A Variant of Pattern Matching for Multiwords

V. Bruyère[1], **O. Carton**[2], **A. Decan**[1], **O. Gauwin**[1], **J. Wijsen**[1]

[1] Institut d'Informatique, Université de Mons – Belgium

[2] LIAFA, Université Paris Diderot - Paris 7 – France

Abstract

Multiwords are words in which a single symbol can be replaced by a nonempty set of symbols. A pattern w is *certain* in a multiword M if it occurs in *every* word that can be obtained by selecting one single symbol among the symbols provided in each position of M. Motivated by a problem on incomplete databases, we investigate a variant of the pattern matching problem which is to decide whether a pattern w is certain in a multiword M. We study the language $\mathsf{CERTAIN}(w)$ of multiwords in which w is certain. We show that this regular language is aperiodic for three large families of words and we study the size of its minimal automaton.

1 Introduction

Given a pattern w and a text t, the *pattern matching problem* is to find all the occurrences of w in t. There exist efficient algorithms that solve this problem, like the well-known Knuth-Morris-Pratt algorithm [1] and Boyer-Moore algorithm [2] (see also Chapters 3 and 4 in [3]).

Several extensions of this problem have been studied. Instead of a single pattern w, the Aho-Corasick algorithm efficiently finds in a text t all the occurrences of words w taken from a finite set of words [4]. A more general problem is the regular expression matching problem where the pattern is a set of words specified by a regular expression (see for instance Chapter 7 in [3]).

Other extensions deal with the pattern matching problem by allowing *don't-care* symbols in the pattern w and/or in the text t. In this case, some positions in the pattern or the text can contain a set of symbols, instead of a single symbol. A word with don't-care symbols represents a finite set of (classical) words obtained by selecting a single symbol among the symbols provided in each don't-care position. Therefore, if w is a pattern with don't-care symbols and t is a text, the problem consists in finding all the occurrences of words represented by w in the text t. When w is a pattern and t is a text with don't-care symbols, we are interested in finding the occurrences of w in t such that in each don't-care position i, the symbol at the corresponding position of w belongs to the set of symbols of t at position i.

When don't-care symbols are allowed, most of the existing exact methods for pattern matching are useless or have to be adapted. One among the first works in this framework has been presented by Fisher and Paterson in [5]. Without being exhaustive, let us also mention the recent references [6, 7, 8].

The interest in words with don't-care symbols is driven by applications in computational biology, cryptanalysis, musicology, and other areas. In computational biology, DNA sequences may still be considered to match each other if letter A (respectively, C) is juxtaposed with letter T (respectively, G); analogous juxtapositions may count as matches in protein sequences. In cryptanalysis, so far undecoded symbols may be known to match one of a specific set of letters in the alphabet. In music, single notes may match chords, or notes separated by an octave may match.

Our problem is motivated by research in incomplete historical databases, as described in [9]. Incomplete databases represent sets of possible databases, also called repairs. Given a query, we are interested in answers that are true in every repair of the incomplete database. The queries ask whether a sequence $w = a_1 \dots a_n$ is encountered in every repair, i.e. whether in every repair we can find a sequence $t_1 \dots t_n$ of successive time points such that a_1 holds at t_1, a_2 holds at t_2, and so on. This problem can be seen as a variant of pattern matching: given a pattern w and a text t with don't-care symbols, does w appear as a factor of each word z represented by t? It is important to notice that we want to be sure that w appears in *each* z, and not in *some* z. In the database scenario, it is significant to ask whether our queries are first-order expressible. The reason for this is that if a query is first-order expressible, then its data complexity is in P (and even in AC^0), and it can be encoded in standard database languages like SQL.

This variant of the pattern matching problem motivated by querying incomplete databases was first stated in [9] together with several partial results. Given a pattern w, the authors provide a deterministic finite automaton $\mathcal{A}(w)$ recognizing the set $\mathsf{CERTAIN}(w)$ of all words t with don't-care symbols such that w is a factor of each word z represented by t. This automaton is a kind of Knuth-Morris-Pratt automaton (see Chapter 9 of [10]), with a more sophisticated use of the prefixes of w. They also prove that for a particular class of words w, the regular set $\mathsf{CERTAIN}(w)$ is aperiodic, or equivalently [11, 12], first-order expressible.

In this article, we prove that $\mathsf{CERTAIN}(w)$ is aperiodic for three large families of words w including powers of primitive words (for a power ≥ 3) and powers of unbordered words. Based on these results, we conjecture the aperiodicity of $\mathsf{CERTAIN}(w)$ for *all* words w. We also investigate the size of the minimal deterministic automaton $\mathcal{A}_{\min}(w)$ recognizing $\mathsf{CERTAIN}(w)$. On one hand, each state of $\mathcal{A}(w)$ consists in a subset of prefixes of w [9]. Hence, the size of $\mathcal{A}(w)$ may be exponential in the length of w. On the other hand, each state of the Knuth-Morris-Pratt automaton is a prefix of w, so the size of this automaton is linear in the length of w. We experimentally compute the size of $\mathcal{A}_{\min}(w)$ for a large number of words w over a two-letter alphabet. Surprisingly, in all our experiments, it is bounded by $|w| + \lfloor \frac{|w|}{2} \rfloor$, and in most cases, it equals $|w| + 1$, the size of the Knuth-Morris-Pratt automaton.

