

Testing Primitivity on Partial Words

F. Blanchet-Sadri* and Arundhati R. Anavekar

Department of Mathematical Sciences, University of North Carolina,
P.O. Box 26170, Greensboro, NC 27412-6170, USA,
blanchet@uncg.edu,
<http://www.uncg.edu/mat/~blanchet>

Abstract. *Primitive words*, or strings over a finite alphabet that cannot be written as a power of another string, play an important role in numerous research areas including formal language theory, coding theory, and combinatorics on words. Testing whether or not a word is primitive can be done in linear time in the length of the word. Indeed, a word is primitive if and only if it is not an inside factor of its square. In this paper, we describe a linear time algorithm to test primitivity on *partial words* which are strings that may contain a number of “do not know” symbols. Our algorithm is based on the combinatorial result that under some condition, a partial word is primitive if and only if it is not *compatible* with an inside factor of its square. The concept of *special*, related to commutativity on partial words, is foundational in the design of our algorithm. A World Wide Web server interface at <http://www.uncg.edu/mat/primitive/> has been established for automated use of the program. . . .

1 Introduction

Words, or strings of symbols over a finite alphabet, are natural objects in several research areas including automata and formal language theory, coding theory, and theory of algorithms. Molecular biology has stimulated considerable interest in the study of *partial words* which are strings that may contain a number of “do not know” symbols or “holes”. The motivation behind the notion of a partial word is the comparison of genes. Alignment of two such strings can be viewed as a construction of two partial words that are said to be *compatible* in a sense that will be discussed in Section 2. While a word can be described by a total function, a partial word can be described by a partial function. More precisely, a partial word of length n over a finite alphabet A is a partial function from $\{0, \dots, n-1\}$ into A . Elements of $\{0, \dots, n-1\}$ without an image are called holes (a word is just a partial word without holes). Research in combinatorics on partial words is underway [1,2,3,4,5,6,7,8,11] and has the potential for impacts in numerous areas, notably in molecular biology, nano-technology, and DNA

* This material is based upon work supported by the National Science Foundation under Grant CCF-0207673. We thank the referees of preliminary versions of this paper for their very valuable comments and suggestions.

computing [13]. Partial words are currently being considered, in particular, for finding good encodings for DNA computations.

Primitive words, those that cannot be written as a power of another word, play an important role in combinatorics on words. A word u is *primitive* if there exists no word v such that $u = v^n$ with $n \geq 2$. A natural algorithmic problem is “How can we decide efficiently whether a given word is primitive?”. The problem has a brute force quadratic solution: divide the input word into two parts and check whether the right part is a power of the left part. But how can we obtain a faster solution to the problem? Fast algorithms for testing primitivity of words can be based on the combinatorial result that a word u is primitive if and only if u is not an inside factor of its square uu , that is, $uu = xuy$ implies that x or y is empty [9]. Indeed, any linear time string matching algorithm can be used to test whether the string u is an inside factor of uu . If the answer is no, then the primitiveness of u has been verified [10].

Primitive partial words were defined in [4]. A partial word u is *primitive* if there exists no word v such that $u \subset v^n$ with $n \geq 2$ (the concept of containment, denoted by \subset , is discussed in Section 2). A partial word u with one hole was shown to be primitive if and only if the compatibility of uu with xuy for some partial words x, y implies that x or y is empty. A linear time algorithm for testing primitivity of partial words with one hole can be based on this combinatorial result which found a nice application in [5]. There, Blanchet-Sadri and Chriscoe extended to partial words with one hole the well known result of Guibas and Odlyzko [12] which states that the sets of periods of words are independent of the alphabet size. As a consequence of their constructive proof, Blanchet-Sadri and Chriscoe obtained a linear time algorithm which, given a partial word with one hole, computes a binary one with the same sets of periods and the same sets of weak periods. The algorithm required primitivity testing of partial words with one hole (see <http://www.uncg.edu/mat/AlgBin/>).

In this paper, we investigate primitivity testing for partial words with an arbitrary number of holes. The partial word $u = ab\diamond bbb\diamond b$ (where the \diamond 's denote holes) illustrates the fact that the above mentioned combinatorial property does not hold in general for primitive partial words with more than one hole (see Example 2). However, we show that if u is a primitive partial word with more than one hole such that uu and xuy are compatible for some non-empty partial words x and y , then u belongs to a *special* class of partial words (see Proposition 2). This concept of special partial word, defined in Section 3, relates to commutativity and is foundational in the design of our linear time algorithm for testing primitivity on partial words which is described in Section 4.

