DECIDABILITY, COMPLEXITY, AND EXPRESSIVENESS OF FIRST-ORDER LOGIC OVER THE SUBWORD ORDERING

SIMON HALFON, PHILIPPE SCHNOEBELEN, AND GEORG ZETZSCHE

Abstract. We consider first-order logic over the subword ordering on finite words where each word is available as a constant. Our first result is that the Σ_1 theory is undecidable (already over two letters).

We investigate the decidability border by considering fragments where all but a certain number of variables are alternation bounded, meaning that the variable must always be quantified over languages with a bounded number of letter alternations. We prove that when at most two variables are not alternation bounded, the Σ_1 fragment is decidable, and that it becomes undecidable when three variables are not alternation bounded. Regarding higher quantifier alternation depths, we prove that the Σ_2 fragment is undecidable already for one variable without alternation bound and that when all variables are alternation bounded, the entire first-order theory is decidable.

1. Introduction

A subsequence of a (finite) sequence u is a sequence obtained from u by removing any number of elements. For example, if $u = (a, b, a, b, a)$ then $u' = (b, b, a)$ is a subsequence of u, a fact we denote with $u' \sqsubseteq u$. Other examples that work for any u are $u \sqsubseteq u$ (remove nothing) and $() \sqsubseteq u$. In the rest of this paper, we shall use the terminology from formal methods and will speak of *words* and their *subwords* rather than finite sequences.

Reasoning about subwords occurs prominently in many areas of computer science, e.g., in pattern matching (of texts, of DNA strings, etc.), in coding theory, in theorem proving, in algorithmics, etc. Closer to our own motivations, the automatic verification of unreliable channel systems and related problems involves the subword ordering or some of its variants [2, 6, 16, 24]. Our experience is that reasoning about subwords and related concepts (e.g., shuffles of words) involves ad hoc techniques quite unlike the standard tools that work well with prefixes and suffixes [22].

The logic of subwords. In this paper we consider the first-order logic $\mathsf{FO}(A^*, \sqsubseteq)$ of words over some alphabet $A = \{a, b, c, \ldots\}$ equipped with the subword relation $\sqsubseteq$. Our main objective is to understand how and when one can decide queries formulated in this logic, or decide whether a given formula is valid.

For example, we consider formulas like

$$\varphi_1: \quad \forall u, u', u'' : u \sqsubseteq u' \wedge u' \sqsubseteq u'' \implies u \sqsubseteq u'' ,$$

$$\varphi_2: \quad \exists u : \; abcd \sqsubseteq u \wedge bcde \sqsubseteq u \wedge abcde \not\sqsubseteq u ,$$

$$\varphi_3: \quad \forall u, v : \exists s : \begin{pmatrix} \quad u \sqsubseteq s \wedge v \sqsubseteq s \\ \wedge \quad \forall t : u \sqsubseteq t \wedge v \sqsubseteq t \implies s \sqsubseteq t \end{pmatrix} .$$

Here φ_1 states that the subword relation is transitive (which it is).

More interesting is φ_2, stating that it is possible that a word contains both $abcd$ and $bcde$ as subwords but not $abcde$. This formula is true and, beyond knowing its validity, one is

The third author is supported by a fellowship within the Postdoc-Program of the German Academic Exchange Service (DAAD) and by Labex DigiCosme, Univ. Paris-Saclay, project VERICONISS.

also interested in *solutions*: can we design a constraint solver that will produce a witness, e.g., $u = bcdeabcd$, or more generally the set of solutions?

Our third example, φ_3, states that words ordered by subwords are an upper semilattice. This is a more complex formula with Π_3 quantifier alternation. It is not valid in general (e.g., ab and ba have no lub) but this depends on the alphabet A at hand: φ_3 holds if A is a singleton alphabet, i.e., $\{a\}^* \models \varphi_3$ but $\{a,b\}^* \not\models \varphi_3$.

We say that formulas like φ_1 or φ_3 where constants from A^* do not appear are in the *pure fragment.* Formally, there are two logics at hand here. The *pure* logic is the logic of the purely relational structure $(A^*, \sqsubseteq)$ while the *extended* logic is over the expansion $(A^*, \sqsubseteq, w_1, \ldots)$ where there is a constant symbol w_i for every word in A^*.

As we just illustrated with φ_3, the validity of a formula may depend on the underlying alphabet even for the pure fragment. We note that this phenomenon is not limited to the degenerate case of singleton alphabets. Indeed, observe that it is possible to state that u is a letter, i.e., is a word of length 1, in the pure fragment:

$$\exists z : \forall x : z \sqsubseteq x \wedge (x \sqsubseteq u \implies \big(u \sqsubseteq x \vee x \sqsubseteq z\big)) .$$

Thus, even in the pure fragment, one can state that A contains 2, 3, ..., or exactly n letters. Similarly one can state that A is infinite by saying that no word contains all letters.

State of the art. Relatively little is known about deciding the validity of $(A^*, \sqsubseteq)$ formulas and about algorithms for computing their solutions. By comparison, it is well known that the Σ_2-theory of $\mathsf{FO}(A^*, \cdot)$, the logic of strings with concatenation, is undecidable [11, 34], and that its Σ_1 fragment (aka "word equations") is decidable in PSPACE [21, 33]. Moreover, introducing counting predicates leads to an undecidable Σ_1 fragment [7].

Regarding the logic of subwords, Comon and Treinen showed undecidability for an extended logic $\mathsf{FO}(A^*, \sqsubseteq, p_\#)$ where $A = \{a, b, \#\}$ has three letters and $p_\#$ is a unary function that prepends # in front of a word, hence is a restricted form of concatenation [9, Prop. 9]. Kuske showed that, when only the subword predicate is allowed, the logic $\mathsf{FO}(A^*, \sqsubseteq)$ is undecidable and already its Σ_3 fragment is undecidable when $|A| \geq 2$. Kudinov *et al.* considered definability in $(A^*, \sqsubseteq)$ and showed that the predicates definable in $(A^*, \sqsubseteq)$ are exactly the arithmetical predicates[1] [29].

Kuske's result on the Σ_3 theory leaves open the question whether smaller fragments are decidable. Karandikar and Schnoebelen showed that the Σ_2 theory is undecidable [23] and this is tight since the Σ_1 fragment is decidable, in fact NP-complete [23, 30].

Karandikar and Schnoebelen also showed that the two-variable fragment $\mathsf{FO}^2(A^*, \sqsubseteq)$ is decidable [23] and that it has an elementary complexity upper bound [25]. Decidability extends to the logic $\mathsf{FO}^2(A^*, \sqsubseteq, R_1, R_2, \ldots)$ where arbitrary regular languages (monadic predicates) are allowed.

Objectives of this paper. We are interested in solving constraints built with the subword ordering. This corresponds to the Σ_1 fragment but beyond deciding validity, we are interested in computing sets of solutions: a formula like φ_2 can be seen as a conjunctive set of constraints, "$abcd \sqsubseteq x \wedge bcde \sqsubseteq x \wedge abcde \not\sqsubseteq x$" that define a set of words (a set of tuples when there are several free variables).

A first difficulty is that Kuske's decidability result for the Σ_1 fragment only applies to the pure fragment, where constants are not allowed. That is, we know how to decide the validity of formulas like φ_1 but not like φ_2. However, using constants inside constraints is natural and convenient. In particular, it makes it easy to express piecewise testable constraints (see below), and we would like to generalise Kuske's result to the extended logic.

[1]Those that are invariant under the automorphisms of the structure

$\Sigma_{i,j}$	0	1	2	3
1	NP	NP	in NEXP	U
$i \geq 2$	$\Sigma^{\mathsf{EXP}}_{i-1}$	U	U	U

TABLE 1. The cell in row i and column j shows the decidability/complexity of the fragment $\Sigma_{i,j}$.

We note that, in principle, the difference between the pure and the extended logic is only superficial since, up to automorphisms, arbitrary words can be defined in the logic,[2] see [23, 29, 30]. However this requires some universal quantification (even when defining the empty word) that are not allowed when restricting to the Σ_1 fragment. So this avenue is closed.

Summary of results. Our first result is that, *when constants are allowed*, the Σ_1 fragment of $\mathsf{FO}(A^*, \sqsubseteq, w_1, \ldots)$ is actually undecidable. In fact the Σ_1 fragment of $\mathsf{FO}(A^*, \sqsubseteq, W)$, where *a single constant* $W \in A^*$ can be named, is undecidable unless W is too simple. These results hold as soon as A contains two distinct letters and exhibit a sharp contrast between the pure and the extended logic. We found this very surprising because, before hitting on undecidability, we had already developed algorithms that solve large classes of Σ_1 constraints.

Our second result identifies a key factor influencing decidability: it turns out that free variables ranging over a "thin" language like $L = a^+bc^*$, are easier to handle than variables ranging over a "wide" language like $L' = (a+b)^*$. The key difference is that a thin language only allows a bounded number of letter changes (in L we have a's, then b's, then c's) while a wide language contains words with arbitrarily many alternations between distinct letters.

These observations lead to a new descriptive complexity measure for the formulas in $\mathsf{FO}(A^*, \sqsubseteq, w_1, \ldots)$. The associated fragments, denoted $\Sigma_{i,j}$ for $i, j \in \mathbb{N}$, consist of all Σ_i formulas where j variables, say $x_1, x_2, \ldots, x_j$ can be used without any restrictions, while all the other variables must be restricted with respect to letter alternations, say using $x \in (a_1^* a_2^* \cdots a_n^*)^\ell$ for some $\ell \in \mathbb{N}$ and assuming that $a_1, \ldots, a_n$ is a fixed enumeration of A. In computer-aided verification, such bounded quantifications occur in the analysis of bounded context-switching protocols.

Within this classification framework, we can delineate a precise undecidability landscape. The $\Sigma_{1,2}$ fragment is decidable while $\Sigma_{1,3}$ is undecidable even for $|A| = 2$. The $\Sigma_{2,0}$ fragment is decidable while $\Sigma_{2,1}$ is not. In fact, when all variables are alternation bounded, the entire first-order theory is decidable.

The computational complexity of all mentioned fragments is summarized in Table 1. Note that, in this table, Σ^{EXP}_n denotes the n-th level of the weak EXP hierarchy, which lies between NEXP and $\mathsf{EXPSPACE}$ [15, 18].

Finally, we offer a series of expressiveness results showing how various predicates like concatenation or length function can, or cannot, be defined in the $\Sigma_{i,j}$ fragments. As demonstrated in the paper, expressiveness results are crucial to obtain hardness results. Beyond their theoretical interest, and since pinning down precise properties of words is not easy when only the subword ordering is available, these results provide a welcome intermediate language for defining more complex formulas.

Related work. We already mentioned works on the logic of concatenation, or the two-variable fragment $\mathsf{FO}^2(A^*, \sqsubseteq)$. Because undecidability appears so easily when reasoning

[2]This is a common situation, shared with, e.g., $\mathsf{FO}(A^*, \cdot)$ and $\mathsf{FO}(\mathbb{N}, <)$.

about words, the focus is often on restricted fragments, typically Σ_1, aka "constraint solving". Decision methods for constraints over words have been considered in several contexts but this usually does not include the subsequence predicate: these works rather consider the prefix ordering, and/or membership in a regular language, and/or functions for taking contiguous subsequences or computing the length of sequences, see, e.g., [1, 13, 19].

Outline of the paper. We provide in Section 2 the basic definitions and results necessary for our later developments. Then we show the undecidability of the Σ_1 fragment (Section 3) before focusing on the decidable fragments (Section 4). Finally, in Section 5, we turn to expressiveness questions.

2. Subwords and their logics

We consider finite words $w, v, ...$ over a given finite alphabet A of letters like $a, b, \ldots$. Concatenation of words is written multiplicatively, with the empty word ε as unit. We freely use regular expressions like $(ab)^* + (ba)^*$ to denote regular languages.

The length of a word w is written $|w|$ while, for a letter $a \in A$, $|w|_a$ denotes the number of occurrences of a in w. The set of all words over A is written A^*.

A word v is a *factor* of w if there exist words w_1 and w_2 such that $w = w_1 v w_2$. If furthermore $w_1 = \varepsilon$ then v is a *prefix* of w, while if $w_2 = \varepsilon$ then v is a *suffix.*

Subwords. We say that a word w is a *subword* (i.e., a subsequence) of v, written $w \sqsubseteq v$, when w is some $a_1 \cdots a_n$ and v can be written as $v_0 a_1 v_1 \cdots a_n v_n$ for some $v_0, v_1, \ldots, v_n \in A^*$, e.g., $\varepsilon \sqsubseteq bba \sqsubseteq ababa$. We write $w \sqsubset v$ for the associated strict ordering, where $w \neq v$. Two words w and v are *incomparable* (with respect to the subword relation), denoted $w \perp v$, if $w \not\sqsubseteq v$ and $v \not\sqsubseteq w$. Factors are a special case of subwords.

With any $w \in A^*$ we associate its upward closure $\uparrow w$, given by $\uparrow w \stackrel{\text{def}}{=} \{v \in A^* \mid w \sqsubseteq v\}$. For example, $\uparrow ab = A^* a A^* b A^*$. The definition of $\uparrow w$ involves an implicit alphabet A that will always be clear from the context.

Piecewise testable languages. Piecewise testable languages (abbreviated PT) constitute a subvariety of the languages of dot-depth one, themselves a subvariety of the star-free languages, which are a subvariety of the regular languages [10]. Among the several characterizations of PT languages, the most convenient for our purposes is the following one: $L \subseteq A^*$ is PT if, and only if, it is a boolean combination of languages of the form $\uparrow w$ for some $w \in A^*$. Thus the PT languages are exactly the monadic predicates that can be defined by a boolean combination of constraints of the form $w_i \sqsubseteq x$ and/or $w_j \not\sqsubseteq x$, or equivalently by a quantifier-free $\varphi_L(x)$ formula in the $\mathsf{FO}(A^*, \sqsubseteq, w_1, \ldots)$ logic. For example, the solutions of φ_2 (from the introduction) form a PT language. In the following, we often write "$x \in L$", where L is a given PT language, as an abbreviation for $\varphi_L(x)$, with the understanding that this is a Σ_0 formula.

Logic of subwords. Let V be the set of variables with typical elements $x, y, \ldots, u, v, \ldots$. For a first-order logic formula φ over a structure with domain D, we denote by $[\![\varphi]\!] \subseteq D^V$ the set of satisfying assignments, with typical elements $\alpha, \beta, \ldots$. If φ has only one free variable, say x, and there is no danger of confusion, we sometimes write $[\![\varphi]\!]$ to mean $\{\alpha(x) \mid \alpha \in [\![\varphi]\!]\}$. Moreover, $\mathsf{fv}(\varphi)$ denotes the set of free variables in φ.

By $\mathsf{FO}(A^*, \sqsubseteq)$, we denote the first-order logic over the structure $(A^*, \sqsubseteq)$. In contrast, $\mathsf{FO}(A^*, \sqsubseteq, w_1, \ldots)$ is the first-order logic over the structure $(A^*, \sqsubseteq, w_1, \ldots)$, where for each word $w \in A^*$, the signature provides a constant symbol. In the case of $\mathsf{FO}(A^*, \sqsubseteq, w_1, \ldots)$ and $\mathsf{FO}(A^*, \sqsubseteq)$, assignments are members of $(A^*)^V$. We will sometimes write w to denote the assignment that maps every variable to the word $w \in A^*$. Moreover, $(x \mapsto w)$ denotes the assignment in $(A^*)^{\{x\}}$ that maps x to w.