In the literature, different terms have been used for words with don't-care symbols like indeterminate words [6], partial words, words with holes or jokers [13, 14, 15]. In each case, either the don't-care symbol means any letter of the alphabet, or it has to be selected among a subset of the alphabet depending on its position in the word. In this article, we follow the second approach and we use the term *multiword* coined in [9].

The remainder of this article is organised as follows. The next section introduces terminology and notations and formalizes the problems we are interested in. Section 3 contains our main results. It establishes the aperiodicity of $\mathsf{CERTAIN}(w)$ for several large families of words w. These families contain powers of primitive words (power $\geq$ 3) and powers of unbordered words. Section 4 concentrates on the size of the minimal automaton recognizing $\mathsf{CERTAIN}(w)$. Based on experiments, we suggest an upper bound on this size which is linear in $|w|$, and we give families of words for which this upper bound is tight.

2 Preliminaries

2.1 Multiwords

In this section, we recall the basic definitions and results on multiwords [9]. Let $\Sigma = \{a, b, c, \dots\}$ be a finite alphabet of symbols.

Definition 1 A *word* of *length* $n \geq 0$ is a sequence $a_1 a_2 \dots a_n$ of symbols. We denote by $|w|$ the length of w. The *empty word*, denoted by ϵ, has length 0. The *concatenation* of words w_1 and w_2 is denoted by $w_1 \cdot w_2$ and naturally extends to sets of words. If S and T are sets of words then $S \cdot T = \{w_1 \cdot w_2 \mid w_1 \in S, w_2 \in T\}$. If $w = p \cdot q$, then p is called a *prefix* of w and q a *suffix*. A prefix (or suffix) of w that is distinct from w is called *proper*. We say that a word w is a *factor* of v, denoted by $v \Vdash w$, if there exist words p and q such that $v = p \cdot w \cdot q$. We denote as usual by Σ^* the set of all words over Σ, and $\Sigma^+ = \Sigma^* \setminus \{\epsilon\}$. A word w is called *unbordered* if no nonempty proper suffix of w is a prefix of w. A word $w \in \Sigma^+$ is *primitive* if $w = v^k$ implies $k = 1$.

Definition 2 We define the *powerset alphabet* as $\widehat{\Sigma} = 2^\Sigma \setminus \{\emptyset\}$. A *multiword* $M = \langle A_1, \dots, A_n \rangle$ is a finite word over the powerset alphabet $\widehat{\Sigma}$, i.e. $A_i \subseteq \Sigma$ and $A_i \neq \emptyset$ for all i. Given a multiword $M = \langle A_1, \dots, A_n \rangle$, we define the set of words represented by M:

$$\mathsf{words}(M) := \{a_1 a_2 \dots a_n \mid \forall i \in \{1, \dots, n\} : a_i \in A_i\}.$$

Let w be a word. We say that a word w is *certain* in M, denoted $M \Vdash_{\mathsf{certain}} w$, if w is a factor of every word in $\mathsf{words}(M)$.

Example 1 The following multiword M contains two symbols with values $\{a, b\}$ and $\{c, d\}$. Curly braces are omitted for symbols that are singletons; for example, $\{a\}$ is written as a.

So, for $M = \langle a, b, d, a, b, c, a, \{a, b\}, b, d, a, b, \{c, d\}, a, b, c, a, b \rangle$, we have:

$$\begin{aligned} \mathsf{words}(M) \quad = \{ \quad & abdabca\underline{abdabcab}cab, \\ & abdabcaabd\underline{abdabcab}, \\ & \underline{abdabcab}bdabcabcab, \\ & \underline{abdabcab}bd\underline{abdabcab} \ \}. \end{aligned}$$

Hence, $M \Vdash_{\mathsf{certain}} abdabcab$ because $abdabcab$ is a factor of each word in $\mathsf{words}(M)$.

Definition 3 Given a word $w \in \Sigma^+$,
we are interested in the language $\mathsf{CERTAIN}(w) \subseteq \widehat{\Sigma}^*$ defined as follows:

$$\mathsf{CERTAIN}(w) := \{M \in \widehat{\Sigma}^* \mid M \Vdash_{\mathsf{certain}} w\}$$

In the next subsection, we recall known results [9] about the language $\mathsf{CERTAIN}(w)$, and we formulate two problems for which we provide partial answers in the remainder of the article.

2.2 Known results and problems

We first recall a procedure for deciding membership of $\mathsf{CERTAIN}(w)$. This procedure is interesting because it directly leads to the construction of a deterministic finite automaton which accepts $\mathsf{CERTAIN}(w)$.

Definition 4 Let $u, w \in \Sigma^*$. We note $\mathsf{sufpre}(u, w)$ the maximal suffix of u that is also a prefix of w. For a set S of words, we define $\mathsf{sufpre}(S, w) = \{\mathsf{sufpre}(u, w) \mid u \in S\}$. We define $\lfloor S \rfloor = S \setminus (\Sigma^+ \cdot S)$, that is, $\lfloor S \rfloor$ is the smallest set of words satisfying $\lfloor S \rfloor \subseteq S$ and $\lfloor S \rfloor$ contains a suffix of every word in S.

