

Extensions of ω -Regular Languages

Mikołaj Bojańczyk
University of Warsaw
Poland
bojan@mimuw.edu.pl

Rafał Stefański
University of Warsaw
Poland
rafal.stefanski@mimuw.edu.pl

Edon Kelmendi
University of Oxford
United Kingdom
edon.kelmendi@cs.ox.ac.uk

Georg Zetsche
Max Planck Institute for Software Systems (MPI-SWS)
Germany
georg@mpi-sws.org

Abstract

We consider extensions of monadic second-order logic over ω -words, which are obtained by adding one language that is not ω -regular. We show that if the added language L has a neutral letter, then the resulting logic is necessarily undecidable. A corollary is that the ω -regular languages are the only decidable Boolean-closed full trio over ω -words.

CCS Concepts: • Theory of computation \rightarrow Logic and verification.

Keywords: extensions of omega-regular, extensions of mso, full-trios, congruences, mso+u

ACM Reference Format:

Mikołaj Bojańczyk, Edon Kelmendi, Rafał Stefański, and Georg Zetsche. 2020. Extensions of ω -Regular Languages. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '20)*, July 8–11, 2020, Saarbrücken, Germany. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3373718.3394779>

1 Introduction

A famous theorem of Büchi [12, Theorem 2] says that the monadic second-order theory of $(\omega, <)$ is decidable. What can be added to this logic while retaining decidability? This question has seen a lot of interest, and we begin by discussing some of the existing results.

What predicates can be added? The first natural idea is to add predicates beyond the order $<$, e.g. a unary predicate for the primes, or a binary addition function. This idea was pursued already by Robinson in [30], in what is possibly the first published paper to mention MSO on $(\omega, <)$. This is before Büchi's theorem about decidability of MSO, and even

before the decidability results about weak MSO of Büchi [11, Corollary 1], Elgot [15, Corollary 5.8] and Trakhtenbrot [35]. After describing MSO, which he credits to Tarski's lectures, Robinson shows that adding the doubling function $n \mapsto 2n$ to MSO results in an undecidable logic [30, p.242]. Other examples of unary functions that lead to undecidability were given by Elgot and Rabin [16, Section 1]. One of these examples is that MSO becomes undecidable after adding any function f such that $f^{-1}(n)$ is infinite for all n . This result was strengthened by Siefkes [32, Theorem 5] who showed that it is enough that $f^{-1}(n)$ is infinite for all n with certain periodicity properties, and then by Thomas [33, Theorem 1] who showed that it is enough for $f^{-1}(n)$ to be infinite for infinitely many n . Another example of undecidability is MSO extended with any unary function f that is monotone and satisfies $f(n+1) > f(n) + 1$ for infinitely many n , see [33, Theorem 2]. This line of research is summarised in [29] as follows: "for most examples of natural functions or binary relations it turned out that the corresponding monadic theory is undecidable, usually shown via an interpretation of first-order arithmetic".

The undecidability issues mentioned above are avoided if one considers unary first-order predicates. The first examples of this kind were given by Elgot and Rabin, who showed that MSO remains decidable after adding unary predicates for the factorials, or squares, or cubes, etc. see [16, Theorem 4]). Following this result, a lot of attention has been devoted to identifying the unary predicates that keep MSO decidable. An equivalent phrasing of this question is: which ω -words have a decidable MSO theory? An interesting example is the Thue-Morse word; its MSO theory is decidable, which follows from [25, Theorem 3]. A general classification of ω -words with a decidable MSO theory was given by Semenov in [31, p. 165], this line of research was continued in [13, 29]. It is worth pointing out that the classification can be hard to apply to some specific cases; an important one being the case of prime numbers. It is unknown if MSO extended with a predicate for the prime numbers has a decidable theory; if

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

LICS '20, July 8–11, 2020, Saarbrücken, Germany

© 2020 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-7104-9/20/07.

<https://doi.org/10.1145/3373718.3394779>

this were the case then one could use the algorithm to decide if there are infinitely many twin primes¹.

It is worth pointing out that in all of the results discussed above, it makes no difference whether one uses MSO or weak MSO. The undecidability proofs for unary functions in [16, 32, 33] use only weak MSO. For the results about unary predicates, it makes no difference if MSO or weak MSO is used, because for every ω -word, its MSO theory is decidable if and only if its weak MSO theory is decidable, which follows from McNaughton’s determinisation theorem [22, p. 524]². In a sense, one could say that the results discussed above are really about extending weak MSO. This will no longer be true when adding quantifiers and languages.

