Testing avoidability on sets of partial words is hard

F. Blanchet-Sadri[1*] Raphaël M. Jungers[2†] Justin Palumbo[3]

August 17, 2007

Abstract

We prove that the problem of deciding whether a finite set of partial words is unavoidable is NP-hard for any alphabet of size larger or equal to two, which is in contrast with the well known feasability results for unavoidability of a set of full words. We raise some related questions on avoidability of sets of partial words.

Keywords: Combinatorics on words; Partial words; Unavoidable sets; 3SAT problem; NP-hard problems.

1 Introduction

A set of (full) words X over a finite alphabet A is *unavoidable* if no two-sided infinite word avoids X, that is, X is unavoidable if every two-sided infinite word over A has a factor in X. This concept was explicitly introduced in 1983 in connection with an attempt to characterize the rational languages among the context-free ones [9]. It is clear from the definition that from each unavoidable set we can extract a finite unavoidable subset, so the study can be reduced to finite unavoidable sets. There is a vast literature on unavoidable sets of words and we refer the reader to [7, 12, 14] for more information.

Partial words, or finite sequences of symbols over a finite alphabet that may contain a number of "do not know" symbols or "holes", appear in natural ways in several areas of current interest such as molecular biology, data

*This material is based upon work supported by the National Science Foundation under Grant No. DMS–0452020.

†Raphaël Jungers is a FNRS fellow (Belgian Fund for Scientific Research). His work is supported by the "Communauté francaise de Belgique - Actions de Recherche Concertées", and by the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office.

[1]University of North Carolina, P.O. Box 26170, Greensboro, NC 27402–6170, USA, blanchet@uncg.edu

[2]Department of Mathematical Engineering, Université catholique de Louvain, Avenue Georges Lemaitre 4, B–1348 Louvain-la-Neuve, Belgium, raphael.jungers@uclouvain.be

[3]UCLA Mathematics Department, Box 951555, Los Angeles, CA 90095–1555

communication, and DNA computing [2, 3, 11]. Unavoidable sets of partial words were introduced recently in [4], where the problem of classifying such sets of small cardinality was initiated and then further studied in [5]. In terms of unavoidability, sets of partial words serve as representations of sets of full words.

Efficient algorithms to determine if a finite set of full words is unavoidable are well known [12, 13]. For example, we can check whether there is a loop in the finite automaton of Aho and Corasick [1] recognizing $A^* \setminus A^*XA^*$. These same algorithms can be used to decide if a finite set of partial words X is unavoidable by determining the unavoidability of $\hat{X}$, the set of all full words *compatible* with an element of X. However this incurs a dramatic loss in efficiency, as each partial word u in X can contribute as many as $\|A\|^{\|H(u)\|}$ elements to $\hat{X}$. In [4], the question was raised as to whether there is an efficient algorithm to determine if a finite set of partial words is unavoidable. In this paper, we show that this problem is NP-hard by using techniques similar to those used in a recent paper of Blondel, Jungers and Protasov on the complexity of computing the capacity of codes that avoid forbidden difference patterns [6].

The contents of our paper are summarized as follows: In Section 2, we review basic concepts such as unavoidable sets of partial words. In Section 3, we discuss testing unavoidability of such sets. We prove that the problem of deciding whether a set of partial words is unavoidable is NP-hard for any alphabet of size larger or equal to two, which is in contrast with the well known feasability results for avoidability of a set of full words. In Section 4, we raise some related questions on avoidability of sets of partial words.

2 Preliminaries

This section reviews some background material.

2.1 Basics on partial words

Throughout this paper A is a fixed finite set called the *alphabet* whose elements we call *letters*. We use A^* (respectively, A^n) to denote the set of finite words over A (respectively, the set of words of length n over A). For $u \in A^*$, we write $|u|$ for the length of u. Under the concatenation operation of words, A^* forms a free monoid whose identity is the empty word which we denote by ε. If there exist $x, y \in A^*$ such that $u = xvy$, then we say that v is a *factor* of u.