For example, $\mathsf{sufpre}(abcd, cde) = cd$ and $\mathsf{sufpre}(ab, c) = \epsilon$. For the set $S = \{aa, ac, abc, bc, c\}$, we have $\lfloor S \rfloor = \{c, aa\}$. The following lemma provides a procedure for deciding membership of $\mathsf{CERTAIN}(w)$ [9]:

Lemma 1 *Let $M = \langle A_1, \ldots, A_n \rangle \in \widehat{\Sigma}^*$ and $w \in \Sigma^+$. Let $\langle S_0, S_1, \ldots, S_n \rangle$ be the sequence such that:*

- *$S_0 = \{\epsilon\}$, and*
- *for every $i \in \{1, \ldots, n\}$, $S_i = \lfloor \mathsf{sufpre}(S_{i-1} \cdot A_i, w) \setminus \{w\} \rfloor$.*

Then, $M \in \mathsf{CERTAIN}(w)$ if and only if S_n is empty.

Example 2 The construction is illustrated in Figure 1 for the word $w = abdabcab$ and the multiword introduced in Example 1:

$$M = \langle a, b, d, a, b, c, a, \{a, b\}, b, d, a, b, \{c, d\}, a, b, c, a, b \rangle$$

The set S_8, for instance, is computed from $S_7 \cdot A_8 = \{abdabcaa, w\}$, in which $abdabcaa$ is replaced with its suffix a, and the word w is removed.

Intuitively, the construction of $\langle S_0, S_1, \ldots, S_n \rangle$ for a word w and a multiword M can be thought of as executing the pattern matching algorithm of Knuth-Morris-Pratt [1] simultaneously on every word in $\mathsf{words}(M)$. In particular, if every symbol of the multiword is a singleton, this algorithm runs in a way that is similar to the Knuth-Morris-Pratt algorithm.

Lemma 1 suggests the following construction of a deterministic finite automaton recognizing $\mathsf{CERTAIN}(w)$ [9].

Definition 5 Let P be the set of proper prefixes of w. We define the deterministic finite automaton $\mathcal{A}(w) = (Q, \widehat{\Sigma}, S_0, F, \delta)$ on the powerset alphabet $\widehat{\Sigma}$. Its finite set of states is $Q = \{\lfloor S \rfloor \mid S \subseteq P\}$. The initial state is $S_0 = \{\epsilon\}$, and the final states are $F = \{\emptyset\}$. The transition function $\delta : Q \times \widehat{\Sigma} \to Q$ is defined by:

$$\delta(S, A) = \lfloor \mathsf{sufpre}(S \cdot A, w) \setminus \{w\} \rfloor$$

	$S_0 = \{\epsilon\}$	$A_{10} = \{d\}$	$S_{10} = \{abd\}$
$A_1 = \{a\}$	$S_1 = \{a\}$	$A_{11} = \{a\}$	$S_{11} = \{abda\}$
$A_2 = \{b\}$	$S_2 = \{ab\}$	$A_{12} = \{b\}$	$S_{12} = \{abdab\}$
$A_3 = \{d\}$	$S_3 = \{abd\}$	$A_{13} = \{c, d\}$	$S_{13} = \{abdabc, abd\}$
$A_4 = \{a\}$	$S_4 = \{abda\}$	$A_{14} = \{a\}$	$S_{14} = \{abdabca, abda\}$
$A_5 = \{b\}$	$S_5 = \{abdab\}$	$A_{15} = \{b\}$	$S_{15} = \{abdab\}$
$A_6 = \{c\}$	$S_6 = \{abdabc\}$	$A_{16} = \{c\}$	$S_{16} = \{abdabc\}$
$A_7 = \{a\}$	$S_7 = \{abdabca\}$	$A_{17} = \{a\}$	$S_{17} = \{abdabca\}$
$A_8 = \{a, b\}$	$S_8 = \{a\}$	$A_{18} = \{b\}$	$S_{18} = \{\}$
$A_9 = \{b\}$	$S_9 = \{ab\}$		

Figure 1: Illustration of the construction in Lemma 1.

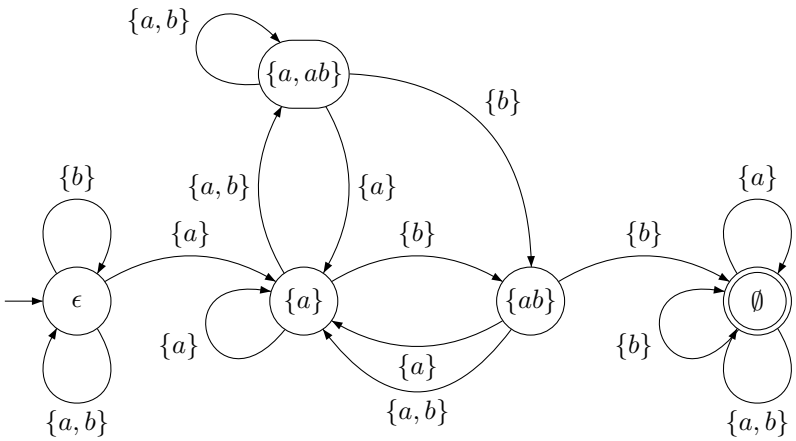


Figure 2: The automaton $\mathcal{A}(abb)$ over $\Sigma = \{a, b\}$.

