 4.2, L' is generated by another trellis automaton M'. Applying the above argument to M', a system of equations specifying the given set of numbers is obtained. This completes the proof for this remaining case. ☐

Finally, the system of equations constructed in Theorem 4.2 can be represented as a conjunctive grammar over a unary alphabet, which yields the following general result on the expressive power of these grammars:

Corollary 4.1. *Let $k \geq 2$. For every trellis automaton M over Σ_k, with $L(M) \cap \mathtt{0}\Sigma_k^* = \varnothing$, there exists and can be effectively constructed a conjunctive grammar over the alphabet $\{a\}$ that generates the language $\{a^n \mid n \in (L(M))_k\}$.*

Part II

Unresolved equations over sets of natural numbers

As already noted in Chapter 1, systems of equations over sets of numbers can be viewed as systems of language equations over a unary alphabet. Thus the upper bound of Theorem 1.1 can be applied to systems of equations as they are, i.e., each unique (least, greatest) solution of such system is recursive (recursively enumerable, co-recursively enumerable, respectively). It may seem surprising, but a matching lower bound is shown in the following chapters.

It should be noted that the original proof of Theorem 1.1 was relatively easy in case of the lower bound and involved and difficult in case of upper bound. Contrary to that, the proof of the new lower bound requires both new mathematical ideas and difficult proofs.

Theorem 1.1 states that language equations are computationally complete even if either union or intersection and concatenation are used. For this reason, the tempting goal is to show the completeness of unresolved system of equations over sets of natural numbers also with only one Boolean operation and addition. The usage of only one operation (union) is essential for the constructions of systems with addition only in Chapter 7; so there is a reasonable motivation behind.

As some non-trivial results concerning the expressive power of the resolved systems of equations were already obtained in the previous chapters, it would be convenient to reuse them in the following chapters. Unfortunately, this is not possible straight-away, as all these constructions essentially use both intersection and union. Their improvement, so that they use only one Boolean operation, is carried out in Chapter 5. Note that the usage of unresolved systems is crucial, as resolved systems with addition and one Boolean operation have only ultimately-periodic sets as least solutions.

The original construction of the lower bound from Theorem 1.1 is recalled and used as a model for a similar construction in case of systems of equations in Chapter 6.

Further restrictions on the computational model are discussed in the last chapter of this part. That is, systems of equations with addition only are considered. It is shown that in such a case, the use of of finite constants only leads to trivial least and greatest solutions. When ultimately periodic constants are allowed, such restricted systems are computationally universal, as shown in Chapter 7. This is achieved by a special encoding of operations of union and addition by addition only.

Chapter 5

Transforming resolved to unresolved

In the previous chapters of the thesis, resolved systems of equations were studied, i.e., those of the form

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

The general systems of equations, that is:

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

clearly generalise the resolved systems, in particular Theorem 4.2 applies to them as well.

As stated already in the introduction to this part of the thesis, the goal is to construct systems using addition and either union or intersection. So Theorem 4.2 cannot be applied directly, as the resolved systems constructed in it essentially use both union and intersection. Nevertheless, due to greater expressive power of general equations, the constructions from this theorem can be improved so that the use of two Boolean operations is not needed. And this is what is done in this chapter: the construction from the previous chapters are revisited and improved, so that the constructed unresolved systems utilise addition and either union or intersection.

To simplify the process, two general translation lemmata are given. They are used to automate the redesigning of the constructions: they are applied to the systems constructed in Chapter 3 and Chapter 4. As the translation lemmata have strong assumptions, the constructions from Chapter 3 and Chapter 4 are reinvestigated and slightly improved.

5.1 Two general translation lemmata

The first basic result is a simulation of a resolved system of a specific form using union, intersection and addition by an unresolved system that does not use intersection.

Consider resolved systems of equations over sets of numbers. They are of the form

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

where φ_i may contain union, intersection and addition, as well as singleton constants.

This section defines a syntactical transformation of resolved equations of a particular kind into unresolved equations using only one Boolean operation (that is, either union or intersection).

A resolved system of equations is said to have a *chain dependency* of X from Y if the equation defining X is of the form $X = Y \cap \varphi$ or $X = Y \cup \varphi$, where φ is an arbitrary expression.

Lemma 5.1. *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 a family $\mathcal{C}$, where every constant contains only positive integers. Let $(S_1, \ldots, S_n)$ be its least solution. Assume that for every variable X_{i_0} there exists a subset of variables $\{X_i\}_{i \in I}$ containing X_{i_0}, such that*

- *the sets $\{S_i\}_{i \in I}$ are pairwise disjoint and their union is in $\mathcal{C}$, and*
- *the equations for all $\{X_i\}_{i \in I}$ are either all of the form $X_i = \bigcup_j \alpha_{ij}$, or all of the form $X_i = \bigcap_j \alpha_{ij} \cup C_i$, where C_i is a constant and $\alpha_{ij} = A_1 + \ldots + A_\ell$, with $\ell \geq 1$ and with each A_t being a constant or a variable.*

In addition, assume that there are no cyclic chain dependencies in the system. Then there exists an unresolved system with union and addition, with constants from $\mathcal{C}$, which has the unique solution $(S_1, \ldots, S_n)$.

Proof. Such a system is given directly by replacing each equation $X_i = \bigcap_j \alpha_{ij} \cup C_i$, where each α_i is a sum of constants and variables, by the following collection of inequalities:

$$X_i \subseteq \alpha_{ij} \cup C_i, \quad \text{for all } j \ . \tag{5.1}$$

In addition, for each group of variables $\{X_i\}_{i \in I}$, whose union of the group is a constant C_I, the following equation is added:

$$\bigcup_{i \in I} X_i = C_I \ . \tag{5.2}$$

The rest of the equations, which are of the form $X_i = \bigcup_j \alpha_{ij}$, with α_{ij} being a sum of variables and constants, are left as they are. Clearly, the least solution $(S_1, \ldots, S_n)$ of the former system is a solution of the new system. It remains to prove that no other solutions exist.

Assume for the sake of contradiction, that there is another solution $(S'_1, \ldots, S'_n)$. So there is a number $n \in S_i \,\Delta\, S'_i$ for some i. Such a number is called *wrong* or *wrong for* X_i. In particular, if $n \in S'_i \setminus S_i$, then n is said to be an *extra number* for X_i, and if $n \in S_i \setminus S'_i$, then n is a *missing number* for X_i.

Note that the supposed solution satisfies $0 \notin S'_i$ for all i. Indeed, every i belongs to some group of variables I, and then, by (5.2), $S'_i \subseteq C_I$. Since $0 \notin C_I$, zero may not be in S'_i. This, in particular, means that 0 cannot be a wrong number: as for all j it holds that $j \notin C_j$, the system is proper and hence by Lemma 2.1 $0 \notin S_i$ for all i.

Fix $n > 0$ as the smallest wrong number. Then it can be proved that if this number is obtained as a nontrivial sum of variables and constants, it is equally obtained under the substitution of both solutions:

Claim 5.1. *If n is the smallest wrong number and $\alpha = A_1 + \ldots + A_\ell$, where $\ell \geq 2$ and all A_j are variables and constants, then $n \in \alpha(\ldots, S_i, \ldots)$ if and only if $n \in \alpha(\ldots, S'_i, \ldots)$.*

Proof. If $n \in \alpha(\ldots, S_i, \ldots)$, then $n = n_1 + \ldots + n_\ell$, with $n_j \in A_j(\ldots, S_i, \ldots)$. As all sets $A_j(\ldots, S_i, \ldots)$ are 0-free, each number n_j must be positive. Furthermore, each of them must be less than n because $\ell \geq 2$. Since n is the smallest wrong number, none of $n_1, \ldots, n_\ell$ is wrong for its respective variable, and hence $n_j \in A_j(\ldots, S'_i, \ldots)$. The same argument applies for the converse implication. ∎

Among all pairs (n, X_i), where n is the smallest wrong number and it is wrong for X_i, choose a pair such that n is an extra number for X_i, and if this is not possible, then a pair such that n is a missing number for X_i is chosen. Let us show that n must be wrong for another variable $X_{i'}$, with a chain dependency of $X_{i'}$ from X_i.

Suppose that X_i has an equation $X_i = \bigcup_j \alpha_{ij}$ in the original system, which is preserved in the new system. So $S_i = \bigcup_j \alpha_{ij}(\ldots, S_t, \ldots)$. Hence there exists α_{ij}, such that $n \in \alpha_{ij}(\ldots, S_t, \ldots) \,\Delta\, \alpha_{ij}(\ldots, S'_t, \ldots)$. Clearly this α_{ij} cannot be a constant. If it is a variable $X_{i'}$ then we replace S_i by $S_{i'}$. Note, that there is a chain dependency of S_i from $S_{i'}$ and n is wrong for $S'_{i'}$ and if n is an extra number, we can choose $S_{i'}$ so that n is still an extra number for $S_{i'}$. By Claim 5.1 α_{ij} cannot be a non-trivial sum of variables and constants.

Suppose now that the equation for X_i in the original system is of the form $X_i = \bigcap_j \alpha_{ij} \cup C_i$, and n is a missing number. We use (5.2) in this case —let $i \in I$ and $\bigcup_{j \in I} X_j = C_I$. Then by substituting S_i into those equations we obtain that $n \in C_I$. On the other hand by substituting S'_i into those equations we obtain that $n \in S'_{i'}$ for some $i' \in I$ and $i' \neq i$. As $n \in S_i$ then $n \notin S_{i'}$, as $i, i' \in I$ and by assumption sets in the same group are pairwise disjoint. Hence we obtain a contradiction, as n is an extra number for $X_{i'}$ and we are supposed to choose an extra number if there is any.

Let the equation for X_i in the original system be $X_i = \bigcap_j \alpha_{ij} \cup C$ and suppose that n is an extra number. So in the new system there are equations $X_i \subseteq \alpha_{ij} \cup C_i$ for $j \in I$, hence $n \in \alpha_{ij}(\ldots, S'_t, \ldots) \cup C_i$ for $j \in I$. On the other hand $n \notin S_i = \bigcap_{j \in I} \alpha_{ij}(\ldots, S_t, \ldots) \cup C_i$. And so there is $j' \in I$ such that $n \notin \alpha_{ij'}(\ldots, S_t, \ldots) \cup C_i$. Hence $n \in \alpha_{ij'}(\ldots, S'_t, \ldots) \setminus \alpha_{ij'}(\ldots, S_t, \ldots)$. Clearly $\alpha_{ij'}$ cannot be a constant, assuming that it is a non-trivial sum would again derive a contradiction by Claim 5.1. And so $\alpha_{ij'}$ is a variable $X_{i'}$. We replace X_i by $X_{i'}$ and continue the process. Note, that there is a chain dependency of X_i from $X_{i'}$ and n is an extra number for $X_{i'}$.

Now the same argument applies to the pair $(n, X_{i'})$, and in this way an infinite sequence of variables with a chain dependency to their successors is obtained. This is a contradiction, as there are no cyclic chain dependencies in the system. $\square$

A similar construction produces equations with intersection instead of union. The next lemma is very similar in spirit and proof technique to Lemma 5.1, but some technical details are different, therefore it is proved separately.

Lemma 5.2. *Under the assumptions of Lemma 5.1, there exists 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. Here the new system is obtained by the following transformation. For every equation $X_i = \bigcup_j \alpha_{ij}$ in the original system, where each α_{ij} is a sum of constants and variables, the new system contains inequalities

$$\alpha_{ij} \subseteq X_i \quad \text{, for each } j \ . \tag{5.3}$$

For every subset of variables $\{X_i\}_{i \in I}$, with union C_I, the following equations are added:

$$X_i \cap X_j = \varnothing, \qquad \text{for each } i, j \in I \text{ with } i \neq j \ , \tag{5.4}$$

$$X_i \subseteq C_I, \qquad \text{for each } i \in I \ . \tag{5.5}$$

The rest of the equations are of the form $X_i = \bigcap_j \alpha_{ij} \cup C_i$, where C_i is a constant and $\alpha_{ij} = A_1 + \ldots + A_k$, with $k \geq 1$ and each A_t being a constant or a variable. They are changed in the way similar to the equations for union, i.e., are replaced by inequalities

$$C_i \subseteq X_i \ \text{ and } \ \bigcap_j \alpha_{i,j} \subseteq X_i \ .$$

Clearly, the least solution $(\dots, S_i, \dots)$ of the former system is still a solution. It should be proved that no other solution exists.

As in Lemma 5.1, Lemma 2.1 is used to show that the least solution $(\dots, S_i, \dots)$ of the resolved system is 0-free. Also, since the assumptions of the lemma are the same as those of Lemma 5.1, then Claim 5.1 holds.

Suppose that there is another solution $(\dots, S'_i, \dots)$. Note that 0 may not be in any S'_i by the equation (5.5).

Define wrong numbers, missing numbers and extra numbers as in the proof of Lemma 5.1. Let n be the smallest wrong number with $n \in S_i \,\Delta\, S'_i$ for some i. By the above arguments, n must be positive. Among all pairs (n, X_i), such that n is the smallest wrong number and it is wrong for X_i, choose the one in which n is a missing number, if there is any such pair. If there is none, then choose a pair (n, X_i), where n is an extra number for X_i. As in the proof of the previous lemma, the idea is to show that there must be another variable $X_{i'}$ which has a chain dependence on X_i, such that n is a wrong number of $X_{i'}$.

Suppose first that n is an extra number.We use the (5.4) and (5.5) in this case: substituting $(\dots, S_t, \dots)$ into (5.5) one obtains that $n \in C_I$, where $i \in I$. On the other hand, by the assumption of the Lemma $\bigcup_{j\in I} S_j = C_I$, hence there exists $i' \neq i$ such that $n \in S_{i'}$. But by (5.4): $n \notin S'_{i'}$, as $S'_i \cap S'_{i'} = \varnothing$. Hence n is a missing number for i', a contradiction, as we were supposed to choose a missing number if there was any.

Assume now that n is a missing number and in the original resolved system the equation defining S_i is of the from $X_i = \bigcup_j \alpha_{ij}$. By the construction there are equations $\alpha_{ij} \subseteq X_i$ for $j \in I$, hence $n \notin \alpha_{ij}(\dots, S'_t, \dots)$ for $j \in I$. On the other hand $n \in S_i = \bigcup_{j\in I} \alpha_{ij}(\dots, S_t, \dots)$. Hence there is $i' \in I$ such that $n \in \alpha_{i'j}(\dots, S_t, \dots)$ and therefore $n \in \alpha_{i'j}(\dots, S_t, \dots) \setminus \alpha_{i'j}(\dots, S'_t, \dots)$. By Claim 5.1 $\alpha_{i'j}$ cannot be a non-trivial sum. Clearly it cannot be a constant, hence it is a variable. And so $\alpha_{i'j} = X_{i'}$. We swap S_i for $S_{i'}$. Note that there is a chain dependency of X_i from $X_{i'}$ and n is a missing number for $X'_{i'}$.

Suppose now that n is a missing number and in the original system the equation for S_i is of the form $X_i = \bigcap \alpha_{ij} \cup C_i$. The construction assures that there are equations $\bigcap_j \alpha_{ij} \subseteq X_i$ and $C_i \subseteq X_i$ in the new system. Then $n \notin \bigcap_j \alpha_{ij}(\dots, S'_t, \dots)$ and $n \notin C_i$. On the other hand, as $n \in S_i$, it holds that $n \in \bigcap_j \alpha_{ij}(\dots, S_t, \dots) \cup C_i$. Since $n \notin C_i$ by previous observation, $n \in \bigcap_j \alpha_{ij}(\dots, S_t, \dots)$. Thus there exists $i' \in I$ such that $n \in \alpha_{i'j}(\dots, S_t, \dots) \setminus \alpha_{i'j}(\dots, S'_t, \dots)$. Similarly to the analysis in the previous case, $\alpha_{i'j}$ cannot be a constant and by Claim 5.1 it cannot be a non-trivial sum as well. Hence $\alpha_{i'j} = X_{i'}$, for some variable $X_{i'}$, i.e., there is a chain dependency of X_i from $X_{i'}$. We replace S_i with $S_{i'}$, note that n is a missing number for $X_{i'}$.

And so for every n and X_i for which it is wrong another $X_{i'}$ can be found such that n is wrong for it as well and there is a chain dependency of S_i

from $S_{i'}$. Contradiction, as there are no cyclic chain dependencies in the system. □

The next task is to apply Lemmata 5.1 and 5.2 to resolved systems constructed in the proofs of Theorems 3.3 and 4.2. For the lemmata to be applicable, these equations need to be decomposed into smaller parts and slightly changed. Then the variables can be grouped into subsets, as required by the lemmata.

5.2 Sets with regular positional notation

Using the lemmata from the previous section, the resolved equations from Chapter 3 and Chapter 4 will now be converted to unresolved equations with sum and either union or intersection. The first task is to reformulate them so that Lemmata 5.1 and 5.2 are applicable.

Now the first result on the expressive power of equations with one Boolean operation asserts representability of finite and co-finite sets of numbers.

Lemma 5.3. *Every finite or co-finite subset of $\mathbb{N}$ is representable by a unique solution of a resolved system with union and addition, as well as by a unique solution of an unresolved system with intersection and addition.*

Proof. The case of union follows from the fact that every ultimately periodic unary language can be specified by a resolved system of language equations with union, one-sided concatenation and constants $\{a\}$ and $\{\varepsilon\}$.

Let us prove the lemma in the case of intersection, where the use of unresolved equations becomes essential. Let $K = \{n_1, n_2, \ldots, n_m\}$, with $0 \leq n_1 < \ldots < n_m$, be any finite set of numbers. First define the following equations for a variable X:

$$n_m + 1 \subseteq X \ , \tag{5.6a}$$
$$X + 1 \subseteq X \ , \tag{5.6b}$$
$$n_m \cap X = \varnothing \ . \tag{5.6c}$$

Here (5.6b) ensures that the solution is of the form $\{n \mid n \geq k\}$ for some k (or empty), (5.6a) states that $n_m + 1$ is in X, while (5.6c) ensures that n 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:

$$X \cap Y = \varnothing \ , \tag{5.6d}$$
$$n_i \subseteq Y, \quad \text{for } i \in \{1, 2, \ldots, m\} \ , \tag{5.6e}$$
$$n \cap Y = \varnothing, \quad \text{for each } n < n_m \text{ with } n \notin K \ . \tag{5.6f}$$

By (5.6d), Y must be a subset of $\{0, \ldots, n_m\}$. The next two equations state 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:

$$X \subseteq Z \ , \tag{5.6g}$$

$$n_i \cap Z = \varnothing, \quad \text{for } i = 1, 2, \ldots, m \ , \tag{5.6h}$$

$$n_i \subseteq Z, \quad \text{for } n_i < n_m, n_i \notin \{n_1, \ldots, n_m\} \ . \tag{5.6i}$$

The equation (5.6g) states that every number greater than n_m must be in Z. The next two equations define, similarly to the equations for Y, 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. □

Consider a set $(ij\mathtt{0}^*)_k$ for $i \neq \mathtt{0}$. It was shown in Chapter 3 how to construct a system of resolved equations such that this set is a component of the least solution of this system. This result will now be reconstructed to use only one Boolean operation, at the expense of turning the resolved equations into unresolved ones.

Theorem 5.1. *For every $k \geq 9$, there exists an unresolved system with union (intersection), sum and singleton constants, which has a unique solution with some of its components being*

$$(ij\mathtt{0}^*)_k \quad \textit{(for all } i, j \in \Sigma_k \textit{ with } i > 0\textit{)} \ .$$

Proof. A resolved system using union, intersection and addition was already constructed in Theorem 3.2, (3.2)–(3.4). However, the obtained sets cannot be grouped to match the conditions of Lemmata 5.1 and 5.2. The proposed solution relies on representing both these sets and the complementary sets $\widetilde{S}_{ij} = (ij(\Sigma_k^* \setminus \mathtt{0}^*))_k$. Then all sets S_{ij} and $\widetilde{S}_{ij}$ will be pairwise disjoint and their union will be co-finite, making the lemmata applicable.

Define the set of variables $X_{i,j}$, $Y_{i,j}$, $X_{i,j,\ell}$ and $Y_{i,j,\ell}$, with $i, j, \ell \in \Sigma_k$ and

$i \neq \mathtt{0}$, and consider the following resolved system of equations:

$$\begin{aligned}
X_{1,j} &= \bigcap_{n=1}^{2} X_{k-n,0}+X_{j+n,0} \cup (\mathtt{1}j)_k, && \text{for } j = \mathtt{0}, \mathtt{1}, \mathtt{2} \ , \\
X_{i,j} &= \bigcap_{n=1}^{2} X_{i-1,k-n}+X_{j+n,0} \cup (ij)_k, && \text{for } j = \mathtt{0}, \mathtt{1}, \mathtt{2}, \ i \geq 2 \ , \\
X_{i,j} &= \Big(\bigcap_{n=1}^{2} X_{i,j-n}+X_{n,0}\Big) \\
&\quad \cap \ X_{i,0}+X_{j,0} \cup (ij)_k, && \text{for } j \geq 3 \ , \\
X_{i,j,\ell} &= \bigcap_{n=0}^{3} X_{i,n}+X_{j-n,\ell}, && \text{for } j \geq 4, \ i \neq \mathtt{0}, \ \ell \in \Sigma_k \ , \\
X_{i,j,\ell} &= \bigcap_{n=1}^{4} X_{i-1,j+n}+X_{k-n,\ell}, && \text{for } j \leq 3, \ i \neq \mathtt{0}, \mathtt{1}, \ \ell \in \Sigma_k \ , \\
X_{1,j,\ell} &= \bigcap_{n=1}^{4} X_{k-n,0}+X_{j+n,\ell}, && \text{for } j \leq 3, \ \ell \in \Sigma_k \ , \\
Y_{i,j} &= \bigcup_{\ell \neq \mathtt{0}} X_{i,j,\ell} \cup \bigcup_{\ell \in \Sigma_k} Y_{i,j,\ell}, && \text{for } j \in \Sigma_k, \ i \neq \mathtt{0} \ , \\
Y_{i,j,\ell} &= \bigcap_{n=0}^{3} X_{i,n}+Y_{j-n,\ell}, && \text{for } j \geq 4, \ i \neq \mathtt{0}, \ \ell \in \Sigma_k \ , \\
Y_{i,j,\ell} &= \bigcap_{n=1}^{4} X_{i-1,j+n}+Y_{k-n,\ell}, && \text{for } j \leq 3, \ i \neq \mathtt{0}, \mathtt{1}, \ \ell \in \Sigma_k \ , \\
Y_{1,j,\ell} &= \bigcap_{n=1}^{4} X_{k-n,0}+Y_{j+n,\ell}, && \text{for } j \leq 3, \ \ell \in \Sigma_k \ .
\end{aligned}$$

It is claimed that the least solution of those equations is:

$$\begin{aligned}
X_{i,j} &= (ij\mathtt{0}^*)_k \ , \\
X_{i,j,\ell} &= (ij\ell\mathtt{0}^*)_k \ , \\
Y_{i,j} &= (ij(\Sigma_k^* \setminus \mathtt{0}^*))_k \ , \\
Y_{i,j,\ell} &= (ij\ell(\Sigma_k^* \setminus \mathtt{0}^*))_k \ .
\end{aligned}$$

The equations for $X_{i,j}$ are equations (3.2)–(3.4) taken directly from Theorem 3.2 and this theorem asserted that their least solution is $(ij\mathtt{0}^*)_k$, as claimed.

The rest of the equations occur in (3.5)–(3.8), yet some explanations are due in order to recognise them in the above system. This is in fact the system (3.5)–(3.8) constructed in Chapter 3 used in Theorem 3.3, to represent a set of numbers $(L(M))_k$ for a given finite automaton M. Consider that $\Sigma_k^* \setminus \mathtt{0}^*$ is a regular language recognised by a finite automaton $M_0 = \langle \Sigma_k, Q_0, \delta_0, F_0, q_0 \rangle$ reading the string of digits from the right to the left. The automaton has two states, q_0 and q_1; it is in state q_0 while all digits encountered so far are zeroes, and once any non-zero digit is read, it enters state q_1 and remains there.

The construction (3.5)–(3.8) specified to this automaton gives a system in variables $X_{i,j}$, $X_{i,j,\ell}$ and $Y_{i,j}$, $Y_{i,j,\ell}$, with the X-variables corresponding to the state q_0 and with the Y-variables representing the state q_0.

From Lemmata 3.2, 3.3 and 3.4 one may infer that the least solution of this system satisfies

$$\begin{aligned} X_{i,j,\ell} &= (\{ij\ell w \mid \delta_0(q_0, w^r) = q_0\})_k = (ij\ell\mathtt{0}^*)_k \ , \\ Y_{i,j,\ell} &= (\{ij\ell w \mid \delta_0(q_0, w^r) = q_1\})_k = (ij\ell(\Sigma_k^* \setminus \mathtt{0}^*))_k \ , \end{aligned}$$

regardless of the actual equation defining $X_{i,j,\ell}$ or $Y_{i,j,\ell}$. Moreover, by Lemma 3.5, the value of $Y_{i,j}$ in the least solution is

$$Y_{i,j} = (\{ijw \mid \delta_0(q_0, w^r) = q_1\})_k = (ij(\Sigma_k^* \setminus \mathtt{0}^*))_k \ .$$

It remains to show that these equations satisfy the assumptions of Lemmata 5.1 and 5.2, with the variables separated into the following two groups:

$$\{X_{i,j}, Y_{i,j} \mid i, j \in \Sigma_k,\ i \neq \mathtt{0}\},\ \{X_{i,j,\ell}, Y_{i,j,\ell} \mid i, j, \ell \in \Sigma_k,\ i \neq \mathtt{0}\} \ .$$

The unions of the corresponding sets in the least solution for the former group is $\{n \mid n \geq k\}$, and for the latter group it is $\{n \mid n \geq k^2\}$; both are co-finite sets. Clearly, in either group all the components are pairwise disjoint. The only chain dependencies are those of variables $X_{i,j}$ on (some) variables $X_{i,j,\ell}$, as well as of $Y_{i,j}$ on some $Y_{i,j,\ell}$; hence there are no cyclic chain dependencies. And so by Lemma 5.1 and Lemma 5.2 there exist unresolved systems with union (intersection), sum 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 5.3. $\square$

Now the construction of Theorem 3.3 can be remade using unresolved equations with only one Boolean operation.

Lemma 5.4. *For every deterministic finite automaton $M = (\Sigma, Q, q_0, \delta, F)$ there exists an unresolved system of equations using union (intersection), sum and singleton constants, in which some of the components of the unique solution are*

$$S_{i,j,q} := \{(ijw)_k \mid \delta(q_0, w^r) = q\} \quad , \text{for } i, j \in \Sigma_k, i \neq \mathtt{0}, q \in Q \ .$$

Proof. Let us recall the construction (3.5)–(3.8), with constants of the form $(ij\mathtt{0}^*)_k$:

$$
\begin{aligned}
X_{i,j,\ell,q} &= \bigcap_{n=0}^{3} (in\mathtt{0}^*)_k + X_{j-n,\ell,q} && \text{, for } j \geq 4,\ i \geq 1\ , \\
X_{i,j,\ell,q} &= \bigcap_{n=1}^{4} ((i-1)(j+n)\mathtt{0}^*)_k + X_{k-n,\ell,q} && \text{, for } j \leq 3,\ i \geq 2\ , \\
X_{1,j,\ell,q} &= \bigcap_{n=1}^{4} ((k-n)\mathtt{0}\mathtt{0}^*)_k + X_{j+n,\ell,q} && \text{, for } j \leq 3\ , \\
X_{i,j,q} &= \bigcup_{(\ell,q'):\delta(q',\ell)=q} X_{i,j,\ell,q'} \cup \{(ij)_k \mid \text{ if } q = q_0\} && \text{, for } i,j \in \Sigma_k,\ i \leq \mathtt{1}\ .
\end{aligned}
$$

Lemmata 3.2, 3.3 and 3.4. asserted that in the least solution those variables were assigned values

$$X_{i,j,\ell,q} = \{(ij\ell w)_k \mid \delta(q_0, w^r) = q\}\ .$$

Similarly, Lemma 3.5 assured that in the least solution the values of variables $X_{i,j,q}$ is

$$X_{i,j,q} = \{(ijw)_k \mid \delta(q_0, w^r) = q\}\ .$$

The plan is to apply Lemmata 5.1 and 5.2 to the above system. To this end, the variables of the system have to be grouped. There are two groups:

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

The union of the least solution in the former group is $\{n \mid n \geq k\}$, and $\{n \mid n \geq k^2\}$ for the latter. The sets within each group are clearly disjoint.

The resulting system uses two co-finite constants obtained as unions of the groups, as well as constants of the form $(ij\mathtt{0}^*)_k$. The former are expressed as in Lemma 5.3, while the latter are replaced by references to equations from Theorem 5.1. □

Theorem 5.2. *For every $k \geq 2$ and for every regular language $L \subseteq \Sigma_k^* \setminus \mathtt{0}\Sigma_k^*$ there exists an unresolved system of equations with union (intersection), addition and singleton constants, which has a unique solution with $(L)_k$ as one of its components.*

Proof. First consider the case of $2 \leq k < 9$. Then, by Lemma 3.1, 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 $k \geq 9$.

Let $M = (\Sigma, Q, q_0, \delta, F)$ be a deterministic finite automaton recognizing L^r. By Lemma 5.4, there exists an unresolved system of the specified form, in which every variable $X_{i,j,q}$ in the unique solution equals $\{(ijw)_k \,|\, \delta(q_0, w^r) = q\}$. Then the set $(L)_k$ can be obtained as the following union:

$$(L)_k = \big(\underbrace{(L)_k \cap \{n \mid n < k\}}_{\text{finite constant}}\big) \cup \bigcup_{\substack{i,j,q:\\ \delta(q,ji)\in F}} \underbrace{\{(ijw)_k \mid \delta(q_0, w^r) = q\}}_{X_{i,j,q}} \ . \tag{5.7}$$

In the case of unresolved equations with union, the equality (5.7) can be directly specified by introducing a new variable Y and adding the following equation:

$$Y = \big((L)_k \cap \{n \mid n < k\}\big) \cup \bigcup_{\substack{i,j,q:\\ \delta(q,ji)\in F}} X_{i,j,q} \ .$$

The finite constant $(L)_k \cap \{n \,|\, n < k\}$ is expressed according to Lemma 5.3.

For the case of intersection, consider that the sets $\{(ijw)_k|\delta(q_0, w^r) = q\}$, along with the finite set $\{n \mid n < k\}$, form a partition of $\mathbb{N}$. Then a new variable Y is added, and its intersection with every element of this partition is expressed:

$$\begin{aligned} Y \cap \{n \mid n < k\} &= (L \cap \Sigma_k^{\leq 1})_k \ , & & \\ Y \cap X_{i,j,q} &= \varnothing & &, \text{for } (i,j,q) \text{ such that } \delta(q,ji) \notin F \ , \\ Y \cap X_{i,j,q} &= X_{i,j,q} & &, \text{for } (i,j,q) \text{ such that } \delta(q,ji) \in F \ . \end{aligned}$$

Because these equalities state the membership of *every* natural number in Y, this representation is equivalent to (5.7), and hence the system has a unique solution with $Y = (L)_k$. Both finite constants are again replaced according to Lemma 5.3. $\square$

5.3 Sets trellis positional notation

The next task is to remake construction form Chapter 4 using only one Boolean operation. As stated in Theorem 4.2, for every trellis automaton M with $L(M) \subseteq \Sigma_k^+ \setminus \mathtt{0}\Sigma_k^*$, 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 the goal is again to refine it so that Lemmata 5.1 and 5.2 could be applied to it.

We follow the three stages of the construction : first, the set $(\mathtt{1}(L(M) \boxminus \mathtt{1})\mathtt{10}^*)_k$ was represented (Lemma 4.3); next, $(\mathtt{1} \cdot L(M))_k$ (Lemma 4.4); and finally, a system for $(L(M))_k$ was obtained (Lemma 4.5). This composition will be followed in the below proof, and each part of the construction will be carefully remade.

Lemma 5.5. *For every $k \geq 5$ and for every trellis automaton M over $\Sigma_k = \{\mathtt{0}, \ldots, k-1\}$ with $L(M) \cap \mathtt{0}\Sigma_k^* = \varnothing$, 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 singleton constants, such that the unique solution of this system contains a component*

$$(\mathtt{1}(L_M(q) \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k = \{(\mathtt{1}w\mathtt{1}\mathtt{0}^\ell)_k \mid \ell \geq 0, w \notin (k-1)^*, w \boxplus \mathtt{1} \in L_M(q)\} \ .$$

Proof. Let $M = (\Sigma_k, Q, I, \delta, F)$ be any trellis automaton. Let us recall the resolved system representing the given sets of numbers introduced in Lemma 4.3. It uses variables X_q over all $q \in Q$ and contains the equations

$$X_q = R_q \ \cup \bigcup_{\substack{q', q'': \delta(q', q'') = q \\ i, j \in \Sigma_k}} \lambda_i(X_{q''}) \cap \rho_j(X_{q'}) \quad (\text{, for all } q \in Q) \ ,$$

where

$$\begin{aligned}
R_q &= \{(\mathtt{1}(w \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k \mid w \in \mathtt{0}^*(\Sigma_k \setminus \mathtt{0}) \cup (\Sigma_k \setminus \mathtt{0})\mathtt{0}^*,\ w \in L_M(q)\} \ , \\
\kappa_{i'}(X) &= \big(X \cap (\mathtt{1}i'\Sigma_k^*\mathtt{1}\mathtt{0}^*)_k\big) + (\mathtt{1}\mathtt{0}^*)_k \ \cap \ (\mathtt{2}i'\Sigma_k^*)_k, \\
&\qquad\qquad\qquad\qquad \text{for all } i' \in \Sigma_k \ , \\
\lambda_i(X) &= \bigcup_{i' \in \Sigma_k} \big(\kappa_{i'}(X) + ((k+i-2)\mathtt{0}^*)_k \cap (\mathtt{1}i\Sigma_k^*)_k\big), \\
&\qquad\qquad\qquad\qquad \text{for } i = 0, 1 \ , \\
\lambda_i(X) &= \bigcup_{i' \in \Sigma_k} \big(\kappa_{i'}(X) + (\mathtt{1}(i-2)\mathtt{0}^*)_k \cap (\mathtt{1}i\Sigma_k^*)_k\big), \\
&\qquad\qquad\qquad\qquad \text{for } i \geq 2 \ ,
\end{aligned}$$

$$\begin{aligned}
\pi_{j'}(X) &= \big(X \cap (\mathtt{1}\Sigma_k^*j'\mathtt{1}\mathtt{0}^*)_k\big) + (\mathtt{1}\mathtt{0}^*)_k \ \cap \ (\mathtt{1}\Sigma_k^*j'\mathtt{2}\mathtt{0}^*)_k, \\
&\qquad\qquad\qquad\qquad \text{for } j' \in \Sigma_k \ , \\
\rho_j(X) &= \bigcup_{j' \in \Sigma_k} \big(\pi_{j'}(X) + ((k+j-2)\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^*j\mathtt{1}\mathtt{0}^*)_k\big), \\
&\qquad\qquad\qquad\qquad \text{for } j = 0, 1 \ , \\
\rho_j(X) &= \bigcup_{j' \in \Sigma_k} \big(\pi_{j'}(X) + (\mathtt{1}(j-2)\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^*j\mathtt{1}\mathtt{0}^*)_k\big), \\
&\qquad\qquad\qquad\qquad \text{for } 2 \leq j \leq k-2 \ , \\
\rho_{k-1}(X) &= \bigcup_{j' \in \Sigma_k} \big(\pi_{j'}(X) + ((k-3)\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^*(k-1)\mathtt{1}\mathtt{0}^*)_k\big) \ .
\end{aligned}$$

All constants used in the system have regular base-k notation.

The least solution is $X_q = S_q$ by Main Claim from Section 4.2, where

$$\begin{aligned}
S_q &= (\mathtt{1}((L_M(q) \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k \\
&= \{(\mathtt{1}w\mathtt{1}\mathtt{0}^\ell)_k \mid \ell \geq 0, w \notin (k-1)^*, w \boxplus \mathtt{1} \in L_M(q)\} \ .
\end{aligned}$$

These sets are pairwise disjoint and their union is a set with a regular base-k notation. In order to prove this, let us establish a more general statement that will be used several times in the following:

Claim 5.2. *Let $x \in \Sigma_k^* \setminus 0\Sigma_k^*$ and $y \in \Sigma_k^+ \setminus 0^*$ be 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*

$$S_t = \{(xuy0^\ell)_k \mid \ell \geq 0,\ u \in K_t\} \ .$$

Then these sets are pairwise disjoint and their union is

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

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 since $K_t \cap K_{t'} = \varnothing$ by assumption. This proves that $S_t \cap S_{t'} = \varnothing$.

The union of these sets is

$$\bigcup_t S_t = \bigcup_t (xK_ty0^*)_k = (x(\textstyle\bigcup_t K_t)y0^*)_k \ ,$$

as stated. □

Now Claim 5.2 can be applied to the particular case of the sets S_q to obtain the following result:

Claim 5.3. *The sets of numbers S_q with different $q \in Q$ are pairwise disjoint, and their union is*

$$\bigcup_q S_q = (1(\Sigma_k^+ \setminus (k-1)^*)10^*)_k \ .$$

Proof. Since a trellis automaton computes a uniquely determined state $\delta(I(w)) \in Q$ on each string $w \in \Sigma_k^+$, it induces a partition of Σ_k^+ into classes corresponding to different states. Define $K_q = (L_M(q) \setminus 0^*) \boxminus 1$; these sets are pairwise disjoint and their union for all $q \in Q$ is $\Sigma_k^* \setminus (k-1)^*$, since every string $w \in \Sigma_k^+$ belongs to some $L_M(q)$. The rest is given by Claim 5.2 with $x = y = 1$. □

Though the values of the variables X_q as they are already satisfy Lemma 5.1 and Lemma 5.2, the equations defining them 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 representing λ and ρ. Each occurrence of $\lambda_i(X_q)$ will be replaced by a new variable $Z^\lambda_{i,q}$, and similarly $\kappa_{i'}(X_q)$ is replaced by $W^\lambda_{i',q}$, where the new variables have the following equations:

$$U^\lambda_{i',q} = X_q \cap (\mathtt{1}i'\Sigma_k^*\mathtt{10}^*)_k \ , \tag{5.8}$$
$$W^\lambda_{i',q} = U^\lambda_{i',q} + (\mathtt{10}^*)_k \cap (\mathtt{2}i'\Sigma_k^*)_k \ , \tag{5.9}$$
$$Y^\lambda_{i,i',q} = W^\lambda_{i',q} + (\mathtt{1}(i-2)\mathtt{0}^*)_k \cap (\mathtt{1}i\Sigma_k^*)_k \qquad , \text{ for } i \geq 3 \ , \tag{5.10}$$
$$Y^\lambda_{i,i',q} = W^\lambda_{i',q} + ((k+i-2)\mathtt{0}^*)_k \cap (\mathtt{1}i\Sigma_k^*)_k \qquad , \text{ for } i \leq 2 \ , \tag{5.11}$$
$$Z^\lambda_{i,q} = \bigcup_{i'} Y^\lambda_{i,i',q} \ . \tag{5.12}$$

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:

$$\{U^\lambda_{i',q} \mid i' \in \Sigma_k, q \in Q\}, \quad \{W^\lambda_{i',q} \mid i' \in \Sigma_k, q \in Q\} \ ,$$
$$\{Y^\lambda_{i,i',q} \mid i, i' \in \Sigma_k, q \in Q\} \quad \{Z^\lambda_{i,q} \mid i \in \Sigma_k, q \in Q\} \ .$$

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

Claim 5.4. *Consider the least solution of* (5.8)–(5.12). *For all* $(i'_1, q_1) \neq (i'_2, q_2)$ *sets* $U^\lambda_{i'_1,q_1}$ *and* $U^\lambda_{i'_2,q_2}$ *are disjoint and and their union is*

$$\bigcup_{i',q} U^\lambda_{i',q} = (\mathtt{1}(\Sigma_k^+ \setminus (k-1)^*)\mathtt{10}^*)_k \ .$$

Proof. By (4.1)

$$U^\lambda_{i',q} = \{(\mathtt{1}i'w\mathtt{10}^\ell)_k \mid \ell \geq 0, i'w \notin (k-1)^*, i'w \boxplus \mathtt{1} \in L_M(q)\} \ .$$

These sets are obtained from the languages $K^\lambda_{i',q} = ((L_M(q) \setminus \mathtt{0}^*) \boxminus \mathtt{1}) \cap i'\Sigma_k^*$ with $i' \in \Sigma_k$ and $q \in Q$ as in the statement of Claim 5.2. To see that the sets $K^\lambda_{i',q}$ 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 words in these sets start from different digits, and if $q_1 \neq q_2$, then $K^\lambda_{i'_1,q_1} \subseteq L_M(q_1) \boxminus \mathtt{1}$ and $K^\lambda_{i'_2,q_2} \subseteq L_M(q_2) \boxminus \mathtt{1}$. In both cases, $K^\lambda_{i_1,q_1} \cap K^\lambda_{i_2,q_2} = \varnothing$.

