Approaching Arithmetic Theories with Finite-State Automata*

Christoph Haase[0000−0002−5452−936X]

University College London, UK
c.haase@ucl.ac.uk

Abstract. The automata-theoretic approach provides an elegant method for deciding linear arithmetic theories. This approach has recently been instrumental for settling long-standing open problems about the complexity of deciding the existential fragments of Büchi arithmetic and linear arithmetic over p-adic fields. In this article, which accompanies an invited talk, we give a high-level exposition of the NP upper bound for existential Büchi arithmetic, obtain some derived results, and further discuss some open problems.

Keywords: Presburger arithmetic · Büchi arithmetic · reachability · automatic structures

1 Introduction

Finite-state automata over finite and infinite words provide an elegant method for deciding linear arithmetic theories such as Presburger arithmetic or linear real arithmetic. Automata-based decision procedures for arithmetic theories have also been of remarkable practical use and have been implemented in tools such as LASH [16] or TaPAS [10]. However, understanding the algorithmic properties of automata-based decision procedures turned out to be surprisingly difficult and tedious, see e.g. [3,19,9,6]. It took, for instance, 50 years to show that Büchi's seminal approach for deciding Presburger arithmetic using finite-state automata runs in triply-exponential time and thus matches the upper bound of quantifier-elimination algorithms [5,6]. Given this history, it is not surprising that, until recently, the author was of the opinion that automata should better be avoided when attempting to prove complexity upper bounds for arithmetic theories.

* This work is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 852769, ARiAT).

The author's opinion drastically changed when appealing to automata-based approaches recently allowed for settling long-standing open problems about the complexity of the existential fragments of Büchi arithmetic and linear arithmetic over p-adic fields, which were both shown NP-complete [8]. The NP upper bounds are the non-trivial part in those results, since, unlike, for instance, in existential Presburger arithmetic, the encoding of smallest solutions can grow super-polynomially. The key result underlying both NP upper bounds is that given two states of a finite-state automaton encoding the set of solutions of a system of linear Diophantine equations, one can decide whether one state reaches the other in NP in the size of the encoding of the system (and without explicitly constructing the automaton).

This article gives a high-level yet sufficiently detailed outline of how the NP upper bound for existential Büchi arithmetic can be obtained. We subsequently show how the techniques used for the NP upper bound can be applied in order to show decidability and complexity results for an extension of Presburger arithmetic with valuation constraints. Those results are somewhat implicit in [8] but seem worthwhile being explicated in written. We conclude with some observations and discussion of open problems.

2 Preliminaries

We denote by $\mathbb{R}$ the real numbers, by $\mathbb{R}_+$ the non-negative reals, by $\mathbb{Q}$ the rational numbers, by $\mathbb{Z}$ the integers, by $\mathbb{N}$ the non-negative integers, and by $\mathbb{N}_+$ the positive integers. For integers $a < b$, we write $[a, b]$ for the set $\{a, a+1, \ldots, b\}$. All numbers in this article are assumed to be encoded in binary. Given a matrix $\mathbf{A} \in \mathbb{Z}^{m\times n}$ with components $a_{ij} \in \mathbb{Z}$, $1 \leq i \leq m$, $1 \leq j \leq n$, the $(1,\infty)$-norm of $\mathbf{A}$ is $\|\mathbf{A}\|_{1,\infty} := \max_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|$. For $\boldsymbol{v} \in \mathbb{R}^n$, we just write $\|\boldsymbol{v}\|_\infty$.

2.1 Büchi arithmetic

Throughout this article, let $p \geq 2$ be a base. Recall that Presburger arithmetic is the first-order theory of the structure $\langle \mathbb{N}, 0, 1, + \rangle$. Büchi arithmetic is the first-order theory of the structure $\langle \mathbb{N}, 0, 1, +, V_p \rangle$ obtained from endowing Presburger arithmetic with a functional binary predicate $V_p \subseteq \mathbb{N} \times \mathbb{N}$ such that $V_p(x, u)$ evaluates to true if and only if u is the largest power of p dividing x without remainder. This definition leaves the case $x = 0$ ambiguous. A sensible approach would be to introduce a special value ∞ and to assert $V_p(0, \infty)$ to hold, many authors choose to assert $V_p(0, 1)$, see e.g. [4]. However, the particular choice has no impact on the sets of naturals definable in Büchi arithmetic.