2 Preliminaries

We first review basic concepts on words and partial words. Let A be a non-empty finite set, or an *alphabet*. A *string* or *word* u over A is a finite concatenation of symbols from A . The number of symbols in u , or *length* of u , is denoted by $|u|$. We assume that, for every word, the first letter is at position 0. For any word u ,

$u[i..j)$ is the *subword* or *factor* of u that starts at position i and ends at position $j-1$ (it is called *proper* if $0 < i$ or $j < |u|$). In particular, $u[0..j)$ is the *prefix* of u that ends at position $j-1$ and $u[i..|u|)$ is the *suffix* of u that begins at position i . The subword $u[i..j)$ is the empty word if $i \geq j$ (the empty word is denoted by ϵ). The set of all words over A of finite length (greater than or equal to 0) is denoted by A^* . It is a monoid under the associative operation of concatenation or product of words (ϵ serves as identity) and is referred to as the *free monoid* generated by A . Similarly, the set of all non-empty words over A is denoted by A^+ . It is a semigroup under the operation of concatenation of words and is referred to as the *free semigroup* generated by A .

For a word u , the powers of u are defined inductively by $u^0 = \epsilon$ and, for any $n \geq 1$, $u^n = uu^{n-1}$. A word u is *primitive* if there exists no word v such that $u = v^n$ with $n \geq 2$. If u is a non-empty word, then there exists a unique primitive word v and a unique positive integer n such that $u = v^n$.

A word of length n over A can be defined by a total function $u : \{0, \dots, n-1\} \rightarrow A$ and is usually represented as $u = a_0a_1 \dots a_{n-1}$ with $a_i \in A$. A partial word u of length n over A is a partial function $u : \{0, \dots, n-1\} \rightarrow A$. For $0 \leq i < n$, 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 $i \in H(u)$). A word over A is a partial word over A with an empty set of holes (we will sometimes refer to words as *full* words). The length of u is denoted by $|u|$.

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

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

The bijectivity of the map $u \mapsto u_\diamond$ allows us to define for partial words concepts such as concatenation and powers in a trivial way. The symbol \diamond is viewed as a “do not know” symbol and not as a “do not care” symbol as in pattern matching. The word $u_\diamond = \diamond ba \diamond abb$ is the companion of the partial word u of length 7 where $D(u) = \{1, 2, 4, 5, 6\}$ and $H(u) = \{0, 3\}$. In the sequel, for convenience, we will consider a partial word over A as a word over the enlarged alphabet $A \cup \{\diamond\}$, where the additional symbol \diamond plays a special role. Thus, we say for instance “the partial word $\diamond ba \diamond abb$ ” instead of “the partial word with companion $\diamond ba \diamond abb$ ”.

A *period* of a partial word u over A is a positive integer p such that $u(i) = u(j)$ whenever $i, j \in D(u)$ and $i \equiv j \pmod{p}$. In such a case, we call u *p-periodic*. Similarly, a *weak period* of u is a positive integer p such that $u(i) = u(i+p)$ whenever $i, i+p \in D(u)$. In such a case, we call u *weakly p-periodic*. The partial word with companion $ab \diamond bcb$ is weakly 2-periodic but is not 2-periodic (this is because a occurs in position 0 while c occurs in position 4). The latter shows a difference between partial words and words since every weakly p -periodic full word is p -periodic. Another difference worth noting is the fact that even if the length of a partial word u is a multiple of a weak period of u , then u is not necessarily a power of a shorter partial word.

If u and v are two partial words “of equal length”, then u is said to be contained in v , denoted by $u \subset v$, if all elements in $D(u)$ are in $D(v)$ and $u(i) = v(i)$ for all $i \in D(u)$. The partial words u and v are called *compatible*, denoted by $u \uparrow v$, if there exists a partial word w such that $u \subset w$ and $v \subset w$. For example, $u = aba\circ a$ and $v = a\circ\circ ba$ are two partial words that are compatible ($w = ababa$).

We can extend the notion of a word being primitive to a partial word being primitive as follows: A partial word u is *primitive* if there exists no word v such that $u \subset v^n$ with $n \geq 2$. Note that if v is primitive and $v \subset u$, then u is primitive as well. If u is a non-empty partial word, then there exists a primitive word v and a positive integer n such that $u \subset v^n$. Uniqueness does not hold for partial words. The partial word u where $u_\circ = \circ a$ serves as a counterexample ($u \subset a^2$ and $u \subset ba$ for distinct letters a, b).

The following rules are useful for computing with partial words [1].

Multiplication: If $u \uparrow v$ and $x \uparrow y$, then $ux \uparrow vy$.

Simplification: If $ux \uparrow vy$ and $|u| = |v|$, then $u \uparrow v$ and $x \uparrow y$.

Weakening: If $u \uparrow v$ and $w \subset u$, then $w \uparrow v$.

The following lemma holds [1].

Lemma 1. *Let u, v, x, y be partial words such that $ux \uparrow vy$.*

1. *If $|u| \geq |v|$, then there exist partial words w, z such that $u = wz$, $v \uparrow w$, and $y \uparrow zx$.*
2. *If $|u| \leq |v|$, then there exist partial words w, z such that $v = wz$, $u \uparrow w$, and $x \uparrow zy$.*

3 Commutativity on Partial Words

It is well known that two non-empty words u and v commute if and only if there exists a word w such that $u = w^m$ and $v = w^n$ for some integers m, n . When dealing with two non-empty partial words u and v , the existence of a word w satisfying $u \subset w^m$ and $v \subset w^n$ for some integers m, n certainly implies $uv \uparrow vu$. The converse is not true in general (take for example $u = \circ bb$ and $v = abb\circ$). However, if uv has at most one hole, then the following result holds [1].

