UNIQUENESS THEOREMS ON WORDS AND SEQUENCE ASSEMBLY

ARTURO CARPI, ALDO DE LUCA, AND STEFANO VARRICCHIO

Abstract. We introduce the notions of (right) *univalent factor*, *minimal univalent factor*, and *starting block* of a word and we study their relationship with the well known notions of *boxes*, *superboxes*, and *minimal forbidden factors*. Moreover, we prove some new uniqueness conditions for words based on univalent factors. In particular, we show that a word is uniquely determined by its starting block, the set of the extensions of its minimal univalent factors, and its length or its terminal box. Finally, we show how the presented results and techniques can be used to solve the problem of sequence assembly for DNA molecules, under reasonable assumptions on the repetitive structure of the considered molecule and on the set of known fragments.

Résumé. On introduit des notions combinatoires sur les mots (facteurs univalents droite, bloc de départ) et on étudie leurs relations avec les notions mieux connues de boites, hyperboites et facteurs interdits minimaux. On démontre des rsultat d'unicité pour un mot satisfaisant des contraintes sur ces facteurs univalents. En particulier on prouve qu'un mot est déterminé de manière unique par son bloc de départ l'ensemble des extensions de ses facteurs univalents minimaux et sa longueur (ou bien sa boite terminale).

Enfin on montre comment les résultats obtenus et les techniques étudiées peuvent être utilisés pour la reconstitution de molcules DNA par l'assemblage de certains de leurs fragments.

1. Introduction

One of the main problems in the combinatorics of finite sequences of symbols, or *words*, consists in determining uniquely a word by knowing a suitable set of 'short' factors, or subwords. In fact, this problem is of great interest both from the theoretical and applicative points of view. As regards the applications we mention here the problem of the assembly of very long biological macromolecules, such as DNA and proteins, starting from a certain number of short fragments. Another application is the problem of transmitting and recovering very long messages using different communication channels each one for a short period.

A mathematical solution of the preceding problem is usually obtained by proving a *uniqueness theorem* on a word by knowing a well defined set of its factors and possibly having some further additional information such as its length or some suitable initial and/or terminal factor. From the applicative point of view a uniqueness theorem is of interest if there exist efficient algorithms in order to construct the preceding factors and, conversely, to recover uniquely the word by knowing the set of factors and the possible additional information on it. In some applications such as DNA assembly the limitations of experimental conditions give some strong constraints on the kind of information about the sequence. This conditions the uniqueness theorem and the corresponding algorithms which may be used.

In some recent papers new and interesting results on this and related subjects have been found in the framework of a suitable analysis of the structure of a finite word based on the notions of the *extendibility* and *repetitiveness* of its factors [2, 3, 4, 7, 8].

Let w be a word over a given finite alphabet. A *right extension* of a factor u of w is any factor ux of w, with x letter. The *right valence* of a factor u of w is the number (possibly 0) of its different right extensions in w. A factor u of w is *right extendable* if its right valence is greater than 0. In a symmetric way one can introduce *left extensions* and the *left valence* of any factor of w. A factor u of w is *left extendable* if its left valence is greater than 0. There exists a unique shortest factor (suffix) of w which is not right extendable. It is denoted by k_w and is called the *terminal box* of w. In a symmetric way there exists a unique shortest factor (prefix) which is not left extendable. It is denoted by h_w and is called the *initial box* of w. The lengths $|h_w|$ and $|k_w|$ of the initial and terminal boxes are denoted, respectively, by H_w and K_w.

A factor u of a given word w is *repeated* if it has at least two different occurrences in w. The shortest repeated prefix (resp. suffix) of w will be denoted by h'_w (resp. k'_w). In the study of repetitions in a word a crucial role is played by the so-called *special factors*. A factor u of w is called *right special* if its right valence is greater than 1, i.e., there exist at least two distinct letters x and y such that ux and uy are factors of w. In a symmetric way one can define *left special* factors. A factor u of w is called *bispecial* if it is right and left special. We denote by R_w (resp. L_w) the minimal natural number such that w has no right (resp. left) special factor of length R_w (resp. L_w).

We recall the following important relation between the parameters R_w, L_w, H_w, K_w, and the maximal length G_w of a repeated factor of a non-empty word w [13].

$$G_w = \max\{R_w, K_w\} - 1 = \max\{L_w, H_w\} - 1. \tag{1}$$

In this paper we consider the notion of *univalent factor* of a word. A factor u of a word w is called *right univalent* if its right valence is equal to 1. A right univalent factor is called *minimal* if no proper suffix of it is right univalent. In a symmetric way one can introduce left univalent factors and minimal left univalent factors.

There exists a relation between minimal univalent factors and special factors. In fact, any proper suffix of a minimal right univalent factor has to be a right special factor.

A particular right univalent factor of a non-empty word w which plays an important role in the theory, is the so-called *starting block* which is the shortest prefix $\overline{h}_w$ of w such that all proper prefixes of w of length $\geq |\overline{h}_w|$ are right univalent. We shall denote by R'_w the length of $\overline{h}_w$. One has $R'_w \leq \min\{H_w, R_w\}$.

In Sec. 2 some important results relating minimal univalent factors, the starting block, and other combinatorial objects such as *boxes*, *superboxes*, and *hyperboxes* are presented. Moreover, an interesting relation existing, in the case of a two-letter alphabet, between minimal univalent factors and minimal forbidden factors of a word [1, 11] is stated.

