Languages of lossless seeds

Karel Břinda

Languages of lossless seeds

Karel Břinda
Laboratoire d'Informatique Gaspard Monge
Université Paris-Est Marne-la-Vallée
Paris, France
karel.brinda@univ-mlv.fr

Several algorithms for similarity search employ seeding techniques to quickly discard very dissimilar regions. In this paper, we study theoretical properties of lossless seeds, i.e., spaced seeds having full sensitivity. We prove that lossless seeds coincide with languages of certain sofic subshifts, hence they can be recognized by finite automata. Moreover, we show that these subshifts are fully given by the number of allowed errors k and the seed margin ℓ. We also show that for a fixed k, optimal seeds must asymptotically satisfy $\ell \in \Theta(m^{\frac{k}{k+1}})$.

1 Introduction

The annual volume of data produced by the Next-Generation Sequencing technologies has been rapidly increasing; even faster than growth of disk storage capacities. Thus, new efficient algorithms and data-structures for processing, compressing and storing these data, are needed.

Similarity search represents the most frequent operation in bioinformatics. In huge DNA databases, a two-phase scheme is the most widely used approach to find all occurrences of a given string up to some Hamming or Levenshtein distance. First of all, most of dissimilar regions are discarded in a fast *filtration phase*. Then, in a *verification phase*, only "hot candidates" on similarity are processed by classical time-consuming algorithms like Smith-Waterman [23] or Needleman-Wunsch [17].

Algorithms for the filtration phase are often based on so-called *seed filters* which make use of the fact that two strings of the same length m being in Hamming distance k must necessarily share some exact patterns. These patterns are represented as strings over the alphabet $\{\texttt{\#},\texttt{-}\}$ called *seeds*, where the "matching" symbol # corresponds to a matching position and the "joker" symbol - to a matching or a mismatching position.

For instance, for two strings of length 15, matching within two errors, shared patterns are, e.g., `##-#--##-#` or `#####`. For illustration, if we consider that two strings match as `===X=====X=====` (where the symbols = and X represent respectively matching and mismatching positions), then the corresponding seed positions can be following:

```
===X=====X=====
.##-#--##-#....
....#####......
```

As the second seed is the longest possible contiguous seed in this case, we observe the main advantage of spaced seeds in comparison to contiguous seeds: for the same task, there exist spaced seeds with higher number of #'s (so-called *weight*).

Two basic characteristics of every seed are *selectivity* and *sensitivity*. Selectivity measures restrictivity of a filter created from the seed. In general, higher weight implies better selectivity of the filter. *Lossless seeds* are those seeds having full sensitivity. They are easier to handle mathematically on one hand, but attain lower weight on the other hand. *Lossy seeds* are employed for practical purposes more since a small decrease in sensitivity can be compensated by considerable improvement of selectivity.

Nevertheless, only lossless seeds are considered in this paper. For a given length m of strings to be compared and a given number of allowed mismatches k (such setting is called (m,k)*-problem*), the aim is to design fully sensitive seeds with highest possible weight.

1.1 Literature

The idea of lossless seeds was originally introduced by Burkhardt and Karkkäinen [3, 4]. Let us remark that lossy spaced seed were used in the same time in the PatternHunter program [16]. Generalization of lossless seeds was studied by Kucherov et al. [11]. given seed are required (the pattern is shared at more positions). The authors also proved that, for a fixed number k of mismatches, *optimal seeds* (i.e., seeds with the highest possible weight among all seeds solving the given problem) must asymptotically satisfy $m - w(m) \in \Theta(m^{\frac{k}{k+1}})$, where $w(m)$ denotes the maximal possible weight of a seed solving the (m,k)-problem. They also started a systematic study of seeds created by repeating of short patterns. Afterwards, the results on asymptotic properties of optimal seeds were generalized by Farach-Colton et al. [10]. Computational complexity of optimal seed construction was derived by Nicolas and Rivals [18, 19].