Atomic formulas of Büchi arithmetic are either linear equations $\boldsymbol{a} \cdot \boldsymbol{x} = c$ or Büchi predicates $V_p(x, u)$. Note that the negation of $\boldsymbol{a} \cdot \boldsymbol{x} = c$ is equivalent to $\boldsymbol{a} \cdot \boldsymbol{x} < c \vee \boldsymbol{a} \cdot \boldsymbol{x} > c$. Since we interpret variables over the non-negative integers, we have $\boldsymbol{a} \cdot \boldsymbol{x} > c \equiv \exists y\, \boldsymbol{a} \cdot x - y = c + 1$. Consequently, we can, with no loss of generality, assume that negation symbols only occur in front of V_p predicates. Now if we consider a negated literal $\neg V_p(x, u)$, we have that $\neg V_p(x, u)$ evaluates to true if and only if either

(i) u is a power of p but not the largest power of p dividing x; or
(ii) u is not a power of p.

The case (i) can easily be dealt with, as it is definable by

$$\exists v\, V_p(u,u) \wedge V_p(x,v) \wedge \neg(u = v)$$

Moreover, $\neg V_p(u,u)$ asserts that u is not a power of p. Thus, we may, without loss of generality, assume that quantifier-free formulas of Büchi arithmetic are positive Boolean combinations of atomic formulas $\boldsymbol{a}\cdot\boldsymbol{x} = c$, $V_p(x,u)$ and $V_p(u,u)$.

2.2 Finite-state automata and p-automata

It is well known that Büchi arithmetic can elegantly be decided using finite-state automata, see [2] for a detailed overview over this approach. In this section, we give a generic definition of deterministic automata and then define p-automata which are used for deciding Büchi arithmetic.

Definition 1. *A* deterministic automaton *is a tuple* $A = (Q, \Sigma, \delta, q_0, F)$*, where*

- Q *is a set of* states*,*
- Σ *is a finite* alphabet*,*
- $\delta\colon Q \times \Sigma \to Q \cup \{\bot\}$*, where* $\bot \notin Q$*, is the* transition function*,*
- $q_0 \in Q$ *is the* initial state*, and*
- $F \subseteq Q$ *is the set of* final states*.*

Note that this definition allows automata to have infinitely many states and to have partially defined transition functions (due to the presence of $\bot$ in the codomain of δ).

For states $q, r \in Q$ and $u \in \Sigma$, we write $q \xrightarrow{u} r$ if $\delta(q,u) = r$, and extend $\to$ inductively to finite words such that for $w \in \Sigma^*$ and $u \in \Sigma$, $q \xrightarrow{w\cdot u} r$ if there is $s \in Q$ such that $q \xrightarrow{w} s \xrightarrow{u} r$. Whenever $q \xrightarrow{w} r$, we say that A *has a run on* w *from* q *to* r. We write $q \xrightarrow{*} r$ if there is some $w \in \Sigma^*$ such that $q \xrightarrow{w} r$.

A *finite-state automaton* A is a deterministic automaton with a finite set of states that accepts finite words. The *language of* A is defined as

$$L(A) \stackrel{\text{def}}{=} \{w \in \Sigma^* : q_0 \xrightarrow{w} q_f, q_f \in F\}.$$

We now introduce p-automata, which are deterministic automata whose language encodes a set of non-negative integers in base p. Furthermore, we recall the construction of the key gadget underlying the automata-based decision procedures for Büchi arithmetic which provides a representation of the set of non-negative integer solutions of a system of linear equations as the language of a finite-state p-automaton

Formally, a *p-automaton* is a deterministic automaton over an alphabet $\Sigma_p^n := \{0, 1, \ldots, p-1\}^n$ for some nonnegative integer n. A finite word over the alphabet Σ_p^n can naturally be seen as encoding an n-tuple of nonnegative

integers in base p. There are two possible encodings: least significant digit first and most-significant digit first. We only consider the latter *msd-first encoding*, in which the most significant digit appears on the left. Formally, given a word $w = \boldsymbol{u}_0 \cdots \boldsymbol{u}_k \in (\Sigma_p^n)^*$, we define $[\![w]\!] \in \mathbb{N}^n$

$$[\![w]\!] := \sum_{j=0}^{k} p^{k-j} \cdot \boldsymbol{u}_j \,.$$

Note that for $w = \varepsilon$, the empty word, we have $[\![w]\!] = \mathbf{0}$.

A system S of *linear Diophantine equations* has the form $S\colon \mathbf{A} \cdot \boldsymbol{x} = \boldsymbol{c}$, where $\mathbf{A}$ is an $m \times n$ matrix with integer coefficients, $\boldsymbol{c} \in \mathbb{Z}^m$, and $\boldsymbol{x} = (x_1, \ldots, x_n)^\top$ is a vector of variables taking values in the nonnegative integers. We write $[\![S]\!] := \{\boldsymbol{u} \in \mathbb{N}^n : \mathbf{A} \cdot \boldsymbol{u} = \boldsymbol{c}\}$ for the set of all nonnegative integer solutions of S. We denote by $\langle S \rangle$ the size of the encoding of S, i.e., the number of symbols required to represent S assuming binary encoding of all numbers.