Bounding alternations. We define a fragment of first-order logic over the relational structure $(A^*, \sqsubseteq, w_1, \ldots)$. Let $A = \{a_1, \ldots, a_n\}$. The starting point for introducing the fragments $\Sigma_{i,j}$ is the observation that if every variable in a sentence φ is introduced by a restricted quantifier of the form $\exists x \in (a_1^* \cdots a_n^*)^\ell$ or $\forall x \in (a_1^* \cdots a_n^*)^\ell$ for some $\ell \in \mathbb{N}$, then one can reduce the truth problem of φ to Presburger arithmetic. Note that the language $(a_1^* \cdots a_n^*)^\ell$ is PT, implying that such restrictions, which we call *alternation bounds*, can be imposed within $\mathsf{FO}(A^*, \sqsubseteq, w_1, \ldots)$ and without any additional quantifiers. This raises the question of how many variables without alternation bound one can allow without losing decidability.

In essence $\Sigma_{i,j}$ contains all formulas in the Σ_i fragment with j variables without alternation bound. A formalization of this *for sentences* could just be a syntactic restriction: Every quantifier for all but at most j variables must be relative to some $(a_1^* \cdots a_n^*)^\ell$. However, this would not restrict free variables, which we need in order to build complex $\Sigma_{i,j}$ formulas from predicates defined in $\Sigma_{i,j}$.

Formally, a *formula with alternation bounds* consists of a formula φ of $\mathsf{FO}(A^*, \sqsubseteq, w_1, \ldots)$ and a function $\ell \colon V \to \mathbb{N} \cup \{\infty\}$, which specifies the alternation bounds. This means, the semantics $[\![(\varphi, \ell)]\!]$ of (φ, ℓ) is defined as $[\![\tilde{\varphi}]\!]$, where $\tilde{\varphi}$ is defined as follows. First, we replace every quantifier $\mathcal{Q}x$ ($\mathcal{Q} \in \{\exists, \forall\}$) in φ by the relativized $\mathcal{Q}x \in (a_1^* \cdots a_n^*)^{\ell(x)}$. Then we add the conjunction $\bigwedge_{x \in \mathsf{fv}(\varphi), \ell(x) < \infty} x \in (a_1^* \cdots a_n^*)^{\ell(x)}$ for the free variables.

The fragment $\Sigma_{i,j}$ consists of those formulas with alternation bounds (φ, ℓ) where φ belongs to the Σ_i fragment of $\mathsf{FO}(A^*, \sqsubseteq, w_1, \ldots)$ and has at most j variables $x \in V$ with $\ell(x) = \infty$. We will always represent a formula in $\Sigma_{i,j}$ by its Σ_i formula and the function ℓ will be clear from the context. Variables $x \in V$ with $\ell(x) < \infty$ will be called *alternation bounded*, the others *alternation unbounded*. In order to permit a polynomial translation into an equivalent formula in ordinary $\mathsf{FO}(A^*, \sqsubseteq, w_1, \ldots)$, the alternation bounds are always encoded in unary. The fragment $\Pi_{i,j}$ is defined similarly, with Π_i instead of Σ_i.

Sometimes we define predicates that are satisfied for words with unbounded alternations (such as "$u \in \{a, b\}^*$" when $A = \{a, b, c\}$), but want to use the corresponding formula in a context where the variables are alternation bounded ("$u \in \{a, b\}^* \wedge ab \not\sqsubseteq u$"). In that situation, we want to record the number of alternation unbounded variables we need for the definition, *disregarding the free variables.* Hence, $\Sigma'_{i,j}$ denotes those formulas with alternation bound in Σ_i, where there are at most j *quantified variables* without alternation bound. The semantics is defined as for $\Sigma_{i,j}$. The fragment $\Pi'_{i,j}$ is defined with Π_i instead of Σ_i.

3. Undecidability

3.1. The $\Sigma_{1,3}$ fragment.

We begin with our main result, the undecidability of the Σ_1 theory of $\mathsf{FO}(A^*, \sqsubseteq, w_1, \ldots)$ for $|A| \geq 2$. In fact, we will even prove undecidability for the $\Sigma_{1,3}$ fragment. We need a few ingredients. A word $w \in A^+$ is called *primitive* if there is no $v \in A^+$, $|v| < |w|$, with $w \in v^*$. The following is a well-known basic fact from word combinatorics (see e.g. [4, Exercise 2.5])

Fact 3.1. *If $p \in A^+$ is primitive, then $pw = wp$ is equivalent to $w \in p^*$.*

We also use the following version of the fact that Diophantine sets are precisely the recursively enumerable sets [31].

Theorem 3.2. *Let $S \subseteq \mathbb{N}$ be a recursively enumerable set. Then there is a finite set of variables $\{x_0, \ldots, x_m\}$ and a finite set E of equations, each of the form*

$$x_i = x_j + x_k \qquad\qquad x_i = x_j \cdot x_k \qquad\qquad x_i = 1$$

with $i, j, k \in [0, m]$, such that

$$S = \{y_0 \in \mathbb{N} \mid \exists y_1, \ldots, y_m \in \mathbb{N} \colon (y_0, \ldots, y_m) \text{ satisfies } E\}.$$

We are now ready to prove our main result.

Theorem 3.3. *Let $|A| \geq 2$ and $a \in A$. For each recursively enumerable set $S \subseteq \mathbb{N}$, there is a $\Sigma_{1,3}$ formula φ over the structure $\mathsf{FO}(A^*, \sqsubseteq, w_1, \ldots)$ with one free variable such that $[\![\varphi]\!] = \{a^k \mid k \in S\}$.*

Proof. We show how to express some basic properties of words and combine these to build more complex predicates, all the time keeping track of what fragments are involved. Here, we always use u, v, w as the free variables of the formula we currently construct.

Recall that for every PT language $L \subseteq A^*$, we can express "$u \in L$" in $\Sigma'_{0,0}$: we will use this silently, mainly for languages of the form ra^*s where a is a letter and r, s are two words.[3] Note also that, since "$u \in (a+b)^*$" can be expressed in $\Sigma'_{0,0}$ for $a, b \in A$, it suffices to prove the theorem in the case $|A| = 2$.

(1) We can express "$|u|_a < |v|_a$" in $\Sigma'_{1,0}$:
$$\exists x \in a^* \colon x \sqsubseteq v \wedge x \not\sqsubseteq u.$$
(2) We can express "$\exists n \colon u = a^n \wedge v = a^{n-1}b$" in $\Sigma_{1,0}$. Clearly, it suffices to show that we can express "$\exists n \geq 2 \colon u = a^n \wedge v = a^{n-1}b$". Consider the formula:
$$u \in aaa^* \wedge v \in a^*b \wedge \exists x \in a^*baa \colon |v|_a < |u|_a \wedge v \not\sqsubseteq x \wedge u \sqsubseteq x.$$
Suppose the formula is satisfied with $u = a^n$, $x = a^\ell baa$ and $v = a^m b$. Then $|v|_a < |u|_a$ implies $m < n$. By $v \not\sqsubseteq x$, we have $\ell < m$ and thus $\ell < m < n$, hence $\ell + 2 \leq n$. On the other hand, $u \sqsubseteq x$ implies $n \leq \ell + 2$ and thus $n = \ell + 2$ and $m = n - 1$.

Conversely, if $u = a^n$ and $v = a^{n-1}b$ for some $n \geq 2$, then the formula is satisfied with $x = a^{n-2}baa$.

(3) We can express "$u, v \in (a+b)^*b \wedge |u|_a = |v|_a$" in $\Sigma'_{1,0}$:
$$\begin{aligned} & u, v \in (a+b)^* \\ \wedge\ & \exists x \in a^* \colon \exists y \in a^*b \colon \left[\exists n \colon x = a^n \wedge y = a^{n-1}b\right] \\ & \wedge\ y \sqsubseteq u \ \wedge\ y \sqsubseteq v \ \wedge\ x \not\sqsubseteq u \ \wedge\ x \not\sqsubseteq v. \end{aligned}$$
Suppose the formula is satisfied. Then $a^{n-1}b \sqsubseteq u$ and $a^n \not\sqsubseteq u$ imply $|u|_a = n - 1$. Moreover, if u ended in a, then $a^{n-1}b \sqsubseteq u$ would entail $a^n \sqsubseteq u$, which is not the case. Since $|u| \geq 1$, we therefore have $u \in \{a, b\}^*b$. By symmetry, we have $|v|_a = n - 1$ and $v \in \{a, b\}^*b$. Hence, $|u|_a = n - 1 = |v|_a$.

If $u, v \in \{a, b\}^*b$ with $|u|_a = |v|_a$, then the formula is satisfied with $n = |u|_a + 1$.

(4) We can express "$\exists n \colon u = aaba^nb \wedge v = aba^{n+1}b \wedge w = ba^{n+2}b$" in $\Sigma_{1,0}$:
$$\begin{aligned} & u \in aaba^*b \wedge v \in aba^*b \wedge w \in ba^*b \\ \wedge\ & \left[u, v, w \in \{a, b\}^*b \wedge |u|_a = |v|_a = |w|_a\right]. \end{aligned}$$
(5) We can express "$\exists n \colon u = ba^nb \wedge v = ba^{n+1}b$" in $\Sigma_{1,0}$. It suffices to show that we can express "$\exists n \geq 1 \colon u = ba^nb \wedge v = ba^{n+1}b$". Consider the formula:
$$\begin{aligned} & \exists x \in aaba^*b, y \in aba^*b, z \in ba^*b \colon \\ & \left[\exists m \colon x = aaba^mb \wedge y = aba^{m+1}b \wedge z = ba^{m+2}b\right] \\ & \wedge\ u, v \in ba^*b \ \wedge\ u \sqsubseteq y \ \wedge\ u \not\sqsubseteq x \ \wedge\ v \sqsubseteq z \ \wedge\ v \not\sqsubseteq y. \end{aligned}$$
Suppose the formula is satisfied for $u = ba^kb$ and $v = ba^\ell b$. Then $u \sqsubseteq y$ and $u \not\sqsubseteq x$ imply $k \leq m + 1$ and $k > m$, hence $k = m + 1$. Moreover, $v \sqsubseteq z$ and $v \not\sqsubseteq y$ imply $\ell \leq m + 2$ and $\ell > m + 1$, hence $\ell = m + 2$. Hence, with $n = m + 1$ we have $u = ba^nb$ and $v = ba^{n+1}b$ and $n \geq 1$.

Conversely, if $u = ba^nb$ and $v = ba^{n+1}b$ for some $n \geq 1$, then the formula is satisfied with $m = n - 1$.

[3] That any language of the form ra^*s is PT is easy to prove, e.g., using the characterization of [27].

(6) We can express "$\exists n\colon u = a^n \wedge v = a^{n+1}$" in $\Sigma_{1,0}$. For this, it suffices to express "$\exists n \geq 1\colon u = a^n \wedge v = a^{n+1}$". As in Item 5, one verifies correctness of the following:

$$\exists x, y, z\colon \left[\exists m\colon x = ba^m b \wedge y = ba^{m+1}b \wedge z = ba^{m+2}b\right]$$
$$\wedge\ u, v \in a^* \ \wedge\ u \sqsubseteq y \ \wedge\ u \not\sqsubseteq x \ \wedge\ v \sqsubseteq z \ \wedge\ v \not\sqsubseteq y.$$

(7) We can express "$v = a^{|u|_a}$" in $\Sigma'_{1,0}$:

$$\exists x \in a^*\colon \left[\exists n\colon v = a^n \wedge x = a^{n+1}\right] \wedge v \sqsubseteq u \wedge x \not\sqsubseteq u.$$

(8) We can express "$|u|_a = |v|_a$" in $\Sigma'_{1,0}$:

$$\exists x\colon x = a^{|u|_a} \wedge x = a^{|v|_a}.$$

(9) For $a \neq b$, we can express "$u \in a^* \wedge v = bu$" in $\Sigma_{1,0}$:

$$u \in a^* \wedge v \in ba^* \wedge |v|_a = |u|_a.$$

(10) For $a \neq b$, we can express "$u \in a^* \wedge v = ub$" in $\Sigma_{1,0}$:

$$u \in a^* \wedge v \in a^*b \wedge |v|_a = |u|_a.$$

(11) We can express "$|w|_a = |u|_a + |v|_a$" for any $a \in A$ in $\Sigma'_{1,0}$. Let $b \in A \setminus \{a\}$:

$$\exists x, y \in a^*\colon \exists z \in a^*ba^*\colon x = a^{|u|_a} \wedge y = a^{|v|_a}$$
$$\wedge\ xb \sqsubseteq z \wedge xab \not\sqsubseteq z \wedge by \sqsubseteq z \wedge bya \not\sqsubseteq z \tag{1}$$
$$\wedge\ |w|_a = |z|_a \tag{2}$$

Note that we can define xb, $(xa)b$ and $b(ya)$ thanks to Items 6, 9 and 10. The constraints in Eq. (1) enforce that $z = xby$ and hence $|z|_a = |x|_a + |y|_a = |u|_a + |v|_a$.

(12) For $k, n_0, \ldots, n_k \in \mathbb{N}$, $a \neq b$, let $r_a(a^{n_0}ba^{n_1} \cdots ba^{n_k}) = n_k$, which defines a function $r_a\colon \{a, b\}^* \to \mathbb{N}$. We can express "$v = a^{r_a(u)}$" in $\Sigma'_{1,0}$:

$$v \in a^* \wedge\ \exists x \in b^*a^*\colon \exists y \in b^*a^*\colon$$
$$|x|_b = |y|_b = |u|_b \ \wedge\ |y|_a = |x|_a + 1$$
$$\wedge\ x \sqsubseteq u \ \wedge\ y \not\sqsubseteq u \ \wedge\ |v|_a = |x|_a.$$

Note that $|x|_b = |y|_b = |u|_b$ can be expressed according to Item 8 and $|y|_a = |x|_a + 1$ can be expressed thanks to Item 11. Write $u = a^{n_0}ba^{n_1} \cdots ba^{n_k}$.

Suppose the formula is satisfied. Then $|x|_b = |y|_b = |u|_b$ and $|y|_a = |x|_a + 1$ imply that $x = b^k a^m$ and $y = b^k a^{m+1}$ for some $m \in \mathbb{N}$. Moreover, $x \sqsubseteq u$ implies $m \leq n_k$ and $y \not\sqsubseteq u$ implies $m + 1 > n_k$, thus $m = n_k$. Thus, $|v|_a = |x|_a$ entails $|v|_a = n_k$.

Conversely, if $v = a^{n_k}$, then the formula is satisfied with $x = b^k a^{n_k}$ and $y = b^k a^{n_k+1}$.

(13) For $a \in A$, we can express "$v \in a^* \wedge w = uv$" in $\Sigma'_{1,0}$. Let $b \neq a$ and consider the formula

$$v \in a^* \ \wedge$$
$$\wedge\ \exists x \in a^*\ \colon \exists y \in a^*\ \colon x = r_a(u) \ \wedge\ y = r_a(w)$$
$$\wedge\ |w|_b = |u|_b \ \wedge\ u \sqsubseteq w \tag{3}$$
$$\wedge\ |y|_a = |x|_a + |v|_a \ \wedge\ |w|_a = |u|_a + |v|_a \tag{4}$$

To show correctness, suppose the formula is satisfied with $u = a^{n_0}ba^{n_1} \cdots ba^{n_k}$ and $w = a^{m_0}ba^{m_1} \cdots ba^{m_\ell}$. The conditions in Eq. (3) imply that $k = \ell$ and $w = a^{m_0}ba^{m_1} \cdots ba^{m_k}$ and $n_i \leq m_i$ for $i \in [0, k]$. The conditions in Eq. (4) then entail $m_k = n_k + |v|_a$ and $\sum_{i=0}^{k} m_i = \sum_{i=0}^{k} n_i + |v|_a$, which together is only possible if $m_i = n_i$ for $i \in [0, k-1]$. This means we have $w = uv$. The converse is clear.