Further, the theory on lossless seed was significantly developed by Egidi and Manzini. First, they studied seeds designed from mathematical objects called perfect rulers [6, 9]. The idea of utilization of some type of "rulers" was later independently extended by KB [2] (cyclic rulers) and, again, Edigi and Manzini [8] (difference sets). In [8], these ideas were extended also to seed families. Cyclic rulers and difference sets mathematically correspond to each other. Edigi and Manzini [7] also showed possible usage of number-theoretical results on quadratic residues for seed design.

In practice, seeds often find their use in short-read mappers implementing hash tables (for more details on read mapping, see, e.g., [12, 22]). ZOOM [13] and PerM [5] are examples of mappers utilizing lossless seeds.

A list of papers on spaced seed is regularly maintained by Noé [20].

1.2 Our object of study

One of the most important theoretical aspects of lossless seeds are their structural properties. Whereas good lossy seeds usually show irregularity, it was observed that good lossless seeds are often repetitions of short patterns ([11, 5, 2, 8]). The question whether optimal seeds can be constructed in all cases by repeating patterns, which would be short with respect to seed length, remains open (see [2, Conjecture 1]). Its answering would have practical impacts in development of bioinformatical software tools since the search space of programs for lossless seeds design could be significantly cut and also indexes in programs using lossless seeds for approximate string matching could be more memory efficient (like [5]).

1.3 Paper organization and results

In this paper, we follow and further develop ideas from [2]. We concentrate on a parameter ℓ called seed margin, which is the difference between the size m of compared strings and the length of a seed.

In Section 2 we recall the notation used in combinatorics on words and symbolic dynamics. In Section 3 we formally define seeds and (m,k)-problems. Then we transform the problem of seed detection into another criterion (Theorem 1) and also show asymptotic properties of ℓ for optimal seeds (Proposition 1). In Section 4 we prove that sets of seeds, obtained by fixing the parameters k and ℓ, coincide with languages of some sofic subshifts. Therefore, those sets of seeds are recognized by finite automata. In Section 5 we show applications of obtained results for seed design. These results provide a new view on lossless seeds and explain their periodic properties.

2 Preliminaries

Throughout the paper, we use a standard notation of combinatorics on words and symbolic dynamics.

2.1 Combinatorics on words

An *alphabet* $\mathcal{A} = \{a_0, \dots, a_{m-1}\}$ is a finite set of symbols called *letters*. In this paper, we will work exclusively with the alphabet $\{\texttt{\#}, \texttt{-}\}$ A finite sequence of letters from $\mathcal{A}$ is called a *finite word* (over $\mathcal{A}$). The set $\mathcal{A}^*$ of all finite words (including the empty word ε) provided with the operation of concatenation is a free monoid. The concatenation is denoted multiplicatively. If $w = w_0 w_1 \cdots w_{n-1}$ is a finite word over $\mathcal{A}$, we denote its length by $|w| = n$. We deal also with bi-infinite sequences of letters from $\mathcal{A}$ called *bi-infinite words* $\mathbf{w} = \cdots \mathbf{w}_{-2}\mathbf{w}_{-1}|\mathbf{w}_0\mathbf{w}_1\mathbf{w}_2 \cdots$ over $\mathcal{A}$. The sets of all bi-infinite words over $\mathcal{A}$ is denoted by $\mathcal{A}^{\mathbb{Z}}$.

A finite word w is called a *factor* of a word $\mathbf{u}$ ($\mathbf{u}$ being finite or bi-infinite) if there exist words p and s (finite or one-side infinite) such that $\mathbf{u} = pws$. For given indexes i and j, the symbol $\mathbf{u}[i,j]$ denotes the factor $\mathbf{u}_i\mathbf{u}_{i+1}\cdots\mathbf{u}_j$ if $i \le j$, or ε if $i > j$. A concatenation of k words w is denoted by w^k. The set of all factors of a word $\mathbf{u}$ ($\mathbf{u}$ being finite or bi-infinite) is called the language of $\mathbf{u}$ and denoted by $\mathcal{L}(\mathbf{u})$. Its subset $\mathcal{L}(\mathbf{u}) \cap \mathcal{A}^n$ containing all factors of $\mathbf{u}$ of length n is denoted by $\mathcal{L}_n(\mathbf{u})$.

