Deciding atomicity of subword-closed languages

A. Atminas[1] and V. Lozin[2]

[1] Department of Mathematical Sciences, Xi'an Jiaotong-Liverpool University, 111 Ren'ai Road, Suzhou 215123, China
Aistis.Atminas@xjtlu.edu.cn
[2] Mathematics Institute, University of Warwick, Coventry, CV4 7AL, UK
V.Lozin@warwick.ac.uk

Abstract. We study languages closed under non-contiguous (scattered) subword containment order. Any subword-closed language L can be uniquely described by its anti-dictionary, i.e. the set of minimal words that do not belong to L. A language L is said to be *atomic* if it cannot be presented as the union of two subword-closed languages different from L. In this work, we provide a decision procedure which, given a language over a finite alphabet defined by its anti-dictionary, decides whether it is atomic or not.

Keywords: Subword-closed language · Joint embedding property · Decidability

1 Introduction

Throughout this paper, A is a finite alphabet and A^* is the set of all finite words over A. A word α is a *subword* of a word β if α can be obtained from β by erasing some (possibly none) letters. We say that a language L is *subword-closed* if $\beta \in L$ implies $\alpha \in L$ for every subword α of β. According to the celebrated Higman's lemma [6], the subword order is a well-quasi-order, and hence every subword-closed language L over a finite alphabet can be uniquely described by a *finite* set of minimal words not in L, called the *anti-dictionary* of L. We will denote the language defined by an anti-dictionary D by $Free(D)$ and call the words in D the *minimal forbidden words* for L.

A subword-closed language L is said to be *atomic* if L cannot be expressed as the union of two non-empty subword-closed languages different from L. It is well-known that atomicity is equivalent to the *joint embedding property* (JEP), which, in case of languages, can be defined as follows: for any two words $\alpha \in L$ and $\beta \in L$ there is a word $\gamma \in L$ containing α and β as subwords. Atomicity, or JEP, is a fundamental property, which frequently appears in the study of various combinatorial structures, for instance, growth rates of permutation classes [11] or hereditary classes of graphs, which are critical with respect to some parameters [1].

The main problem we study in this paper is deciding whether a subword-closed language given by its anti-dictionary is atomic or not. Decidability of

atomicity, or of JEP, is a question, which was addressed in various contexts. In particular, in [3] Braunfeld has shown that this question is undecidable for hereditary classes of graphs defined by finitely many forbidden induced subgraphs. One more undecidability result appeared in [2], where Bodirsky et al. have shown that the joint embedding property is undecidable for the class of all finite models of a given universal Horn sentence. On the other hand, several positive results have been obtained by McDevitt and Ruškuc in [9], where the authors studied classes of words and permutations closed under taking consecutive subwords, also known as factors, and consecutive subpermutations. In both cases, atomicity of classes of words or permutations defined by finitely many forbidden factors or consecutive subpermutations has been shown to be decidable. We observe that every subword-closed language is also factor-closed. However, for languages defined by finitely many forbidden factors or subwords the two families are incomparable. There are languages defined by finitely many forbidden factors that are not subword-closed, and there are subword-closed languages that are not defined by *finitely many* forbidden factors. For instance, for the subword-closed language $Free(101)$ the set of minimal forbidden factors is infinite and contains all words of the form $10\dots01$.

The main result of this paper, proved in Section 2, states that atomicity of subword-closed languages is decidable. We discuss possible applications of this result in Section 3.

2 Main result

We start with some notational remarks. For a word $w \in A^*$, we denote by $|w|$ the number of letters in the word. Also, to simplify the notation $Free(D)$ we omit curly brackets when listing the elements of D. The main result is the following.

Theorem 1. *Let $L = Free(w_1, w_2, \dots, w_n)$ be a language. It is algorithmically decidable whether L is atomic or not. In particular, there exists a decision procedure of complexity $O(n \times m^2)$ where $m = |w_1| + |w_2| + \ldots + |w_n|$.*

The proof of this theorem will be given by induction on $m = |w_1| + |w_2| + \ldots + |w_n|$, i.e. on the total number of letters in the forbidden words. If any of the forbidden words consists of a single letter, then we claim that we can remove this word from the anti-dictionary without changing atomicity, which is proved in the following lemma.