Therefore, Claim 5.2 with $x = y = \mathtt{1}$ asserts that $U^{\lambda}_{i',q}$ are pairwise disjoint and the union of this group of sets is

$$\begin{aligned}
\bigcup_{i',q} U^{\lambda}_{i',q} &= (\mathtt{1}(\textstyle\bigcup_{i',q} K_{i',q})\mathtt{1}\mathtt{0}^*)_k \\
&= (\mathtt{1}\big(\textstyle\bigcup_{i',q}((L_M(q) \setminus \mathtt{0}^*) \boxminus \mathtt{1}) \cap i'\Sigma_k^*\big)\mathtt{1}\mathtt{0}^*)_k \\
&= (\mathtt{1}((\textstyle\bigcup_q L_M(q) \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k \\
&= (\mathtt{1}((\Sigma_k^* \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k \\
&= (\mathtt{1}(\Sigma_k^+ \setminus (k-1)^*)\mathtt{1}\mathtt{0}^*)_k \ ,
\end{aligned}$$

which completes the proof. □

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

Claim 5.5. *Consider the least solution of* (5.8)–(5.12)*. For all* $(i_1, q_1) \neq (i_2, q_2)$ *the sets* $W^{\lambda}_{i'_1,q_1}$ *and* $W^{\lambda}_{i'_2,q_2}$ *are disjoint and the union of all sets in the group is:*

$$\bigcup_{i',q} W^{\lambda}_{i',q} = (\mathtt{2}(\Sigma_k^+ \setminus (k-1)^*)\mathtt{1}\mathtt{0}^*)_k \ .$$

Proof. By (4.2)

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

These sets are induced by $K^{\lambda}_{i',q} = ((L_M(q) \setminus \mathtt{0}^*) \boxminus \mathtt{1}) \cap i'\Sigma_k^*$ with $i' \in \Sigma_k$ and $q \in Q$ as in Claim 5.2 with $x = \mathtt{2}$ and $y = \mathtt{1}$. It has been proved in Claim 5.4 that $K^{\lambda}_{i',q}$ are pairwise disjoint and their union is $\Sigma_k^+ \setminus (k-1)^+$. Both statements of the present claim follow. □

Claim 5.6. *Consider the least solution of* (5.8)–(5.12)*. For all* $(i_1, i'_1, q_1) \neq (i_2, i'_2, q_2)$ *it holds that* $Y^{\lambda}_{i_1,i'_1,q_1} \cap Y^{\lambda}_{i_1,i'_1,q_1} = \varnothing$*. Their union is:*

$$\bigcup_{i,i',q} Y^{\lambda}_{i,i',q} = (\mathtt{1}\Sigma_k(\Sigma_k^+ \setminus (k-1)^*)\mathtt{1}\mathtt{0}^*)_k \ .$$

Proof. By (4.3)–(4.4)

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

Then, for each fixed i, those sets are obtained from the languages $K^{\lambda}_{i',q} = ((L_M(q) \setminus \mathtt{0}^*) \boxminus \mathtt{1}) \cap i'\Sigma_k^*$ as in Claim 5.2 with $x = \mathtt{1}i$ and $y = \mathtt{1}$. It was shown in Claim 5.4 that $K^{\lambda}_{i',q}$ are pairwise disjoint and their union is $\Sigma_k^+ \setminus (k-1)^*$. Thus, for each i,

$$\bigcup_{i',q} Y^{\lambda}_{i,i',q} = (\mathtt{1}i(\Sigma_k^+ \setminus (k-1)^*)\mathtt{1}\mathtt{0}^*)_k \ ,$$

and for all $(i'_1, q_1) \neq (i'_2, q_2)$ the sets Y^λ_{i,i'_1,q_1} and Y^λ_{i,i'_2,q_2} are disjoint. Then, clearly,

$$\bigcup_{i,i',q} Y^\lambda_{i,i',q} = \bigcup_i (\mathtt{1}i(\Sigma_k^+ \setminus (k-1)^*)\mathtt{10}^*)_k = (\mathtt{1}\Sigma_k(\Sigma_k^+ \setminus (k-1)^*)\mathtt{10}^*)_k \ ,$$

What is left to show is that for $(i_1, i'_1, q_1) \neq (i_2, i'_2, q_2)$, the sets Y_{i_1,i'_1,q_1} and Y_{i_2,i'_2,q_2} are disjoint. If $i_1 = i_2$, then $(i'_1, q_1) \neq (i'_2, q_2)$, and such sets were already shown to have empty intersection. If $i_1 \neq i_2$, then these sets consist of numbers with a different second leading digit, and are bound to be disjoint as well. $\square$

Claim 5.7. *Consider the least solution of* (5.8)–(5.12). *For all* $(i_1, q_1) \neq (i_2, q_2)$*, the sets* $Z^\lambda_{i_1,q_1}$ *and* $Z^\lambda_{i_2,q_2}$ *are disjoint, and their union equals*

$$\bigcup_{i,q} Z^\lambda_{i,q} = (\mathtt{1}\Sigma_k(\Sigma_k^+ \setminus (k-1)^*)\mathtt{10}^*)_k \ .$$

Proof. The equation (5.12) defines $Z^\lambda_{i,q}$ as the union of $Y^\lambda_{i,i',q}$ over all i', and the values of the latter variables are known from Claim 5.6. Then the value of $Z^\lambda_{i,q}$ is calculated as follows:

$$\begin{aligned}\bigcup_{i,q} Z^\lambda_{i,q} &= \bigcup_{i,q} (\textstyle\bigcup_{i'} Y^\lambda_{i,i',q}) \\ &= \bigcup_{i,i',q} Y^\lambda_{i,i',q} \\ &= (\mathtt{1}\Sigma(\Sigma_k^+ \setminus (k-1)^*)\mathtt{10}^*)_k \ .\end{aligned}$$

The sets $Z^\lambda_{i,q}$ are pairwise disjoint as disjoint unions of pairwise disjoint sets. $\square$

The equations for ρ will now undergo a similar reconstruction. Every $\rho_j(X_q)$ is replaced by $U^\rho_{j,q}$ and each $\pi_{j'}(X_q)$ by $W^\rho_{j',q}(X_q)$. The new variables are defined by the following resolved equations:

$$\begin{aligned}
U^\rho_{j',q} &= X_q \cap (\mathtt{1}\Sigma_k^* j' \mathtt{10}^*)_k \ , && (5.13)\\
W^\rho_{j',q} &= U^\rho_{j',q} + (\mathtt{10}^*)_k \ \cap \ (\mathtt{1}\Sigma_k^* j' \mathtt{20}^*)_k \ , && (5.14)\\
Y^\rho_{j,j',q} &= W^\rho_{j',q} + (\mathtt{1}(k+j-2)\mathtt{10}^*)_k \ \cap \ (\mathtt{1}\Sigma_k^* j \mathtt{10}^*)_k \quad , \text{for } j < 2 \ , && (5.15)\\
Y^\rho_{j,j',q} &= W^\rho_{j',q} + (\mathtt{1}(j-2)\mathtt{10}^*)_k \ \cap \ (\mathtt{1}\Sigma_k^* j \mathtt{10}^*)_k \quad , \text{for } 2 \leq j < k-1 \ , && (5.16)\\
Y^\rho_{k-1,j',q} &= W^\rho_{j',q} + ((k-3)\mathtt{10}^*)_k \ \cap \ (\mathtt{1}\Sigma_k^*(k-1)\mathtt{10}^*)_k \ , && (5.17)\\
Z^\rho_{j,q} &= \bigcup_{j'} Y^\rho_{j,j',q} \ . && (5.18)
\end{aligned}$$

As the new equations represent the subexpressions of $\rho_j(X_q)$, the values of the variables X_q in the least solution of the new system are the same as in the least solution of the old system.

These variables are grouped as follows:

$$\{U^\rho_{j',q} \mid j' \in \Sigma_k, q \in Q\}, \quad \{W^\rho_{j',q} \mid j' \in \Sigma_k, q \in Q\} \ ,$$
$$\{Y^\rho_{j,j',q} \mid j, j' \in \Sigma_k, q \in Q\}, \quad \{Z^\rho_{j,q} \mid j \in \Sigma_k, q \in Q\} \ .$$

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 notation.

Claim 5.8. *Consider the least solution of* (5.13)–(5.18). *For all* $(j'_1, q_1) \neq (j'_2, q_2)$, *the sets* $U^\rho_{j'_1,q_1}$ *and* $U^\rho_{j'_2,q_2}$ *are disjoint, and their union is*

$$\bigcup_{j',q} U^\rho_{j',q} = (\mathtt{1}(\Sigma_k^+ \setminus (k-1)^*)\mathtt{10}^*)_k \ .$$

Proof. By (4.6) in the least solution

$$U^\rho_{j',q} = \{(\mathtt{1}wj'\mathtt{10}^\ell)_k \mid \ell \geq 0, wj' \notin (k-1)^*, wj' \boxplus \mathtt{1} \in L_M(q)\} \ .$$

These sets can be obtained from the languages $K^\rho_{j',q} = (L_M(q) \boxminus \mathtt{1}) \cap \Sigma_k^* j'$ as in Claim 5.2 with $x = \mathtt{1}$ and $y = \mathtt{1}$. The languages $K^\rho_{j'_1,q_1}$ and $K^\rho_{j'_2,q_2}$ are disjoint for all $(j'_1, q_1) \neq (j'_2, q_2)$, as for $j'_1 \neq j'_2$ their last digits are different, while for $q_1 \neq q_2$ it holds that $K^\rho_{j'_1,q_1} \subseteq L_M(q_1) \boxminus \mathtt{1}$ and $K^\rho_{j'_2,q_2} \subseteq L_M(q_2) \boxminus \mathtt{1}$, and the supersets are disjoint. Then, by Claim 5.2, $U^\rho_{j'_1,q_1} \cap U^\rho_{j'_2,q_2} = \varnothing$ for $(j'_1, q_1) \neq (j'_2, q_2)$, while the union of these sets is

$$\begin{aligned}
\bigcup_{j' \in \Sigma_k, q \in Q} U^\rho_{j',q} &= (\mathtt{1}(\textstyle\bigcup_{j' \in \Sigma_k, q \in Q} K^\rho_{j',q})\mathtt{10}^*)_k \\
&= (\mathtt{1}(\bigcup_{j' \in \Sigma_k, q \in Q} (L_M(q) \boxminus \mathtt{1}) \cap \Sigma_k^* j')\mathtt{10}^*)_k \\
&= (\mathtt{1}(\textstyle\bigcup_{q \in Q} (L_M(q) \boxminus \mathtt{1}) \cap \Sigma_k^+)\mathtt{10}^*)_k \\
&= (\mathtt{1}((\Sigma_k^* \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{10}^*)_k \\
&= (\mathtt{1}(\Sigma_k^+ \setminus (k-1)^*)\mathtt{10}^*)_k \ ,
\end{aligned}$$

and the claim follows. ∎

Claim 5.9. *Consider the least solution of* (5.13)–(5.18). *For* $(j'_1, q_1) \neq (j'_2, q_2)$, *the sets* $W^\rho_{j'_1,q_1}$ *abd* $W^\rho_{j'_2,q_2}$ *are disjoint, and*

$$\bigcup_{j',q} W^\rho_{j',q} = (\mathtt{1}(\Sigma_k^+ \setminus (k-1)^*)\mathtt{20}^*)_k \ .$$

Proof. By (4.7) the value of this variable in the least colution is

$$W^{\rho}_{j',q} = \{(\mathtt{1}wj'\mathtt{2}\mathtt{0}^{\ell})_k \mid wj' \boxplus \mathtt{1} \in L_M(q),\ wj' \notin (k-1)^*,\ \ell \geq 0\} \ .$$

These sets are induced by the languages $K^{\rho}_{j',q} = (L_M(q) \boxminus \mathtt{1}) \cap \Sigma_k^* j'$ as in Claim 5.2 with $x = \mathtt{1}$ and $y = \mathtt{2}$. These languages appeared already in Claim 5.8, where it was shown that they are pairwise disjoint and their union is $\Sigma_k^+ \setminus (k-1)^*$. Then, by Claim 5.2, for all $(j'_1, q_1) \neq (j'_2, q_2)$, the sets $W^{\rho}_{j'_1,q_1}$ and $W^{\rho}_{j'_2,q_2}$ are disjoint, and

$$\bigcup_{j',q} W^{\rho}_{j',q} = (\mathtt{1}(\textstyle\bigcup_{j',q} K^{\rho}_{j',q})\mathtt{2}\mathtt{0}^*)_k = (\mathtt{1}(\Sigma_k^+ \setminus (k-1)^*)\mathtt{2}\mathtt{0}^*)_k \ ,$$

which completes the proof. $\square$

Claim 5.10. *Consider the least solution of* (5.13)–(5.18)*. The sets $Y^{\rho}_{j_1,j'_1,q_1}$ and $Y^{\rho}_{j_2,j'_2,q_2}$ are disjoint for all $(j_1, j'_1, q_1) \neq (j_2, j'_2, q_2)$, and the union in the group equals*

$$\bigcup_{j,j',q} Y^{\rho}_{j,j',q} = (\mathtt{1}((\Sigma_k^* \setminus \mathtt{0}^*)\Sigma_k \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k.$$

Proof. By (4.8) and (4.9)

$$\begin{aligned} Y^{\rho}_{j,j',q} = \{&(\mathtt{1}(w'j' \boxplus \mathtt{1})j\mathtt{1}\mathtt{0}^{\ell-1})_k \mid \\ &\ell \geq 1,\ w'j' \notin (k-1)^*,\ w'j' \boxplus \mathtt{1} \in L_M(q)\} \ , \end{aligned}$$

for all $j \neq k-1$, and by (4.10)

$$\begin{aligned} Y^{\rho}_{k-1,j',q} = \{&(\mathtt{1}w'j'(k-1)\mathtt{1}\mathtt{0}^{\ell-1})_k \mid \\ &\ell \geq 1,\ w'j' \notin (k-1)^*,\ w'j' \boxplus \mathtt{1} \in L_M(q)\} \ . \end{aligned}$$

Fix any $j \neq k-1$. Then the sets $Y^{\rho}_{j,j',q}$ are obtained from the languages $K^{\rho}_{j,j',q} = (L_M(q) \setminus \mathtt{0}^*) \cap (\Sigma_k^* j' \boxplus \mathtt{1})$ as in Claim 5.2 with $x = \mathtt{1}$ and $y = j\mathtt{1}$. Then, for all $(j'_1, q_1) \neq (j'_2, q_2)$, the languages K^{ρ}_{j,j'_1,q_1} and K^{ρ}_{j,j'_2,q_2} are disjoint, as for $q_1 \neq q_2$ $K^{\rho}_{j,j'_1,q_1} \subseteq L_M(q_1)$ and $K^{\rho}_{j,j'_2,q_2} \subseteq L_M(q_2)$, and the supersets are disjoint. If $j'_1 \neq j'_2$, then the strings from these languages differ in the last digit. Therefore, by Claim 5.2,

$$\begin{aligned} \bigcup_{j',q} Y^{\rho}_{j,j',q} &= (\mathtt{1}(\textstyle\bigcup_{j',q} K^{\rho}_{j,j',q})j\mathtt{1}\mathtt{0}^*)_k \\ &= (\mathtt{1}(\bigcup_{j',q}(L_M(q) \setminus \mathtt{0}^*) \cap (\Sigma_k^* j' \boxplus \mathtt{1}))j\mathtt{1}\mathtt{0}^*)_k \\ &= (\mathtt{1}(\textstyle\bigcup_{j'}(\Sigma_k^+ \setminus \mathtt{0}^*) \cap (\Sigma_k^* j' \boxplus \mathtt{1}))j\mathtt{1}\mathtt{0}^*)_k \\ &= (\mathtt{1}((\Sigma_k^+ \setminus \mathtt{0}^*) \cap (\Sigma_k^+ \boxplus \mathtt{1}))j\mathtt{1}\mathtt{0}^*)_k \\ &= (\mathtt{1}(\Sigma_k^+ \setminus \mathtt{0}^*)j\mathtt{1}\mathtt{0}^*)_k \\ &= (\mathtt{1}((\Sigma_k^+ \setminus \mathtt{0}^*)(j+1) \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k \ , \end{aligned}$$

and $Y^\rho_{j,j'_1,q_1} \cap Y^\rho_{j,j'_2,q_2} = \varnothing$ for all $(j'_1, q_1) \neq (j'_2, q_2)$.

Next, consider the case of $j = k-1$ and recall the languages $K^\rho_{j',q} = (L_M(q) \boxminus \mathtt{1}) \cap \Sigma_k^* j$ introduced in Claim 5.8, where it was shown that these languages are pairwise disjoint and their union is

$$\bigcup_{j',q} K^\rho_{j',q} = \Sigma_k^+ \setminus (k-1)^* \ .$$

Now the sets Y^ρ_{k-1,j'_1,q_1} can be obtained from the languages $K^\rho_{j',q}$ by the method of Claim 5.2 with $x = \mathtt{1}$ and $y = (k-1)\mathtt{1}$. Therefore,

$$\begin{aligned}\bigcup_{j',q} Y^\rho_{k-1,j'_1,q_1} &= (\mathtt{1}(\Sigma_k^+ \setminus (k-1)^*)(k-1)\mathtt{1}\mathtt{0}^*)_k \\ &= (\mathtt{1}((\Sigma_k^+ \setminus \mathtt{0}^*)\mathtt{0} \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k \ ,\end{aligned}$$

where the the first equality comes from Claim 5.2 and the second one is a simple calculation. Also, for different $(j'_1, q_1) \neq (j'_2, q_2)$, the sets Y^ρ_{k-1,j'_1,q_1} and Y^ρ_{k-1,j'_2,q_2} are disjoint.

Finally, in order to prove the claim, consider any two sets $Y^\rho_{j_1,j'_1,q_1}$ and $Y^\rho_{j_2,j'_2,q_2}$ with $(j_1, j'_1, q_1) \neq (j_2, j'_2, q_2)$. If $j_1 \neq j_2$, then these sets are disjoint, as their elements differ in the second from the last non-zero digit. If $j_1 = j_2$ and $(j'_1, q_1) \neq (j'_2, q_2)$, then these two sets have been proved to be disjoint in one of the cases above.

The union of all these sets is

$$\begin{aligned}\bigcup_{j,j',q} Y^\rho_{j,j',q} &= \bigcup_{j \neq k-1} \bigcup_{j',q} Y^\rho_{j,j',q} \cup \bigcup_{j',q} Y^\rho_{k-1,j',q} \\ &= \bigcup_{j \neq k-1} (\mathtt{1}((\Sigma_k^+ \setminus \mathtt{0}^*)(j+1) \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k \cup (\mathtt{1}((\Sigma_k^+ \setminus \mathtt{0}^*)\mathtt{0} \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k \\ &= (\mathtt{1}((\Sigma_k^+ \setminus \mathtt{0}^*)\Sigma_k \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k \ ,\end{aligned}$$

which establishes the claim. □

Claim 5.11. *Consider the least solution of* (5.13)–(5.18)*. The sets assigned to $Z^\rho_{j_1,q_1}$ and $Z^\rho_{j_2,q_2}$ are disjoint for $(j_1, q_1) \neq (j_2, q_2)$. Their union equals*

$$\bigcup_{j,q} Z^\rho_{j,q} = (\mathtt{1}((\Sigma_k^* \setminus \mathtt{0}^*)\Sigma_k \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k \ .$$

Proof. The variable $Z^\rho_{j,q}$ is defined by (5.18) as the union of $Y^\rho_{j,j',q}$ for all j'. Then

$$\begin{aligned}\bigcup_{j,q} Z^\rho_{j,q} &= \bigcup_{j,q} \bigcup_{j'} Y^\rho_{j,j',q} \\ &= \bigcup_{j,j',q} Y^\rho_{j,j',q} \\ &= (\mathtt{1}((\Sigma_k^* \setminus \mathtt{0}^*)\Sigma_k \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k \ ,\end{aligned}$$

where the second equality is given by Claim 5.10. The latter claim also states that the sets $Y^{\rho}_{j,j',q}$ are pairwise disjoint, and hence so are the sets $Z^{\rho}_{j,q}$. ☐

Thus the expressions $\lambda_i(X_q)$ and $\rho_j(X_q)$ have been expressed by equations of the form satisfying the assumptions of Lemma 5.1 and Lemma 5.2. It remains to transform the equation defining X_q to the same form. The original equation was

$$X_q = R_q \cup \bigcup_{\substack{q',q'' \colon \delta(q',q'')=q \\ i,j \in \Sigma_k}} \lambda_i(X_{q''}) \cap \rho_j(X_{q'}) \ ,$$

The subexpression corresponding to every i, q'', j and q' shall be represented by a new variable $X_{i,q'',j,q'}$ with the equation

$$X_{i,q'',j,q'} = Z^{\lambda}_{i,q''} \cap Z^{\rho}_{j,q'} \ , \tag{5.19}$$

while the equation for X_q is accordingly replaced by

$$X_q = R_q \cup \bigcup_{\substack{q',q'' \colon \delta(q',q'')=q \\ i,j \in \Sigma_k}} X_{i,q'',j,q'} \ . \tag{5.20}$$

The variables are divided into two groups,

$$\{X_{i,q'',j,q'} \mid i,j \in \Sigma_k, q',q'' \in Q\}, \quad \{X_q \mid q \in Q\}$$

and it remains to show the required properties of the variables in each group.

Claim 5.12. *Consider the least solution of* (5.13)–(5.18). *For all* $(i_1,q''_1,j_1,q'_1) \neq (i_2,q''_2,j_2,q'_2)$, *the sets* X_{i_1,q''_1,j_1,q'_1} *and* X_{i_2,q''_2,j_1,q'_2} *are disjoint, and the union of all these sets is*

$$\bigcup_{i,q'',j,q'} X_{i,q'',j,q'} = (\mathtt{1}((\Sigma_k^* \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{10}^*)_k \setminus \textstyle\bigcup_q R_q \ .$$

Proof. According to (5.19), $X_{i,q'',j,q'} = Z^{\lambda}_{i,q''} \cap Z^{\rho}_{j,q'}$. By Claim 5.7, $Z^{\lambda}_{i_1,q''_1} \cap Z^{\lambda}_{i_2,q''_2} = \varnothing$ for $(i_1,q''_1) \neq (i_2,q''_2)$. Similarly, by Claim 5.11, $Z^{\rho}_{j_1,q'_1} \cap Z^{\rho}_{j_2,q'_2} = \varnothing$ for $(j_1,q'_1) \neq (j_2,q'_2)$. Thus for $(i_1,q''_1,j_1,q'_1) \neq (i_2,q''_2,j_2,q'_2)$ it holds that $X_{i_1,q''_1,j_1,q'_1} \cap X_{i_2,q''_2,j_2,q'_2} = \varnothing$.

By (5.19), the union of all these sets is

$$\begin{aligned}\bigcup_{i,q'',j,q'} X_{i,q'',j,q'} &= \bigcup_{i,q'',j,q'} Z^{\lambda}_{i,q''} \cap Z^{\rho}_{j,q'} \\ &= \Big(\bigcup_{i,q''} Z^{\lambda}_{i,q''}\Big) \cap \Big(\bigcup_{j,q'} Z^{\rho}_{j,q'}\Big) \ ,\end{aligned}$$

and using the values of both unions given by Claim 5.7 and Claim 5.11, this can be calculated as follows:

$$\begin{aligned}
&(\mathtt{1}\Sigma_k(\Sigma_k^+ \setminus (k-1)^*)\mathtt{10}^*)_k \cap (\mathtt{1}((\Sigma_k^+ \setminus \mathtt{0}^*)\Sigma_k \boxminus \mathtt{1})\mathtt{10}^*)_k \\
&= (\mathtt{1}(\Sigma_k(\Sigma_k^+ \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{10}^*)_k \cap (\mathtt{1}((\Sigma_k^+ \setminus \mathtt{0}^*)\Sigma_k \boxminus \mathtt{1})\mathtt{10}^*)_k \\
&= (\mathtt{1}((\Sigma_k^+ \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{10}^*)_k \setminus (\mathtt{1}((\Sigma_k\mathtt{0}^* \cup \mathtt{0}^*\Sigma_k) \boxminus \mathtt{1})\mathtt{10}^*)_k \\
&= (\mathtt{1}((\Sigma_k^+ \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{10}^*)_k \setminus \textstyle\bigcup_{q \in Q} R_q \quad ,
\end{aligned}$$

which concludes the proof. ☐

Since the new equations represent the subexpressions of the original system, the value of the least solution of the common variables (i.e., X_q) remains the same, that is $X_q = S_q$. Moreover, Claim 5.3 asserts that the sets S_q are pairwise disjoint and that their union is a set with a regular notation. Thus the only thing remaining to be checked is that there are no cyclic chain dependencies in the defined system.

Claim 5.13. *There are no cyclic chain dependencies in the equations* (5.8)–(5.20)*.*

Proof. The constructed system contains the following chain dependencies:

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

Consider the following groups of variables:

$$\begin{aligned}
\mathcal{G}_1 &= \{U^\lambda_{i',q}, U^\rho_{j',q} \mid q \in Q;\ i', j'' \in \Sigma_k\} \ , \\
\mathcal{G}_2 &= \{X_q \mid q \in Q\} \ , \\
\mathcal{G}_3 &= \{X_{i,q'',j,q'} \mid q', q'' \in Q;\ i, j \in \Sigma_k\} \ , \\
\mathcal{G}_4 &= \{Z^\lambda_{i,q}, Z^\rho_{j,q} \mid q \in Q;\ i, j \in \Sigma_k\} \ , \\
\mathcal{G}_5 &= \{Y^\lambda_{i,i',q}, Y^\rho_{j,j',q} \mid q \in Q;\ i, j, i', j' \in \Sigma_k\} \ .
\end{aligned}$$

Then it can be easily seen that if a variable from a group $\mathcal{G}_m$ depends from a variable in a group $\mathcal{G}_n$, then $m < n$. Therefore, there are no chain dependencies in the system. ☐

According to the Claims 5.2–5.13, there exists a resolved system of equations satisfying the assumption of Lemma 5.1 and Lemma 5.2, such that one of the components in its least solution is

$$(\mathtt{1}(L_M(q) \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k = \{(\mathtt{1}w\mathtt{1}\mathtt{0}^\ell)_k \mid \ell \geq 0, w \notin (k-1)^*, w \boxplus \mathtt{1} \in L_M(q)\} \ .$$

Then, by the aforementioned lemmata, there exist unresolved systems either with union and addition, or with intersection and addition, which have the same unique solution. Finally, using Theorem 5.2, regular constants used in these systems are replaced by singleton constants, which completes the proof of Lemma 5.5. □

The next task is to represent the set $(\mathtt{1}L_M(q))_k$ as a unique solution of systems of equations. for any trellis automaton M and its state q. Similarly to Lemma 5.5, this will be done by transforming an existing construction introduced in Lemma 4.4.

Lemma 5.6. *For every $k \geq 5$ and for every trellis automaton M over 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 singleton constants, such that its unique solution contains a component $(\mathtt{1}L_M(q))_k$ for each state q of this automaton.*

Proof. The argument will use a simple technical claim, similar to Claim 5.2 from the proof of Lemma 5.5.

Claim 5.14. *Let $x \in \Sigma_k^* \setminus \mathtt{0}\Sigma_k^*$ and $y \in \Sigma_k^*$ be a string of digits (possibly empty), let $K_1, \ldots, K_m \subseteq \Sigma_k^+$ be any pairwise disjoint languages, and let $S_1, \ldots, S_m$ be sets of numbers defined by*

$$S_t = \{(xuy)_k \mid u \in K_t\} \ .$$

Then these sets are pairwise disjoint and their union is

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

The proof is nearly obvious and is omitted. A stronger statement will be proved in the following as Claim 5.17.

Consider the trellis automaton M over Σ_k. For every state q and for every digit $j \in \Sigma_k$, construct a trellis automaton $M_{q,j}$ with the set of states $Q_{q,j}$ recognizing the language $L_M(q)\{j\}^{-1}$, by the (effective) closure properties of trellis languages. Then, by Lemma 5.5, 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, in which $Y_{q,j,p} = (\mathtt{1}((L_{M_{q,j}}(p) \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k$.

The first goal is to combine these systems into a larger system of equations containing variables $Y_{q,j}$ for each state q of M and for each digit j, so that it has $Y_{q,j} = (\mathtt{1}((L(M_{q,j}) \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{10}^*)_k$ in its unique solution.

When union and addition are allowed, the construction is immediate: if $F_{q,j}$ is the set of accepting states of $M_{q,j}$, then

$$Y_{q,j} = \bigcup_{p \in F_{q,j}} Y_{q,j,p} \tag{5.21}$$

merged with subsystems defining $Y_{q,j,p}$ satisfies the goal.

If the allowed operations are intersection and addition, then the following system is constructed:

$$Y_{q,j} \cap Y_{q,j,p} = \varnothing \qquad , \text{ for } p \notin F_{q,j} \ , \tag{5.22}$$

$$Y_{q,j} \cap Y_{q,j,p} = Y_{q,j,p} \qquad , \text{ for } p \in F_{q,j} \ , \tag{5.23}$$

$$Y_{q,j} \cap \left[\mathbb{N} \setminus (\mathtt{1}((\Sigma_k^* \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{10}^*)_k\right] = \varnothing \ , \tag{5.24}$$

where the variables $Y_{q,j,p}$ are defined in subsystems. As the sets $\{(\mathtt{1}((L_{M_{q,j}}(p) \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{10}^*)_k\}_{p \in Q_{q,j}}$ together with $\mathbb{N} \setminus (\mathtt{1}((\Sigma_k^+ \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{10}^*)_k$ form a partition of natural numbers, these equations effectively represent the union of $Y_{q,j,p}$ over all p. The additional constant $(\mathtt{1}((\Sigma_k^* \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{10}^*)_k$ used in the construction is a set of numbers with a regular base-k notation, and hence it can be expressed by Theorem 5.2.

The sets $(\mathtt{1}(((L_M(q)\{j\}^{-1}) \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{10}^*)_k$ are used in Lemma 4.4 to represent the set $(\mathtt{1} \cdot L_M(q))_k$. This equation is of the form

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

which uses the constant $C_q = (\mathtt{1}L_M(q))_k \cap (\mathtt{10}^*\Sigma_k)_k$ with a regular base-k notation. These constants are similar to the constants R_q in Lemma 5.5, in the sense that they represent strings of digits of a simple form not handled by the main formula. This equation also refers to variables $Y_{q,j}$ defined in their own subsystems, so that their least solution satisfies

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

By (4.12) this equation, together with the aforementioned subsystems for variables $Y_{q,j}$, has a least solution with

$$Z_q = (\mathtt{1} \cdot L_M(q))_k \ .$$

Then, by Proposition 2.1, the equation for Z_q with variables $Y_{q,j}$ replaced by constants $Y_{q,j} = (\mathtt{1}(((L_M(q)\{j\}^{-1}) \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{10}^*)_k$ has the least solution with $Z_q = (\mathtt{1} \cdot L_M(q))_k$. The equations for Z_q for all $q \in Q$ can be turned into a system satisfying the assumption of Lemma 5.1 and Lemma 5.2 by introducing new variables $Z_{q,j}$ and rewriting the equations as:

$$\begin{aligned} Z_{q,j} &= Y_{q,j} \cap (\mathtt{1}\Sigma_k^*\mathtt{1})_k \ , \\ Z_q &= C_q \cup \bigcup_{j=0}^{k-1} Z_{q,j} + (\mathtt{1}j \boxminus \mathtt{1})_k \ , \end{aligned}$$

The grouping of variables required by Lemmata 5.1 and 5.2 is

$$\{Z_q \mid q \in Q\}, \quad \{Z_{q,j} \mid q \in Q\}_{j \in \Sigma_k} \ .$$

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

Claim 5.15. *Consider the least solution of the system defining $Z_{q,j}$ and Z_q. Then for every $j \in \Sigma_k$ and $q_1 \neq q_2$, the sets $Z_{q_1,j}$ and $Z_{q_2,j}$ are disjoint and*

$$\bigcup_q Z_{q,j} = (\mathtt{1}(\Sigma_k^* \setminus (k-1)^*)\mathtt{1})_k \ .$$

Proof. The value of $Z_{q,j}$ is determined from its equation as follows:

$$\begin{aligned} Z_{q,j} &= Y_{q,j} \cap (\mathtt{1}\Sigma_k^*\mathtt{1})_k \\ &= (\mathtt{1}(((L_M(q)\{j\}^{-1}) \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^*)_k \cap (\mathtt{1}\Sigma_k^*\mathtt{1})_k \\ &= (\mathtt{1}(((L_M(q)\{j\}^{-1}) \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{1}\mathtt{0}^* \cap \mathtt{1}\Sigma_k^*\mathtt{1})_k \\ &= (\mathtt{1}(((L_M(q)\{j\}^{-1}) \setminus \mathtt{0}^*) \boxminus \mathtt{1})\mathtt{1})_k \ . \end{aligned}$$

Fix any digit j. The sets $Z_{q,j}$ satisfy the assumption of Claim 5.14 with $K_q = ((L_M(q)\{j\}^{-1}) \setminus \mathtt{0}^*) \boxminus \mathtt{1}$ and $x = y = \mathtt{1}$. For $q_1 \neq q_2$ the languages $L_M(q_1)$ and $L_M(q_2)$ are disjoint, and hence the sets also $K_{q_1} = ((L_M(q_1)\{j\}^{-1}) \setminus \mathtt{0}^*) \boxminus \mathtt{1}$ and $K_{q_2} = ((L_M(q_2)\{j\}^{-1}) \setminus \mathtt{0}^*) \boxminus \mathtt{1}$ are disjoint as well. Thus $Z_{q_1,j} \cap Z_{q_2,j} = \varnothing$ by Claim 5.14. Also

$$\bigcup_q Z_{q,j} = (\mathtt{1}(\textstyle\bigcup_q K_q)\mathtt{1})_k = (\mathtt{1}\Sigma_k^+ \setminus (k-1)^*\mathtt{1})_k \ ,$$

since every string not in $(k-1)^*$ belongs to some K_q. ☐

Claim 5.16. *Consider the least solution of the system defining $Z_{q,j}$ and Z_q. For all $q_1 \neq q_2$, the sets Z_{q_1} and Z_{q_2} are disjoint and*

$$\bigcup_q Z_q = (\mathtt{1}\Sigma_k^+)_k \ .$$

Proof. By Proposition 2.3, the value assigned to Z_q remains the same as in the original system, i.e., $Z_q = (\mathtt{1}L_M(q))_k$. Thus Z_q's satisfy the assumption of Claim 5.14 with $K'_q = L_M(q)$, $x = \mathtt{1}$ 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 5.14, $Z_{q_1} \cap Z_{q_2} = \varnothing$ for $q_1 \neq q_2$ and

$$\bigcup_q Z_q = (\mathtt{1}(\textstyle\bigcup_{q \in Q} K'_q))_k = (\mathtt{1}\Sigma_k^+)_k \ .$$

as claimed. □

Lemma 5.1 and Lemma 5.2 require that there are no chain cyclic dependencies in the constructed system. As the only chain dependencies are those of Z_q from (some) $Z_{q,j}$, there are no cycles among them.

Therefore, the new system satisfies the assumption of the Lemma 5.1 and Lemma 5.2, and accordingly, there exists an unresolved system using addition and either union or intersection, which has a unique solution with $(\mathtt{1} \cdot L_M(q))_k$ as one of its components. The system uses regular constants, which can be eliminated using Theorem 5.2, and constants $Y_{q,j}$, which are represented in (5.21) in the case of union and addition, and in (5.22)–(5.24) using intersection and addition. □

The final step of the construction from Chapter 4 was to specify the set $(L(M))_k$ with minimal assumptions on the language $L(M)$. This step will now be similarly replicated using unresolved systems.

Theorem 5.3. *For every $k \geq 2$ and for every trellis automaton M over Σ_k, such that $L(M) \cap \mathtt{0}\Sigma_k^* = \varnothing$, 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 singleton constants, such that its unique solution contains a component $(L(M))_k$.*

Proof. First of all note that by Lemma 4.2 it is enough to consider $k \geq 9$. For smaller k, let $k' = k^4 > 9$, $L = L(M)$ and $S = (L)_k$. Consider $L' \subseteq \Sigma_{k'}^+ \setminus \mathtt{0}\Sigma_{k'}^*$ such that $(L')_{k'} = S$. By Lemma 4.2 L' is recognised by a trellis automaton.

The following slightly more complicated version of Claim 5.14 will be used in the proof:

Claim 5.17. *Let $x \in \Sigma_k^+ \setminus \mathtt{0}\Sigma_k^*$ and $y, z \in \Sigma_k^*$ be strings of digits (possibly empty), let $K_1, \ldots, K_m \subseteq \Sigma_k^+$ be any pairwise disjoint languages, and let $S_1, \ldots, S_m$ be sets of numbers defined by*

$$S_t = (x(z^{-1}K_t)y)_k \ .$$

Then these sets are pairwise disjoint and their union is

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

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 $n = (x(z^{-1}u')y)_k$ with $u' \in K_{t'}$. Clearly, z is a prefix of both u and u', that is, $u = zv$ and $u' = zv'$. Then $n = (x(z^{-1}u)y)_k = (xvy)_k$ and $n = (x(z^{-1}u')y)_k = (xv'y)_k$, and therefore $v' = v$ and $u = u'$. It is a contradiction, as K_t and $K_{t'}$ are disjoint. This proves that $S_t \cap S_{t'} = \varnothing$.

The union of these sets is

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

as desired. □

The proof of Lemma 4.5 begins with the following system of equations (4.13)–(4.14):

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

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

The system refers to the variables $Z_{i,p}$; the values of these variables are defined in their own subsystems with the solution $Z_{i,q} = (\mathtt{1}\{i\}^{-1}L_M(q))_k$.