Following Wolper and Boigelot [19], we define a p-automaton whose language is the msd-first encoding all nonnegative integer solutions of systems of linear equations.

Definition 2. *Let $S\colon \mathbf{A} \cdot \boldsymbol{x} = \boldsymbol{c}$ be a system of linear equations with integer coefficients such that $\mathbf{A}$ has dimension $m \times n$. Corresponding to S, we define a p-automaton $A(S) := (Q, \Sigma_p^n, \delta, \boldsymbol{q}_0, F)$ such that*

- $Q = \mathbb{Z}^m$,
- $\delta(\boldsymbol{q}, \boldsymbol{u}) = p \cdot \boldsymbol{q} + \mathbf{A} \cdot \boldsymbol{u}$ *for all* $\boldsymbol{q} \in Q$ *and* $\boldsymbol{u} \in \Sigma_p^n$,
- $\boldsymbol{q}_0 = \mathbf{0}$, *and*
- $F = \{\boldsymbol{c}\}$.

Although the automaton $A(S)$ has infinitely many states, it defines a regular language since there are only finitely many *live states*, i.e., states that can reach the set F of accepting states. The reason is that no state $\boldsymbol{q} \in Q$ such that $\|\boldsymbol{q}\|_\infty > \|\mathbf{A}\|_{1,\infty}$ and $\|\boldsymbol{q}\|_\infty > \|\boldsymbol{c}\|_\infty$ can reach an accepting state [1,8], and hence Q can be restricted to a finite number of states. A rough upper bound on the number $\#Q$ of states of $A(S)$ is

$$\#Q \leq 2^m \cdot \max(\|\mathbf{A}\|_{1,\infty}, \|\boldsymbol{c}\|_\infty)^m \,, \tag{1}$$

where m is the number of equations in the system S [19,8].

A key reachability property of the automaton $A(S)$ is the following: Let $\boldsymbol{q}, \boldsymbol{r} \in \mathbb{Z}^m$ be states of $A(S)$. Then for all $k \in \mathbb{N}$ and words $w \in (\Sigma_p^n)^k$ we have

$$\boldsymbol{q} \xrightarrow{w} \boldsymbol{r} \iff \boldsymbol{r} = p^k \cdot \boldsymbol{q} + \mathbf{A}\,[\![w]\!] \tag{2}$$

From this characterization, it follows that the language of $A(S)$ is an msd-first encoding of the set of solutions of the system $\mathbf{A} \cdot \boldsymbol{x} = \boldsymbol{c}$. Indeed, choosing $\boldsymbol{q}$ as $\mathbf{0}$ and the final state $\boldsymbol{c}$ as $\boldsymbol{r}$, we have that $\mathbf{0} \xrightarrow{w} \boldsymbol{c}$ if and only if $\mathbf{A} \cdot [\![w]\!]_m = \boldsymbol{c}$.

If we wish to emphasize the underlying system S of linear Diophantine equations of a p-automaton $A(S)$ we annotate the transition relation with the subscript S and, e.g., write $\boldsymbol{q} \xrightarrow{*}_S \boldsymbol{r}$.

2.3 Semi-linear sets

Given a *base vector* $\boldsymbol{b} \in \mathbb{N}^n$ and a finite set of period vectors $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_m\} \subseteq \mathbb{N}^n$, define

$$L(\boldsymbol{b}, P) := \left\{ \boldsymbol{b} + \sum_{i=1}^{m} \lambda_i \cdot \boldsymbol{p}_i : \lambda_i \in \mathbb{N} \right\}.$$

We call $L(\boldsymbol{b}, P)$ a *linear set* and we say that a subset of $\mathbb{N}^n$ is *semi-linear* if it can be written as a finite union of linear sets. It is well-known that the set of nonnegative integer solutions of a system of linear Diophantine equations is a semi-linear set [7]. Also note that a linear set is definable by a formula of existential Presburger arithmetic of linear size.

A special subclass of semi-linear sets are ultimately periodic sets, which are an equivalent presentation of semi-linear sets in dimension one. A set $M \subseteq \mathbb{N}$ is *ultimately periodic* if there is a threshold $t \in \mathbb{N}$ and a period $\ell \in \mathbb{N}$ such that for all $a, b \in \mathbb{N}$ with $a, b \geq t$ and $a \equiv b \bmod \ell$ we have $a \in M$ if and only if $b \in M$.