Lemma 1. *Let $L = Free(w_1, w_2, \dots, w_n)$ be a language. If $|w_i| = 1$ for some i, then L is atomic if and only if $L' = Free(w_1, w_2, \dots\ w_{i-1}, w_{i+1}, w_{i+2} \dots, w_n)$ is atomic.*

Proof. Suppose w_i is the word consisting of only one letter $a \in A$. As the set of words defining the language is assumed to be minimal, we can see that letter a does not appear in any of the words w_j with $j \neq i$. Suppose first that L is not atomic, i.e. $L = L_1 \cup L_2$ for some non-empty languages $L_1 \neq L$ and

$L_2 \neq L$. Then clearly, L_1 and L_2 do not contain letter a, so they can be written as $Free(a, x_1, x_2 \ldots, x_k)$ and $Free(a, y_1, y_2, \ldots, y_l)$ for some words x_i and y_i not containing letter a. But then $L' = Free(x_1, x_2, \ldots, x_k) \cup Free(y_1, y_2, \ldots, y_l)$, and hence L' is not atomic either. On the other hand, suppose that L is atomic. Pick any two words $x', y' \in L'$. Let the words x and y be the subwords of x' and y' obtained by deleting all letters a in x and y, respectively. Then $x, y \in L$ and since L is atomic, by JEP there exists $z \in L$ such that z contains x and y. By adding $|x|+|y|$ copies of letter a between any two consecutive letters of z as well as in the prefix and suffix of z, we obtain a new word $z' \in L'$, which contains x' and y'. Hence L' is atomic as well. This finishes the proof. □

Let $W = \{w_1, w_2, \ldots, w_n\}$ be a set of incomparable words over A each of which has at least two letters, and let $L = Free(w_1, w_2, \ldots, w_n)$ be the language defined by forbidding the words in W. For each $i \in \{1, 2, \ldots, n\}$, we denote by w_{i1} the first letter of w_i and by $w'_i \in A^*$ the word obtained from w_i by removing w_{i1}, i.e. $w_i = w_{i1}w'_i$. Let

$A' = \{w_{i1} : i = 1, 2, \ldots, n\}$ be the set of the first letters appearing in the words $w_1, w_2, \ldots, w_n$.

We call the letters in A' the *leading letters*. Also, we will say that a word $w \in L$ is *leader-free* if it contains no leading letters, and that w is an *a-word* if $a \in A'$ is the first (when reading from left to right) leading letter in w. For each letter $a \in A'$, we denote by

$I_a = \{i \in \mathbb{N} : w_{i1} = a\}$ the set of indices of the words in W that start with letter a,
$S_a = \{w_i : i \in I_a\}$ the subset of words from W that start with letter a,
$S'_a = \{w'_i : i \in I_a\}$ the set of words obtained from the words in S_a by removing the first letter a,
$W_a = \{w'_i : i \in I_a\} \cup \{w_i : i \notin I_a\}$ the set of words obtained from the words in W by removing the first appearance of letter a from all the words that start with a,
$L_a = \{pws : p \in Free(A'), w \in \{a, \emptyset\}, s \in Free(W_a)\}$. Informally, L_a is the subword closure of the set of a-words in L. We observe that all leader-free words from L belong to L_a.

Clearly, each L_a is a subword-closed language and $L = Free(w_1, w_2, \ldots, w_n) = \cup_{a \in A'} L_a$.

Lemma 2. *L is atomic only if $L = L_a$ for some $a \in A'$.*

Proof. Assume that for each $a \in A'$ the language L_a is a proper sublanguage of L. Then take the minimal set $A'' \subseteq A'$ such that $\cup_{a \in A''} L_a = L$. Such a set exists as $\cup_{a \in A'} L_a = L$ and has size $|A''| \geq 2$ as each L_a is a proper sublanguage of L. Fixing any $b \in A''$ we obtain two proper sublanguages L_b and $\cup_{a \in A'' \setminus \{b\}} L_a$ of L whose union is L. So L is not atomic. Hence, L can be atomic only if for some $a \in A'$ we have $L = L_a$. □

To be able to determine whether $L_a = L$ we will determine the list of minimal forbidden subwords for the language L_a. For that purpose, let us define a simple binary relation $\circ : A \times A^* \to A^*$ as follows: for any letter $a \in A$ and any word $w \in A^*$ we define

$$a \circ w = \begin{cases} w, & \text{if } w \text{ starts with letter } a, \\ aw, & \text{otherwise.} \end{cases}$$

Given a letter $b \in A'$, we define $S_a^b = \{b \circ w_i' : i \in I_a\}$ to be the set of words obtained from the words in S_a' by adding letter b in front of all words that do not start with b.