(14) We can express "u is prefix of v" in $\Sigma_{1,3}$:

$$\bigwedge_{a\in A} \exists x\colon \exists y \in a^*\colon x = uy \ \wedge\ x \sqsubseteq v \ \wedge\ |x|_a = |v|_a.$$

Suppose the formula is satisfied. Then $uy \sqsubseteq v$ for some $y \in A^*$, which implies $u \sqsubseteq v$. Let p be the shortest prefix of v with $u \sqsubseteq p$. Observe that whenever $uw \sqsubseteq v$, we also have $pw \sqsubseteq v$, because the leftmost embedding of uw in v has to match up u with p. Now towards a contradiction, assume $|p| > |u|$. Then there is some $a \in A$ with $|p|_a > |u|_a$. The formula tells us that for some $m \in \mathbb{N}$, we have $ua^m \sqsubseteq v$ and $|u|_a + m = |v|_a$. Our observation yields $pa^m \sqsubseteq v$, and hence $|v|_a \geq |p|_a + m > |u|_a + m = |v|_a$, a contradiction. The converse is clear.

(15) We can express "$w = uv$" in $\Sigma_{1,3}$: Since expressibility is preserved by mirroring, we can express prefix and suffix by Item 14. Let $\sqsubseteq_{\mathsf{p}}$ and $\sqsubseteq_{\mathsf{s}}$ denote the prefix and suffix relation, respectively. We can use the formula

$$u \sqsubseteq_{\mathsf{p}} w \ \wedge\ v \sqsubseteq_{\mathsf{s}} w \ \wedge\ \bigwedge_{a\in A} |w|_a = |u|_a + |v|_a.$$

(16) For $a, b \in A$, $a \neq b$, we can express "$u \in (ab)^*$" in $\Sigma_{1,3}$: By Item 15, we can use the formula $\exists v\colon v = uab \wedge v = abu$, which, according to Fact 3.1, is equivalent to $u \in (ab)^*$.

(17) For $a, b \in A$, $a \neq b$, we can express "$|u|_a = |v|_b$" in $\Sigma_{1,3}$ by using

$$\exists x \in (ab)^*\colon |u|_a = |x|_a \wedge |v|_b = |x|_b.$$

(18) We can express "$\exists m, n\colon u = a^n \wedge v = a^m \wedge w = a^{m\cdot n}$" in $\Sigma_{1,3}$:

$$\begin{aligned} & u, v, w \in a^* \\ \wedge\ \exists x\colon & [\exists y, z\colon y = bu \ \wedge\ z = yx \ \wedge\ z = xy] \\ & \wedge\ |x|_b = |v|_a \ \wedge\ |w|_a = |x|_a. \end{aligned}$$

The conditions in brackets require $(bu)x = x(bu)$. Since $bu \in ba^*$ is primitive, this is equivalent to $x \in (bu)^*$ (*cf* Fact 3.1).

(19) We use the fact that every recursively enumerable set of natural numbers is Diophantine. Applying Theorem 3.2 to S yields a finite set E of equations over the variables $\{x_0, \ldots, x_m\}$. The formula φ is of the form

$$\varphi \equiv \exists x_1, x_2, \ldots, x_m \in a^*\colon \psi,$$

where ψ is a conjunction of the following $\Sigma_{1,3}$ formulas. For each equation $x_i = 1$, we add $x_i = a$. For each equation $x_i = x_j + x_k$, we add a formula expressing $|x_i|_a = |x_j|_a + |x_k|_a$. For each equation $x_i = x_j \cdot x_k$, we add a formula expressing $x_i = a^{|x_j|\cdot|x_k|}$. Then we clearly have $[\![\varphi]\!] = \{a^k \mid k \in S\}$. $\square$

As an immediate consequence, one sees that the truth problem is also undecidable for the Σ_1 fragment of the logic of subwords without constants but enriched with predicates like "$|u|_a = 2$" for counting letter occurrences.

It can even be shown that there is a fixed word $W \in \{a, b\}^*$ such that the truth problem of $\Sigma_{1,3}$ over $\mathsf{FO}(\{a,b\}^*, \sqsubseteq, W)$ is undecidable. In order to show undecidability with a single constant, we will need the fact that each word of length at least 3 is determined by its length and its strict subwords. For two words $u, v \in A^*$, we write $u \sim_n v$ if $\downarrow\{u\} \cap A^{\leq n} = \downarrow\{v\} \cap A^{\leq n}$.

Lemma 3.4 ([23])**.** *Let $n \geq 2$ and $|u| = |v| = n + 1$. Then $u = v$ if and only if $u \sim_n v$.*

Theorem 3.5. *There is a word $W \in \{a, b\}^*$ such that for every recursively enumerable set $S \subseteq \mathbb{N}$, there is a $\Sigma_{1,3}$-formula τ over the structure $\mathsf{FO}(\{a,b\}^*, \sqsubseteq, W)$ such that*

$$[\![\tau]\!] = \{a^k \mid k \in S\}.$$

In particular, the truth problem for the $\Sigma_{1,3}$ fragment over $\mathsf{FO}(\{a,b\}^*, \sqsubseteq, W)$ is undecidable.

Proof. In the proof of Theorem 3.3, we have constructed $\Sigma_{1,3}$ formulas over $\mathsf{FO}(A^*, \sqsubseteq, w_1, \ldots)$ expressing successor, addition, and multiplication, more precisely: expressing "$\exists n \geq 0\colon u = a^n \wedge v = a^{n+1}$" and "$\exists m,n \geq 0\colon u = a^m \wedge v = a^n \wedge w = a^{m+n}$" and "$\exists m,n \geq 0\colon u = a^n \wedge v = a^m \wedge w = a^{m\cdot n}$". Let $W_1, \ldots, W_r \in \{a,b\}^*$ be the constants occurring in these three $\Sigma_{1,3}$ formulas, plus ε. Let m the maximal length of any of these words, and let $W = a^{m+1}b^{m+2}$.

Let $S \subseteq \mathbb{N}$ be recursively enumerable. Then, according to Theorem 3.2 and by the choice of $W_1, \ldots, W_r$, there is a $\Sigma_{1,3}$ formula φ that only uses constants from $W_1, \ldots, W_r$ and with $[\![\varphi]\!] = \{a^k \mid k \in S\}$. We shall prove that using W, we can define all the words $W_1, \ldots, W_r$. Consider the formula

$$\begin{array}{ll}
\exists x_0, y_0, \ldots, x_{2m+3}, y_{2m+3}\colon & x_0 \sqsubset \cdots \sqsubset x_{2m+3} \sqsubseteq W \wedge y_0 \sqsubset \cdots \sqsubset y_{2m+3} \sqsubseteq W \wedge \\
\exists x'_0, \ldots, x'_{m+1}\colon & x'_0 \sqsubset \cdots \sqsubset x'_{m+1} \sqsubseteq W \;\wedge \\
\exists y'_0, \ldots, y'_{m+2}\colon & y'_0 \sqsubset \cdots \sqsubset y'_{m+2} \sqsubseteq W \;\wedge \\
 & x_1 \neq y_1 \;\wedge\; x_1 \not\sqsubseteq y'_{m+2} \;\wedge\; y_1 \not\sqsubseteq x'_{m+1} \;\wedge \\
\exists z_{01}\colon & x'_1 \sqsubseteq z_{01} \;\wedge\; x'_2 \not\sqsubseteq z_{01} \;\wedge\; y'_1 \sqsubseteq z_{01} \;\wedge\; y'_2 \not\sqsubseteq z_{01} \;\wedge \\
\exists z_{10}\colon & x'_1 \sqsubseteq z_{10} \;\wedge\; x'_2 \not\sqsubseteq z_{10} \;\wedge\; y'_1 \sqsubseteq z_{10} \;\wedge\; y'_2 \not\sqsubseteq z_{10} \;\wedge \\
 & z_{01} \sqsubseteq W \;\wedge\; z_{01} \neq z_{10}
\end{array}$$

If it is satisfied, then $|x_i| = |y_i| = i$ for $i \in [0, 2m+3]$ and since $x_1 \neq y_1$, we have $\{x_1, y_1\} = \{a, b\}$. Since $x_1 \not\sqsubseteq y'_{m+2}$ we get $y'_i \in y_1^*$ and thus $y'_i = y_1^i$ for $i \in [0, m+2]$, which is only possible with $y_1 = b$. This implies $x_1 = a$. In particular, we get $\{z_{01}, z_{10}\} = \{ab, ba\}$. Since $z_{01} \sqsubseteq W$, we have $z_{01} = ab$ and $z_{10} = ba$. On the other hand, if $|x_i| = |y_i| = i$ for $i \in [0, 2m+3]$, $x_1 = a$, $y_1 = b$, $x'_i = a^i$, $y'_j = b^j$ for $i \in [0, m+1]$, $j \in [0, m+2]$, $z_{01} = ab$, and $z_{10} = ba$, then the formula is clearly satisfied.

Hence, we can already define all words of length at most 2 and all words a^i and b^i for $i \in [0, m+1]$. This lets us define other predicates.

(1) For each $0 \leq \ell \leq m$, we can express "$|u|_a = \ell$" using the formula
$$a^\ell \sqsubseteq u \;\wedge\; a^{\ell+1} \not\sqsubseteq u$$
Note that since $\ell + 1 \leq m+1$, we can already define $a^{\ell+1}$. The same way, we can express "$|u|_b = \ell$".

(2) For each $0 \leq \ell \leq m$, we can express "$|u| = \ell$" using the formula
$$\bigvee_{i+j=\ell} |u|_a = i \wedge |u|_b = j.$$

(3) For each word $w \in A^{\leq m}$, $|w| > 2$, we can define w. We proceed by induction. For $|w| \leq 2$, we can already define w. Thus, suppose we can define every $v \in A^{\leq n}$ and let $w \in A^{n+1}$ with $n+1 \leq m$. Consider the formula
$$|u| = n+1 \;\wedge \bigwedge_{v \in A^{\leq n},\; v \sqsubseteq w} v \sqsubseteq u \;\wedge \bigwedge_{v \in A^{\leq n},\; v \not\sqsubseteq w} v \not\sqsubseteq u.$$
Clearly, if $u = w$, then the formula is satisfied. On the other hand, suppose the formula is satisfied. It expresses that $\downarrow\{u\} \cap A^{\leq n} = \downarrow\{w\} \cap A^{\leq n}$. According to Lemma 3.4, this implies $u = w$.

Note that all the variables we introduced to define the words in $A^{\leq m}$ carry words of length at most $2m+3$, meaning that we may assume that they are alternation bounded. Thus, we can define all words in $A^{\leq m}$ using forumlas in $\Sigma_{1,0}$. Therefore, we can turn φ into a $\Sigma_{1,3}$ formula τ that contains W as its only constant and satisfies $[\![\tau]\!] = [\![\varphi]\!]$.

It remains to show the second statement of the theorem. Let $S \subseteq \mathbb{N}$ be recursively enumerable but undecidable and let $k \in \mathbb{N}$ be given. We choose the formula φ as above, but we modify it as follows. Let $\varphi_0 \equiv \varphi$, and for $i \in [1,k]$, let φ_i express

$$\exists y\colon \varphi_{i-1}(y) \wedge \exists n\colon x = a^n \wedge y = a^{n+1}.$$

Finally, let φ_{k+1} be the formula $\exists x\colon \varphi_k(x) \wedge x = \varepsilon$. Note that by the choice of $W_1, \dots, W_r$, we may assume that φ_{k+1} contains only the constants $W_1, \dots, W_r$. Note that φ_{k+1} has no free variables and is true if and only if $k \in S$. Now τ_{k+1} is obtained from φ_{k+1} just as τ is obtained from φ. It follows as above that τ_{k+1} is true if and only if $k \in S$. This proves the second statement of the theorem. ☐

Here W must be complex enough: For instance, the Σ_1 fragment of $\mathsf{FO}(\{a,b\}^*, \sqsubseteq, \varepsilon)$ and of $\mathsf{FO}(\{a,b\}^*, \sqsubseteq, a)$, respectively, is decidable.

Theorem 3.6. *The Σ_1-fragment of $\mathsf{FO}(\{a,b\}^*, \sqsubseteq, \varepsilon, a)$ is decidable.*

Proof. We may assume that the input formula is of the form $\varphi \equiv \exists x_1, \dots, x_n\colon \psi$, where ψ is a conjunction of literals of the following forms:

$$\begin{array}{cccc} c \sqsubseteq x & c \not\sqsubseteq x & x \sqsubseteq c & x \not\sqsubseteq c \\ & x \sqsubseteq y & x \not\sqsubseteq y & \end{array}$$

where $c \in \{\varepsilon, a\}$ and $x \in X = \{x_1, \dots, x_n\}$. For each literal $x \sqsubseteq c$, we can guess whether $x = \varepsilon$ or $x = a$ and hence assume that these do not occur. Literals of the form $\varepsilon \sqsubseteq x$ are always satisfied, whereas $\varepsilon \not\sqsubseteq x$ is never satisfied. Hence, without loss of generality, these do not occur either and we may assume that all literals are of the form

$$\begin{array}{cccc} a \sqsubseteq x & a \not\sqsubseteq x & x \not\sqsubseteq \varepsilon & x \not\sqsubseteq a \\ & x \sqsubseteq y & x \not\sqsubseteq y. & \end{array}$$

Moreover $x \not\sqsubseteq \varepsilon$ is equivalent to $x \neq \varepsilon$ and the literal $a \not\sqsubseteq x$ is equivalent to $x \in b^*$. We can therefore assume that all literals are of the form

$$\begin{array}{cccc} a \sqsubseteq x & x \in b^* & x \neq \varepsilon & x \not\sqsubseteq a \\ & x \sqsubseteq y & x \not\sqsubseteq y & \end{array}$$

Let $L \subseteq X$ be the set of those variables for which we have a $x \in b^*$ literal. Clearly, $x \in b^*$ and $x \neq \varepsilon$ together mean $x \in b^+$. In the same way, $x \in b^*$ and $x \not\sqsubseteq a$ together mean $x \in b^+$. Furthermore, $a \sqsubseteq x$ and $x \in b^*$ are mutually exclusive. Hence, we can rewrite our constraint system as follows:

- For each $x \in L$, we have either a constraint $x \in b^*$ or $x \in b^+$.
- For each $x \in X \setminus L$, we have a set of constraints of the form $a \sqsubseteq x$, $x \neq \varepsilon$, or $x \not\sqsubseteq a$.
- We have constraints of the form $x \sqsubseteq y$ and $x \not\sqsubseteq y$.