Let us remark that this notation will be used extensively in the whole text. For instance $\mathbf{w}[2,5]\texttt{-}^4$ denotes the word created by concatenation of the factor $\mathbf{w}_2\mathbf{w}_3\mathbf{w}_4\mathbf{w}_5$ of a bi-infinite word $\mathbf{w}$ and the word $\texttt{----}$. Similarly, for a finite word v of length n, by $\cdots\texttt{--}|v\texttt{--}\cdots$ we denote the bi-infinite word $\mathbf{u}$ such that for all $i \in \{0,\dots,n-1\}(\mathbf{u}_i = w_i)$ and for all $i \in \mathbb{Z} \setminus \{0,\dots,n-1\}(\mathbf{u}_i = \texttt{-})$. For more information about combinatorics on words, we can refer to Lothaire I [15].

2.2 Symbolic dynamics

Consider an alphabet $\mathcal{A}$. On the set $\mathcal{A}^{\mathbb{Z}}$ of bi-infinite words over $\mathcal{A}$, we define a so-called Cantor metric d as

$$d(\mathbf{u},\mathbf{v}) = \begin{cases} 0 & \text{if } \mathbf{u} = \mathbf{v}, \\ 2^{-s} & \text{if } \mathbf{u} \neq \mathbf{v}, \text{ where } s := \min\left\{|i| \;\middle|\; \mathbf{u}_i \neq \mathbf{v}_i\right\}. \end{cases}$$

We define a *shift* operation σ as $[\sigma(\mathbf{u})]_i = \mathbf{u}_{i+1}$ for all $i \in \mathbb{Z}$. The map σ is invertible, and the power σ^k is defined by composition for all $k \in \mathbb{Z}$. The map σ is continuous on $\mathcal{A}^{\mathbb{Z}}$, therefore, $(\mathcal{A}^{\mathbb{Z}}, \sigma)$ is a dynamical system, which is called a *full shift*.

A bi-infinite word $\mathbf{u} \in \mathcal{A}^{\mathbb{Z}}$ *avoids* a set of finite words X if $\mathcal{L}(\mathbf{u}) \cap X = \emptyset$. By S_X we denote the set of all bi-infinite words that avoid X and we call it a *subshift*. If X is a regular language, S_X is called *sofic subshift*; if X is finite, S_X is called a *subshift of finite type*. The *language* $\mathcal{L}(S)$ of a subshift S is the union of languages of all bi-infinite words from S. By $\mathcal{L}_n(S)$ we denote the set $\mathcal{L}(S) \cap \mathcal{A}^n$. It holds that a set $S \subseteq \mathcal{A}^{\mathbb{Z}}$ is a subshift if and only if it is invariant under the shift map σ (that means $\sigma(S) = S$) and it is closed with respect to the Cantor metric. A general theory of subshifts is well summarized in [14].

3 Lossless seeds

In this section, we introduce basic definition formalizing lossless seeds. Then we introduce a parameter ℓ called seed margin and show its asymptotic properties for optimal seeds. Let us recall that m denotes the length of strings to be compared and k denotes the number of allowed mismatches.

Definition 1. *The binary alphabet $\mathcal{A} = \{\texttt{\#}, \texttt{-}\}$ is called* seed alphabet. *Every finite word over this alphabet is a* seed. *The* weight *of a seed Q is the number of occurrences of the letter # in Q.*

Definition 2. *Let m and k be positive integers. Every set $\{i_1, \dots, i_k\} \subseteq \{0, \dots, m-1\}$ is called* error combination *of k errors.*

Consider a seed Q such that $|Q| < m$ and denote $\ell := m - |Q|$, which is the so-called seed margin. *Then Q* detects *an error combination $\{i_1, \dots, i_k\} \subseteq \{0, \dots, m-1\}$ at position $t \in \{0, \dots, \ell\}$ if for all $j \in \{0, \dots, |Q|-1\}$ it holds $(Q_j = \texttt{\#} \implies j + t \notin \{i_1, \dots, i_k\})$.*

The seed Q is said to solve *the (m,k)-problem if every error combination $\{i_1, \dots, i_k\} \subseteq \{0, \dots, m-1\}$ of k errors is detected by Q at some position $t \in \{0, \dots, \ell\}$.*

Many combinatorial properties of seeds can be studied from the perspective of bi-infinite words. First, we need a seed analogy of the logical function `OR` applied on bi-infinite words and producing, again, a bi-infinite word.