Lemma 3. $L_a = Free(W \cup_{b \in A' \setminus \{a\}} S_a^b)$.

Proof. We denote $L' = Free(W \cup_{b \in A' \setminus \{a\}} S_a^b)$ and show first that L_a is a subset of L', i.e. we show that every word which is forbidden for L' is also forbidden for L_a. Since L_a is a subset of L, every word from W is forbidden for L_a. Now let $b \in A' \setminus \{a\}$ and assume, to the contrary, that a word $bw \in S_a^b$ belongs to L_a. Then, by definition, bw is contained in an a-word $w' \in L_a$. But then w' contains abw as a subword, which is impossible, because aw (if $w \in S_a'$) or abw (if $bw \in S_a'$) belongs to W and hence is forbidden for words in L_a. This contradiction proves that $L_a \subseteq L'$.

Conversely, consider a word $w \in L'$. Clearly, w belongs to L, since $L' \subseteq L$. If w is an a-word or leader-free, then it also belongs to L_a. Suppose w is a b-word for a letter $b \in A' \setminus \{a\}$. Then by inserting an a right before the leading b in w we obtain a word w', which still belongs to L, since otherwise a forbidden word from S_a^b can be found in w. Therefore, w' and hence w belong to L_a, proving that $L' \subseteq L_a$. □

By the lemma above, to check whether $L_a = L$ we only need to check whether each element of $\cup_{b \in A' \setminus \{a\}} S_a^b$ contains some of the words $w_1, w_2, \ldots, w_n$. If there is an element $w \in \cup_{b \in A' \setminus \{a\}} S_a^b$ which does not contain any of the words $w_1, \ldots, w_n$, then we can readily conclude that $L_a \neq L$, because in this case $w \in L$ and $w \notin L_a$. The result below describes a procedure which makes the checking efficient.

Lemma 4. *For every word $w \in S_a'$ perform the following procedure:*

1. *If the first letter of w is in $A' \setminus \{a\}$ then stop, $L_a \neq L$.*
2. *Otherwise, for every letter $b \in A' \setminus \{a\}$ do the following:*
   - *Check whether there exists a word $v \in S_b'$ contained in w. If yes, proceed to the next b, if no then stop, $L_a \neq L$.*

If the algorithm has successfully run through all the words $w \in S_a'$ and did not stop, then $L_a = L$. The algorithm has running time $O(|S_a'|nm)$ where $m = |w_1| + |w_2| + \ldots + |w_n|$.

Proof. Consider any word in $w \in S_a'$. If the first letter of w is b, for some $b \in A' \setminus \{a\}$, then $b \circ w = w$ and by definition of S_a^b it follows that $w \in S_a^b \subseteq$

$\cup_{b\in A'\setminus\{a\}} S_a^b$. As $w = w'_i \in S'_a$ is a proper subword of some word $w_i \in S_a$ and $w_1, w_2, \ldots, w_n$ are incomparable, w cannot contain any word w_j with $j \neq i$. Therefore,

$$L_a \subseteq Free(w_1, w_2, \ldots, w_{i-1}, w'_i, w_{i+1}, \ldots, w_n) \neq L.$$

Next, consider the case when the first letter of w is not in $A'\setminus\{a\}$. Pick any $b \in A'\setminus\{a\}$. Then $b \circ w = bw$. Again, as $bw \in S_a^b$, $L \neq L_a$, unless bw contains some element of $\{w_1, w_2, \ldots, w_n\}$. Clearly bw cannot contain a word $w_j \in \{w_1, w_2, \ldots, w_n\}\setminus S_b$, since otherwise $w = w'_i$ contains w_j, which is a contradiction to the fact that w_i and w_j are incomparable for $i \neq j$. Therefore, $L = L_a$ only if bw contains a word $w_j \in S_b$, i.e. only if w contains a word $v = w'_j \in S'_b$. Note that this has to hold for each $b \in A'\setminus\{a\}$, since otherwise we obtain a word in S_a^b that does not contain any of $w_1, w_2, \ldots, w_n$, in which case L_a is a proper sublanguage of L.