In Sec. 3 we prove some new uniqueness theorems on words based on univalent factors. The main result is (cf. Theorem 9) that *a word is uniquely determined by its starting block, the set of the (right) extensions of its minimal right univalent factors, and its length or its terminal box.*

This uniqueness theorem is interesting since it is possible to device an efficient algorithm in order to reconstruct the sequence. Some refinements and generalizations of this theorem are proved. Moreover, we give a simple and different proof of the following important result proved in [3, 5]: *Let w be a word. If all factors of a word v up to length $1 + \max\{R_w, K_w\}$ are also factors of w, then v is a factor of w. Moreover, if w and v have the same set of factors up to length $1 + \max\{R_w, K_w\}$, then $w = v$.*

This latter result is of particular interest if one has to reconstruct a sequence knowing only factors without any further information on the sequence such as length, initial or terminal parts.

In Sec. 4 the problem of the sequence assembly is considered in the case of DNA molecules. The aim of the section is that of showing how, by using results and techniques of the previous sections, one can solve the problem by making reasonable assumptions on the repetitive structure of the considered DNA and on the set of considered DNA fragments. An efficient algorithm which is able to reconstruct the word from the fragments is described.

2. Univalent factors and hyperboxes

In the sequel, if not differently specified, we shall refer only to right univalent factors, that we simply call *univalent factors*. The minimal right univalent factors will be called *minimal univalent factors*. For any result concerning right univalent factors, a dual one, concerning left univalent factors holds.

Let w be a word over the alphabet A. The set of minimal univalent factors of w will be denoted by $\mathcal{U}_w$. The set

$$\mathcal{U}_w A \cap \mathrm{Fact}(w),$$

where $\mathrm{Fact}(w)$ denotes the set of all factors of w, will be called the set of the *extensions of the minimal univalent factors* of w.

Lemma 1. *Let w be a word. For any $u \in \mathcal{U}_w$, one has $|u| \leq R_w$.*

Proof. If u is the empty word, the result is trivial. If $u \neq \epsilon$, then u can be written as $u = as$, with a a letter and s a right special factor of w. Since $|s| \leq R_w - 1$, it follows $|u| \leq R_w$. □

Let us now introduce the notion of (right) *hyperbox* of a word w. A factor u of w is called a *hyperbox* if $u = asb$ where a and b are letters, s is a right special factor of w and, for any letter $x \neq b$, one has $asx \notin \mathrm{Fact}(w)$.

We shall denote by $\mathcal{Q}_w$ the set of the hyperboxes of w. A hyperbox is called *maximal* (with respect to the factorial order) if it is not a factor of another hyperbox. For any word $w \in A^*$ we set $\mathrm{alph}(w) = \mathrm{Fact}(w) \cap A$.

Proposition 1. *Let $w \in A^*$ be a word such that* $\mathrm{Card}(\mathrm{alph}(w)) > 1$. *Then,*

$$\mathcal{U}_w A \cap \mathrm{Fact}(w) = \mathcal{Q}_w.$$

Proof. Let v be a factor of w such that $v \in \mathcal{U}_w A$. We can write $v = ub$, with $u \in \mathcal{U}_w$ and $b \in A$. Since $\mathrm{Card}(\mathrm{alph}(w)) > 1$, u is non-empty. This implies that $u = as$ with $a \in A$ and s a right special factor of w. Thus, there exists a letter $c \neq b$ such that $sc \in \mathrm{Fact}(w)$. Since u is univalent one has that, for all $x \in A \setminus \{b\}$, $asx \notin \mathrm{Fact}(w)$. Hence, $v \in \mathcal{Q}_w$.

Let $v \in \mathcal{Q}_w$. There exist letters $a, b, c \in A$, $b \neq c$, and $s \in A^*$ such that $v = asb \in \mathrm{Fact}(w)$, $sc \in \mathrm{Fact}(w)$, and for all letters $x \neq b$, $asx \notin \mathrm{Fact}(w)$. Hence, as is a univalent factor of w which is minimal since s is right special. Thus $as \in \mathcal{U}_w$ and $v \in \mathcal{U}_w A \cap \mathrm{Fact}(w)$. □

For any set of words X, we denote by $\mathrm{Max}\, X$ the set of maximal elements of X with respect to the factorial order. The following proposition [9] shows that the maximal elements in the set of the extensions of minimal univalent factors of a word w coincides with the set of the extensions in w of the maximal elements in the set of minimal univalent factors of w.

Proposition 2. *For any word $w \in A^*$,*

$$(\mathrm{Max}\mathcal{U}_w)A \cap \mathrm{Fact}(w) = \mathrm{Max}(\mathcal{U}_w A \cap \mathrm{Fact}(w)).$$

From Propositions 1 and 2 one has that the set of maximal hyperboxes of a word w which is not a power of a single letter coincides with the set of extensions in w of the maximal elements in the set of minimal univalent factors of w.

Example 1. Let $w = ababcacbabccaa$. The minimal univalent factors of w are ac, ba, cb, cc, bab, bca, cca, $ababc$, $cbabc$. The hyperboxes are acb, bab, cba, cca, $babc$, $bcac$, $ccaa$, $ababca$, $cbabcc$. The elements of $\mathrm{Max}\mathcal{U}_w$ are ac, bca, cca, $ababc$, $cbabc$. The maximal hyperboxes are acb, $bcac$, $ccaa$, $ababca$, $cbabcc$.