A *two-sided infinite word* w is a function $w : \mathbb{Z} \to A$. A finite word u is a factor of w if u is a finite subsequence of w, that is, if there exists some $i \in \mathbb{Z}$ such that $u = w(i) \dots w(i + |u| - 1)$. For a positive integer p, we say that w has *period* p, or that w is *p-periodic*, if $w(i) = w(i+p)$ for all integers i. If w has period p for some p, then we call w *periodic*. If v is a nonempty

finite word, then we denote by $v^{\mathbb{Z}}$ the unique two-sided infinite word w such that w has period $|v|$ and $w(0)\dots w(|v|-1) = v$.

A word of finite length n over an alphabet A can be defined as a total function $w : \{0,\dots,n-1\} \to A$. Analogously a *partial word* of length n over A is a partial function $u : \{0,\dots,n-1\} \to A$. For $0 \le i \le n-1$, if $u(i)$ is defined, then we say that i belongs to the domain of u (denoted by $i \in D(u)$). Otherwise we say that i belongs to the *set of holes* of u (denoted by $i \in H(u)$). In cases where $H(u)$ is empty, we say that u is a *full word*.

If u is a partial word of length n over A, then the *companion* of u is the total function $u_\diamond : \{0,\dots,n-1\} \to A_\diamond$ defined by

$$u_\diamond(i) = \begin{cases} u(i) & \text{if } i \in D(u) \\ \diamond & \text{otherwise} \end{cases}$$

where $A_\diamond = A \cup \{\diamond\}$. Throughout this paper we identify a partial word with its companion. We reserve the term *letter* for elements of A. We will refer to an occurrence of the symbol $\diamond$ in a partial word as a *hole*.

Two partial words u and v of equal length are said to be *compatible*, denoted by $u \uparrow v$, if $u(i) = v(i)$ for every $i \in D(u) \cap D(v)$. If X is a set of partial words, then we use $\hat{X}$ to denote the set of all full words compatible with an element of X.

2.2 Unavoidable sets of partial words

The main concept of this paper is that of unavoidable set of partial words. We start by recalling the full word case.

Definition 1 *Let $X \subset A^*$.*

- *A two-sided infinite word w* avoids *X if no factor of w is an element of X.*
- *The set X is* unavoidable *if no two-sided infinite word avoids X, that is, X is unavoidable if every two-sided infinite word has a factor in X.*

Following are two useful facts giving alternative characterizations of a finite unavoidable set of full words X over A: (1) X is unavoidable if and only if there are only finitely many words in A^* with no member of X as a factor; and (2) X is unavoidable if and only if no periodic two-sided infinite word avoids X.

Unavoidable sets of partial words are defined as follows.

Definition 2 *Let $X \subset A_\diamond^*$.*

- *A two-sided infinite word w* avoids *X if no factor of w is a member of $\hat{X}$.*

- *The set X is* unavoidable *if no two-sided infinite word avoids X, that is, X is unavoidable if every two-sided infinite word has a factor in $\hat{X}$.*

Clearly if every member of X is full, then the concept of unavoidable set in Definition 2 is equivalent to the one in Definition 1. By the definition of $\hat{X}$, a two-sided infinite word w has a factor in $\hat{X}$ if and only if that factor is compatible with a member of X. Thus the two-sided infinite words which avoid $X \subset A_\diamond^\star$ are exactly those which avoid $\hat{X} \subset A^*$, and $X \subset A_\diamond^\star$ is unavoidable if and only if $\hat{X} \subset A^*$ is unavoidable.

3 Testing unavoidability

Testing the unavoidability of a finite set of full words X can be done in different ways. As said earlier, we can construct a finite automaton recognizing $A^* \setminus A^*XA^*$ and then check whether or not there is a loop in the automaton. Another approach is described as follows [8]: A set of words Y is obtained from a finite set of words X by an elementary derivation if

1. *Type 1 elementary derivation:* There exist words $x, y \in X$ such that x is a proper prefix of y, and $Y = X \setminus \{y\}$ (this will be denoted by $X \xrightarrow{1} Y$).

2. *Type 2 elementary derivation:* There exists a word $x = ya \in X$ with $a \in A$ such that, for each letter $b \in A$ there is a suffix z of y such that $zb \in X$, and $Y = (X \setminus \{x\}) \cup \{y\}$ (this will be denoted by $X \xrightarrow{2} Y$).