3 Existential Büchi arithmetic

One of the main results of [8] is that deciding existential formulas of Büchi arithmetic is NP-complete. A main obstacle is that the magnitude of satisfying variable assignments may grow super-polynomially. It is known that for infinitely many primes q the multiplicative order $\mathrm{ord}_q(2)$ of 2 modulo q is at least $\sqrt{q}$ [13]. For such a prime the predicate *x is a strictly positive power of* 2 *that is congruent to 1 modulo q* can easily be expressed as a formula of existential Büchi arithmetic of base 2:

$$\Phi(x) \stackrel{\text{def}}{=} \exists y\, x > 1 \wedge V_2(x, x) \wedge x = q \cdot y + 1$$

Observe that $\Phi(x)$ has a constant number of literals and that its length linear in the bit-length of q, while the smallest satisfying assignment is $x = 2^{\mathrm{ord}_q(2)}$. Thus satisfying assignments in existential Büchi arithmetic may have super-polynomial bit-length in the formula size, even for a fixed base and a fixed number of literals. This rules out the possibility of showing NP membership by a non-deterministic guess-and-check algorithm. We nevertheless have the following theorem:

Theorem 1 ([8]). *Existential Büchi arithmetic is NP-complete.*

Existential Büchi arithmetic inherits the NP lower bound from integer programming when the number of variables is not fixed. While existential Presburger arithmetic can be decided in polynomial time when the number of variables is fixed [15], showing such a result for Büchi arithmetic would likely require major breakthroughs in number theory, even when fixing the number of literals. Given $a, b, c \in \mathbb{N}$, we can express discrete logarithm problems of the kind, *does there exist $x \in \mathbb{N}$ such that $a^x \equiv b \bmod c$*, in a similar way as above:

$$\exists x\, \exists y\, V_a(x, x) \wedge x = c \cdot y + b$$

Such discrete logarithm problems are believed to possibly be even more difficult than those underlying the Diffie-Hellman key exchange [14]. Of course, it may well be that existential Büchi arithmetic with a fixed number of variables (and even literals) is NP-hard. For instance, existential Presburger arithmetic with a full divisibility predicate is NP-hard already for a fixed number of variables and literals [11], shown via a reduction from a certain NP-complete problem involving a special class of quadratic congruences [12].

We now give an exposition of the NP upper bound of Theorem 1 developed in [8]. It clearly suffices to only consider quantifier-free formulas. Let $\Phi(\boldsymbol{x})$ be a quantifier-free formula of Büchi arithmetic, and let us first consider the special case of a system of linear Diophantine equations together with a single V_p assertion

$$\Phi(\boldsymbol{x}) \stackrel{\text{def}}{=} A \cdot \boldsymbol{x} = \boldsymbol{c} \wedge V_p(x,u),$$

where x and u are variables occurring in $\boldsymbol{x}$. From Section 2.2, we know that we can construct a p-automaton $A(S)$ whose language encodes all solutions of $S\colon A \cdot \boldsymbol{x} = \boldsymbol{c}$. A key insight enabling showing decidability of Büchi arithmetic is that the set of solutions of $V_p(x,u)$ for $x > 0$ can be encoded by a regular language over the alphabet $\Sigma_p \times \Sigma_p$:

$$\begin{bmatrix} \Sigma_p \\ 0 \end{bmatrix}^* \begin{bmatrix} \Sigma_p \setminus \{0\} \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}^*$$

Thus, in order to decide whether $\Phi(\boldsymbol{x})$ is satisfiable, we can check whether we can find a run through the automaton $A(S)$ that can be partitioned into three parts. In the first part, x can have any digit and u has only zeros as digits. The second part is a single transition in which x can have any non-zero digit and u has digit one, and in the third part both x and u have digits zero.

To make this argument more formal, it will be useful to introduce a mild generalization of the reachability relation for p-automata. Suppose we are given a system of linear equations $S\colon \mathbf{A}\cdot\boldsymbol{x} = \boldsymbol{c}$ and an additional system of constraints $T\colon \mathbf{B}\cdot\boldsymbol{x} = \boldsymbol{d}$. For all pairs of states $\boldsymbol{q}, \boldsymbol{r}$ of the automaton $A(S)$, write $\boldsymbol{q} \xrightarrow{w}_{S[T]} \boldsymbol{r}$ if $\boldsymbol{q} \xrightarrow{w}_S \boldsymbol{r}$ and $\mathbf{B} \cdot [\![w]\!] = \boldsymbol{d}$. Plainly $\boldsymbol{q} \xrightarrow{w}_{S[T]} \boldsymbol{r}$ if and only if

$$\begin{pmatrix} \boldsymbol{q} \\ \mathbf{0} \end{pmatrix} \xrightarrow{w}_{S \wedge T} \begin{pmatrix} \boldsymbol{r} \\ \boldsymbol{d} \end{pmatrix},$$

where $S \wedge T$ is the system of equations

$$S \wedge T\colon \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix} \boldsymbol{x} = \begin{pmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{pmatrix}.$$

With the new notation at hand, the observations made above now enable us to reduce satisfiability of $\Phi(\boldsymbol{x})$ to three reachability queries in p-automata: $\Phi(\boldsymbol{x})$ is satisfiable if and only if there are states $\boldsymbol{d}$ and $\boldsymbol{e}$ of $A(S)$, and $a \in \Sigma_p \setminus \{0\}$ such that

$$\mathbf{0} \xrightarrow{*}_{S[u=0]} \boldsymbol{d} \rightarrow_{S[x=a,u=1]} \boldsymbol{e} \xrightarrow{*}_{S[x=u=0]} \boldsymbol{c}. \tag{3}$$

Note that by (1), the encoding of the binary representation of the states $\boldsymbol{d}$ and $\boldsymbol{e}$ of $A(S)$ is polynomial in the encoding of S, and hence both states can be guessed in NP.