By Claim 4.5

$$\tau_i(Z_{i,q}) = (L_M(q) \cap i\Sigma_k^+)_k \ .$$

for $i > 1$ and $q \in Q$. The system of equations formed by the above equation for T_q and the subsystems for all variables $Z_{i,p}$ has a least solution with

$$T_q = (L_M(q) \setminus \mathtt{0}^*)_k \ .$$

Consider the following decomposition of this equation:

$$\begin{aligned} U_{i,i',q} &= Z_{i,q} \cap (\mathtt{1}i'\Sigma_k^*)_k && , \text{for } i \geq 2, i' \in \Sigma_k \ , \\ W_{i,i',q} &= U_{i,i',q} + ((i-1)\mathtt{0}^*)_k \cap (ii'\Sigma_k^*)_k && , \text{for } i \geq 2, i' \in \Sigma_k \ , \\ T_q &= \bigcup_{\substack{i \geq 2 \\ i' \in \Sigma_k}} W_{i,i',q} \cup Z_{1,q} \cup (L_M(q) \cap \Sigma_k)_k \ . \end{aligned}$$

Let the set of variables be split into the following $2k-3$ groups:

$$\{U_{i,i',q} \mid i' \in \Sigma_k, q \in Q\}_{2 \leq i < k}, \ \{W_{i,i',q} \mid i, i' \in \Sigma_k, q \in Q\}_{2 \leq i < k}, \ \{T_q \mid q \in Q\} \ .$$

As in the previous proofs, it is claimed that the union of the sets from each group is a set with a regular base-k notation, and that the sets in each group are pairwise disjoint.

By (4.15)

$$U_{i,i',q} = \{(\mathtt{1}i'w)_k \mid ii'w \in L_M(q)\} \ .$$

Fix $i \geq 2$. Then the sets $\{U_{i,i',q}\}$ for all $i' \in \Sigma_k$ and q satisfy the assumption of Claim 5.17 with $K_{i',q} = ii'\Sigma_k^* \cap L_M(q)$, $x = \mathtt{1}$, $y = \varepsilon$ and $z = i$. The intersection $K_{i'_1,q_1} \cap K_{i'_2,q_2}$ is empty for $(i'_1, q_1) \neq (i'_2, q_2)$, as for $i'_1 \neq i'_2$ it holds that $\mathtt{1}i'_1\Sigma_k^* \cap \mathtt{1}i'_2\Sigma_k^* = \varnothing$, and for $q_1 \neq q_2$ it holds that $L_M(q_1) \cap L_M(q_2) = \varnothing$, as trellis automata are deterministic. Hence,

$$\begin{aligned}\bigcup_{i',q} U_{i,i',q} &= (\mathtt{1}(i^{-1}\bigcup_{i',q} K_{i',q}))_k \\ &= (\mathtt{1}(i^{-1}i\Sigma_k^+))_k \\ &= (\mathtt{1}\Sigma_k^+)_k \ ,\end{aligned}$$

for each $i \geq 2$.

Also, by (4.16)

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

Consider any two variables W_{i_1,i'_1,q_1} and W_{i_2,i'_2,q_2} with $(i_1, i'_1, q_1) \neq (i_2, i'_2, q_2)$. If $q_1 \neq q_2$, then $L_M(q_1) \cap L_M(q_2) = \varnothing$. 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 $W_{i_1,i'_1,q_1} \cap W_{i_2,i'_2,q_2} = \varnothing$. The union of these sets equals:

$$\bigcup_{i,i',q} W_{i,i',q} = \bigcup_{i,i',q} (L_M(q))_k \cap (ii'\Sigma_k^*)_k = (\Sigma_k^{\geq 2})_k \ .$$

The proof that any sets T_{q_1} and T_{q_2} with $q_1 \neq q_2$ are disjoint is immediate, as the new equations represent subexpressions of the former system, hence in the least solution $T_q = (L_M(q) \setminus \mathtt{0}^*)_k$. Thus for all $q_1 \neq q_2$

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

while the union of all these sets is

$$\bigcup_q T_q = \bigcup_q (L_M(q) \setminus \mathtt{0}^*)_k = (\Sigma_k^+ \setminus \mathtt{0}^*)_k \ .$$

The only chain dependency in the constructed system is that of T_q from $W_{i,i',q}$. Hence there are no cyclic chain dependencies and the system obtained satisfies the assumptions of Lemma 5.1 and Lemma 5.2 with constants $Z_{i,q}$ and regular constants.

Hence 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 5.6 and the latter by Theorem 5.2. ∎

Chapter 6

Completeness of systems of equations

In this chapter the computational completeness of systems of equations over sets of natural numbers using addition and *either* union *or* intersection is shown:

Theorem 6.1. *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, is exactly the family of recursive (recursively enumerable, co-recursively enumerable, respectively) sets. The same result holds for systems with intersection and addition.*

These solutions are recursive (recursively enumerable, co-recursively enumerable, respectively) because so are the solutions of language equations with union, intersection and concatenation, as asserted by Theorem 1.1. So the task is to take any recursive (recursively enumerable, co-recursively enumerable) set of numbers and to construct two systems of equations representing this set by a solution of the corresponding kind: one system with union and addition, and the other using intersection and addition.

The general plan of proving Theorem 6.1 for systems of equations is as follows. The starting point is, just as in the case of Theorem 1.1, the construction of two (linear) context-free-grammars, such that their intersection is VALC(T)—the language of transcriptions of a TM T. This is a trellis language, as the class of trellis languages includes all linear context-free languages and is closed under intersection. By Theorem 4.2, there exists a resolved system of equations using union, intersection and addition such that $(\mathrm{VALC}(T))_k$ is one of the components of the least solution. It remains to extract S_T out of VALC(T): the the actual set of numbers recognised by TM. To this end, VALC(T) is represented as an addition of two sets, with the intention that one of them is S_T and the other is a slightly modified

$(\mathrm{VALC}(T))_k$. Moreover, some additional conditions on the base-k notations of those two sets are imposed. Using the base-k notation properties of these three sets it is shown that such a representation is unique and so S_T is extracted.

The proposed plan has a little flaw, though. It uses two Boolean operations, while it was intended to use only one. In fact two operations have been used from the very beginning: in the construction of the resolved system specifying VALC(T). Nevertheless, for the clarity of presentation, Theorem 6.1 is first established in a weaker version, with the constructed systems using *both* union and intersection. This is to streamline the proof. It turns out that both union and intersection are used only twice: firstly in the construction of the language of computation of the TM, secondly in one particular equation in the construction. The former is dealt with using methods developed in Chapter 5, the latter by an ad-hoc solution in the last section of this chapter.

6.1 Sketch

A short sketch of the proof of lower bound of Theorem 1.1 is given to present its general idea; in the next sections proof of Theorem 6.1 is modelled on it. The main technical device used in the construction is the language of transcription of computation of a Turing machine, as defined and used by Hartmanis [16]. In short, for every TM T over an input alphabet Γ one can construct an alphabet $\Sigma \supseteq \Gamma$ and an encoding of computations $C_T : \Gamma^* \to \Sigma^*$, so that for every $w \in L(T)$ the string $C_T(w)$ lists the configurations of T on each step of its accepting computation on w. The language

$$\mathrm{VALC}(T) = \{C_T(w)\natural w \mid C_T(w) \text{ is an accepting computation}\} \ ,$$

where $\natural \notin \Sigma$, is an intersection of two linear context-free languages [43]. Since unresolved equations can directly simulate context-free grammars and are equipped with intersection, for every Turing machine it is relatively easy to construct a system in variables $(X_1, \dots, X_n)$ with a unique solution $(L_1, \dots, L_n)$, so that $L_1 = \mathrm{VALC}(T)$.

It remains to 'extract' $L(T)$ out of VALC(T) using a language equation. Let Y be a new variable and consider the inequality

$$\mathrm{VALC}(T) \subseteq \Sigma^*\natural Y,$$

which can be formally rewritten as an equation $X_1 \cup \Sigma^*\natural Y = \Sigma^*\natural Y$. This inequality states that for every $w \in L(T)$, the string $C_T(w)\natural Y$ should be in $\Sigma^*\natural Y$, that is, w should be in Y. This makes $L(T)$ the least solution of this inequality and proves the second part of Theorem 1.1 with respect to recursively enumerable sets and least solutions. The construction for a co-recursively enumerable set and a greatest solution is established by a dual argument, and these two constructions can be then combined to represent every recursive set [41].

At the first glance, the idea that the same result holds if the alphabet consists of a single letter sounds odd. However, this is proved in the following sections, and, moreover, the general plan of the argument remains essentially the same.

6.2 Unresolved systems with $\{\cup, \cap, +\}$

The construction for Theorem 6.1 is based upon an arithmetisation of Turing machines, which proceeds in several stages. First, valid accepting computations of a Turing machine are represented as numbers, so that these numbers could be recognised by a trellis automaton working on base-6 positional notation of these numbers, which are regarded as strings over the alphabet $\Sigma_6 = \{\mathtt{0}, \mathtt{1}, \mathtt{2}, \mathtt{3}, \mathtt{4}, \mathtt{5}\}$. There is nothing specific about base-6, simply this is the smallest number for which the construction is relatively easy to present.

While trellis automata are rather flexible and could accept many different encodings of such computations, the subsequent constructions require a set of numbers of a very specific form. This form will now be defined.

Consider the following standard encoding of computations as strings:

Definition 6.1. Let T be a Turing machine recognizing numbers given to it in base-6 notation. Let $V \supset \Sigma_6$ be its tape alphabet, let Q be its set of states, and define $\Gamma = V \cup Q \cup \{\sharp\}$. Let $S(T) \subseteq \mathbb{N}$ be the set of numbers accepted by T.

For every number $n \in S(T)$, denote the instantaneous description of T after i steps of computation on n as a string $ID_i = \alpha q a \beta \subseteq V^*QVV^*$, where T is in state q scanning $a \in \Gamma$ and the tape contains $\alpha a \beta$. Define

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

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

The language $\{\widetilde{C}_T(n) \mid n \in S(T)\} \subseteq \Gamma^*$ is an intersection of two linear context-free languages [43] and hence is recognised by a trellis automaton [11, 40]. By the known closure of trellis automata under codes [45], the language $\{C_T(n) \mid n \in S(T)\} \subseteq \Sigma_6^+$ is recognised by a trellis automaton as well.

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

Definition 6.2. Let T be a Turing machine recognizing numbers given in base-6 notation. For every $i \in \{\mathtt{1}, \mathtt{2}, \mathtt{3}, \mathtt{4}, \mathtt{5}\}$, the valid accepting computations of T on numbers $n \geq 6$ with their base-6 notation beginning with the digit i is

$$\mathrm{VALC}_i(T) = \{(C_T(n)\mathtt{1}w)_6 \mid n = (iw)_6,\ n \in S(T)\} \ .$$

The computations of T on numbers $n \in \{0, 1, 2, 3, 4, 5\}$, provided that they are accepting, are represented by the following finite set of numbers:

$$\mathrm{VALC}_0(T) = \{(C_T(n))_6 + n \mid n \in \{0, 1, 2, 3, 4, 5\} \text{ and } n \in S(T)\} \ .$$

For example, under this encoding, the accepting computation on a number $n = (\mathtt{543210})_6$ will be represented by a number $(\mathtt{30300300}\ldots\mathtt{30300143210})_6 \in \mathrm{VALC}_5(T)$, where the whole computation is encoded by blocks of digits $\mathtt{30}$ and $\mathtt{300}$, the digit $\mathtt{1}$ acts as a separator and the lowest digits $\mathtt{43210}$ represent n with its leading digit cut.

A crucial property of this encoding is that it allows simulating *concatenation of strings of digits* representing the computation and the input number, which is simulated by adding these numbers to each other. Clearly, the number representing the computation of T on $(iw)_6 \in L(T)$ is representable as a sum of $(\mathtt{1}w)_6$ and an appropriate number in $(\{\mathtt{30}, \mathtt{300}\}^*\mathtt{3000}^*)_6$. What is important is that the converse statement holds as well: that is, whenever the sum of a number $(\mathtt{1}w)_6$ and any number in $(\{\mathtt{30}, \mathtt{300}\}^*\mathtt{3000}^*)_6$ is of the form $(x\mathtt{1}u)_6$ with $x \in \{\mathtt{30}, \mathtt{300}\}^*\mathtt{300}$, the string u must be equal to w. The following lemma rules out the hypothetical possibility that the number $(x\mathtt{1}u)_6$ could be obtained in any other way.

Lemma 6.1. *Let $S \subseteq (\mathtt{1}\Sigma_6^+)_6$. Then for all strings $x \in \{\mathtt{30}, \mathtt{300}\}^*\mathtt{300}$ and $u \in \Sigma_6^+$, if*

$$(x\mathtt{1}u)_6 \in (\{\mathtt{30}, \mathtt{300}\}^*\mathtt{3000}^*)_6 + S \ ,$$

then $(\mathtt{1}u)_6 \in S$.

Proof. Let $(x\mathtt{1}u)_6 = (y\mathtt{0}^\ell)_6 + (\mathtt{1}v)_6$, where $y \in \{\mathtt{30}, \mathtt{300}\}^*\mathtt{300}$, $\ell \geq 0$ and $(\mathtt{1}v)_6 \in S$. The goal is to show that $u = v$, $x = y$ and $|\mathtt{1}v| = \ell$, that is, the only way by obtaining $(x\mathtt{1}u)_6$ is by adding $(x\mathtt{0}^{|u|+1})_6$ to $(\mathtt{1}u)_6$, and no other numbers from $(\{\mathtt{30}, \mathtt{300}\}^*\mathtt{3000}^*)_6$ and S could be used to fake this number. Depending on the number of digits in $|\mathtt{1}v|$, consider the following cases:

- $|\mathtt{1}v| < \ell$. Then $(y\mathtt{0}^\ell)_6 + (\mathtt{1}v)_6 = (y\mathtt{0}^{\ell - |\mathtt{1}v|}\mathtt{1}v)_6$, which is a number with a base-6 notation containing at least three consecutive zeroes to the left of the leftmost digit $\mathtt{1}$. Since $(x\mathtt{1}u)_6$ has two zeroes to the left of the leftmost $\mathtt{1}$, it follows that $(y\mathtt{0}^\ell)_6 + (\mathtt{1}v)_6 \neq (\mathtt{1}u)_6$, which makes this case impossible.

- $|\mathtt{1}v| = \ell$. This is the case of addition done as intended. Then $(y\mathtt{0}^\ell)_6 + (\mathtt{1}v)_6 = (y\mathtt{1}v)_6$, and thus $(y\mathtt{1}v)_6 = (x\mathtt{1}u)_6$. The leftmost instance of $\mathtt{1}$ in $(y\mathtt{1}v)_6$ and in $(x\mathtt{1}u)_6$ is at the first position of $\mathtt{1}v$ and $\mathtt{1}u$, respectively. Therefore, $y = x$ and $v = u$.

- $\ell < |\mathtt{1}v| \leq |y| + \ell$. Then the leading $\mathtt{1}$ from $\mathtt{1}v$ is at the same position as some digit of y in $y\mathtt{1}\mathtt{0}^\ell$. Let $y = y_1 i y_2$, where $|y_2| + \ell = |v|$. The digit i is either $\mathtt{0}$ or $\mathtt{3}$.

  - If $i = \mathtt{0}$, then y_1 ends with $\mathtt{3}$ or $\mathtt{30}$. The sum $(y_1 i y_2 \mathtt{0}^\ell)_6 + (\mathtt{1}v)_6$ is thus of the form $(y_1 i' z)_6$, where $i' \in \{\mathtt{1}, \mathtt{2}\}$ ($\mathtt{2}$ can appear due to a possible carry from the earlier position), and the prefix $y_1 i'$ is in $\{\mathtt{30}, \mathtt{300}\}^*\{\mathtt{31}, \mathtt{32}, \mathtt{301}, \mathtt{302}\}$. On the other hand, in $(x\mathtt{1}u)_6$, the leftmost occurrence of digits outside of $\{\mathtt{3}, \mathtt{0}\}$ must be of the form $\mathtt{3001}$.
  - If $i = \mathtt{3}$, then the sum $(y_1 i y_2 \mathtt{0}^\ell)_6 + (\mathtt{1}v)_6$ is of the form $(y_1 i' z)_6$, where $|z| = |v|$ and $i' \in \{\mathtt{4}, \mathtt{5}\}$ ($\mathtt{5}$ can appear due to a possible carry from the earlier position). Consider the leftmost digits of the numbers $(y_1 i' z)_6$ and $(x\mathtt{1}u)_6$ different from $\mathtt{0}$ and $\mathtt{3}$. For $(x\mathtt{1}u)_6$ it is $\mathtt{1}$, while for $(y_1 i' z)_6$ it is $\mathtt{4}$ or $\mathtt{5}$, and thus these numbers cannot be the same.

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

- $|\mathtt{1}v| > |y| + \ell$. Then the leading digit of $(y\mathtt{0}^\ell)_6 + (\mathtt{1}v)_6$ is $\mathtt{1}$ or $\mathtt{2}$ (due to a possible carry). As the leading digits are different, $(y\mathtt{0}^\ell)_6 + (\mathtt{1}v)_6 \neq (x\mathtt{1}u)_6$, which rules out this case.

It has thus been established that $y = x$ and $\mathtt{1}v = \mathtt{1}u$ in the only possible case, which yields the claim. ☐

A trellis automaton recognizing the base-6 notation of numbers in $\mathrm{VALC}_i(T)$, by Theorem 4.2, can be used to obtain a system of equations with union, intersection and addition representing $\mathrm{VALC}_i(T)$ as the unique solution in the positive numbers.

The system given by Theorem 4.2 is actually resolved; casting away that property and adding additional equations of the form $X \subseteq \{1, 2, 3, \ldots\}$, gives a new system with a unique solution. The co-finite constant $\{1, 2, 3, \ldots\}$ an be expressed as a unique solution of a subsystem of equations using Lemma 5.3.

This result can be proved in the following stronger form using only one Boolean operation:

Lemma 6.2. *For every TM T recognizing numbers there exists a system of equations*

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

over sets of natural numbers using union and addition (intersection and addition), such that its least solution is $(S_0, S_1, \ldots, S_5, S_6, \ldots, S_n)$ with $S_i = \mathrm{VALC}_i(T)$ for $0 \leq i \leq 5$.

Proof, although short, is deferred to the next section. At the time being the weaker for of this lemma, which asserts representability of the sets $\mathrm{VALC}_i(T)$ by equations with union, intersection and addition.

The next task is to use these sets of numbers as constants in order to construct equations representing $S(T)$. The first case to be established is the one of least solutions and recursively enumerable sets.

Lemma 6.3. *For every TM T accepting a set $S_0 \subseteq \mathbb{N}$ there exists a system of equations of the form*

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

with union and addition (or equally with intersection and addition), which has the set of solutions

$$\{ (S, f_1(S), \dots, f_m(S)) \mid S_0 \subseteq S \} ,$$

where $f_1, \dots, f_m : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ are some monotone functions on sets of numbers defined with respect to S_0. In particular, there is the least solution with $Y = S_0$.

The below argument proves a weaker form of Lemma 6.3, with the constructed system using both union and intersection. Most of the given equations are stated as inclusions of the form $V \subseteq V'$ or $V \subseteq V' + V''$, and thus can be equally expressed using union and using intersection. There is only one equation in the proof that explicitly uses both union and intersection, and it will be shown in Section 6.3 that this equation can be rephrased using either intersection (Lemma 6.6), or union (Lemma 6.7). With that correction, the below proof will establish Lemma 6.3 in its full form stated above.

Proof of Lemma 6.3 (weaker form). The proof is by constructing a system in variables $(Y, Y_1, \dots, Y_5, Y_0, X_7, \dots, X_m)$, where the number m will be determined below, and the set of solutions of this system is defined by the following conditions, which ensure that the statement of the lemma is fulfilled:

$$S(T) \cap \{0,1,2,3,4,5\} \subseteq Y_0 \subseteq \{0,1,2,3,4,5\} \ , \tag{6.1a}$$

$$\{(\mathtt{1}w)_6 \mid w \in \Sigma_6^+,\ (iw)_6 \in S(T)\} \subseteq Y_i \subseteq (\mathtt{1}\Sigma_6^+)_6, \quad \text{for } 1 \leq i \leq 5 \ , \tag{6.1b}$$

$$Y = Y_0 \cup \bigcup_{i=1}^{5} \{(iw)_6 \mid (\mathtt{1}w)_6 \in Y_i\} \ , \tag{6.1c}$$

$$X_j = K_j, \quad \text{for } 7 \leq j \leq m \ . \tag{6.1d}$$

The sets $K_7, \dots, K_m$ 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 $i \in \{1, 2, 3, 4, 5\}$, consider the above definition of $\text{VALC}_i(T)$, which can be constructed by Lemma 6.2, and define a variable Y_i with the equations

$$Y_i \subseteq (\mathtt{1}\Sigma_6^+)_6 \ , \tag{6.2a}$$

$$\text{VALC}_i(T) \subseteq (\{\mathtt{30}, \mathtt{300}\}^*\mathtt{3000}^*)_6 + Y_i \ . \tag{6.2b}$$

Both constants are given by regular languages of base-6 representations, and therefore can be specified by equations according to Theorem 5.2, using only sum and union (sum and intersection). It is claimed that this system is equivalent to (6.1b).

Suppose (6.1b) holds for Y_i. Then (6.2a) immediately follows. To check (6.2b), consider any $(C_T^i(iw)\mathtt{1}w)_6 \in \text{VALC}_i(T)$. Since this number represents the computation of T on $(iw)_6$, this implies $(iw)_6 \in S(T)$, and hence $(\mathtt{1}w)_6 \in Y_i$ by (6.1b). Then $(C_T^i(iw)\mathtt{1}w)_6 \in (\{\mathtt{30}, \mathtt{300}\}^*\mathtt{3000}^{|\mathtt{1}w|})_6 + (\mathtt{1}w)_6 \subseteq (\{\mathtt{30}, \mathtt{300}\}^*\mathtt{3000})_6 + Y_i$, which proves the inclusion (6.2b).

Conversely, assuming (6.2), it has to be proved that for every $(iw)_6 \in S(T)$, where $w \in \Sigma_6^+$, the number $(\mathtt{1}w)_6$ is in Y_i. Since $(iw)_6 \in S(T)$, there exists an accepting computation of T: $(C_T^i(iw)\mathtt{1}w)_6 \in \text{VALC}_i(T)$. Hence, $(C_T^i(iw)\mathtt{1}w)_6 \in (\{\mathtt{30}, \mathtt{300}\}^*\mathtt{3000}^*)_6 + Y_i$ due to the inclusion (6.2b), and therefore $(\mathtt{1}w)_6 \in Y_i$ by Lemma 6.1.

Define one more variable Y_0 with the equations

$$Y_0 \subseteq \{0, 1, 2, 3, 4, 5\} \ , \tag{6.3a}$$

$$\text{VALC}_0(T) \subseteq (\{\mathtt{30}, \mathtt{300}\}^*\mathtt{300})_6 + Y_0 \ . \tag{6.3b}$$

The claim is that (6.3) holds if and only if (6.1a).

Assume (6.1a). Then (6.3a) follows automatically. Consider (6.3b) and any number $(C_T(n))_6 + n \in \text{VALC}_0(T)$, where $n \in \{0, 1, 2, 3, 4, 5\}$ by definition. Then n is accepted by T, and, by (6.1a), $n \in Y_0$. As $(C_T(w))_6 + n \in (\{\mathtt{30}, \mathtt{300}\}^*\mathtt{300})_6 + n \subseteq (\{\mathtt{30}, \mathtt{300}\}^*\mathtt{300})_6 + Y_0$. This proves (6.3b).

The converse claim is that (6.3) implies that every $n \in S(T) \cap \{0, 1, 2, 3, 4, 5\}$ must be in Y_0, The corresponding $(C_T(n))_6 + n \in \text{VALC}_0(T)$ is in $(\{\mathtt{30}, \mathtt{300}\}^*\mathtt{300})_6 + n$ by (6.3b). Since n is represented by a single digit, the number $(C_T(n))_6 + n$ ends with this digit. The set $(\{\mathtt{30}, \mathtt{300}\}^*\mathtt{300})_6 + Y_0$ contains a number of such a form only if $n \in Y_0$.

Next, combine the above six systems together and add a new variable Y with the following equation:

$$Y = Y_0 \cup Y_1 \cup \bigcup_{\substack{i \in \{\mathtt{2},\mathtt{3},\mathtt{4},\mathtt{5}\} \\ i' \in \Sigma_6}} \Big((Y_i \cap (\mathtt{1}i'\Sigma_6^*)_6) + ((i-1)\mathtt{0}^*)_6 \cap (ii'\Sigma_6^*)_6 \Big) \ . \tag{6.4}$$

This equation is in fact (4.13) from Lemma 4.5, where it was proved equivalent to $Y = Y_0 \cup \{(iw)_6 \mid (\mathtt{1}w)_6 \in Y_i\}$, that is, to (6.1c). Note that this is the only equation in this proof that uses explicit union or intersection; it will be shown later, in Lemmata 6.6–6.7, that this equation can be equivalently represented using only one Boolean operation.

The final step of the construction is to express constants used in the above systems through singleton constants, which can be done by Theorem 5.2 and Lemma 6.2. The variables needed to specify these languages are denoted $(X_7, \ldots, X_n)$, and the equations for these variables have a unique solution $X_j = K_j$ for all j.

This completes the description of the set of solutions of the system. It is easy to see that there is a least solution in this set, with $Y = S(T)$, $Y_0 = S(T) \cap \{0, 1, 2, 3, 4, 5\}$, $Y_i = \{(\mathtt{1}w)_6 \mid w \in \Sigma_6^+,\ (iw)_6 \in S(T)\}$ and $X_j = K_j$. $\square$

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.

Denote the complements of the languages $\mathrm{VALC}_i(T)$ $(0 \leq i \leq 5)$ by $\mathrm{INVALC}_i(T)$. Base-6 notations of numbers in these sets are recognised by trellis automata due to the closure of trellis automata under complementation. Therefore, analogously to Lemma 6.2, the sets $\mathrm{INVALC}_i(T)$ are representable as unique solutions of systems of resolved equations.

Lemma 6.4. *For every TM T recognizing a recursively enumerable set of numbers $S_0 \subseteq \mathbb{N}$ 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 equally with intersection and addition), which has the set of solutions

$$\left\{\ (S, f_1(S), \ldots, f_m(S)) \ \middle|\ S \subseteq \overline{S_0}\ \right\}\ ,$$

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

As in the previous lemma, only a weaker form of Lemma 6.4 is proved at the moment, with the system using both union and intersection. The full version will follow by improving one of the equations in the below argument according to the later established Lemmata 6.6–6.7.

proof of weaker version of Lemma 6.4. The system has a set of variables $(Z, Z_1, \ldots, Z_5, Z_0, X_7, \ldots, X_m)$, and its set of solutions is characterised by the following conditions:

$$Z_0 \subseteq \overline{S_0} \cap \{0,1,2,3,4,5\} \ , \tag{6.5a}$$
$$Z_i \subseteq \{(\mathtt{1}w)_6 \mid w \in \Sigma_6^+,\ (iw)_6 \notin S_0\} \quad (1 \leq i \leq 5) \ , \tag{6.5b}$$
$$Z = Z_0 \cup \bigcup_{i=1}^{5} \{(iw)_6 \mid (\mathtt{1}w)_6 \in Z_i\} \ , \tag{6.5c}$$
$$X_j = K_j \quad (7 \leq j \leq n) \ . \tag{6.5d}$$

The number m and the vector of languages $(K_7, \ldots, K_m)$ will be determined below. This set of solutions will satisfy the statement of the lemma.

The equations defining the value of each Z_i $(1 \leq i \leq 5)$ are as follows:

$$Z_i \subseteq (\mathtt{1}\Sigma_6^+)_6 \ , \tag{6.6a}$$
$$(\{\mathtt{30}, \mathtt{300}\}^*\mathtt{3000}^*)_6 + Z_i \subseteq \mathrm{INVALC}_i(T) \ . \tag{6.6b}$$

It is claimed that (6.6) holds if and only if (6.5b).

If Z_i satisfies (6.5b), then (6.6a) follows immediately, and in order to prove (6.6b), one has to consider any number not in $\mathrm{INVALC}_i(T)$ and show that it is not in $(\{\mathtt{30}, \mathtt{300}\}^*\mathtt{3000}^*)_6 + Z_i$. By definition, a number is not in $\mathrm{INVALC}_i(T)$ if it is in $\mathrm{VALC}_i(T)$, so take any number $n = (iw)_6 \in S_0$, for which $(C_T(n)\mathtt{1}w)_6 \in \mathrm{VALC}_i(T)$ with $C_T(iw) \in \{\mathtt{30}, \mathtt{300}\}^*\mathtt{300}$. Suppose $(C_T(iw)\mathtt{1}w)_6 \in (\{\mathtt{30}, \mathtt{300}\}^*\mathtt{3000}^*)_6 + Z_i$. Then, by Lemma 6.1, $(\mathtt{1}w)_6 \in Z_i$, hence $(iw)_6 \notin S_0$ by (6.5b), which yields a contradiction.

The converse is established as follows. Assuming (6.6), consider any number $n \in S_0$ and let $n = (iw)_6$ for some $i \in \{1,2,3,4,5\}$ and $w \in \Sigma_6^+$. It is sufficient to prove that $(\mathtt{1}w)_6 \notin Z_i$. Suppose $(\mathtt{1}w)_6 \in Z_i$, then $(C_T(n)w)_6 \in (\{\mathtt{30}, \mathtt{300}\}^*\mathtt{3000}^*)_6 + Z_i \subseteq \mathrm{INVALC}_i(T)$ by (6.6b). However, $(C_T(n)w)_6$ is in $\mathrm{VALC}_i(T)$ and thus cannot be in $\mathrm{INVALC}_i(T)$. The contradiction obtained proves this case.

Define the following equations for the variable Z_0:

$$Z_0 \subseteq \{0,1,2,3,4,5\} \ , \tag{6.7a}$$
$$(\{\mathtt{30}, \mathtt{300}\}^*\mathtt{300})_6 + Z_0 \subseteq \mathrm{INVALC}_0(T) \ . \tag{6.7b}$$

Again, the claim is that these equations are equivalent to (6.5a).

Let Z_0 be a subset of $\{0,1,2,3,4,5\} \setminus S_0$, as stated in (6.5a). This immediately implies (6.7a). Consider any number not in $\mathrm{INVALC}_0(T)$; proving that it is not in $(\{\mathtt{30}, \mathtt{300}\}^*\mathtt{300})_6 + Z_0$ will establish (6.7b). 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 $C_T(n) + n \in (\{\mathtt{30}, \mathtt{300}\}^*\mathtt{300})_6 + Z_0$. The last digit of $C_T(n) + n$ is n, and hence $n \in Z_0$. Therefore, by (6.5a), $n \notin S_0$, which contradicts the accepting computation $C_T(n)$.

Conversely, assume (6.7) and suppose there exists $n \in \{0,1,2,3,4,5\}$, which is at the same time in S_0 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, $C_T(n) + n \in (\{\mathtt{30}, \mathtt{300}\}^*\mathtt{300})_6 + Z_0$, because $C_T(n) \in (\{\mathtt{30}, \mathtt{300}\}^*\mathtt{300})_6$ and $w \in Z_0$ by assumption, which contradicts (6.7b). The contradiction obtained proves that no such w exists, which establishes (6.5a).

The equation for Z is the same as (6.4) in Lemma 6.3:

$$Z = Z_0 \cup Z_1 \cup \bigcup_{\substack{i \in \{\mathtt{2},\mathtt{3},\mathtt{4},\mathtt{5}\} \\ i' \in \Sigma_6}} \Big((Z_i \cap (\mathtt{1}i'\Sigma_6^*)_6) + ((i-1)\mathtt{0}^*)_6 \cap (ii'\Sigma_6^*)_6 \Big) \ . \quad (6.8)$$

As in Lemma 6.3, this equation is equivalent to (6.5c). Again, this is the only equation using union and intersection, which will later be replaced by simpler equations given in Lemmata 6.6–6.7.

To conclude the proof, trellis constants are expressed by Theorem 5.3, using extra variables $(X_7, \ldots, X_n)$ and only one boolean operation. The set of solutions has been described, and, clearly, the greatest of them is $Z_0 = \overline{S_0} \cap \{0, 1, 2, 3, 4, 5\}$, $Z_i = \{(\mathtt{1}w)_6 \mid w \in \Sigma_6^+,\ (iw)_6 \notin S_0\}$, $Z = \overline{S_0}$. ☐

Finally, the case of recursive languages and unique solutions can be established by combining the constructions of Lemmata 6.3 and 6.4 as follows:

Lemma 6.5. *For every TM T halting on every input and recognizing 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 and addition, such that its unique solution is $Y = Z = S$, $X_i = K_i$, where $(K_1, \ldots, K_n)$ is some vector of sets.*

Proof. A TM T' recognizing $\overline{S}$ is easily constructed out of T. Then Lemma 6.3 is applied to T and Lemma 6.4 is applied to T'. Consider both systems of language equations given by these lemmata, let Y be the variable from Lemma 6.3, let Z be the variable from Lemma 6.4, and let $X_1, \ldots, X_n$ be the rest of the variables in these systems combined. The set of solutions of the system obtained is

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

that is

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

Adding one more equation

$$Y = Z$$

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

$$\big\{ (S, S, f_1(S, S), \ldots, f_n(S, S)) \big\} \ ,$$

which completes the proof. ☐

The weaker form of the above three lemmata yields a weaker form of Theorem 6.1, which asserts computational completeness of equations with union, intersection and addition.

6.3 Equations with $\cup$, + or $\cap$, +

We now refine the construction of the previous section so that only one boolean operation is used.

Firstly we give a proof of Lemma 6.2 and then show how to remake the construction.

proof of Lemma 6.2. As already stated, VALC(T) is an intersection of two linear context-free grammars [43]. Although VALC$_i$(T)'s are slightly modified versions of VALC(T):

$$\mathrm{VALC}(T) = \{C_T(w)\natural w \mid w \in S(T)\} \ ,$$

as compared to

$$\mathrm{VALC}_i(T) = \{(C_T(iw)\mathtt{1}w)_6 \mid n = (iw)_6,\ n \in S(T)\} \ ,$$

it can be easily verified that the original construction of VALC(T) can be improved to generate VALC$_i$(T). Then VALC$_i$(T) is an intersection of linear context-free grammars, and so it is recognised by some trellis automaton.

Theorem 5.3 assures that $(\mathrm{VALC}_i(T))_6$ is the unique solution of a system of equations using union and sum (or intersection and sum). $\square$

The next (and final) step of the argument is to modify the systems defined in the proofs of Lemmata 6.3 and 6.4 to use these sets of operations.

The only equations using Boolean operations in those proofs are (6.4) and (6.8), and since they are identical, it is sufficient to rephrase a single equation (6.4). Its reformulation using addition and intersection is immediate:

Lemma 6.6. *Let $Y_i \subseteq (\mathtt{1}\Sigma_6^+)_6$ for $1 \le i \le 5$ and let $Y_0 \subseteq \{0,1,2,3,4,5\}$. Then, for every set $Y \subseteq \mathbb{N}$,*

$$Y = Y_0 \cup Y_1 \cup \bigcup_{\substack{i\in\{\mathtt{2},\mathtt{3},\mathtt{4},\mathtt{5}\}\\ j\in\Sigma_6}} \Big((Y_i \cap (\mathtt{1}j\Sigma_6^*)_6) + ((i-1)\mathtt{0}^*)_6 \cap (ij\Sigma_6^*)_6\Big) \ . \quad (6.9)$$

if and only if

$$Y \cap (ij\Sigma_6^*)_6 = (Y_i \cap (\mathtt{1}j\Sigma_6^*)_6) + ((i-1)\mathtt{0}^*)_6 \ \cap \ (ij\Sigma_6^*)_6, \ \textit{for } i,j \in \Sigma_6,\ i > \mathtt{1} \ ,$$
$$Y_0 = Y \cap \{0,1,2,3,4,5\} \ ,$$
$$Y_1 = Y \cap (\mathtt{1}\Sigma_6^+)_6 \ .$$

Proof. $\Leftarrow$ Assume that the sets Y_i satisfy the latter three equations. Then, since $\mathbb{N} = \{0, \dots, 5\} \cup (\mathtt{1}\Sigma_6^+)_6 \cup \bigcup_{i>1,j} (ij\Sigma_6^*)_6$,

$$\begin{aligned}
Y &= Y \cap \mathbb{N} \\
&= Y \cap (\{0, \dots, 5\} \cup (\mathtt{1}\Sigma_6^+)_6) \\
&= (Y \cap \{0, \dots, 5\}) \cup (Y \cap (\mathtt{1}\Sigma_6^+)_6) \cup \bigcup_{i>1,j} (Y \cap (ij\Sigma_6^*)_6) \\
&= Y_0 \cup Y_1 \cup \bigcup_{\substack{i \in \{\mathtt{2},\mathtt{3},\mathtt{4},\mathtt{5}\} \\ j \in \Sigma_6}} \Big((Y_i \cap (\mathtt{1}j\Sigma_6^*)_6) + ((i-1)\mathtt{0}^*)_6 \ \cap \ (ij\Sigma_6^*)_6 \Big) \ .
\end{aligned}$$

$\Rightarrow$ Conversely, assume that (6.9) holds. Then, intersecting both sides of (6.9) with $(ij\Sigma_6^*)_6$, $\{0, \dots 5\}$ and $(\mathtt{1}\Sigma_6^+)_6$, one obtains:

$$\begin{aligned}
Y \cap (ij\Sigma_6^*)_6 &= (Y_i \cap (\mathtt{1}j\Sigma_6^*)_6) + ((i-1)\mathtt{0}^*)_6 \cap (ij\Sigma_6^*)_6 \ , \\
Y \cap \{0, \dots 5\} &= Y_0 \ , \\
Y \cap (\mathtt{1}\Sigma_6^+)_6 &= Y_1 \ .
\end{aligned}$$

∎

An analogous result for addition and union requires introducing new variables, and so the statement looks more complicated:

Lemma 6.7. *There exist monotone functions $f_{i,j}, g_{i,j}, h_{i,j} : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$, with $i \in \{2, \dots, 5\}$ and $j \in \{0, \dots, 5\}$ and a system of equations in variables $\{Y, Y_0, \dots, Y_5\} \cup \{Y_{i,j}, Y'_{i,j}, Y''_{i,j} \mid 2 \leq i \leq 5,\ 0 \leq j \leq 5\}$ using the operations of union and addition, such that the two following conditions are equivalent*

1. *$Y = S$, $Y_i = S_i$, $Y_{i,j} = S_{i,j}$, $Y'_{i,j} = S'_{i,j}$, $Y''_{i,j} = S''_{i,j}$ with $i \in \{2, \dots, 5\}$ and $j \in \{0, \dots, 5\}$ is a solution of that system*
2. *$S_0 \subseteq \{0, 1, 2, 3, 4, 5\}$, $S_1, S_2, S_3, S_4, S_5 \subseteq (\mathtt{1}\Sigma_6^+)_6$, and $Y = S$, $Y_i = S_i$ is a solution of the equation*

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

and $S_{i,j} = f_{i,j}(S_i)$, $S'_{i,j} = g_{i,j}(S_i)$ and $S''_{i,j} = h_{i,j}(S_i)$ for $i \in \{2, \dots, 5\}$ and $j \in \{0, \dots, 5\}$.