Definition 3. *Consider k bi-infinite words $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(k)}$ over $\mathcal{A}$. We define a k-nary operation $\oplus$ as*

$$\forall i \in \mathbb{Z}: \quad (\oplus(\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(k)}))_i = \begin{cases} \texttt{\#} & \text{if } (\mathbf{u}^{(j)})_i = \texttt{\#} \text{ for some } j \in \{1, \dots, k\} \\ \texttt{-} & \text{otherwise} \end{cases}$$

The following theorem will be crucial for seed analysis in the rest of the text. It is mainly a translation of basic definitions to the formalism of shifts and logical operations, but it enables us to easily observe on which parameters (and how) the structure of lossless seeds really depends.

Theorem 1. *Let m and k be positive integers and Q be a seed such that $|Q| < m$. Denote $\ell := m - |Q|$ and $\mathbf{w} := \cdots \texttt{--}|\texttt{-}^{\ell} Q \texttt{--} \cdots$. Then Q detects an error combination $\{i_1, \dots, i_k\} \subseteq \{0, \dots, m-1\}$ at a position $t \in \{0, \dots, \ell\}$ if and only if*

$$\Big(\oplus(\sigma^{i_1}(\mathbf{w}), \dots, \sigma^{i_k}(\mathbf{w})\Big)_{\ell - t} = \texttt{-}. \tag{1}$$

Proof. Q detects $\{i_1, \dots, i_k\}$ at position t if $\forall j \in \{0, \dots, |Q|-1\}\big(Q_j = \texttt{\#} \implies j + t \notin \{i_1, \dots, i_k\}\big)$. This is equivalent to $\forall p \in \{i_1, \dots, i_k\}(\mathbf{w}_{p-t+\ell} = \texttt{-})$, which is equivalent to (1). ☐

Corollary 1. *Q does not detect a combination $\{i_1, \dots, i_k\}$ at any position $t \in \{0, \dots, \ell\}$ if and only if $(\oplus(\sigma^{i_1}(\mathbf{w}), \dots, \sigma^{i_k}(\mathbf{w}))[0, \ell] = \texttt{\#}^{\ell+1}$.*

Let us mention that in the case of two errors, Corollary 1 corresponds to the Laser method [2, Section 4.1] (a JavaScript implementation is available at [1]) as we illustrate in the following example.

Example 1. *Consider a seed $Q = \texttt{\#\#-\#-----\#-\#\#}$ of length* 14 *and the $(19, 2)$-problem. In Figure 1 we show a corresponding schematic table. Denote $\mathbf{w} := \cdots \texttt{--}|\texttt{-}^5 Q \texttt{--} \cdots$. The words $\oplus(\sigma_i(\mathbf{w}), \sigma_j(\mathbf{w}))$ occur diagonally. It is easily seen from Corollary 1 that Q does not detect the error combination $\{5, 13\}$ since $\ell = 5$ and $\big(\oplus(\sigma^5(\mathbf{w}), \sigma^{13}(\mathbf{w}))\big)[0, 5] = \texttt{\#}^{\ell+1}$.*