A *derivation* is a sequence of elementary derivations. We say that Y is derived from X if Y is obtained from X by a derivation. If Y is derived from X, then X is unavoidable if and only if Y is unavoidable.

Example 1 *The following sequence of elementary derivations shows that $\{\varepsilon\}$ is derived from $X = \{a\diamond\diamond a, b\diamond b\}$:*

$$
\begin{aligned}
\{a\diamond\diamond a, b\diamond b\} &\xrightarrow{2} \{aa\diamond a, aba, abba, b\diamond b\}\\
&\xrightarrow{2} \{aa\diamond a, ab\diamond, b\diamond b\}\\
&\xrightarrow{2} \{aa\diamond a, ab, b\diamond b\}\\
&\xrightarrow{2} \{aaa, aaba, ab, b\diamond b\}\\
&\xrightarrow{2} \{aa, aaba, ab, b\diamond b\}\\
&\xrightarrow{1} \{a\diamond, b\diamond b\}\\
&\xrightarrow{2} \{a, ab, b\diamond b\}\\
&\xrightarrow{2} \{a, b\diamond b\}\\
&\xrightarrow{2} \{a, b\diamond\}\\
&\xrightarrow{2} \{a, b\}\\
&\xrightarrow{2} \{\varepsilon, b\}\\
&\xrightarrow{1} \{\varepsilon\}.
\end{aligned}
$$

The notion of a derivation gives an algorithm to check whether a set is unavoidable: A finite set X is unavoidable if and only if there is a derivation from X to the set $\{\varepsilon\}$. The above derivation shows that $\{a\diamond\diamond b, b\diamond b\}$ is unavoidable.

These algorithms to determine if a finite set of full words is unavoidable, like the one just described, can be used to decide if a finite set of partial words X is unavoidable by determining the unavoidability of $\hat{X}$. However this incurs a dramatic loss in efficiency, as each partial word u in X can contribute as many as $\|A\|^{\|H(u)\|}$ elements to $\hat{X}$. In [4], the authors raised the following question: Is there an efficient algorithm to determine if a finite set of partial words is unavoidable? The following theorem shows that this problem is hard for any alphabet of size larger or equal to three.

Theorem 1 *The problem of determining if a finite set of partial words over a k-letter alphabet where $k \geq 3$ is unavoidable is NP-hard.*

Proof. The proof proceeds by reduction from the 3SAT problem that is known to be NP-complete (see [10]). In the 3SAT problem, we are given n binary variables $x_1, \ldots, x_n$ and m clauses that each contain three literals (a literal can be a variable or its negation), and we search a truth assignment for the variables such that each clause has at least one true literal.

Suppose that we are given a set of clauses x. We construct a set of partial words X over the alphabet $A = \{0, T, F\}$ such that X is avoidable if and only if the instance of 3SAT x has a solution. The first part of X is given by $\{0^{n+1}\}$. The second part is

$$\{0T0, 0T\diamond 0, \ldots, 0T\diamond^{n-2}0, 0F0, 0F\diamond 0, \ldots, 0F\diamond^{n-2}0\}.$$

The third part of X is as follows:

$$\{T\diamond^{n-1}T, T\diamond^{n-1}F, F\diamond^{n-1}T, F\diamond^{n-1}F\}.$$

Two-sided infinite words over $\{0, T, F\}$ that avoid these patterns are exactly words that are two-sided infinite concatenations of factors of the form $0^n v$ where $v \in \{T, F\}^n$. The remainder of the construction is such that the n consecutive nonzero symbols encode possible truth assignments for the variables in their natural order.

We add one pattern for every clause. These patterns are of length $n+2$, begin and end with a zero, and are otherwise entirely composed of $\diamond$'s except for the positions corresponding to the three variables of the clause, which we set to F if the clause contains the variable itself, or to T if the clause contains the negation of the variable.