The reduction to reachability queries in p-automata is easily seen to generalize to the case where we have an arbitrary number k of constraints $V_p(x_i, u_i)$ in $\Phi(\boldsymbol{x})$. To check satisfiability, all we have to do is to guess a relative order between the u_i, $a_i \in \Sigma_p \setminus \{0\}$, states $\boldsymbol{d}_i$ and $\boldsymbol{e}_i$ of $A(S)$, resulting in $O(k)$ reachability queries in p-automata. We illustrate the reachability queries for the case in which $u_i > u_{i+1}$ for all $1 \leq i \leq k$, the remaining cases follow analogously:

$$\begin{aligned}\boldsymbol{0} &\xrightarrow{*}_{S[u_1,\ldots,u_k=0]} \boldsymbol{d}_1 \rightarrow_{S[x_1=a_1,u_1=1,u_2,\ldots,u_k=0]} \boldsymbol{e}_1 \\ &\xrightarrow{*}_{S[x_1,u_1,\ldots,u_k=0]} \boldsymbol{d}_2 \rightarrow_{S[x_2=a_2,u_2=1,x_1,u_1,u_3,\ldots u_k=0]} \boldsymbol{e}_2 \xrightarrow{*}_{S[x_1,x_2,u_1,\ldots,u_k=0]} \cdots \\ &\cdots \boldsymbol{d}_k \rightarrow_{S[x_k=a_k,u_k=1,x_1,\ldots,x_{k-1},u_1,\ldots,u_{k-1}=0]} \boldsymbol{e}_k \xrightarrow{*}_{S[x_1,\ldots,x_k,u_1,\ldots,u_k=0]} \boldsymbol{c} \quad (4)\end{aligned}$$

Finally, we observe that the set of solutions of a literal $\neg V_p(u, u)$, stating that u is not a power of p, is encoded by the regular language given by the following regular expression:

$$\overline{0^*10^*} \equiv 0^*(\Sigma_p \setminus \{0,1\})\Sigma_p^* \,+\, 0^*10^*(\Sigma_p \setminus \{0\})\Sigma_p^*$$

Observe that this regular expression induces a decomposition similar to (3). Hence, we can non-deterministically polynomially reduce deciding conjunctions of the form

$$\mathbf{A} \cdot \boldsymbol{x} = \boldsymbol{c} \wedge \bigwedge_{i \in I} V_p(x_i, u_i) \wedge \bigwedge_{j \in J} \neg V_p(u_j, u_j) \quad (5)$$

for finite index sets I, J to a linear number of state-to-state reachability queries in p-automata implicitly given by systems of linear Diophantine equations. We now invoke the following theorem:

Theorem 2 ([8]). *Deciding state-to-state reachability in a p-automaton $A(S)$ given by a system of linear Diophantine equations S is in NP (with respect to the encoding of S).*

In particular, the NP upper bound does *not* require the explicit construction of $A(S)$. By application of this result and the arguments above, the NP upper bound for existential Büchi arithmetic follows. Given a quantifier-free formula $\Phi(\boldsymbol{x})$, as discussed in Section 2.1, we can assume that Φ is a positive Boolean combination of literals $\boldsymbol{a} \cdot \boldsymbol{x} = c$, $V_p(x, u)$ and $\neg V_p(u, u)$. Hence we can guess in NP a clause of the disjunctive normal form of Φ, which is of the form (5), and in turn check in NP a series of guessed reachability queries in p-automata induced by the guessed clause.