What quantifiers can be added? Another line of research concerns adding new quantifiers. If the added quantifier has some implicit arithmetic, such as the Härtig quantifier [18], which expresses the existence of two sets of equal size with a given property, then MSO immediately becomes undecidable. This follows directly from Robinson’s result about $n \mapsto 2n$, and it is also discussed in more detail in [20, Theorem 14]. However, there are quantifiers which describe only topological or asymptotic behaviour, and for such quantifiers proving undecidability can be much harder. One example of an asymptotic quantifier is the bounding quantifier from [5], which expresses the property “for each $k \geq 0$, $\varphi(X)$ is true for some finite set X with $|X| \geq k$ ” [5]. The resulting logic, called MSO+U, is undecidable [9, Theorem 1.1]. However, it is close to the decidability border; in particular weak MSO with the bounding quantifier is decidable [6, Theorems 3 and 5]; and the same is true for its variants and extensions [10, Theorems 11 and 13]. Other extensions of MSO with asymptotic quantifiers were proposed by Michalewski, Mio and Skrzypczak in [23, 24], including a quantifier related to Baire category and a quantifier related to probability. The Baire quantifier does not add to the expressive power of MSO [24, Theorem 4.1]. On the other hand, the probability quantifier leads to an undecidable logic [23, Theorem 1], because it can express the undecidable problem of checking if a probabilistic Büchi automaton accepts some word with nonzero probability [3, Theorem 7.2]. The theme for quantifiers seems to be that adding a well-behaved quantifier to (non-weak) MSO either does not change the expressive power, or leads to an undecidable logic; but in the latter case the undecidability proof can be hard.

¹This theory is known to be decidable if one assumes Schinzel’s Hypothesis – a conjecture from number theory which implies that there are infinitely many twin primes [4, Theorem 4].

²The equivalence of MSO and wMSO need not hold after adding non-unary predicates. For example, wMSO with addition can only define languages in the Borel hierarchy, while MSO with addition can easily be shown to contain the logic MSO+U that will be discussed later in the paper, and MSO+U can define languages beyond the Borel hierarchy [19, Theorem 2.1].

What languages can be added? We now turn to the final kind of feature that can be added to MSO, namely languages. This is the main topic of this paper. A language $L \subseteq \{a, b\}^\omega$ over a binary alphabet can be viewed as a second-order unary predicate $L(X)$, which inputs a set $X \subseteq \omega$ and returns true if the language L contains the word where positions from X have label a and the remaining positions have label b . This can be generalized to alphabets with k letters: Then, the predicate inputs k sets and returns true if the k sets form a partition of ω and L contains the word encoding this partition. Let us write MSO+L for the extension of MSO which has the predicate described above.

Another equivalent way of describing the logic MSO+L uses closure properties of languages. A folklore fact about MSO is that existential monadic quantification is the same as taking the image of a language under a letter-to-letter homomorphism, see [28, p. 2] or [34, Section 2.3]. It follows that MSO+L is exactly the smallest class of languages of ω -words which contains L and all ω -regular languages, and which is closed under (a) Boolean combinations; (b) images of letter-to-letter homomorphisms; and (c) inverse images of letter-to-letter homomorphisms. We will return to this language theoretic approach in Section 2.

Example 1.1. Suppose that $L \subseteq \Sigma^\omega$ is a singleton language, i.e. it contains only one word w . Then MSO+L has the same expressive power as MSO extended with unary predicates

$$P_a = \{n \in \omega : \text{the } n\text{-th letter of } w \text{ is } a\} \quad \text{for } a \in \Sigma.$$

Therefore, in the case of singleton languages MSO+L, corresponds to the MSO extensions with unary first-order predicates that were studied in [13, 16, 29, 31, 33]. In particular, there are known decidable examples.

Example 1.2. Define $U \subseteq \{a, b\}^\omega$ to be the ω -words where blocks of a ’s have unbounded size:

$$U \stackrel{\text{def}}{=} \{a^{k_1} b a^{k_2} \dots : \limsup k_n = \infty\}.$$

In [8, Theorem 1.3] it is shown that adding the language U to MSO gives exactly the logic MSO+U. Hence it is unambiguous to write MSO+U, with both meanings (adding a quantifier or a language) being equivalent. As mentioned before, this logic is undecidable.

Example 1.3. Consider the ultimately periodic words, i.e.

$$P = \{wv^\omega : w, v \in \{a, b\}^* \text{ and } v \text{ is nonempty}\}.$$

It is not hard to see that MSO+P can express the language U from Example 1.2, and therefore this logic is undecidable [8, Theorem 1.4]. In contrast, adding P to weak MSO yields a decidable logic [10, Theorem 13].

Our goal in this paper is to classify the languages L such that MSO+L is undecidable. By the discussion in Example 1.1, this project is at least as difficult as classifying the ω -words with a decidable MSO theory. However, in the spirit of “asymptotic” conditions, we restrict attention to languages which

have a neutral letter, which means that there is a letter in the alphabet, denoted by 1, such that

$$w_1 1 w_2 1 \cdots \in L \quad \Leftrightarrow \quad w_1 w_2 \cdots \in L$$

holds for every words $w_1, w_2, \dots \in \Sigma^*$ where infinitely many w_i are nonempty³. We now state our main theorem.