Proof. Define

$$\begin{aligned}
f_{i,j}(X) &= X \cap (\mathtt{1}j\Sigma_6^*)_6 \ , \\
g_{i,j}(X) &= f_{i,j}(X) + ((i-1)\mathtt{0}^*)_6 \ \cap \ (ij\Sigma_6^*)_6 \ , \\
h_{i,j}(X) &= f_{i,j}(X) + ((i-1)\mathtt{0}^*)_6 \ \cap \ (\Sigma_6^* \setminus ij\Sigma_6^*)_6 \ .
\end{aligned}$$

Note that these are monotone functions. The system of equations is constructed as follows:

$$Y = Y_0 \cup Y_1 \cup \bigcup_{i,j} Y'_{i,j} \;, \tag{6.10}$$

$$Y_0 \subseteq \{0,1,2,3,4,5\} \;, \tag{6.11a}$$
$$Y_1 \subseteq (\mathtt{1}\Sigma_6^+)_6 \;, \tag{6.11b}$$

$$\bigcup_{j=0}^{5} Y_{i,j} = Y_i, \qquad \text{for } 2 \le i \le 5 \;, \tag{6.12a}$$
$$Y_{i,j} \subseteq (\mathtt{1}j\Sigma_6^*)_k, \qquad \text{for } 2 \le i \le 5; 0 \le j \le 5 \;, \tag{6.12b}$$

$$Y'_{i,j} \subseteq Y_{i,j} + ((i-1)\mathtt{0}^*)_6, \qquad \text{for } 2 \le i \le 5; 0 \le j \le 5 \;, \tag{6.13a}$$
$$Y'_{i,j} \subseteq (ij\Sigma_6^*)_6, \qquad \text{for } 2 \le i \le 5; 0 \le j \le 5 \;, \tag{6.13b}$$
$$Y''_{i,j} \subseteq Y_{i,j} + ((i-1)\mathtt{0}^*)_6, \qquad \text{for } 2 \le i \le 5; 0 \le j \le 5 \;, \tag{6.13c}$$
$$Y''_{i,j} \subseteq (\Sigma_6^* \setminus ij\Sigma_6^*)_6, \qquad \text{for } 2 \le i \le 5; 0 \le j \le 5 \;, \tag{6.13d}$$
$$Y'_{i,j} \cup Y''_{i,j} = Y_{i,j} + ((i-1)\mathtt{0}^*)_6, \qquad \text{for } 2 \le i \le 5; 0 \le j \le 5 \;. \tag{6.13e}$$

The statement of the lemma is proved separately in two directions.

$\Rightarrow$ Suppose $(S, S_0 \dots, S_5, \dots, S_{i,j}, S'_{i,j}, S''_{i,j}, \dots)$ is a solution of the system (6.10)–(6.13). Then, by (6.12b), for each $i \in \{2, \dots, 5\}$ and $j \in \{0, \dots, 5\}$,

$$S_{i,j} \subseteq (\mathtt{1}j\Sigma_6^*)_k \;,$$

and taking into account that $\bigcup_{j=0}^{5} S_{i,j} = S_i$ for $2 \le i \le 5$, by (6.12a), it follows that $S_i \subseteq (\mathtt{1}\Sigma_6^+)_k$ holds for $i = 2, \dots, 5$. The inclusions $S_0 \subseteq \{0,1,2,3,4,5\}$ and $S_1 \subseteq (\mathtt{1}\Sigma_6^+)_6$ are explicitly stated in the system as (6.11a) and (6.11b).

To see that $S_{i,j} = f_{i,j}(S_i)$, consider that, by (6.12a), $S_i = \bigcup_j S_{i,j}$, and further, by (6.12a) and (6.12b), for each j it holds that $S_{i,j} \subseteq S_i \cap (\mathtt{1}j\Sigma_6^*)_6$. Taking the union over j: $\bigcup_j S_{i,j} \subseteq \bigcup_j S_i \cap (\mathtt{1}j\Sigma_6^*)_6$. The latter is, clearly, a subset of S_i, and thus

$$S_i \overset{(6.12a)}{=} \bigcup_j S_{i,j} \overset{(6.12b)}{\subseteq} \bigcup_j S_i \cap (\mathtt{1}j\Sigma_6^*)_6 \subseteq S_i \;.$$

hence the inequalities are in fact equalities. Since $S_{i,j} \subseteq (\mathtt{1}j\Sigma_6^*)_6$ and for $j \neq j'$ the sets $(\mathtt{1}j\Sigma_6^*)_6$ and $(\mathtt{1}j'\Sigma_6^*)_6$ are disjoint, for each j it holds that $S_{i,j} = S_i \cap (\mathtt{1}j\Sigma_6^*)_6$, that is $S_{i,j} = f_{i,j}(S_i)$.

The proof of equalities $S'_{i,j} = g_{i,j}(S_i)$ and $S''_{i,j} = h_{i,j}(S_i)$ is done by a similar chain of inclusions:

$$\begin{aligned}
S_{i,j} + ((i-1)\mathtt{0}^*)_6 &\overset{(6.13e)}{=} S'_{i,j} \cup S''_{i,j} \\
&\overset{(6.13a)\text{–}(6.13d)}{\subseteq} \Big(S_{i,j} + ((i-1)\mathtt{0}^*)_6 \,\cap\, (ij\Sigma_k^*)_6\Big) \\
&\qquad \cup \Big(S_{i,j} + ((i-1)\mathtt{0}^*)_6 \,\cap\, (\Sigma_6^* \setminus ij\Sigma_k^*)_6\Big) \\
&= \Big(S_{i,j} + ((i-1)\mathtt{0}^*)_6\Big) \cap \Big((ij\Sigma_k^*)_6 \,\cup\, (\Sigma_6^* \setminus ij\Sigma_k^*)_6\Big) \\
&= S_{i,j} + ((i-1)\mathtt{0}^*)_6 \ .
\end{aligned}$$

Therefore, the inequalities turn into equalities:

$$\begin{aligned}
S'_{i,j} &= S_{i,j} + ((i-1)\mathtt{0}^*)_6 \cap (ij\Sigma_6^*)_6 \\
&= g_{i,j}(S_i) \ , \\
S''_{i,j} &= S_{i,j} + ((i-1)\mathtt{0}^*)_6 \cap (\Sigma_6^* \setminus ij\Sigma_6^*)_6 \\
&= h_{i,j}(S_i) \ .
\end{aligned}$$

Since $(S, \ldots, S_i, \ldots, S_{i,j}, \ldots, S'_{i,j}, \ldots, S''_{i,j}, \ldots)$ satisfies (6.10),

$$S = S_0 \cup S_1 \cup \bigcup_{\substack{i \in \{2,3,4,5\} \\ j \in \Sigma_6}} S'_{i,j} \ ,$$

and it can be concluded that

$$\begin{aligned}
S &= S_0 \cup S_1 \cup \bigcup_{\substack{i \in \{2,3,4,5\} \\ j \in \Sigma_6}} S'_{i,j} \\
&= S_0 \cup S_1 \cup \bigcup_{\substack{i \in \{2,3,4,5\} \\ j \in \Sigma_6}} g_{i,j}(S_i) \\
&= S_0 \cup S_1 \cup \bigcup_{\substack{i \in \{2,3,4,5\} \\ j \in \Sigma_6}} \Big((S_i \cap (\mathtt{1}j\Sigma_6^*)_6) + ((i-1)\mathtt{0}^*)_6 \cap (ij\Sigma_6^*)_6\Big) \ .
\end{aligned}$$

Hence $(S, S_0, \ldots, S_5)$ is a solution of the equation.

$\Leftarrow$ Conversely, assume that $(S, S_0 \ldots, S_5)$ with $S_0 \subseteq \{0, 1, 2, 3, 4, 5\}$, $S_1, S_2, S_3, S_4, S_5 \subseteq (\mathtt{1}\Sigma_6^+)_6$ is a solution of the former equation. To show that $(S, S_0 \ldots, S_5, \ldots, f_{i,j}(S_i), \ldots, g_{i,j}(S_i), \ldots, h_{i,j}(S_i), \ldots)$ is a solution of the latter system, these values should be substituted into (6.10)–(6.13). The equality (6.10) follows by the assumption that $(S, S_0, \ldots, S_5)$ is a solution of the original system:

$$\begin{aligned}
S_0 \cup S_1 \cup \bigcup_{i,j} g_{i,j}(S_i) &= S_0 \cup S_1 \cup \bigcup_{i,j} f_{i,j}(S_i) + ((i-1)\mathtt{0}^*)_6 \cap (ij\Sigma_6^*)_6 \\
&= S_0 \cup S_1 \cup \bigcup_{i,j} (S_i \cap (\mathtt{1}j\Sigma_6^*)_6) + ((i-1)\mathtt{0}^*)_6 \cap (ij\Sigma_6^*)_6 \\
&= S \quad .
\end{aligned}$$

The next equation, (6.11), is explicitly stated in the former system, so it holds.

For (6.12), the equality holds by the following calculations:

$$\begin{aligned}
\bigcup_j f_{i,j}(S_i) &= \bigcup_j S_i \cap (\mathtt{1}j\Sigma_6^*)_6 \\
&= S_i \cap \bigcup_j (\mathtt{1}j\Sigma_6^*)_6 \\
&= S_i \cap (\mathtt{1}j\Sigma_6^*)_6 \\
&= S_i \quad , \\
f_{i,j}(S_i) &= S_i \cap (\mathtt{1}j\Sigma_6^*)_6 \\
&\subseteq (\mathtt{1}j\Sigma_6^*)_6 \quad .
\end{aligned}$$

In the same manner, all five equations in (6.13) hold true:

$$\begin{aligned}
g_{i,j}(S_i) &= f_{i,j}(S_i) + ((i-1)\mathtt{0}^*)_6 \cap (ij\Sigma_6^*)_6 \\
&\subseteq f_{i,j}(S_i) + ((i-1)\mathtt{0}^*)_6 \quad , \\
g_{i,j}(S_i) &= f_{i,j}(S_i) + ((i-1)\mathtt{0}^*)_6 \cap (ij\Sigma_6^*)_6 \\
&\subseteq (ij\Sigma_6^*)_6 \quad , \\
h_{i,j}(S_i) &= f_{i,j}(S_i) + ((i-1)\mathtt{0}^*)_6 \cap (\Sigma_6^* \setminus ij\Sigma_6^*)_6 \\
&\subseteq f_{i,j}(S_i) + ((i-1)\mathtt{0}^*)_6 \quad , \\
h_{i,j}(S_i) &= f_{i,j}(S_i) + ((i-1)\mathtt{0}^*)_6 \cap (\Sigma_6^* \setminus ij\Sigma_6^*)_6 \\
&\subseteq (\Sigma_6^* \setminus ij\Sigma_6^*)_6 \quad , \\
g_{i,j}(S_i) \cup h_{i,j}(S_i) &= \Big(f_{i,j}(S_i) + ((i-1)\mathtt{0}^*)_6 \cap (ij\Sigma_6^*)_6\Big) \\
&\quad \cup \Big(f_{i,j}(S_i) + ((i-1)\mathtt{0}^*)_6 \cap (\Sigma_6^* \setminus ij\Sigma_6^*)_6\Big) \\
&= f_{i,j}(S_i) + ((i-1)\mathtt{0}^*)_6 \cap \Big((ij\Sigma_6^*)_6 \cup (\Sigma_6^* \setminus ij\Sigma_6^*)_6\Big) \\
&= f_{i,j}(S_i) + ((i-1)\mathtt{0}^*)_6 \cap (\Sigma_6^*)_6 \\
&= f_{i,j}(S_i) + ((i-1)\mathtt{0}^*)_6 \quad .
\end{aligned}$$

Hence the whole system holds. $\square$

Using these equivalent reformulations of equations (6.4) and (6.8), the constructions in the proofs of Lemmata 6.3 and 6.4 can be modified to use either union only or intersection only, thus proving those lemmata in their full form. This completes the proof of Theorem 6.1 in its full statement.

Chapter 7

Equations with addition only

It is already known that the systems of equations with union and sum (or intersection and sum) are computationally complete, which is the strongest possible result for such equations. Still, in case of language equations, even simpler types of equations are known to be computationally universal: for example Kunc [29] has shown that the greatest solution of the equation $XL = LX$ for a finite L can be co-recursively enumerable hard. The considered alphabet was $\{a, b\}$, and using at least two letters was essential.

In case of equations over sets of natural numbers, the set of allowed operations can be limited in a similar way: this chapter is concerned with equations over sets of numbers that use *only addition and no Boolean operations*. These are systems of equations of the form

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

in variables $(X_1, \ldots, X_n)$, where $C, D \subseteq \mathbb{N}$ are ultimately periodic constants. This is the simplest form of equations, and at the first glance it seems out of question that such equations could have any non-trivial solutions. However, it is shown that they can have not only non-periodic solutions, but in fact are computationally universal.

The idea is to take an arbitrary system using addition and union and *encode* it in another system using addition only. The solutions of the two systems are not identical, but there is a (linear) bijection between solutions based upon an encoding of sets of numbers.

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

7.1 Overview of the argument

The general idea of the construction is as follows. Define an abstract encoding, that is an injective function

$$\sigma \colon 2^{\mathbb{N}} \to 2^{\mathbb{N}} \ .$$

There are two goals to be acquired by this encoding. Firstly, it should be *verifiable by an equation*, meaning, that there is an equation such that X satisfies it if and only if $X = \sigma(\widehat{X})$ for some $\widehat{X}$. Secondly, this encoding should allow *simulating operations*, i.e., for union (addition, respectively) there is a system of equations such that $X = \sigma(\widehat{X})$, $Y = \sigma(\widehat{Y})$ and $Z = \sigma(\widehat{Z})$ satisfy this system of equations if and only if $\widehat{X} = \widehat{Y} \cup \widehat{Z}$ ($\widehat{X} = \widehat{Y} + \widehat{Z}$, respectively).

More precisely, the encoding is expected to be of the general form

$$X = \{pn + i \mid n \in \mathbb{N}, i \in I\} \cup \{pn + j \mid n \in \widehat{X}\} \cup C \ ,$$

for some fixed p, $j \in \mathbb{N}$ and I, $C \subseteq \mathbb{N}$. The exact values are given in the next section.

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

7.2 Encoding of sets

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

Definition 7.1. For each $i \in \{0, 1, \dots, 15\}$,

$$\text{TRACK}_i(S) = \{n \mid 16n + i \in S\} \ ,$$
$$\tau_i(S') = \{16n + i \mid n \in S'\} \ .$$

The subset $S \cap \{16n + i \mid n \geq 0\}$ is called the i^{th} *track of* S. A set S is said to have an *empty (full) track* i if $\text{TRACK}_i(S) = \varnothing$ ($\text{TRACK}_i(S) = \mathbb{N}$, respectively).

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

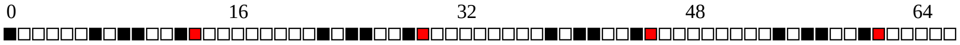

Figure 7.1: The encoding $\sigma(\widehat{S})$. The red cells represent numbers that not necessarily belong to $\sigma(\widehat{S})$. The black cells represent the numbers that certainly are in $\sigma(\widehat{S})$; the white cells represent the numbers which cannot be in $\sigma(\widehat{S})$.

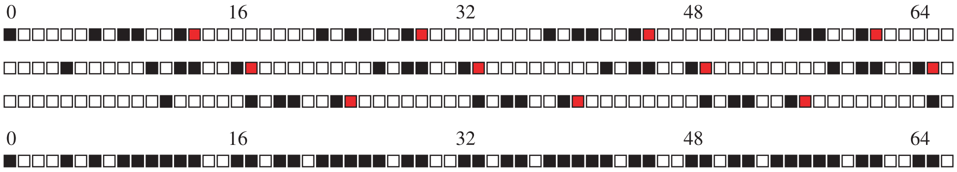

Figure 7.2: The addition of $\sigma(\widehat{X}) + \{0, 4, 11\}$. The following rows represent $\sigma(\widehat{X})$, $\sigma(\widehat{X}) + \{4\}$, $\sigma(\widehat{X}) + \{11\}$ and finally $\sigma(\widehat{X}) + \{0, 4, 11\}$. It can be seen that the last sum has only black and white cells.

Definition 7.2. For every set $\widehat{S} \subseteq \mathbb{N}$, its *encoding* is the set

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

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

Lemma 7.1. *A set $X \subseteq \mathbb{N}$ satisfies an equation*

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

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

Proof. For an illustration, see Fig. 7.2.

$\bigcirc\!\!\!\!\Rightarrow$ Let X be any set that satisfies the equation. Then $X + \{0, 4, 11\}$ has empty tracks 2, 5, 14 and 15:

$$\begin{aligned} \textsc{track}_2(X + \{0, 4, 11\}) &= \textsc{track}_5(X + \{0, 4, 11\}) \\ &= \textsc{track}_{14}(X + \{0, 4, 11\}) \\ &= \textsc{track}_{15}(X + \{0, 4, 11\}) \\ &= \varnothing \ . \end{aligned}$$

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

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

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

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

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

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

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

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

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

track is full:

$$\begin{aligned}
\mathbb{N} &= \textsc{track}_{10}(X + \{0, 4, 11\}) \\
&= \textsc{track}_{10}(X) \cup \textsc{track}_{6}(X) \cup (\textsc{track}_{15}(X) + 1) \\
&= \varnothing \cup \textsc{track}_{6}(X) \cup \varnothing \\
&= \textsc{track}_{6}(X) \ .
\end{aligned}$$

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

$\Leftarrow$ It remains to show the converse, that is, that if $X = \sigma(\widehat{X})$, then

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

Since $X = \bigcup_{i=0}^{15} \tau_i(\textsc{track}_i(X))$, then

$$\begin{aligned}
X + \{0, 4, 11\} = \Big(\bigcup_i \tau_i(\textsc{track}_i(X)) + 0\Big) \cup \Big(\bigcup_i \tau_i(\textsc{track}_i(X)) + 4\Big) \\
\cup \Big(\bigcup_i \tau_i(\textsc{track}_i(X)) + 11\Big) \ ,
\end{aligned}$$

and Table 7.1 presents the form of each particular term in this union. Each i^{th} row represents track number i in X, and each column labelled $+j$ for $j \in \{0, 4, 11\}$ corresponds to the addition of a number j. The cell (i, j) gives the set $\textsc{track}_i(X) + j$ and the number of the track in which this set appears in the result (this is track $i + j \pmod{16}$). Then each ℓ^{th} track of $X + \{0, 4, 11\}$ is obtained as a union of all the appropriate sets in the Table 7.1.

According to the table, the values of the set $\widehat{X}$ are reflected in three tracks of the sum $X + \{0, 4, 11\}$: in tracks 13, 1 and 8 (in the last two cases, with offset 1). However, at the same time the sum contains full tracks 8 and 13, as well as $\mathbb{N} + 1$ in track 1, and the contributions of $\widehat{X}$ to the sum are subsumed by these numbers, as $\tau_{13}(\widehat{X}) \subseteq \tau_{13}(\mathbb{N})$, $\tau_1(\widehat{X} + 1) \subseteq \tau_1(\mathbb{N} + 1)$ and $\tau_8(\widehat{X} + 1) \subseteq \tau_8(\mathbb{N})$. Therefore, the value of the expression does not depend on $\widehat{X}$. Taking the union of all entries of the Table 7.1 proves that $X + \{0, 4, 11\}$ equals

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

as stated in the lemma. $\square$

	$+0$	$+4$	$+11$
$0\colon \{0\}$	$0\colon \{0\}$	$4\colon \{0\}$	$11\colon \{0\}$
$6\colon \mathbb{N}$	$6\colon \mathbb{N}$	$10\colon \mathbb{N}$	$1\colon \mathbb{N}+1$
$8\colon \mathbb{N}$	$8\colon \mathbb{N}$	$12\colon \mathbb{N}$	$3\colon \mathbb{N}+1$
$9\colon \mathbb{N}$	$9\colon \mathbb{N}$	$13\colon \mathbb{N}$	$4\colon \mathbb{N}+1$
$12\colon \mathbb{N}$	$12\colon \mathbb{N}$	$0\colon \mathbb{N}+1$	$7\colon \mathbb{N}+1$
$13\colon \widehat{X}$	$13\colon \widehat{X}$	$1\colon \widehat{X}+1$	$8\colon \widehat{X}+1$

Table 7.1: Tracks in the sum $\sigma(\widehat{X}) + \{0, 4, 11\}$, only non-empty tracks of $\sigma(\widehat{X})$ are included.

7.3 Simulating operations

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

Lemma 7.2. *For all sets $X, Y, Z \subseteq \mathbb{N}$,*

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

and

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

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

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

encodes the set $Y + Z + 1$ on one of its tracks, while the contents of all other tracks do not depend on Y or on Z, see Fig. 7.3 for an example. Similarly, the sum

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

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

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

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

by the distributivity of union over addition, the sum $\sigma(Y) + \sigma(Z)$ is a union of 36 terms, each being a sum of two individual tracks. Every such sum is

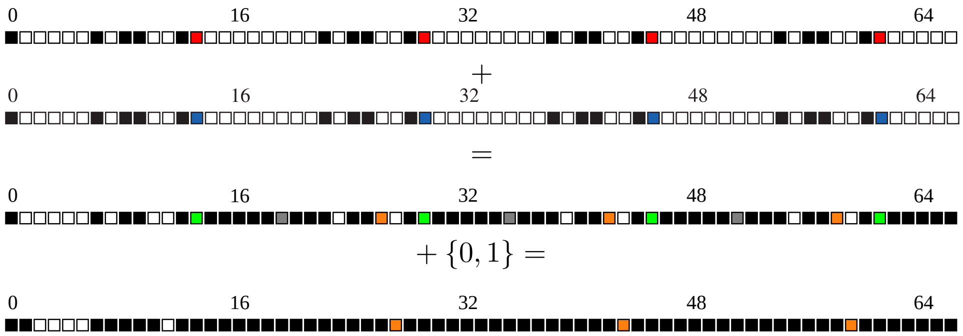

Figure 7.3: The graphical illustration of addition $\sigma(Y)+\sigma(Z)+\{0,1\}$. The first row represents $\sigma(Y)$, with red cells encoding the elements of Y. The second row represents $\sigma(Z)$, with blue cells encoding the elements of Z. In the sum $\sigma(Y)+\sigma(Z)$ the green cells are the encoded elements of $Y \cup Z$, the orange cells are the encoded elements of $\sigma(Y)+\sigma(Z)$ while the grey cells are some auxiliary "junk" numbers. Then the addition of $\{0,1\}$ leaves only encoded numbers from $\sigma(Y)+\sigma(Z)$ and the numbers not depending on Y nor Z.

contained in a single track as well, and Table 7.2 gives a case inspection of the form of all these terms. Each of its six rows corresponds to one of the nonempty tracks of $\sigma(Y)$, while its six columns refer to the nonempty tracks in $\sigma(Z)$. Then the cell gives the sum of these tracks, in the form of the track number and track contents: that is, for row representing $\textsc{track}_i(\sigma(Y))$ and for column representing $\textsc{track}_j(\sigma(Z))$, the cell (i,j) represents the set $\textsc{track}_i(\sigma(Y)) + \textsc{track}_j(\sigma(Z))$, which is bound to be on track $i+j \pmod{16}$. For example, the sum of track 8 of $\sigma(Y)$ and track 9 of $\sigma(Z)$ falls onto track $1 = 8+9 \pmod{16}$ and equals

$$\begin{aligned}\tau_8(\mathbb{N}) + \tau_9(\mathbb{N}) &= \{8+9+16(m+n) \mid m,n \geq 0\} \\ &= \{1+16n \mid n \geq 1\} \\ &= \tau_1(\mathbb{N}+1) \ ,\end{aligned}$$

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

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

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

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

Table 7.2: Tracks in the sum $\sigma(Y) + \sigma(Z)$. Question marks denote subsets of $\mathbb{N} + 1$ that depend on Y or Z and whose actual values are unimportant.

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

Table 7.3: Tracks in the sums of $\sigma(Y) + \sigma(Z)$ with constants.

Now the value of each i^{th} track of $\sigma(Y) + \sigma(Z)$ is obtained as the union of all sums in Table 7.2 that belong to the i^{th} track. The final values of these tracks are presented in the corresponding column of Table 7.3.

Now the contents of the tracks in $\sigma(Y)+\sigma(Z)+\{0,1\}$ can be completely described. The calculations are given in Table 7.3, and the result is that for all Y and Z,

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

It easily follows that

$$X = Y + Z$$

if and only if

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

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

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

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

and therefore for all X, Y, Z,

$$X = Y \cup Z$$

if and only if

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

since $X = X \cup X$.

Both claims of the lemma follow. □

7.4 Simulating a system of equations

Using the encoding defined above, it is now possible to represent a system with union and addition by a system with addition only. Since Lemma 7.2 on the simulation of individual operations is applicable only to equations of a simple form, the first task is to convert a given system to such a form:

Lemma 7.3. *For every system of equations over sets of numbers in variables $(X_1, \ldots, X_n)$ using union, addition and constants from a set $\mathcal{C}$ there exists a system in variables $(X_1, \ldots, X_n, X_{n+1}, \ldots, X_{n+m})$ with all equations of the form $X_i = X_j + X_k$, $X_i = X_j \cup X_k$ or $X_i = C$ with $C \in \mathcal{C}$, such that the set of solutions of this system is*

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

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

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

Lemma 7.4. *For every system of equations over sets of numbers in variables $(\ldots, X, \ldots)$ and with all equations of the form $X = Y + Z$, $X = Y \cup Z$ and $X = C$, there exists a system in variables $(\ldots, X', \ldots)$, using only addition and constants $\{0,1\}$, $\{0,2\}$, $\{0,4,11\}$, $\sigma(\{0\})$, $\sigma(C)$ with C used in the original system and the ultimately periodic constant from Lemma 7.1, such that $(\ldots, S'_X, \ldots)$ is a solution of the latter system if and only if $S'_X = \sigma(S_X)$ for each variable X, for some solution $(\ldots, S_X, \ldots)$ of the former system.*

Proof. The proof is by a direct transformation of this system according to Lemmata 7.1 and 7.2. First, the new system contains the following equation for each variable X':

$$X' + \{0,4,11\} = \bigcup_{\substack{i \in \{0,4,6,8,9, \\ 10,12,13\}}} \tau_i(\mathbb{N}) \cup \bigcup_{i \in \{1,3,7\}} \tau_i(\mathbb{N}+1) \cup \{11\} \ . \tag{7.1}$$

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

$$X' + \sigma(\{0\}) + \{0,1\} = Y' + Z' + \{0,1\} \tag{7.2}$$

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

$$X' + X' + \{0,2\} = Y' + Z' + \{0,2\} \ . \tag{7.3}$$

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

$$X' = \sigma(C) \ . \tag{7.4}$$

By Lemma 7.1, (7.1) ensures that each solution $(\ldots, S'_X, \ldots)$ of the constructed system satisfies $S'_X = \sigma(S_X)$ for some sets S_X. It is claimed that $(\ldots, S_X, \ldots)$ satisfies each equation of the original system if and only if $(\ldots, \sigma(S_X), \ldots)$ satisfies the corresponding equation (7.2)–7.4 of the constructed system. Consider each pair of corresponding equations:

- Consider an equation $X = Y + Z$ from the original system. Then there is a corresponding equation (7.2), and, by Lemma 7.2, $(\ldots, S_X, \ldots)$ satisfies the original equation if and only if $(\ldots, \sigma(S_X), \ldots)$ satisfies (7.2).

- Similarly, by Lemma 7.2, an equation of the form $X = Y \cup Z$ is satisfied by $(\ldots, S_X, \ldots)$ if and only if $(\ldots, \sigma(S_X), \ldots)$ satisfies the corresponding equation (7.3).

- For each equation of the form $X = C$ it is claimed that a set S_X satisfies it if and only if $\sigma(S_X)$ satisfies the corresponding equation (7.4). Indeed, $\sigma(S_X) = \sigma(C)$ if and only if $\textsc{track}_{13}(\sigma(S_X)) = \textsc{track}_{13}(\sigma(C))$, and since $\textsc{track}_{13}(\sigma(S_X)) = S_X$ and $\textsc{track}_{13}(\sigma(C)) = C$, this is equivalent to $S_X = C$.

This shows that $(\ldots, S_X, \ldots)$ satisfies the original system if and only if $(\ldots, \sigma(S_X), \ldots)$ satisfies the constructed system, which proves the correctness of the construction. □

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

Furthermore, it is important that the encoding σ respects inclusion, that is, if $X \subseteq Y$, then $\sigma(X) \subseteq \sigma(Y)$. If one solution of the original system is less than another, then the corresponding solutions of the constructed system maintain this relation. Therefore, if the original system has a least (greatest) solution then so does the new one, and its least (greatest) solution is the image of the least (greatest) solution of the original system.

These observations allow applying Lemmata 7.3 and 7.4 to encode each system in Theorem 6.1 (in case of sum and addition) within a system using addition only.

Theorem 7.1. *For every recursive (recursively enumerable, co-recursively enumerable) set $S \subseteq \mathbb{N}$ there exists a system of equations*

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

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

7.5 Systems with finite constants

The constructions above essentially use three infinite ultimately periodic constants: one of them is the right-hand side of the equation from Lemma 7.1, and the other two are the sets $\sigma(\{0\})$ and $\sigma(\{1\})$ used in Lemma 7.4 to represent constants $\{0\}$ and $\{1\}$. It will now be shown that the use of such constants is necessary, and systems using only addition and *finite* constants cannot specify any non-trivial infinite sets.

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

Lemma 7.5. *If a system of equations in variables* $(\ldots, X_j, \ldots, Y_i, \ldots)$ *using addition and only finite constants has a solution* $(\ldots, F_j, \ldots, S_i, \ldots)$*, where each* F_j *is finite and each* S_i *infinite, then* $(\ldots, F_j, \ldots, \varnothing, \ldots)$ *is a solution of this system.*

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

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

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

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

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

This completes the proof that $(\ldots, F_j, \ldots, \varnothing, \ldots)$ is a solution. □

In a similar way, infinite components of a solution can be augmented to co-finite sets. For every nonempty set $S \subseteq \mathbb{N}$, consider its *upward closure* $S + \mathbb{N}$, which is always co-finite.

Lemma 7.6. *If a system of equations in variables* $(\ldots, X_j, \ldots, Y_i, \ldots)$ *using addition and only finite constants has a solution* $(\ldots, F_j, \ldots, S_i, \ldots)$*, where each* F_j *is finite and each* S_i *infinite, then* $(\ldots, F_j, \ldots, S_i + \mathbb{N}, \ldots)$ *is a solution as well.*

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

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

If both sides evaluate to $\varnothing$ or to any finite nonempty set, these cases are handled as in Lemma 7.5.

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

Theorem 7.2. *If a system of equations using addition and finite constants has a least (greatest, unique) solution* $(\ldots, S_i, \ldots)$*, then each* S_i *is finite (finite or co-finite, finite, respectively).*

7.6 Afterthought

One may wonder, how this particular encoding was devised and what is the idea behind it. The sad truth is, that while there is an idea in using an encoding in general, there is none behind this particular encoding.

The idea that such an encoding can be made was born during the discussion on possible simplifications of the unresolved systems devised so forth. It was proposed by A. Okhotin that maybe an encoding of such a type can be constructed. The needed properties were identified and formalised. Then an exhaustive search by means of a computer programme was made, and the exact encoding used in this chapter was found. In particular, this is the encoding with the smallest number of tracks that was found, as the number of tracks was increased one by one in the search. Note, that no guarantee that such numbers actually exist was given beforehand. I must admit that I was quite sceptical about their existence until I verified their properties by hand.

Taking this into account, it would be nice to remove the element of blind luck out of this construction. So the question is, whether it is possible to devise an encoding, in which the number of tracks and their contents are specified by a means of some external combinatorial properties and not by an explicit construction (found by an exhaustive search).

Part III

Decision Problems

Each formalism defining sets of numbers or languages is judged from two main perspectives. On one hand, its expressive power is measured, i.e., how complicated and useful sets can be defined. This was addressed already in Chapter I and Chapter II. On the other hand, one wants to know the difficulty of the decision problems for such a formalism, or, informally speaking, how hard is to check the properties of a given instance of a formalism. It is natural to expect some trade-off between those two.

In this part of the thesis we focus on the decision problems for both unresolved and resolved systems of equations and even unary conjunctive grammars. All usual problems, such as emptiness, equivalence, membership etc., are considered. It is natural to expect that in all cases majority of the decision problems are hard, as these systems define a rich class of sets. It is the case—most of the results presented here will calculate the exact level of undecidability, or at least narrow down the position of the problem in the arithmetical hierarchy.

The complexity of decision problems usually decrease when some restrictions on the systems are imposed. This restrictions can be, for example, limiting the number of the equations or variables. Contrary to this intuition, it is shown that almost all considered problems are equally hard for a single equation with single variable.

This first two chapters of this parts deal with reductions of the number of variables. Afterwards only the decision problems are considered. Special place is reserved for the complexity of the membership problem for resolved equations as, on one hand, it looks as the most important problem considered in this part of the thesis. On the other hand this result is much more technically involved than the other.

Chapter 8

Single nonterminal grammars

For each formalism describing languages in formal language theory, one can introduce its *descriptional complexity*—a number of states for automata, length for the regular expressions, use of some particular operation, etc. For grammars, the number of its nonterminals is usually used. Its usefulness is proved in the theorem claiming that for each k the class of languages generated by the context-free grammars using at most $k+1$ nonterminals properly contains the class of languages generated using at most k nonterminals [15]. For the language equations, the most natural measures are the number of used variables and the amount of the equations in the system.

Already in my first paper on unary conjunctive grammars [19], the question of minimising the number of non-terminals required to describe a non-regular language was raised. In particular, it was somehow expected that one non-terminal symbol is not enough to obtain non-regular solution. It should be noted that I failed in both attempts: minimising the number of non-terminals as well as proving that only regular languages can be obtained.

Surprisingly, Okhotin and Rondogiannis [46] gave a construction of a one non-terminal unary conjunctive grammar with a non-regular solution. It will be given and described later in the chapter, it is enough to note now that it was based on encoding of the system (3.1) in one non-terminal using tracks, in a way similar to the one in the Chapter 7. Moreover, Okhotin and Rondogiannis presented two classes of conjunctive languages that are not representable by any conjunctive grammar with a single nonterminal.

The goal of this chapter is to generalise the example of a nonregular language given by Okhotin and Rondogiannis [46] to a representability theorem: for every unary conjunctive grammar the languages generated by all of its nonterminal symbols can be encoded together in a single unary language generated by a one-nonterminal conjunctive grammar. This allows proving the claims on complexity of decision problems for unary conjunctive grammars in much stronger form, i.e., even for grammars with one non-terminal.

8.1 First example

The first example of non-regular language generated by a conjunctive grammar (3.1), essentially use all four nonterminal symbols, and there seems to be no apparent way to replicate it using a single nonterminal. The same can be said also for all other systems devised so far. However, this was achieved in the following example:

Example 8.1 (Okhotin, Rondogiannis [46]). The conjunctive grammar

$$S \to a^{22}SS\&a^{11}SS \mid a^{9}SS\&aSS \mid a^{7}SS\&a^{12}SS \mid \\ a^{13}SS\&a^{14}SS \mid a^{56} \mid a^{113} \mid a^{181}$$

generates the language $\{a^{4^n-8} \mid n \geq 3\} \cup \{a^{2 \cdot 4^n-15} \mid n \geq 3\} \cup \{a^{3 \cdot 4^n-11} \mid n \geq 3\} \cup \{a^{6 \cdot 4^n-9} \mid n \geq 3\}$.

This grammar is actually derived from (3.1), and the language it generates encodes the languages of all four nonterminals of this grammar. Each of the four components in the generated language represents one of the nonterminals of (3.1) with a certain *offset* (8, 15, 11 and 9).

Note that the language of the grammar from Example 8.1 has exponential growth. At the same time, it has been proved that if a language grows faster than exponentially (for example, $\{a^{n!} \mid n \geq 1\}$), then it is not representable by univariate equations:

Proposition 8.1 (Okhotin, Rondogiannis [46]). *Let* $L = \{a^{n_1}, a^{n_2}, \dots, a^{n_i}, \dots\}$ *with* $0 \leq n_1 < n_2 < \dots < n_i < \dots$ *be an infinite language, for which* $\liminf_{i \to \infty} \frac{n_i}{n_{i+1}} = 0$. *Then* L *is not the generated by a conjunctive grammar with a single non-terminal.*

On the other hand, it is known that unary conjunctive grammars can generate a set that grows much faster: for example consider a language $L = \{1^n 2^{2^n} 3^{2^{2^n}} \mid n \in \mathbb{N}\}$. It is recognised by a trellis automata and hence $\{a^n \mid n \in (L)_k\}$ is generated by some unary conjunctive grammar.

Thus one-nonterminal conjunctive grammars are weaker in power than conjunctive grammars of an arbitrary form. However, even though one-nonterminal conjunctive grammars cannot generate *all* unary conjunctive languages, it is now demonstrated that they can represent a certain encoding of every conjunctive language.

8.2 One-nonterminal conjunctive grammars

The goal is to simulate an arbitrary conjunctive grammar over $\{a\}$ by a conjunctive grammar with a single nonterminal symbol. The construction elaborates and formalises the intuitive idea of Example 8.1, making it provably work for any grammar.

The intended encoding of languages $(L_1, \dots L_m)$ is the language of the form $L = \bigcup_{i=1}^{m} \{a^{pn-d_i} \mid a^n \in L_i\}$, for some properly chosen numbers p and $d_1, \dots, d_m$. This encoding use the idea of tracks: for a langauge $L \subseteq a^*$, its subset of the form $L \cap \{a^{pn-t} \mid n \in \mathbb{N}\}$ will be called its t^{th} *track*, where t is the *offset* of the track. The langauge L' is encoded on the t^{th} track of L if $L' = \{a^n \mid a^{pn-t} \in L\}$. Similar notation is used for sets of numbers.

Let us shortly compare this encoding with the one used in Chapter 7. First of all, the offset is of opposite sign. While it may cause some initial confusion, it is much easier to handle hear. Secondly, no tracks are filled with gadgets. Moreover, several tracks of the single non-terminal are used to encode many non-terminals of the original grammar, while in Chapter 7 only one track per variable was meaningful.

The second step towards the construction is a small refinement of the known normal form for conjunctive grammars. It is already known by Theorem 2.2 that every conjunctive language $L \subseteq \Sigma^+$ over any alphabet Σ can be generated by a conjunctive grammar in the binary normal form. The following stronger form is required by construction given later on.

Lemma 8.1. *For every conjunctive grammar $G = (\Sigma, N, P, S)$ with $\varepsilon \notin L(G)$ there exists a conjunctive grammar $G' = (\Sigma, N', P', S')$ generating the same language, in which every rule is of the form $A \to a$ with $a \in \Sigma$, or*

$$A \to B_1C_1 \& \dots \& B_nC_n \quad \textit{with } n \geq 2 \ ,$$

in which the sets $\{B_1, C_1\}$, $\dots$, $\{B_n, C_n\}$ are pairwise disjoint.

Proof. This stronger form is obtained from the ordinary binary normal form by making multiple copies of each nonterminal.

Assume, without loss of generality, that the grammar G is already in the binary normal form. If there is a non-terminating rule with no intersection, that is, $A \to \alpha$ for some nonterminal A and $\alpha \in N^2$, it can be replaced by a trivial intersection $A \to \alpha \& \alpha$.

Let m be the greatest number of conjuncts in the rules in P. Define m copies of every nonterminal: $N' = N \times \{1, \dots, m\}$. Replace every rule

$$A \to B_1C_1 \& \dots \& B_\ell C_\ell$$

with

$$(A, i) \to (B_1, 1)(C_1, 1) \& \dots \& (B_\ell, \ell)(C_\ell, \ell)$$

for all $i \in \{1, \dots, m\}$. For every rule $A \to a$ in the original grammar, define new rules $(A, i) \to a$, again for each $i \in \{1, \dots, m\}$. Let $S' = (S, 1)$ be the new start symbol. The resulting grammar generates the same language. ☐

The next theorem is the main result of this chapter.

Theorem 8.1. *For every conjunctive grammar $G = (\{a\}, \{A_1, \ldots, A_m\}, P, A_1)$ of the form given in Lemma 8.1 there exist numbers $0 < d_1 < \ldots < d_m < p$ depending only on m and a conjunctive grammar $G' = (\{a\}, \{B\}, P', B)$ generating the language $L(G') = \{a^{np-d_i} \mid 1 \leq i \leq m,\, a^n \in L_G(A_i)\}$.*

Accordingly, the corresponding equation $X = \varphi(X)$ over sets of natural numbers has a unique solution $S = \bigcup_{i=1}^m S_i$, where $S_i = \{np - d_i \mid a^n \in L_G(A_i)\}$.