We close this section with a brief discussion of the main ideas behind the NP upper bound of Theorem 2. The first observation is that reachability in p-automata reduces to satisfiability in a certain class of systems of linear-exponential Diophantine equations. From (2), we can deduce that for a word $w \in (\Sigma_p^n)^k$,

$$\boldsymbol{q} \xrightarrow{w} \boldsymbol{r} \iff \boldsymbol{r} = p^k \cdot \boldsymbol{q} + \mathbf{A} \cdot [\![w]\!] \iff \boldsymbol{r} = p^k \cdot \boldsymbol{q} + \mathbf{A} \cdot \boldsymbol{x}, \|\boldsymbol{x}\|_\infty < p^k.$$

Let $\boldsymbol{x} = (x_1, \ldots, x_n)^\top$, replacing p^k by a fresh variable y, it follows that $\boldsymbol{q} \xrightarrow{*} \boldsymbol{r}$ if and only if the following system of linear Diophantine inequalities has a solution in which y is a power of p:

$$\boldsymbol{r} = y \cdot \boldsymbol{q} + \mathbf{A} \cdot \boldsymbol{x}, x_i < y,\ 1 \leq i \leq n.$$

This is now a problem that is not difficult to decide, since we can guess in NP a linear set $L(\boldsymbol{b}, P) \subseteq \mathbb{N}^m$ with a small description that generates a subset of the set of solutions of this system. Checking whether $L(\boldsymbol{b}, P)$ contains a point in which the y-coordinate is a power of p can easily be done in NP, we refer the reader to [8] for further details.

4 Presburger arithmetic with valuation constraints

The definition of V_p ensures that p-recognizable sets are equivalent to those definable in Büchi arithmetic. Note that it is possible to enrich Presburger arithmetic with an even more general predicate which does, however, not change the definable sets of natural numbers, see e.g. [4, p. 209]. But the predicate V_p also has a close connection to the valuation function $v_p \colon \mathbb{Q} \to \mathbb{Z}$ underlying the definition of the p-adic numbers. Given a prime p and a non-zero rational number x, the p-adic valuation $v_p(x)$ is defined to be the unique integer $e \in \mathbb{Z}$ such that $x = p^e \cdot \frac{a}{b}$ with $a, b \in \mathbb{Z}$ and $p \nmid a, b$. Intuitively $v_p(x)$ is the exponent of the greatest power of p that divides x. Now the p-adic valuation v_p and the V_p predicate of Büchi arithmetic (viewing V_p as a function) are related as follows: for a natural number $n \in \mathbb{N}$ we have $V_p(n) = p^{v_p(n)}$. Thus, we could view $v_p(n)$ as a succinct representation of $V_p(n)$.

In arithmetic theories over p-adic numbers, it is common to consider two-sorted logics with one sort for the p-adic numbers and another sort for the valuation ring $\mathbb{Z}$, together with additional (restricted) arithmetic over the valuation ring, see e.g. [18]. One can naturally transfer this concept to arithmetic theories over numerical domains other than the p-adic numbers. The decompositions established in the previous section together with classical results on finite-state automata then give decidability and complexity results.

As a concrete illustrating example, we introduce in this section Presburger arithmetic with valuation constraints. Since $v_p(n) \in \mathbb{N}$ for all $n \in \mathbb{N}_+$, technically we are not dealing with a multi-sorted logic.[1] We use the following notational convention: a variable x is interpreted as a natural number, and $\overline{x} \stackrel{\text{def}}{=} v_p(x)$ is interpreted as the valuation of x. A formula Φ of *Presburger arithmetic with valuation constraints* is then given by a tuple

$$\Phi = (\Psi(x_1, \ldots, x_n); \Gamma(\overline{x}_1, \ldots, \overline{x}_n)),$$

where both Ψ and Γ are formulas of Presburger arithmetic. We say that Φ is existential if both Ψ and Γ are formulas of existential Presburger arithmetic.

[1] And for brevity, we do not delve into different ways of defining $V_p(0)$, the results given work for any sensible choice of defining $V_p(0)$.

Moreover, Φ is satisfiable with respect to a fixed $p > 1$ given as input whenever we can find a variable assignment $\sigma\colon \{x_1, \ldots, x_n\} \to \mathbb{N}$ such that both $\Psi(\sigma(x_1), \ldots, \sigma(x_n))$ and $\Gamma(v_p(\sigma(x_1)), \ldots, v_p(\sigma(x_n)))$ evaluate to true.

It is not surprising and easy to see that satisfying assignments are not semi-linear since, e.g.,

$$\Phi = (x > 0; \exists y\, \overline{x} = 2y \wedge y > 0)$$

has the set of all positive integers n with $v_p(n)$ even and greater than zero as its set of satisfying assignments, i.e., Φ defines the set $\{p^{2k} \cdot n : k, n \in \mathbb{N}_+, p \nmid n\}$ which, for any base $p > 1$, is obviously not ultimately periodic and hence not semi-linear.