Finally, note that if the procedure runs through all the words $w \in S'_a$ without deducing that $L \neq L_a$, then every word in S'_a starts with a letter in $A\setminus A' \cup \{a\}$, implying that for each letter $b \in A'\setminus\{a\}$, every word in the set $S_a^b = \{b \circ w : w \in S'_a\} = \{bw : w \in S'_a\}$ contains some word from the set $\{w_1, w_2, \ldots, w_n\}$. This means that none of the words in $\cup_{b\in A'\setminus\{a\}} S_a^b$ is minimal and hence

$$L_a = Free(\{w_1, w_2, \ldots, w_n\} \cup_{b\in A'\setminus\{a\}} S_a^b) = Free(\{w_1, w_2, \ldots, w_n\}) = L.$$

The main step of algorithm is checking whether a word $w \in S'_a$ contains a word from the set $\{w'_1, w'_2, \ldots, w'_n\}$. To check whether w contains w'_i, one can go through the letters of w until the first appearance of the first letter of w'_i in w is found, then proceed to the first appearance of the second letter of w'_i in w and so on. It takes $O(|w|)$ steps to check whether w contains w'_i, and it is performed for at most n different words $w'_i s$. Hence for each $w \in S'_a$ it takes $O(|w|n)$ steps and hence in total it takes $O(|S'_a||w|n)$ steps. Noting that $|w| \leq m$, completes the proof of the lemma. □

By Lemma 2, L is atomic only if $L = L_a$ for some $a \in A'$. Rather than checking whether $L = L_a$ for each $a \in A'$, one can, in fact, quickly determine one specific letter $a \in A'$ for which it suffices to verify whether $L = L_a$. In the lemma below, for two vectors of integers $v = (v_1, \ldots, v_n)$ and $u = (u_1, \ldots, u_m)$ we say that v *majorizes* u if either $n \leq m$ and $v_i = u_i$ for all $i = 1, \ldots, n$ or there exists a p such that $v_p > u_p$ and $v_i = u_i$ for all $i = 1, \ldots, p-1$

Lemma 5. *L is atomic only if $L = L_a$ for a letter $a \in A'$ which can be found using the following procedure:*

- *For each letter $b \in A'$, let $(w_{b1}, w_{b2}, \ldots, w_{bk})$ be the list of words in S_b ordered so that $|w_{b1}| \leq |w_{b2}| \leq \ldots \leq |w_{bk}|$. Define vector $v_b = (|w_{b1}|, |w_{b2}|, \ldots, |w_{bk}|)$.*
- *Find a letter b such that v_b majorizes all vectors v_c with $c \in A'$.*
- *Look at the second letter of each word in S_b, if any of these letters belong to A', say $c \in A'$, then choose $a = c$, otherwise choose $a = b$.*

Proof. Suppose that vector v_c does not majorize v_b and assume, for contradiction, $L = L_c$. We list the words of S_c as $(w_{c1}, w_{c2}, \ldots, w_{cl})$ with $|w_{c1}| \leq |w_{c2}| \leq \ldots \leq |w_{cl}|$ and the words of S_b as $(w_{b1}, w_{b2}, \ldots, w_{bk})$ with $|w_{b1}| \leq |w_{b2}| \leq \ldots \leq |w_{bk}|$. Let w'_{ci} and w'_{bi} denote the words obtained from w_{ci} and w_{bi} by removing first letters c and b, respectively.