A factor v of w will be called a *right proper box* or, simply, *right box* of w if $v = asb$ with $a, b \in A$ and s right special in w. Symmetrically, a *left box* of w is any factor $v = asb$ with $a, b \in A$ and s a left special factor. A right (resp. left) box of w is *maximal* (with respect to the factorial order) if it is not a factor of another right (resp. left) box. We notice that any hyperbox of w is a right box.

We recall [3] that a factor of w is a *proper box* if it is both a right and a left box. A proper box is *maximal* if it is not a factor of another proper box. The set of maximal proper boxes of a word w will be denoted by $\mathcal{B}_w$.

Proposition 3. *If* $\mathrm{Card}(\mathrm{alph}(w)) > 1$, *then the set $\mathcal{B}_w^r$ of the maximal right boxes of w is given by*

$$\mathcal{B}_w^r = \mathrm{Max}(\{\overline{h}_w\} \cup \mathcal{Q}_w).$$

Proof. First, we prove that any maximal right box of a word w is either a factor of a hyperbox of w or a factor of $\overline{h}_w$. Let α be a maximal right box of w. We can write $\alpha = asb$ with a and b letters and s a right special factor of w. If as is not right special, then as is a minimal univalent factor of w and, by Proposition 1, α is a hyperbox. Let us then suppose that as is right special. If α is left extendable, i.e., $casb \in \mathrm{Fact}(w)$ for a suitable letter c, then $c\alpha = casb$ is a right box, contradicting the maximality of α. Hence, α is not left extendable in w and, consequently, it is a prefix of w. Since, moreover, as is right special in w, we conclude that α is a prefix of $\overline{h}_w$.

Conversely, by definition, any hyperbox is a right box. Now, we prove that $\overline{h}_w$ is a factor of a right box. We first assume that $|\overline{h}_w| \leq 2$. Since $\mathrm{Card}(\mathrm{alph}(w)) > 1$, ϵ is a bispecial factor, so that all the factors of w of length 2 are right boxes. Hence, $\overline{h}_w$ is a factor of a right box. Now suppose that $|\overline{h}_w| > 2$. Then $\overline{h}_w$ can be written as $\overline{h}_w = asb$, with $a, b \in A$ and $s \in A^+$. By the definition of $\overline{h}_w$, as is a right special factor of w. Thus s is a right special factor of w and, therefore, asb is a right box of w.

Thus we have shown that $\mathrm{Fact}(\mathcal{B}^r_w) = \mathrm{Fact}(\{\overline{h}_w\} \cup \mathcal{Q}_w)$. Hence, one derives that the maximal elements, with respect to the factorial order, in the set of right boxes of w coincide with the maximal elements in the set $\{\overline{h}_w\} \cup \mathcal{Q}_w$. □

We recall the notion of *superbox* of a word w [3]. A factor v of w is called a *superbox* of w if one can write

$$v = asb,$$

with a and b letters, where s is a repeated factor of w while as and sb are unrepeated in w. The set of superboxes of a word w is a factor code, i.e., no superbox is a factor of another superbox. Superboxes of a word w can be constructed by a simple algorithm described in [3].

The following proposition summarizes some results proved in [9] which relate the notions of boxes, hyperboxes, and superboxes.

Proposition 4. *Let w be a word. If $v = asb$ is a superbox of w with $s \neq k'_w$, then v is a hyperbox of w. If $u = ctd$ is a hyperbox of w with $t \neq h'_w$, then u is a proper box of w.*

Conversely, any proper box of w is a factor of a hyperbox of w or of $\overline{h}_w$ and any hyperbox of w is a factor of a superbox of w or of h_w or of k_w.

Example 2. In the case of the word considered in Example 1, one easily verifies that the set of superboxes coincides with the set of maximal hyperboxes.

In the case of the word $w = abaababaaba$ one has that $h_w = abaabab$, $k_w = babaaba$, the set of superboxes of w is empty and the hyperboxes are ba, aab, $aabab$, $babaa$. The words ba and aab are factors of both h_w and k_w, while $aabab$ is a factor of h_w and $babaa$ is a factor of k_w.

Let w be the word $w = abccbabcab$. One has that $h_w = \overline{h}_w = abcc$. The maximal proper boxes are abc, bca, bcc, cba, ccb. The hyperboxes are ab, abc, cba, ccb, $babca$. All the proper boxes of w are factors of a hyperbox, with the only exception of bcc, which is a factor of $\overline{h}_w$.

By the previous results one simply derives the following proposition proved in [3].

Proposition 5. *Let $u = asb$ be a superbox of w, with $a, b \in A$. If $s \neq h'_w$ and $s \neq k'_w$, then u is a maximal proper box.*

A word $u \in A^*$ is called a *minimal forbidden factor* of $w \in A^*$ if $u \notin \mathrm{Fact}(w)$ while any proper factor of u is a factor of w. This notion has been considered in [1] in the setting of symbolic dynamics and in [11] for problems of text-compression in the case of a binary alphabet. In this latter case, there exists an interesting relationship between minimal forbidden factors and minimal univalent factors of a word [9].

Proposition 6. *Let $v = ux$, with $u \in \{a,b\}^*$ and $x \in \{a,b\}$, be a minimal forbidden factor of a word $w \in \{a,b\}^*$. Then either $u = k_w$ or u is a minimal univalent factor of w. Conversely, if u is a minimal univalent factor of w, then there exists a letter $x \in \{a,b\}$ such that $v = ux$ is a minimal forbidden factor of w.*

3. Uniqueness theorems

In this section, we prove some uniqueness theorems on words based on univalent factors. The main result shows that a word is uniquely determined by its starting block, the set of the extensions of its minimal univalent factors, and its length or its terminal box. We start by stating the following technical lemma [9].