As a final reformulation step, note that every $u \in \{a,b\}^*$ satisfies either $u \in b^*$ or $a \sqsubseteq u$. Therefore, we may assume that for every $x \in X$, we have either a constraint $x \in b^*$ or $x \in b^+$ (and hence $x \in L$) or $a \sqsubseteq x$. Notice that if we already have $a \sqsubseteq x$, then $x \neq \varepsilon$ is redundant. Thus, we have the following constraints:

(1) For each $x \in L$, we have either $x \in b^*$ or $x \in b^+$.
(2) For each $x \in X \setminus L$, we have $a \sqsubseteq x$ and possibly $x \not\sqsubseteq a$.
(3) A set of constraints of the form $x \sqsubseteq y$ or $x \not\sqsubseteq y$.

We say that a partial order $(X, \leq)$ is *compatible* if

(1) L is downward closed and linearly ordered,
(2) for each constraint $x \sqsubseteq y$ ($x \not\sqsubseteq y$), we have $x \leq y$ ($x \not\leq y$).

We claim that φ is satisfied in $\mathsf{FO}(\{a,b\}^*, \sqsubseteq, \varepsilon, a)$ if and only if there is a compatible partial order on X. Since the latter is clearly decidable, this implies the theorem.

Of course, if φ is satisfied, then the subword ordering induces a compatible partial order on X. So let us prove the converse and suppose $(X, \leq)$ is a compatible partial order and let $P = X \setminus L$. Then, $(P, \leq)$ is a partial order and we can find some $m \geq 0$ such that $(P, \leq)$

embeds into the lattice $\{0,1\}^m$ of m-tuples over $\{0,1\}$ with componentwise comparison. Consider such an embedding with $m \geq 2$. This embedding allows us to assign to each $x \in P$ a word $u_x \in a\beta_1 \cdots a\beta_m$, where $\beta_1, \ldots, \beta_m \in \{\varepsilon, b\}$ such that $x \leq y$ if and only if $u_x \sqsubseteq u_y$.

Now write $L = \{\ell_1, \ldots, \ell_k\}$ with $\ell_1 \leq \cdots \leq \ell_k$. We now define a function $f\colon X \to \{0, \ldots, k\}$. Note that for each $x \in P$, the set $\downarrow\{x\} \cap L$ is a downward closed subset of L and hence of the form $\{\ell_1, \ldots, \ell_i\}$ for some $i \geq 0$. In this case, set $f(x) = i$. Moreover, let

$$P_i = \{x \in P \mid \downarrow\{x\} \cap L = \{\ell_1, \ldots, \ell_i\}\}.$$

This allows us to construct an assignment of words v_x to variables x. Let us explain the intuition. In order ensure that $v_{\ell_1} \not\sqsubseteq v_x$ for all $x \in P_0$, we let $v_x = u_x$ and notice that then, the words v_x all contain at most m-many b's. Hence, we set $v_{\ell_i} = b^{m+1}$. Now, we have to make sure that the words for v_x with $x \in P_1$ all contain v_{ℓ_1} as a subword, so we pad the words u_x with b's on the left: We set $v_x = b^{m+1}u_x$. Now, in turn, we need to make sure that v_{ℓ_2} contains more than $(m+1)+m$-many b's, leading to $v_{\ell_2} = b^{2(m+1)}$, and so on. Thus, we set:

$$v_{\ell_i} = b^{i(m+1)} \qquad\qquad v_x = b^{f(x)\cdot(m+1)}u_x$$

for $i \in \{1, \ldots, k\}$ and $x \in P$. Let us show that this assignment sastisfies our constraint system.

- Consider a constraint $x \sqsubseteq y$. We have $x \leq y$.
  - If $x, y \in P$, then $\downarrow\{x\} \cap L \subseteq \downarrow\{y\} \cap L$ and thus $f(x) \leq f(y)$. Since also $u_x \sqsubseteq u_y$, we have $v_x \sqsubseteq v_y$.
  - If $y \in L$, then also $x \in L$ (since L is downward closed) and thus clearly $v_x \sqsubseteq v_y$.
  - If $y \in P$ and $x \in L$. Suppose $x = y_i$ and $f(y) = j$. By definition of f, we have $i \leq j$ and hence $v_x = v_{\ell_i} = b^{i(m+1)} \sqsubseteq b^{j(m+1)}u_y = v_y$.
- Consider a constraint $x \not\sqsubseteq y$. Then $x \not\leq y$.
  - If $x, y \in P$, then $u_x \not\sqsubseteq u_y$ by choice of u_x and u_y. In particular, we have $v_x = b^{f(x)\cdot(m+1)}u_x \not\sqsubseteq b^{f(y)\cdot(m+1)}u_y = v_y$.
  - If $x \in P$ and $y \in L$, then $v_y \in b^*$ and $a \sqsubseteq v_x$. Thus $v_x \not\sqsubseteq v_y$.
  - If $x \in L$ and $y \in P$, say $x = \ell_i$. Then $x \not\leq y$ means that $\ell_i \notin \downarrow\{y\} \cap L$ and hence $f(y) < i$. Note that $v_y = b^{f(y)(m+1)}u_y$ and that $|u_y|_b \leq m$. Therefore $|v_y|_b \leq f(y)(m+1) + m < i(m+1)$ and hence $v_x = v_{\ell_i} = b^{i(m+1)} \not\sqsubseteq v_y$.
  - If $x, y \in L$, then $x = y_i$ and $y = y_j$ with $j < i$. Then clearly $v_x \not\sqsubseteq v_y$.
- Constraints $x \in b^*$ or $x \in b^+$ with $x \in L$ are of course satisfied.
- Constraints $x \not\sqsubseteq a$ with $x \in P$ are satisfied because $|u_x|_a \geq m \geq 2$ for every $x \in P$.

This established our claim and thus the theorem. □

This raises an interesting question: For which sets $\{W_1, W_2, \ldots\} \subseteq A^*$ of constants is the truth problem for Σ_1 sentences over $\mathsf{FO}(A^*, \sqsubseteq, W_1, W_2, \ldots)$ decidable?

3.2. The $\Sigma_{2,1}$ fragment.

Our next result is that if we allow one more quantifier alternation, then already one variable without alternation bound is sufficient to prove undecidability.

Theorem 3.7. *Let $|A| \geq 2$ and $a \in A$. For each recursively enumerable set $S \subseteq \mathbb{N}$, there is a $\Sigma_{2,1}$ formula φ over the structure $\mathsf{FO}(A^*, \sqsubseteq, w_1, \ldots)$ with one free variable such that $[\![\varphi]\!] = \{a^k \mid k \in S\}$. In particular, the truth problem for $\Sigma_{2,1}$ is undecidable.*

Proof. (1) We can express "$|u|_a \leq |v|_a$" in $\Pi'_{1,0}$:

$$\forall x \in a^*\colon x \not\sqsubseteq u \vee x \sqsubseteq v.$$

Hence, we can express "$|u|_a = |v|_a$" in $\Pi'_{1,0}$.

(2) We can express "$|u|_a > |v|_a$" in $\Pi'_{1,0}$: This follows from the fact that $|u|_a \leq |v|_a$ is expressible in $\Sigma'_{1,0}$.

(3) We can express "$|u|_a \neq |v|_a$" in $\Pi'_{1,0}$ according to the previous item.

(4) We can express "$u \in a^* \wedge v \in (bu)^*$" in $\Pi_{1,1}$. It clearly suffices to express "$u \in a^* \wedge v \in (bu)^+$" in $\Pi_{1,1}$. Consider the formula

$$v \in b\{a,b\}^* \ \wedge\ \forall x \in b^+a^*b^* \colon \big[|x|_b \neq |v|_b \\ \vee\ (|x|_a > |u|_a \wedge x \not\sqsubseteq v)\ \vee\ (|x|_a \leq |u|_a \wedge x \sqsubseteq v)\big].$$

Note that "$v \in b\{a,b\}^*$" is expressible in $\Pi'_{1,0}$ because $v \in a\{a,b\}^*$ is expressible in $\Sigma'_{1,0}$ (see Item 3 in the proof of Theorem 3.3).

Moreover, notice that since $b^*a^*b^* = \{a,b\}^* \smallsetminus \uparrow aba$, the language $b^+a^*b^* = (b^*a^*b^*) \cap (\uparrow ba \cup (\uparrow b \smallsetminus \uparrow a))$ is piecewise testable and thus definable in $\Sigma'_{0,0}$.

(5) We can express "$|u|_a = |v|_b$" in $\Sigma'_{2,1}$:

$$\exists x \in (ab)^* \colon |u|_a = |x|_a \wedge |v|_b = |x|_b.$$

(6) We can express "$\exists m,n \colon u = a^m \wedge v = a^n \wedge w = a^{m \cdot n}$" in $\Sigma_{2,1}$:

$$u \in a^* \wedge v \in a^* \wedge w \in a^* \wedge \exists y \in b^* \colon \exists x \in (bu)^* \colon \\ |x|_b = |y|_b \wedge |y|_b = |v|_a \wedge |x|_a = |w|_a.$$

Note that we employ the variable y because directly expressing $|x|_b = |v|_a$ (using the previous item) would require an additional alternation unbounded variable besides x, but we can only use one.

(7) Recall that "$|w|_a = |u|_a + |v|_a$" is expressible in $\Sigma'_{1,0}$ (see Item 11 of Theorem 3.3) and hence "$\exists m,n \colon u = a^m \wedge v = a^n \wedge w = a^{m+n}$" in $\Sigma_{1,0}$. Thus, we can implement Diophantine equations as in the proof of Theorem 3.3.

□

3.3. **The Σ_2 fragment over two letters in the pure logic.** The final result in this section settles the question of how many letters are needed to make the Σ_2 fragment of $\mathsf{FO}(A^*, \sqsubseteq)$ undecidable. We show here that two letters suffice. Observe that if $|A| = 1$, $\mathsf{FO}(A^*, \sqsubseteq)$ can be interpreted in $\mathsf{FO}(\mathbb{N}, <)$ and is thus decidable.

Let $\mu \colon A^* \to A^*$ be a map. It is called a *morphism* if $\mu(uv) = \mu(u)\mu(v)$ for all $u, v \in A^*$. It is an *anti-morphism* if $\mu(uv) = \mu(v)\mu(u)$ for all $u, v \in A^*$. Finally, it is an *automorphism* of $(A^*, \sqsubseteq)$ if for any $u, v \in A^*$, we have $u \sqsubseteq v$ if and only if $\mu(u) \sqsubseteq \mu(v)$.

Note that if we have no constants, we cannot define a language in a^*, because all definable subsets are closed under automorphisms of $(A^*, \sqsubseteq)$. It will be useful for the next proof to have a classification of all automorphisms of $(A^*, \sqsubseteq)$. The following is shown implicitly by Kudinov et. al. [29], but we include a short proof for completeness.

Lemma 3.8. *A map μ is an automorphism of $(A^*, \sqsubseteq)$ if and only if*

(i) μ is either a morphism or an anti-morphism and
(ii) μ permutes A.

Proof. Clearly, maps as described in the lemma are automorphisms. Assume μ is an automorphism. Since μ has to preserve the minimal element, we have $\mu(\varepsilon) = \varepsilon$. It also has to preserve the set of minimal elements of $A^* \smallsetminus \{\varepsilon\}$, hence the set A. Repeating this argument yields that μ has to preserve length. Therefore, according to Lemma 3.4, if μ is identical on $A^{\leq n}$ for $n \geq 2$, then it is the identity on $A^{\leq n+1}$. By induction, this implies that if an automorphism is identical on $A^{\leq 2}$, then it is the identity on A^*. Hence, if two automorphisms agree on $A^{\leq 2}$, then they are the same. It therefore suffices to show that every automorphism μ agrees on $A^{\leq 2}$ with a map as described in the lemma.

Since μ preserves the set A, the map $\pi = \mu|_A$ is a permutation of A. Moreover, for any $a, b \in A$, we have $\mu(ab) = \mu(a)\mu(b)$ or $\mu(ab) = \mu(b)\mu(a)$. If $\mu(ab) = \mu(a)\mu(b)$, then we cannot have $\mu(bc) = \mu(c)\mu(b)$, because the two words ab and bc have only one common upper bound of length three (namely abc), whereas the words $\mu(a)\mu(b)$ and $\mu(c)\mu(b)$ have two, namely $\mu(c)\mu(a)\mu(b)$ and $\mu(a)\mu(c)\mu(b)$. Therefore, if $\mu(ab) = \mu(a)\mu(b)$, then $\mu(bc) = \mu(b)\mu(c)$. In particular, if $\mu(ab) = \mu(a)\mu(b)$, then we have $\mu(cd) = \mu(c)\mu(d)$ for all $c, d \in A$. Hence on $A^{\leq 2}$, μ agrees with a map as decribed. ☐

Corollary 3.9. *Let $|A| \geq 2$. For each recursively enumerable set $S \subseteq \mathbb{N}$, there is a Σ_2 formula τ over the structure $\mathsf{FO}(A^*, \sqsubseteq)$ that defines the language*

$$[\![\tau]\!] = \{a^k \mid a \in A,\ k \in S\}.$$

In particular, the truth problem for Σ_2 is undecidable.

Proof. Fix a letter $a \in A$. Let $S \subseteq \mathbb{N}$ be recursively enumerable, let φ be the $\Sigma_{1,3}$ formula provided by Theorem 3.3 with one free variable x and with $[\![\varphi]\!] = \{a^k \mid k \in S\}$, and let $w_1, \ldots, w_m \in A^*$ be the constants used in the formula φ.

It was shown in [23] that from $w_1, \ldots, w_m$, one can construct a Σ_2 formula ψ over $\mathsf{FO}(A^*, \sqsubseteq)$ with free variables $V = \{x_1, \ldots, x_m\}$ such that for $\alpha \in (A^*)^V$, we have $\alpha \in [\![\psi]\!]$ if and only if there is an automorphism $\bar{\cdot}\colon A^* \to A^*$ such that $\alpha(x_i) = \overline{w_i}$ for every $i \in [1, m]$.

Let φ' be the formula obtained from φ by replacing every occurrence of w_i with x_i. Moreover, let $\tau \equiv \exists z_1, \ldots, z_r \colon \psi \wedge \varphi'$.

Then, τ is clearly a Σ_2 formula and has exactly one free variable, say x. We claim that

$$[\![\tau]\!] = \{b^k \mid b \in A,\ k \in S\}. \tag{5}$$

If $k \in S$, then $a^k \in [\![\varphi]\!]$ and hence clearly $b^k \in [\![\tau]\!]$ for each $b \in A$. Moreover, if $w \in [\![\tau]\!]$, then for some $\alpha \in [\![\psi]\!]$, we have $\alpha_w \models \varphi'$, where α_w denotes the assignment with $\alpha_w|_V = \alpha$ and $\alpha(x) = w$. This means, there is an automorphism $\bar{\cdot}$ of $(A^*, \sqsubseteq)$ such that $\alpha(x_i) = \overline{w_i}$ for $i \in [1, m]$. Therefore, there is some $w' \in A^*$ that satisfies φ such that $w = \overline{w'}$. In particular, $w' = a^k$ for some $k \in S$ and hence $w = b^k$ for some $b \in A$. This proves Eq. (5).

We can now proceed as in Theorem 3.5 to show undecidability of the truth problem. ☐

4. Complexity

In this section, we study the complexity of the truth problem for the $\Sigma_{i,j}$ fragments of $\mathsf{FO}(A^*, \sqsubseteq, w_1, \ldots)$.

4.1. **Complexity of $\Sigma_{i,0}$.** We begin with the case $j = 0$. In the following, $\mathsf{\Sigma}_n^{\mathsf{EXP}}$ denotes the n-th level of the weak EXP hierarchy [15, 18].