Theorem 1.4. *If $L \subseteq \Sigma^\omega$ has a neutral letter and is not definable in MSO, then MSO+L is undecidable.*

In the above theorem, by undecidable we mean that there is no algorithm which decides the sentences of MSO+L that are true in $(\omega, <)$. Since MSO can quantify over words, this is the same as saying that satisfiability is undecidable for MSO+L for ω -words. Theorem 1.4 will be proven in Section 4.

Example 1.5. Define $U' \subseteq \{a, b, 1\}^\omega$ to be the words such that eliminating all 1's gives a word in the language U from Example 1.2. We claim that the logic is obtained by extending MSO with (a) the bounding quantifier; or (b) the language U ; or (c) the language U' . The equality of (a) and (b) was discussed in Example 1.2. The language U' can be defined using the bounding quantifier, hence the inclusion (c) \subseteq (a). The language U is the intersection of U' with language of words that do not contain the letter 1, hence the inclusion (b) \subseteq (c). The equality of these three logics is discussed in more detail in [8].

The language U will play an important role in the proof of Theorem 1.4. We will show that if L is not definable in MSO and has a neutral letter, then U is definable in MSO+L. Undecidability will then follow by Theorems [9, Theorem 1.1] and [8, Theorem 1.3]. In this sense, U is the simplest undecidable extension of MSO.

The paper is structured as follows. In Section 2, we discuss a version of our main theorem for finite words, which was proved by the fourth author, Kuske and Lohrey in [36]. Like [36], we prove our main theorem using syntactic congruences, and therefore Section 3 is devoted to a discussion of syntactic congruences for ω -languages. In Section 4, we prove our main result, and in Section 5 we show that the main theorem implies that the ω -regular languages are the only Boolean-closed full trios that are decidable.

2 Finite words

In this section, we describe the starting point for our work, which is a theorem by the fourth author, Kuske and Lohrey. It says that the regular languages of finite words are the only decidable Boolean-closed full trio. We will define full trios⁴

³In the definition of neutral letters, we require that the language is stable under inserting or deleting infinitely many neutral letters. However, this also implies that the language is stable under inserting or deleting finitely many neutral letters.

⁴Full trios are sometimes called (*rational*) *cones* in formal languages literature, they are meant to be a formalisation of robust classes of languages.

in Theorem 2.1. Recall that a *homomorphism* is a function

$$h : \Sigma^* \rightarrow \Gamma^* \quad \text{such that } h(wv) = h(w)h(v).$$

Define the *arithmetic hierarchy*, see [36, Section 2], to be the least class of languages of finite words that contains all recursively enumerable languages, and which is closed under complementation and homomorphic images.

Theorem 2.1. [36, Corollary 3.2] *Let \mathcal{L} be a class of languages of finite words which is a full trio, i.e. it is closed under:*

1. *images under homomorphisms; and*
2. *inverse images under homomorphisms; and*
3. *intersections with regular languages.*

If \mathcal{L} is additionally Boolean-closed (closed under union and complementation) and it contains at least one non-regular language, then it contains the arithmetic hierarchy.

In the above theorem, there is no assumption on neutral letters. This is because condition 2 deprecates the assumption, since a neutral letter can be added to any language by taking the inverse image under the homomorphism which eliminates the neutral letter. The closure properties used in Theorem 1.4 are weaker, and hence the assumption on neutral letters is needed. As a warm-up for the case of ω -words, we give below a proof sketch for the above theorem.

Proof sketch. The proof uses rational relations [14, p.236]. Recall that a rational relation is a binary relation on words that is recognised by a nondeterministic automaton where each transition is labelled by a pair (input word, output word), with both words being possibly empty. By Nivat's theorem (Propositions 1 and 2 in [26]), if a language class \mathcal{L} is a full trio, then it is closed under images under rational relations, which can be visualised as the following reasoning rule:

$$\frac{K \subseteq \Sigma^* \text{ is in } \mathcal{L} \quad R \subseteq \Sigma^* \times \Gamma^* \text{ is a rational relation}}{\{v \in \Gamma^* : \exists w \in L \text{ with } (w, v) \in R\} \text{ is in } \mathcal{L}}$$

Because \mathcal{L} is closed under complementation, we can also use a variant of the above rule where \forall is used instead of \exists in the conclusion of the rule (i.e. below the line).