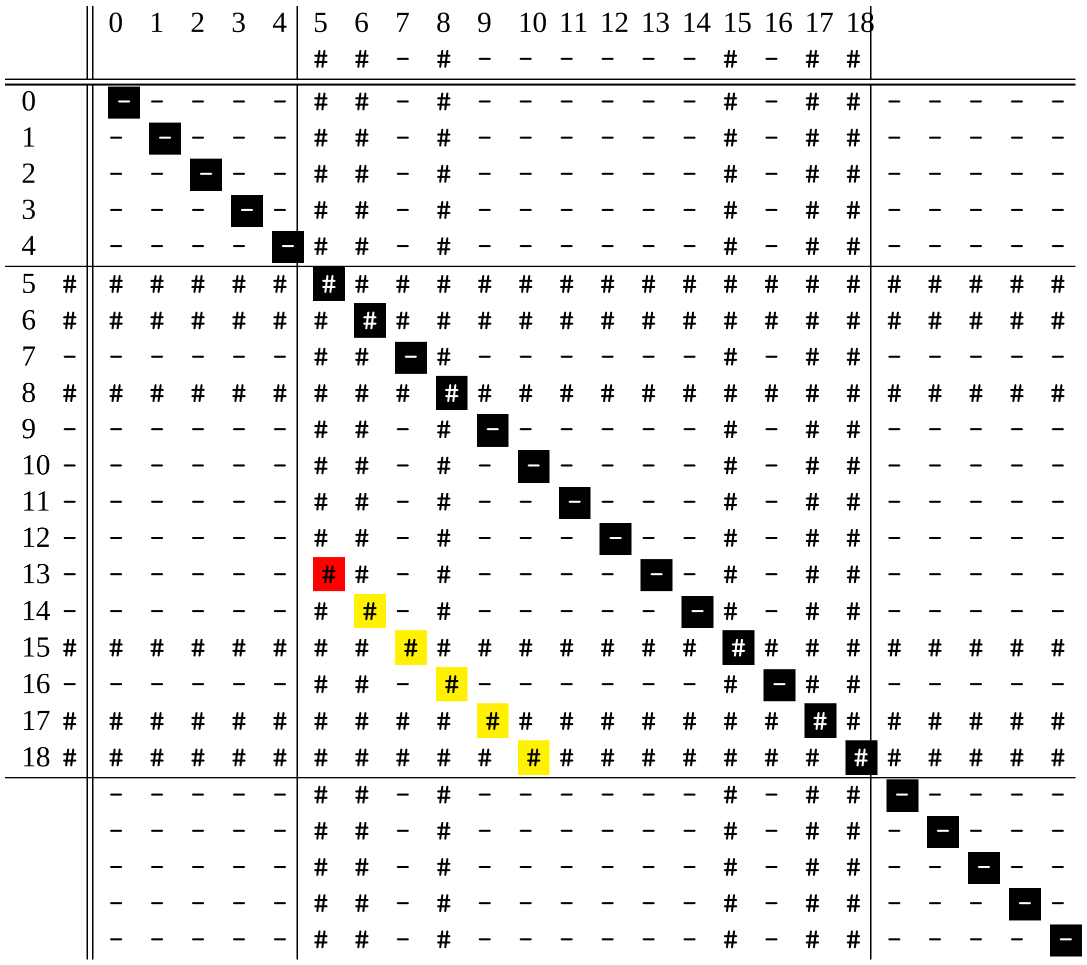


Figure 1: The Laser method for the $(19,2)$-problem and the seed $Q =$ ##-#------#-## in Example 1.

Even though the basic parameters in the concept of (m,k)-problems are m and k; as follows from Theorem 1, the parameters determining structure of seeds are ℓ and k. Therefore, in the next section, we will fix them and study seeds Q solving $(|Q|+\ell,k)$-problems. Hence, when increasing m, the seed must be extended in order to keep ℓ constant.

To complete this section, we show the asymptotic relation of m, ℓ, and k for optimal seeds. Let us fix the parameter k. Let $w(m)$ denote the maximal weight of a seed solving the (m,k)-problem. It was proved in [11, Lemma 4] that $m - w(m) \in \Theta(m^{\frac{k}{k+1}})$. We show that ℓ has the same asymptotic behavior.