Theorem 4.1. *If $|A| \geq 2$, then the truth problem for $\Sigma_{i,0}$ is NP-complete for $i = 1$ and $\mathsf{\Sigma}_{i-1}^{\mathsf{EXP}}$-complete for $i > 1$.*

We provide a polynomial inter-reduction with the Σ_i fragment of $\mathsf{FO}(\mathbb{N}, 0, 1, +, <)$, a.k.a. Presburger Arithmetic (PA), for which Haase [17] has recently proven $\mathsf{\Sigma}_{i-1}^{\mathsf{EXP}}$-completeness for $i > 1$. The Σ_1 fragment of PA is NP-complete [32].

The reduction from PA to $\Sigma_{1,0}$ fixes a letter $a \in A$ and encodes every number $k \in \mathbb{N}$ by a^k. Addition can then be expressed in $\Sigma_{1,0}$ (Item 11 of Theorem 3.3). Note that although this ostensibly works with one letter, we need another letter in A to express addition. This is crucial: If $|A| = 1$, then $\mathsf{FO}(A^*, \sqsubseteq, w_1, \ldots)$ is just $\mathsf{FO}(\mathbb{N}, <)$, which has a PSPACE-complete truth problem [12, 35]. Moreover, an inspection of the proof of Theorem 3.3 shows that an alternation bound of $\ell = 2$ suffices to define addition, which is tight: if we only use a bound of 1, we can also easily reduce to $\mathsf{FO}(\mathbb{N}, <)$.

The reduction from $\Sigma_{i,0}$ to Presburger arithmetic encodes a word w known to belong to $(a_1^* \cdots a_n^*)^\ell$, i.e., of the form

$$\prod_{i=1}^{\ell} \prod_{j=1}^{n} a_j^{x_{i,j}},$$

by the vector $(x_{1,1}, \ldots) \in \mathbb{N}^{\ell \cdot n}$ of exponents. With this encoding, it suffices to show how to express literals $w \sqsubseteq w'$ (and also $w \not\sqsubseteq w'$) by polynomial-size existential Presburger formulas for $w, w' \in (a_1^* \cdots a_n^*)^\ell$. For a vector $x = (x_{1,1}, \ldots, x_{\ell,n})$ from $\mathbb{N}^{\ell \cdot n}$, let $w_x = \prod_{i=1}^{\ell} \prod_{j=1}^{n} a_j^{x_{i,j}}$.

Proposition 4.2. *There are existential Presburger formulas φ and ψ of size polynomial in n and ℓ such that*

$$\varphi(x_{1,1},\dots,x_{\ell,n},y_{1,1},\dots,y_{\ell,n}) \iff w_x \sqsubseteq w_y,$$
$$\psi(x_{1,1},\dots,x_{\ell,n},y_{1,1},\dots,y_{\ell,n}) \iff w_x \not\sqsubseteq w_y.$$

Let us briefly describe these formulas. Let $I = [1,\ell]\times[1,n]$ and order the pairs $(i,j)\in I$ lexicographically: $(i',j') < (i,j)$ if $i' < i$ or $i = i'$ and $j' < j$. This captures the order of the $a_j^{x_{i,j}}$ factors in w_x. We now define formulas $\tau_{i,j}$ and $\eta_{i,j}$ where the $t_{i,j,k}$'s and $e_{i,j,k}$'s are extra free variables:

$$\tau_{i,j}: \qquad \bigwedge_{1\le k\le \ell} t_{i,j,k} = \begin{cases} 0 & \text{if } e_{i',j',k'} > 0 \text{ for some} \\ & \quad (i',j') < (i,j) \text{ and } k' > k \\ y_{k,j} - \sum_{i'=1}^{i-1} e_{i',j,k} & \text{otherwise} \end{cases}$$

$$\eta_{i,j}: \qquad \bigwedge_{1\le k\le \ell} e_{i,j,k} = \min\left\{ t_{i,j,k}\ ,\ x_{i,j} - \sum_{r=1}^{k-1} e_{i,j,r} \right\}$$

These expressions define the leftmost embedding of w_x into w_y: the variable $t_{i,j,k}$ describes how many letters from $a_j^{y_{k,j}}$ are available for embedding the $a_j^{x_{i,j}}$ factor of w_x into w_y. The variable $e_{i,j,k}$ counts how many of these available letters are actually used for the $a_j^{x_{i,j}}$ factor in the left-most embedding of w_x into w_y. Since i and j,k are bounded by n and ℓ, we have polynomially many formulas of polynomial size.

Define $\xi = \bigwedge_{(i,j)\in I} \tau_{i,j} \wedge \eta_{i,j}$ and the formulas φ,ψ as:

$$\begin{array}{r}\exists t_{1,1,1}\cdots\exists t_{\ell,n,\ell} \\ \exists e_{1,1,1}\cdots\exists e_{\ell,n,\ell}\end{array} : \xi \wedge \bigwedge_{(i,j)\in I}\left(x_{i,j} \le \sum_{k=1}^{\ell} e_{i,j,k}\right) \tag{φ}$$

$$\begin{array}{r}\exists t_{1,1,1}\cdots\exists t_{\ell,n,\ell} \\ \exists e_{1,1,1}\cdots\exists e_{\ell,n,\ell}\end{array} : \xi \wedge \bigvee_{(i,j)\in I}\left(x_{i,j} > \sum_{k=1}^{\ell} e_{i,j,k}\right) \tag{ψ}$$

Since formulas $\tau_{i,j}$ and $\eta_{i,j}$ are inductive equations that uniquely define the values of $t_{i,j,k}$ and $e_{i,j,k}$ as functions of the x and y vectors, ψ is equivalent to the negation of φ. Moreover, φ expresses that there is enough room to embed each factor $a_j^{x_{i,j}}$ in w_y, i.e., that $w_x \sqsubseteq w_y$ as claimed, and both formulas are easily constructed in polynomial time.

4.2. **Complexity of $\Sigma_{1,1}$.**

Theorem 4.3. *The truth problem for the $\Sigma_{1,1}$ fragment is* NP*-complete.*

Of course, hardness is inherited from $\Sigma_{1,0}$. Conversely, NP-membership is shown by a reduction to the $\Sigma_{1,0}$ fragment. For this reduction, we first explain how a single "unbounded" word can be made alternation bounded while respecting its relationships with other alternation bounded words.

For this we use a slightly different measure of alternation levels for words: we factor words in *blocks* of repeating letters, writing $u = \prod_{i=1}^{k} {a_i}^{\ell_i}$ with $\ell_i > 0$ and $a_i \neq a_{i+1}$ for all i. By "an a-block of u" we mean an occurrence of a factor $a_i^{\ell_i}$ with $a_i = a$. We note that requiring some bound in the number of blocks is equivalent to bounding the number of alternations when it comes to defining the $\Sigma_{i,j}$ fragments. However, counting blocks is more precise.

Lemma 4.4. *Let $t, x_1,\dots,x_n, y_1,\dots,y_m \in A^*$ such that:*

- *for all i, $x_i \sqsubseteq t$,*
- *for all j, $y_j \not\sqsubseteq t$,*
- *for all i and j, x_i and y_j have less than ℓ blocks,*
- *t has $k > (m+n)\cdot\ell + |A|$ blocks.*

Then there exists $t' \in A^*$ such that:

- *for all i, $x_i \sqsubseteq t'$,*
- *for all j, $y_j \not\sqsubseteq t'$,*
- *t' has either $k-1$ or $k-2$ blocks.*

Proof. Given $u \in A^*$, we write $\operatorname{Im} u$ for the image of the left-most embedding of u into t. This is a set of positions in t and, in case $u \not\sqsubseteq t$, these positions only account for the longest prefix of u that can be embedded in t. In particular, and since we assumed $x_i \sqsubseteq t$ and $y_j \not\sqsubseteq t$, then $|\operatorname{Im} x_i| = |x_i|$ and $|\operatorname{Im} y_j| < |y_j|$ for all i, j.

Let b_0 be an a-block of t. This block is said to be *irreducible* if either (1) it is the last, i.e. right-most, a-block of t, or (2) writing t under the form $t = t_0 b_0 t_1 b_1 t_2$ where b_1 is the next a-block, i.e. $a \notin t_1$, one of the following holds:

- there is some i s.t. $b_0 \cap \operatorname{Im} x_i \neq \emptyset$ and $t_1 \cap \operatorname{Im} x_i \neq \emptyset$.
- there is j s.t. $b_0 \cap \operatorname{Im} y_j = \emptyset$ and $t_1 \cap \operatorname{Im} y_j \neq \emptyset$ and $b_1 \cap \operatorname{Im} y_j \neq \emptyset$.

Otherwise b_0 is said to be *reducible.*

Claim: t contains a reducible block.

Indeed, every irreducible block is either a right-most a-block for some a, or can be associated with a letter alternation in some x_i, or in some y_j. Furthermore, this association is injective. Thus there are at most $(n+m)\cdot\ell$ irreducible blocks that are not right-most (and at most $|A|$ right-most blocks). Since $k > (n+m)\cdot\ell + |A|$, t has a reducible block.

So let us pick one such reducible block, say an a-block b_0, write t under the form $t = t_0 b_0 t_1 b_1 t_2$ as above, and let $t' = t_0 t_1 b_0 b_1 t_2$.

Claim: t' fulfills the requirements of Lemma 4.4.

Since b_1 is an a-block, $b_0 b_1$ is now a block of t' and t' has less than k blocks. Moreover, the only other possible block merge is in $t_0 t_1$, thus t' has at least $k-2$ blocks. We now show that $x_i \sqsubseteq t'$ and $y_j \not\sqsubseteq t'$ for all i, j.

- Pick some i. Since $x_i \sqsubseteq t$, there is a unique decomposition $x_i = u_0 u_1 u_2 u_3 u_4$ of x_i such that $\operatorname{Im} u_0 \subseteq t_0$, $\operatorname{Im} u_1 \subseteq b_0$, $\operatorname{Im} u_2 \subseteq t_1$, $\operatorname{Im} u_3 \subseteq b_1$ and $\operatorname{Im} u_4 \subseteq t_2$. Since b_0 is reducible one of $\operatorname{Im} u_1$ or $\operatorname{Im} u_2$ is empty. Thus one of u_1 or u_2 is the empty word, allowing $x_i \sqsubseteq t'$.
- Assume, by way of contradiction, that for some j, $y_j \sqsubseteq t'$. Let z_1 be the maximal prefix of y_j that embeds into t. We proceed to show that b_0 is irreducible.
  - First, $b_0 \cap \operatorname{Im} z_1 = \emptyset$. Otherwise, since $a \notin t_1$, the left-most embedding of z_1 into $t' = t_0 t_1 b_0 b_1 t_2$ does not use t_1 at all and we would have $y_j \sqsubseteq t_0 b_0 b_1 t_2 \sqsubseteq t$.
  - Secondly, $t_1 \cap \operatorname{Im} z_1$ is not empty. If it were, since b_0 is made of a's only and $a \notin t_1$, the left-most embedding of z_1 into $t_0 t_1 b_0 b_1 t_2$ would not use t_1 and again we would have $y_j \sqsubseteq t_0 b_0 b_1 t_2 \sqsubseteq t$.
  - Lastly, $b_1 \cap \operatorname{Im} z_1 \neq \emptyset$. Otherwise, the already established fact $b_0 \cap \operatorname{Im} z_1 = \emptyset$ implies that y_j embeds not only in t' but in $t_0 t_1 t_2$, which is a subword of t.

  Since b_0 is reducible, we conclude that the original assumption that $y_j \sqsubseteq t'$ does not hold, i.e., that $y_j \not\sqsubseteq t'$ as required.

□

We now proceed to prove Theorem 4.3. Let φ be a $\Sigma_{1,1}$ sentence, where t is the only variable which is not alternation bounded. As a first step of our NP algorithm, we guess the set of literals occurring in φ that is satisfied. After guessing this subset, we check whether the formula would be satisfied if exactly this subset were true (which essentially amounts to evaluating a formula in propositional logic). If this is the case, it remains to check whether it is possible to choose words for all existential quantifiers of φ so that exactly this subset of literals is true. This means, we are left with the task of checking satisfiability of a formula φ of the form $\varphi \equiv \exists t \colon \psi$: Here, ψ begins with existential quantifiers for alternation bounded variables, which are followed by a conjunction of literals.

Every literal in ψ that involves t is of one of the following types:

(i) $x \sqsubseteq t$, where x is an alternation bounded variable,
(ii) $y \not\sqsubseteq t$, where x is an alternation bounded variable,
(iii) $t \not\sqsubseteq u$, where u is an alternation bounded variable,
(iv) $t \sqsubseteq z$, where z is an alternation bounded variable,
(v) $t \sqsubseteq t$,
(vi) $t \not\sqsubseteq t$.

Assertions of types (v) and (vi) can be replaced by their truth value. If a literal $t \sqsubseteq z$ of type (iv) occurs in ψ, then φ is equivalent to $\exists t \in (a_1^* \cdots a_n^*)^\ell \colon \psi$, where ℓ is the alternation bound of variable z.

We can thus assume that only literals of types (i) to (iii) occur in ψ. Let n be the number of variables x that occur in literals of type (i), m the number of variables y that occur in literals of type (ii), ℓ the maximum alternation level of these variables, and k the maximum alternation bound of all variables u that appear in literals of type (iii). Let

$$p = \max\{(m+n)\cdot(\ell\cdot|A|) + |A|,\ k\cdot|A|+3\}.$$

(here ℓ and k are multiplied by $|A|$ to obtain a number of blocks from a maximum alternation). Then φ is equivalent to $\exists t \in (a_1^* \cdots a_n^*)^p \colon \psi$. Indeed, if the restricted formula has a solution, it is a solution for ψ. Conversely, if ψ is satisfiable via some $t \in A^*$ having more than p blocks, then by Lemma 4.4, one can also use a t having between $k\cdot|A|$ and p blocks. The fact that t has more than $k\cdot|A|$ blocks ensures that all literals $t \not\sqsubseteq u$ are still satisfied.

Finally, we can replace every $\exists t$ in φ by a bounded quantification and obtain an equivalent formula in $\Sigma_{1,0}$ which proves Theorem 4.3

To the authors' knowledge, the following was not known.

Corollary 4.5. *If PT languages are represented as boolean combinations of sets of the form $\uparrow w$ with $w \in A^*$, then their non-emptiness problem is* NP*-complete.*

Membership in NP follows from Theorem 4.3. For hardness, we can reduce CNF-SAT as follows. We encode an assignment $\alpha\colon \{x_1,\dots,x_n\} \to \{0,1\}$ as a word $ba^{\alpha(x_1)}ba^{\alpha(x_2)}\cdots ba^{\alpha(x_n)}$. With literals x_i and $\neg x_i$, we associate the languages $K_{x_i} = \uparrow b^i a b^{n-i}$ and $K_{\neg x_i} = \{a,b\}^* \smallsetminus \uparrow b^i a b^{n-i}$. A clause $C = L_1 \vee \cdots \vee L_m$ is then translated to $K_C = K_{L_1} \cup \cdots \cup K_{L_n}$ and a conjunction of clauses $C_1 \wedge \cdots \wedge C_k$ is satisfiable if, and only if, the PT language $(b(a+\varepsilon))^n \cap K_{C_1} \cap \cdots \cap K_{C_k}$ is nonempty.