Since $L = L_c$, by Lemma 4, we have that each word w'_{cj} for $j = 1, 2, \ldots, l$ contains a word w'_{bi} for some $i = 1, 2, \ldots, k$. Then $|w_{c1}| = |w_{b1}|$, since otherwise w'_{c1} is strictly shorter than any word in S'_b, in which case it cannot contain a word in S'_b. Let p be the largest integer such that $|w_{b_1}| = |w_{b2}| = \ldots = |w_{bp}|$. Clearly, as v_c does not majorize v_b, we must also have $|w_{c1}| = |w_{c2}| = \ldots = |w_{cp}|$. For each $i \leq p$ and $j > p$, we have $|w'_{ci}| < |w'_{bj}|$. Therefore, for each $i \leq p$ the word w'_{ci} contains a word w'_{bj} with $j \leq p$, and since these words have the same length, we conclude that the set of words w'_{ci} for $i = 1, 2, \ldots p$ is just a permutation of the set of words w'_{bj} with $j = 1, 2, \ldots, p$. Now, take a word $w_{c(p+1)}$, which must exist, since v_c does not majorize v_b. If $w'_{c(p+1)}$ contains a word w'_{bj} with $j \leq p$, then $w'_{c(p+1)}$ must contain a word w'_{ch} with $h \leq p$, which is not possible, as the words in the set S_c are incomparable. This means, similarly as before, that the words in S'_c of length $|w'_{c(p+1)}|$ must form a permutation of words in S'_b of the same length. Continuing this way, we must conclude that S_c has the same number of words as S_b and $v_c = v_b$, which is a contradiction to the assumption that v_c does not majorize v_b.

Finally, consider the set $A'' = \{b \in A' : v_b$ majorizes all v_c with $c \in A'\}$. Then for any $b, c \in A''$, we have $v_b = v_c$. Moreover, if for some letter $a \in A''$ we have $L = L_a$, then, by the arguments in the previous paragraph, for any letter $b \in A''$ we have $S'_b = S'_a$. Since for all letters $b \in A''$ we have the same set S'_b, the second condition of Lemma 4, is either satisfied or not, regardless of the choice of $b \in A''$. We need to check the first condition of Lemma 4 by looking at the first letter of each word in the set S'_b. If such letter c belongs to A', then the only chance for $L = L_a$ for some $a \in A''$ is when $a = c$, since otherwise the first condition of Lemma 4 is not satisfied. On the other hand, if none of the first letters of S'_b belongs to A', then the first condition of Lemma 4 is satisfied for all sets S'_b with $b \in A''$, and since all these sets are equal, we have that either $L = L_a$ holds for all $a \in A''$ or for none of them, so it is enough to pick one of them, say $a = b$ to check whether $L_a = L$ or not. This finishes the proof. □

The final ingredient for our inductive argument is the following simple observation.

Lemma 6. *L_a is atomic if and only if $Free(W_a)$ is atomic.*

Proof. We recall that L_a can be presented as

$$L_a = \{pws : p \in Free(A'), w \in \{a, \emptyset\}, s \in Free(W_a)\}.$$

Suppose first that $Free(W_a)$ is atomic. Pick $x, y \in L_a$. Then $x = p_x w_x s_x$ and $y = p_y w_y s_y$ with $p_x, p_y \in Free(A')$, $w_x, w_y \in \{a, \emptyset\}$ and $s_x, s_y \in Free(W_a)$. Since $Free(W_a)$ is atomic, by JEP we have that there exists a word $s_z \in Free(W_a)$

containing s_x and s_y. Letting $p_z = p_x p_y$, we can define $z = p_z a s_z$. Clearly z contains both x and y and since $p_z \in Free(A')$, $s_z \in Free(W_a)$ we also have $z \in L_a$. So L_a satisfies JEP, and so it is atomic.

Now suppose L_a is atomic. Pick $x, y \in Free(W_a)$. Then since the words ax and ay both belong to L_a and L_a is atomic, by JEP there exists $z \in L_a$ which contains both ax and ay. Let us denote $z = pws$ with $p \in Free(A')$, $w \in \{a, \emptyset\}$ and $s \in Free(W_a)$. As ax is a subword of z, and a does not appear in p, we have that ax is a subword of ws, and since $w \in \{a, \emptyset\}$ we conclude that x is a subword of s. For the same reason, we have y is a subword of s. Since $s \in Free(W_a)$, we see that $Free(W_a)$ satisfies JEP, hence $Free(W_a)$ is atomic. Thus we conclude that L_a is atomic if and only if $Free(W_a)$ is atomic. □

We are now ready to prove the main result of the paper.

Proof of Theorem 1. Let $L = Free(w_1, w_2, \ldots, w_n)$ be a given language with $w_1, w_2, \ldots, w_n \in A^*$ incomparable words. If $|w_i| = 1$ for some $i = 1, 2, \ldots, n$, then remove such a word, as by Lemma 1 this operation does not affect atomicity. So assume, without loss of generality, that $|w_i| \geq 2$ for all $i = 1, 2, \ldots, n$. Now perform the procedure of Lemma 5 to find a letter $a \in A'$ such that L is atomic only if $L = L_a$.