The key idea is to use the closure properties to formalise the syntactic right congruence of the language. Let $L \subseteq \Sigma^*$ be some non-regular language in \mathcal{L} , which exists by assumption, and let \sim be its syntactic right congruence, i.e. the equivalence relation defined by

$$u \sim u' \stackrel{\text{def}}{=} \forall v \in \Sigma^* \quad uv \in L \Leftrightarrow u'v \in L.$$

By the Myhill-Nerode theorem, \sim has infinite index, i.e. infinitely many equivalence classes. Using the reasoning rule with rational relations, one shows that \mathcal{L} contains the language

$$L_1 = \{u\#u' : u \sim u'\},$$

where # is a fresh separator symbol. Consider now two separator symbols # and #. Define L_2 to be the language

$$\{w_1\#\cdots\#w_n\#v_1\#\cdots\#v_m\#\} : \begin{cases} w_1 \sim v_1 \\ w_n \sim v_m \\ w_i \not\sim w_j \text{ for } i \neq j \\ v_i \not\sim v_j \text{ for } i \neq j \\ w_i \sim v_j \Rightarrow w_{i+1} \sim v_{j+1} \end{cases} \}.$$

Using the closure properties, one shows $L_2 \in \mathcal{L}$. A short analysis of the conditions defining L_2 reveals that every word in L_2 satisfies

$$m = n \quad \text{and} \quad w_1 \sim v_1, w_2 \sim v_2, \dots, w_n \sim v_n.$$

Furthermore, since \sim has infinitely many equivalence classes, it follows that $n = m$ can be arbitrarily large. By projecting away the words w_i, v_i using a homomorphism, it follows that \mathcal{L} contains the language

$$L_3 = \{\#^n \#^n : n \in \{1, 2, \dots\}\}.$$

A string encoding of runs of two-counter machines, see [17, Theorem 2], can be used to show that \mathcal{L} contains every recursively enumerable language. The arithmetic hierarchy follows, by closure of \mathcal{L} under homomorphic images and complementation. \square

3 Congruences for ω -words

Like in Theorem 2.1, the proof of Theorem 1.4 also uses congruences. However, there are several issues with congruences for ω -words, which mean that some new ideas are needed. The main problem is that there is no good notion of syntactic congruence for languages of ω -words.

We begin by discussing several existing approaches to congruences for ω -words, see also [21]. In all cases, we begin with a language $L \subseteq \Sigma^\omega$, and use it to define an equivalence relation on finite words. The first candidate is the *right congruence*, which identifies two finite words $u, u' \in \Sigma^*$ if

$$\forall v \in \Sigma^\omega \quad uv \in L \Leftrightarrow u'v \in L.$$

This right congruence does not characterise the ω -regular languages, although it does have some use, for example in automata learning [1]. There could be finitely many equivalence classes despite a language not being ω -regular. For example, every prefix independent language will have one equivalence class of right congruence, but there are prefix independent languages which are not ω -regular, such as

$$\{w(a^n b)^\omega : w \in \{a, b\}^* \text{ and } n \text{ is prime}\}.$$

By induction on formula size one can show that if L has a right congruence of finite index, then the same is true for every language definable in $\text{MSO}+L$; which shows that right congruences will not be useful for our main result. Similar problems arise for the two-sided version of right congruence.

A more useful congruence for ω -words uses two-sided environments and ω -iteration; this leads to the *Arnold congruence* [2, Section 2], which identifies u, u' if

$$\wedge \begin{cases} \forall w \in \Sigma^* \forall v \in \Sigma^\omega & wuv \in L \Leftrightarrow wu'v \in L \\ \forall w, v \in \Sigma^* & w(uv)^\omega \in L \Leftrightarrow w(u'v)^\omega \in L. \end{cases}$$

The Arnold congruence still does not characterise the ω -regular languages. For example, the language U is not ω -regular, but it has two equivalence classes under Arnold congruence: words which contain b , and words which do not contain b .

Fortunately, there is a successful characterisation of ω -regular languages via congruences. This characterisation is stated below, and it corresponds to ω -semigroups.

Theorem 3.1. [27, Theorem 7.5] *A language $L \subseteq \Sigma^\omega$ is ω -regular if and only if there is an equivalence relation \sim which has finite index and satisfies the following conditions for all sequences of finite words u_i :*

$$\left(\bigwedge_{i \in \{1,2\}} u_i \sim u'_i \right) \Rightarrow u_1 u_2 \sim u'_1 u'_2 \quad (1)$$

$$\left(\bigwedge_{i \in \mathbb{N}} u_i \sim u'_i \right) \Rightarrow (u_1 u_2 \cdots \in L \Leftrightarrow u'_1 u'_2 \cdots \in L) \quad (2)$$

We use the name *ω -congruence* for an equivalence relation that satisfies conditions (1) and (2) in the above theorem. We use the above theorem in our main result. Although promising, the characterisation in terms of ω -congruences has one important drawback, namely non-uniqueness. For the right congruence, the defining property

$$\forall v \in \Sigma^\omega \quad uv \in L \Leftrightarrow u'v \in L.$$

gives a unique equivalence relation, and hence it makes sense to speak of *the* right congruence. A similar property holds for the Arnold congruence. The uniqueness of the definition of right congruence, and the fact that its definition can be formalised using rational relations, is what drives the proof of Theorem 2.1.

In contrast, there is no uniqueness in Theorem 3.1, and there cannot be. A language might not have a unique coarsest ω -congruence (such an equivalence relation is called the *syntactic ω -congruence*). An example is the language U , see [7, Running Example 2]. For ω -regular languages, the syntactic ω -congruence exists and coincides with Arnold congruence, see [27, Proposition 8.8], but this is not very helpful in our setting, since we want to study congruences for languages that are *not* ω -regular. These are issues that we will need to overcome in the proof of our main result.