Let $p = 4^{m+2}$ and let $d_i = \frac{p}{4} + 4^i$ for every nonterminal A_i. For every number $t \in \{0, \ldots, p\}$, the set $\{np - t \mid n \geq 0\}$ is called *track number* t. The goal of the construction is to represent each set S_i in the track d_i. The rest of the tracks should be empty.

For every rule $A_i \to A_{j_1} A_{k_1} \& \ldots \& A_{j_\ell} A_{k_\ell}$ in G, the new grammar G' contains the rule

$$B \to a^{d_{j_1}+d_{k_1}-d_i} BB \& \ldots \& a^{d_{j_\ell}+d_{k_\ell}-d_i} BB \ , \tag{8.1}$$

and for every rule $A_i \to a$ in G, let G' have a rule

$$B \to a^{p-d_i} \ .$$

The proof of correctness of the construction shall be done in terms of equations over sets of numbers. Consider the equation $X = \varphi(X)$ corresponding to G', with the unknown $X \subseteq \mathbb{N}$. Every "long" rule $A \to \mathscr{A}$ in G, where $\mathscr{A} = A_{j_1} A_{k_1} \& \ldots \& A_{j_\ell} A_{k_\ell}$, is represented in the new grammar by a rule (8.1), which contributes the following subexpression to φ:

$$\varphi_{i,\mathscr{A}}(S) = \bigcap_{t=1}^{\ell} (d_{j_t} + d_{k_t} - d_i) + S + S \ .$$

Altogether, the equation $X = \varphi(X)$ takes the following form:

$$X = \bigcup_{A_i \to \mathscr{A} \in P} \varphi_{i,\mathscr{A}}(X) \cup \bigcup_{A_i \to a \in P} \{p - d_i\} \ .$$

Now the task is to prove that the unique solution of this equation is $S = \bigcup_i S_i$, where $S_i = \{np - d_i \mid a^n \in L_G(A_i)\}$.

Each time X appears in the right-hand side of the equation, it is used in the context of an expression $\psi_{i,\mathscr{A}}(X) = X + X + (d_i + d_j - d_k)$. The proof of the theorem is based upon the following property of these expressions.

Lemma 8.2. *Let $i, j, k, \ell \in \{1, \ldots, m\}$ with $\{i, j\} \cap \{k, \ell\} = \varnothing$. Then*

$$(S+S+d_i+d_j) \cap (S+S+d_k+d_\ell) = (S_i+S_j+d_i+d_j) \cap (S_k+S_\ell+d_k+d_\ell) \ .$$

Proof. As addition is distributive over union and union is distributive over intersection,

$$\begin{aligned}&(S + S + d_i + d_j) \cap (S + S + d_k + d_\ell)\\ &\qquad = \bigcup_{i',j',k',\ell'} (S_{i'} + S_{j'} + d_i + d_j) \cap (S_{k'} + S_{\ell'} + d_k + d_\ell) \ .\end{aligned}$$

It is sufficient to prove that if $\{i', j'\} \neq \{i, j\}$ or $\{k', \ell'\} \neq \{k, \ell\}$, then the intersection is empty. Consider any such intersection

$$\begin{aligned}&(S_{i'} + S_{j'} + d_i + d_j) \cap (S_{k'} + S_{\ell'} + d_k + d_\ell) =\\ &\quad = \big(\{np \mid a^n \in L(A_{i'})\} - d_{i'} + \{np \mid a^n \in L(A_{j'})\} - d_{j'} + d_i + d_j\big)\\ &\qquad \cap \big(\{np \mid a^n \in L(A_{k'})\} - d_{k'} + \{np \mid a^n \in L(A_{\ell'})\} - d_{\ell'} + d_k + d_\ell\big) \ ,\end{aligned}$$

and suppose it contains any number, which must consequently be equal to $d_i + d_j - d_{i'} - d_{j'}$ modulo p and to $d_k + d_\ell - d_{k'} - d_{\ell'}$ modulo p. As each d_t satisfies $\frac{p}{4} < d_t \leq \frac{p}{2}$, both offsets are strictly between $-\frac{p}{2}$ and $\frac{p}{2}$, and therefore they must be equal to each other:

$$d_i + d_j - d_{i'} - d_{j'} = d_k + d_\ell - d_{k'} - d_{\ell'} \ .$$

Equivalently,

$$d_i + d_j + d_{k'} + d_{\ell'} = d_k + d_\ell + d_{i'} + d_{j'} \ ,$$

and since each d_t is defined as $\frac{p}{4} + 4^t$, this holds if and only if

$$4^i + 4^j + 4^{k'} + 4^{\ell'} = 4^k + 4^\ell + 4^{i'} + 4^{j'} \ .$$

Consider the largest of these eight numbers, let its value be d. Without loss of generality, assume that it is on the left-hand side. Then the left-hand side is greater than d. On the other hand, if no number on the right-hand side is d, then the sum is at most $4 \cdot \frac{d}{4} = d$. Thus at least one number on the right-hand side must be equal to d as well. Removing those two numbers and giving the same argument for the sum of 3, 2 and 1 summands yields that

$$\{d_i, d_j, d_{k'}, d_{\ell'}\} = \{d_k, d_\ell, d_{i'}, d_{j'}\} \ .$$

Then, by the assumption that $\{i, j\} \cap \{k, \ell\} = \varnothing$,

$$\{d_i, d_j\} = \{d_{i'}, d_{j'}\} \quad \text{and} \quad \{d_{k'}, d_{\ell'}\} = \{d_k, d_\ell\},$$

and since the addition is commutative, without loss of generality one can take

$$i = i', \quad j = j', \quad k = k' \quad \text{and} \quad \ell = \ell' \ .$$

Therefore,

$$\begin{aligned}(S + S + d_i + d_j) &\cap (S + S + d_k + d_\ell) \\ &= \bigcup_{i',j',k',j'} \big(S_{i'} + S_{j'} + d_i + d_j\big) \cap (S_{k'} + S_{\ell'} + d_k + d_\ell) \\ &= (S_i + S_j + d_i + d_j) \cap (S_k + S_\ell + d_k + d_\ell) \quad ,\end{aligned}$$

which completes the proof of the lemma. □

With this property established, it can be verified that every rule for every A_i in G is correctly simulated by the corresponding rule of G', and that the data from different tracks is never mixed.

Proof of Theorem 8.1. Let $P = P_1 \cup P_0$, where P_0 contains rules of the form $A_i \to a$, while P_1 consists of multiple-conjunct rules. The associated equation is strict and thus has a unique solution, so it is enough to show that S is a solution, that is,

$$S = \varphi(S) = \bigcup_{A_i \to \mathscr{A} \in P_1} \varphi_{i,\mathscr{A}}(S) \cup \bigcup_{A_i \to a \in P_0} \{p - d_i\} \quad .$$

Consider first each "long" rule of the grammar: $A_i \to \mathscr{A} \in P_1$, where $\mathscr{A} = A_{j_1} A_{k_1} \& \ldots \& A_{j_t} A_{k_t}$. Then

$$\begin{aligned}\varphi_{i,\mathscr{A}}(S) &= \\ \bigcap_{t=1}^{\ell} (d_{j_t} + d_{k_t} - d_i) + S + S &= \\ \bigcap_{t=1}^{\ell} (d_{j_t} + d_{k_t} - d_i) + S_{j_t} + S_{k_t} & \quad .\end{aligned}$$

by Lemma 8.2, and it is easy to calculate that

$$\bigcap_{t=1}^{\ell} (d_{j_t} + d_{k_t} - d_i) + S_{j_t} + S_{k_t} = \{np - d_i \mid a^n \in L_G(\mathscr{A})\} \quad ,$$

as

$$\begin{aligned}
&\bigcap_{t=1}^{\ell}(d_{j_t} + d_{k_t} - d_i) + S_{j_t} + S_{k_t} \\
&\qquad =\bigcap_{t=1}^{\ell}(d_{j_t} + d_{k_t} - d_i) + \{pn_j - d_{j_t} \mid a^{n_j} \in L(A_{j_t})\} \\
&\qquad \quad + \{pn_k - d_{k_t} \mid a^{n_k} \in L(A_{k_t})\} \\
&\qquad =\bigcap_{t=1}^{\ell}\{p(n_j + n_k) - d_i \mid a^{n_j} \in L(A_{j_t}), a^{n_k} \in L(A_{k_t})\} \\
&\qquad =\bigcap_{t=1}^{\ell}\{np - d_i \mid a^n \in L(A_{j_t} \cdot A_{k_t})\} \\
&\qquad =\{np - d_i \mid a^n \in L(\mathscr{A})\} \ .
\end{aligned}$$

Similarly, for a "short" rule $A_i \to a$,

$$\{p - d_i\} = \{np - d_i \mid a^n \in L_G(\{a\})\} \ .$$

Substituting this to the equation for $\varphi(S)$: and altogether,

$$\begin{aligned}
\varphi(S) &= \bigcup_i \left(\bigcup_{A_i \to \mathscr{A} \in P_1} \varphi_{i,\mathscr{A})}(S) \cup \bigcup_{A_i \to a \in P_0} \{p - d_i\} \right) \\
&= \bigcup_i \Big(\bigcup_{A_i \to \mathscr{A} \in P_1} \varphi_{i,\mathscr{A}}(S) \cup \bigcup_{A_i \to a \in P_0} \{p - d_i\} \Big) \\
&= \bigcup_i \Big(\bigcup_{A_i \to \mathscr{A} \in P_1} \{np - d_i \mid a^n \in L(\mathscr{A})\} \cup \{np - d_i \mid a^n \in L(a)\} \Big) \\
&= \bigcup_i \bigcup_{A_i \to \mathscr{B} \in P} \{np - d_i \mid a^n \in L(\mathscr{B})\} \\
&= \bigcup_i \{np - d_i \mid a^n \in L(A_i)\} \\
&= \bigcup_i S_i \\
&= S \ .
\end{aligned}$$

This completes the proof of Theorem 8.1. □

Chapter 9

Equation with one variable

In the previous chapter it was shown that one non-terminal unary conjunctive grammar can encode an arbitrary unary conjunctive grammar. Knowing this result it is natural to ask, how many variables and equations in unresolved system of equations are necessary to attain computational universality. In this chapter it is shown that a single equation

$$\varphi(X) = \psi(X) \ ,$$

with a unique variable X is already sufficient to represent a certain encoding of every recursive (recursively enumerable, co-recursively enumerable) set. This encoding is similar to the one used in Chapter 8, in the sense that it uses tracks of one variable to simulate several variables. Still, the details of the encoding as well as the construction of the equation are quite different and are described in full in the rest of this chapter.

For the sake of clarity, this result is presented in two stages. First it is established in the weaker form: in Section 9.1 it is shown that every system of unresolved equations can be encoded in a single univariate equation using ultimately periodic constants. This construction is improved in Section 9.2 to use singleton constants only.

9.1 Equations $\varphi(X) = \psi(X)$ with periodic constants

The first goal is to replicate Theorem 6.1 using a unique equation with a unique variable. This is achieved by taking an arbitrary system of equations with solutions of the form $X_1 = S_1, \ldots, X_m = S_m$, and constructing an equation whose solutions is the set $S = \bigcup_{i=1}^{m} \{pn - d_i \mid n \in S_i\}$, for some properly chosen numbers p and $d_1, \ldots, d_m$. This is essentially the same encoding as the one used in the Chapter 8, differing only in the choice of p and d_i's.

The encoding of multiple sets on tracks of one set requires constructing a new equation out of the old system. This equation extracts the tracks contents out of a single variable and performs the operations on them, avoiding mixing the contents of several tracks. Also it checks that all tracks of the variable that should be empty according to the encoding are indeed empty, which ensures a bijection between the solutions of the original system and the solutions of the constructed equation. Such an encoding is defined in the following theorem.

Theorem 9.1. *For every system of equations over sets of numbers* $E_j(Y_1, \ldots, Y_m) = F_j(Y_1, \ldots, Y_m)$ *with* $j \in \{1, \ldots, \ell\}$, *in which every expression* E_j *and* F_j *is of the form*

$$Y_i \cap Y_{i'} \quad \text{or} \quad Y_i \cup Y_{i'} \quad \text{or} \quad Y_i + Y_{i'} \quad \text{or} \quad \{1\},$$

there exist numbers $0 \leq d_1 < \ldots < d_m < p$ *and an equation*

$$\varphi(X) = \psi(X)$$

using singleton constants and the constant $\{kp \mid k \geq 0\}$, *such that a set* S *is its solution if and only if* $S = \bigcup_{i=1}^{m} \{kp - d_i \mid k \in S_i\}$ *for some solution* $(S_1, \ldots, S_m)$ *of the original system.*

The numbers d_i are offsets of tracks for X, and the statement of the theorem already specifies that S is split into tracks. That is, each set S_i is represented in a track $S \cap \{kp - d_i \mid k \geq 1\}$. For each variable Y_i, a unique offset d_i is assigned. Define $T = \{0, \ldots, p-1\} \setminus \{d_1, \ldots, d_m\}$; each t^{th} track of S with $t \in T$ should be empty.

The set $\varphi(S) = \psi(S)$ is as well split into tracks of its own, which are not directly related to the tracks of S. These tracks correspond to equations of the original system. A track $\{kp - e_j \mid k \geq 1\}$ of $\varphi(S) = \psi(S)$ with a unique offset e_j is assigned to an equation number j. Some additional care is taken to ensure that the designated empty tracks of S are in fact empty. This is checked in designated tracks of $\varphi(S) = \psi(S)$.

Let $p = 2(m + \ell + 3)$. Then d_i and e_j can be defined so that

- $1 < d_i < \frac{p}{2} - 1$ and $1 < e_j < \frac{p}{2} - 1$ for all i and j;
- variable offsets are greater than equation offsets, that is, $d_i > e_j$ for all i, j.

Define the following expression, which extracts the track t from X:

$$f_t(X) = \begin{cases} X \cap \{kp \mid k \geq 0\}, & \text{if} \quad t = 0 \ , \\ X \cap \big(\{kp \mid k \geq 0\} + (p - t)\big), & \text{if} \quad 1 \leq t \leq p - 1 \ . \end{cases}$$

Provided that $X \subseteq \mathbb{N}$, this definition is equivalent to

$$f_t(X) = X \cap \{kp - t \mid k \geq 0\} \ .$$

Define the encoding of the system into a single equation. First the left- and right-hand sides of each equation are to be translated. For every expression E as in the statement, define φ_E as follows:

$$\begin{aligned}
\varphi_{j,\{1\}}(X) &= \{p - e_j\} \ , \\
\varphi_{j,X_i+X_{i'}}(X) &= f_{d_i}(X) + f_{d_{i'}}(X) + (d_i + d_{i'} - e_j) \ , \\
\varphi_{j,X_i \cup X_{i'}}(X) &= \big(f_{d_i}(X) + (d_i - e_j)\big) \ \cup \ \big(f_{d_{i'}}(X) + (d_{i'} - e_j)\big) \ , \\
\varphi_{j,X_i \cap X_{i'}}(X) &= \big(f_{d_i}(X) + (d_i - e_j)\big) \ \cap \ \big(f_{d_{i'}}(X) + (d_{i'} - e_j)\big) \ .
\end{aligned}$$

Next, these translated expressions are used to define a single equation:

$$\bigcup_{j=1}^{\ell} \varphi_{j,E_j}(X) = \bigcup_{j=1}^{\ell} \varphi_{j,F_j}(X) \cup \bigcup_{t \in T} \big(f_t(X) + t + 1\big) \ . \tag{9.1}$$

Its left-hand side and the first big union on its right-hand side encode the equations of the original system, while the second big union on the right ensures that there is no “garbage” on any other track of X.

Before proceeding with the main proof, let us state some technical properties of this construction. The constructed expressions φ_{j,E_j} simulate the original expressions E_j as follows:

Lemma 9.1. *Let $S \subseteq \mathbb{N}$ and $S_1, \ldots, S_m \subseteq \mathbb{N}$ satisfy*

$$S = \bigcup_{t \in T} C_t \cup \bigcup_{i=1}^{m} \{kp - d_i \mid k \in S_i\} \ , \tag{9.2}$$

for some constants $\{C_t\}_{t \in T}$. Let E_j be as defined in Theorem 9.1, i.e., the left-hand side of equation number j in the original system.. Then

$$\varphi_{j,E_j}(S) = \{kp - e_j \mid k \in E_j(S_1, \ldots, S_m)\} \ .$$

Similar correspondence holds for F_j.

Proof. Since the definitions of E_j and F_j are similar, it is enough to consider E_j. Suppose that $E_j(X_1, \ldots, X_m) = X_i \cup X_{i'}$. The value of $\varphi_{j,E_j}(S)$ is calculated as follows:

$$\begin{aligned}
\varphi_{j,X_i \cup X_{i'}}(S) &= [f_{d_i}(S) + d_i - e_j] \cup [f_{d_{i'}}(S) + d_{i'} - e_j] \\
&= \big[(S \cap \{kp - d_i \mid k \geq 0\}) + d_i - e_j\big] \\
&\cup \big[(S \cap \{kp - d_{i'} \mid k \geq 0\}) + d_{i'} - e_j\big] \\
&= \{kp - d_i + d_i - e_j \mid k \in S_i\} \cup \{kp - d_{i'} + d_{i'} - e_j \mid k \in S_{i'}\} \\
&= \{kp - e_j \mid k \in S_i\} \cup \{kp - e_j \mid k \in S_{i'}\} \\
&= \{kp - e_j \mid k \in S_i \cup S_{i'}\} \ .
\end{aligned}$$

Similar straightforward calculations yield

$$\begin{aligned}
\varphi_{j,\{1\}}(S) &= \{p - e_j\} \ , \\
\varphi_{j,X_i \cap X_{i'}}(S) &= \{kp - e_j \mid k \in S_i \cap S_{i'}\} \ , \\
\varphi_{j,X_i + X_{i'}}(S) &= \{kp - e_j \mid k \in S_i + S_{i'}\} \ .
\end{aligned}$$

□

It follows that the equations formed from the new expressions simulate the original equations:

Lemma 9.2. *Let $S \subseteq \mathbb{N}$ and $S_1, \ldots, S_m \subseteq \mathbb{N}$ satisfy* (9.2). *Then for every j*

$$E_j(S_1, \ldots, S_m) = F_j(S_1, \ldots, S_m) \text{ if and only if } \varphi_{j,E_j}(S) = \varphi_{j,F_j}(S) \ .$$

Proof. By Lemma 9.1,

$$\varphi_{j,E_j}(S) = \{pk - e_j \mid k \in E_j(S_1, \ldots, S_m)\}$$

and

$$\varphi_{j,F_j}(S) = \{pk - e_j \mid k \in F_j(S_1, \ldots, S_m)\} \ .$$

Thus an equality

$$E_j(S_1, \ldots, S_m) = F_j(S_1, \ldots, S_m)$$

holds if and only if

$$\varphi_{j,E_j}(S) = \varphi_{j,F_j}(S) \ .$$

□

Then the original system of equations is represented by the following system of equations:

Lemma 9.3. *Let a system of equations*

$$E_j(X_1, \ldots, X_m) = F_j(X_1, \ldots, X_m) \quad , \text{for } j = 1, \ldots, \ell$$

be as in Theorem 9.1, denote $T = \{0, \ldots, p-1\} \setminus \{d_1, \ldots, d_m\}$, and let $C_t \subseteq \{kp - t \mid k \geq 0\}$, for all $t \in T$, be any constant sets. Then a system of equations

$$X \cap \{kp - t \mid k \geq 0\} = C_t \qquad , \text{for } t \in T \ , \tag{9.3}$$

$$\varphi_{j,E_j}(X) = \varphi_{j,F_j}(X) \qquad , \text{for } j = 1, \ldots, \ell \tag{9.4}$$

has a solution S if and only if

$$S = \bigcup_{t \in T} C_t \cup \bigcup_{i=1}^{m} \{kp - d_i \mid k \in S_i\} \ , \tag{9.5}$$

for some solution $(S_1, \ldots, S_m)$ of the original system.

Proof. $\Rightarrow$ Let $(S_1, \ldots, S_m)$ be a solution of the original system, construct the set S as in (9.5). Then S clearly satisfies (9.3). Then, since every j^{th} equation $E_j(X_1, \ldots, X_m) = F_j(X_1, \ldots, X_m)$ is satisfied by $(S_1, \ldots, S_m)$, it follows by Lemma 9.2 that S satisfies the j^{th} equation (9.4).

$\Leftarrow$ Suppose now that S satisfies (9.3)–(9.4). Then, for every $t \in T$,

$$S \cap \{kp - t \mid k \geq 0\} = C_t \ ,$$

by (9.3). Let $S_i = \{k \mid kp - d_i \in S\}$ for $i = 1, \ldots, m$. Then S is obtained from S_i as in (9.5). It remains to show that $(S_1, \ldots, S_m)$ is a solution of the first system, that is,

$$E_j(S_1, \ldots, S_m) = F_j(S_1, \ldots, S_m) \ .$$

for each j. This follows from Lemma 9.2, as by (9.4)

$$\varphi_{j,E_j}(X) = \varphi_{j,F_j}(X) \quad \text{, for } j = 1, \ldots, \ell$$

and by Lemma 9.2 this implies

$$E_j(S_1, \ldots, S_m) = F_j(S_1, \ldots, S_m) \quad \text{, for } j = 1, \ldots, \ell \ .$$

$\square$

Note that the actual equation (9.1) mixes the statements (9.3),(9.4) in a single equality. Showing that they are indeed equivalent yields the proof of the theorem.

Proof of Theorem 9.1. Assume that S is a solution of (9.1) and consider intersections of both sides of (9.1) with $\{kp+1 \mid k \geq 0\}$. Since by Lemma 9.1 $\varphi_{j,E_j}(S) \subseteq \{kp - j \mid k \geq 0\}$, there is $\varnothing$ on the left-hand side, and for the same reason the first big union in the right-hand side vanishes. Thus the equation turns into

$$\varnothing = \varnothing \cup \bigcup_{t \in T} \big(S \cap \{kp - t \mid k \geq 0\}\big) + t + 1 \ ,$$

and therefore

$$S \cap \{kp - t \mid k \geq 0\} = \varnothing \quad \text{, for } t \in T \ . \tag{9.6}$$

Consider the intersection of (9.1) with the set $\{kp - e_j \mid k \geq 0\}$. As by Lemma 9.1, $\varphi_{j',E}(S) \subseteq \{kp - e_{j'} \mid k \geq 1\}$ then on the left-hand side only $\varphi_{j,E_j}(S)$ remains and on the right-hand side only $\varphi_{j,F_j}(S)$, as for each $t \in T$: $f_t(S) + t + 1 \subseteq \{kp + 1 \mid k \geq 0\}$. Thus the following system of equations is obtained

$$\varphi_{j,E_j}(S) = \varphi_{j,F_j}(S) \quad \text{, for } j = 1, \ldots, \ell \ . \tag{9.7}$$

Hence every solution S also satisfies the system (9.6)–(9.7).

Conversely, consider any S satisfying both (9.6) and (9.7). Then S clearly satisfies (9.1), as it is obtained as a union of sides of (9.6) (with additional $+\, t + 1$ at both sides of the equation for t) and (9.7).

As (9.6) and (9.7) satisfy the assumptions of Lemma 9.3 with constants

$$C_t = \varnothing \quad \text{, for } t \in T \ ,$$

every solution of (9.1) is of the form

$$S = \bigcup_{i=1}^{m} \{kp - d_i \mid k \in S_i\} \ ,$$

for some solution $S_1, \ldots, S_m$ of the system

$$E_j(X_1, \ldots, X_m) = F_j(X_1, \ldots, X_m) \quad \text{, for } j = 1, \ldots, \ell \ ,$$

which completes the proof of the theorem. □

9.2 Equations $\varphi(X) = \psi(X)$ with singleton constants

The construction from the previous section will now be refined by eliminating infinite periodic constants from the equation: that is, only singleton constants will be used.

Theorem 9.2. *For every system of equations $E_j(Y_1, \ldots, Y_m) = F_j(Y_1, \ldots, Y_m)$ with $j \in \{1, \ldots, \ell\}$, in which every expression E_j and F_j are as in Theorem 9.1, there exist numbers $0 \leq d_1 < \ldots < d_m < p$ and an equation $\lambda(X) = \rho(X)$ using singleton constants, such that a set $S \subseteq \mathbb{N}$ is its solution if and only if*

$$S = \{kp, kp + \tfrac{p}{2} + 1 \mid k \geq 0\} \cup \bigcup_{i=1}^{m} \{kp - d_i \mid k \in S_i\} \ ,$$

for some solution $(S_1, \ldots, S_m)$ of the original system.

The definitions and the assumptions on d_i and e_j are as in Theorem 9.1. Similarly to previous section, denote the set of offsets of unused tracks by

$$T = \{0, \ldots, p - 1\} \setminus \{0, d_1, \ldots, d_m, \tfrac{p}{2} - 1\} \ .$$

The equation (9.1) uses only one infinite constant, $\{kp \mid k \geq 0\}$. So the first step of the construction is to encode this constant in S and define expression extracting it out of S. The former was already explicitly stated in Theorem 9.2, as S contains $\{kp, kp + \frac{p}{2} + 1 \mid k \geq 0\}$. So it is left to define the expression that extract these constants out of S. These are the following:

$$\begin{aligned}
\pi(X) &= X \cap (X + \tfrac{p}{2} - 1) \ , \\
\pi'(X) &= X \cap (X + \tfrac{p}{2} + 1) \ .
\end{aligned}$$

Indeed, from the intended form of solutions stated in the theorem, $\pi(S) = \{kp \mid k \geq 1\}$ and $\pi'(S) = \{kp - (\frac{p}{2} - 1) \mid k \geq 1\}$.

Using these subexpressions, the expressions $f_t(X)$ from the previous construction is replaced by the following:

$$f'_t(X) = \begin{cases} X \cap (\pi'(X) + \frac{p}{2} - 1 - t) & \text{, for } 0 \leq t \leq \frac{p}{2} - 1 \ , \\ \Big(X \cap (\pi(X) + p - t)\Big) \cup \Big(X \cap \{p - t\}\Big) & \text{, for } \frac{p}{2} \leq t \leq p - 1 \ . \end{cases}$$

The goal is to construct such an equation that $f'_i(S) = f_i(S)$ for each of its solutions S, which allows reusing parts of the construction and the proof from Theorem 9.1. In particular, the expressions $\varphi'_{j,E}$ are defined in the same way as $\varphi_{j,E}$ in Theorem 9.1, this time using $f'_i(X)$ instead of $f_i(X)$. Furthermore, define the following three new expressions:

$$\begin{aligned}
\psi(X) &= \{\tfrac{p}{2} + 1\} \cup (\pi'(X) + \tfrac{p}{2} - 1) \cup (\pi(X) + \tfrac{p}{2} + 1) \ , \\
\psi'(X) &= \pi(X) \cup \pi'(X) \ , \\
\theta(X) &= \bigcup_{j=1}^{\ell} \big(f'_{e_j}(X) + e_j - 1\big) \cup \bigcup_{t \in T \setminus \{e_1, \ldots, e_\ell\}} f'_t(X) \ .
\end{aligned}$$

The expressions $\psi(X)$ and $\psi'(X)$ are used to generate the sets $\{kp \mid k \geq 1\} \cup \{kp - (\frac{p}{2} - 1) \mid k \geq 1\}$. The expression $\theta(X)$ deals with the “garbage” in the same way as the second part of the right-hand side of (9.1).

Now the new equation is constructed the follows:

$$\psi(X) \cup \bigcup_{j=1}^{\ell} \big(\varphi'_{j,E_j}(X) + p\big) = \psi'(X) \cup \bigcup_{j=1}^{\ell} \big(\varphi'_{j,F_j}(X) + p\big) \cup \theta(X) \ . \qquad (9.8)$$

The main technical property of this equation is that in each of its solutions the tracks 0 and $\frac{p}{2} - 1$ are full, while all tracks besides these two and $d_1, \ldots, d_m$ are empty.

Lemma 9.4. *If S is a solution of* (9.8)*, then*

$$\begin{aligned}
\{kp \mid k \geq 0\} &\subseteq S \ , \\
\{kp - (\tfrac{p}{2} - 1) \mid k \geq 1\} &\subseteq S \ , \\
S \cap \{kp - t \mid k \geq 1\} &= \varnothing \quad \text{, for all } t \in T \ .
\end{aligned}$$

Proof. The following terminology for any deviations from this rule is adopted. For any set of numbers S_0, a number $n \in S_0$ is said to be *extra* if $n = kp - t$ for $t \in T$. A number n is *missing* if $n \notin S_0$ and $n = kp - t$ for $t \in \{0, \frac{p}{2} - 1\}$. Then it has to be proved that there cannot be any extra or missing numbers in any solution of the equation (9.8).

The proof begins with the following technical claim:

Claim 9.1. *Let $S_0 \subseteq \mathbb{N}$ be any set that has no extra numbers. Then, for every expression $E \in \{E_j, F_j\}$ in every j^{th} equation, it holds that*

$$\varphi'_{j,E}(S_0) \subseteq \{kp - e_j \mid k \geq 1\} \ .$$

Proof. The main step towards establishing the claim is showing that, under the assumptions, $\pi'(S_0) \subseteq \{kp - (\frac{p}{2} - 1) \mid k \geq 1\}$.

Consider any $n \in \pi'(S_0) = S_0 \cap (S_0 + (\frac{p}{2} + 1))$. Then $n \in S_0$ and $n' = n - \frac{p}{2} - 1 \in S_0$. Let $n = kp - t$, then, as there are no extra numbers in S_0, t and $(t + \frac{p}{2} + 1 \mod p)$ are in $\{0, d_1, \ldots, d_m, \frac{p}{2} - 1\}$, which is only possible if $t = \frac{p}{2} - 1$. Hence $n = kp - (\frac{p}{2} - 1)$ and $n' = (k-1)p$, which shows that

$$\pi'(S_0) = S_0 \cap (S_0 + (\tfrac{p}{2} + 1)) \subseteq \{kp - (\tfrac{p}{2} - 1) \mid k \geq 0\} \ .$$

Then the definition of f'_{d_i} can be expanded as

$$\begin{aligned} f'_{d_i}(S_0) &= S_0 \cap (\pi'(S_0) + \tfrac{p}{2} - 1 - d_i) \\ &\subseteq \pi'(S_0) + \tfrac{p}{2} - 1 - d_i \\ &\subseteq \{kp - (\tfrac{p}{2} - 1) \mid k \geq 1\} + \tfrac{p}{2} - 1 - d_i \\ &= \{kp - d_i \mid k \geq 1\} \ , \end{aligned}$$

and therefore, for an expression $E = X_i \cap X_{i'}$,

$$\begin{aligned} \varphi'_{j,E}(S_0) &= \big(f'_{d_i}(S_0) + d_i - e_j\big) \cap \Big(f'_{d_{i'}}(S_0) + d_{i'} - e_j\Big) \\ &\subseteq f'_{d_i}(S_0) + d_i - e_j \\ &\subseteq \{kp - d_i \mid k \geq 1\} + d_i - e_j \\ &= \{kp - e_j \mid k \geq 1\} \ . \end{aligned}$$

Similar calculations can be made for $E = X_i \cup X_{i'}$, $E = X_i + X_{i'}$ and $E = \{1\}$. ☐

Let S be any solution of the equation. The first claim is that there are no extra or missing numbers in S that are smaller than p.

Claim 9.2. *It holds that $0, \frac{p}{2} + 1 \in S$ and $p - t \notin S$ for all $t \in T$.*

Proof. Firstly the small numbers are considered, i.e., smaller than p belonging to both sides of (9.8). The properties of such numbers are used as a base for considerations of larger numbers.

Consider a number $n < p$ appearing on the left-hand side of (9.8) under the substitution $X = S$. Then $n \in \psi(S)$, as the rest of the left-hand side cannot produce any number less than p. As $\psi(S)$ is a union of three subexpressions, consider each of them. First suppose that $n \in \pi(S) + \frac{p}{2} + 1$, that is, $n - \frac{p}{2} - 1 \in \pi(S)$. Then, by the definition of π, $n - \frac{p}{2} - 1 \in S + \frac{p}{2} - 1$, and so $n \geq p$. Next, suppose that $n \in \pi'(S) + \frac{p}{2} - 1$. Then $n - \frac{p}{2} + 1 \in \pi'(S)$ and thus, by the definition of π', $n - \frac{p}{2} + 1 \in S + \frac{p}{2} + 1$, and again $n \geq p$. The only remaining possibility is $n = \frac{p}{2} + 1$.

Therefore, the only number smaller than p that appears on the right-hand side is $\frac{p}{2} + 1$. Based on this fact, it will be shown which small numbers must belong to S in order to obtain $\frac{p}{2} + 1$ on the right-hand side, and which may not belong to S, as they would produce other small numbers on the right-hand side.

First, note that the only number $n' \leq \frac{p}{2}$ that may be in S is 0, as otherwise n' would be in $S \cap (p - t)$ for some $\frac{p}{2} \leq t \leq p - 1$, which is a part of $\theta(S)$, and clearly n' does not occur on the left-hand side.

Let us now consider how the number $\frac{p}{2} + 1$ is obtained on the right-hand side of (9.8). Every number in $\varphi'_{j,F_j}(S) + p$ for any j is at least p and hence is of no concern. Consider $\theta(S)$ and suppose that $\frac{p}{2} + 1 \in \theta(S)$. Then it belongs to one of the subexpressions in $\theta(S)$.

- Suppose that $\frac{p}{2} + 1 \in f'_{e_j}(S) + e_j - 1$ for some j. Let $n' \in f'_{e_j}(S)$ be such that $\frac{p}{2} + 1 = n' + (e_j - 1)$. Then

$$\begin{aligned} n' &\in S \cap (\pi'(S) + \tfrac{p}{2} - 1 - e_j) \\ &\supseteq \pi'(S) + \tfrac{p}{2} - 1 - e_j \\ &= (S \cap (S + \tfrac{p}{2} + 1)) + \tfrac{p}{2} - 1 - e_j \\ &\supseteq S + \tfrac{p}{2} + 1 + \tfrac{p}{2} - 1 - e_j \\ &\supseteq S + p - e_j \end{aligned}$$

  and so $n' \geq p - e_j$. Hence $\frac{p}{2} - 1 = n' + (e_j - 1) \geq p - 1$, a contradiction.

- Suppose $\frac{p}{2} + 1 \in f'_t(S)$ for $t \in T \setminus \{e_1, \ldots, e_\ell\}$. According to the definition of $f'_t(S)$, there are two cases.

  - If $0 \leq t \leq \frac{p}{2} - 1$, then $\frac{p}{2} + 1 \in \pi'(S) + \frac{p}{2} - 1 - t$ and hence $t + 2 \in \pi'(S)$, which implies that $t - \frac{p}{2} + 1 \in S$. As $t \leq \frac{p}{2} - 1$ by case assumption, we obtain that $t = \frac{p}{2} - 1 \notin T$, which is not an appropriate value of t.

  - In the second case of $\frac{p}{2} \leq t \leq p - 1$, $\frac{p}{2} + 1 \in f'_t(S)$ means that $\frac{p}{2} + 1 \in \pi(S) + p - t$ or $\frac{p}{2} = p - t$. Consider the former. Then

$t \in \pi(S) + \frac{p}{2} - 1$. As t is not $\frac{p}{2} - 1$,

$$t - \tfrac{p}{2} + 1 \in \pi(S) \subset S \ ,$$

which is impossible because $1 \leq t - \frac{p}{2} + 1 \leq \frac{p}{2}$ and it was shown before that 0 is the only number smaller than $\frac{p}{2}$ that can be in S. Consider the latter, i.e., the number $\frac{p}{2} + 1 = (p - t)$ for some $t \in T \setminus \{e_1, \ldots, e_\ell\}$. This cannot hold, as $t = \frac{p}{2} - 1 \notin T$.

Thus we have shown that $\frac{p}{2} + 1 \notin \theta(S)$

The only remaining possibility is $\frac{p}{2} + 1 \in \psi'(S)$, that is, $\frac{p}{2} + 1 \in \pi(S) \cup \pi'(S)$. If

$$\tfrac{p}{2} + 1 \in \pi(S) = S \cap (S + \tfrac{p}{2} - 1) \ ,$$

then $2 \in S$, which is not the case, as it is smaller than $\frac{p}{2} + 1$. Thus $\frac{p}{2} + 1 \in \pi'(S) = S \cap (S + \frac{p}{2} + 1)$, and therefore $0, \frac{p}{2} + 1 \in S$. Hence the goal of showing that those two numbers are in S has been obtained.

The only thing left to show is that for each $t \in T$, the number $p - t$ is not in S. If $t \geq \frac{p}{2}$, this has already been proved above. Suppose that $p - t \in S$ for any $t \in T$ with $t < \frac{p}{2}$ and $t \notin \{e_1, \ldots, e_\ell\}$. As $0 \in S$ and $\frac{p}{2} + 1 \in S$, $\frac{p}{2} + 1 \in \pi'(S)$. Consequently

$$p - t \in \pi'(S) + (\tfrac{p}{2} - 1 - t)$$

and hence $p - t \in f'_t(S) \subseteq \theta(S)$ is on the right-hand side. However, there is no corresponding number on the left-hand side, which is a contradiction.

Finally, suppose $p - e_j \in S$ for some j. Then, as $0, \frac{p}{2} + 1 \in S$, $\frac{p}{2} + 1 \in \pi'(S)$ and accordingly

$$\begin{aligned} p - e_j &= (\tfrac{p}{2} + 1) + (\tfrac{p}{2} - 1 - e_j) \\ &\in \pi'(S) + \tfrac{p}{2} - 1 - e_j \\ &= f'_{e_j}(S) \ . \end{aligned}$$

Therefore, $p - e_j + e_j - 1 = p - 1 \in \theta(S)$, and thus, again, $p - 1$ must appear on the left-hand side, which is a contradiction. □

In order to show that there are no missing or extra numbers, the smallest among them is considered. Then it is proved that for every extra number there is a smaller missing one and for each extra number, there is a smaller extra one. Thus there is no such a number at all.

Claim 9.3. *Let $n \notin S$ be the least missing number. Then there exists a number $n' < n$ that is extra.*

Proof. Let n be the least missing number.

- If it is of the form $n = kp$, then $n \geq p$, since $0 \in S$ by Claim 9.2. The numbers $n - p$ and $n - \frac{p}{2} + 1$ are in S, because, by assumption, there are no missing numbers less than n. Therefore, $n - \frac{p}{2} + 1 \in \pi'(S)$ and hence $n = (n - \frac{p}{2} + 1) + \frac{p}{2} - 1 \in \psi(S)$, that is, n belongs to the left-hand side of (9.8).

- A similar analysis applies for $n = kp - \frac{p}{2} + 1$. By Claim 9.2, $n \geq \frac{3p}{2} + 1$, since there are no missing numbers less than p. The numbers $n - p$ and $n - \frac{p}{2} - 1$ must be in S, because they are smaller than the least missing number. Then $n - \frac{p}{2} - 1 \in \pi(S)$ and, accordingly, $n = (n - \frac{p}{2} - 1) + \frac{p}{2} + 1 \in \psi(S)$.

In both cases, since n appears on the left-hand side of (9.8), it should also appear on the right-hand side. Consider the subexpression in which n is obtained.

First suppose that $n \in \varphi'_{j,F_j}(S) + p$ for some j, and define the finite set $S_0 = S \cap \{n' \mid n' \leq n - p\}$. Then $n \in \varphi'_{j,F_j}(S_0) + p$, because the membership of numbers larger than $n - p$ in the argument of φ'_{j,F_j} does not influence the value of this expression. If S_0 contains an extra number n', this establishes the claim, as $n' < n$. So suppose, for the sake of a contradiction, that S_0 contains no extra numbers. Thus $\varphi'_{j,F_j}(S_0) + p \subseteq \{kp - e_j \mid k \geq 2\}$ by Claim 9.1. It follows that $n \in \{kp - e_j \mid k \geq 2\}$, which contradicts the form of missing numbers. Hence S_0 contains an extra number.