Then perform the procedure of Lemma 4 to check whether $L_a = L$. If not, then we know that L is not atomic. Now consider the case when $L = L_a$. In this case, by Lemma 6, L_a is atomic if and only if $Free(W_a)$ is atomic, and to determine whether $Free(W_a)$ is atomic we can proceed inductively, as the total number of letters in the set W_a is smaller than in the original set of forbidden words.

Note that the most expensive step in terms of algorithmic complexity is the application of the procedure in Lemma 4, which takes $O(|S'_a|nm)$ steps. After completing the induction step we have a set of forbidden words with $|S'_a|$ fewer letters than the original set of forbidden words. Since the removal of $|S'_a|$ letters takes $O(|S'_a|nm)$ steps to complete, to finish the procedure, i.e. to remove all m letters, we will have the computational complexity of order $O(m \times nm) = O(nm^2)$. This finishes the proof. □

We finish this section with a couple of corollaries that follow from the proof of the main theorem. The first corollary gives a simple representation of all atomic subword-closed languages. Following the algorithm of the main theorem, one can efficiently move between this representation and the representation of the atomic language given by forbidden subwords.

Corollary 1. Let L be a subword-closed language over a finite alphabet A. Then L is atomic if and only if there exists a sequence of subsets $A_i \subseteq A$ for $i = 1, 2, \ldots, m+1$ and letters $a_i \in A_i$ for $i = 1, 2, \ldots, m$, such that

$$L = \{w_1 a'_1 w_2 a'_2 \ldots w_m a'_m w_{m+1} : a'_i \in \{a_i, \emptyset\} \text{ for all } i \in \{1, 2, \ldots, m\} \text{ and } w_i \in Free(A_i) \text{ for all } i \in \{1, 2, \ldots, m+1\}\}.$$

The second corollary gives a simple description of all atomic languages defined by one or two forbidden subwords.

Corollary 2. Let $w, w_1, w_2 \in A^*$ be some words over a finite alphabet A with w_1 and w_2 incomparable. Then

- $Free(w)$ is atomic.
- $Free(w_1, w_2)$ is atomic if and only if $w_1 = pw's$, $w_2 = pw''s$ for some words $p, s \in A^*$ and some words $w', w'' \in A^*$ such that either $|w'| = 1$ or $|w''| = 1$.

Proof. Applying the algorithm for deciding atomicity to the language $Free(w)$ with $w = x_1x_2 \dots x_k$, for some $x_1, x_2, \dots, x_k \in A$, we see that $Free(w)$ is atomic, if and only if $Free(x_2x_3 \dots x_k)$ is atomic, if and only if $Free(x_3 \dots x_k)$ is atomic, ..., if and only if $Free(x_k)$ is atomic. Clearly, $Free(x_k)$ is atomic and hence $Free(w)$ is atomic. Moreover, we can represent this language as

$$\{w_1x'_1w_2x'_2 \dots w_{k-1}x'_{k-1}w_k : x'_i \in \{x_i, \emptyset\} \text{ for all } i = \{1, 2, \dots, k-1\} \text{ and } w_i \in Free(x_i) \text{ for all } i = \{1, 2, \dots, k\}\}.$$

Let us now write $w_1 = pw's$ and $w_2 = pw''s$, where p and s are the longest common prefix and the longest common suffix of w_1 and w_2, respectively. Note that $w' \neq \emptyset$ and $w'' \neq \emptyset$, as otherwise one of w_1 and w_2 would be a subword of the other, which is not allowed. Following the algorithm we see that $Free(w_1, w_2)$ is atomic if and only if $Free(w's, w''s)$ is atomic. Suppose that $|w''| \geq |w'|$. Let $w' = x_1x_2 \dots x_k$ and $w'' = y_1y_2 \dots y_l$ with $l \geq k$. Then, if $l > k$ the algorithm removes the letter from w'' and checks whether $y_2y_3 \dots y_ls$ contains $x_2 \dots x_ks$, which happens if and only if $y_2y_3 \dots y_l$ contains $x_2 \dots x_k$. If it does, then the length of $y_2y_3 \dots y_l$ is still bigger than of w', in which case it removes one more letter and checks whether $y_3y_4 \dots y_l$ contains $x_2 \dots x_k$. The process continues until the length of the words $y_{l-k+2}y_{l-k+3} \dots y_l$ and $x_2 \dots x_k$ are the same, in which case to contain one another means to be equal. Now, if $k \geq 2$, this means $x_k = y_l$ and this contradicts the fact that s is the longest suffix. Thus if $k \geq 2$ the two words cannot contain each other, and we conclude that the language is not atomic. On the other hand, if $k = 1$, then clearly all containments are satisfied trivially and algorithm proceeds without stopping, thus showing that for $k = 1$ the language is atomic. This finishes the proof. □