Lemma 2. *Let u and v be non-empty partial words such that uv has at most one hole. If $uv \uparrow vu$, then there exists a word w such that $u \subset w^m$ and $v \subset w^n$ for some integers m, n .*

We now describe an extension of Lemma 2 when uv has at least two holes. Without loss of generality, we may assume that $|u| \leq |v|$. Our extension is based on the concept of uv being (k, ℓ) -special where k, ℓ denote the lengths of u, v respectively. For $0 \leq i < k + \ell$, we define the sequence of i relative to k, ℓ as $seq_{k, \ell}(i) = (i_0, i_1, i_2, \dots, i_n, i_{n+1})$ where $i_0 = i = i_{n+1}$ and where

For $1 \leq j \leq n$, $i_j \neq i$,

For $1 \leq j \leq n + 1$, i_j is defined as

$$i_j = \begin{cases} i_{j-1} + k & \text{if } i_{j-1} < \ell, \\ i_{j-1} - \ell & \text{otherwise.} \end{cases}$$

Note that $seq_{k,\ell}(i)$ is stopped at the first occurrence of i , which defines $n + 1$. For example, if $k = 4$ and $\ell = 10$, then $seq_{4,10}(1) = (1, 5, 9, 13, 3, 7, 11, 1)$. Now, the concept of (k, ℓ) -special partial word is defined as follows.

Definition 1. Let k, ℓ be positive integers satisfying $k \leq \ell$ and let w be a partial word of length $k + \ell$. We say that w is (k, ℓ) -special if there exists $0 \leq i < k$ such that $seq_{k,\ell}(i) = (i_0, i_1, i_2, \dots, i_n, i_{n+1})$ contains two positions that are holes of w while $w(i_0)w(i_1)w(i_2) \dots w(i_{n+1})$ is not 1-periodic.

Example 1. If $k = 4$ and $\ell = 10$, then the partial word $u = a \diamond baab \diamond aabaa \diamond \diamond$ is $(4, 10)$ -special since $seq_{4,10}(0)$ contains the positions 6 and 12 which are in $H(u) = \{1, 6, 12, 13\}$ while $u(0)u(4)u(8)u(12)u(2)u(6)u(10)u(0) = aaa \diamond b \diamond aa$ is not 1-periodic. However, the partial word $v = \diamond babab \diamond babab \diamond b$ is not $(4, 10)$ -special.

Remark 1. The above defined concept of (k, ℓ) -special partial word is different from an earlier concept of $\{k, \ell\}$ -special partial word that was introduced in [8]. There, w is $\{k, \ell\}$ -special if there exists $0 \leq i < k$ such that $seq_{k,\ell}(i)$ satisfies the condition of Definition 1 or the condition that it contains two consecutive positions that are holes of w . This extra condition was needed to prove the following combinatorial property: If w is a partial word and u, v are full words such that $w \subset uv$ and $w \subset vu$ and w is non- $\{|u|, |v|\}$ -special, then $uv = vu$. For instance, if $k = 3$ and $\ell = 6$, then the partial word $w = ab \diamond bc \diamond bc$ is $\{3, 6\}$ -special since $seq_{3,6}(0) = (0, 3, 6, 0)$ contains the consecutive positions 3 and 6 which are in $H(w) = \{2, 3, 6\}$ (but w is not $(3, 6)$ -special). Here, by letting $u = abc$ and $v = abcbbc$, we have $w \subset uv$ and $w \subset vu$ and $uv \neq vu$.

Remark 2. For the counterexample to Lemma 2 where $u = \diamond bb$ and $v = abb \diamond$, we have $seq_{3,4}(0) = (0, 3, 6, 2, 5, 1, 4, 0)$ which contains the holes 0, 6 of uv while

$$(uv)(0)(uv)(3)(uv)(6)(uv)(2)(uv)(5)(uv)(1)(uv)(4)(uv)(0) = \diamond a \diamond bbb \diamond$$

is not 1-periodic showing that uv is $(3, 4)$ -special.

We now prove our extension of Lemma 2.

Theorem 1. Let u, v be non-empty partial words such that $|u| \leq |v|$. If $uv \uparrow vu$ and uv is not $(|u|, |v|)$ -special, then there exists a word w such that $u \subset w^m$ and $v \subset w^n$ for some integers m, n .