Lemma 2. *Let $f, g \in A^*$ be two distinct words such that $|f| = |g|$ or f and g have a common suffix of length $\max\{K_f, K_g\}$. Then there exist words $v, r, s \in A^*$ and two distinct letters $a, b \in A$ such that*

$$f = var \quad \textit{and} \quad g = vbs.$$

Theorem 7. *Let $f, g \in A^*$ be two words such that*

1. *$|f| = |g|$ or f and g have a common suffix of length $\max\{K_f, K_g\}$,*
2. *$\mathcal{U}_f A \cap \mathrm{Fact}(g) \subseteq \mathrm{Fact}(f)$,*
3. *$\overline{h}_f$ is a prefix of g.*

Then $f = g$.

Proof. Let us assume, by contradiction, that $f \neq g$. By Condition 1, in view of Lemma 2 there exist words $v, r, s \in A^*$ and two distinct letters a and b such that $f = var$ and $g = vbs$. Since v is the longest common prefix of f and g, by Condition 3, one has $|v| \geq |\overline{h}_f| = R'_f$. Thus, the word v is univalent in f. Consequently, there exists a suffix u of v which is a minimal univalent factor of f. One has $ub \in \mathcal{U}_f A \cap \mathrm{Fact}(g)$ and, therefore, by Condition 2, $ub \in \mathrm{Fact}(f)$. Since $ua \in \mathrm{Fact}(f)$, u would be a right special factor of f, which is a contradiction. □

By the previous theorem, one derives the following:

Proposition 8. *Let $f, g \in A^*$ be two words such that*

1. *$|f| = |g|$ or f and g have a common suffix of length $\max\{K_f, K_g\}$,*
2. *$\mathcal{U}_f A \cap \mathrm{Fact}(f) \subseteq \mathcal{U}_g A \cap \mathrm{Fact}(g)$,*
3. *$\overline{h}_f$ is a prefix of g.*

Then $f = g$.

Proof. It is sufficient to verify that Condition 2 of the previous theorem is satisfied. Indeed, let $v \in \mathcal{U}_f A \cap \mathrm{Fact}(g)$. We can write v as $v = ua$, with $u \in \mathcal{U}_f$ and a a letter. Let b be the only letter such that $ub \in \mathrm{Fact}(f)$. Thus $ub \in \mathcal{U}_f A \cap \mathrm{Fact}(f)$ and, by Condition 2, $ub \in \mathcal{U}_g A \cap \mathrm{Fact}(g)$. This implies $u \in \mathcal{U}_g$. Thus, since $ua, ub \in \mathrm{Fact}(g)$, it follows $a = b$ and then $v \in \mathrm{Fact}(f)$. □

A straightforward consequence of the previous proposition is the following noteworthy theorem:

Theorem 9. *A word is uniquely determined by its starting block, the set of the extensions of its minimal univalent factors, and its length or its terminal box.*

Let us give now a simple procedure based on the preceding theorem, which allows us to construct the word w by knowing its starting block $\overline{h}_w$, the set of the extensions of its minimal univalent factors, and the length of w or the terminal box k_w. Initially, we set $p = \overline{h}_w$. Now suppose that we have already constructed a prefix p of w of length $|p| = n \geq R'_w$. If $n = |w|$ or p has the suffix k_w, then $w = p$ and the procedure ends. Otherwise, let $u \in \mathcal{U}_w$ be the unique suffix of p which is a minimal univalent factor of w. Let a be the unique letter extending u in w. Then the prefix of w of length $n + 1$ is pa.

Example 3. Let $w = ababcacbabccaa$ be the word considered in Example 1. In this case, $\overline{h}_w = h_w = aba$ and the set $\mathcal{U}_w A \cap \mathrm{Fact}(w)$ coincides with the set of hyperboxes acb, bab, cba, cca, $babc$, *bcac*, *ccaa*, *ababca*, *cbabcc*. Let us start with $\overline{h}_w = aba$. We can construct the following chain of derivations, where we underline the unique suffix which is a minimal univalent factor of w.

$$\begin{aligned} a\underline{ba} &\to a\underline{bab} \to \underline{ababc} \to aba\underline{bca} \to ababc\underline{ac} \to ababca\underline{cb} \to ababcac\underline{ba} \\ &\to ababcac\underline{bab} \to ababca\underline{cbabc} \to ababcacbab\underline{cc} \to ababcacbab\underline{cca} \\ &\to ababcacbabccaa. \end{aligned}$$

The following recent result [7] is a noteworthy improvement of the so called *maximal box theorem* proved in [3].

Theorem 10. *Let f and g be two words such that*

(1) $|f| = |g|$ *or f and g have a common suffix of length* $\max\{K_f, K_g\}$,
(2) *f and g have a common prefix of length* $\max\{1, R'_f, R'_g\}$,
(3) $\mathcal{B}_f \subseteq \mathrm{Fact}(g)$ *and* $\mathcal{B}_g \subseteq \mathrm{Fact}(f)$.

Then $f = g$.

By the preceding theorem one derives the following proposition.