Proposition 1. *Let k be a fixed positive integer and $w(m)$ denote the maximal weight of a seed solving the (m,k)-problem. Let $H(m)$ be the set of all seeds with weight $w(m)$ solving the (m,k)-problem. For every positive m, set $\ell(m) := m - |Q|$, where Q is an arbitrary seed from $H(m)$. Then $\ell(m) \in \Theta(m - w(m))$.*

Proof. Since $m \geq \ell(m) + w(m)$, we get trivially the upper bound as $\ell(m) \in \mathcal{O}(m - w(m))$. Now let us prove the lower bound. Let k be fixed. Since $m - w(m) \in \Theta(m^{\frac{k}{k+1}})$ for optimal seeds, it also holds that $(m - w(m))^{k+1} \in \mathcal{O}(m^k)$. From combinatorial considerations on seed detection, we get $\binom{m}{k} \leq \binom{m-w(m)}{k}(\ell+1)$. By combining the last two formulas, we obtain $\ell(m) \in \Omega(m - w(m))$, which concludes the proof. $\square$

4 Seed subshifts

In this section, we show the relation between lossless seeds and subshifts. First, we denote sets of seeds obtained by fixing the parameters ℓ and k. Afterwards, we prove that they coincide with languages of certain sofic subshifts. After defining functions checking the criterion given by Corollary 1 globally on bi-infinite words, we show that the subshift are exactly the sets of bi-infinite words, for which these functions have the upper bound ℓ.

Definition 4. *Let ℓ and k be positive integers. The set of all seeds such that each seed Q solves the $(|Q|+\ell,k)$-problem is denoted by* Seed_k^ℓ.

Example 2. $\mathrm{Seed}_2^3 = \{\varepsilon, \#, -, \#-, -\#, --, \#--, -\#-, --\#, ---, \#--\#, \#---, -\#--, --\#-, ---\#, ----, \dots\}$.

4.1 Functions sh_k and (ℓ,k)-valid bi-infinite words

Definition 5. *Consider a positive integer k. We define a function* $\mathrm{sh}_k : (\mathcal{A}^{\mathbb{Z}})^k \to \mathbb{N}_0 \cup \{+\infty\}$ *as:*

$$\mathrm{sh}_k(\mathbf{u}^{(1)},\dots,\mathbf{u}^{(k)}) = \sup_{i_1,\dots,i_k\in\mathbb{Z}} \sup_{p\in\mathbb{N}_0} \left\{ p \mid \mathbf{v}[0,p-1] = \#^p,\ \textit{where } \mathbf{v} = \oplus\left(\sigma^{i_1}(\mathbf{u}^{(1)}),\dots,\sigma^{i_k}(\mathbf{u}^{(k)})\right)\right\}. \quad (2)$$

We extend the range of the function $\mathrm{sh}_k(\cdot,\dots,\cdot)$ to $(\mathcal{A}^*)^k$. Finite words w are transformed into bi-infinite words $\mathbf{v}$ as $\mathbf{v} := \cdots --|w--\cdots$.

Informally said, $sh_k(\mathbf{u}^{(1)},\dots,\mathbf{u}^{(k)})$ is equal to

- a finite $s \in \mathbb{N}_0$ if after arbitrary "aligning" of the words followed by the logical OR operation (in the Laser method the diagonal bi-infinite words), each run of #'s has length at most s and the value s is attained for some "alignment";
- $+\infty$ if there exists an "alignment" with run of infinitely many #'s (e.g., $\mathrm{sh}_2(\cdots vv|vv\cdots, \cdots ww|ww\cdots)$ with $v = \#\#-$ and $w = \#--$).

Every function sh_k is symmetric and shift invariant with respect to all variables. The following observations show how to estimate their values for given k bi-infinite words.

Observation 1 (Lower estimate)**.** *Let $\mathbf{u}^{(1)},\dots,\mathbf{u}^{(k)}$ be bi-infinite words. If $\oplus(\sigma^{i_1}(\mathbf{u}^{(1)}),\dots,\sigma^{i_k}(\mathbf{u}^{(k)}))$ has a factor $\#^p$ for some $i_1,\dots,i_k$; then $\mathrm{sh}_k(\mathbf{u}^{(1)},\dots,\mathbf{u}^{(k)}) \geq p$.*