We finish this section with a simple observation, which says that condition (1) in Theorem 3.1 is superfluous. This observation will be useful later on, since condition (2) will be easier to formalise.

Lemma 3.2. *If there is an equivalence relation of finite index which satisfies (2), then there is an equivalence relation of finite index which satisfies both (1) and (2).*

Proof. Induction on the number of equivalence classes in the equivalence relation, call it \sim . In the base case, when \sim has one equivalence class, condition (1) holds vacuously. Consider the induction step. For this proof, it is easier to work with the following equivalent form of (1):

$$u \sim u' \quad \Rightarrow \quad (uw \sim u'w) \wedge (wu \sim wu').$$

If \sim satisfies the above implication, then we are already done. Otherwise, choose a violation of the implication, i.e. words $u \sim u'$ which do not satisfy the conclusion of the implication. Assume $uw \not\sim u'w$ (the case $wu \not\sim wu'$ is analogous). Define \approx to be the equivalence relation obtained from \sim by merging the equivalence classes of uw and $u'w$. We claim that \approx still satisfies condition (2), and therefore the induction assumption can be applied. We visualize (2) as follows:

$$\begin{array}{ccccccc} u_0 & u_1 & u_2 & \cdots & \in L \\ \approx & \approx & \approx & \approx & \Downarrow \\ u'_0 & u'_1 & u'_2 & \cdots & \in L \end{array}$$

By definition of \approx and assumption (2) for \sim , we can replace every u_n in the equivalence class of uw by uw and every u_n in the equivalence class of $u'w$ by $u'w$, without affecting membership in L . Therefore we can assume without loss of generality that every u_n is either uw , or $u'w$, or a word that is \sim -equivalent to neither of these. The same can be done for u'_n . By definition of \approx , if $u_n = uw$ then u'_n has to be one of uw or $u'w$. We can now split each $u_n = uw$ into two words u and w , likewise for $u_n = u'w$, and then use again the assumption that \sim satisfies (2) to finish the proof. \square

4 Proof of the main theorem

In this section we prove Theorem 1.4. We will prove that for a language L that has a neutral letter, and is not definable in MSO , $\text{MSO}+L$ contains the language U , and therefore undecidability follows thanks to the results about the logic $\text{MSO}+U$. To explain how U can be defined, we use a game called the *congruence game*. This game is played by two players called Spoiler and Duplicator, and it is parameterised by an ω -word $u \in \{a, b\}^\omega$. The congruence game is designed so that player Duplicator wins if and only if $u \in U$, which means that u has a -labelled intervals of unbounded size. This is achieved as follows. Roughly speaking, the goal of player Duplicator is to show that from the perspective of the language L , each finite word $w \in \Sigma^*$ is equivalent to some word $v \in \Sigma^*$ which can fit infinitely often into a -labelled intervals in the word u . Since the language L is not ω -regular, Duplicator needs intervals of unbounded size to win.

Define an *interval* to be a finite connected subset of ω , i.e. it contains all positions between its first and last position. If W, V are intervals, then we write $W < V$ if the last position of W is strictly before the first position of V .

Definition 4.1 (Congruence Game). The *congruence game* for $u \in \{a, b\}^\omega$ is the following game played by two players, called Spoiler and Duplicator.

1. Spoiler chooses an infinite family \mathcal{W} of pairwise disjoint intervals.
2. Duplicator chooses intervals

$$W_1 < V_1 < W_2 < V_2 < \cdots$$

such that W_1, W_2, \dots are from \mathcal{W} and V_1, V_2, \dots contain only positions with label a in the word u .

3. Spoiler chooses words

$$w_1, w_2, \dots \in \Sigma^*$$

such that $|w_i| < |W_i|$ for every $i \in \{1, 2, \dots\}$.

4. Duplicator chooses words

$$v_1, v_2, \dots \in \Sigma^*$$

such that $|v_i| < |V_i|$ for every $i \in \{1, 2, \dots\}$.

5. Spoiler chooses a sequence of natural numbers

$$i_1 < i_2 < \cdots$$

6. Duplicator wins the game if and only if:

$$w_{i_1} w_{i_2} w_{i_3} \cdots \in L \quad \Leftrightarrow \quad v_{i_1} v_{i_2} v_{i_3} \cdots \in L.$$

The key result about the congruence game is the following lemma. The lemma does not use the assumption that L has a neutral letter; this assumption will be used later when formalising the congruence game in $\text{MSO}+L$.

Lemma 4.2. *Assume that $L \subseteq \Sigma^\omega$ is not ω -regular. Then*

$$\text{Duplicator wins the congruence game for } u \Leftrightarrow u \in U.$$

Proof.