Suppose that $n \in \psi'(S)$, then $n \in \pi(S) \cup \pi'(S)$, but as $\pi(S) \subseteq S$ and $\pi'(S) \subseteq S$ then $n \in S$ and this is not possible, as n is a missing number. For the same reason n cannot belong to the second part of $\theta(S)$, as $f'_t(S) \subseteq S$ by the definition of $f'_t(X)$.

Therefore, n must belong to the first part of $\theta(S)$. Then there exists equation number j, such that $n \in f'_{e_j}(S) + e_j - 1$. This implies

$$\begin{aligned} n &\in f'_{e_j}(S) + e_j - 1 \\ &\subseteq (\pi'(S) + \tfrac{p}{2} - 1 - e_j) + e_j - 1 \\ &= \pi'(S) + \tfrac{p}{2} - 2 \\ &\subseteq S + \tfrac{p}{2} - 2 \ , \end{aligned}$$

from whence it follows that $n - \frac{p}{2} + 2 \in S$. To see that this is the promised extra number in S, consider two cases of n: if $n = kp$, then $n - \frac{p}{2} + 2$ belongs to track $\frac{p}{2} - 2 \in T$, and if $n = kp - \frac{p}{2} + 1$, then $n - \frac{p}{2} + 2 = kp - p + 3$ is in track $p - 3 \in T$. □

Claim 9.4. *If $n \in S$ is the least extra number, then there exists a number $n' < n$ that is missing.*

Proof. As it has already been shown that there are no extra numbers smaller than p, and p cannot be an extra number, it follows that $n > p$.

Let $n = kp - t$ and suppose there are no missing numbers smaller than n. Then it can be inferred that $n \in f'_t(S)$.

- If $0 < t < \frac{p}{2} - 1$, then $n + t - (\frac{p}{2} - 1), n + t - p \in S$ (as they are smaller than n). Hence $n + t - (\frac{p}{2} - 1) \in \pi'(S)$ and thus
$$n = \big(n + t - (\tfrac{p}{2} - 1)\big) + \tfrac{p}{2} - 1 - t \in f'_t(S) \ .$$
- The second case is that $\frac{p}{2} \leq t < p$. Since there are no missing numbers smaller than n, then $n+t-p, n+t-p-(\frac{p}{2}-1) \in S$. Thus $n+t-p \in \pi(S)$ and hence
$$n = n + t - p + (p - t) \in f'_t(S) \ .$$

The rest of the proof is split into two cases depending on t.

Let $t \in T \setminus \{e_1, \ldots, e_\ell\}$. Then $n \in f'_t(S) \subseteq \theta(S)$. Therefore, n is present on the right-hand side of (9.8), and so it should appear on the left-hand side of (9.8). Consider the expressions on the left-hand side from which n is obtained.

- If $n \in \psi(S)$, then, in particular, $n \in S + p$. But this means that $n - p \in S$, which is a contradiction, as n was supposed to be the smallest extra number.
- If $n \in f'_{e_j}(S) + p$ for some j. Let $S_0 = S \cap \{n'' \mid n'' \leq n - p\}$. Then $n \in f'_{e_j}(S_0) + p$. Since there are no extra numbers in S_0, by Claim 9.1, $f'_{e_j}(S_0) + p$ contains only numbers on the equation tracks, and thus $n \notin f'_{e_j}(S_0) + p$. Contradiction.

Consider now the other case of $t \in \{e_1, \ldots e_\ell\}$. Then $n = kp - e_j$ for some j. Since $n \in f'_{e_j}(S)$, the number $n' = n + e_j - 1$ is in $\theta(S)$, hence n' is on the right-hand side of (9.8). Note, that there is no extra number smaller than $n' - \frac{p}{2}$.

- Suppose that $n' \in \psi(S)$. Then, in particular $n' \in S + p$. But this means that $n' - p$ is an extra number, which is a contradiction, there is no extra number smaller than $n' - \frac{p}{2}$. Thus n' is not in $\psi(S)$.
- Suppose that $n' \in f'_{e_j}(S)+p$ for some e_j. Let $S_0 = S \cap \{n'' | n'' \leq n' - p\}$. Then $n' \in f'_{e_j}(S_0) + p$. As there are no extra numbers in S_0, by Claim 9.1 $f'_{e_j}(S_0) + p$ contains only numbers on the equation tracks, and thus $n' \notin f'_{e_j}(S_0) + p$. Contradiction. □

Hence, by Claim 9.2, Claim 9.3 and Claim 9.4, there are no missing and extra numbers. Then the first and second inclusions in the statement of Lemma 9.4 hold, as they state that there are no missing numbers. The third inclusion states that there is no extra numbers, which completes the proof of Lemma 9.4. □

Lemma 9.4 has established the basic structure of any solution S, which must contain all numbers in tracks 0 and $\frac{p}{2}-1$ and no elements of any tracks besides these two and the tracks $d_1, \ldots, d_m \in \{2, \ldots, \frac{p}{2}-2\}$. Because of this, S can be shifted and intersected with itself to obtain a certain periodic set. This is what is done in the expressions π and π':

Lemma 9.5. *If S is a solution of* (9.8), *then*

$$\pi(S) = \{kp \mid k \geq 1\} \ ,$$
$$\pi'(S) = \{kp - (\tfrac{p}{2} - 1) \mid k \geq 1\} \ .$$

Proof. The proof is based upon Lemma 9.4, which assures that if $n = kp-t \in S$, then $t \in \{0, d_1, \ldots, d_m, \frac{p}{2} - 1\}$.

Consider any $n \in \pi(S) = S \cap (S+(\frac{p}{2}-1))$. Then $n \in S$ and $n-\frac{p}{2}+1 \in S$. Let $n = kp - t$, then t and $(t + \frac{p}{2} - 1 \mod p) \in \{0, d_1, \ldots, d_m, \frac{p}{2} - 1\}$, which is only possible if $t = 0$. Hence $n = kp$ and $n' = kp - (\frac{p}{2} - 1)$, which shows that $S \cap (S + (\frac{p}{2} - 1)) \subseteq \{kp \mid k \geq 0\}$.

Conversely, let $n = kp$ for some $k \geq 1$. Then $n, n - (\frac{p}{2} - 1) \in S$ by Lemma 9.4, and thus $n \in \pi(S)$.

A similar calculation can be done for the second equality. If $n \in \pi'(S) = S \cap (S + (\frac{p}{2} + 1))$, then $n \in S$ and $n - (\frac{p}{2} + 1) \in S$. By the same argument, $n = -(\frac{p}{2} - 1) \pmod p$, and hence it belongs to the given set. In the other direction, if $n = kp - (\frac{p}{2} - 1)$ for some $k \geq 1$, then $n \in S$ and $n - (\frac{p}{2} + 1) = (k-1)p \in S$ by Lemma 9.4, which shows that $n \in \pi'(S)$. $\square$

Now the values of the auxiliary expressions $\psi(X)$, $\psi'(X)$ and $\theta(X)$ can be determined by direct calculations based on the result of Lemma 9.5:

Lemma 9.6. *If $\pi(S) = \{kp \mid k \geq 1\}$ and $\pi'(S) = \{kp - (\frac{p}{2} - 1) \mid k \geq 1\}$, then*

$$\begin{aligned} \psi(S) = \psi'(S) &= \{kp \mid k \geq 0\} \cup \{kp - (\tfrac{p}{2} - 1) \mid k \geq 1\} \ , \\ f'_t(S) &= f_t(S) \quad , \textit{for all } t \in \{0, \ldots, p-1\} \ , \\ \varphi'_{j,E}(S) &= \varphi_{j,E}(S) \quad 1 \leq j \leq \ell, \ E \in \{E_j, F_j\} \ . \end{aligned}$$

Moreover, if $S \cap \{kp - t \mid k \geq 0\} = \varnothing$ for $t \in T$ then also

$$\theta(S) = \varnothing \ .$$

Proof. Using Lemma 9.5, the following direct calculations are carried out:

$$\begin{aligned} \psi(S) &= \{\tfrac{p}{2} + 1\} \cup (\pi'(S) + \tfrac{p}{2} - 1) \cup (\pi(S) + \tfrac{p}{2} + 1) \\ &= \{\tfrac{p}{2} + 1\} \cup (\{kp - (\tfrac{p}{2} - 1) \mid k \geq 1\} + \tfrac{p}{2} - 1) \cup (\{kp \mid k \geq 1\} + \tfrac{p}{2} + 1) \\ &= \{kp \mid k \geq 1\} \cup \{kp + \tfrac{p}{2} + 1 \mid k \geq 0\} \ . \end{aligned}$$

Similarly,

$$\begin{aligned}\psi'(S) &= \pi(S) \cup \pi'(S) \\ &= \{kp - \tfrac{p}{2} - 1 \mid k \geq 1\} \cup \{kp \mid k \geq 1\} \ .\end{aligned}$$

Consider now $f'_t(S)$. For $0 \leq t \leq \frac{p}{2} - 1$,

$$\begin{aligned}f'_t(S) &= S \cap (\pi'(S) + \tfrac{p}{2} - 1 - t) \\ &= S \cap (\{kp - (\tfrac{p}{2} - 1) \mid k \geq 1\} + \tfrac{p}{2} - 1 - t) \\ &= S \cap (\{kp - t \mid k \geq 1\} \\ &= f_t(S) \ ,\end{aligned}$$

and for $\frac{p}{2} \leq t \leq p - 1$, similarly,

$$\begin{aligned}f'_t(S) &= S \cap (\pi(S) + p - t) \cup S \cap (\{p - t\}) \\ &= \big(S \cap (\{kp \mid k \geq 1\} + p - t)\big) \cup \big(S \cap \{p - t\}\big) \\ &= S \cap \{kp - t \mid k \geq 1\} \\ &= f_t(S) \ .\end{aligned}$$

When additionally $S \cap \{kp - t \mid k \geq 0\} = \varnothing$ for $t \in T$, by Lemma 9.5 the value of $\theta(S)$ is calculated as: recall that

$$\begin{aligned}\theta(S) &= \bigcup_{j=1}^{\ell} \big(f'_{e_j}(S) + e_j - 1\big) \cup \bigcup_{t \in T \setminus \{e_1, \ldots, e_\ell\}} f'_t(S) \\ &= \bigcup_{j=1}^{\ell} \big(f_{e_j}(S) + e_j - 1\big) \cup \bigcup_{t \in T \setminus \{e_1, \ldots, e_\ell\}} f_t(S) \\ &= \bigcup_{j=1}^{\ell} \varnothing \cup \bigcup_{t \in T \setminus \{e_1, \ldots, e_\ell\}} \varnothing \\ &= \varnothing \ .\end{aligned}$$

As φ' is defined analogously to φ, with $f_t(X)$ replaced by $f'_t(X)$, and as it has already been proved that $f'_t(S) = f_t(S)$, it follows that $\varphi'_{j,E}(S) = \varphi_{j,E}(S)$ for $E \in \{E_j, F_j\}$. ☐

The proof of Theorem 9.2 is generally similar to the proof of Theorem 9.1, though there are more details to consider. Roughly speaking, once Lemma 9.5 and Lemma 9.6 determine the values of the auxiliary expressions and establish the equality of φ' with the earlier expression φ, Lemma 9.3 from the previous section becomes applicable, and it yields the equivalence.

proof of Theorem 9.2. $\Rightarrow$ Suppose S satisfies the equation (9.8). Then, by Lemma 9.4, it satisfies the following system of equations as well:

$$X \cap \{kp, kp + \tfrac{p}{2} + 1 \mid k \geq 0\} = \{kp, kp + \tfrac{p}{2} + 1 \mid k \geq 0\} \ , \tag{9.9}$$
$$X \cap \{kp - t \mid k \geq 0\} = \varnothing \quad , \text{for } t \in T \ . \tag{9.10}$$

Let us substitute S into (9.8) and intersect both of its sides with the set $\{kp - e_j \mid k \geq 0\}$. According to Lemma 9.6,

$$\psi(S) \cap \{kp - e_j \mid k \geq 0\} = \psi'(S) \cap \{kp - e_j \mid k \geq 0\} = \theta(S) = \varnothing \ .$$

On the other hand, by Lemma 9.6 and by Lemma 9.1,

$$\varphi'_{j,E}(S) = \varphi_{j,E}(S) \subseteq \{kp - e_j \mid k \geq 1\} \ . \tag{9.11}$$

This gives the following equality:

$$\varphi_{j,E_j}(S) + p = \varphi_{j,F_j}(S) + p, \quad \text{for } j = 1, \ldots, \ell \ . \tag{9.12}$$

Then (9.9), (9.10), (9.12) satisfy the assumptions of Lemma 9.3 with constants

$$C_0 = \{kp \mid k \geq 0\} \ ,$$
$$C_{\frac{p}{2}-1} = \{kp - (\tfrac{p}{2} - 1) \mid k \geq 0\} \ ,$$
$$C_t = \varnothing \quad , \text{for } t \in T \ ,$$

and Lemma 9.3 states that S is of the form

$$S = \{kp, kp + \tfrac{p}{2} + 1 \mid k \geq 0\} \cup \bigcup_{i=1}^{m} \{kp - d_i \mid k \in S_i\} \ ,$$

where $(S_1, \ldots, S_m)$ is a solution of the original system.

$\Leftarrow$ Conversely, assume that $(S_1, \ldots, S_m)$ is a solution of the original system, and let

$$S = \{kp, kp + \tfrac{p}{2} + 1 \mid k \geq 0\} \cup \bigcup_{i=1}^{m} \{kp - d_i \mid k \in S_i\} \ .$$

Then S satisfies (9.9), (9.10), (9.12) by Lemma 9.3. Under these premises, $\pi(S)$ and $\pi'(S)$ can be calculated in the same way as in Lemma 9.5, resulting in

$$\pi(S) = \{kp \mid k \geq 1\} \quad \text{and} \quad \pi'(S) = \{kp - (\tfrac{p}{2} - 1) \mid k \geq 1\} \ .$$

Then, by Lemma 9.6,

$$\psi(S) = \psi'(S) \ , \tag{9.13}$$
$$\varphi'_{j,E}(S) = \varphi_{j,E}(S), \quad (\text{ for } 1 \leq j \leq \ell, \ E \in \{E_j, F_j\}) \ , \tag{9.14}$$
$$\theta(S) = \varnothing \ . \tag{9.15}$$

Now it can be verified that a substitution $X = S$ turns (9.8) into an equality:

$$\begin{aligned}
\psi(S) \cup \bigcup_{j=1}^{\ell} (\varphi'_{j,E_j}(S)) &= \psi(S) \cup \bigcup_{j=1}^{\ell} (\varphi_{j,E_j}(S)) \cup \varnothing \\
&= \psi'(S) \cup \bigcup_{j=1}^{\ell} (\varphi_{j,F_j}(S)) \cup \theta(S) \\
&= \psi'(S) \cup \bigcup_{j=1}^{\ell} (\varphi'_{j,F_j}(S)) \cup \theta(S) \ .
\end{aligned}$$

by Lemma 9.2 and equations (9.13), (9.15) and (9.14). Hence S is a solution. $\square$

Now the constructions from of systems of equations representing recursive, recursively enumerable and co-recursively enumerable sets from Chapter 6 by their unique, least and greatest solutions are immediately extended to univariate equations:

Corollary 9.1. *For every recursive (recursively enumerable, co-recursively enumerable) set $S_0 \subseteq \mathbb{N}$ there exist numbers $0 \leq d < p$ and an equation $\varphi(X) = \psi(X)$ using union, intersection, addition and singleton constants, such that its unique (least, greatest) solution S satisfies*

$$S \cap \{kp - d \mid k \geq 1\} = \{kp - d \mid k \in S_0\} \ .$$

In particular, there exists an equation $\varphi(X) = \psi(X)$ with an recursively enumerable complete least solution, as well as one with a co-recursively enumerable complete greatest solution.

Chapter 10

Membership for resolved equations

Each formalism defining subsets of natural numbers is made with one goal in mind—to describe sets. As such, the most important question that can be asked about it, is the complexity of its membership problem. In this chapter the complexity of the membership problem for resolved systems of equations is studied. It is shown that it is EXPTIME-complete and this result is put in the general context of similar and related studies. It shall be used later as a building block for the considerations of the membership problem for unary conjunctive grammars.

10.1 Related work

The resolved equations over sets of natural numbers can be seen as a continuation and generalisation of long chain of studies of expressions over natural numbers. This study began in the seminal paper by Stockmeyer and Meyer [51]. Stockmeyer and Meyer established that the membership problem for expressions with union and addition is NP-complete, and if the expressions may also contain complementation, then the problem becomes PSPACE-complete. Some follow-up work was done by Yang [56], who considered *circuits* computing sets of numbers (that is, expressions in which subexpressions may be shared) with an extra operation of elementwise multiplication, and established similar complexity results. A systematic study of complexity of expressions and circuits with different sets of operations was carried out by McKenzie and Wagner [34].

This work has inspired some related studies. Breunig [5] investigated the same formalisms defined over sets of positive integers and showed that in some cases the membership problem becomes computationally easier. Similarly, the complexity of the membership problem for expressions and circuits over sets of any integers was studied by Travers [54], who found cases where the problem becomes harder and also cases where it becomes easier. Glaßer et al. [14] determined the complexity of the equivalence problem for the classes studied by McKenzie and Wagner [34].

It is expected that the membership problem for resolved equations with union, intersection and addition should be harder, as it allows a larger set of operations. This intuition is also supported that it was already shown (Theorem 4.2), that resolved systems with these operations define much larger class of sets than resolved systems with addition and union only, i.e., the class of ultimately periodic sets.

10.1.1 Fixed membership problem

The core result of this chapter, established in Section 10.2, is a construction of a fixed resolved system, such that testing the membership of numbers in its least solution is an EXPTIME-hard problem, with the numbers given in binary notation. The result is obtained by an arithmetisation of an alternating linear-space Turing machine, i.e., encoding the description of a configuration of such machine by a number and construction of equations which simulate the transition by manipulating the encoding numbers. It is also shown that for every system of resolved equations, the membership of numbers in its least solution can be tested in exponential time, which makes the constructed set the hardest. This yields EXPTIME-completeness of the *fixed membership problem*, in which the system is not a part of the input. Comparing this with circuits over sets of numbers, the latter may only generate ultimately periodic sets (unless multiplication of sets is employed), and thus the computational complexity of those sets is trivial.

10.1.2 General membership problem

This result easily leads to the complexity of the general membership problem for resolved equations with union, intersection and addition, which is stated as follows: *"Given a resolved system of equations and a number $n \geq 0$ in binary notation, determine whether n is in the first component of the least solution of the system"*. For integer expressions and integer circuits with the same operations on sets, which can be regarded as an acyclic case of systems of resolved equations, it is known from McKenzie and Wagner [34] that the membership problem is PSPACE-complete. Another weaker model are systems of resolved equations with union and addition, that is, without intersection, for which the corresponding problem is NP-complete due to the result of Huynh [17] on the commutative case of the context-free grammars. For equations with union, intersection and addition, the general membership problem is shown to be EXPTIME-complete in Section 10.3.

10.2 Arithmetisation of EXPTIME-completeness

In this section a construction of a particular resolved system of equations over sets of numbers is given. Testing membership of numbers in its least solution of this system is an EXPTIME-complete problem. This construction is accompanied by a matching upper bound:

Theorem 10.1. *The family of sets of numbers representable by resolved systems of equations with union, intersection and addition, as well as singleton constants, is a subset of EXPTIME and contains an EXPTIME-complete language.*

An EXPTIME-complete language can be generated even by a single equation using one variable.

The proof is by constructing a system of equations that encodes a computation of any linearly bounded alternating Turing machine (ATM). It is known that such machines recognise some EXPTIME-complete sets [7].

Furthermore, an ATM shall operate on a *circular tape* and move to the right at every step. Its tape shall originally contain the input string, and the cells containing it constitute all space available to the machine. Such a machine can simulate an arbitrary linear-bounded ATM by marking the position of its head on the tape, and by making one transition of the simulated ATM per each traversal of the circular tape. Hence, these restricted ATMs are as powerful as linear-bounded ATMs of the general form.

Formally, such a machine is defined as $M = (\Omega, \Gamma, Q_E, Q_A, \delta, q_0, q_{fin})$, where Ω is the input alphabet, $\Gamma = \{a_0, a_1, \ldots, a_{\max}\} \supseteq \Omega$ is the tape alphabet, Q_E and Q_A are disjoint sets of existential and universal states, respectively, $Q = Q_E \cup Q_A = \{q_0, q_1, \ldots, q_{max}\}$ and $q_{fin} \in Q$. Given an input $w \in \Omega^+$, M starts in state q_0 with the head over the first symbol of w. The transition function is $\delta : Q \times \Gamma \to 2^{Q \times \Gamma}$, and the head is moved one symbol to the right at every step. Once the head moves beyond the right-most symbol, it is moved back to the first symbol of w, maintaining its current state; this implements a circular tape. For technical reasons, assume that $(q, a') \notin \delta(q, a)$ for all $q \in Q$ and $a, a' \in \Gamma$ (that is, the machine never stays in the same state), and that $\delta(q, a) \neq \varnothing$ for all $q \in Q_A$ and $a \in \Gamma$. Such a change can be easily done by duplicating the set of states and naming the original ones as odd and the new ones as even; then each transition from odd state is to an even one and vice versa. Universal states with no successor for a should lead to a rejecting state.

The construction of a system of equations over sets of numbers simulating a computation is based upon representing instantaneous descriptions of the ATM as *numbers*. These numbers shall be considered in positional notation with base $8 + |Q| + \max(|Q| + 7, |\Gamma|)$, and the entire argument is based upon mapping the symbols used by the machine to digits, and then using addition to manipulate individual digits in the positional notation of numbers. As usually, this positional notation is only a tool for understanding the constructions, while the actual equations, deal with numbers as they are.

Let $\Sigma = \{\mathtt{0}, \mathtt{1}, \ldots, 7 + |Q| + \max(|Q| + 7, |\Gamma|)\}$ be the alphabet of digits and $k = |\Sigma|$. Define the mapping of symbols to digits,

$$\langle\cdot\rangle : Q \cup \Gamma \to \Sigma \ ,$$

as follows:

$$\begin{aligned}
\langle q_i \rangle &= 7 + i, \quad \text{for } q_i \in Q \ , \\
\langle a_i \rangle &= 7 + |Q| + i, \quad \text{for } a_i \in \Gamma \ .
\end{aligned}$$

The notation $\langle\cdot\rangle$ is naturally extended to strings over $Q \cup \Gamma$ by $\langle s_1 \ldots s_\ell \rangle = \langle s_1 \rangle \ldots \langle s_\ell \rangle$. Furthermore, let $\langle Q \rangle = \{\langle q \rangle \mid q \in Q\}$ and $\langle \Gamma \rangle = \{\langle a \rangle \mid a \in \Gamma\}$. We identify the *tape* of the ATM containing symbols $a_{i_1} \ldots a_{i_n}$ and the head in state q over the j^{th} symbol with the following string of digits:

$$\mathtt{0}\langle a_{i_1} \rangle \ldots \mathtt{0}\langle a_{i_{j-1}} \rangle \langle q \rangle \langle a_{i_j} \rangle \mathtt{0} \langle a_{i_{j+1}} \rangle \ldots \mathtt{0} \langle a_{i_n} \rangle \mathtt{0} \in \Sigma^* \ ,$$

and denote both of them as a *configuration*. For technical reasons, configurations, in which the head has just moved over the last symbol but has not yet jumped to the first position, are considered separately. These are the strings:

$$\mathtt{0}\langle a_{i_1} \rangle \ldots \mathtt{0}\langle a_{i_n} \rangle \langle q \rangle \ ,$$

where q is the current state. Note that digits denoting letters are written only in even positions, while odd positions are reserved for the states of the Turing machine. The set of all valid configurations is specified by the following regular expression over Σ:

$$\text{Tape} = (\mathtt{0}\langle\Gamma\rangle)^* \langle Q \rangle (\langle\Gamma\rangle \mathtt{0})^* \setminus \langle Q \rangle \ .$$

The set Tape should be considered as a formal language over Σ, which will be used later as a part of representations by some sets of numbers. Subsets of this set of tapes with different states will be denoted using the following notation:

$$\begin{aligned}
\text{Tape}_\alpha &= \{w \mid w \in \text{Tape},\ \alpha \in (\Gamma \cup Q)^*,\ \langle\alpha\rangle \text{ is a substring of } w\} \ , \\
\text{Tape}^\ell_\alpha &= \{w \mid w \in \text{Tape},\ \alpha \in (\Gamma \cup Q)^*,\ \langle\alpha\rangle \text{ is a prefix of } w\} \ .
\end{aligned}$$

Besides the configuration, we include in the *description* of a Turing machine a *counter of rotations* of the circular tape. This counter specifies the number of circles through the tape the machine is still allowed to make before it must halt. It is given in binary notation using zeroes and ones, and the set of valid counters is

$$\text{Counter} = \mathtt{1}\{\mathtt{0}, \mathtt{1}\}^* \ ,$$

where the digits are still in base-k notation. Normally, the counter uses only digits $\{\mathtt{0}, \mathtt{1}\}$, but in order to implement its incrementation, strings containing a single digit $\mathtt{2}$, which is a *zero with carry*, shall be used as well. The set of valid counters with a carry is

$$\text{Counter}' = \mathtt{1}\{\mathtt{0}, \mathtt{1}\}^*\mathtt{2}\mathtt{0}^* \cup \mathtt{2}\mathtt{0}^* \quad .$$

For every string $c_{k-1} \ldots c_0 \in \text{Counter} \cup \text{Counter}'$, define its value as

$$\text{Value}(c_{k-1} \ldots c_0) = \sum_{j=0}^{k-1} c_j \cdot 2^j \quad .$$

Now define the mapping from descriptions of the Turing machines to numbers. A description with configuration given by $w \in$ Tape and counter value given by $x \in$ Counter is encoded by a string of digits

$$x\mathtt{55}w \quad ,$$

where two marker digits $\mathtt{55}$ separate the counter from the tape. This string of digits specifies a number

$$(x\mathtt{55}w)_k \quad ,$$

which accordingly *represents the description.*

The key property of this representation is that *every transition of the machine reduces the numerical value of its representation.* Indeed, if the head is moved to the right, then a digit $\langle q \rangle$ is replaced with $\mathtt{0}$ and all other modifications are done on less significant digits. If the head jumps from the end to the beginning, then the counter is decremented, and since the counter occupies higher positions in the notation of the number than the tape, this transition decreases the value of the description as well. Such a monotonicity allows encoding the dependence of descriptions on each other by using addition of nonnegative numbers only. This dependence is inductively expressed in the equations defined below.

The construction of a system of equations representing the computation of the ATM begins with some expressions that will be used in the right-hand sides of equations. These expressions contain some constant sets of numbers given as regular languages over the alphabet Σ. Every such language is used to denote the set of all numbers with k-ary notation of the given form. According to Theorem 3.3, every such set can be represented by a separate system of equations using only singleton constants. All these subsystems are assumed to be included in the constructed system, and each of the regular expressions in the system can be formally regarded as a reference to one of the auxiliary variables.

Under these conventions, the following four expressions are defined, each representing a function of one set argument:

$$\begin{aligned}
\mathrm{Step}(X) = {} & \bigcup_{\substack{q \in Q_E \\ a \in \Gamma}} \bigcup_{\substack{(q',a') \in \\ \delta(q,a)}} \mathrm{Move}_{q',a',q,a}(X) \\
& \cup \bigcup_{\substack{q \in Q_A \\ a \in \Gamma}} \bigcap_{\substack{(q',a') \in \\ \delta(q,a)}} \mathrm{Move}_{q',a',q,a}(X) \ , \\
\mathrm{Move}_{q',a',q,a}(X) = {} & \Big[(X \cap (\mathrm{Counter}\, \mathtt{55}\, \mathrm{Tape}_{a'q'})_k) \\
& + \big((\langle q\rangle\langle a\rangle \mathtt{0} \boxminus \langle a'\rangle\langle q'\rangle)(\mathtt{00})^* \big)_k \Big] \cap (\mathrm{Counter}\, \mathtt{55}\, \mathrm{Tape}_{qa})_k \ , \\
\mathrm{Jump}(X) = {} & \bigcup_q \Big[\big(X \cap (\mathrm{Counter}\, \mathtt{55}\, \mathrm{Tape}^{\ell}_q)_k \big) \\
& + ((\mathtt{1000} \boxminus \langle q\rangle)(\mathtt{00})^+)_k + (\langle q\rangle)_k \Big] \\
& \cap ((\mathrm{Counter} \cup \mathrm{Counter}')\mathtt{55}\, \mathrm{Tape}_q)_k \ , \\
\mathrm{Carry}(Y) = {} & \Big[\Big(\big[(Y \cap (\{\mathtt{0},\mathtt{1}\}^*\mathtt{2}\{\mathtt{0},\mathtt{1}\}^*\, \mathtt{55}\, \mathrm{Tape})_k) + (\mathtt{10}^*)_k \big] \\
& \cap (\{\mathtt{0},\mathtt{1}\}^*\mathtt{3}\{\mathtt{0},\mathtt{1}\}^*\, \mathtt{55}\, \mathrm{Tape})_k \Big) + \big((\mathtt{10} \boxminus \mathtt{3})\mathtt{0}^* \big)_k \Big] \\
& \cap \big((\{\mathtt{0},\mathtt{1}\}^+ \cup \{\mathtt{0},\mathtt{1}\}^*\mathtt{2}\{\mathtt{0},\mathtt{1}\}^*)\, \mathtt{55}\, \mathrm{Tape} \big)_k \ .
\end{aligned}$$

Whenever the functions $\mathrm{Move}_{q',a',q,a}$ and Jump are applied to a set of X representing the descriptions, they manipulate the symbols in each description in this set in order to reconstruct the description at the previous step (one with a larger numerical value). In particular, Jump moves the head back over the edge of the tape, incrementing the counter, while $\mathrm{Move}_{q',a',q,a}$ reverses a transition from (q, a) to (q', a') by moving the head by one position to the left, restoring the letter a and returning to the state q. The function Step transcribes the logic of a single step of the ATM, taking the transition table and the alternation into account, while the function Carry is used to implement incrementation of the counter.

The set of final descriptions of the machine is defined as follows:

$$\mathrm{Final} = (\mathrm{Counter}\, \mathtt{55}\, \mathrm{Tape}_{q_{\mathit{fin}}})_k \ .$$

The system of equations uses two variables, X and Y. Either variable represents the set of proper descriptions of the machine, starting from which the machine accepts. The difference between these variables is that X represents descriptions belonging to the set Counter $\mathtt{55}$ Tape, while Y represents descriptions from $(\mathrm{Counter} \cup \mathrm{Counter}')\mathtt{55}\,\mathrm{Tape}$, in which the counter may contain one carry digit $\mathtt{2}$ that needs to be propagated to higher positions.

The equations, using the above auxiliary functions, are as follows:

$$\begin{cases} X &= \text{Final} \cup \text{Step}(X) \cup \big(Y \cap (\text{Counter}\,\texttt{55}\,\text{Tape})_k\big) \\ Y &= \text{Jump}(X) \cup \text{Carry}(Y) \ . \end{cases} \tag{10.1}$$

The equation for X inductively defines representation of descriptions of Turing machines leading to acceptance as either having accepting description, or descriptions from which the machine can go to a description with representation in X. Once the head is moved over the edge of the tape and the counter in incremented, the equation for Y is used to propagate the carry digit.

In order to determine the least solution of this system, let us establish some properties of the auxiliary functions. First of all, by Lemma 2.2, $\text{Move}_{q',a',q,a}$, Jump and Carry are distributive. On the other hand, as Step contains an intersection of two expressions involving X, then it need not be distributive.

A common expression used in these functions is an addition of a constant set of numbers $(u\texttt{0}^*)_k$ with one, two or three non-zero leading digits in u. The following lemma establishes that this addition can never rewrite the double markers 55, that is, every sum in which these markers are altered does not represent a valid description. This means that such additions manipulate the counter and the tape separately, and the changes do not mix.

Lemma 10.1 (Marker preservation)**.** *For every* $x, x' \in \{\texttt{0},\texttt{1},\texttt{2},\texttt{3}\}^* \setminus \texttt{0}\Sigma^*$ *and* $w, w' \in$ Tape*, if* $(x'\texttt{55}w')_k \in (x\texttt{55}w)_k + ((\Sigma^3 \cup \Sigma^2 \cup \Sigma)\texttt{0}^*)_k$*, then* $|w| = |w'|$*.*

Proof. Let $y = ij\tau\texttt{0}^\ell$, with $i, j, \tau \in \Sigma$, be a string representing a number, and assume that $(x'\texttt{55}w')_k = (x\texttt{55}w)_k + (y)_k$. The ℓ least significant digits of $x\texttt{55}w$ and of $x'\texttt{55}w'$ are then the same.

Consider the $(\ell+4)^{\text{th}}$ digit of $x\texttt{55}w$, let it be c. Since y has fewer than $\ell + 4$ digits, any change at this position can only be due to a carry from the position $\ell + 3$. As the digit $k - 1$ is not used in any proper encoding, $c < k - 1$. Because the carry digit is at most 1, the $(\ell+4)^{\text{th}}$ digit in $x'\texttt{55}w'$ is less or equal to $c + 1$, that is, it is less or equal to $k - 1$. Therefore, there is no carry to the position $\ell+5$ in $(x\texttt{55}w)_k + (y)_k$, and all digits in positions higher than $\ell + 4$ in $(x'\texttt{55}w')_k$ are the same as in $(x\texttt{55}w)_k$. Hence, $x'\texttt{55}w'$ has at most four digits different from $x\texttt{55}w$, which may be at the positions $\ell + 1$, $\ell + 2$, $\ell + 3$ and $\ell + 4$.

Assume for the sake of contradiction that $|w| \neq |w'|$. Since w and w' are both of odd length, the 5's in the strings $x\texttt{55}w$ and $x'\texttt{55}w'$ occur in different positions. Hence $x\texttt{55}w$ and $x'\texttt{55}w'$ differ at exactly four positions, which are the positions of 5s in them.

Note that if four digits are modified by adding $(y)_k$, then the digit in the position $\ell + 4$ can only be incremented by 1 due to a carry from the previous position. Since one of the strings $x\texttt{55}w$, $x'\texttt{55}w'$ has the digit 5 in the position $\ell + 4$, the other string should have a digit $\mathtt{4}$ or $\mathtt{6}$ in the same position. Because the latter digits are not encodings of any symbols, this yields a contradiction. ☐

The next statement describes the operation of Carry: when applied to number representing a description $x\mathtt{55}w$ with the counter x having a single carry digit $\mathtt{2}$, Carry changes this digit to $\mathtt{0}$ and increments the next digit, turning it to $\mathtt{1}$ or $\mathtt{2}$. The tape contents is not altered, only the carry digit is propagated to the next higher position. Note that all operations are in k-ary notation.

Lemma 10.2 (Carry propagation)**.** *For every $x \in \mathrm{Counter}'$ and for every $w \in \mathrm{Tape}$, $\mathrm{Carry}((x\mathtt{55}w)_k) = \{(x'\mathtt{55}w)_k\}$ for some $x' \in \mathrm{Counter} \cup \mathrm{Counter}'$ with $\mathrm{Value}(x') = \mathrm{Value}(x)$.*

For every string $\alpha \in \Sigma^*$ of any different form, $\mathrm{Carry}((\alpha)_k) = \varnothing$.

Proof. The inner intersection with $(\{\mathtt{0},\mathtt{1}\}^*\mathtt{2}\{\mathtt{0},\mathtt{1}\}^*\,\mathtt{55}\,\mathrm{Tape})_k$ ensures that the set $\mathrm{Carry}((\alpha)_k)$ is non-empty only for $\alpha = x\mathtt{55}w$ with $x \in \mathrm{Counter}'$ and $w \in \mathrm{Tape}$.

The goal is to prove that if $x = \mathtt{2}\widetilde{x} \in \mathrm{Counter}'$ and $w \in \mathrm{Tape}$, then

$$\mathrm{Carry}((\mathtt{2}\widetilde{x}\mathtt{55}w)_k) = \{(\mathtt{10}\widetilde{x}\mathtt{55}w)_k\} \ ,$$

and if $x = \widehat{x}c\mathtt{2}\widetilde{x} \in \mathrm{Counter}'$ and $w \in \mathrm{Tape}$, then

$$\mathrm{Carry}((\widehat{x}c\mathtt{2}\widetilde{x}\mathtt{55}w)_k) = \{(\widehat{x}(c+1)\mathtt{0}\widetilde{x}\mathtt{55}w)_k\} \ .$$

If a number $(x\mathtt{55}w)_k$, with $x \in \mathrm{Counter}' \cup \mathrm{Counter}$ and $w \in \mathrm{Tape}$, is substituted into the expression Carry, then the first subexpression produces all numbers of the form

$$(u)_k \in ((x\mathtt{55}w)_k + (\mathtt{10}^*)_k) \,\cap\, (\{\mathtt{0},\mathtt{1}\}^*\mathtt{3}\{\mathtt{0},\mathtt{1}\}^*\,\mathtt{55}\,\mathrm{Tape})_k \ .$$

Consider the possible changes made to $x\mathtt{55}w$ to obtain u. As $\mathtt{1}$ is added only to one digit, there cannot be a carry, because the digit $k-1$ is not used for encoding. Therefore, only one digit is modified in $x\mathtt{55}w$. Since $x\mathtt{55}w$ does not contain the digit $\mathtt{3}$ that occurs in u, the unique digit $\mathtt{2}$ in x must be replaced by $\mathtt{3}$. Denote $u = \widetilde{x}\mathtt{55}w$.

Consider u' (any such string if it is not unique) such that:

$$(u')_k \in \big((u)_k + ((\mathtt{10} \boxminus \mathtt{3})\mathtt{0}^*)_k\big) \,\cap\, \big((\{\mathtt{0},\mathtt{1}\}^+ \cup \{\mathtt{0},\mathtt{1}\}^*\mathtt{2}\{\mathtt{0},\mathtt{1}\}^*)\,\mathtt{55}\,\mathrm{Tape}\big)_k \ .$$

Let $(u')_k = (u)_k + (y)_k$, with $y \in (\mathtt{10} \boxminus \mathtt{3})\mathtt{0}^* = (k-3)\mathtt{0}^*$. By Lemma 10.1, $u' = x'\mathtt{55}w'$ and $|w'| = |w|$.

Consider the changes in $x'\mathtt{55}w'$ as compared to $\widetilde{x}\mathtt{55}w$. Since there is a digit $\mathtt{3}$ in $\widetilde{x}$ and there is no such digit in x', the position of $\mathtt{3}$ in $\widetilde{x}$ is one of the modified positions. Denote the number of this position by ℓ. Because the addition of $(y)_k$ has modified the digit $\mathtt{3}$, this means that the unique non-zero digit in y is in position ℓ or $\ell-1$. If it is in the position $\ell-1$, then the digit $\mathtt{3}$ can only be modified by adding 1 as a carry from the position $\ell-1$. This cannot be the case, as the digit $\mathtt{4}$ is not used in the encoding. Therefore, the non-zero digit in y is in the position ℓ; then adding $(y)_k$ to $(\widetilde{x}\mathtt{55}w)_k$ replaces $\mathtt{3}$ with $\mathtt{0}$ and results in a carry, thus increasing the digit in the position $\ell+1$ by 1. Note that, in particular, no changes were made to w, and hence $w' = w$.