Observation 2 (Upper estimate)**.** *Let $\mathbf{u}^{(1)},\dots,\mathbf{u}^{(k)},\mathbf{v}^{(1)},\dots,\mathbf{v}^{(k)}$ be bi-infinite words such that $\mathbf{u}^{(1)} \preceq \mathbf{v}^{(1)},\dots,\mathbf{u}^{(k)} \preceq \mathbf{v}^{(k)}$, where $\preceq$ is a relation defined as*

$$\mathbf{u} \preceq \mathbf{v} \qquad \Longleftrightarrow \qquad (\mathbf{u}_i = \# \implies \mathbf{v}_i = \#) \ \textit{ holds for all } i \in \mathbb{Z}. \quad (3)$$

Then $\mathrm{sh}_k(\mathbf{u}^{(1)},\dots,\mathbf{u}^{(k)}) \leq \mathrm{sh}_k(\mathbf{v}^{(1)},\dots,\mathbf{v}^{(k)})$.

Bi-infinite words for which the sh_k function is bounded by some ℓ, will be the "bricks" of our subshifts. Their factors Q are exactly those seeds solving $(|Q|+\ell,k)$-problems.

Definition 6. *A bi-infinite word $\mathbf{u}$ satisfying $\mathrm{sh}_k(\mathbf{u},\dots,\mathbf{u}) \leq \ell$ is called an $(\ell;k)$*-valid bi-infinite word. *For fixed positive integers ℓ and k, we denote the set of all $(\ell;k)$-valid words by* V_k^ℓ.

Lemma 1. *A seed Q solves the $(|Q|+\ell,k)$-problem if and only if it is a factor of an (ℓ,k)-valid bi-infinite word.*

Proof. $\Longrightarrow$: The word $\mathbf{w} := \cdots --|-^\ell Q--\cdots$ must be (ℓ,k)-valid since otherwise Q would not solve the $(|Q|+\ell,k)$-problem by Corollary 1.

$\Longleftarrow$: For a contradiction assume that there exists a factor Q of a bi-infinite word $\mathbf{u}$, which does not solve the $(|Q|+\ell,k)$-problem. Let the non-detected error combination be $\{i_1,\dots,i_k\}$. Denote $\mathbf{w} = \cdots\texttt{-}\texttt{-}|\texttt{-}^{\ell}Q\texttt{-}\texttt{-}\cdots$.

We use shift invariance of sh_k and Observation 2 to get

$$\mathrm{sh}_k(\mathbf{w},\dots,\mathbf{w}) \le \mathrm{sh}_k(\mathbf{u},\dots,\mathbf{u}) \le \ell. \tag{4}$$

Since Q does not detect the error combination $\{i_1,\dots,i_k\}$, it follows from Corollary 1 that

$$(\oplus(\sigma^{i_1}(\mathbf{w}),\dots,\sigma^{i_k}(\mathbf{w}))[0,\ell] = \texttt{\#}^{\ell+1}.$$

Nevertheless, this gives us a lower estimate on $\mathrm{sh}_k(\mathbf{w},\dots,\mathbf{w})$, which is contradicting (4). □

4.2 Subshifts of (ℓ,k)-valid words

The property of (ℓ,k)-validity is preserved under the shift operation. Moreover, the sets V_k^{ℓ} of (ℓ,k)-valid words are subshifts. To prove it, we need to find a criterion for verifying (ℓ,k)-validity based on comparing finite factors of a given bi-infinite word.

Lemma 2. *Let* $\mathbf{u}$ *be a bi-infinite word over the seed alphabet* $\mathcal{A}$. *Then the following statements are equivalent:*

1. $\mathbf{u}$ *is* $(\ell;k)$*-valid;*
2. $\forall v^{(1)},\dots,v^{(k)} \in \mathcal{L}_{\ell+1}(\mathbf{u})\big(\mathrm{sh}_k(v^{(1)},\dots,v^{(k)}) \le \ell\big)$*;*
3. $\forall w^{(1)},\dots,w^{(k)} \in \mathcal{L}_{\ell+1}(\mathbf{u})\big(\oplus(w^{(1)},\dots,w^{(k)}) \ne \texttt{\#}^{\ell+1}\big)$*.*

Proof. We prove three implications.