Proof. Since $uv \uparrow vu$, there exists a word x such that $uv \subset x$ and $vu \subset x$. Put $|u| = k$ and $|v| = \ell$. The proof is split into three cases that refer to a given position i of x . Case 1 refers to $0 \leq i < k$, Case 2 to $k \leq i < \ell$, and Case 3 to $\ell \leq i < \ell + k$ (Cases 1 and 3 are symmetric as is seen by putting $i = \ell + j$ where $0 \leq j < k$). The following diagram pictures the containments $uv \subset x$ and $vu \subset x$:

$$\begin{array}{l}
(uv)_\diamond \mid u_\diamond(0) \dots u_\diamond(k-1) \mid v_\diamond(0) \dots v_\diamond(\ell-k-1) \mid v_\diamond(\ell-k) \dots v_\diamond(\ell-1) \\
(vu)_\diamond \mid v_\diamond(0) \dots v_\diamond(k-1) \mid v_\diamond(k) \dots v_\diamond(\ell-1) \mid u_\diamond(0) \dots u_\diamond(k-1) \\
x \mid x(0) \dots x(k-1) \mid x(k) \dots x(\ell-1) \mid x(\ell) \dots x(\ell+k-1)
\end{array}$$

Put $\ell = mk + r$ where $0 \leq r < k$. We first assume that $r = 0$.

Case 1: Since $uv \subset x$ and $vu \subset x$, we have

$$\begin{array}{l}
u_\diamond(i) \subset x(i) \text{ and } v_\diamond(i) \subset x(i), \\
v_\diamond(i) \subset x(i+k) \text{ and } v_\diamond(i+k) \subset x(i+k), \\
v_\diamond(i+k) \subset x(i+2k) \text{ and } v_\diamond(i+2k) \subset x(i+2k), \\
v_\diamond(i+2k) \subset x(i+3k) \text{ and } v_\diamond(i+3k) \subset x(i+3k), \\
\vdots \\
v_\diamond(i+(m-2)k) \subset x(i+(m-1)k) \text{ and } v_\diamond(i+(m-1)k) \subset x(i+(m-1)k), \\
v_\diamond(i+(m-1)k) \subset x(i+mk) \text{ and } u_\diamond(i) \subset x(i+mk).
\end{array}$$

Put $u_\diamond(i)v_\diamond(i)v_\diamond(i+k) \dots v_\diamond(i+(m-1)k)u_\diamond(i) = y_i$. We claim that the partial word y_i is 1-periodic, say with letter a_i in $A \cup \{\diamond\}$. The claim easily follows from the above list of containments in case y_i has less than two holes.

For the case where y_i has at least two holes, the claim follows since uv is not (k, ℓ) -special. By letting $w = a_0 a_1 \dots a_{k-1}$, we get $u \subset w$ and $v \subset w^m$ as desired.

Case 2: Put $i = nk + s$ where $0 \leq s < k$. Since $uv \subset x$ and $vu \subset x$, we have

$$\begin{array}{l}
v_\diamond(nk+s) \subset x((n+1)k+s) \text{ and } v_\diamond((n+1)k+s) \subset x((n+1)k+s), \\
v_\diamond((n+1)k+s) \subset x((n+2)k+s) \text{ and } v_\diamond((n+2)k+s) \subset x((n+2)k+s), \\
v_\diamond((n+2)k+s) \subset x((n+3)k+s) \text{ and } v_\diamond((n+3)k+s) \subset x((n+3)k+s), \\
\vdots \\
v_\diamond((m-2)k+s) \subset x((m-1)k+s) \text{ and } v_\diamond((m-1)k+s) \subset x((m-1)k+s), \\
v_\diamond((m-1)k+s) \subset x(mk+s) \text{ and } u_\diamond(s) \subset x(mk+s), \\
u_\diamond(s) \subset x(s) \text{ and } v_\diamond(s) \subset x(s), \\
v_\diamond(s) \subset x(k+s) \text{ and } v_\diamond(k+s) \subset x(k+s), \\
v_\diamond(k+s) \subset x(2k+s) \text{ and } v_\diamond(2k+s) \subset x(2k+s), \\
\vdots \\
v_\diamond((n-2)k+s) \subset x((n-1)k+s) \text{ and } v_\diamond((n-1)k+s) \subset x((n-1)k+s), \\
v_\diamond((n-1)k+s) \subset x(nk+s) \text{ and } v_\diamond(nk+s) \subset x(nk+s).
\end{array}$$

Put $u_\diamond(s)v_\diamond(s)v_\diamond(k+s) \dots v_\diamond((m-1)k+s)u_\diamond(s) = y_s$. As in Case 1. the partial word y_s is 1-periodic, say with letter a_s in $A \cup \{\diamond\}$. By letting $w = a_0 a_1 \dots a_{k-1}$, we get $u \subset w$ and $v \subset w^m$ as desired.

We now assume that $r > 0$.