We now show NP-completeness of existential formulas of Presburger arithmetic with valuation constraints from which we can then conclude decidability of the general case. Given $\Phi = (\Psi, \Gamma)$, let us first consider the case in which Ψ is a system of linear Diophantine equations $S\colon A \cdot \boldsymbol{x} = \boldsymbol{c}$ with $\boldsymbol{x} = (x_1, \ldots, x_n)$, and Γ is existential. A solution of S is encoded by a path in $A(S)$ from $\mathbf{0}$ to $\boldsymbol{c}$, and if we assume without loss of generality that $x_i > x_{i+1}$ for all $1 \leq i < n$ then similarly as in (4) we can decompose this path as

$$\begin{aligned}\mathbf{0} \xrightarrow{*} \boldsymbol{d}_1 \to_{S[x_1=a_1]} \boldsymbol{e}_1 \xrightarrow{w_1}_{S[x_1=0]} \boldsymbol{d}_2 \to_{S[x_2=a_2,x_1=0]} \boldsymbol{e}_2 \xrightarrow{w_2}_{S[x_1,x_2=0]} \cdots \\ \cdots \boldsymbol{d}_n \to_{S[x_n=a_n,x_1,\ldots,x_{n-1}=0]} \boldsymbol{e}_n \xrightarrow{w_n}_{S[x_1,\ldots,x_n=0]} \boldsymbol{c} \quad (6)\end{aligned}$$

for some $w_i \in (\Sigma_p^n)^*$ and with all $a_i \neq 0$. Note that this decomposition implies that $v_p(x_n) = |w_n| + 1$, $v_p(x_{n-1}) = |w_n| + |w_{n-1}| + 2$, etc. In particular, each $|w_i|$ is the length of a path between the states $\boldsymbol{e}_i$ and $\boldsymbol{d}_{i+1}$. It is well-known that the set of lengths of paths between two states in a non-deterministic finite-state automaton is semi-linear and that the encoding of each linear set in such a semi-linear set is logarithmic in the number states, see e.g. [17]. Moreover, semi-linear sets are closed under taking finite sums. Recall that by the estimation in Equation (1) the number of states of a p-automaton $A(S)$ is exponentially bounded and that each state has an encoding linear in the encoding of S. It follows that given a decomposition as in (6), we can for each x_i guess in NP a linear set $L(b, P) \subseteq \mathbb{N}$ such that $v_p(x_i) \in L(b, P)$. Also recall from Section 2.3 that a linear set is definable by a formula of existential Presburger arithmetic of linear size. Consequently, we obtain the following non-deterministic polynomial-time algorithm deciding satisfiability of Φ above:

- guess the states occurring in a decomposition of a run from $\mathbf{0}$ to $\boldsymbol{c}$ in $A(S)$ of the form (6) (again note that this does not require constructing $A(S)$);
- from this decomposition, guess linear sets $L(b_i, P_i)$ such that $v_p(x_i) \in L(b_i, P_i)$ for each x_i;
- check whether Γ is satisfiable with each $v_p(x_i)$ constrained to lie in $L(b_i, P_i)$.

If $\Phi = (\Psi, \Gamma)$ is an arbitrary existential formula of Presburger arithmetic with valuation constraints, an NP upper bound also follows: we only need to guess a clause of the disjunctive normal form of Ψ and then proceed as before. The case where Φ is arbitrary obviously reduces to the existential case since Presburger arithmetic has quantifier elimination.

Theorem 3. *Presburger arithmetic with valuation constraints is decidable, and its existential fragment is NP-complete.*

5 Conclusion

This article provided an exposition of the results of [8] together with some results that follow but are not explicated in [8]. We described the proof of NP-completeness of existential Büchi arithmetic and showed how this proof can be applied to obtain decidability of Presburger arithmetic with valuation constraints and NP-completeness of its existential fragment. We close this article with a couple of remarks and open questions for future work:

- There is an analogue of Büchi arithmetic for the reals that was studied by Boigelot, Rassart and Wolper [1]. This analogue builds upon a predicate $X_p \subseteq \mathbb{R}_+ \times \mathbb{Q} \times [0, p-1]$ such that $X_p(x,u,k)$ is true if and only if u is a (possibly negative) integer power of p, and there is an encoding of x such that the digit at the position specified by u is k:

$$X_p(x,u,k) \iff \text{there are } \ell \in \mathbb{Z}, a_\ell, a_{\ell-1}, \ldots \in [0,p-1] \text{ s.t. } x = \sum_{i=\ell}^{-\infty} a_i p^i$$
$$\text{and there is } q \in \mathbb{Z} \text{ s.t. } q \leq \ell, u = p^q \text{ and } a_q = k.$$

  The real analogue of Büchi arithmetic is the first-order theory of the structure $\langle \mathbb{R}_+, 0, 1, +, X_p \rangle$ (*BRW arithmetic* after the authors of [1] for short).[2] Looking at the similarities of the definitions of $X_p(x,u,k)$ and V_p, it seems conceivable that existential BRW arithmetic is also NP-complete, though this is likely more tedious to prove mainly because some real numbers have multiple encodings (e.g., $1.0000\cdots = 0.9999\cdots$).
- Presburger arithmetic with valuation constraints is a powerful logic which can be used to reason about sets of integers which are not semi-linear. Decidability in such contexts is rare, and NP-completeness of its existential fragment means that this logic could potentially find practical applications in areas such as formal verification, as we seemingly can, for instance, express some problems typically arising in bit-vector arithmetic. Generally speaking, what are natural applications of Presburger arithmetic with valuation constraints?
- Is Büchi arithmetic with valuation constraints decidable? It can be derived from the approach presented in Section 4 that this is the case for existential Büchi arithmetic. However, the author is not aware of a quantifier-elimination procedure for Büchi arithmetic that given a formula of Büchi arithmetic allows for obtaining an equivalent formula of *existential* Büchi arithmetic.