(\Leftarrow) Assume $u \in U$. We will show a winning strategy for player Duplicator. Suppose that player Spoiler has chosen a family \mathcal{W} in round 1. Since $u \in U$, intervals with only a -labelled positions have unbounded size, and therefore in round 2, player Duplicator can choose the intervals so that $|V_i| \geq |W_i|$ for all i . For every choice of words w_i made by player Spoiler in round 3, Duplicator's response in round 4 is to choose the words v_i so that $v_i = w_i$ for all i . This guarantees victory for Duplicator, regardless of Spoiler's move in round 5.

(\Rightarrow) Assume $u \notin U$. We will show a winning strategy for player Spoiler. In round 1, Spoiler picks \mathcal{W} so that the lengths of the intervals tend to infinity, i.e. no size appears infinitely often. Let W_i and V_i be the intervals that are chosen in round 2 by player Duplicator. By choice of \mathcal{W} , the lengths of the intervals W_i tend to infinity, while by assumption that $u \notin U$, the lengths of the intervals V_i are bounded. In round 3, Spoiler chooses the words w_i so that every word from Σ^* appears infinitely often. This can be done because the lengths of the intervals W_i tend to infinity. Suppose that Duplicator chooses some words v_i in round 4. Since the intervals V_i have bounded size, the words v_i chosen by Duplicator come

from a finite set $F \subseteq \Sigma^*$. This means that for every $w \in \Sigma^*$, there is some $v \in \Sigma^*$ such that infinitely often $w_i = w$ and $v_i = v$. Choose some function

$$f : \Sigma^* \rightarrow F$$

which realises the dependency $w \mapsto v$, i.e. for every $w \in \Sigma^*$,

$$w = w_i \quad \text{and} \quad f(w) = v_i \quad \text{for infinitely many } i. \quad (3)$$

Apply Lemma 3.2 with \sim being the kernel of f , i.e. the equivalence relation that identifies two words if they have the same image under f . Since L is not ω -regular, then by Lemma 3.2 there must be a violation of (2), i.e. there must be words u_1, u_2, \dots such that

$$u_1 u_2 \cdots \in L \quad \not\equiv \quad f(u_1) f(u_2) \cdots \in L. \quad (4)$$

By (3), in Round 5, Spoiler can choose the indices $i_1 < i_2 < \dots$ so that

$$u_n = w_{i_n} \quad f(u_n) = f(v_{i_n}) \quad \text{for all } n \in \{1, 2, \dots\}$$

and therefore win thanks to (4). \square

The following corollary, and undecidability of the logic $\text{MSO}+\text{U}$, complete the proof of Theorem 1.4.

Corollary 4.3. *If $L \subseteq \Sigma^\omega$ is not ω -regular and has a neutral letter, then $\text{MSO}+\text{U} \subseteq \text{MSO}+L$.*

Proof. By [8, Theorem 1.3], it is enough to show that the language U is definable in $\text{MSO}+L$. By Lemma 4.2, it is enough to show that $\text{MSO}+L$ can express that Duplicator has a winning strategy in the congruence game.

A family of disjoint intervals is represented by two sets of positions: the set X of leftmost positions in the intervals, and the set Y of rightmost positions in the intervals. The condition that all intervals are disjoint means that

$$\forall x_1, x_2 \in X \quad x_1 < x_2 \implies \exists y \in Y \quad x_1 \leq y < x_2.$$

Using this representation, the choices of intervals in rounds 1 and 2 can be represented by set quantification, with choices of player Duplicator using existential quantifiers and choices of player Spoiler using universal quantifiers. The words w_i chosen in round 3 are represented by colouring the intervals W_i with letters from Σ ; and using the neutral letter for positions not in the intervals W_i . The same goes for round 4. The subsequence in round 5 is represented by a subset of leftmost positions in the intervals W_i . The winning condition in round 6 is checked by using the predicate for L , and using the neutral letter for positions not belonging to the subsequence of round 5. \square

5 Boolean closed full trios

We finish with a corollary of our main theorem, which is an analog of Theorem 2.1 for infinite words: if a Boolean-closed full trio of languages of infinite words contains at least one non-regular language, then it includes the entire arithmetic hierarchy, subject to a certain representation.

As mentioned in the introduction, a language of ω -words is definable in $\text{MSO}+L$ if and only if it belongs to the smallest class of languages that contains L , contains all ω -regular languages, is closed under Boolean combinations, as well as images and inverse images under letter-to-letter homomorphisms. If we lift the restriction on homomorphisms being letter-to-letter, then we get a Boolean-closed full trio, as discussed below.

When applying a homomorphism that may erase some letters, an ω -word can be mapped to a finite word. Therefore, in the presence of such homomorphisms, it makes sense to consider languages of words of length $\leq \omega$, i.e. words which are either finite or ω -words. For such words, define a *regular language* to be a union of two languages: a regular language of finite words, plus a regular language of ω -words. Define a *homomorphism* for words of length $\leq \omega$ to be a function

$$h : \Sigma^{\leq \omega} \rightarrow \Gamma^{\leq \omega}$$

which is obtained by applying to each letter a function of type $\Sigma \rightarrow \Gamma^*$.