For instance, Figure 2 shows the reachable states of $\mathcal{A}(abb)$ for the alphabet $\Sigma = \{a, b\}$. The following proposition states that $\mathsf{CERTAIN}(w)$ is regular.

Proposition 1 ([9]) *For every word w, $\mathcal{A}(w)$ recognizes $\mathsf{CERTAIN}(w)$.*

Note that the size of $\mathcal{A}(w)$ may be exponential in $|w|$, as the states are subsets of prefixes of w.

Problem 1 *Is the size of the minimal automaton recognizing $\mathsf{CERTAIN}(w)$ polynomially bounded in $|w|$?*

We recall that the Knuth-Morris-Pratt automaton of w associated to the pattern matching algorithm of Knuth-Morris-Pratt (as done in [10]) has a linear size equal to $|w| + 1$.

Another result of [9] states that $\mathsf{CERTAIN}(w)$ is aperiodic for a particular family of words w.

Definition 6 A language L over an alphabet Σ is *aperiodic* if there exists $k > 0$ such that for every word $p, u, q \in \Sigma^*$, we have:

$$p \cdot u^k \cdot q \in L \iff p \cdot u^{k+1} \cdot q \in L$$

Proposition 2 ([9]) *Let $a \in \Sigma$. If w is a word over $\Sigma\backslash\{a\}$, then* $\mathsf{CERTAIN}(a \cdot w)$ *and* $\mathsf{CERTAIN}(w \cdot a)$ *are aperiodic.*

The question whether $\mathsf{CERTAIN}(w)$ is aperiodic for all w has already been raised in [9].

Problem 2 *Determine if, for every word $w \in \Sigma^+$,* $\mathsf{CERTAIN}(w)$ *is* aperiodic. *This corresponds to proving that for every $w \in \Sigma^+$, there exists $k > 0$ such that for all $P, U, Q \in \widehat{\Sigma}^*$,*

$$P \cdot U^k \cdot Q \Vdash_{\mathsf{certain}} w \iff P \cdot U^{k+1} \cdot Q \Vdash_{\mathsf{certain}} w$$

3 Aperiodicity

In this article we show that $\mathsf{CERTAIN}(w)$ is aperiodic for several large families of words w. Our proofs are based on two main lemmas. Lemma 2 states that one can avoid one of the two implications of Definition 6 for proving aperiodicity. Lemma 3 is the core tool in all our proofs.

Lemma 2 *Given a word $w \in \Sigma^+$,* $\mathsf{CERTAIN}(w)$ *is aperiodic if there exists $k > 0$ such that for all $P, U, Q \in \widehat{\Sigma}^*$,*

$$P \cdot U^k \cdot Q \Vdash_{\mathsf{certain}} w \implies P \cdot U^{k+1} \cdot Q \Vdash_{\mathsf{certain}} w$$

Proof.[Sketch] The proof is based on the following property of regular languages, that can be applied to $\mathsf{CERTAIN}(w)$ since it is regular (see Chapter 1 of [16]). There exist $n \geq 0$, $p \geq 1$ such that $\forall m \geq n$, $\forall P, U, Q \in \widehat{\Sigma}^*$,

$$P \cdot U^m \cdot Q \Vdash_{\mathsf{certain}} w \iff P \cdot U^{m+p} \cdot Q \Vdash_{\mathsf{certain}} w$$

□

Definition 7 Given a word $w = a_1 a_2 \cdots a_n$, we define the *positions* in w as the set $\{0, 1, \ldots, n\}$. For $1 \leq i \leq n-1$, position i of w can be seen as the position *between* a_i and a_{i+1}. Position 0 precedes the first letter, while position n follows the last letter.

Lemma 3 *Let $w \in \Sigma^+$ be a word. Let $P, U, Q \in \widehat{\Sigma}^*$ be multiwords such that $P \cdot U^k \cdot Q \Vdash_{\mathsf{certain}} w$. Let $p \in \mathsf{words}(P)$, $q \in \mathsf{words}(Q)$ and $u \in \mathsf{words}(U^{k+1})$. If $m = p \cdot u \cdot q$ does not contain w as a factor, then for every position π in m, if $|P| \leq \pi \leq |P \cdot U^k|$ then there exist positions $\pi' < \pi$ and $\pi'' > \pi + |U|$ in m such that the nonempty factors x (resp. y) between π' and π (resp. $\pi + |U|$ and π'') satisfy $w = x \cdot y$.*

The situation in Lemma 3 is depicted in Figure 3. The couple (x, y) mentioned in Lemma 3 is called a *decomposition of w at position π*.