Finally, consider the values of the counters x and x'. The value of x is $\sum c_i 2^i$, where c_i is the digit in the i^{th} position. If x has no digit in the position $\ell+1$, then assume for the purposes of calculation that $c_{\ell+1} = 0$ (this does not influence the value of the counter). In x', the digit $\mathtt{2}$ was replaced by $\mathtt{0}$, hence $c'_\ell = \mathtt{0}$. In the position $\ell+1$, the digit $c_{\ell+1}$ was replaced with $c_{\ell+1}+1$. If there was no actual digit $c_{\ell+1}$ in x, then a new digit $c'_{\ell+1} = \mathtt{1}$ has been created. In any case x' contains the digit $c'_{\ell+1} = c_{\ell+1}+1$ in this position. All other digits of the counters are left intact. Then the difference of the values of the counters is determined by the positions ℓ and $\ell+1$, and

$$\text{Value}(x) - \text{Value}(x') = (c_{\ell+1} \cdot 2^{\ell+1} + \mathtt{2} \cdot 2^\ell) - ((c_{\ell+1}+1) \cdot 2^{\ell+1} + \mathtt{0} \cdot 2^\ell) = 0 \ ,$$

that is, the value of the counter has been preserved. □

According to Lemma 10.2, Carry basically moves the carry digit higher by one position. The next lemma shows that sufficiently many iterations of Carry always eliminate the carry digit: given a counter with the notation $x' = \widetilde{x}\mathtt{0}\mathtt{1}^{\ell-1}\mathtt{2} \in \text{Counter}'$, Carry^ℓ transforms it to $x = \widetilde{x}\mathtt{1}\mathtt{0}^{\ell-1}\mathtt{0} \in \text{Counter}$.

Lemma 10.3 (Termination of carry propagation)**.** *For every* $x' \in \text{Counter} \cup \text{Counter}'$ *and* $w \in \text{Tape}$ *there exist* $x \in \text{Counter}$ *and a number* $\ell \geq 0$*, such that* $\text{Carry}^\ell((x'\mathtt{55}w)_k) = \{(x\mathtt{55}w)_k\}$ *and* $\text{Value}(x) = \text{Value}(x')$*.*

Proof. If $x' \in \text{Counter}$, then the statement of the lemma is satisfied for $\ell = 0$ and $x = x'$.

Let $x' \in \text{Counter}'$ and construct a sequence $x_0, x_1, \ldots, x_\ell$, with $x_i \in \text{Counter}'$ and $\text{Value}(x_i) = \text{Value}(x')$, where ℓ shall be determined below, as follows. Let $x_0 = x'$. For every $i \geq 1$, consider $\text{Carry}((x_{i-1}\mathtt{55}w)_k)$, which, by Lemma 10.2, equals $\{(x_i\mathtt{55}w)_k\}$ for some $x_i \in \text{Counter}' \cup \text{Counter}$ with $\text{Value}(x_i) = \text{Value}(x_{i-1})$. If $x_i \in \text{Counter}$, then $\ell = i$ and $x = x_i$ satisfy the statement of the lemma. Otherwise, if $x_i \in \text{Counter}'$, then the construction of the sequence continues.

Note that $(x_{i+1}\mathtt{55}w)_k > (x_i\mathtt{55}w)_k$ and hence all elements of the sequence are distinct. Since there exist only finitely many elements of $\text{Counter}'$ with the same value, the sequence cannot be infinite and eventually $x_i \in \text{Counter}$ is obtained. □

The next lemma determines the operation of Jump, which can be described as follows. This function is applicable to numbers representing descriptions in which the head scans the first symbol, and the result of Jump on every such number is the number representing the *previous* description, in which the head is at the right-most position beyond the end of the string, while the value of the counter x is greater by 1.

Lemma 10.4. *Let $x = \widetilde{x}c \in$ Counter with $c \in \{\mathtt{0}, \mathtt{1}\}$ and let $w = \langle q\rangle\widetilde{w}\mathtt{0} \in$ Tape with $q \in Q$, that is, w is a configuration with the head over the first symbol. Then*

$$\text{Jump}((x\mathtt{55}w)_k) = \{(\widetilde{x}(c+1)\mathtt{550}\widetilde{w}\langle q\rangle)_k\} \ .$$

For any string $\alpha \in \Sigma^*$ of a different form, $\text{Jump}((\alpha)_k) = \varnothing$.

Proof. The inner subexpression of $\text{Jump}((\alpha)_k)$,

$$\{(\alpha)_k\} \cap \text{Counter}\,\mathtt{55}\,\text{Tape}_q^\ell \ ,$$

ensures that $\alpha = x\mathtt{55}w$ where $w = \langle q\rangle\widetilde{w}\mathtt{0}$ for some $\widetilde{w} \in \langle\Gamma\rangle(\mathtt{0}\langle\Gamma\rangle)^*$, that is, that the digit specifying the state of the machine is in the left-most position. If w is of a different form, then $\text{Jump}(x\mathtt{55}w) = \varnothing$. Fix an arbitrary state $q \in Q$; as the outermost operation in Jump, a union over all q will be taken.

The next subexpression performs an addition

$$(x\mathtt{55}\langle q\rangle\widetilde{w}\mathtt{0})_k + ((\mathtt{1000} \boxminus \langle q\rangle)(\mathtt{00})^+)_k + (\langle q\rangle)_k \ ,$$

which is meant to remove q from the beginning of the tape, increment the counter and place q in the end of the tape. Consider an arbitrary $(y)_k = ((\mathtt{1000} \boxminus \langle q\rangle)(\mathtt{00})^\ell)_k + (\langle q\rangle)_k$, with $\ell \geq 1$. Denote $(u)_k = (x\mathtt{55}w)_k + (y)_k$ and assume that

$$u \in (\text{Counter}' \cup \text{Counter})\mathtt{55}\,\text{Tape}_q \ ,$$

as the subsequent operation in Jump is an intersection with $((\text{Counter}' \cup \text{Counter})\mathtt{55}\,\text{Tape}_q)_k$. Let $u = x'\mathtt{55}w'$ with $x' \in \text{Counter} \cup \text{Counter}'$ and $w' \in$ Tape. Also note that $w' = \mathtt{0}\widetilde{w}'\langle q\rangle$, as the right-most digit in y is $\langle q\rangle$ and the right-most digit in w is $\mathtt{0}$, and there is only one digit from $\langle Q\rangle$ in w'.

Since y has non-zero digits only in the positions $2\ell+1, 2\ell+2, 2\ell+3$ and 1, and the digit $k-1$ does not occur in $x\mathtt{55}w$, adding y cannot change any digit in $x\mathtt{55}w$ in positions higher than $2\ell+4$. Let $2\ell' = |\widetilde{w}\mathtt{0}|$. Then adding y to w modifies the digit in the position $2\ell'+1$, which is $\langle q\rangle$. Hence, either $2\ell'+1 = 2\ell+1$ or $2\ell'+1 = 2\ell+3$.

If $2\ell'+1 = 2\ell+3$, then there is either $\langle q\rangle$ or $\langle q\rangle - 1$ in the position $2\ell'+1$ in $x\mathtt{55}w+y$. Moreover, the digit $\mathtt{5}$ in the position $2\ell'+3 = 2\ell+5$ in $(x\mathtt{55}w)_k$

was not modified by adding $(y)_k$. Since the position $2\ell' + 1$ is to the right of $\mathtt{55}$ in $x'\mathtt{55}w'$, it contains $\mathtt{0}$. This is a contradiction, as $\langle q\rangle > \langle q\rangle - 1 > \mathtt{0}$.

Hence, $2\ell' + 1 = 2\ell + 1$. Let $x = \widetilde{x}c$. Then

$$(x\mathtt{55}w)_k + (y)_k = (\widetilde{x}(c+1)\mathtt{550}\widetilde{w}\langle q\rangle)_k \ ,$$

and therefore $x' = \widetilde{x}(c+1)$ and $w' = \mathtt{0}\widetilde{w}\langle q\rangle$, as stated in the lemma. ☐

The next operation is Move, which represents symbol manipulation, head movement and state change of a Turing machine according to the transitions specified in δ. Generally, when $\text{Move}_{q',a',q,a}$ is applied to a valid description, it computes the *preceding description* of the machine. This description is unique because of the restriction built in $\text{Move}_{q',a',q,a}$: the intersections therein ensure that in the current description the machine is in the state q' and the symbol to the left rewritten at the previous step is a', while in the previous description the machine was in the state q and used to scan the symbol a. Then the previous description is obtained simply by rewriting the three digits $\mathtt{0}\langle a'\rangle\langle q'\rangle$ with $\langle q\rangle\langle a\rangle\mathtt{0}$. For all other descriptions and in all other cases, the function produces the empty set.

Lemma 10.5. *Let $q, q' \in Q$ with $q \neq q'$, and let $a, a' \in \Gamma$. Let $x \in$ Counter and $w = \widehat{w}\mathtt{0}\langle a'\rangle\langle q'\rangle\widetilde{w} \in$ Tape for some $\widehat{w} \in (\mathtt{0}\langle\Gamma\rangle)^*$ and $\widetilde{w} \in (\langle\Gamma\rangle\mathtt{0})^*$. Then*

$$\text{Move}_{q',a',q,a}((x\mathtt{55}w)_k) = \{(x\mathtt{55}\widehat{w}\langle q\rangle\langle a\rangle\mathtt{0}\widetilde{w})_k\} \ .$$

For every string $\alpha \in \Sigma^*$ of a different form, $\text{Move}_{q',a',q,a}(\alpha) = \varnothing$.

Proof. Fix a', q', a and q. The inner subexpression of $\text{Move}_{q',a',q,a}$,

$$(\alpha)_k \cap (\text{Counter}\,\mathtt{55}\,\text{Tape}_{a'q'})_k \ ,$$

ensures that $\alpha = x\mathtt{55}w$ and $w = \widehat{w}\mathtt{0}\langle a'\rangle\langle q'\rangle\widetilde{w}$ for some $\widehat{w} \in (\mathtt{0}\langle\Gamma\rangle)^*$ and $\widetilde{w} \in (\langle\Gamma\rangle\mathtt{0})^*$. For any α of a different form $\text{Move}_{q',a',q,a}((\alpha)_k)$ is empty, similarly for w of a different form, $\text{Move}_{q',a',q,a}((x\mathtt{55}w)_k)$ is empty.

The next subexpression performs the operation

$$(x\mathtt{55}w)_k + ((\langle q\rangle\langle a\rangle\mathtt{0} \boxminus \langle a'\rangle\langle q'\rangle)(\mathtt{00})^*)_k \cap (\text{Counter}\,\mathtt{55}\,\text{Tape}_{qa})_k \ ,$$

which is designed to replace the digits $\mathtt{0}\langle a'\rangle\langle q'\rangle$ in w with the digits $\langle q\rangle\langle a\rangle\mathtt{0}$. The task is to show that the addition always proceeds according to this plan.

Let

$$(y)_k = ((\langle q\rangle\langle a\rangle\mathtt{0} \boxminus \langle a'\rangle\langle q'\rangle)\mathtt{0}^{2\ell})_k \in ((\langle q\rangle\langle a\rangle\mathtt{0} \boxminus \langle a'\rangle\langle q'\rangle)(\mathtt{00})^*)_k$$

and consider the string $(x'\mathtt{55}w')_k = (x\mathtt{55}w)_k + (y)_k$, where $x' \in$ Counter and $w' \in$ Tape. By Lemma 10.1, $|w'| = |w|$.

As y has non-zero digits only in positions $2\ell+1, 2\ell+2, 2\ell+3$, while the digit $k-1$ is not a valid encoding of any symbol and may not occur in

$x\mathtt{55}w$, adding $(y)_k$ cannot change any digits in $(x\mathtt{55}w)_k$ in positions higher than $2\ell+4$.

Let $2\ell' = |\widetilde{w}|$. Since $q \neq q'$, w and w' must differ in the position $2\ell'+1$, where w has the digit $\langle q'\rangle$. Therefore, $2\ell'+1 = 2\ell+3$ or $2\ell'+1 = 2\ell+1$.

Suppose $2\ell'+1 = 2\ell+3$, that is, the digit $\langle q'\rangle$ in w is added to $\langle q\rangle$ or $\langle q\rangle - 1$ in y, with a possible carry from the lower digits. Then $(x\mathtt{55}w)_k + (y)_k$ has a digit $(\langle q\rangle + \langle q'\rangle - 1)$, $(\langle q\rangle + \langle q'\rangle)$ or $(\langle q\rangle + \langle q'\rangle + 1)$ (modulo k in each case) in the position $2\ell'+1$. Since $(\langle q\rangle)_k, (\langle q'\rangle)_k \leq 6 + |Q|$ and $q \neq q'$, it follows that $(\langle q\rangle + \langle q'\rangle)_k \leq 11 + 2|Q| < k$ and $(\langle q\rangle + \langle q'\rangle + 1)_k \leq 12 + 2|Q| < k$. Each sum is smaller than k and is therefore represented by a single digit. However, each of these digits is greater than $(\langle q\rangle)_k$, and hence all of them are filtered out by the intersection with $(\mathrm{Counter}\,\mathtt{55}\,\mathrm{Tape}_{qa})_k$.

In the other case of $2\ell'+1 = 2\ell+1$, the addition proceeds as expected, and $x' = x$ and $w' = \widehat{w}\langle q\rangle\langle a\rangle\mathtt{0}\widetilde{w}$, as stated in the lemma. □

The flow control of an alternating Turing machine includes existential and universal nondeterminism in the corresponding states, and a single step is in fact a disjunction or a conjunction of several transitions as specified in Move. This logic is transcribed in the expression $\mathrm{Step}(X)$, which computes the set of all *previous descriptions*, from which machines in a universal state make all their transitions to descriptions in X, and machines in an existential state make at least one of their transitions to some description in X. This implements one step of the computation of the machine, backwards.

Lemma 10.6. *Let $x \in \mathrm{Counter}$ and $w \in \mathrm{Tape}$, let $q \in Q$ be the state encoded in w. Let $X \subseteq \mathbb{N}$. Then $(x\mathtt{55}w)_k \in \mathrm{Step}(X)$ if and only if the following conditions hold:*

- *the configuration w has the head* ***not*** *in the position beyond the rightmost symbol, that is, $w = \widehat{w}\langle q\rangle\langle a\rangle\mathtt{0}\widetilde{w}$ for some $\widehat{w} \in (\mathtt{0}\langle\Gamma\rangle)^*$ and $\widetilde{w} \in (\langle\Gamma\rangle\mathtt{0})^*$ and $a \in \Gamma$;*
- *if $q \in Q_E$, then $(x\mathtt{55}w')_k \in X$ for some configuration w' among successors to w;*
- *if $q \in Q_A$, then $(x\mathtt{55}w')_k \in X$ for every configuration w' among successors to w.*

Proof. ⇔ Consider the definition of Step:

$$\mathrm{Step}(X) = \Big(\bigcup_{\substack{\widehat{q}\in Q_E\\ \widehat{a}\in\Gamma}} \bigcup_{\substack{(q',a')\in\\ \delta(\widehat{q},\widehat{a})}} \mathrm{Move}_{q',a',\widehat{q},\widehat{a}}(X)\Big)$$
$$\cup \Big(\bigcup_{\substack{\widehat{q}\in Q_A\\ \widehat{a}\in\Gamma}} \bigcap_{\substack{(q',a')\in\\ \delta(\widehat{q},\widehat{a})}} \mathrm{Move}_{q',a',\widehat{q},\widehat{a}}(X)\Big) .$$

Assume that $(x\mathtt{55}w)_k \in \mathrm{Step}(X)$, let $a \in \Gamma$ be the symbol scanned by the head of the machine in the configuration w, and let $q \in Q$ be the current state. Then, according to Lemma 10.5, $(x\mathtt{55}w)_k \in \mathrm{Move}_{q',a',\widehat{q},\widehat{a}}(X)$ only if $(\widehat{q},\widehat{a}) = (q,a)$, and hence the subexpressions $\mathrm{Move}_{q',a',\widehat{q},\widehat{a}}(X)$ with $(\widehat{q},\widehat{a}) \neq (q,a)$ need not be taken into account.

First suppose that q is an existential state. Then

$$(x\mathtt{55}w)_k \in \bigcup_{(q',a')\in\delta(q,a)} \mathrm{Move}_{q',a',q,a}(X) \ ,$$

that is, there exist $q' \in Q$ and $a' \in \Gamma$ with $x\mathtt{55}w \in \mathrm{Move}_{q',a',q,a}(X)$ for some $(q',a') \in \delta(q,a)$. Note that $q \neq q'$ by the technical assumption that the machine changes its state upon every transition. Since $\mathrm{Move}_{q',a',q,a}$ is distributive over union there exists a number $n \in X$ with $(x\mathtt{55}w)_k \in \mathrm{Move}_{q',a',q,a}(\{n\})$. Then, by Lemma 10.5, n must be of the form $(x\mathtt{55}w')_k$ with $w' = \widehat{w}\mathtt{0}\langle a'\rangle\langle q'\rangle\widetilde{w}$ for some $\widehat{w} \in (\mathtt{0}\langle\Gamma\rangle)^*$ and $\widehat{w} \in (\langle\Gamma\rangle\mathtt{0})^*$, and with $w = \widehat{w}\langle q\rangle\langle a\rangle\mathtt{0}\widetilde{w}$. Since $(q',a') \in \delta(q,a)$, w' is a successor configuration to w, and $(x\mathtt{55}w')_k \in X$. The position of the head in w is to the left of the right-most symbol.

The case of $q \in Q_A$ is similar. It follows from $(x\mathtt{55}w)_k \in \mathrm{Step}(X)$ that

$$(x\mathtt{55}w)_k \in \bigcap_{(q',a')\in\delta(q,a)} \mathrm{Move}_{q',a',q,a}(X) \ ,$$

that is, for all $q' \in Q$ and $a' \in \Gamma$ with $(q',a') \in \delta(q,a)$ it holds that $(x\mathtt{55}w)_k \in \mathrm{Move}_{q',a',q,a}(X)$. As in the previous case, this implies that $w = \widehat{w}\langle q\rangle\langle a\rangle\mathtt{0}\widetilde{w}$ and there is $(x\mathtt{55}w'_{q',a'})_k \in X$ with $w'_{q',a'} = \widehat{w}\mathtt{0}\langle a'\rangle\langle q'\rangle\widetilde{w}$. These are consecutive configurations, and every successor configuration to w is of this form for some $(q',a') \in \delta(q,a)$. Then the required element $(x\mathtt{55}\widehat{w}\mathtt{0}\langle a'\rangle\langle q'\rangle\widetilde{w})_k$ is in X for all q' and a' with $(q',a') \in \delta(q,a)$. Also note that $\delta(q,a) \neq \varnothing$ by assumption, and hence there is at least one such pair (q',a'). So w is of the required form with the head not beyond the right-most symbol.

$\Leftarrow$ Let $w = \widehat{w}\langle q\rangle\langle a\rangle\mathtt{0}\widetilde{w}$ and first consider the case of $q \in Q_E$. Let w' be one of the next configurations of the machine with $(x\mathtt{55}w')_k \in X$. Then $w' = \widehat{w}\mathtt{0}\langle a'\rangle\langle q'\rangle\widetilde{w}$ for some $(q',a') \in \delta(q,a)$, and it is known that $q \neq q'$. By Lemma 10.5, $\mathrm{Move}_{q',a',q,a}((x\mathtt{55}w')_k) = \{(x\mathtt{55}w)_k\}$. Since $\mathrm{Move}_{q',a',q,a}((x\mathtt{55}w')_k) \subseteq \mathrm{Step}(X)$, this shows that $(x\mathtt{55}w)_k \in \mathrm{Step}(X)$.

If $q \in Q_A$, then, by assumption, $(x\mathtt{55}w')_k \in X$ for all configurations w' that are immediate successors to the configuration w. That is, for all $(q',a') \in \delta(q,a)$, $(x\mathtt{55}w'_{q',a'})_k \in X$, where $w'_{q',a'} = \widehat{w}\mathtt{0}\langle a'\rangle\langle q'\rangle\widetilde{w}$. For every such pair, by Lemma 10.5, $(x\mathtt{55}w)_k \in \mathrm{Move}_{q',a',q,a}((x\mathtt{55}w'_{q',a'})_k)$. Hence,

$$(x\mathtt{55}w)_k \in \bigcap_{(q',a')\in\delta(q,a)} \mathrm{Move}_{q',a',q,a}(X) \ ,$$

and therefore $(x55w)_k \in \mathrm{Step}(X)$. $\square$

Thus the formal meaning of all auxiliary operations has been established, and the equations can now be analysed. The equation for X states that a description leads to acceptance if and only if it is itself accepting (Final), or one can directly proceed from it to a description leading to acceptance (Step(X)), or that it is a description obtained in Y. The equation for Y specifies circular rotation of the tape by Jump(X) and implements iterated carry propagation as in Lemma 10.3 by a self-reference Carry(Y). Altogether, the least solution of these equations corresponds to the computation of the machine as follows:

Lemma 10.7. *Let* (S_X, S_Y) *be the least solution of the system* (10.1).

I. *Let* $x \in \mathrm{Counter}$*,* $w \in \mathrm{Tape}$ *and* $(x55w)_k \in S_X$*. Then* M *accepts starting from the configuration* w*.*

II. *Conversely, if* M *accepts starting from a configuration* $w \in \mathrm{Tape}$*, and the longest path in the tree of the accepting computation has length* m*, then* $(x55w)_k \in S_X$ *for each* $x \in \mathrm{Counter}$ *with* $\mathrm{Value}(x) \geq m$*.*

Proof. As the least solution of the system is computed by fixpoint iteration (2.2), denote by $S_X^{(\ell)}$ and $S_Y^{(\ell)}$ the X- and Y-components of the vector $\varphi^\ell(\varnothing, \ldots, \varnothing)$ obtained after $\ell \geq 0$ iterations. Then $(x55w)_k \in S_X$ if and only if $(x55w)_k \in S_X^{(\ell)}$ for some $\ell \geq 1$.

(I) Assume that $(x55w)_k \in S_X^{(\ell)}$. It has to be proved that the Turing machine accepts starting from the configuration w. The proof is an induction on ℓ.

By the equation for X, $(x55w)_k \in S_X^{(\ell)}$ means that either $(x55w)_k \in (\mathrm{Final})_k$, or $(x55w)_k \in \mathrm{Step}(S_X^{(\ell-1)})$, or $(x55w)_k \in S_Y^{(\ell-1)}$. If $(x55w)_k \in (\mathrm{Final})_k$, then w is an accepting configuration, as the Turing machine is already in an accepting state. Consider the other two cases.

Let $(x55w)_k \in \mathrm{Step}(S_X^{(\ell-1)})$, and let $w = \widehat{w}\langle q\rangle\langle a\rangle 0\widetilde{w}$; the configuration is of this form by Lemma 10.6. Consider the set of numbers $S = \{(x55\widehat{w}0\langle a'\rangle\langle q'\rangle\widetilde{w})_k \mid (q', a') \in \delta(q, a)\}$ representing all possible next descriptions of the machine. Suppose first that $q \in Q_A$. Then, by Lemma 10.6, $S \subseteq S_X^{(\ell-1)}$, and by the induction hypothesis, all numbers in $S_X^{(\ell-1)}$ represent descriptions with configurations from which the machine accepts. Hence the machine accepts starting from all successor configurations to w, and then, by definition, it accepts starting from w.

The case of $q \in Q_E$ is treated similarly. Again, by Lemma 10.6, at least one number from S is in $S_X^{(\ell-1)}$, and every number in $S_X^{(\ell-1)}$ represents a description with configuration from which the machine accepts, by the induction hypothesis. Accordingly, the machine accepts starting from the configuration w, because it accepts starting from one of its successor configurations.

Consider the other case of $(x\texttt{55}w)_k \in S_Y^{(\ell-1)}$, that is, of $x\texttt{55}w$ obtained by processing the carry in the counter. This processing may be reconstructed as a finite sequence $x_{\ell-1}$, $x_{\ell-2}$, $\ldots$, $x_{\ell_0} \in \text{Counter} \cup \text{Counter}'$, where the number $\ell_0 \geq 0$ is determined later, and, for all $i \in \{\ell-1, \ell-2, \ldots, \ell_0\}$,

$$\begin{aligned} &\text{Value}(x_i) = \text{Value}(x) \ , \\ &(x_i\texttt{55}w)_k \in S_Y^{(i)} \ , \\ &(x_i\texttt{55}w)_k = \text{Carry}((x_{i-1}\texttt{55}w)_k), \quad \text{unless } i = \ell_0 \ . \end{aligned}$$

Let $x_{\ell-1} = x$. Each string of digits x_i for $i = \ell-2$, $\ell-3$, $\ldots$ is defined by a backward induction as follows.

Assume that $(x_i\texttt{55}w)_k \in S_Y^{(i)}$, and hence $(x_i\texttt{55}w)_k \in \text{Carry}(S_Y^{(i-1)})$ or $(x_i\texttt{55}w)_k \in \text{Jump}(S_X^{(i-1)})$. In the former case, as Carry is distributive, there exists a number $n \in S_Y^{(i-1)}$ with $(x_i\texttt{55}w)_k \in \text{Carry}(n)$. Then, by Lemma 10.2, $n = (x'\texttt{55}w)_k$ for some x' with $\text{Value}(x') = \text{Value}(x_i)$, and it holds that $\text{Carry}((x'\texttt{55}w)_k) = \{(x_i\texttt{55}w)_k\}$. Then $x_{i-1} = x'$ forms the next element of the sequence.

In the latter case, $(x_i\texttt{55}w)_k \in \text{Jump}(S_X^{(i-1)})$. As Jump is distributive, there is a number $n \in S_X^{(i-1)}$ with $(x_i\texttt{55}w)_k \in \text{Jump}(n)$. Then, according to Lemma 10.4, $n = (x'\texttt{55}w')_k$, where $x' \in \text{Counter}'$ with $\text{Value}(x') = \text{Value}(x_i) - 1 = \text{Value}(x) - 1$, $w' = \langle q \rangle \widetilde{w}\texttt{0}$ and $w = \texttt{0}\widetilde{w}\langle q \rangle$; that is, $\text{Jump}((x'\texttt{55}w')_k) = \{(x_i\texttt{55}w)_k\}$. Then ℓ_0 is defined as i, which completes the construction of the sequence.

It has been shown that there exists $(x'\texttt{55}w')_k \in S_X^{(\ell_0-1)}$ with $\text{Value}(x') = \text{Value}(x) - 1$, such that the machine goes from the configuration w to the configuration w'. By the induction hypothesis for $x'\texttt{55}w'$, the machine accepts from the configuration w'. Therefore, the machine accepts starting from w, as claimed.

It is left to mention that the case of $(x_i\texttt{55}w)_k \in \text{Jump}(S_X^{(i-1)})$ in the above proof eventually occurs, because otherwise the sequence would continue until $(x_0\texttt{55}w)_k \in S_Y^{(0)} = \varnothing$, which is impossible.

(II) For the converse statement, let w be a configuration, and assume that there is an accepting computation starting from w, with the longest path of length ℓ. The claim is that $x\texttt{55}w \in S_X$ holds for every $x \in \text{Counter}$ with the value at least ℓ. This is proved by induction on ℓ.

If $\ell = 0$, then w is a configuration in the accepting state, and therefore, by the equation for X in the system, $(x\texttt{55}w)_k \in (\text{Final})_k \subseteq S_X$ for all $x \in \text{Counter}$.

Assume there is an accepting computation starting from w with the longest path of length $\ell+1$. Suppose first that $w = \widehat{w}\langle q \rangle \langle a \rangle \texttt{0}\widetilde{w}$ with $\widehat{w}, \widetilde{w} \in \Sigma^*$, $a \in \Gamma$ and $q \in Q$, that is, the configuration w has the head anywhere except in the position beyond the right-most symbol. Consider the case of $q \in Q_A$. Then the machine accepts from each successor configuration to w, and longest path in each of these accepting computations is of length at most ℓ. Hence all strings of the form $x\mathtt{55}w'$, where w' is a successor configuration to w and $x \in \text{Counter}$ is a counter of value at least ℓ, are in S_X by the induction hypothesis. Then, by Lemma 10.6, $(x\mathtt{55}w)_k \in \text{Step}(S_X)$. By the equation for X, $(x\mathtt{55}w)_k \in S_X$, which proves this case.

Now consider the case when $q \in Q_E$. Fix any $x \in \text{Counter}$ of value at least ℓ. At least for one successor configuration to w, the Turing machine accepts starting from it, with the longest path of length at most ℓ. Accordingly, at least one number of the form $(x\mathtt{55}w')_k$, where w' is one of the successor configurations to w, is in S_X by the induction hypothesis. Therefore, by Lemma 10.6, $(x\mathtt{55}w)_k \in \text{Step}(S_X)$, and, by the equation for X, $(x\mathtt{55}w)_k \in S_X$, as stated in the lemma.

Finally, consider the case where the head of the Turing machine is in the position beyond the right-most symbol, and let $w = \mathtt{0}\widetilde{w}\langle q\rangle$. Let $w' = \langle q\rangle\widetilde{w}\mathtt{0}$ be the next configuration, from which the machine accepts with the longest path of length ℓ. Let $x' \in \text{Counter}$ be a counter of value at least ℓ. By the induction hypothesis, $(x'\mathtt{55}w')_k \in S_X$. Then, by Lemma 10.4, there is a string $(x''\mathtt{55}w)_k \in \text{Jump}(S_X) \subseteq S_Y$, where $x'' \in \text{Counter}' \cup \text{Counter}$ and $\text{Value}(x'') = \text{Value}(x') + 1$. Hence, by Lemma 10.3, there exists $\ell \geq 0$, for which $(x\mathtt{55}w)_k \in \text{Carry}^\ell(S_Y)$, where x is the unique element of Counter with $\text{Value}(x'') = \text{Value}(x)$. By the equation for Y in the system, $\text{Carry}(S_Y) \subseteq S_Y$, and since Carry is monotone, this implies the following chain of inclusions:

$$\text{Carry}^\ell(S_Y) \subseteq \text{Carry}^{\ell-1}(S_Y) \subseteq \text{Carry}^{\ell-2}(S_Y) \subseteq \ldots \subseteq \text{Carry}(S_Y) \subseteq S_Y \ .$$

Therefore, $(x\mathtt{55}w)_k \in S_Y$, which, by the equation for X in the system (10.1), implies $(x\mathtt{55}w)_k \in S_X$, as claimed. □

It remains to observe that the number of steps made by the machine is exponentially bounded, and hence the acceptance of a string by the machine is represented by the following number in the least solution of the constructed system:

Main Claim. *The ATM M accepts a string $a_1 \ldots a_n \in \Omega^+$ if and only if*

$$(\mathtt{10}^{n\log(|\Gamma|)+\log(n+1)+\log(|Q|)}\mathtt{55}\langle q_0\rangle\langle a_1\rangle\mathtt{0}\langle a_2\rangle\mathtt{0}\ldots\langle a_n\rangle\mathtt{0})_k \in S_X \ .$$

Proof. The initial configuration of M on $a_1 \ldots a_n$ is

$$w = \langle q_0\rangle\langle a_1\rangle\mathtt{0}\langle a_2\rangle\mathtt{0}\ldots\langle a_n\rangle\mathtt{0} \in \text{Tape} \ .$$

⇛ If M accepts starting from this configuration, then the longest path in the accepting computation consists of at most

$$(n+1)\cdot|Q|\cdot|\Gamma|^n \leq 2^{\log(n+1)} \cdot 2^{\log|Q|+n\log|\Gamma|}$$

steps, since all configurations forming this path must be different. Then, by Lemma 10.7, for $x = \mathtt{10}^{\log(n+1)+n\log|\Gamma|+\log|Q|}$ with $\mathrm{Value}(x) = 2^{\log(n+1)+n\log|\Gamma|+\log|Q|}$ it holds that $(x\mathtt{55}w)_k \in S_X$.

$\circleddash$ Conversely, if there exists $x \in \mathrm{Counter}$ with $x\mathtt{55}w \in S_X$, then, according to Lemma 10.7, M accepts starting from the configuration w. ☐

Proof of Theorem 10.1. The system of equations constructed above has an EXPTIME-hard least solution. It uses constant sets of numbers with a regular base-k notation, which are expressed in additional equations for additional variables constructed according to Theorem 3.3.

Theorem 8.1 is used to show that one terminal is enough to define an EXPTIME-complete set. It follows from it that one can efficiently construct an equation $Z = \psi(Z)$ with a least solution S and numbers $p, d \geq 1$ such that $n \in S_X$ if and only if $pn - d \in S$. Hence the membership problem for S_X is polynomially reducible to the membership problem to S, and hence S is EXPTIME-hard.

To see that the least solution of every system is in EXPTIME, it is sufficient to represent it as a conjunctive grammar over a unary alphabet. Then, given a number n, its membership in the least solution can be tested by supplying the string a^n to a known cubic-time parsing algorithm for conjunctive grammars [36]. Its time is cubic in n, hence exponential in the length of the binary notation of n. ☐

This establishes the computational complexity of sets of numbers specified by resolved systems of equations with union, intersection and addition, which is the main technical result of this chapter.

10.3 The membership problem

Consider the *general membership problem* for these equations, stated as follows: "Given a system $X_i = \varphi_i(X_1, \ldots, X_m)$ and given a number n *in binary notation*, determine whether n is in the first component of the least solution of the system".

Theorem 10.2. *The general membership problem for resolved systems of equations over sets of numbers with the operations of union, intersection and addition is EXPTIME-complete.*

It remains EXPTIME-complete for instances with a single equation using a single variable.

Proof. **Membership in EXPTIME.** The existence of such an algorithm can be inferred from the known polynomial-time algorithm for solving the membership problem for conjunctive grammars [37]. It is sufficient to represent the given system as a conjunctive grammar over a unary alphabet,

		Representable sets	Membership problem
$\{\cup, +\}$	expressions	Finite	NP-complete [51]
	circuits	Finite	NP-complete [17, 34]
	equations	Ult. periodic [13]	NP-complete [17, 48]
$\{\cup, \cap, +\}$	expressions	Finite	PSPACE-complete [34]
	circuits	Finite	PSPACE-complete [34]
	equations	$\subsetneq$ **EXPTIME**	**EXPTIME-complete**

Table 10.1: Comparison of formalisms over sets of nonnegative integers.

with a linearly bounded blow-up, and then represent the given number n as a string a^n, with an exponential blow-up.

An exponential-time algorithm for equations over sets of numbers can be constructed directly as follows. Given a number n and a resolved system with m variables, the algorithm will simulate fixpoint iteration as in (2.2), but all sets will be computed as subsets of $\{0, \ldots, n\}$. The algorithm thus uses variables $X_i \subseteq \{0, \ldots, n\}$, which are initially empty, and which are updated at every step by substituting their values into the right-hand side of the system. Up to $m(n + 1)$ such iterations can be done until the sets stabilise, when the algorithm can answer whether n is in X_1. Each iteration is polynomial in $n + m$, and so is the entire algorithm. Since n is given to the algorithm in binary notation, the size of the instance of the general membership problem is $\log n + m$, and hence the algorithm makes at most exponentially many iterations each working in exponential time.

The **EXPTIME-hardness** of the general membership problem immediately follows from Theorem 10.1 by fixing the system of equations. As this theorem in fact assures that one equation with a single variable is enough, similar claim holds in the general case. $\square$

Recalling that *equations* over sets of numbers are a generalisation of *circuits* and *expressions* over sets of numbers, Theorem 10.2 can be directly compared to the existing results on the complexity of these formalisms, summarised in the right column of Table 10.1. The left column of the table characterises the families of sets representable by each of these six formalisms. Obviously, expressions and circuits can represent only finite sets, as any Boolean combinations and sums of finite sets are finite. The sets represented by equations with union and addition are bound to be ultimately periodic, because all context-free languages over a unary alphabet are regular [13]. By Theorem 10.1 and Theorem 10.2 that equations with union, intersection and addition represent some EXPTIME-complete sets. At the same time, these equations cannot represent the whole class EXPTIME $= \bigcup_{k \geq 1} \mathrm{DTIME}(2^{n^k})$, because their solutions lie in DTIME(2^{n^2}), which is a proper subset of EXPTIME due to the time hierarchy theorem.

Chapter 11

Conjunctive grammars

In general, undecidability results are obtained by simulations of Turing Machines. As languages of computations of Turing Machines can be encoded in languages generated by unary conjunctive grammars, it looks that some decision problems for them must be undecidable. In this chapter, the exact levels of undecidability of the decision problems are calculated as well. Then the same problems are considered when only one non-terminal is allowed. In such a setting, some of them retain their complexity, while others become much simpler. For example, equality to a constant language for conjunctive grammars with one variable is shown to be decidable for a large class of constants, in contrast to the multiple-nonterminal case where it is undecidable for every fixed conjunctive constant.

We consider the standard decisions problems for grammars:

- the (fixed) compressed membership problem—for a fixed grammar G and a word $w = a^n$, given by a context-free grammar G_a such that $L(G_w) = \{w\}$, decide, whether $a^n \in L(G)$.

- the general compressed membership problem—for a given pair G and context free grammar G_w it is asked, whether $w \in L(G)$

- equivalence problem—for two given grammars G, G' it is asked, whether $L(G) = L(G')$

- equivalence to a fixed (finite/regular) language—for a fixed (finite/regular) language L_0 test if $L(G) = L_0$

- finiteness problem—for a given grammar G check, whether $L(G)$ is finite

- co-finiteness problem—for a given grammar G decide, whether $L(G)$ is co-finite

11.1 Membership

Among all decision problems, the membership problem is the most important and prominent one. Formalism are devised to define languages and one needs to know if a word in in the constructed languages.

Luckily, the complexity of equations over sets of numbers established in Chapter 10 has direct implications on the complexity of conjunctive grammars over a one-letter alphabet.

Every conjunctive language can be parsed by a cubic-time algorithm and thus is in P [36], and some conjunctive languages over a multiple-letter alphabet are known to be P-complete [38]. The case of a unary alphabet is special, as it is known that no sparse language, in particular no unary language, can be P-complete unless DLOGSPACE = P [35, 6], that is, unless the notion of P-completeness is trivial. However, from Theorem 10.1 one can infer the following result slightly weaker than P-completeness:

Corollary 11.1. *There exists an EXPTIME-complete set of numbers $S \subseteq \mathbb{N}$, such that the language $L = \{a^n \mid n \in S\}$ of unary notations of numbers from S is generated by a conjunctive grammar.*

Note that for every unary language generated by a conjunctive grammar, the corresponding set of numbers is in EXPTIME. The set from in Corollary 11.1 can thus be regarded as the computationally hardest among unary conjunctive languages.

This has a straightforward consequence referring to the complexity of parsing for conjunctive grammars. For context-free languages, it is known each of them is in NC^2, that is, can be parsed by a polynomial-size circuit of depth $O(\log^2 n)$, which was discovered independently by Brent and Goldschlager [4] and by Rytter [50]. The known examples of P-complete conjunctive languages imply that, unless P = NC, there are no polylogarithmic-time parallel parsing algorithm for conjunctive languages [38]. Now a similar result can be claimed with respect to grammars over a one-letter alphabet.

Corollary 11.2. *Unless* PSPACE = EXPTIME*, there is no logarithmic-space parsing algorithm for conjunctive languages over a unary alphabet.*

Indeed, having such an algorithm for the particular language L from Corollary 11.1 would give a polynomial-space algorithm for the EXPTIME-complete set S.

Let us now consider the complexity of the *compressed membership problem* for conjunctive grammars. This is a problem of testing whether a string w is generated by a grammar G, but unlike the ordinary membership problem, here the string w is given in a compressed form constructed by a data compression algorithm. The standard abstraction for data compression, which captures algorithms such as LZ77 and LZW, is the notion of a *straight-line program* (SLP). Following Plandowski and Rytter [48], a straight-line program over the alphabet Σ is a context-free grammar $G_w = (\Sigma, N, P, S)$ with $L(G_w) = \{w\}$, and G_w is considered as a compressed representation of w. Note that the length of w may be exponentially larger than the description of the grammar G_w.