Theorem 5.1. *Let \mathcal{L} be a class of languages of words of length at most $\leq \omega$ which is closed under:*

1. *images under homomorphisms; and*
2. *inverse images under homomorphisms; and*
3. *intersections with regular languages.*

If \mathcal{L} is additionally Boolean-closed (closed under union and complementation) and it contains at least one non-regular language, then for every $L \subseteq \Sigma^$ in the arithmetic hierarchy,*

$$\underbrace{\{wv^\omega : w \in \Sigma^*, v \in L\}}_{\text{loop representation of } L} \in \mathcal{L}.$$

loop representation of L

Proof. By closure under inverse images of homomorphisms and under intersection with Σ^* and Σ^ω for any Σ , if \mathcal{L} contains some non-regular language, then it contains some non-regular ω -language with a neutral letter. By Corollary 4.3, \mathcal{L} contains all languages definable in $\text{MSO}+\text{U}$. By [9, Lemma 3.2], for every recursively enumerable language $L \subseteq \Sigma^*$, the logic $\text{MSO}+\text{U}$ defines some ω -language K , over an alphabet extended with a neutral letter, such that

$$\text{loop representation of } L = h(K),$$

where h is the homomorphism that eliminates the neutral letter. Since \mathcal{L} is closed under homomorphic images, it follows that \mathcal{L} contains the loop representations of all recursive enumerable languages. For the arithmetic hierarchy, it is enough to observe that the class

$$\{L \subseteq \Sigma^* : \mathcal{L} \text{ contains the loop representation of } L\}$$

is closed under Boolean combinations and homomorphic images. \square

In contrast to the finite word setting, we cannot conclude that \mathcal{L} contains L itself: Standard arguments show that if a language $K \subseteq \Sigma^{\leq \omega}$ has a finite-index right congruence, then

every language obtained from K using Boolean combinations and full trio operations also has a finite-index right congruence. Thus, for example, if we start with U , all obtainable languages over finite words are regular.

One could also consider Boolean closed full trios of ω -languages. Then, homomorphisms would be defined by functions of type $\Sigma \rightarrow \Gamma^*$ and the (inverse) image of a homomorphism on a subset of Σ^ω would contain only those resulting words that are infinite. For this notion of (inverse) image, Theorem 5.1 follows with essentially the same proof.

Acknowledgments

The first three authors have been supported by ERC under the ERC Consolidator Grant LIPA 683080.