In particular, given a finite number of PT languages, the problem of deciding whether they intersect non-vacuously is NP-complete. This is in contrast with general regular languages represented by DFAs (or NFAs), for which the problem is well-known to be PSPACE-complete [28].

4.3. Complexity of $\Sigma_{1,2}$.

Our next result is an upper bound for the truth problem of $\Sigma_{1,2}$.

Theorem 4.6. *The truth problem for $\Sigma_{1,2}$ is in* NEXP.

We prove Theorem 4.6 in two steps. The first step of our decidability result is to transform a $\Sigma_{1,2}$ formula into a system of constraints where the relations among those variables without an alternation bound have a tree shape. In the second step, we exploit the tree shape to construct an exponential-size counter automaton for the set of satisfying assignments.

Tree-shaped constraints. Let $A = \{a_1,\dots,a_n\}$ and let V be a set of variables. A *constraint system* is a set of constraints of the form $x \sqsubseteq y$, $x \not\sqsubseteq y$, $x = y$, $x \in (a_1^* \cdots a_n^*)^\ell$, or $x = w$, where $x,y \in V$, $\ell \in \mathbb{N}$ and $w \in A^*$. A constraint of the form $x \sqsubseteq y$, $x \not\sqsubseteq y$, or $x = y$ is also called (x,y)*-constraint* or (y,x)*-constraint*. Constraints of the form $x \in (a_1^* \cdots a_n^*)^\ell$ are called *alternation constraints*. The set of assignments $\alpha \in (A^*)^V$ that satisfy S is denoted by $[\![S]\!]$. For a subset $U \subseteq V$, by existentially quantifying all variables outside of U, the constraint system S also defines a set of assignments in $(A^*)^U$, which we denote by $[\![S]\!]_U$.

For a constraint system S over V, we define the graph $\Gamma(S) = (V, E)$ where $\{x, y\} \in E$ if and only if S contains an (x, y)-constraint. We say that S is *tree-shaped* if $\Gamma(S)$ is a forest. Furthermore, S is called *alternation bounded* if every variable occurring in S also has an alternation constraint in S.

Proposition 4.7. *For any disjunction-free $\Sigma_{1,2}$-formula φ, one can construct polynomial-size constraint systems T and S over variables $V' \supseteq \mathsf{fv}(\varphi)$ such that*

(1) T is tree-shaped,
(2) S is alternation bounded, and
(3) $[\![T \cup S]\!]_{\mathsf{fv}(\varphi)} = [\![\varphi]\!]$.

Proof. We will need the notion of quotients of constraint systems. The idea is to identify certain pairs of variables. Suppose S is a constraint system over V. Furthermore, let $\sim \subseteq V \times V$ be an equivalence relation that specifies which variables we want to identify with each other. Then we define the *quotient* $S/{\sim}$ as a constraint system over the variable set $V/{\sim}$ with the constraints

$$\begin{aligned} S/{\sim} \ &= \{[x]\delta[y] \mid \delta \in \{\sqsubseteq, \not\sqsubseteq\},\ (x\delta y) \in S\}. \\ &\cup\ \{[x] = w \mid (x = w) \in S\} \\ &\cup\ \{[x] \in (a_1^* \cdots a_n^*)^\ell \mid (x \in (a_1^* \cdots a_n)^\ell) \in S\} \end{aligned}$$

In the course of constructing the constraint systems, it will be convenient to assume that two constraint systems are defined over disjoint sets of variables. To this end, we need some notation to state that a constraint system is equivalent to a formula even though its variables have different names. Suppose we have a constraint system S over the set of variables V' and $\psi\colon \mathsf{fv}(\varphi) \to V'$ is an injective map. Then, via ψ, the formula φ defines a set of assignments $[\![\varphi]\!]_\psi \subseteq (A^*)^{\mathsf{im}(\psi)}$. We say that φ and S are *ψ-equivalent* if $[\![S]\!]_{\mathsf{im}(\psi)} = [\![\varphi]\!]_\psi$.

We may clearly assume that all literals are of the form $x \sqsubseteq y$, $x \not\sqsubseteq y$, or $x = w$ for $w \in A^*$ (and there are no literals $w \sqsubseteq x$ etc.).

We show the following stronger statement. Let B be the set of variables in φ that are alternation bounded. For each disjunction-free $\Sigma_{1,2}$-formula φ, there is a set of variables V', constraint systems T and S over V', an injective map $\psi\colon \mathsf{fv}(\varphi) \to V'$ such that the following holds. If $B' \subseteq V'$ denotes the set of variables for which there is an alternation bound in S, then

(i) T is tree-shaped,
(ii) S is alternation-bounded,
(iii) $T \cup S$ and φ are ψ-equivalent,
(iv) for every $x \in B$ we have $\psi(x) \in B'$, and
(v) if $|\mathsf{fv}(\varphi) \setminus B| = 2$ with $\mathsf{fv}(\varphi) \setminus B = \{x, y\}$, then $\psi(x)$ and $\psi(y)$ are either neighbors in $\Gamma(T)$ or in distinct components.

To prove this statement by induction, we need to consider three cases.

(1) Literals, i.e. $x \sqsubseteq y$, $x \not\sqsubseteq y$, or $x = w$. There are only two variables so we can just take the literal as the set T and let S contain the global alternation constraints for the variables in the literal.
(2) Existentially quantified formulas $\exists x\colon \varphi$. Here, we just reduce the set of free variables, so it suffices to adjust the map ψ.
(3) Conjunctions $\varphi \equiv \varphi_0 \wedge \varphi_1$. Suppose we have constructed $T_i, S_i, V_i', \psi_i, B_i'$ as above for $i = 0, 1$. We may clearly assume $V_0 \cap V_1 = \emptyset$. We construct T, S, V', ψ, B' as follows. Let $\sim \subseteq (V_0' \cup V_1') \times (V_0' \cup V_1')$ be the smallest equivalence relation with $\psi_0(x) \sim \psi_1(x)$ for all $x \in \mathsf{fv}(\varphi) \setminus B$. Then we take $V' = (V_0' \cup V_1')/{\sim}$ and define $T = (T_0 \cup T_1)/{\sim}$. Moreover, let

$$S = S_0 \cup S_1 \cup \{[\psi_0(x)] = [\psi_1(x)] \mid x \in \mathsf{fv}(\varphi) \cap B\}.$$

Moreover, we choose $\psi\colon \mathsf{fv}(\varphi) \to V'$ so that $\psi(x) = \psi_0(x)$ if $x \in \mathsf{fv}(\varphi_0)$ and $\psi(x) = \psi_1(x)$ if $x \notin \mathsf{fv}(\varphi_0)$.

It is clear that Items (ii) to (iv) above are satisfied. It remains to verify Items (i) and (v). We distinguish the following cases.

- $|\mathsf{fv}(\varphi_i) \setminus B| \leq 1$ for some $i \in \{0,1\}$. Then there is at most one variable in V'_i that is identified with a variable in V'_{1-i} by $\sim$. Hence, $\Gamma(T)$ is obtained from $\Gamma(T_0)$ and $\Gamma(T_1)$ either by disjoint union or by identifying one vertex from $\Gamma(T_0)$ with one vertex from $\Gamma(T_1)$. In any case, $\Gamma(T)$ is a forest. Hence, Item (i) is satisfied.
  Let us now show Item (v). Hence, assume $|\mathsf{fv}(\varphi) \setminus B| = 2$ with $\mathsf{fv}(\varphi) \setminus B = \{x, y\}$.
  Clearly, if $\mathsf{fv}(\varphi_0) \setminus B$ and $\mathsf{fv}(\varphi_1) \setminus B$ are disjoint, then no variables are identified and hence $\psi(x)$ and $\psi(y)$ are in distinct components of $\Gamma(T)$. Hence, we assume that $\mathsf{fv}(\varphi_0) \setminus B$ and $\mathsf{fv}(\varphi_1) \setminus B$ have a variable in common, say x.
  Since $|\mathsf{fv}(\varphi_0) \setminus B| \leq 1$, this means $\mathsf{fv}(\varphi_1) \setminus B = \{x, y\}$ and hence $\psi_1(x)$ and $\psi_1(y)$ are neighbors in $\Gamma(T_1)$ or they are in distinct components of $\Gamma(T_1)$.
  $\Gamma(T)$ is obtained from $\Gamma(T_0)$ and $\Gamma(T_1)$ by identifying $\psi_0(x)$ and $\psi_1(x)$. Therefore, $\psi(x)$ and $\psi(y)$ are neighbors in $\Gamma(T)$ if and only if they are neighbors in $\Gamma(T_1)$. Moreover, they are in disjoint components of $\Gamma(T)$ if and only if they are in disjoint components of $\Gamma(T_1)$. This proves Item (v).
- $|\mathsf{fv}(\varphi_0)| = |\mathsf{fv}(\varphi_1)| = 2$. Write $\mathsf{fv}(\varphi_0) \setminus B = \mathsf{fv}(\varphi_1) = \{x, y\}$. Note that $\Gamma(T)$ is obtained from $\Gamma(T_0)$ and $\Gamma(T_1)$ by identifying $\psi_0(x)$ with $\psi_1(x)$ and identifying $\psi_0(y)$ with $\psi_1(y)$. Since for each $i \in \{0,1\}$ we know that $\psi_i(x)$ and $\psi_i(y)$ are either neighbors in $\Gamma(T_i)$ or in distinct components, this clearly implies that $\Gamma(T)$ is a forest. Hence, we have shown Item (i).
  Moreover, if for some $i \in \{0,1\}$, $\psi_i(x)$ and $\psi_i(y)$ are neighbors in $\Gamma(T_i)$, then $\psi(x)$ and $\psi(y)$ are neighbors in $\Gamma(T)$. Otherwise, $\psi(x)$ and $\psi(y)$ are in disjoint components of $\Gamma(T)$. This proves Item (v).

□

Counter automata. In the next step, we exploit the decomposition into a tree-shaped constraint system and an alternation-bounded constraint system to reduce satisfiability to non-emptiness of counter automata. To this end, we use a type of counter automata known as *Parikh automata* [8, 26]. In terms of expressiveness, these are equivalent to the classical reversal-bounded counter automata [20], but their syntax makes them convenient for our purposes.

Let V be a finite set of variables. A *counter automaton over* V is a tuple $\mathcal{A} = (Q, A, C, E, q_0, F)$, where Q is a finite set of *states*, A is the input alphabet, C is a set of *counters*,

$$E \subseteq Q \times (A \cup \{\varepsilon\})^V \times \mathbb{N}^C \times Q$$

is the finite set of *edges*, $q_0 \in Q$ is the *initial state*, and F is a finite set of pairs (q, φ), where $q \in Q$ and φ is an existential Presburger formula with free variables in C. A *configuration of* $\mathcal{A}$ is a tuple (q, α, μ), where $q \in Q$, $\alpha \in (A^*)^V$, $\mu \in \mathbb{N}^C$. The step relation is defined as follows. We have $(q, \alpha, \mu) \to_{\mathcal{A}} (q', \alpha', \mu')$ iff there is an edge $(q, \beta, \nu, q') \in E$ such that $\alpha' = \alpha\beta$ and $\mu' = \mu + \nu$. A counter automaton accepts a set of assignments, namely

$$L(\mathcal{A}) = \{\alpha \in (A^*)^V \mid \exists (q, \varphi) \in F\colon (q_0, \varepsilon, 0) \xrightarrow{*}_{\mathcal{A}} (q, \alpha, \mu),\ \mu \models \varphi\}\,.$$

We call a subset $R \subseteq (A^*)^V$ a *counter relation* if there is a counter automaton $\mathcal{A}$ with $R = L(\mathcal{A})$. If $|V| = 1$, say $V = \{x\}$, then $\mathcal{A}$ defines a subset of A^*, namely the language $\{w \in A^* \mid (x \mapsto w) \in L(\mathcal{A})\}$. Languages of this form are called *counter languages*.

Suppose V_0, V_1 are sets of variables with $|V_0 \cap V_1| \leq 1$. Let $\mathcal{A}_i = (Q_i, A, C_i, E_i, q_{0,i}, F_i)$ be a counter automaton over V_i for $i = 0, 1$ such that $C_0 \cap C_1 = \emptyset$. Then a simple product construction yields a counter automaton $\mathcal{A}_0 \otimes \mathcal{A}_1 = (Q_0 \times Q_1, A, C_0 \cup C_1, E, (q_{0,0}, q_{0,1}), F)$ over $V_0 \cup V_1$ such that $((p_0, p_1), \alpha, \mu) \xrightarrow{*}_{\mathcal{A}_0 \otimes \mathcal{A}_1} ((p_0', p_1'), \alpha', \mu')$ iff

$$(p_i, \alpha|_{V_i}, \mu|_{C_i}) \xrightarrow{*}_{\mathcal{A}_i} (p_i', \alpha'|_{V_i}, \mu'|_{C_i})$$

for $i = 0, 1$ and

$$F = \{((p_0, p_1), \varphi_0 \wedge \varphi_1) \mid (p_i, \varphi_i) \in F_i \text{ for } i = 0, 1\}.$$

Proposition 4.8. *Given a tree-shaped constraint system T, one can construct in exponential time a counter automaton $\mathcal{A}$ with $L(\mathcal{A}) = [\![T]\!]$.*

Proof. First, observe that it suffices to consider the case where every constraint in T involves two variables: The other constraints have the form $x = w$ for some $w \in A^*$ or $x \in (a_1^* \cdots a_n^*)^\ell$ for some $\ell \in \mathbb{N}$ and can easily be imposed afterwards in the counter automaton.

We construct the automaton inductively. The statement is trivial if T involves only one variable, so assume $|V| \geq 2$ from now on.

Since $\Gamma(T)$ is a forest, it contains a vertex $x \in V$ with at most one neighbor. Let T' be the constraint system obtained from T by removing all constraints involving x and suppose we have already constructed a counter automaton $\mathcal{A}'$ with $L(\mathcal{A}') = [\![T']\!]$.

Now if x has no neighbor, it is easy to construct the automaton for T. So suppose x has a unique neighbor y. Then, the additional constraints imposed in T are all (x, y)-constraints. Let T'' be the set of all (x, y)-constraints in T. Now note that if $\mathcal{A}''$ is a counter automaton with $L(\mathcal{A}'') = [\![T'']\!]$, then we have

$$L(\mathcal{A}' \otimes \mathcal{A}'') = [\![T' \cup T'']\!] = [\![T]\!].$$

Therefore, it suffices to construct in polynomial time a counter automaton $\mathcal{A}''$ with $L(\mathcal{A}'') = [\![T'']\!]$.

Observe that any set of (x, y)-constraints can be written as a disjunction of one of the following constraints:

(i) $x = y$ (ii) $x \sqsubset y$ (iii) $y \sqsubset x$ (iv) $x \perp y$

Since it is easy to construct a counter automaton for the union of two relations accepted by counter automata, it suffices to construct a counter automaton for the set of solutions to each of the constraints (i)—(iv). This is obvious in all cases but the last. In that last case, one can notice that $x \perp y$ holds if either (1) $|x| < |y|$ and $x \not\sqsubseteq y$ or (2) $|y| < |x|$ and $y \not\sqsubseteq x$ or (3) $|x| = |y|$ and $x \neq y$. Note that each of these cases is easily realized in a counter automaton since we can use the counters to guarantee the length constraints. Moreover, the resulting counter automaton can clearly be constructed in polynomial time, which completes the proof. □

We can now prove Theorem 4.6 by taking the constraint system provided by Proposition 4.7 and construct a counter automaton just for T using Proposition 4.8. Then, we can impose the constraints in S by using additional counters. Note that since all variables in S are alternation bounded, we can store these words, in the form of their occurring exponents, in counters. We can then install the polynomial-size Presburger formulas from Proposition 4.2 in the counter automaton to impose the binary constraints required by S. This results in an exponential size counter automaton that accepts the satisfying assignments of φ. The NEXP upper bound then follows from the fact that non-emptiness for counter automata is NP-complete.