Proposition 11. *Let $f, g \in A^*$ be two words such that*

(1) $|f| = |g|$ *or f and g have a common suffix of length* $\max\{K_f, K_g\}$,
(2) *f and g have a common prefix of length* $\max\{R'_f, R'_g\}$,
(3) $\mathcal{U}_f A \cap \mathrm{Fact}(f) \subseteq \mathrm{Fact}(g)$ *and* $\mathcal{U}_g A \cap \mathrm{Fact}(g) \subseteq \mathrm{Fact}(f)$.

Then $f = g$.

Proof. Let us verify that Condition 3 of Theorem 10 is satisfied. Indeed, if $\mathrm{Card}(\mathrm{alph}(f)) = 1$, then $\mathcal{B}_f = \emptyset \subseteq \mathrm{Fact}(g)$. If $\mathrm{Card}(\mathrm{alph}(f)) > 1$, then by Proposition 4, one has that $\mathcal{B}_f \subseteq \mathrm{Fact}(\{\overline{h}_f\} \cup \mathcal{Q}_f)$. Moreover, by Proposition 1, $\mathcal{Q}_f = \mathcal{U}_f A \cap \mathrm{Fact}(f)$. By Condition 3, $\mathcal{Q}_f \subseteq \mathrm{Fact}(g)$ and by Condition 2, $\overline{h}_f \in \mathrm{Fact}(g)$. Thus, $\mathcal{B}_f \subseteq \mathrm{Fact}(g)$. In a similar way, one proves that $\mathcal{B}_g \subseteq \mathrm{Fact}(f)$.

Now, let us verify that Condition 2 of Theorem 10 is satisfied. This is trivially true if $R'_f > 0$ or $R'_g > 0$. Let us then suppose that $R'_f = R'_g = 0$. This implies that f and g are powers of single letters. Thus $\mathcal{U}_f = \mathcal{U}_g = \{\epsilon\}$ so that Condition 3 becomes $\mathrm{alph}(f) \subseteq \mathrm{Fact}(g)$ and $\mathrm{alph}(g) \subseteq \mathrm{Fact}(f)$. Thus, f and g are powers of the same letter and they have the same prefix of length 1. Since all conditions of Theorem 10 are satisfied, it follows $f = g$. □

As a consequence of the preceding proposition, one has that a word f is uniquely determined by its starting block, a set T such that $\mathcal{U}_f A \cap \mathrm{Fact}(f) \subseteq \mathrm{Fact}(T) \subseteq \mathrm{Fact}(f)$, and its length or its terminal box. In particular, one can take $T = (\mathrm{Max}\,\mathcal{U}_f)A \cap \mathrm{Fact}(f)$.

Let $u, v \in A^*$ and $n \geq 0$. We denote by $A^{[n]}$ the set of all words of A^* of length $\leq n$. We write $u \preceq_n v$ if $\mathrm{Fact}(u) \cap A^{[n]} \subseteq \mathrm{Fact}(v)$. If $u \preceq_n v$ and $v \preceq_n u$, then we write $u \sim_n v$.

A further consequence of Theorem 7 is the following noteworthy proposition.

Proposition 12. *Let $f, g \in A^*$ be two words satisfying the following conditions:*

(1) $|f| = |g|$ *or f and g have a common suffix of length* $\max\{K_f, K_g\}$,
(2) $g \preceq_{1+R_f} f$,
(3) *f and g have the same prefix of length R'_f.*

Then $f = g$.

Proof. In view of Lemma 1, one has that $\mathcal{U}_f A \cap \mathrm{Fact}(g) \subseteq \mathrm{Fact}(g) \cap A^{[1+R_f]}$. By Condition 2, one derives $\mathcal{U}_f A \cap \mathrm{Fact}(g) \subseteq \mathrm{Fact}(f)$, so that the statement follows from Theorem 7. □

We remark that a weaker version of the previous proposition, where, in Condition 3, R'_f is replaced by R_f, is in [6]. Moreover, the following example shows that the bound R'_f in Condition 3 is optimal.

Example 4. Let f and g be the words $f = abbcababbc$ and $g = ababbcabab$. One has $|f| = |g|$, $R_f = 3$, and $R'_f = 3$. One easily checks that $g \preceq_4 f$ and the longest common prefix of f and g has length 2.

Proposition 13. *Let $f, g \in A^*$ be two words satisfying the following conditions:*
(1) *the prefix of length R_f of g is a factor of f,*
(2) $(\mathcal{U}_f \cup \{k_f\})A \cap \mathrm{Fact}(g) \subseteq \mathrm{Fact}(f)$.

Then g is a factor of f.

Proof. Suppose that $g \notin \mathrm{Fact}(f)$ and let v be the longest prefix of g such that $v \in \mathrm{Fact}(f)$. Then we can write $g = va\lambda$, with $a \in A$, $\lambda \in A^*$, and $va \notin \mathrm{Fact}(f)$. The word v is right extendable in f, otherwise, k_f would be a suffix of v so that $k_f a \in \mathrm{Fact}(g)$ and, by Condition 2, $k_f a \in \mathrm{Fact}(f)$, which is a contradiction. Thus, there exists a letter $b \neq a$ such that $vb \in \mathrm{Fact}(f)$. Since by Condition 1, $|v| \geq R_f$, v is univalent in f. This implies that there exists a word $u \in \mathcal{U}_f$ which is a suffix of v. Clearly, one has $ub \in \mathrm{Fact}(f)$. Moreover, by Condition 2, $ua \in \mathrm{Fact}(f)$, which contradicts the fact that u is univalent in f. □

The following theorem synthesizes some results proved in [3, 5]. We give here a simple proof based on the previous proposition.