Case 1: We consider the cases where $i < r$ and $i \geq r$. If $i < r$, then

$$\begin{array}{l}
u_\diamond(i) \subset x(i) \text{ and } v_\diamond(i) \subset x(i), \\
v_\diamond(i) \subset x(i+k) \text{ and } v_\diamond(i+k) \subset x(i+k), \\
v_\diamond(i+k) \subset x(i+2k) \text{ and } v_\diamond(i+2k) \subset x(i+2k), \\
v_\diamond(i+2k) \subset x(i+3k) \text{ and } v_\diamond(i+3k) \subset x(i+3k), \\
\vdots \\
v_\diamond(i+(m-1)k) \subset x(i+mk) \text{ and } v_\diamond(i+mk) \subset x(i+mk),
\end{array}$$

$$\begin{aligned} v_\diamond(i+mk) &\subset x(i+(m+1)k) \text{ and } u_\diamond(i+k-r) \subset x(i+(m+1)k), \\ u_\diamond(i+k-r) &\subset x(i+k-r) \text{ and } v_\diamond(i+k-r) \subset x(i+k-r), \\ v_\diamond(i+k-r) &\subset x(i+2k-r) \text{ and } v_\diamond(i+2k-r) \subset x(i+2k-r), \end{aligned}$$

$$\vdots$$

If $i \geq r$, then

$$\begin{aligned} u_\diamond(i) &\subset x(i) \text{ and } v_\diamond(i) \subset x(i), \\ v_\diamond(i) &\subset x(i+k) \text{ and } v_\diamond(i+k) \subset x(i+k), \\ v_\diamond(i+k) &\subset x(i+2k) \text{ and } v_\diamond(i+2k) \subset x(i+2k), \\ v_\diamond(i+2k) &\subset x(i+3k) \text{ and } v_\diamond(i+3k) \subset x(i+3k), \end{aligned}$$

$$\vdots$$

$$\begin{aligned} v_\diamond(i+(m-2)k) &\subset x(i+(m-1)k) \text{ and } v_\diamond(i+(m-1)k) \subset x(i+(m-1)k), \\ v_\diamond(i+(m-1)k) &\subset x(i+mk) \text{ and } u_\diamond(i-r) \subset x(i+mk), \\ u_\diamond(i-r) &\subset x(i-r) \text{ and } v_\diamond(i-r) \subset x(i-r), \\ v_\diamond(i-r) &\subset x(i+k-r) \text{ and } v_\diamond(i+k-r) \subset x(i+k-r), \end{aligned}$$

$$\vdots$$

If $i < r$, then let $u_\diamond(i)v_\diamond(i)v_\diamond(i+k) \dots v_\diamond(i+mk)u_\diamond(i+k-r) \dots u_\diamond(i) = y_i$, and if $i \geq r$, then let $u_\diamond(i)v_\diamond(i)v_\diamond(i+k) \dots v_\diamond(i+(m-1)k)u_\diamond(i-r) \dots u_\diamond(i) = y_i$. In either case, we claim that y_i is 1-periodic, say with letter a_i in $A \cup \{\diamond\}$. The claim follows from the above containments in case y_i has less than two holes. For the case where y_i has at least two holes, the claim follows since uv is not (k, ℓ) -special. It turns out that $a_j = a_{j+r} = \dots$ for $0 \leq j < r$. Let $w = a_0 a_1 \dots a_{r-1}$. If r divides k , then $u \subset w^{k/r}$ and $v \subset w^{(mk/r)+1}$. If r does not divide k , then w is 1-periodic with letter a say. In this case, $u \subset a^k$ and $v \subset a^\ell$.

Case 2: Put $i = nk + s$ where $0 \leq s < k$. We have

$$\begin{aligned} v_\diamond(nk+s) &\subset x((n+1)k+s) \text{ and } v_\diamond((n+1)k+s) \subset x((n+1)k+s), \\ v_\diamond((n+1)k+s) &\subset x((n+2)k+s) \text{ and } v_\diamond((n+2)k+s) \subset x((n+2)k+s), \\ v_\diamond((n+2)k+s) &\subset x((n+3)k+s) \text{ and } v_\diamond((n+3)k+s) \subset x((n+3)k+s), \end{aligned}$$

$$\vdots$$

$$v_\diamond((m-2)k+s) \subset x((m-1)k+s) \text{ and } v_\diamond((m-1)k+s) \subset x((m-1)k+s).$$

If $s < r$, then we also get

$$\begin{aligned} v_\diamond((m-1)k+s) &\subset x(mk+s) \text{ and } v_\diamond(mk+s) \subset x(mk+s), \\ v_\diamond(mk+s) &\subset x((m+1)k+s) \text{ and } u_\diamond(k-r+s) \subset x((m+1)k+s), \\ u_\diamond(k-r+s) &\subset x(k-r+s) \text{ and } v_\diamond(k-r+s) \subset x(k-r+s), \\ v_\diamond(k-r+s) &\subset x(2k-r+s) \text{ and } v_\diamond(2k-r+s) \subset x(2k-r+s), \end{aligned}$$

$$\vdots$$

$$v_\diamond((n-1)k+s) \subset x(nk+s) \text{ and } v_\diamond(nk+s) \subset x(nk+s),$$

The result follows similarly as in Case 1 ($r > 0$).

