Journal of Computer and System Sciences 82 (2016) 1007–1019

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Equations over sets of integers with addition only ☆

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

[a] Institute of Computer Science, University of Wrocław, Poland
[b] Department of Mathematics and Statistics, University of Turku, Finland

ARTICLE INFO

Article history:
Received 18 October 2010
Received in revised form 19 November 2014
Accepted 2 February 2016
Available online 17 March 2016

Keywords:
Language equations
Unary languages
Concatenation
Computability

ABSTRACT

Systems of equations of the form $X = Y + Z$ and $X = C$ are considered, in which the unknowns are sets of integers, the plus operator denotes element-wise sum of sets, and C is an ultimately periodic constant. For natural numbers, such equations are equivalent to language equations over a one-symbol alphabet using concatenation and regular constants. It is shown that such systems are computationally universal: for every recursive (r.e., co-r.e.) set $S \subseteq \mathbb{N}$ there exists a system with a unique (least, greatest) solution containing a component T with $S = \{n \mid 16n + 13 \in T\}$. Testing solution existence for these equations is Π^0_1-complete, solution uniqueness is Π^0_2-complete, and finiteness of the set of solutions is Σ^0_3-complete. A similar construction for integers represents any hyper-arithmetical set $S \subseteq \mathbb{Z}$ by a set $T \subseteq \mathbb{Z}$ satisfying $S = \{n \mid 16n \in T\}$, whereas testing solution existence is Σ^1_1-complete.

1. Introduction

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

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

and using union, concatenation and singleton constants, define the semantics of the context-free grammars [2]. If intersection is also allowed, such equations characterise an extension of the context-free grammars known as *conjunctive grammars* [14], which have an increased expressive power and are at the same time notable for preserving efficient parsing algorithms [15,19].

The expressive power of language equations of the general form, that is, with arbitrary expressions on both sides,

☆ The main results of this paper were presented at the STACS 2009 conference held in Freiburg, Germany on February 26–28, 2009, under the title "Equations over sets of natural numbers with addition only". The result in Section 6 was announced at STACS 2010.

* Corresponding author.
E-mail address: aje@cs.uni.wroc.pl (A. Jeż).

http://dx.doi.org/10.1016/j.jcss.2016.02.003

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

was determined by Okhotin [18,17], who proved that a language is representable by a unique solution of a system with concatenation, Boolean operations and singleton constants if and only if this language is recursive. Further characterisation of recursively enumerable (r.e.) and co-r.e. sets was given in terms of *least* and *greatest* (with respect to component-wise inclusion) solutions of such systems [18]. The same expressive power is attained using concatenation with constants and union [17]. It was subsequently discovered that language equations can be computationally universal even without any Boolean operations: Kunc [9] constructed a finite language $L \subseteq \{a,b\}^*$, for which the greatest solution of a language equation $LX = XL$ is Π^0_1-hard (that is, hard for co-r.e. sets). This paper establishes a similar result in the seemingly trivial case of a one-symbol alphabet.

Unary languages, defined over an alphabet $\{a\}$, form a theoretically important special class of formal languages. It is well-known that context-free grammars over this alphabet generate only regular languages [2]. The first example of a language equation over a unary alphabet with a non-regular unique solution was constructed by Leiss [12]: this was an equation $X = \varphi(X)$, with φ containing concatenation, complementation and constant $\{a\}$. Turning to systems with union, intersection and concatenation—in other words, conjunctive grammars—the question of whether these grammars can generate any non-regular languages had been a long-standing open problem [14], until Jeż [3] constructed a conjunctive grammar generating $\{a^{4^n} \mid n \geqslant 0\}$. The ideas of this example were used by Jeż and Okhotin [4] to establish some general results on the expressive power of these equations; in particular, they are capable of having EXPTIME-complete solutions [5]. For systems of the general form (**) using concatenation and union, it has recently been shown by the authors [6] that they are computationally complete; on a higher level, this result can be considered a remake of the computational completeness proof for language equations [18], but carrying out that proof using only sets of numbers required much more difficult constructions.

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

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

an operation that has been a subject of much study in number theory and combinatorics [24]. Computational complexity of expressions and circuits over sets of natural numbers with addition and different sets of Boolean operations has first been investigated by Stockmeyer and Meyer [23] and then extensively studied by McKenzie and Wagner [13]. A similar study for expressions and circuits over sets of both positive and negative integers was done by Travers [25].

As compared to these circuits, equations over sets of numbers are a more general formalism already in the resolved case (*), where the definition of a set may recursively refer to the same set. The exact expressive power of equations over sets of natural numbers with addition and Boolean operations was determined in the aforementioned work on language equations over a unary alphabet [3–6]: they are computationally complete. Equations over sets of integers were recently investigated by the authors [7], and proved to define exactly the *hyper-arithmetical sets*, which is a class situated at the bottom of the analytical hierarchy and properly containing the sets representable in first-order Peano arithmetic.

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

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

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

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

with regular constants $K, L \subseteq a^*$. This is the ultimately simplest case of language equations, and at the first glance it seems out of question that such equations could have any non-trivial unique solutions (and considering least or greatest solutions makes little difference). Probably for that reason no one has ever proclaimed their expressive power to be an open problem. However, as proved in this paper, these equations can have not only non-periodic unique solutions, but in fact are computationally universal. Furthermore, their main decision problems are as hard as similar problems for language equations over multiple-letter alphabets and using all Boolean operations [18,17].

The new results are directly based on the authors' recent proof of the computational completeness of equations over sets of numbers with addition and union [6], though it is established using completely different methods. The idea is to take an arbitrary system using addition and union and *encode* it in another system using addition only. The solutions of the two systems will not be identical, but there will be a bijection σ between them, which represents every number n of the encoded set as the number $16n + 13$ in the encoding. The encoding of a set also contains other numbers that do not depend on the encoded set and form a periodic structure.

The first goal is to guarantee that each solution of the constructed system consists only of valid encodings of sets. To this end, a special equation is constructed, which is satisfied exactly by those sets of natural numbers that are valid encodings