The compressed membership problem is defined as follows:

(Membership problem). *Given a conjunctive grammar $G = (\Sigma, N, P, S)$ and a context-free grammar $G_w = (\Sigma, N, P, S)$ generating a singleton language $\{w\}$, determine whether $w \in L(G)$.*

The complexity of this problem for the common families of languages is known from the literature. For regular expressions, as well as for deterministic finite automata, the problem was shown to be in P by Plandowski and Rytter [48], while Markey and Schnoebelen [33] demonstrated its P-hardness already for a fixed regular language. Plandowski and Rytter [48] also showed that the compressed membership problem for context-free grammars is in PSPACE, and Lohrey [32] proved that it is PSPACE-hard even for a fixed deterministic linear context-free language. Furthermore, Lohrey [32] established the EXPSPACE-completeness of the same problem for context-sensitive grammars, showing that it is EXPSPACE-hard already for a fixed language.

Theorem 10.1 and Theorem 10.2 can be used to establish the complexity of the same problem for conjunctive grammars.

Theorem 11.1. *The compressed membership problem for conjunctive grammars is EXPTIME-complete. It remains EXPTIME-complete for a fixed conjunctive language $L_0 \subseteq a^*$ and one non-terminal unary conjunctive grammar.*

Proof. An exponential-time algorithm for this problem is straightforward. Given a conjunctive grammar G and a context-free grammar G_w with $L(G_w) = \{w\}$, the algorithm first decompresses the string w, that is, constructs it explicitly. Its length is at most exponential in the size of G_w. Then the known polynomial-time algorithm for solving the membership problem for a conjunctive grammar [37] is applied.

To show the EXPTIME-hardness of the problem for a particular language, let $S \subseteq \mathbb{N}$ be the set of numbers represented in Theorem 10.1. This theorem asserts that we may assume that S is the least solution of a single equation using one variable. Define $L_0 = \{a^n \mid n \in S\}$, by the duality between unary conjunctive grammars and systems of equations over natural numbers, it is a conjunctive language generated by a unary conjunctive grammar with a single non-terminal.

Then the problem of testing whether a number n is in S can be reduced to the compressed membership problem in L_0 as follows.

Let $b_{\ell-1} \ldots b_1 b_0$ with $b_i \in \{0, 1\}$ be the binary notation of n, that is, $n = \sum_{i=0}^{\ell-1} b_i 2^i$. Let $i_1 < \ldots < i_k$ be all numbers with $b_{i_j} = 1$. Then the

	Membership	Compressed membership
Regular languages		
DFA	DLOGSPACE-comp.	P-comp. [33, 48]
Regular expressions	NLOGSPACE-comp.	P-comp. [33, 48]
Grammars		
Linear context-free	NLOGSPACE-comp.	PSPACE-comp. [32, 48]
Context-free	P-comp. [27]	PSPACE-comp. [32, 48]
Linear conjunctive	P-comp. [37, 38]	?
Conjunctive	P-comp. [37]	**EXPTIME-comp.**
Context-sensitive	PSPACE-comp.	EXPSPACE-comp. [32]

Table 11.1: Complexity of membership problems for automata and grammars.

singleton language $\{a^n\}$ is generated by the following context-free grammar G_n:

$$\begin{aligned} S &\to A_{i_1} \dots A_{i_k} \\ A_0 &\to a \\ A_{i+1} &\to A_i A_i \quad , \text{ for } 0 \le i < i_k \ . \end{aligned}$$

Accordingly, the description of G_n is a yes-instance of the compressed membership problem for L_0 if and only if $n \in S$, which completes the reduction. $\square$

This result is compared to the similar earlier cited results on other families of formal grammars in Table 11.1.

11.2 Equivalence

We begin with making some technical assumptions on the encoding of language of the transition of TM. Recall the language VALC(T) defined in the beginning of Part II as:

$$\text{VALC}(T) = \{C_T(w)\natural w \mid C_T(w) \text{ is an accepting computation}\} \ .$$

Without loss of generality, VALC(T) it is defined over an alphabet of digits $\Sigma_k = \{\mathtt{0}, \dots, k-1\}$, for a suitable k, and $w \in (\Sigma_k \setminus \{\mathtt{0}\})^*$, so that no string in VALC(T) has a leading zero. Define the language INVALC(T) as $(\Sigma_k^* \setminus \mathtt{0}\Sigma_k^*) \setminus$ VALC(T). These elaborations do not affect the fact that both VALC(T) and INVALC(T) are intersection of linear context-free grammars [43] and thus are recognised by trellis automata, so that Theorem 4.2 is still applicable and can be used to obtain the following result:

Corollary 11.3. *For every Turing machine T there exist and can be effectively constructed conjunctive grammars G and G' over the alphabet $\{a\}$, such that $L(G) = \{a^n \mid n \in (\text{VALC}(T))_k\}$ and $L(G') = \{a^n \mid n \in (\text{INVALC}(T))_k\}$, where k is the size of the alphabet used for encoding the computations.*

This result is used in the rest of this chapter while proving the undecidability of various decision problems.

It is first applied to the fixed equivalence problem stated as:

(Fixed equivalence problem). *Given a conjunctive grammar determine whether $L(G) = L_0$.*

Theorem 11.2. *For every fixed unary conjunctive language $L_0 \subseteq a^*$, the fixed equivalence problem is Π_1-complete.*

Proof. The containment of the problem in Π_1 is evident, since the equivalence problem for two given recursive languages is in Π_1. It is the Π_1-hardness that has to be established.

We reduce the problem of the emptiness of TM, a known Π_1-complete problem. Let $G_0 = (\Sigma, N_0, P_0, S_0)$ be a fixed conjunctive grammar generating L_0. Depending on the form of L_0, let us consider two cases.

Case I: L_0 contains no subset of the form $a^\ell(a^p)^*$, where $\ell \geq 0$ and $p \geq 1$. Given a Turing machine T, construct a conjunctive grammar $G_T = (\{a\}, N_T, P_T, S_T)$ for $\{a^n \mid n \in (\text{VALC}(T))_k\}$. On the basis of G_T and G_0, construct a new conjunctive grammar $G = (\{a\}, N_T \cup N_0 \cup \{S, A\}, P_T \cup P_0 \cup P, S)$, where P contains the following new rules:

$$\begin{aligned}
S &\to S_0 \mid S_T A \\
A &\to aA \mid \varepsilon \\
S_T &\to \dots \quad \text{(rules generating } \{a^n \mid n \in (\text{VALC}(T))_k\}) \\
S_0 &\to \dots \quad \text{(rules generating } L_0)
\end{aligned}$$

Now, if $L(T) = \varnothing$, then $L(G_T) = \varnothing$, the rule $S \to S_T A$ in G generates nothing, and therefore $L(G) = L(G_0) = L_0$.

Suppose $L(T) \neq \varnothing$. Then there is a string $C_T(w)\natural w \in \text{VALC}(T)$, and accordingly there exists $a^n \in L(G_T)$. Hence the rule $S \to S_T A$ in G can be used to generate all strings in $a^n a^*$, and therefore $L(G)$ contains the subset $a^\ell(a^p)^*$ for $\ell = n$ and $p = 1$. As L_0 contains no such subset by assumption, $L(G) \neq L_0$.

It has been proved that $L(G) = L_0$ if and only if $L(T) = \varnothing$, which completes the reduction for this form of L_0.

Case II: L_0 contains a subset $a^\ell(a^p)^*$, where $\ell \geq 0$ and $p \geq 1$. Assume that $p \geq k$, i.e., it is larger than the cardinality of the alphabet used for INVALC(T) (if p is too small, any of its multiples can be taken). Define INVALC(T) over a p-letter alphabet, consider the set of numbers $(\text{INVALC}(T) \cdot \mathtt{0})_p$, and construct a conjunctive grammar $G'_T = (\{a\}, N'_T, P'_T, S'_T)$ generating $\{a^n \mid n \in (\text{INVALC}(T) \cdot \mathtt{0})_p\}$. Using G_0 and G'_T, construct a new grammar $G = (\{a\}, N'_T \cup N_0 \cup \{S, B, C\}, P_T \cup P_0 \cup P, S)$, where the new rules in P are as follows:

$$\begin{aligned}
S &\to S_0\&B \mid a^\ell S'_T \\
B &\to a^i \quad (\text{for all } 0 \leq i < \ell) \\
B &\to a^{\ell+i}C \quad (\text{for all } 1 \leq i < p) \\
C &\to a^pC \mid \varepsilon \\
S'_T &\to \ldots \quad (\text{rules generating } \{a^n \mid n \in (\text{INVALC}(T) \cdot \mathtt{0})_p\}) \\
S_0 &\to \ldots \quad (\text{rules generating } L_0)
\end{aligned}$$

Note that $L_G(B) = a^* \setminus a^\ell(a^p)^*$ and therefore $L_G(S_0\&B) = L_0 \setminus a^\ell(a^p)^*$. Thus

$$L(G) = L_0 \iff L_G(a^\ell S'_T) \supseteq a^\ell(a^p)^*$$

and as $L_G(S'_T) \subseteq (a^p)^*$

$$\begin{aligned}
&\iff L_G(S'_T) = (a^p)^* \\
&\iff \{a^n \mid n \in (\text{INVALC}(T) \cdot \mathtt{0})_p\} \supseteq (a^p)^* \\
&\iff \{a^n \mid n \in (\text{INVALC}(T))_p\} \supseteq a^* \\
&\iff \text{INVALC}(T) = \Sigma_p^* \setminus \mathtt{0}\Sigma_p^* \\
&\iff \text{VALC}(T) = \varnothing \\
&\iff L(T) = \varnothing \ .
\end{aligned}$$

Therefore, again, $L(G) = L_0$ if and only if $L(T) = \varnothing$, which proves the Π_1-hardness of the problem. □

If L_0 is not generated by a conjunctive grammar, then the problem of testing whether a given conjunctive grammar generates L_0 becomes trivial. Hence, the following characterisation is obtained:

Corollary 11.4. *For every fixed language $L_0 \subseteq a^*$, the problem of testing whether a given conjunctive grammar over $\{a\}$ generates L_0 is either* Π_1-*complete or trivial.*

In contrast, in the case of one-nonterminal grammars, the equality to any fixed ultimately periodic set is clearly decidable: it is sufficient to substitute it into the equation and check whether it is turned into an equality. This approach extends to a larger class of fixed languages:

Theorem 11.3. *There exists an algorithm, which, given a unary conjunctive grammar with one non-terminal symbol and a finite automaton M over an alphabet $\Sigma_k = \{0, 1, \ldots, k-1\}$, determines whether $\{a^n \mid \text{the } k\text{-ary notation of } n \text{ is in } L(M)\}$ is the language generated by the given grammar.*

The algorithm works by substituting the set of numbers defined by M into the equation corresponding to the given conjunctive grammar. The value of each subexpression is computed in the form of a finite automaton over Σ_k representing base-k notation. For Boolean operations this is clearly possible, while the addition of sets can be done symbolically on finite automata for their base-k representation according to the following lemma:

Proposition 11.1 (Jirásková, Okhotin [26]). *Let L_1 and L_2 be regular languages over an alphabet $\Sigma = \{0, 1, \ldots, k-1\}$, with $L_1 \cap 0\Sigma^* = L_2 \cap 0\Sigma^* = \varnothing$. Then the language*

$$L_1 \boxplus L_2 = \{w_1 \boxplus w_2 \mid w_1 \in L_1,\ w_2 \in L_2\} \subseteq \Sigma^*$$

is regular.

This shows that equality to a fixed language is decidable for one-nonterminal conjunctive grammars for a fairly large class of constants.

Consider now the general equivalence problem, formulated as:

(General equivalence problem). *Given two grammars G_1, G_2 decide, whether $L(G_1) = L(G_2)$.*

While the fixed equivalence problem is tractable for one non-terminal symbol and hard for many non-terminal symbols, the general equivalence problem is hard already for one non-terminal unary conjunctive grammars

Theorem 11.4. *The general equivalence problem for one-nonterminal unary conjunctive grammars is Π_1-complete.*

Proof. The proof is by reduction from the equivalence problem for unary conjunctive grammars with multiple nonterminals. Two grammars are combined into one, the construction of Theorem 8.1 is applied, and then the start symbols of the two grammars are exchanged and the construction is applied again. The two resulting one-nonterminal grammars are equivalent if and only if the original grammars generate the same language.

Before approaching the equivalence problem for one-nonterminal conjunctive grammars, let us establish the undecidability of the following technical problem:

Claim 11.1. *The problem of testing whether for a given conjunctive grammar $G = (\{a\}, N, P, S)$ with two designated nonterminals S and S', $L_G(S) = L_G(S')$, is undecidable.*

Proof. Theorem 11.2 states that the problem of whether two unary conjunctive grammars generate the same language is Π_1-complete. Let $G_1 = (\{a\}, P_1, N_1, S_1)$ and $G_2 = (\{a\}, P_2, N_2, S_2)$ be any two conjunctive grammars over $\{a\}$. Assume, without loss of generality, that $N_1 \cap N_2 = \varnothing$. Construct a new conjunctive grammar $G = (\{a\}, P_1 \cup P_2, N_1 \cup N_2, S_1)$. Then $L_G(S_1) = L(G_1)$ and $L_G(S_2) = L(G_2)$, and therefore testing the equality of $L_G(S_1)$ and $L_G(S_2)$ solves the equivalence problem for G_1 and G_2. $\square$

Now this technical problem may be easily reduced to the equivalence problem for one-nonterminal conjunctive grammars over $\{a\}$. Let a grammar $G = (\{a\}, \{A_1, A_2, \ldots, A_m\}, P, A_1)$ be given, and assume without loss of generality that it is of the form required in Lemma 8.1; it is asked whether $L_G(A_1) = L_G(A_2)$. Construct a one-nonterminal unary conjunctive grammar G' that encodes G according to Theorem 8.1, with

$$\begin{aligned} L(G') = {} & \{a^{np-d_1} \mid a^n \in L_G(A_1)\} \cup \{a^{np-d_2} \mid a^n \in L_G(A_2)\} \\ & \cup \bigcup_{i \geq 3} \{a^{np-d_i} \mid a^n \in L_G(A_i)\} \ . \end{aligned}$$

Next, the same transformation is applied to the grammar $G = (\{a\}, \{A_2, A_1, A_3, \ldots, A_m\}, P, A_2)$, with nonterminals A_1 and A_2 exchanged. The values of $p, d_1, \ldots, d_m$ are the same, as they depend only on m, so the generated language is

$$\begin{aligned} L(G'') = {} & \{a^{np-d_2} \mid a^n \in L_G(A_1)\} \cup \{a^{np-d_1} \mid a^n \in L_G(A_2)\} \\ & \cup \bigcup_{i \geq 3} \{a^{np-d_i} \mid a^n \in L_G(A_i)\} \ . \end{aligned}$$

Clearly, the two languages are the same if and only if $L_G(A_1) = L_G(A_2)$. $\square$

11.3 Finiteness

The grammar finiteness problem is now investigated.

(Grammar finiteness). *Given a grammar G decide, whether $L(G)$ is finite.*

The the finiteness problem is Σ_1-hard already for the one-non-terminal case. For the general case only a trivial Σ_2 upper bound is supplied.

Theorem 11.5. *The finiteness problem for one-nonterminal unary conjunctive grammars is Σ_1-complete.*

Proof. To see that the problem is in Σ_1 consider the following nondeterministic Turing machine that tests whether a given conjunctive grammar $G = (\{a\}, \{S\}, P, S)$ generates a finite language. The machine starts with guessing a finite language $F \subset a^*$ and then uses the method of Theorem 11.3 to check whether $L(G) = F$.

The Σ_1-hardness is shown by reduction from the problem of whether a given unary conjunctive grammar $G = (\{a\}, \{A_1, \dots, A_n\}, P, A_1)$ generates a language *other than* a^+. This problem is Σ_1-complete by Theorem 11.2.

Assume without loss of generality that G contains a nonterminal that generates an infinite language and that G is of the form given in Lemma 8.1. By Theorem 8.1, there exist numbers $1 \leq d_1 < \dots < d_n < p$, and a one-nonterminal grammar $G_1 = (\{a\}, \{S\}, P_1, S)$ generating the language $\{a^{np-d_i} \mid a^n \in L_G(A_i)\}$ can be constructed. Accordingly, $\{a^{np-d_1} \mid n \geq 1\} \subseteq L(G_1)$ if and only if $L(G) = a^+$.

Note that, according to the theorem, for each rule

$$S \to a^{\ell_1} SS \& \dots \& a^{\ell_k} SS \tag{11.1}$$

of this grammar there exists a number i with $L(a^{\ell_1} SS \& \dots \& a^{\ell_k} SS) \subseteq \{a^{np-d_i} \mid n \geq 1\}$. Let such a rule be called an i-rule. Also note that $L(G_1)$ is always infinite, because of a nonterminal generating an infinite language.

Now construct a new grammar $G_2 = (\{a\}, \{S\}, P_2, S)$, where P_2 contains all rules of the form $S \to a^m$ from G_1, as well as a rule of the form

$$S \to a^{\ell_1} SS \& \dots \& a^{\ell_k} SS \& a^{p+d_1-d_i} S \tag{11.2}$$

for each i-rule (11.1).

Clearly, $L(G_2) \subseteq L(G_1)$, as every rule in P_2 is a more restrictive version of some rule from P_1 containing an extra conjunct, and thus every derivation in G_2 can be simplified down to a derivation of the same string in G_1. Furthermore, the grammar G_2 inherits the property of G_1 that each rule (11.2) generates a subset of $\{a^{np-d_i} \mid n \geq 1\}$, for d_i given in the last conjunct. The purpose of the additional conjunct in (11.2) is to make the membership of a^{np-d_1} in $L(G_2)$ a necessary condition for generating the number $a^{(n+1)p-d_i}$. In this way, if any number in track d_1 is missing, then no larger numbers will be generated, and the language will be finite.

Formally, it is claimed that G_2 generates an infinite language if and only if $\{a^{np-d_1} \mid n \geq 1\} \subseteq L(G_1)$.

$\Leftarrow$ Assume that $\{a^{np-d_1} \mid n \geq 1\} \subseteq L(G_1)$, that is, every string a^{np-d_1} with $n \geq 1$ is in $L(G_1)$. It is claimed that $L(G_1) \subseteq L(G_2)$, and as $L(G_1)$ is infinite, this would prove that so is $L(G_2)$.

Suppose the contrary, that $L(G_1) \setminus L(G_2) \neq \varnothing$, and let a^{np-d_i} with $n \geq 1$ and $1 \leq i \leq m$ be the shortest string in $L(G_1)$ that is not in $L(G_2)$. This string must be produced by a long rule of G_1 (because all the short rules of G_1 are in G_2 as well), and therefore $n \geq 2$. Consider the i-rule (11.1) by which

a^{np-d_i} is generated. Then $a^{np-d_i} \in a^{\ell_j} L(G_1)^2$ and hence $a^{np-d_i} \in a^{\ell_j} L(G_2)^2$, as $\ell_j \geq 1$ and $L(G_1)$ and $L(G_2)$ do not differ on strings shorter than a^{np-d_i}. For the same reason, $a^{(n-1)p-d_1} \in L(G_1)$ implies $a^{(n-1)p-d_1} \in L(G_2)$ (note that $(n-1)p - d_1 > 0$, as $n \geq 2$), and, accordingly, $a^{np-d_i} \in a^{p+d_1-d_i} L(G_1)$. Therefore, a^{np-d_i} is generated in G_2 by the rule (11.2) corresponding to (11.1), which contradicts the assumption that $a^{np-d_i} \notin L(G_2)$.

Ⓡ⇒ Conversely, if $\{a^{np-d_1} \mid n \geq 1\} \nsubseteq L(G_1)$, then there is a number $n \geq 1$ with $a^{np-d_1} \notin L(G_1)$, and hence with $a^{np-d_1} \notin L(G_2)$ (as $L(G_2) \subseteq L(G_1)$). Now the claim is that no string longer than a^{np-d_1} is in $L(G_2)$.

Let $a^{n'p-d_i} \in L(G_2)$ for some $n' > n$ and $1 \leq i \leq m$ be the shortest string of length greater than $np - d_1$ generated by G_2, and let $a^{n'p-d_i}$ be generated by an i-rule (11.2). According to the last conjunct of this rule, $a^{n'p-d_i} \in a^{p+d_1-d_i} L(G_2)$ and hence $a^{(n'-1)p-d_1} \in L(G_2)$. Now if $n' - 1 = n$, then this does not hold by assumption, and if $n' - 1 > n$, then $a^{(n'-1)p-d_1}$ is a string shorter than $a^{n'p-d_i}$ satisfying the assumptions, and in both cases a contradiction is obtained.

The above claims imply that $L(G_2)$ is finite if and only if $L(G) \neq a^+$, which completes the reduction. □

Fact 11.1. *The grammar finiteness problem for unary conjunctive grammar is in* Σ_2.

Proof. This can expressed as a formula

$$\exists n \forall m \, m > n \Rightarrow a^m \notin L(G) \ .$$

Since checking whether $a^n \in L(G)$ can be done in polynomial time, the whole problem is in Σ_2. □

11.4 Co-finiteness

As opposed to the grammar finiteness problem, now the complementary grammar co-finiteness problem is inspected.

(Grammar co-finiteness problem). *Given a grammar G determine, whether $L(G)$ is co-finite.*

The results for the co-finiteness problem are similar as for the finiteness. Already in the one-nonterminal case the problem is Σ_1-complete, while in case of many nonterminals only a trivial Σ_2 upper bound is provided.

Theorem 11.6. *The grammar co-finiteness problem for one-nonterminal unary conjunctive grammars is Σ_1-complete.*

Proof. The problem is in Σ_1 by Theorem 11.3: an algorithm solving this problem can nondeterministically guess an DFA M recognizing a co-finite language and test whether the grammar generates $L(M)$.

The Σ_1-hardness of the co-finiteness problem is established by a reduction from the emptiness problem for unary conjunctive grammars with unrestricted number of nonterminals, which is Σ_1-hard by Theorem 11.2. Without loss of generality, assume that all the words produced by the given unary conjunctive grammar are of length at least one: it is enough to introduce a new start symbol A' and a production $A' \to Aa$, where A is the old start symbol. The proof is given in terms of equations over sets of numbers.

Let G_0 be a unary conjunctive grammar with a start symbol A_1, and assume $\varepsilon \notin L(G_0)$. Construct a grammar G by introducing an additional non-terminal A_2 with the same set of productions as A_1. It is easy to see that $L_G(A_1) = L_G(A_2) = L_{G_0}(A_1)$. Then, according to Theorem 8.1, G is transformed into a univariate equation over sets of natural numbers $X = \varphi(X)$ with a unique solution $S = \bigcup_i S_i$, where $S_i = \{np - d_i \mid n \in L_G(A_i)\}$. Now construct a new expression φ' by introducing another term to φ:

$$\varphi'(X) = \varphi(X) \cup \bigcup_{i=0}^{p-1} (X{+}d_1{+}i \, \cap \, X{+}d_2{+}i) \ .$$

In other words, the following p extra rules are added to G':

$$S \to a^{d_1+i} S \& a^{d_2+i} S \quad , \text{ for } 0 \leq i \leq p-1$$

The idea of the construction is that if any string appears on track d_1, these rules will generate all longer strings, thus 'spamming' the language to make it co-finite.

Let S' be the unique solution of the equation $X = \varphi'(X)$.

Claim 11.2. *If $L(G_0) = \varnothing$, then $S' = S$, and accordingly S' is not co-finite.*

Proof. It is easy to see that $S \subseteq S'$, since $\varphi(X) \subseteq \varphi'(X)$ for all X. Thus it remains to show that $S' \setminus S = \varnothing$.

Suppose the contrary and consider the smallest number $n \in S' \setminus S$. Then also $n \in \varphi'(S') \setminus \varphi(S)$. But $n \in \varphi'(S')$ implies that $n \in \varphi'(S)$, as the numbers used to obtain n in $\varphi'(S')$ are smaller than n. Since $\varphi'(S) = \varphi(S) \cup \bigcup_{t=0}^{p-1}(S + d_1 + t \cap S + d_2 + t)$, it follows that $n \in \bigcup_{t=0}^{p-1}(S + d_1 + t \cap S + d_2 + t)$. It is shown that if $n \in S + d_1 + t \cap S + d_2 + t$, then $n \in S_1 + d_1 + t \cap S_2 + d_2 + t$, similarly as in Lemma 8.2. By distributivity,

$$\begin{aligned} S{+}d_1 \, \cap \, S{+}d_2 &= \bigcup_i (S_i + d_1) \cap \bigcup_j (S_j + d_2) \\ &= \bigcup_{i,j} (S_i + d_1) \cap (S_j + d_2) \ , \end{aligned}$$

and the value of each subexpression is

$$\begin{aligned} (S_i + d_1) \cap (S_j + d_2) = \big(&\{np \mid n \in L_G(A_i)\} - d_i + d_1\big) \\ &\cap \big(\{np \mid n \in L_G(A_j)\} - d_j + d_2\big) \ . \end{aligned}$$

Since $d_1, d_2, d_i, d_j \in \{\frac{p}{4}+1, \ldots, \frac{p}{2}\}$, the differences $-d_i + d_1$, $-d_j + d_2$ are in $\{-\frac{p}{4}+1, -\frac{p}{4}+2, \ldots, \frac{p}{4}-1\}$, and thus any number that belongs to this intersection is equal modulo p both to $d_1 - d_i$ and to $d_2 - d_j$. Accordingly, the latter two numbers must be equal, that is,

$$4^1 - 4^i = 4^2 - 4^j \ ,$$

which is true only for $i = 1$ and $j = 2$. Therefore, both S_1 and S_2 are nonempty, which yields a contradiction, as $S_1 = S_2 = \varnothing$ by the assumption.

The contradiction obtained proves that $S = S'$. For the definition of S according to Theorem 8.1, it is easy to see that it is never co-finite, as by definition it has empty tracks. □

Claim 11.3. *If $a^n \in L(G_0)$ for $n \geq 1$, then every number greater or equal to pn is in S', and thus S' is co-finite.*

Proof. By Theorem 8.1, $pn - d_1, pn - d_2 \in S$, and accordingly $pn - d_1, pn - d_2 \in S'$, since $S \subseteq S'$.

Let $m = pn' + i$ for some $0 \leq i < p$ and $n' \geq n$. By an induction on n' it will be proved that $m \in S'$. If $n' = n$ then, as stated above, $pn' - d_1, pn' - d_2 \in S'$, and if $n' > n$, then $pn' - d_1, pn' - d_2 \in S$ by the induction assumption. In each case m is produced by the subexpression $X{+}d_1{+}i \cap X{+}d_2{+}i$ as follows:

$$pn' + i = (pn' - d_1) + d_1 + i \in S' + d_1 + i \ ,$$
$$pn' + i = (pn' - d_2) + d_2 + i \in S' + d_2 + i \ ,$$

and thus

$$\begin{aligned} m &\in (S' + d_1 + i) \cap (S' + d_2 + i) \\ &\subseteq \varphi'(S') = S' \ . \end{aligned}$$

□

It follows from Claim 11.2 and Claim 11.3 that $L(G_0)$ is non-empty if and only if the solution of $X = \varphi(X)$ is co-finite, which shows the correctness of the reduction. □

Fact 11.2. *The grammar co-finiteness problem for unary conjunctive grammars is in Σ_2*

Proof. This can be expressed as a formula

$$\exists n \forall m \ m > n \Rightarrow a^m \notin L(G) \ .$$

And as testing whether $a^m \notin L(G)$ can be done in polynomial time, the whole problem is in Σ_2. □

	equiv. to reg. L_0	equivalence	finiteness	co-finiteness
any N	Π_1-complete	Π_1-complete	$\Sigma_1 \leq \cdot \leq \Sigma_2$	$\Sigma_1 \leq \cdot \leq \Sigma_2$
$N = \{S\}$	decidable	Π_1-complete	Σ_1-complete	Σ_1-complete

Table 11.2: Decision problems for conjunctive grammars over $\{a\}$.

The decidability status of decision problems for one-nonterminal and multiple-nonterminal unary conjunctive grammars is summarised in Table 11.2.

Chapter 12

General systems of equations

Consider basic properties of unresolved equations, such as the existence and the uniqueness of solutions. For the more general case of language equations it is known that these properties, as well as a few others, are undecidable [47, 41, 42], and their exact position in the arithmetical hierarchy has been determined. These results will now be re-created for equations over sets of numbers, based upon the constructions from the previous chapters.

All the corollaries follow from straightforward application of Lemma 7.4 to the systems mentioned in the construction of the appropriate theorems.

Theorem 12.1. *The problem of whether a system of equations $\varphi_i(X_1, \dots, X_n) = \psi_i(X_1, \dots, X_n)$ over sets of natural numbers has a solution is Π_1-complete.*

It remains Π_1-hard if there is only one equation with single variable allowed and the used operations are union and addition, or intersection and addition.

Proof. The problem is in Π_1 in the more general case of language equations [47].

Its Π_1-hardness is proved by a reduction from the emptiness problem for Turing machines. Let T be a TM and construct a system of equations in variables $(Y_0, \dots, Y_5, X_1, \dots, X_m)$ with the unique solution $Y_i = \mathrm{VALC}_i(T)$, $X_j = K_j \subseteq \mathbb{N}$ as described in Lemma 6.3. Since $S(T) = \varnothing$ if and only $\bigcup_{i=0}^{5} \mathrm{VALC}_i(T) = \varnothing$, it is sufficient to add six new equations $Y_i = \varnothing$ for $i \in \{0, 1, \dots, 5\}$, so that the resulting system has a solution if and only if $S(T) = \varnothing$.

Then Theorem 9.2 is applied and a single equation with a single variable is obtained. According to this theorem, it has a solution if and only if the original system has a solution. $\square$

Corollary 12.1. *Testing whether system of equations $\varphi_i(X_1, \dots, X_n) = \psi_i(X_1, \dots, X_n)$ over sets of natural numbers, where φ_i, ψ_i use ultimately periodic constants and addition only, has a solution is Π_1-complete.*

Theorem 12.2. *Testing whether a system $\varphi_i(X_1, \ldots, X_n) = \psi_i(X_1, \ldots, X_n)$ over sets of natural numbers has a unique solution is a Π_2-complete problem.*

It is still Π_2-hard if the operations are limited to union (intersection) and addition and there is a single equation using one variable.

Proof. The Π_2 upper bound is known from the case of language equations [47].

Π_2-hardness is proved by a reduction from the known Π_2-complete Turing machine universality problem [49, §14.8], which can be stated as follows: "Given a TM M working on natural numbers, determine whether it accepts every $n \in \mathbb{N}_0$". Given M, construct the system of equations as in Lemma 6.3. It has a unique solution if and only if the bounds $S(T) \subseteq S \subseteq \mathbb{N}$ are tight, that is, if and only if the TM accepts every number.

We apply Theorem 9.2 to this system. Thus a single equation with a single variable is obtained. According to Theorem 9.2, the obtained equation has a unique solution if and only if the original system has a unique solution. This completes the reduction. □

Corollary 12.2. *For a system $\varphi_i(X_1, \ldots, X_n) = \psi_i(X_1, \ldots, X_n)$ using only ultimately periodic constants and addition the problem of having a unique solution is Π_2-complete*

Theorem 12.3. *The problem whether a system $\varphi_i(X_1, \ldots, X_n) = \psi_i(X_1, \ldots, X_n)$ over sets of natural numbers has finitely many solutions is Σ_3-complete. Its Σ_3-hardness is maintained in the case of a single equation using one variable and the operations of union (intersection) and addition.*

Proof. The problem is in Σ_3 for language equations [42].

To prove Σ_3-hardness, consider the co-finiteness problem for Turing machines, which is stated as "Given a TM T working on natural numbers, determine whether $\mathbb{N} \setminus S(T)$ is finite", which is known to be Σ_3-complete [49, Cor. 14-XVI]. Given M, use Lemma 6.3 to construct the system of equations with the set of solutions $\{(S, f_1(S), \ldots, f_k(S)) \mid S(T) \subseteq S\}$. This set is finite if and only if $\mathbb{N} \setminus S(T)$ is finite.

Theorem 9.2 is applied to this system of equations and a single equation with one variable is obtained. Since solutions of the constructed equation are in bijective correspondence with the solutions of the original system, testing whether this single equation has finitely many solution is Σ_3-hard as well. This completes the reduction. □

Corollary 12.3. *For a system $\varphi_i(X_1, \ldots, X_n) = \psi_i(X_1, \ldots, X_n)$ using only ultimately periodic constants and addition testing whether it has finitely many solutions is Σ_3-complete*

Bibliography

[1] J. Autebert, J. Berstel, L. Boasson, “Context-free languages and pushdown automata”, in: Rozenberg, Salomaa (Eds.), *Handbook of Formal Languages*, Vol. 1, Springer-Verlag, 1997, 111–174.

[2] J. W. Backus, “The syntacs and semantics of the proposed of the proposed international algebraic language of the Zurich ACM-GAMM conference” Proceedings of the International Conference on Information Processing, UNESCO, 1959, 125–132.

[3] J. W. Backus, F. L. Bauer, J. Green, C. Katz, J. McCarthy, A. J. Perlis, H. Rutishauser, K. Samelson, B. Vauquois, J. H. Wegstein, A. van Wijngaarden, M. Woodger, “Revised report on the algorithm language ALGOL 60”, *Communications of ACM*, 6:1 (1963) 1–17.

[4] R. P. Brent, L. M. Goldschlager, “A parallel algorithm for context-free parsing”, *Australian Computer Science Communications*, 6:7 (1984), 7.1–7.10.

[5] H.-G. Breunig, “The complexity of membership problems for circuits over sets of positive numbers”, *Fundamentals of Computation Theory* (FCT 2007, Budapest, Hungary, August 27–30, 2007), LNCS 4639, 125–136.

[6] J.-Y. Cai, D. Sivakumar, “Sparse hard sets for P: resolution of a conjecture of Hartmanis”. *Journal of Computer and System Sciences*, 58:2 (1999), 280–296.

[7] A. K. Chandra, D. C. Kozen, L. J. Stockmeyer, “Alternation”, *Journal of the ACM*, 28:1 (1981), 114–133.

[8] W. Charatonik, “Set constraints in some equational theories”, *Information and Computation*, 142:1 (1998), 40–75.

[9] N. Chomsky, “Three models for the description of language” *IRE Transactions on Information Theory*, 2:3 (1956), 113–124.

[10] J. H. Conway, *Regular Algebra and Finite Machines*, Chapman and Hall, 1971.

[11] K. Culik II, J. Gruska, A. Salomaa, "Systolic trellis automata", I and II, *International Journal of Computer Mathematics*, 15 (1984), 195–212, and 16 (1984), 3–22.

[12] K. Culik II, J. Gruska, A. Salomaa, "Systolic trellis automata: stability, decidability and complexity", *Information and Control*, 71 (1984), 218–230.

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

[14] C. Glaßer, K. Herr, C. Reitwießner, S. D. Travers, M. Waldherr, "Equivalence problems for circuits over sets of natural numbers", *Computer Science in Russia* (CSR 2007, Ekaterinburg, Russia, September 3–7, 2007), LNCS 4649, 127–138.

[15] J. Gruska, "Descriptional complexity of context-free languages", *Mathematical Foundations of Computer Science* (MFCS 1973, Strbské Pleso, Czechoslovakia, September 3–8, 1973), Mathematical Institute of the Slovak Academy of Sciences, 71–83.

[16] J. Hartmanis, "Context-free languages and Turing machine computations", *Proceedings of Symposia in Applied Mathematics*, Vol. 19, AMS, Providence, RI, 1967, 42–51.

[17] D. T. Huynh, "Commutative grammars: the complexity of uniform word problems", *Information and Control*, 57:1 (1983), 21–39.

[18] O. H. Ibarra, S. M. Kim, "Characterisations and computational complexity of systolic trellis automata", *Theoretical Computer Science*, 29 (1984), 123–153.

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

[20] A. Jeż, A. Okhotin, "Complexity of equations over sets of natural numbers", *25th Annual Symposium on Theoretical Aspects of Computer Science* (STACS 2008, Bordeaux, France, 21–23 February, 2008), 373–383, to appear in *Theory of Computing Systems*, .

[21] A. Jeż, A. Okhotin, "On the computational completeness of equations over sets of natural numbers" *35th International Colloquium on Automata, Languages and Programming* (ICALP 2008, Reykjavik, Iceland, July 7–11, 2008), LNCS 5126, 63–74.

[22] A. Jeż, A. Okhotin, "Equations over sets of natural numbers with addition only", *STACS 2009* (Freiburg, Germany, 26–28 February, 2009), 577–588.

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

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

[25] A. Jeż, A. Okhotin, "Univariate equations over sets of natural numbers", *Fundamenta Informaticae*, accepted for publication.

[26] G. Jirásková, A. Okhotin, "Nondeterministic state complexity of positional addition", *DCFS 2009*.

[27] N. D. Jones, W. T. Laaser, "Complete problems for deterministic polynomial time", *Theoretical Computer Science*, 3:1 (1976), 105–117.

[28] S. C. Kleene, *Introduction to metamathematics*, North-Holland, Amsterdam, (1952).

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

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

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

[32] M. Lohrey, "Word problems and membership problems on compressed words", *SIAM Journal on Computing*, 35:5 (2006), 1210–1240.

[33] N. Markey, Ph. Schnoebelen, "A PTIME-complete matching problem for SLP-compressed words", *Information Processing Letters*, 90:1 (2004), 3–6.

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

[35] M. Ogihara, "Sparse hard sets for P yield space-efficient algorithms", *Chicago Journal of Theoretical Computer Science*, 1996, article 2.

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

[37] A. Okhotin, "A recognition and parsing algorithm for arbitrary conjunctive grammars", *Theoretical Computer Science*, 302 (2003), 365–399.

[38] A. Okhotin, "The hardest linear conjunctive language", *Information Processing Letters*, 86:5 (2003), 247–253.

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

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

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

[42] A. Okhotin, "Strict language inequalities and their decision problems", *Mathematical Foundations of Computer Science* (MFCS 2005, Gdańsk, Poland, August 29–September 2, 2005), LNCS 3618, 708–719.

[43] A. Okhotin, "Language Equations with Symmetric Difference", *First International Computer Science Symposium in Russia* (CSR 2006, St. Petersburg, Russia, June 8–12, 2006) LNCS 3967, 292–303

[44] A. Okhotin, "Nine open problems for conjunctive and Boolean grammars", *Bulletin of the EATCS*, 91 (2007), 96–119.

[45] A. Okhotin, "Homomorphisms Preserving Linear Conjunctive Languages", *Journal of Automata, Languages and Combinatorics*, 13:3-4 (2008), 299–305.

[46] A. Okhotin, P. Rondogiannis, "On the expressive power of univariate equations over sets of natural numbers", *IFIP Intl. Conf. on Theoretical Computer Science* (TCS 2008, Milan, Italy, 8–10 September, 2008), IFIP vol. 273, 215–227.

[47] A. Okhotin, "Decision problems for language equations", *Journal of Computer and System Sciences*, to appear.

[48] W. Plandowski, W. Rytter, "Complexity of language recognition problems for compressed words", in: J. Karhumäki, H. A. Maurer, G. Păun, G. Rozenberg (Eds.), *Jewels are Forever*, Springer, 1999, 262–272.

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

[50] W. Rytter, "On the recognition of context-free languages", *Fundamentals of Computation Theory* (FCT 1985, Cottbus, Germany), LNCS 208, 315–322.

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

[52] A. Tarski, "A lattice theoretical fixpoint theorem and its applications", *Pacific Journal of Mathematics*, 5 (1955), 285–310.

[53] Véronique Terrier "On Real Time One-Way Cellular Array", *Theoretical Computer Science*, 141:1&2 (1995), 331–335.

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

[55] D. Wotschke, personal communication to A. Okhotin, August 2000.

[56] K. Yang, "Integer circuit evaluation is PSPACE-complete", *Journal of Computer and Systems Sciences*, 63:2 (2001), 288–303.