If $s \geq r$, then we also get

$$\begin{aligned} v_\diamond((m-1)k+s) &\subset x(mk+s) \text{ and } u_\diamond(s-r) \subset x(mk+s), \\ u_\diamond(s-r) &\subset x(s-r) \text{ and } v_\diamond(s-r) \subset x(s-r), \\ v_\diamond(s-r) &\subset x(k-r+s) \text{ and } v_\diamond(k-r+s) \subset x(k-r+s), \end{aligned}$$

$$\vdots$$

$$v_\diamond((n-1)k+s) \subset x(nk+s) \text{ and } v_\diamond(nk+s) \subset x(nk+s),$$

Again, the result follows similarly as in Case 1 ($r > 0$).

□

4 Our Algorithm

The property of being primitive is testable on a word of n symbols in $O(n)$ time [10]. A linear time algorithm can be based on the combinatorial property that no primitive word u can be an inside factor of uu . Indeed, u is primitive if and only if u is not a proper factor of uu , that is, $uu = xuy$ implies $x = \epsilon$ or $y = \epsilon$. The following proposition shows that the property also holds for partial words with one hole.

Proposition 1. *Let u be a partial word with one hole. Then u is primitive if and only if $uu \uparrow xuy$ for some partial words x, y implies $x = \epsilon$ or $y = \epsilon$.*

Proof. Assume that u is primitive and that $uu \uparrow xuy$ for some non-empty partial words x, y . Since $|x| < |u|$, by Lemma 1, there exist non-empty partial words z, v such that $u = zv$, $z \uparrow x$, and $vu \uparrow uy$. Then $zvzv \uparrow xzvy$ yields $vz \uparrow zv$ by simplification. By Lemma 2, v and z are contained in powers of a common word, a contradiction with the fact that u is primitive.

Now, assume that $uu \uparrow xuy$ for some partial words x, y implies $x = \epsilon$ or $y = \epsilon$. Suppose to the contrary that u is not primitive. Then there exists a non-empty word v and an integer $n \geq 2$ such that $u \subset v^n$. But then $uu \uparrow v^{n-1}uv$, and using our assumption we get $v^{n-1} = \epsilon$ or $v = \epsilon$, a contradiction. \square

In the case of partial words with at least two holes, the following holds.

Proposition 2. *Let u be a partial word with at least two holes.*

1. *If $uu \uparrow xuy$ for some partial words x, y implies $x = \epsilon$ or $y = \epsilon$, then u is primitive.*
2. *If $uu \uparrow xuy$ for some non-empty partial words x and y satisfying $|x| \leq |y|$, then the following hold:*
 - (a) *If $|x| = |y|$, then u is not primitive.*
 - (b) *If u is not $(|x|, |y|)$ -special, then u is not primitive (it is contained in a power of a word of length $|x|$).*
 - (c) *If u is $(|x|, |y|)$ -special, then u is not contained in a power of a word of length $|x|$.*

Proof. Statement 1 follows as in Proposition 1. For Statement 2, assume that $uu \uparrow xuy$ for some non-empty partial words x, y . Let u_1 be the prefix of length $|x|$ of u and u_2 be the suffix of length $|y|$ of u ($u = u_1u_2$). The compatibility relation $u_1u_2u_1u_2 \uparrow xu_1u_2y$ yields $u_1u_2 \uparrow u_2u_1$. For Statement 2(a), since $|x| = |y|$, $u_1u_2 \uparrow u_2u_1$ implies $u_1 \uparrow u_2$. By definition, there exists a partial word w such that $u_1 \subset w$ and $u_2 \subset w$. We get $u = u_1u_2 \subset w^2$, and the statement follows. For Statement 2(b), since $u = u_1u_2$ is not $(|u_1|, |u_2|)$ -special, by Theorem 1, u_1 and u_2 are contained in powers of a common word, showing that u is not primitive. Here, for $0 \leq i < |x|$, $seq_{|x|, |y|}(i)$ is 1-periodic with letter a_i for some $a_i \in A \cup \{\diamond\}$. We conclude that u is contained in a power of $a_0a_1 \dots a_{|x|-1}$. For Statement 2(c), put $|y| = m|x| + r$ where $0 \leq r < |x|$. If $r > 0$, then u is obviously not contained in a power of a word of length $|x|$. And if $r = 0$, then there exists

$0 \leq i < |x|$ such that $seq_{|x|,|y|}(i) = (i, i + |x|, i + 2|x|, \dots, i + m|x|, i)$ contains two positions that are holes of u while $u_{\diamond}(i)u_{\diamond}(i + |x|)u_{\diamond}(i + 2|x|) \dots u_{\diamond}(i + m|x|)u_{\diamond}(i)$ is not 1-periodic. \square

Example 2. This example illustrates Proposition 2(2(c)). The primitive partial word $u = ab\triangleright bbb\triangleright b$ is compatible with an inside factor of its square uu as illustrated in the following diagram:

$$\begin{array}{c} a b \triangleright b b b \triangleright b a b \triangleright b b b \triangleright b \\ a b \triangleright b b b \triangleright b \end{array}$$

Here u is $(2, 6)$ -special since $seq_{2,6}(0) = (0, 2, 4, 6, 0)$ contains the holes 2 and 6 while $u(0)u(2)u(4)u(6)u(0) = a\triangleright b\triangleright a$ is not 1-periodic. Here, u is not contained in a power of a word of length 2.