For example, if there are five variables in our instance of 3SAT, the clause $x_1 \vee \bar{x_2} \vee x_4$ implies the presence of the partial word $0FT\diamond F\diamond 0$ in X. Such a set X has always a length polynomial in the number of clauses and the number of variables. We now prove that there is a solution to the instance x of 3SAT if and only if X is avoidable.

For the forward implication, suppose that there exists a satisfying truth assignment: $x_1, \ldots, x_n \in \{T, F\}$ satisfy the instance of 3SAT. We claim that the following two-sided infinite word

$$w = (0^n x_1 \ldots x_n)^{\mathbb{Z}}$$

avoids the set X. Indeed the first three parts of X are avoided because it is a two-sided infinite concatenation of factors of the form $0^n v$ where $v \in \{T, F\}^n$, and the last part is avoided because the only factors of length $n+2$ in w whose first and last letter is zero are equal to $0x_1 \ldots x_n 0$. They cannot be compatible with a partial word in the last part of X, since it would imply that the corresponding clause is not satisfied in the instance of 3SAT.

For the reverse implication, assume that X is avoidable. So there exists a two-sided infinite word w that avoids X. Due to the first three parts in X, this word has a factor $0w(i)\ldots w(i+n-1)0$ where $w(i), \ldots, w(i+n-1)$ are not equal to 0. Let us now assign to $x_j, 1 \leq j \leq n$, the value corresponding to $w(i+j-1)$. Suppose that this assignment does not satisfy the instance of 3SAT. This implies that the word $0w(i)\ldots w(i+n-1)0$ is compatible with the word in X corresponding to the violated clause, and we have a contradiction. This concludes the proof for $k = 3$. For $k \geq 3$, one just has to forbid the other letters in the construction of X. □

Since the case $k = 1$ is trivial, only the case $k = 2$ remains to be discussed. The following theorem adapts our reduction to the binary alphabet. Even if Theorem 2 implies Theorem 1, we have given the proof of both seperatly, because the following proof is more technical, and is better understood in view of the more intuitive Theorem 1.

Theorem 2 *The problem of determining if a finite set of partial words over a binary alphabet is unavoidable is NP-hard.*

Proof. We mirror the proof used for the ternary case, again proceeding by a reduction from the 3SAT problem. We use the words $111, 101$ to represent T and F respectively. Let a set of clauses x be given each containing three literals. We construct a set of partial words X over the binary alphabet $\{0, 1\}$ which will be avoidable precisely when the given instance of 3SAT x has a solution. Say x has m clauses and n variables $x_1, \ldots, x_n$.

There are two parts to our construction of X. First, we will make sure that a two-sided infinite word avoiding X is necessarily an element of the set W consisting of infinite concatenations of factors of the form $(000)^n v$ where $v \in \{111, 101\}^n$. The first triple of letters after the 0's will then correspond to an assignment to x_1, the second triple to x_2, and so forth. The second part of the construction carries out this correspondence.

The next steps all refer to the first part of this construction of X:

1. First, put $(000)^n 0$ in X to prevent a longer string of 0's in an avoider than desired.

2. Second, add 000100 and 000110 to X. Additionally, for each of the six binary words s of length three other than 101 or 111, add to X the words
$$000111s, 000111(\diamond\diamond\diamond)s, \ldots, 000111(\diamond\diamond\diamond)^{n-2}s$$
and also
$$000101s, 000101(\diamond\diamond\diamond)s, \ldots, 000101(\diamond\diamond\diamond)^{n-2}s$$
This will keep out triples that do not represent anything, and will also make sure n triples occur between our stretches of consecutive 0's.

3. Now, add all words of the form $s(\diamond\diamond\diamond)^{n-1}t$ where s, t are binary strings of length three other than 000. We must make some exceptions because of boundaries occurring in words of W against the consecutive zeros, that is, do **not** add in the strings
$$\begin{array}{lll} 001 & (\diamond\diamond\diamond)^{n-1} & 010 \\ 001 & (\diamond\diamond\diamond)^{n-1} & 110 \\ 010 & (\diamond\diamond\diamond)^{n-1} & 100 \\ 011 & (\diamond\diamond\diamond)^{n-1} & 100 \\ 010 & (\diamond\diamond\diamond)^{n-1} & 001 \\ 110 & (\diamond\diamond\diamond)^{n-1} & 001 \\ 100 & (\diamond\diamond\diamond)^{n-1} & 010 \\ 100 & (\diamond\diamond\diamond)^{n-1} & 011. \end{array}$$
This step is to guarantee that the stretches of 0's come in the right length.