References

- [1] Dana Angluin and Dana Fisman. 2018. Regular ω -languages with an informative right congruence. In *GandALF 2018, Saarbrücken, Germany, 2018*. 265–279.
- [2] André Arnold. 1985. A syntactic congruence for rational ω -languages. *Theoret. Comput. Sci.* 39, 2-3 (1985), 333–335.
- [3] Christel Baier, Marcus Grösser, and Nathalie Bertrand. 2012. Probabilistic ω -automata. *J. ACM* 59, 1 (March 2012), 1:1–1:52.
- [4] P. T. Bateman, C. G. Jockusch, and A. R. Woods. 1993. Decidability and undecidability of theories with a predicate for the primes. *Journal of Symbolic Logic* 58, 2 (June 1993), 672–687.
- [5] Mikołaj Bojańczyk. 2004. A bounding quantifier. In *Computer Science Logic, CSL 2004, Karpacz (Lecture Notes in Computer Science)*, Jerzy Marcinkowski and Andrzej Tarlecki (Eds.), Vol. 3210. Springer, 41–55.
- [6] Mikołaj Bojańczyk. 2011. Weak MSO with the unbounding quantifier. *Theory Comput. Syst.* 48, 3 (2011), 554–576.
- [7] Mikołaj Bojańczyk. 2015. Recognisable languages over monads. *CoRR* abs/1502.04898 (2015).
- [8] Mikołaj Bojańczyk, Laure Daviaud, Bruno Guillon, Vincent Penelle, and A. V. Sreejith. (to appear). Undecidability of MSO+“ultimately Periodic”. *Logical Methods in Computer Science* abs/1807.08506 ((to appear)).
- [9] Mikołaj Bojańczyk, Paweł Parys, and Szymon Toruńczyk. 2016. The MSO+U theory of $(\mathbb{N}, <)$ is undecidable. In *33rd STACS Orléans, France (LIPIcs)*, Nicolas Ollinger and Heribert Vollmer (Eds.), Vol. 47. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 21:1–21:8.
- [10] Mikołaj Bojańczyk and Szymon Toruńczyk. 2009. Deterministic automata and extensions of Weak MSO. In *FSTTCS 2009, IIT Kanpur, India (LIPIcs)*, Vol. 4. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 73–84.
- [11] J. Richard Büchi. 1960. Weak second-order arithmetic and finite automata. *Z. Math. Logik und Grundl. Math.* 6 (1960), 66–92.
- [12] J. Richard Büchi. 1962. On a decision method in restricted second order arithmetic. In *Logic, Methodology and Philosophy of Science (Proc. 1960 Internat. Congr.)*. Stanford Univ. Press, Stanford, Calif., 1–11.
- [13] Olivier Carton and Wolfgang Thomas. 2002. The monadic theory of morphic infinite words and generalizations. *Information and Computation* 176, 1 (2002), 51–65.
- [14] Samuel Eilenberg. 1974. *Automata, languages, and machines. Vol. A*. Academic Press [A subsidiary of Harcourt Brace Jovanovich, Publishers], New York.
- [15] Calvin C. Elgot. 1961. Decision problems of finite automata design and related arithmetics. *Trans. Amer. Math. Soc.* 98 (1961), 21–51.
- [16] Calvin C. Elgot and Michael O. Rabin. 1966. Decidability and undecidability of second (first) order theory of (generalized) successor. *J. of Symbolic Logic* 31 (1966), 169–181.
- [17] Juris Hartmanis and John E. Hopcroft. 1970. What makes some language theory problems undecidable. *J. Comput. Syst. Sci.* 4, 4 (1970), 368–376.
- [18] Heinrich Herre, Michał Krynicki, Alexandr Pinus, and Jouko Väänänen. 1991. The Härtig quantifier: a survey. *The Journal of symbolic logic* 56, 4 (1991), 1153–1183.
- [19] Szczepan Hummel and Michał Skrzypczak. 2012. The topological complexity of MSO+U and related automata models. *Fundamenta Informaticae* 119, 1 (2012), 87–111.
- [20] Felix Klaedtke and Harald Ruef. 2003. Monadic second-order logics with cardinalities. In *30th ICALP 2003, Eindhoven, The Netherlands*. 681–696.
- [21] Oded Maler and Ludwig Staiger. 1997. On syntactic congruences for ω -languages. *Theoret. Comput. Sci.* 183, 1 (1997), 93–112.
- [22] Robert McNaughton. 1966. Testing and generating infinite sequences by a finite automaton. *Information and Control* 9 (1966), 521–530.
- [23] Henryk Michalewski and Matteo Mio. 2016. Measure quantifier in monadic second order logic. In *International Symposium on Logical Foundations of Computer Science*. Springer, 267–282.
- [24] Matteo Mio, Michał Skrzypczak, and Henryk Michalewski. 2018. Monadic Second Order Logic with Measure and Category Quantifiers. *Logical Methods in Computer Science* 14, 2 (2018).
- [25] Andrei Muchnik, Alexei Semenov, and Maxim Ushakov. 2003. Almost periodic sequences. *Theoretical Computer Science* 304, 1-3 (2003), 1–33.
- [26] Maurice Nivat. 1968. Transductions des langages de Chomsky. *Annales de l'Institut Fourier* 18, 1 (1968), 339–455. <https://doi.org/10.5802/aif.287>
- [27] Dominique Perrin and Jean-Eric Pin. 2004. *Infinite words - automata, semigroups, logic and games*. Pure and applied mathematics series, Vol. 141. Elsevier Morgan Kaufmann.
- [28] Michael O. Rabin. 1969. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.* 141 (1969), 1–35.
- [29] Alexander Rabinovich and Wolfgang Thomas. 2006. Decidable theories of the ordering of natural numbers with unary predicates. In *Computer Science Logic, Zoltán Ésik (Ed.)*. Springer Berlin Heidelberg, Berlin, Heidelberg, 562–574.
- [30] Raphael M. Robinson. 1958. Restricted set-theoretical definitions in arithmetic. *Proc. Amer. Math. Soc.* 9, 2 (Feb. 1958), 238–238.
- [31] Alexei L. Semenov. 1984. Decidability of monadic theories. In *Mathematical foundations of computer science, 1984 (Prague, 1984)*. Springer, Berlin, 162–175.
- [32] Dirk Siefkes. 1971. Undecidable extensions of monadic second order successor arithmetic. *Mathematical Logic Quarterly* 17, 1 (1971), 385–394.
- [33] Wolfgang Thomas. 1975. A note on undecidable extensions of monadic second order successor arithmetic. *Archive for Mathematical Logic* 17, 1 (1975), 43–44.
- [34] Wolfgang Thomas. 1997. Languages, automata, and logic. In *Handbook of Formal Languages, Volume 3: Beyond Words.*, Grzegorz Rozenberg and Arto Salomaa (Eds.). Springer, 389–455.
- [35] Boris A. Trakhtenbrot. 1961. Finite automata and the logic of single-place predicates. *Dokl. Akad. Nauk SSSR* 140, 2 (1961), 326–329.
- [36] Georg Zetsche, Dietrich Kuske, and Markus Lohrey. 2017. On Boolean closed full trios and rational Kripke frames. *Theory of Computing Systems* 60, 3 (2017), 438–472. <https://doi.org/10.1007/s00224-016-9694-0>