Since counter automata are only a slight extension of reversal-bounded counter automata, the following is well-known.

Lemma 4.9. *The non-emptiness problem for counter automata is* NP*-complete.*

Proof. Given a counter automaton $\mathcal{A} = (Q, A, C, E, q_0, F)$, and a state $q \in Q$, we can construct an existential Presburger formula θ_q with a free variable for each edge in $\mathcal{A}$ that is satisfied for an assignment $\nu \in \mathbb{N}^E$ iff there is a run from q_0 to q where each edge $e \in E$

occurs exactly $\nu(e)$ times. This is just the fact that we can construct in polynomial time an existential Presburger formula for the Parikh image of a finite automaton [36].

For each edge $e = (p, \alpha, \mu, p')$, define $\mu_e = \mu$. Then the formula

$$\bigvee_{(q,\varphi)\in F} \theta_q \wedge \bigwedge_{c\in C} c = \mu_e(c)\cdot e \wedge \varphi$$

expresses precisely that there is an accepting run. The fact that the satisfiability problem for existential Presburger arithmetic is NP-complete [32] now gives us the upper bound as well as the lower bound. □

We are now ready to prove Theorem 4.6.

Proof. First we use Proposition 4.7 to turn the formula φ into constraint systems T and S such that T is tree-shaped, S is alternation bounded, and $[\![T \cup S]\!]_{\mathsf{fv}(\varphi)} = [\![\varphi]\!]$. Then we use Proposition 4.8 to to obtain in exponential time a counter automaton $\mathcal{A}$ with $L(\mathcal{A}) = [\![T]\!]$.

Let $U = \{x_1, \dots, x_m\}$ be the set of variables occurring in S. It remains to impose the constraints in S. We do this by first building the product with one automaton $\mathcal{A}_i$ for each x_i. This automaton imposes the alternation constraint on x_i and stores the word read into x_i in a set of counters. Note that this is possible because the word is alternation bounded. After taking the product with all these automata, we impose the remaining constraints from S (which are binary constraints or of the form $x = w$ with $x \in U$, $w \in A^*$) by adding existential Presburger formulas that express subword constraints as provided by Proposition 4.2.

We may clearly assume that whenever there is a variable x, an alternation constraint $x \in (a_1^* \cdots a_n^*)^\ell$, and a constraint $x = w$, then $w \in (a_1^* \cdots a_n^*)^\ell$: Otherwise, the system is not satisfiable and clearly has an equivalent counter automaton.

Let $A = \{a_1, \dots, a_n\}$. Since S is alternation bounded, S contains an alternation constraint $x_i \in (a_1^* \cdots a_n^*)^{\ell_i}$ for each $i \in [1, m]$. Let ℓ be the maximum of all these ℓ_i. We will use the counter variables $c_{i,j,k}$ for each $i \in [1, m]$, $j \in [1, \ell]$, and $k \in [1, n]$. We set up the automaton $\mathcal{A}_i$ over $V_i = \{x_i\}$ such that it has an initial state q_0, a state q_1, and satisfies

$$(q_0, \varepsilon, 0) \xrightarrow{*}_{\mathcal{A}_i} (q_1, \alpha, \mu)$$

if and only if α maps x_i to the word

$$a_1^{\mu(c_{i,1,1})} \cdots a_n^{\mu(c_{i,1,n})} \cdots a_1^{\mu(c_{i,\ell_i,1})} \cdots a_n^{\mu(c_{i,\ell_i,n})}. \tag{6}$$

This can clearly be done with $n \cdot \ell$ states. Moreover, let $F_i = \{(q_1, \top)\}$. Note that $\mathcal{A}_i$ has the counters $c_{i,j,k}$ even for $j > \ell_i$ although it never adds to them. The reason we have the variables $c_{i,j,k}$ for $j > \ell_i$ is that this way, the formulas from Proposition 4.2 are applicable.

Note that since each V_i is a singleton, the automaton $\mathcal{B} = \mathcal{A} \times \mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_m$ is defined. It satisfies $L(\mathcal{B}) = [\![S']\!]$, where S' is the set of alternation constraints in S.

It remains to impose the remaining constraints from S, namely the binary constraints and those of the form $x = w$ with $x \in U$, $w \in A^*$. Let $R \subseteq S$ be the set of these remaining constraints. For each i and for $\mu \in (A^*)^V$, let $w_{\mu,i}$ be the word in Eq. (6). According to Proposition 4.2, for each constraint $r \in R$, we can construct a polynomial size existential Presburger formula κ_r such that $\mu \models \kappa_r$ if and only if the constraint is satisfied for the assignment α with $\alpha(x_i) = w_{\mu,i}$. Moreover, let $\kappa = \bigwedge_{r\in R} \kappa_r$.

Suppose $\mathcal{B} = (Q, A, C, E, q_0, F)$. In the last step, we construct the counter automaton $\mathcal{B}' = (Q, A, C, E, q_0, F')$, where

$$F' = \{(q, \psi \wedge \kappa) \mid (q, \psi) \in F\}.$$

Now $\mathcal{B}'$ clearly satisfies $L(\mathcal{B}') = [\![T \cup S]\!]$. Thus, if we obtain $\mathcal{B}''$ from $\mathcal{B}'$ by projecting the input to those variables that occur freely in φ, then $L(\mathcal{B}'') = [\![T \cup S]\!]_{\mathsf{fv}(\varphi)}$. Moreover, $\mathcal{B}''$ can clearly be constructed in exponential time.

The membership of the truth problem in NEXP follows from the fact that emptiness of counter automata is in NP (Lemma 4.9). □

5. Expressiveness

In this section, we shed some light on which predicates or languages are definable in our fragments $\Sigma_{i,j}$.

5.1. **Expressiveness of the $\Sigma_{1,0}$ fragment.**

A language L definable in $\Sigma_{1,0}$ always satisfies $L \subseteq (a_1^* \cdots a_n^*)^\ell$ for some $\ell \in \mathbb{N}$. Hence, it can be described by the set of vectors that contain the occurring exponents. As can be derived from results in Section 4, these sets are always semilinear. In this section, we provide a decidable characterization of the semilinear sets that are expressible in this way. Stating the characterization requires some terminology.

Let V be a set of variables. By $\mathbb{N}^V$, we denote the set of mappings $V \to \mathbb{N}$. By a *partition of* V, we mean a set $P = \{V_1, \ldots, V_n\}$ of subsets $V_1, \ldots, V_n \subseteq V$ such that $V_i \cap V_j = \emptyset$ for $i \neq j$ and $V_1 \cup \cdots \cup V_n = V$. If $U \cap V = \emptyset$ and $\alpha \in \mathbb{N}^U$, $\beta \in \mathbb{N}^V$, we write $\alpha \times \beta$ for the map $\gamma \in \mathbb{N}^{U \cup V}$ such that $\gamma|_U = \alpha$ and $\gamma|_V = \beta$. Furthermore, if $S \subseteq \mathbb{N}^U$, $T \subseteq \mathbb{N}^V$, then $S \times T = \{\alpha \times \beta \mid \alpha \in S,\ \beta \in T\}$. A semilinear set $S \subseteq \mathbb{N}^V$ is *P-compatible* if it has a semilinear representation where each occurring period vector belongs to $\mathbb{N}^{V_i}$ for some $i \in [1, n]$.

Theorem 5.1. *Suppose $L \subseteq (a_1^* \cdots a_n^*)^\ell$. Let $V = \{x_{i,j} \mid i \in [1,\ell],\ j \in [1,n]\}$ and consider the partition $P = \{V_1, \ldots, V_n\}$ where $V_j = \{x_{i,j} \mid i \in [1,\ell]\}$ for $j \in [1,n]$. The language L is definable in $\Sigma_{1,0}$ if, and only if, the set*

$$\{\alpha \in \mathbb{N}^V \mid a_1^{\alpha(x_{1,1})} \cdots a_n^{\alpha(x_{1,n})} \cdots a_1^{\alpha(x_{\ell,1})} \cdots a_n^{\alpha(x_{\ell,n})} \in L\}$$

is a P-compatible semilinear set.

For example, this means we can define $\{a^n b a^n \mid n \in \mathbb{N}\}$, but not $\{a^n b^n \mid n \in \mathbb{N}\}$: A semilinear representation for the latter requires a period that produces both a's and b's.

The proof of Theorem 5.1 employs a characterization of P-compatible sets in terms of Presburger arithmetic. Let V be a set of variables and φ be a Presburger formula whose variables are in V. Let $P = \{V_1, \ldots, V_n\}$ be a partition of V. We say φ is *P-compatible* if there is a set of variables $V' \supseteq V$ and a partition $P' = \{V_1', \ldots, V_n'\}$ of V' such that

(1) $V_j \subseteq V_j'$ for each $j \in [1,n]$ and
(2) in each literal in φ, all variables belong to the same set V_j' for some $j \in [1,n]$.

The following is a simple observation.

Theorem 5.2. *Let $P = \{V_1, \ldots, V_n\}$ be a partition of V. For sets $S \subseteq \mathbb{N}^V$, the following conditions are equivalent:*

(1) S is a P-compatible semilinear set.
(2) $S = [\![\varphi]\!]$ for some P-compatible existential Presburger formula φ.
(3) S is a finite union of sets of the form $A_1 \times \cdots \times A_n$ where each A_j is a semilinear subset of $\mathbb{N}^{V_j}$.

Proof. The directions "3⇒1" and "1⇒2" are easy to see, so we show "2⇒3".

If a set satisfies the condition of 3, then projecting to a subset of the coordinates yields again a set of this form. Therefore, it suffices to consider the case where in φ, there are no quantifiers. Now, bring φ into disjunctive normal form. Since each literal in φ only mentions variables from V_j for some $j \in [1,n]$, we can sort the literals of each co-clause of the DNF according to the subset V_j they mention. Hence, we arrive at the form

$$\varphi \equiv \bigvee_{i=1}^{k} \bigwedge_{j=1}^{n} \varphi_{i,j},$$

where $\varphi_{i,j}$ only mentions variables from V_j. This implies

$$[\![\varphi]\!] = \bigcup_{i=1}^{k} [\![\varphi_{1,j}]\!] \times \cdots \times [\![\varphi_{n,j}]\!],$$

which is the form required in 3. ☐

We are now ready to prove Theorem 5.1.

Proof. If L is definable in $\Sigma_{1,0}$, we can write down a Presburger formula that defines S. Here, in order to express the subword ordering (and its negation), we use the formulas from Proposition 4.2. Observe that these formulas are P-compatible. This means that S is P-compatible.

For the converse, suppose S is P-compatible. According to Theorem 5.21, S is defined by a P-compatible existential Presburger formula φ. Hence, φ has free variables $V = \{x_{i,j} \mid i \in [1,\ell], j \in [1,n]\}$ and uses variables $V' \supseteq V$ that are partitioned as $V' = \biguplus_{j=1}^{n} V'_j$ so that in each literal, all occurring variables belong to the same V'_j.

In the first step, we turn φ into a $\Sigma_{1,0}$ formula $\bar{\varphi}$ with the same number of free variables. For each $x \in V'$, we take a fresh variable $\bar{x}$, which will hold words in a_j^*. More precisely, we have our new variables $\bar{V} = \{\bar{x} \mid x \in V'\}$ and a mapping $\iota\colon \mathbb{N}^{V'} \to (A^*)^{\bar{V}}$ defined by $\iota(\alpha)(\bar{x}) = a_j^{\alpha(x)}$, where j is the unique index with $x \in V'_j$. We want to construct $\bar{\varphi}$ so that $[\![\bar{\varphi}]\!] = \iota([\![\varphi]\!])$.

We obtain $\bar{\varphi}$ from φ as follows. For each literal $x = y + z$, we know that there is a $j \in [1,n]$ with $x, y, z \in V'_i$, so we can replace the literal with $|\bar{x}|_{a_j} = |\bar{y}|_{a_j} + |\bar{z}|_{a_j}$, which is expressible in $\Sigma_{1,0}$ according to Item 11 in the proof of Theorem 3.3 (note that in this case, we actually are in $\Sigma_{1,0}$ because the variables $\bar{x}$, $\bar{y}$, and $\bar{z}$ range over a_j^* and are thus alternation bounded). Since we can clearly also express $|\bar{x}|_{a_j} \neq |\bar{y}|_{a_j}$ in $\Sigma_{1,0}$, we use this to implement literals $x \neq y$ with $x, y \in V'_j$. Literals of the form $x = k$ with $k \in \mathbb{N}$ and $x \in V'_j$ can just be replaced by $\bar{x} = a_j^k$. Then we clearly have $[\![\bar{\varphi}]\!] = \iota([\![\varphi]\!])$.

In the second step, we construct the words

$$a_1^{\alpha(x_{1,1})} \cdots a_n^{\alpha(x_{1,n})} \cdots a_1^{\alpha(x_{\ell,1})} \cdots a_n^{\alpha(x_{\ell,n})}$$

for $\alpha \in [\![\varphi]\!]$. This is possible thanks to Item 13 of the proof of Theorem 3.3. We can express

$$u = \bar{x}_{1,1} \cdots \bar{x}_{1,n} \cdots \bar{x}_{\ell,1} \cdots \bar{x}_{\ell,n}$$

by applying Item 13 exactly $\ell \cdot n - 1$ times, once to append each $x_{i,j}$ to the word defined so far, using $\ell \cdot n - 1$ additional variables. Of course, all these variables can be restricted to $(a_1^* \cdots a_n^*)^\ell$, which means the resulting formula belongs to $\Sigma_{1,0}$. Moreover, it clearly defines L. ☐

Our characterization of $\Sigma_{1,0}$ is decidable. We use a technique from [14], where it is shown that recognizability is decidable for semilinear sets. The idea is to characterize P-compatibility as the finiteness of the index of certain equivalence relations, which can be expressed in Presburger arithmetic.

Theorem 5.3. *Given a semilinear subset $S \subseteq \mathbb{N}^V$ and a partition P of V, it is decidable whether S is P-compatible.*

Proof. For $\alpha \in \mathbb{N}^{V_i}$ and $\gamma \in \mathbb{N}^V$, we write $\gamma[i/\alpha]$ to be the element of $\mathbb{N}^V$ with

$$\gamma[i/\alpha](v) = \begin{cases} \alpha(v) & \text{if } v \in V_i, \\ \gamma(v) & \text{otherwise.} \end{cases}$$

For $\alpha, \beta \in \mathbb{N}^{V_i}$, we write $\alpha \sim_i \beta$ if for every $\gamma \in \mathbb{N}^V$, we have $\gamma[i/\alpha] \in S$ if and only if $\gamma[i/\beta] \in S$. Moreover, for $\gamma \in \mathbb{N}^V$, we will use the norm $\|\cdot\|$ as defined by $\|\gamma\| = \sum_{v \in V} \gamma(v)$. We claim that S is P-compatible if and only if

$$\exists k \in \mathbb{N}\colon \bigwedge_{i=1}^{n} \forall \alpha \in \mathbb{N}^{V_i} \colon \exists \beta \in \mathbb{N}^{V_i} \colon \|\beta\| \leq k,\ \alpha \sim_i \beta. \qquad (7)$$

Suppose Eq. (7) holds. For each $\beta \in \mathbb{N}^{V_i}$, we define $S_{i,\beta} = \{\alpha \in \mathbb{N}^{V_i} \mid \alpha \sim_i \beta\}$. Then $S_{i,\beta} \subseteq \mathbb{N}^{V_i}$ is semilinear and we have

$$S = \bigcup_{\beta_1 \in \mathbb{N}^{V_1}, \|\beta_1\| \le k} \cdots \bigcup_{\beta_n \in \mathbb{N}^{V_n}, \|\beta_n\| \le k} S_{1,\beta_1} \times \cdots \times S_{n,\beta_n}.$$

Hence, S is P-compatible.