Proof. Let π be a position in m such that $|P| \leq \pi \leq |P \cdot U^k|$. We can assume $m = p \cdot v_1 \cdot u_1 \cdot v_2 \cdot q$ with $|p \cdot v_1| = \pi$ and $|u_1| = |U|$. Let $m' = p \cdot v_1 \cdot v_2 \cdot q$. From $P \cdot U^k \cdot Q \Vdash_{\mathsf{certain}} w$ and $m' \in \mathsf{words}(P \cdot U^k \cdot Q)$, we have $m' \Vdash w$. The situation is:

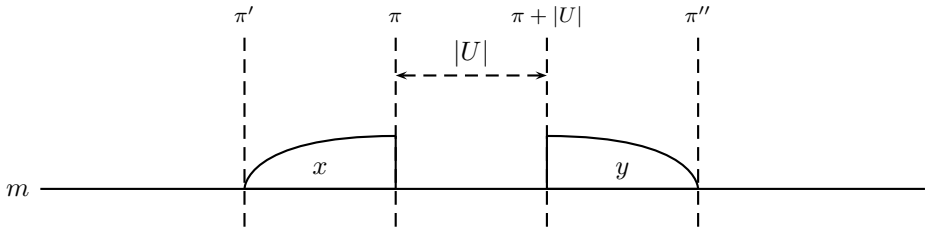


Figure 3: Decomposition (x, y) of w at position π.

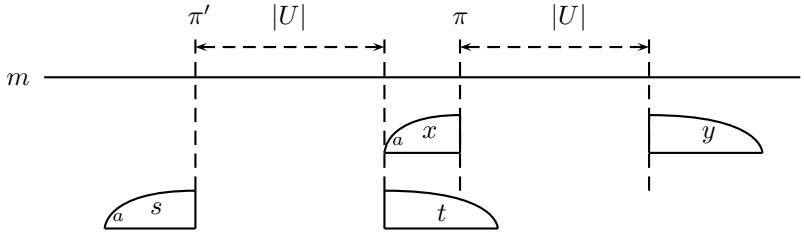


Figure 4: Configuration in the proof of Proposition 2.

$$m = \overbrace{p \cdot v_1 \cdot u_1 \cdot v_2 \cdot q}^{\not\Vdash w}$$

$$m' = \underbrace{\overbrace{p \cdot v_1}^{\not\Vdash w} \quad \overbrace{v_2 \cdot q}^{\not\Vdash w}}_{\Vdash w}$$

But $m \not\Vdash w$ implies $p \cdot v_1 \not\Vdash w$ and $v_2 \cdot q \not\Vdash w$, so it must be the case that $p \cdot v_1$ ends with some nonempty prefix x of w and $v_2 \cdot q$ starts with some nonempty suffix y of w and that $w = x \cdot y$. We can take $\pi' = \pi - |x|$ and $\pi'' = \pi + |u_1| + |y|$. □

In other words, this lemma states that, for every position π of m (under the hypotheses), there exist a prefix x of w just before π and a suffix y of w just after $\pi + |U|$, such that $x \cdot y = w$. To illustrate this lemma, we can use it to rewrite the proof of Proposition 2.

Proof.[Proposition 2] We show that Lemma 2 can be applied, with $k = |w| + 1$. Let P, U and Q be multiwords such that $P \cdot U^k \cdot Q \Vdash_{\mathsf{certain}} w$.
Let $m = p \cdot u \cdot q$ be such that $p \in \mathsf{words}(P)$, $u \in \mathsf{words}(U^{k+1})$ and $q \in \mathsf{words}(Q)$. Assume, for contradiction, that $m \not\Vdash w$. Let us consider the position $\pi = |p| + |w| + |U|$.
To apply Lemma 3 at π, we have to show that $|P| \leq \pi \leq |P \cdot U^k|$. Obviously, $|P| \leq \pi$. Notice that $|w| \leq |U^{k-1}|$, because $|w| = k-1$ and $|U| > 0$ (otherwise we directly obtain a contradiction from $P \cdot U^k \cdot Q \Vdash_{\mathsf{certain}} w$ and $m \not\Vdash w$). By adding $|p| + |U|$ on both sides of the inequality $|w| \leq |U^{k-1}|$, we get $\pi \leq |P \cdot U^k|$. Thus we can apply Lemma 3 at position π, and get a decomposition (x, y) of w at π, as depicted in Figure 4. Let v_1, v_2 be such that $m = p \cdot v_1 \cdot v_2 \cdot q$ and $|p \cdot v_1| = \pi$. Hence, $p \cdot v_1$ ends with $x = a \cdot x'$. But since x is a proper nonempty prefix of w, and $\pi = |p| + |w| + |U|$, v_1 ends with x, i.e. $v_1 = v_1' \cdot a \cdot x'$ for

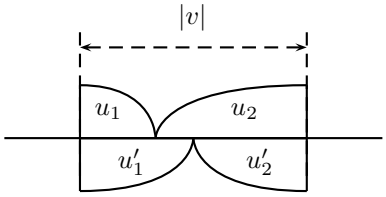


Figure 5: A forbidden situation for a primitive word v.

some v'_1. Consider now the position $\pi' = |p \cdot v'_1| - |U|$. Obviously, $\pi' \leq |P \cdot U^k|$ because $\pi' \leq \pi$. We also have $|P| \leq \pi'$, because $\pi' \geq \pi - |U| - |w|$. So we use Lemma 3 to obtain the decomposition (s, t) of w at position π' (see Figure 4). Therefore, t is a proper nonempty suffix of w, and its first letter is "a" because it is the first letter of x. This contradicts the definition of w. □

We now present three large families of words w for which we prove the aperiodicity of $\mathsf{CERTAIN}(w)$. As the related proofs are long and quite technical, we only give a sketch for convenience. The three proofs are based on two main arguments. The first one is Lemma 3, and the second one uses a notion of synchronization, that differs for each family.