Theorem 14. *Let $f, g \in A^*$ be two words and set $n = 1 + \max\{R_f, K_f\}$. If $g \preceq_n f$, then g is a factor of f. If $g \sim_n f$, then $f = g$.*

Proof. If $|g| \leq n$, then since $g \preceq_n f$, one has trivially $g \in \mathrm{Fact}(f)$. Let us then suppose that $|g| > n$. In this case, the prefix of g of length R_f is also a factor of f. Moreover, in view of Lemma 1, all words of $(\mathcal{U}_f \cup \{k_f\})A$ have length $\leq n$, so that Condition 2 of the preceding proposition is satisfied. This implies $g \in \mathrm{Fact}(f)$. Now, suppose that $g \sim_n f$. Since g is a factor of f, the length G_g of the maximal repetition in g is less than or equal to the length G_f of the maximal repetition in f. This implies, by Eq. (1) that $n' = 1 + \max\{R_g, K_g\} \leq n$. Thus, $f \preceq_{n'} g$ so that $f \in \mathrm{Fact}(g)$. We conclude that $f = g$. □

We remark that in [3] it was also proved that the bound $n = 1 + \max\{R_f, K_f\}$ in the previous theorem is optimal.

4. Applications to sequence assembly

Sequence assembly is one of the most important algorithmic problem in molecular biology. Indeed, DNA sequences can be viewed as words over a four letter alphabet $\{A, C, G, T\}$, where each letter is the initial of the chemical name of a *nucleotide*. DNA sequences can be very long; their length can vary from $\approx 10^6$ to $\approx 10^9$. Current technologies do not allow one to read a very long sequence; for instance, gel electrophoresis permits to determine directly *fragments* of DNA sequences of length ≈ 500. Therefore, the main task consists in reconstructing the entire sequence having only access to short fragments of it. Moreover, a further difficulty is due to the fact that we do not know the exact position of a given fragment in the entire sequence.

This problem can be formulated as follows: given a target (unknown) word w, try to reconstruct it from the knowledge of some suitable factors of w (also called *fragments* or *reads*) $s_1, s_2, \ldots, s_l$.

There exists a large literature on DNA sequencing and assembling. We refer to [14, 15] and references therein. A more recent survey on this subject is in [16].

In this section we present an approach to this problem purely based on the notions and techniques considered in the previous sections. The first problem consists in making reasonable assumptions on the target word w and on the set of reads $\{s_1, s_2, \ldots, s_l\}$ in order to achieve all the information that uniquely determines the word w. A second problem is to find efficient algorithms that on inputs $s_1, s_2, \ldots, s_l$ reconstruct the word w.

Let G_w be the maximal length of a repeated factor of w (cf. Eq. (1)). We shall assume that: (a) the value G_w is small, (b) there is an integer $k > G_w$ such that the set $\{s_1, s_2, \ldots, s_l\}$ is a *covering* of the factors of length $k+1$ of w, i.e., any factor of w of length $k+1$ occurs in some s_i, $i = 1, \ldots, n$.

Assumption (a) is reasonable, since many microorganisms such as prokaryotes (bacteria) and lower eukaryotes have DNA sequences without long repetitions. For instance, in the case of prokaryotes, the length of maximal repetitions is ≤ 15. Assumption (b) can be acquired taking the reads coming from many random fragmentations of the word w.

As we have seen in the previous section (cf. Theorem 14), the knowledge of the factors of w up to $k+1$ is sufficient to reconstruct the word w. We shall show that an efficient way to realize this reconstruction may consists of the following two stages: first build a deterministic automaton accepting the language $\mathrm{Fact}(w)$ and next search for the longest word accepted by such an automaton, which indeed has to be equal to w.

Let $\mathcal{A} = (Q, A, \delta, q_0, Q_+)$ be an incomplete deterministic finite automaton, where Q is the set of the states, A is the alphabet, δ is the transition function, q_0 is the initial state, and Q_+ the set of terminal states. A state q of $\mathcal{A}$ will be called *univalent* if q has only one outgoing edge. For a univalent state q we define $\mathrm{out}(q) = a$, where a is the unique letter for which $\delta(q, a)$ is defined. Now, let us refer to an incomplete deterministic automaton $\mathcal{A} = (Q, A, \delta, q_0, Q)$ accepting $\mathrm{Fact}(w)$ and such that all states are terminal. We observe that if $v \in \mathrm{Fact}(w)$, then there is a one-to-one correspondence between the right extensions of v in w and the edges outgoing from the state $q = \delta(q_0, v)$. In particular, v is univalent in w if and only if q is a univalent state of $\mathcal{A}$ and, in such a case, the only extension of v in w is $v\,\mathrm{out}(q)$. Moreover, v is not right extendable in w if and only if q has no outgoing edge.

Clearly, the word w is the longest word accepted by $\mathcal{A}$, that is, the label of the longest path starting from q_0, in the graph of the automaton $\mathcal{A}$. We notice that all states encountered in such a path, after the first R'_w steps, have at most one outgoing edge. Indeed, they are equal to $\delta(q_0, u)$, for some prefix u of w of length $\geq R'_w$. Thus, a simple procedure to reconstruct w by knowing $\mathcal{A}$ and a prefix v of w of length $\geq R'_w$ consists in extending the path starting from q_0 and labeled v in the only possible way, until a state with no outgoing edge is found: w will be the label of such a path.