[2] For presentational convenience, we chose $\mathbb{R}_+$ as the domain of BRW arithmetic, unlike [1] who actually use $\mathbb{R}$.

– Is existential Büchi arithmetic with a fixed number of variables (and possibly even a fixed number of literals) NP-complete? As discussed in Section 3, showing membership in P would require breakthroughs that currently (and likely over the next decades) seem out of reach, and would moreover break some public key cryptographic systems.

References

1. Boigelot, B., Rassart, S., Wolper, P.: On the expressiveness of real and integer arithmetic automata (extended abstract). In: Automata, Languages and Programming, ICALP. Lect. Notes Comp. Sci., vol. 1443, pp. 152–163. Springer (1998)
2. Boigelot, B., Wolper, P.: Representing arithmetic constraints with finite automata: An overview. In: Logic Programming, ICLP. Lect. Notes Comp. Sci., vol. 2401, pp. 1–19. Springer (2002)
3. Boudet, A., Comon, H.: Diophantine equations, presburger arithmetic and finite automata. In: Trees in Algebra and Programming - CAAP. Lect. Notes Comp. Sci., vol. 1059, pp. 30–43. Springer (1996)
4. Bruyère, V., Hansel, G., Michaux, C., Villemaire, R.: Logic and p-recognizable sets of integers. Bull. Belg. Math. Soc. Simon Stevin **1**(2), 191–238 (1994)
5. Durand-Gasselin, A., Habermehl, P.: On the use of non-deterministic automata for presburger arithmetic. In: Concurrency Theory - CONCUR. Lect. Notes Comp. Sci., vol. 6269, pp. 373–387. Springer (2010)
6. Durand-Gasselin, A., Habermehl, P.: Ehrenfeucht-fraïssé goes elementarily automatic for structures of bounded degree. In: Symposium on Theoretical Aspects of Computer Science, STACS. LIPIcs, vol. 14, pp. 242–253. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2012)
7. Ginsburg, S., Spanier, E.H.: Bounded ALGOL-like languages. T. Am. Math. Soc. pp. 333–368 (1964)
8. Guépin, F., Haase, C., Worrell, J.: On the existential theories of Büchi arithmetic and linear p-adic fields. In: Logic in Computer Science, LICS. IEEE (2019)
9. Klaedtke, F.: Bounds on the automata size for presburger arithmetic. ACM Trans. Comput. Log. **9**(2), 11:1–11:34 (2008)
10. Leroux, J., Point, G.: Tapas: The talence presburger arithmetic suite. In: Tools and Algorithms for the Construction and Analysis of Systems, TACAS. Lect. Notes Comp. Sci., vol. 5505, pp. 182–185. Springer (2009)
11. Lipshitz, L.M.: Some remarks on the Diophantine problem for addition and divisibility. In: Proc. Model Theory Meeting. vol. 33, pp. 41–52 (1981)
12. Manders, K.L., Adleman, L.M.: NP-complete decision problems for binary quadratics. J. Comput. Syst. Sci. **16**(2), 168–184 (1978)
13. Matthews, C.R.: Counting Points Modulo p for some Finitely Generated Subgroups of Algebraic Groups. Bull. Lond. Math. Soc. **14**(2), 149–154 (1982)
14. McCurley, K.S.: The discrete logarithm problem. In: Proc. of Symp. in Applied Math. vol. 42, pp. 49–74 (1990)
15. Scarpellini, B.: Complexity of subcases of Presburger arithmetic. T. Am. Math. Soc. **284**, 203–218 (1984)
16. The Liège automata-based symbolic handler (LASH): Available at `http://www.montefiore.ulg.ac.be/~boigelot/research/lash/`
17. To, A.W.: Unary finite automata vs. arithmetic progressions. Inf. Process. Lett. **109**(17), 1010–1014 (2009)

18. Weispfenning, V.: The complexity of linear problems in fields. J. Symb. Comput. **5**(1/2), 3–27 (1988)
19. Wolper, P., Boigelot, B.: On the construction of automata from linear arithmetic constraints. In: Tools and Algorithms for the Construction and Analysis of Systems, TACAS. Lect. Notes Comp. Sci., vol. 1785, pp. 1–19. Springer (2000)