We now give an algorithm for testing whether a partial word is primitive.

Algorithm Primitivity Testing

```

input: partial word  $u$ 
output: primitive (if  $u$  is primitive) and non-primitive (otherwise)
 $U \leftarrow uu$ 
count  $\leftarrow \|H(u)\|$ 
if count  $< 2$  then
  check compatibility of  $u$  with a substring of  $U[1..2|u| - 1]$ 
  if successful then
    return non-primitive
  else
    return primitive
else
   $k \leftarrow 1$  and  $\ell \leftarrow |u| - 1$ 
  while  $k \leq \ell$  do
    check compatibility of  $u$  with  $U[k..k + |u|]$ 
    if successful then
      if  $u$  is  $(k, \ell)$ -special and  $k < \ell$  then
         $k \leftarrow k + 1$  and  $\ell \leftarrow \ell - 1$ 
      if  $u$  is not  $(k, \ell)$ -special or  $k = \ell$  then
        return non-primitive
    else
       $k \leftarrow k + 1$  and  $\ell \leftarrow \ell - 1$ 
  return primitive

```

Remark 3. Note that if u is primitive, then its reversal $rev(u)$ defined by $(rev(u))_{\diamond} = rev(u_{\diamond})$ (where $rev(u_{\diamond})$ is u_{\diamond} written backwards) is also primitive. This fact justifies the while loop being for $k \leq \ell$.

The following example illustrates our algorithm.

Example 3. Consider the partial word $u = a \diamond ab a \diamond$ where $D(u) = \{0, 3, 4, 5\}$ and $H(u) = \{1, 2, 6\}$. The algorithm proceeds as follows:

- $k = 1, \ell = 6$: Compatibility of u with $U[1..8]$ is non-successful.
 $k = 2, \ell = 5$: Compatibility of u with $U[2..9]$ is successful.

$$\begin{array}{c} a \diamond a b a \diamond a \diamond a b a \diamond \\ a \diamond a b a \diamond \end{array}$$

Here, the partial word u is $(2, 5)$ -special.

- $k = 3, \ell = 4$: Compatibility of u with $U[3..10]$ is non-successful.

Thus the partial word u is primitive.

Now, consider the partial word $u = ab \diamond bc \diamond bc$ where $D(u) = \{0, 1, 4, 5, 7, 8\}$ and $H(u) = \{2, 3, 6\}$. The algorithm proceeds as follows:

- $k = 1, \ell = 8$: Compatibility of u with $U[1..10]$ is non-successful.
 $k = 2, \ell = 7$: Compatibility of u with $U[2..11]$ is non-successful.
 $k = 3, \ell = 6$: Compatibility of u with $U[3..12]$ is successful.

$$\begin{array}{c} a b \diamond b c \diamond b c a b \diamond b c \diamond b c \\ a b \diamond b c \diamond b c \end{array}$$

Here, the partial word u is not $(3, 6)$ -special and is thus non-primitive ($u \subset (abc)^3$).

In conclusion, the following theorem holds.

Theorem 2. *The property of being primitive is testable on a partial word of length n in $O(n)$ time.*

Proof. The correctness of our algorithm follows from Propositions 1 and 2. To see that primitivity can be tested in linear time in the length of a given partial word u , any linear time pattern matching algorithm, refer for instance to Reference [10], can be easily adapted to test whether the string u is compatible with an inside substring of uu . The algorithm finds the leftmost occurrence, if any, of a factor of uu , $U[k..k+|u|]$, compatible with u . For a full word u , the comparisons done are of the type $a \stackrel{?}{=} b$, for letters a and b in the alphabet A . For a partial word u , we can overload the comparison operator in $a \stackrel{?}{=} b$ to return all comparisons of the special symbol \diamond with any letter a or b as true. (For example, both $\diamond \stackrel{?}{=} b$ and $a \stackrel{?}{=} \diamond$ returns true for all letters a and b in A , while $a \stackrel{?}{=} b$ only returns true if both a and b are the same symbol.) Overloading the operator does not change the time complexity of the algorithm any more than by a constant factor. Thus, the discovery of the leftmost occurrence, if any, of a substring $U[k..k+|u|]$ compatible with u can be performed in linear time. This part of the algorithm needs to be altered slightly to handle partial words with at least two holes.