4. We finally add 1001 to ensure the presence of at least a triple of zeros.

We claim that the set of avoiders of the construction so far is exactly W. It is easy enough to see that any member of W avoids X. We now show that an avoider w of X must be in W. We first see that a factor of the form 000 must occur in w. If w has a factor in the set $\{101, 111\}$, then it also has a factor 000 due to the words added to X in Step 3. Now remark that an infinite word that avoids $\{101, 111\}$ and $\{1001\}$ must have three consecutive zeros.
Given an occurrence of 000, by Step 1 we know we will eventually find a final 0 in the sequence of consecutive 0's it is part of. By Step 2, only 101 or 111 can follow in the next triple of positions. In fact, Step 2 guarantees that there is a word in $\{101, 111\}^n$ following that final 0. By Step 3, there is a factor $(000)^n$ immediately following that. And so on.

The rest of the proof goes precisely as in the ternary case, by finishing our construction of X depending on the actual clauses in x, using 111 (respectively, 101) in place of T (respectively, F). □

4 Conclusion

In this paper, we have shown that testing avoidability on sets of partial words is much harder to handle than the similar problem for full words. An interesting open question is whether the decision problem of the avoidability of a set of partial words is in NP. A similar (stronger) question is this one: For any set of partial words X, does there always exist a two-sided infinite periodic word that avoids X, whose period is polynomial in the size of X?

References

[1] Aho, A.V., Corasick, M.J.: Efficient string machines, an aid to bibliographic research. Comm. ACM **18** (1975) 333–340

[2] Berstel, J., Boasson, L.: Partial words and a theorem of Fine and Wilf. Theoret. Comput. Sci. **218** (1999) 135–141

[3] Blanchet-Sadri, F.: Algorithmic Combinatorics on Partial Words. Chapman & Hall/CRC Press (2007)

[4] Blanchet-Sadri, F., Brownstein, N.C., Palumbo, J.: Two element unavoidable sets of partial words. In Harju, T., Karhumäki, J., Lepistö, A. (eds.): DLT 2007, LNCS **4588** (Springer-Verlag, Berlin, 2007) 96–107 `www.uncg.edu/mat/research/unavoidablesets`

[5] Blanchet-Sadri, F., Kalcic, A., Weyand, T.: Unavoidable sets of partial words of size three. Preprint `www.uncg.edu/cmp/research/unavoidablesets2`

[6] Blondel, V.D., Jungers, R., Protasov, V.: On the complexity of computing the capacity of codes that avoid forbidden difference patterns. IEEE Trans. Information Theory **52:**11 (2006) 5122–5127

[7] Choffrut, C., Culik II, K.: On extendibility of unavoidable sets. Discrete Appl. Math. **9** (1984) 125–137

[8] Choffrut, C., Karhumäki, J.: Combinatorics of Words. In Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages. Vol. 1. Springer-Verlag, Berlin (1997) 329–438

[9] Ehrenfeucht, A., Haussler, D., Rozenberg, G.: On regularity of context-free languages. Theoret. Comput. Sci. **27** (1983) 311–322

[10] Garey, M.R., Johnson, D.S.: Computers and Intractability - A Guide to the Theory of NP-Completeness. Freeman (1979)

[11] Leupold, P.: Partial words for DNA coding. LNCS **3384** (Springer-Verlag, Berlin, 2005) 224–234

[12] Lothaire, M.: Algebraic Combinatorics on Words. Cambridge University Press, Cambridge (2002)

[13] Rosaz, L.: Unavoidable languages, cuts and innocent sets of words. RAIRO Theoret. Inform. Appl. **29** (1995) 339–382

[14] Rosaz, L.: Inventories of unavoidable languages and the word-extension conjecture. Theoret. Comput. Sci. **201** (1998) 151–170