Now assume S is P-compatible. Then we can write $S = \bigcup_{j=1}^{\ell} A_{j,1} \times \cdots \times A_{j,n}$, where each $A_{j,i} \subseteq \mathbb{N}^{V_i}$ is semilinear. For each $i \in [1,n]$, consider the function $\kappa_i \colon \mathbb{N}^{V_i} \to 2^{\{1,\ldots,\ell\}}$ with

$$\kappa_i(\alpha) = \{j \in [1,\ell] \mid \alpha \in A_{j,i}\}.$$

Observe that if $\kappa_i(\alpha) = \kappa_i(\beta)$, then $\alpha \sim_i \beta$. Since κ_i has a finite codomain, this means the equivalence relation $\sim_i$ on $\mathbb{N}^{V_i}$ has finite index. This immediately implies Eq. (7).

Since we can clearly formulate the condition Eq. (7) in Presburger arithmetic, P-compatibility is decidable. □

In fact, it is not hard to see that if P consists only of singletons, a semilinear set is P-compatible iff it is recognizable. Hence, Theorem 5.3 generalizes the decidability of recognizability. Let M be a monoid. A subset $S \subseteq M$ is called *recognizable* if there is a finite monoid F and a morphism $\varphi\colon M \to F$ such that $S = \varphi^{-1}(\varphi(S))$.

Theorem 5.4. *Suppose P consists only of singletons. Then $S \subseteq \mathbb{N}^V$ is P-compatible if and only if it is recognizable.*

Proof. Mezei's Theorem [4] states that if $M_1, \ldots, M_n$ are monoids, then a subset of $M_1 \times \cdots \times M_n$ is recognizable if and only if it is a finite union of sets $S_1 \times \cdots \times S_n$ such that $S_i \subseteq M_i$ is recognizable for $i \in \{1, \ldots, n\}$.

Combined with the fact that a subset of $\mathbb{N}$ is semilinear if and only if it is recognizable, the condition 3 of Theorem 5.2 yields the result. □

5.2. Expressiveness of $\Sigma_{1,0}$ vs. $\Sigma_{1,1}$.

It is obvious that $\Sigma_{1,1}$ is strictly more expressive than $\Sigma_{1,0}$, because it permits the definition of languages with unbounded alternations, such as $\{a,b\}^*$. But is this the only difference between the two fragments? In other words: Restricted to alternation bounded languages, is $\Sigma_{1,1}$ more expressive? The answer is no.

Theorem 5.5. *If $L \subseteq (a_1^* \cdots a_n^*)^\ell$, then L is definable in $\Sigma_{1,1}$ if and only if it is definable in $\Sigma_{1,0}$.*

Proof. Let φ be a $\Sigma_{1,1}$ formula where the free variable is alternation bounded and the variable t is not alternation bounded. We can transform φ into a disjunction $\bigvee_{i=1}^k \varphi_i$, where each φ_i belongs to $\Sigma_{1,1}$ and consists of a block of existential quantifiers followed by a conjunction of literals. Then, the proof of Theorem 4.3 yields for each φ_i a (polynomial) bound p_i so that if we replace the quantifier $\exists t$ in φ_i by $\exists t \in (a_1^* \cdots a_n^*)^{p_i}$, the resulting $\Sigma_{1,0}$ formula is equivalent. □

This allows us to reason beyond alternation bounded languages. We have seen in the proof of Theorem 3.3 that one can express "$|u|_a = |v|_b$" in $\Sigma_{1,3}$, which required significantly more steps, and two more alternation unbounded variables, than the ostensibly similar "$|u|_a = |v|_a$". This raises the question: Can we define the former in $\Sigma'_{1,1}$? We cannot:

Corollary 5.6. *The predicate "$|u|_a = |u|_b$" is not definable in $\Sigma_{1,1}$.*

Proof. Otherwise, we could define the set $\{a^n b^n \mid n \ge 0\}$ in $\Sigma_{1,1}$, hence in $\Sigma_{1,0}$, contradicting Theorem 5.1. □

5.3. **Expressiveness of $\Sigma_{1,2}$ vs. $\Sigma_{1,3}$.** We have seen in Theorem 3.3 that $\Sigma_{1,3}$ can express all recursively enumerable unary languages. Moreover, Theorem 4.6 tells us that the languages definable in $\Sigma_{1,2}$ are always counter languages.

How do the fragments compare with respect to natural (binary) predicates on words. We already know from Item 14 in Theorem 3.3 that over two letters, the prefix relation is expressible in $\Sigma_{1,3}$. Note that the following theorem does not follow directly from the fact that for any $\Sigma_{1,2}$ formula φ, the set $[\![\varphi]\!]$ is a counter relation, as shown in Section 4. This is because the prefix relation is a counter relation (and even rational).

Theorem 5.7. *In $\Sigma_{1,2}$, "u is a prefix of v" is not expressible.*

Proof. Suppose the prefix relation were expressible using a $\Sigma_{1,2}$ formula φ. Then, by reversing all constants in φ, we obtain a formula expressing the suffix relation. Let $\sqsubseteq_{\mathsf{p}}$ denote the prefix relation and $\sqsubseteq_{\mathsf{s}}$ the suffix relation. We can now express

$$\exists v \in \{a,b\}^* \colon v \sqsubseteq_{\mathsf{p}} u \wedge v \sqsubseteq_{\mathsf{s}} u \wedge |u|_a = 2 \cdot |v|_a \wedge |u|_b = 2 \cdot |v|_b,$$

which is equivalent to $u \in S$, where $S = \{vv \mid v \in \{a,b\}^*\}$. Note that $|u|_a = 2 \cdot |v|_a$ can be expressed by using $|u|_a = |v|_a + \cdot |v|_a$, which can be done in $\Sigma'_{1,0}$ according to Item 11 in Theorem 3.3.

However, S is not a counter language. This is due to the fact that the class of recursively enumerable languages is the smallest language class that contains S, is closed under rational transductions, union, and intersection (this can be shown as in the case of the set of palindromes [3]). However, the class of counter languages also has these closure properties and is properly contained in the recursively enumerable languages. Hence, S is indeed not a counter language. This is in contradiction with the fact that for any $\Sigma_{1,2}$ formula φ, the set $[\![\varphi]\!]$ is a counter relation, as shown in Section 4. □

6. Conclusion

We have shown that the Σ_1 theory of the subword ordering is undecidable (already for two letters), if all words are available as constants. This implies that the Σ_2 theory is undecidable already for two letters, even without constants.

In order to shed light on decidable fragments of first-order logic over the structure $(A^*, \sqsubseteq, w_1, \ldots)$, we introduced the fragments $\Sigma_{i,j}$. We have completely settled their decidability status. In terms of complexity, the only open case is the $\Sigma_{1,2}$ fragment. We have an NP lower bound and an NEXP upper bound.

This aligns with the situation for expressiveness. We have a decidable characterization for the expressiveness of $\Sigma_{1,0}$ and, obvious exceptions aside, $\Sigma_{1,1}$ is as expressive as $\Sigma_{1,0}$. However, we do not know whether $\Sigma_{1,1}$ and $\Sigma_{1,2}$ differ significantly: Of course, $\Sigma_{1,2}$ can have two alternation unbounded free variables, but it is conceivable that $\Sigma_{1,1}$ and $\Sigma_{1,2}$ define the same languages.

References

[1] P. A. Abdulla, M. F. Atig, Y. Chen, L. Holík, A. Rezine, P. Rümmer, and J. Stenman. "String Constraints for Verification". In: *Proc. CAV 2014, LNCS 8559.* Springer, 2014, pp. 150–166.

[2] P. A. Abdulla, A. Collomb-Annichini, A. Bouajjani, and B. Jonsson. "Using Forward Reachability Analysis for Verification of Lossy Channel Systems". In: *Form. Methods Sys. Des.* 25.1 (2004), pp. 39–65.

[3] B. S. Baker and R. V. Book. "Reversal-bounded multipushdown machines". In: *J. Comput. System Sci.* 8.3 (1974), pp. 315–332.

[4] J. Berstel. *Transductions and Context-Free Languages.* Stuttgart: B. G. Teubner, 1979.

[5] A. Boudet and H. Comon. "About the Theory of Tree Embedding". In: *Proc. TAPSOFT '93, LNCS 668.* Springer, 1993, pp. 376–390.

[6] P. Bouyer, N. Markey, J. Ouaknine, Ph. Schnoebelen, and J. Worrell. “On Termination and Invariance for Faulty Channel Machines”. In: *Form. Asp. Comp.* 24.4–6 (2012), pp. 595–607.

[7] J. R. Büchi and S. Senger. “Definability in the Existential Theory of Concatenation and Undecidable Extensions of this Theory”. In: *Z. Math. Logik Grundlag. Math.* 34.4 (1988), pp. 337–342.

[8] M. Cadilhac, A. Finkel, and P. McKenzie. “Affine Parikh automata”. In: *RAIRO Theor. Inf. Appl.* 46.4 (2012), pp. 511–545.

[9] H. Comon and R. Treinen. “Ordering Constraints on Trees”. In: *Proc. CAAP ’94, LNCS 787.* Springer, 1994, pp. 1–14.

[10] V. Diekert, P. Gastin, and M. Kufleitner. “A Survey on Small Fragments of First-Order Logic over Finite Words”. In: *Int. J. Found. Comput. Sci.* 19.3 (2008), pp. 513–548.

[11] V. G. Durnev. “Undecidability of the positive $\forall\exists^3$-theory of a free semigroup”. In: *Sib. Math. J.* 36.5 (1995), pp. 917–929.

[12] J. Ferrante and C. W. Rackoff. *The computational complexity of logical theories.* Vol. 718. Lecture Notes in Mathematics. Springer, 1979.

[13] V. Ganesh, M. Minnes, A. Solar-Lezama, and M. C. Rinard. “Word Equations with Length Constraints: What’s Decidable?” In: *Proc. HVC 2012, LNCS 7857.* Springer, 2013, pp. 209–226.

[14] S. Ginsburg and E. H. Spanier. “Bounded Regular Sets”. In: *Proc. Amer. Math. Soc.* 17.5 (1966), pp. 1043–1049.

[15] G. Gottlob, N. Leone, and H. Veith. “Second Order Logic and the Weak Exponential Hierarchies”. In: *Proc. MFCS ’95, LNCS 969.* Springer, 1995, pp. 66–81.

[16] Ch. Haase, S. Schmitz, and Ph. Schnoebelen. “The Power of Priority Channel Systems”. In: *Log. Methods Comput. Sci.* 10.4:4 (2014).

[17] C. Haase. “Subclasses of Presburger Arithmetic and the Weak EXP Hierarchy”. In: *Proc. CSL-LICS 2014.* ACM, 2014, 47:1–47:10.

[18] L. Hemachandra. “The strong exponential hierarchy collapses”. In: *J. Comput. System Sci.* 39.3 (1989), pp. 299–322.

[19] P. Hooimeijer and W. Weimer. “StrSolve: solving string constraints lazily”. In: *Autom. Softw. Eng.* 19.4 (2012), pp. 531–559.

[20] O. H. Ibarra. “Reversal-bounded multicounter machines and their decision problems”. In: *J. ACM* 25.1 (1978), pp. 116–133.

[21] A. Jeż. “Recompression: A Simple and Powerful Technique for Word Equations”. In: *J. ACM* 63.1 (2016), 4:1–4:51.

[22] P. Karandikar, M. Niewerth, and Ph. Schnoebelen. “On the state complexity of closures and interiors of regular languages with subwords and superwords”. In: *Theoret. Comput. Sci.* 610 (2016), pp. 91–107.

[23] P. Karandikar and Ph. Schnoebelen. “Decidability in the Logic of Subsequences and Supersequences”. In: *Proc. FST&TCS 2015, LIPIcs 45.* Leibniz-Zentrum für Informatik, 2015, pp. 84–97.

[24] P. Karandikar and Ph. Schnoebelen. “Generalized Post Embedding Problems”. In: *Theory of Computing Systems* 56.4 (2015), pp. 697–716.

[25] P. Karandikar and Ph. Schnoebelen. “The height of piecewise-testable languages with applications in logical complexity”. In: *Proc. CSL 2016, LIPIcs 62.* Leibniz-Zentrum für Informatik, 2016, 37:1–37:22.

[26] F. Klaedtke and H. Rueß. “Monadic Second-Order Logics with Cardinalities”. In: *Proc. ICALP 2003, LNCS 2719.* Springer, 2003, pp. 681–696.

[27] O. Klíma and L. Polák. “Alternative Automata Characterization of Piecewise Testable Languages”. In: *Proc. DLT 2013, LNCS 7907.* Springer, 2013, pp. 289–300.

[28] D. C. Kozen. “Lower bounds for natural proof systems”. In: *Proc. FOCS ’77.* IEEE, 1977, pp. 254–266.

[29] O. V. Kudinov, V. L. Selivanov, and L. V. Yartseva. “Definability in the Subword Order”. In: *Proc. CiE 2010, LNCS 6158.* Springer, 2010, pp. 246–255.

[30] D. Kuske. “Theories of orders on the set of words”. In: *RAIRO Theor. Inf. Appl.* 40.1 (2006), pp. 53–74.

[31] Y. V. Matiyasevich. *Hilbert’s Tenth Problem.* Cambridge, Massachusetts: MIT Press, 1993.

[32] D. C. Oppen. “A $2^{2^{2^{pn}}}$ upper bound on the complexity of Presburger Arithmetic”. In: *J. Comput. System Sci.* 16.3 (1978), pp. 323–332.

[33] W. Plandowski. “An efficient algorithm for solving word equations”. In: *Proc. STOC 2006.* ACM Press, 2006, pp. 467–476.

[34] W. V. Quine. “Concatenation as a basis for arithmetic”. In: *J. Symb. Logic* 11.4 (Dec. 1946), pp. 105–114. ISSN: 1943–5886.

[35] L. J. Stockmeyer. “The complexity of decision problems in automata theory and logic”. Available as Report MAC-TR-133. PhD thesis. Department of Electical Engineering, MIT, July 1974.

[36] K. N. Verma, H. Seidl, and T. Schwentick. “On the Complexity of Equational Horn Clauses”. In: *Proc. CADE 2005, LNCS 3632.* Springer, 2005, pp. 337–352.

Lab. Specification & Verification (LSV), CNRS & ENS Paris-Saclay, Cachan, France
Email address: `halfon@lsv.fr`

Lab. Specification & Verification (LSV), CNRS & ENS Paris-Saclay, Cachan, France
Email address: `phs@lsv.fr`

Lab. Specification & Verification (LSV), CNRS & ENS Paris-Saclay, Cachan, France
Email address: `zetzsche@lsv.fr`