The first family contains powers (≥ 3) of a primitive word. This is a very interesting family since it is well known [17] that every word is a power (≥ 1) of a primitive word.

Theorem 1 *If $w = v^k \cdot v'$ where v is primitive, $k \geq 3$, and v' is a (possibly empty) proper prefix of v, then $\mathsf{CERTAIN}(w)$ is aperiodic.*

Proof.[Sketch] Consider a word m over Σ, with a factor u of size $|v|$. As v is primitive, there is at most one pair $(u_1, u_2) \in \Sigma^* \times \Sigma^+$ such that $u = u_1 \cdot u_2$ and $u_2 \cdot u_1 = v$. In other words, for the considered w, when two prefixes of w share a common window of size $|v|$, they have to synchronize according to v. For instance, the situation depicted in Figure 5 cannot hold for a primitive word v. This property still holds for a window of size $|v| - 1$. This can be proved using Fine and Wilf's theorem [18].

The complete proof is based on this synchronizing property and our key lemma (Lemma 3). It is rather long and technical. □

The second family is composed of every power of an unbordered word. Notice that it contains the words of Proposition 2.

Theorem 2 *If $w = v^k$ with v an unbordered word and $k \geq 1$, then $\mathsf{CERTAIN}(w)$ is aperiodic.*

Proof.[Sketch] The proof is in the vein as the previous one. It is based on Lemma 3 and the following synchronization property. As v is unbordered, a prefix of w cannot overlap with a suffix of w, except if it is a power of v. The complete proof is again long. □

Given a symbol $a \in \Sigma$, the last family contains words w in which one among the distances between two consecutive a's is smaller than the other ones.

Proposition 3 *Let $a \in \Sigma$. Let $w = r_0 \cdot a \cdot r_1 \cdot a \cdots a \cdot r_n$ with*

1. *a not in $r_0 \cdot r_1 \cdots r_n$*
2. *$\exists i \in \{1, \ldots, n-1\}, \forall j \neq i \in \{0, \ldots, n\}, |r_i| < |r_j|$*

Then $\mathsf{CERTAIN}(w)$ *is aperiodic.*

PROOF.[Sketch] We show Lemma 2 for k chosen large enough. Let P, U and Q be multiwords such that $P \cdot U^k \cdot Q \Vdash_{\mathsf{certain}} w$. Assume by contradiction that there exists a word $m = p \cdot u \cdot q$ such that $p \in \mathsf{words}(P)$, $u \in \mathsf{words}(U^{k+1})$, $q \in \mathsf{words}(Q)$, and $m \not\Vdash w$. The proof is in two parts. By Lemma 3, the first part states that there exists some position π in m such that in the decomposition (x, y) at position π, either x or y contains the factor $a \cdot r_i \cdot a$ of w. Again by Lemma 3, the second part shows that m must contain w as a factor by choosing appropriate positions π' for decompositions around the factor $a \cdot r_i \cdot a$ located in x or y in the first step. This leads to the contradiction. □

The three preceding families do not cover all the words $w \in \Sigma^+$. Nevertheless, we are convinced that aperiodicity of $\mathsf{CERTAIN}(w)$ holds for every w.

4 Size of the minimal automaton

This section investigates Problem 1. The size of the minimal deterministic automaton[1] recognizing a language gives insights on the algebraic structure of this language. A small size of this automaton for $\mathsf{CERTAIN}(w)$ could imply strong properties of this language. We did a number of experiments in order to study the size of the minimal deterministic automaton $\mathcal{A}_{\min}(w)$ recognizing $\mathsf{CERTAIN}(w)$, over the alphabet $\Sigma = \{a, b\}$. Other experiments on larger alphabets confirmed these observations.

Exhaustive experiments. The first experiments concern an exhaustive study of the set of words w over $\Sigma = \{a, b\}$ such that $1 \leq |w| \leq 16$.

We obtained some surprising results:

- No minimal automaton $\mathcal{A}_{\min}(w)$ has more than $|w| + \lfloor \frac{|w|}{2} \rfloor$ states for $|w| \geq 2$.
- A majority of minimal automata $\mathcal{A}_{\min}(w)$ has $|w| + 1$ states, as in Knuth-Morris-Pratt automata.

  This means that for these words w, each state of $\mathcal{A}(w)$ having several prefixes can be merged with a state having only one prefix, i.e. a state (equivalent to a state) of the Knuth-Morris-Pratt automaton.

  The situation is depicted in Figure 6, for all words of length 14. For these words, the size of $\mathcal{A}_{\min}(w)$ ranges from 15 to 21. For each size in this range, the number of minimal automata of this size is indicated.
- The distribution of the number of minimal automata for each size of automata is quite similar for each length of w. Figure 7a presents this distribution for $|w|$ ranging from 1 to 16 (the number of automata is in logscale).

[1] In this article, the size of an automaton is its number of states.