Fixing $k > 0$, the following diagram pictures the alignment of u with $U[k..k+|u|]$:

$$\begin{array}{cccccccc} u_{\diamond}(0) & u_{\diamond}(1) & \dots & u_{\diamond}(|u| - k - 1) & u_{\diamond}(|u| - k) & u_{\diamond}(|u| - k + 1) & \dots & u_{\diamond}(|u| - 1) \\ u_{\diamond}(k) & u_{\diamond}(k + 1) & \dots & u_{\diamond}(|u| - 1) & u_{\diamond}(0) & u_{\diamond}(1) & \dots & u_{\diamond}(k - 1) \end{array}$$

Now, let $\ell = |u| - k$. If $k < \ell$, then the checking of whether or not u is compatible with $U[k..k + |u|]$ can be done simultaneously with the checking of whether or not u is (k, ℓ) -special. Indeed, for any $0 \leq i < k$, consecutive positions in $seq_{k,\ell}(i)$ turn out to be aligned positions in the above diagram. The algorithm starts by considering $i = 0$ and repeats the following, increasing i until $i = k$ (whenever $i = k$, both u is compatible with $U[k..k + |u|]$ and u is not (k, ℓ) -special). While considering i , the algorithm computes $seq_{k,\ell}(i) = (i_0, i_1, i_2, \dots, i_{n+1})$ along with its letter *seqletter* initialized with $u_{\diamond}(i)$. Whenever the position i_j is added to the sequence, the algorithm compares $u_{\diamond}(i_j)$ with $u_{\diamond}(i_{j-1})$. If not compatible, then the compatibility of u with $U[k..k + |u|]$ is non-successful and the algorithm increases k by 1 and decreases ℓ by 1. If compatible, then the algorithm updates *seqletter* depending on the value of $u_{\diamond}(i_j)$. There are four cases that can arise while updating *seqletter* (here a, b denote distinct letters in A): (1) *seqletter* = \diamond and $u_{\diamond}(i_j) = \diamond$ (no update is needed); (2) *seqletter* = \diamond and $u_{\diamond}(i_j) = a$ (*seqletter* is updated with a); (3) *seqletter* = a and $u_{\diamond}(i_j) = a$ (no update is needed); and (4) *seqletter* = a and $u_{\diamond}(i_j) = b$ (here it is discovered that $u(i_0)u(i_1)u(i_2) \dots u(i_{n+1})$ is not 1-periodic). If any of Cases (1), (2) or (3) occurs and $j < n + 1$, then the algorithm repeats the process by adding the position i_{j+1} to the sequence. If any of Cases (1), (2) or (3) occurs and $j = n + 1$, then the algorithm increases i . If Case (4) occurs, then we claim that the algorithm will increase k by 1 and decrease ℓ by 1. To see this, if the number of holes seen so far in the sequence, or *seqholes*, is not less than 2, then u is (k, ℓ) -special and regardless of whether or not u is compatible with $U[k..k + |u|]$, the algorithm will increase k by 1 and decrease ℓ by 1. If *seqholes* < 2 , then u is (k, ℓ) -special or u is not compatible with $U[k..k + |u|]$, and again regardless of which case happens, the algorithm will increase k by 1 and decrease ℓ by 1. These changes in the original algorithm increase the time complexity by at most a constant factor. \square

References

1. Berstel, J., Boasson, L.: Partial words and a theorem of Fine and Wilf. *Theoret. Comput. Sci.* **218** (1999) 135–141
2. Blanchet-Sadri, F.: Periodicity on partial words. *Comput. Math. Appl.* **47** (2004) 71–82
3. Blanchet-Sadri, F.: Codes, orderings, and partial words. *Theoret. Comput. Sci.* **329** (2004) 177–202
4. Blanchet-Sadri, F.: Primitive partial words. *Discrete Appl. Math.* **148** (2005) 195–213
5. Blanchet-Sadri, F., Chriscoe, Ajay: Local periods and binary partial words: an algorithm. *Theoret. Comput. Sci.* **314** (2004) 189–216
<http://www.uncg.edu/mat/AlgBin/>
6. Blanchet-Sadri, F., Duncan, S.: Partial words and the critical factorization theorem. *J. Combin. Theory Ser. A* **109** (2005) 221–245
<http://www.uncg.edu/mat/cft/>

7. Blanchet-Sadri, F., Hegstrom, Robert A.: Partial words and a theorem of Fine and Wilf revisited. *Theoret. Comput. Sci.* **270** (2002) 401–419
8. Blanchet-Sadri, F., Luhmann, D.K.: Conjugacy on partial words. *Theoret. Comput. Sci.* **289** (2002) 297–312
9. 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
10. Crochemore, M., Rytter, W.: *Text Algorithms*. Oxford University Press (1994)
11. Giancarlo, R., Mignosi, F.: Generalizations of the periodicity theorem of Fine and Wilf. *Trees in algebra and programming - CAAP'94 (Edinburgh, 1994)*. Lecture Notes in Comput. Sci. Vol. 787. Springer, Berlin (1994) 130–141
12. Guibas, L.J., Odlyzko, A.M.: Periods in strings. *J. Combin. Theory Ser. A* **30** (1981) 19–42
13. Leupold, P.: Partial words for DNA coding. In *DNA 10 Tenth International Meeting on DNA Computing* (2004)
14. Lothaire, M.: *Algebraic Combinatorics on Words*. Cambridge University Press (2002)