If we do not know a prefix of w of length $\geq R'_w$, then we can still reconstruct w starting by a factor v of w of length $> G_w$ and using two automata $\mathcal{A}$ and $\mathcal{A}'$ recognizing respectively $\mathrm{Fact}(w)$ and $\mathrm{Fact}(\tilde{w})$, where $\tilde{w}$ is the *reversed* (or *mirror image*) of w. We proceed in the following way: also in this case, we extend the path in $\mathcal{A}$ starting from the initial state and labeled v in the only possible way, until a state with no outgoing edge is found. The label u of such a path will be a factor of w which is not right extendable. One can easily verify that this is equivalent to say that $\tilde{u}$ is a factor of $\tilde{w}$ which is not left extendable. Thus, $\tilde{u}$ is a prefix of $\tilde{w}$ of length $\geq R'_{\tilde{w}}$ and we can use the procedure described before to compute $\tilde{w}$. Finally, reversing $\tilde{w}$, we obtain w.

Now the main problem consists in the construction of an automaton recognizing $\mathrm{Fact}(w)$.

The *failure function* is the map $f : A^* \to A^*$ defined as

$$f(\epsilon) = \epsilon, \qquad \text{and} \qquad \text{for all } a \in A,\ u \in A^*,\ f(au) = u.$$

Let k be a positive integer, and consider the deterministic (incomplete) finite automaton $\mathcal{A}_k(w) = (Q, A, \delta, q_0, Q)$, where the input alphabet is A, the set of states is $Q = \mathrm{Fact}(w) \cap A^{[k]}$, all states are terminal, the initial state is $q_0 = \epsilon$, and the transition function $\delta : Q \times A \to Q$ is the partial map, defined as follows: for any $u \in Q$, $a \in A$,

$$\delta(u, a) = \begin{cases} ua & \text{if } ua \in \mathrm{Fact}(w) \text{ and } |u| < k \\ f(ua) & \text{if } ua \in \mathrm{Fact}(w) \text{ and } |u| = k \\ \text{undefined} & \text{otherwise.} \end{cases}$$

This is a standard construction (see, e.g., [12, 10]). We observe that the automaton $\mathcal{A}_k(w)$ can be obtained by the tree of the factors (also called *trie*) of w of length $\leq k+1$, simply replacing the edges

$u \xrightarrow{a} v$, $u \in \mathrm{Fact}(w) \cap A^k$, $a \in A$, $v = ua$, with $u \xrightarrow{a} f(v)$, and then deleting the corresponding nodes ua.

We remark that for any language L over the alphabet A we can define the automaton $\mathcal{A}_k(L)$ taking as set of states the set $Q = \mathrm{Fact}(L) \cap A^{[k]}$ and defining q_0 and δ in a similar way. A proof of the following proposition can be found in [9].

Proposition 15. *The automaton $\mathcal{A}_k(w)$ accepts the language*

$$\{v \in A^* \mid v \preceq_{1+k} w\}.$$

Proposition 16. *If $k > G_w$, then the automaton $\mathcal{A}_k(w)$ accepts the language* $\mathrm{Fact}(w)$.

Proof. By Eq. (1) if $k > G_w$, then $k \geq \max\{R_w, K_w\}$. By Theorem 14, under this condition one has $v \preceq_{1+k} w$ if and only if $v \in \mathrm{Fact}(w)$. The conclusion follows from the previous proposition. □

As we have seen, $\mathcal{A}_k(w)$ can be easily obtained by the trie of the factors of w of length $\leq k+1$. An algorithm (BUILD-FACT) to construct such a trie, given w, is in [12].

Suppose that given the factors $s_1, s_2, \ldots, s_l$ of w we want to construct $\mathcal{A}_k(w)$. We can use a simple variant of the quoted algorithm BUILD-FACT in order to construct the trie of the set

$$S = \bigcup_{i=1}^{l} \mathrm{Fact}(s_i) \cap A^{[1+k]}.$$

Now, replacing the edges $u \xrightarrow{a} v$, $u \in S \cap A^k$, $a \in A$, $v = ua$, with $u \xrightarrow{a} f(v)$ and deleting the corresponding nodes ua, we obtain the automaton

$$\mathcal{A}_k(\{s_1, s_2, \ldots, s_l\}).$$

If we suppose that the words s_1, s_2, …, s_l are a *covering* of the factors of length $k+1$ of w, then one has trivially $\mathcal{A}_k(\{s_1, s_2, \ldots, s_l\}) = \mathcal{A}_k(w)$.

Now we discuss briefly the complexity of the previous algorithms. In our analysis we consider the size of the alphabet to be constant; indeed for DNA this size is exactly 4. Then we study the complexity in terms of the three integer parameters k, n, and m, where $n = |w|$, $m = \sum_{i=1}^{l} |s_i|$, and $k > G_w$.

We begin by considering the *space memory* necessary in the execution of the algorithm. An analysis of the algorithm (cf. [9]) shows that this is essentially the space necessary to memorize the automaton $\mathcal{A}_k(w)$ and some auxiliary data, associated with the states of the automaton, and requiring constant space. Therefore, an upper bound to the size of the memory is proportional to the number of states of the automaton, i.e., to $\mathrm{Card}(\mathrm{Fact}(w) \cap A^{[k]})$ which is $\leq kn$.