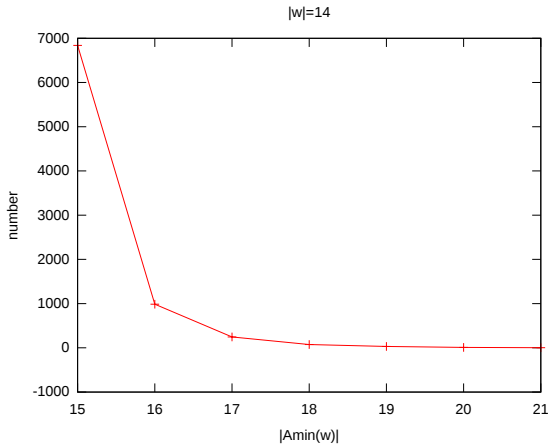
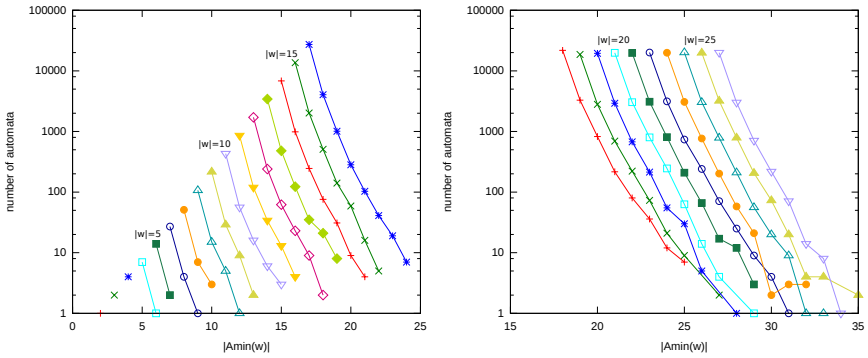


Figure 6: Distribution of the number of automata for words w of length 14.



(a) Exhaustive from length 1 to 16

(b) Random from length 17 to 26

Figure 7: Distribution of the number of minimal automata for each size of automata

Non-exhaustive experiments. We also ran a lot of experiments for random words of length up to 26. These experiments (about 250.000 words) never gave a counter-example for these three observations.

Figure 7b shows the distribution of the number of automata for the random experiments. Data are represented in the same way as in Figure 7a.

Based on these experiments, we want to reformulate Problem 1 in a more precise way:

Problem 3 *Determine if, for all $w \in \Sigma^*$ with $|w| \geq 2$, the size of the minimal automaton recognizing* $\mathsf{CERTAIN}(w)$ *is at most* $|w| + \lfloor \frac{|w|}{2} \rfloor$.

Families of words w maximizing $|\mathcal{A}_{\min}(w)|$. Through our experiments we identified several families of words w for which the minimal automaton $\mathcal{A}_{\min}(w)$ reaches what we think to be the highest number of states. These families are:

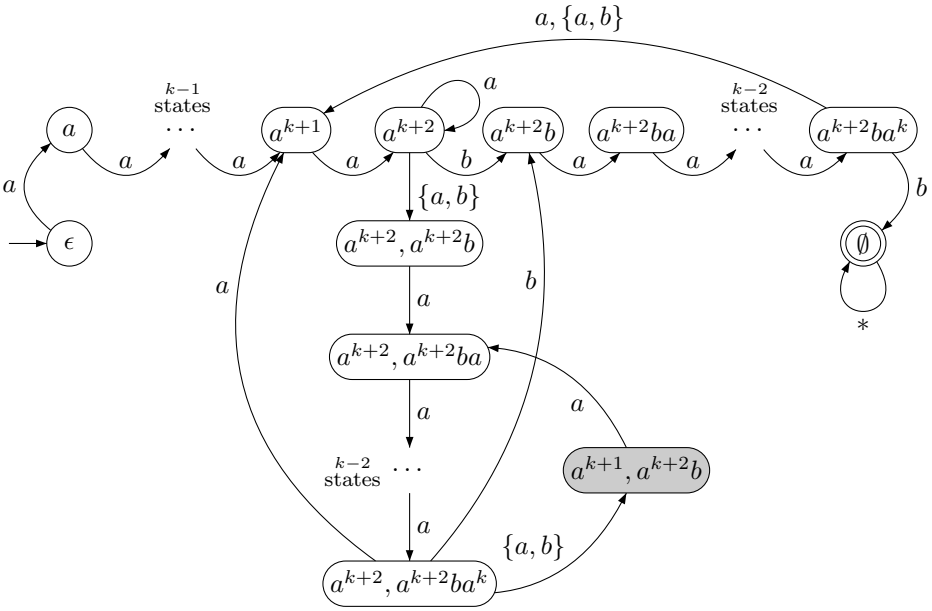


Figure 8: Automaton $\mathcal{A}(w)$ for $w = a^{k+2}ba^kb$.

- $a^{k+2}ba^kb$ (for $0 \leq k$)
- $a^{k+2}ba^jba^kba^jb$ (for $0 < j < k$)
- $a^{k+3}ba^kb$ (for $0 \leq k$)
- $a^{k+3}ba^jba^kba^jb$ ($0 < j < k$)

For the first family, i.e. $w = a^{k+2}ba^kb$, we explicitly construct the minimal automaton $\mathcal{A}_{\min}(w)$ and we show that it has a size of $3k + 6 = |w| + \lfloor \frac{|w|}{2} \rfloor$.