3 Concluding remarks and open problems

In this paper we have proved that atomicity, or equivalently the joint embedding property, is algorithmically decidable for subword-closed languages. However, the question of computing a decomposition of a non-atomic language into two proper subword-closed sublanguages remains open.

The decidability procedure developed in this paper implies, in particular, that atomicity is decidable for hereditary subclasses of threshold graphs [8], since there is a bijection between threshold graphs on n vertices and binary words of length $n - 1$. Note that for general hereditary classes this question is undecidable [3].

Threshold graphs constitute a prominent example of graphs of bounded *lettericity* [10] and we conjecture that our result implies decidability of atomicity for all hereditary classes in this family.

Clique-width [4] is a notion which is more general than lettericity in the sense that bounded lettericity implies bounded clique-width but not necessarily vice versa. Graphs of bounded clique-width can be described by words (algebraic expressions) over a finite alphabet, and we believe that decidability of atomicity can be extended to graphs of bounded clique-width.

The main result of this paper also implies that atomicity is decidable for classes of linear read-once Boolean functions closed under renaming variables and erasing variables from linear read-once expressions defining the functions, because, similarly to threshold graphs, linear read-once Boolean functions can be uniquely (up to renaming variables) described by binary words. Linear read-once functions appeared in the literature under various other names such as nested canalyzing functions, unate cascade functions [7], 1-decision lists [5], and we conjecture that decidability of atomicity can be extended to classes of d-decision lists for any fixed d. To support this conjecture, we observe that the main result of this paper is valid for subword-closed languages over *infinite* alphabets, provided that the set of minimal forbidden words is finite.

References

1. B. Alecu, V. Lozin, D. de Werra, The micro-world of cographs, *Discrete App. Math.*, accepted. https://doi.org/10.1016/j.dam.2021.11.004
2. M. Bodirsky, J. Rydval, A. Schrottenloher, Universal Horn sentences and the joint embedding property, preprint available at arXiv:2104.11123v3.
3. S. Braunfeld, The undecidability of joint embedding and joint homomorphism for hereditary graph classes. *Discrete Math. Theor. Comput. Sci.* 21 (2019), no. 2, Paper No. 9, 17 pp.
4. B. Courcelle, J. Engelfriet, G. Rozenberg, Handle-rewriting hypergraph grammars, *J. Computer and System Sci.* 46 (2) (1993) 218–270.
5. T. Eiter, T. Ibaraki, K. Makino. Decision lists and related Boolean functions. *Theoretical Computer Science*, 270(1) (2002) 493–524.
6. G. Higman, Ordering by divisibility in abstract algebras, *Proceedings of the London Mathematical Society*, (3) 2 (1952) 326–336.
7. A. S. Jarrah, B. Raposa, and R. Laubenbacher. Nested canalyzing, unate cascade, and polynomial functions. *Physica D: Nonlinear Phenomena*, 233(2) (2007) 167–174.
8. N. V. R. Mahadev, U. N. Peled, Threshold graphs and related topics. Annals of Discrete Mathematics, 56. North-Holland Publishing Co., Amsterdam, 1995. xiv+543 pp.
9. M. McDevitt, N. Ruškuc, N. Atomicity and well quasi-order for consecutive orderings on words and permutations. *SIAM J. Discrete Math.* 35 (2021), no. 1, 495–520.
10. M. Petkovšek, Letter graphs and well-quasi-order by induced subgraphs. *Discrete Math.* 244 (2002), 375–388.
11. V. Vatter, Growth rates of permutation classes: from countable to uncountable. *Proc. Lond. Math. Soc.* (3) 119 (2019), no. 4, 960–997.