Now we consider the *time complexity* of our algorithm. Observe that the search for the longest path in an automaton recognizing $\mathrm{Fact}(w)$, given a prefix of length $\geq R'_w$ or a factor of length $\geq G_w$ requires linear time with respect to n. Moreover, it is possible to show (cf. [9]) that the time necessary to construct the automaton $\mathcal{A}_k(\{s_1, s_2, \ldots, s_l\})$ is $O(kn + m)$.

In practical cases, one has to fix the integer k in such a way that one can reasonably expect that Condition (b) stated at the beginning of this section is satisfied. If this is the case, then the automaton $\mathcal{A}_k(\{s_1, s_2, \ldots, s_l\})$ accepts the language $\mathrm{Fact}(w)$, so that our algorithm actually reconstructs w. However, with a bad choice of k, it may happen that either $k \leq G_w$ or $\{s_1, s_2, \ldots, s_l\}$ is not a covering of the factors of length $k+1$ of w. We remark that, if $\{s_1, s_2, \ldots, s_l\}$ is a covering of the factors of length $k+1$ of w, then a suitable design of the algorithm can allow one to detect whether $k \leq G_w$ or not.

In conclusion we stress that our approach to DNA assembly is based on assumptions (a) and (b) which are very strong if we consider the real situations. In fact, the sequence assembly is much more complicated for superior eukaryotes, whose genes contain large repetitions. Moreover, in the real situations one has to consider the presence of sequencing errors (1 to 5 % of the sequence), so that our approach can be considered just as an essential initial step for a further research in this direction.

References

[1] M.-P. Béal, F. Mignosi, and A. Restivo, Minimal forbidden words and symbolic dynamics, Proc.s STACS '96, Lecture Notes in Computer Science, vol. 1046, Springer-Verlag (Berlin, 1996) pp. 555–566.

[2] A. Carpi and A. de Luca, Words and repeated factors, *Séminaire Lotharingien de Combinatoire*, **B42l** (1998) 24 pp. and in: D. Foata and G. Han, (Eds.), *The Andrews Festschrift*, Springer-Verlag (Berlin, to appear) pp. 295–306.

[3] A. Carpi and A. de Luca, Words and special factors, Preprint 98/33, Dipartimento di Matematica dell'Università di Roma "La Sapienza", 1998, *Theoretical Computer Science*, to appear.

[4] A. Carpi and A. de Luca, Repetitions and boxes in words and pictures, in: J. Karhumäki, H. Maurer, G. Păun, and G. Rozenberg, (Eds.), *Jewels are Forever*, Springer-Verlag (Berlin, 1999), pp. 295–306.

[5] A. Carpi and A. de Luca, Special factors, periodicity, and an application to Sturmian words, *Acta Informatica* **36** (2000) 983–1006.

[6] A. Carpi and A. de Luca, Semiperiodic words and root-conjugacy, preprint 10/2000, Dipartimento di Matematica dell'Università di Roma "La Sapienza", 2000, *Theoretical Computer Science*, to appear.

[7] A. Carpi and A. de Luca, Periodic-like words, in: Lecture Notes in Computer Science, vol. 1893, Springer-Verlag (Berlin, 2000) pp. 264–275.

[8] A. Carpi, A. de Luca, and S. Varricchio, Special factors and uniqueness conditions in rational trees, Preprint 99/24, Dipartimento di Matematica dell'Università di Roma "La Sapienza", 1999, *Theory of Computing Systems*, to appear.

[9] A. Carpi, A. de Luca, and S. Varricchio, Words, univalent factors, and boxes, preprint 41/2000, Dipartimento di Matematica dell'Università di Roma "La Sapienza", 2000.

[10] M. Crochemore and C. Hancart, Automata for matching patterns, in: *Handbook of Formal Languages*, G. Rozenberg and A. Salomaa (ed.s), Springer-Verlag (Berlin, 1997) vol. 2, pp. 399–462.

[11] M. Crochemore, F. Mignosi, and A. Restivo, Automata and forbidden words, *Information Processing Letters* **67** (1998) 111–117.

[12] M. Crochemore, F. Mignosi, A. Restivo, and S. Salemi, Text compression using antidictionaries, in: Lecture Notes in Computer Science, vol. 1664, Springer-Verlag (Berlin, 1999) pp. 261–270.

[13] A. de Luca, On the combinatorics of finite words, *Theoretical Computer Science* **218** (1999) 13–39.

[14] D. Gusfield, *Algorithms on strings, trees and sequences,* Cambridge University Press, 1997.

[15] J. D. Kececioglu and E. W. Myers, Combinatorial algorithms for DNA sequence assembly, *Algorithmica* **13** (1995) 7–51.

[16] E. W. Myers, Whole-genome DNA sequencing, *Computing in Science & Engineering*, May–June 1999, 33–43.

Arturo Carpi, Istituto di Cibernetica del CNR, via Toiano 6, 80072 Arco Felice (NA), Italy
E-mail address: arturo@arturo.cib.na.cnr.it

Aldo de Luca, Dipartimento di Matematica dell'Università di Roma "La Sapienza", piazzale Aldo Moro 2, 00185 Roma, Italy
E-mail address: deluca@mat.uniroma1.it

Stefano Varricchio, Dipartimento di Matematica dell'Università di Roma "Tor Vergata", viale della Ricerca Scientifica, 00133 Roma, Italy
E-mail address: varricch@mat.uniroma2.it