Proposition 4 *Let $k \geq 0$. Let $w = a^{k+2}ba^kb$. The minimal automaton that recognizes* $\mathsf{CERTAIN}(w)$ *has $|w| + \lfloor \frac{|w|}{2} \rfloor$ states.*

Proof.[Sketch] Let $w = a^{k+2}ba^kb$ with $k \geq 0$. We use the procedure described in Definition 5 to construct the automaton $\mathcal{A}(w)$. The resulting automaton is illustrated in Figure 8. Notice that for simplicity, the lacking transitions all lead to state $\{\epsilon\}$ and the non-reachable states have not been drawn. The procedure ensures that this automaton recognizes exactly the language $\mathsf{CERTAIN}(w)$.

It is easy to see that the state in gray is equivalent to the state $\{a^{k+2}, a^{k+2}b\}$ so it can be merged with it. It can also be shown that the remaining states cannot be merged. This is done by exhibiting words that belong to the residual language of a first state, but not to the residual language of a second state. □

5 Conclusions and perspectives

A multiword M represents a set $\mathsf{words}(M)$ of words. Given a word w, we said that w is certain in M if it occurs in every word represented by M. We study

the set $\mathsf{CERTAIN}(w)$ which contains every multiword in which w is certain. In particular, motivated by the problem of querying incomplete databases, we are interested in the aperiodicity of $\mathsf{CERTAIN}(w)$. We prove it for three large families of words w that contain in particular powers of primitive words (power $\geq$ 3) and powers of unbordered words.

We also consider the size of the minimal deterministic automaton $\mathcal{A}_{\min}(w)$ recognizing $\mathsf{CERTAIN}(w)$. Regarding our numerous experiments, we suggest an upper bound for this size and we show that it is reached by the family of words $w = a^k b a^{k+2} b$.

Based on these results, we conjecture the aperiodicity of $\mathsf{CERTAIN}(w)$ for all words w and a size of $\mathcal{A}_{\min}(w)$ lower than or equal to $|w| + \lfloor \frac{|w|}{2} \rfloor$.

References

[1] D. E. Knuth, J. H. Morris, and V. R. Pratt, "Fast pattern matching in strings," *SIAM Journal on Computing*, vol. 6, no. 2, pp. 323–350, 1977.

[2] R. S. Boyer and J. S. Moore, "A fast string searching algorithm," *Communications of the ACM*, vol. 20, no. 10, pp. 762–772, 1977.

[3] M. Crochemore and W. Rytter, *Text Algorithms*. Oxford University Press, 1994.

[4] A. V. Aho and M. J. Corasick, "Efficient string matching: An aid to bibliographic search," *Communications of the ACM*, vol. 18, no. 6, pp. 333–340, 1975.

[5] M. Fischer and M. Paterson, "String matching and other products," *SIAM-AMS Proceedings, Complexity of Computation*, vol. 7, pp. 113–125, 1974.

[6] J. Holub, W. F. Smyth, and S. Wang, "Fast pattern-matching on indeterminate strings.," *Journal of Discrete Algorithms*, vol. 6, no. 1, pp. 37–50, 2008.

[7] M. S. Rahman, C. S. Iliopoulos, and L. Mouchard, "Pattern matching in degenerate DNA/RNA sequences," in *Workshop on Algorithms and Computation (WALCOM)* (M. Kaykobad and M. S. Rahman, eds.), pp. 109–120, Bangladesh Academy of Sciences (BAS), 2007.

[8] G. Kucherov, L. Noé, and M. A. Roytberg, "Subset seed automaton," in *Proceedings of the 12th International Conference on Implementation and Application of Automata (CIAA)*, pp. 180–191, Springer, 2007.

[9] V. Bruyère, A. Decan, and J. Wijsen, "On first-order query rewriting for incomplete database histories," in *Proceedings of the 16th International Symposium on Temporal Representation and Reasoning (TIME)*, pp. 54–61, 2009.

[10] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[11] M. P. Schützenberger, "On finite monoids having only trivial subgroups," *Information and Control*, vol. 8, no. 2, pp. 190–194, 1965.

[12] R. McNaughton and S. Papert, *Counter-free Automata*. Cambridge, MA: MIT Press, 1971.

[13] J. Berstel and L. Boasson, "Partial words and a theorem of fine and wilf," *Theor. Comput. Sci.*, vol. 218, no. 1, pp. 135–141, 1999.

[14] F. Blanchet-Sadri, *Algorithmic Combinatorics on Partial Words (Discrete Mathematics and Its Applications)*. Chapman & Hall/CRC, 2007.

[15] M. Crochemore, C. Hancart, and T. Lecroq, *Algorithms on Strings*. Cambridge University Press, 2007. 392 pages.

[16] J.-É. Pin, *Varieties of Formal Languages*. North Oxford, London and Plenum, New-York, 1986.

[17] M. Lothaire, *Combinatorics on words*. Cambridge University Press, 1997.

[18] N. J. Fine and H. S. Wilf, "Uniqueness theorems for periodic functions," *Proceedings of the American Mathematical Society*, vol. 16, pp. 109–114, 1965.