$1 \Longrightarrow 2$: Consider any such factors $v^{(1)},\dots,v^{(k)}$. Find their positions $i_1,\dots,i_k$ in $\mathbf{u}$. It holds that

$$\cdots\texttt{-}\texttt{-}|v^{(1)}\texttt{-}\texttt{-}\cdots \preceq \sigma^{i_1}(\mathbf{u}),\ \ \dots,\ \ \cdots\texttt{-}\texttt{-}|v^{(k)}\texttt{-}\texttt{-}\cdots \preceq \sigma^{i_k}(\mathbf{u}),$$

where $\preceq$ is the relation defined by (3). By combining the assumption, shift invariance of sh_k, and Observation 2, we obtain $\mathrm{sh}_k(v^{(1)},\dots,v^{(k)}) \le \mathrm{sh}_k(\mathbf{u},\dots,\mathbf{u}) \le \ell$.

$2 \Longrightarrow 3$: It is an easy consequence of the definition of the sh_k function.

$3 \Longrightarrow 1$: For a contradiction assume that $\mathbf{u}$ is not (ℓ,k)-valid. Then there exist integers $i_1,\dots,i_k$ such that $\oplus(\sigma^{i_1}(\mathbf{u}),\dots,\sigma^{i_1}(\mathbf{u}))[0,\ell] = \texttt{\#}^{\ell+1}$. □

The main consequence of Lemma 2 is the fact that every seed must be constructed from reciprocally compatible tiles of length $\ell+1$. To describe this property, we define a relation of compatibility on the set $\mathcal{A}^{\ell+1}$.

Definition 7. *For given positive integers* ℓ *and* k*, we define the* k*-nary* compatibility relation C_k^{ℓ} *on* $\mathcal{A}^{\ell+1}$ *as*

$$\mathrm{C}_k^{\ell}(v^{(1)},\dots,v^{(k)}) \quad\Longleftrightarrow\quad \mathrm{sh}_k(v^{(1)},\dots,v^{(k)}) \le \ell.$$

Corollary 2. *Let* $\mathbf{u}$ *be a bi-infinite word over the seed alphabet* $\mathcal{A}$. *The word* $\mathbf{u}$ *is* (ℓ,k)*-valid if and only if* $\forall v^{(1)},\dots,v^{(k)} \in \mathcal{L}_{\ell+1}(\mathbf{u})\left(\mathrm{C}_k^{\ell}(v^{(1)},\dots,v^{(k)})\right)$.

Now let us prove that (ℓ,k)-valid words really form subshifts. We only need to show that (ℓ,k)-valid words are exactly those words, which can be created from compatible “tiles”.

Lemma 3. *Let ℓ and k be positive integers. The set V_k^ℓ of all (ℓ,k)-valid words is a subshift.*

Proof. We prove the lemma by construction of a set X of forbidden words (as they are introduced in 2.2). Take

$$X := \left\{x \in \mathcal{A}^* \mid \exists v^{(1)},\dots,v^{(k)} \in \mathcal{L}_{\ell+1}(x)\left(\neg \mathrm{C}_k^\ell(v^{(1)},\dots,v^{(k)})\right)\right\}.$$

The set X contains all possible finite words having some factors, which are "incompatible" with respect to the given ℓ and k. Hence, the subshift S_X contains exactly all bi-infinite words $\mathbf{u}$ satisfying $\forall v_1,\dots,v_k \in \mathcal{L}_{\ell+1}(\mathbf{u})\left(\mathrm{C}_k^\ell(v_1,\dots,v_k)\right)$ and we obtain $S_X = \mathrm{V}_k^\ell$ by Corollary 2. □

Example 3. *Even though both of the seeds $Q^{(1)} =$ ##-#-- and $Q^{(2)} =$ --#-## solve the $(11,2)$-problem, the seed $Q = Q^{(1)}$--$Q^{(2)}$ does not solve the $(19,2)$-problem as we have seen in Example 1. Since $Q^{(1)} \oplus Q^{(2)} = \#^6$, any seed $\tilde{Q}$ of the form $\tilde{Q} = Q^{(1)}\text{-}^p Q^{(2)}$ cannot solve the $(|\tilde{Q}|+5,2)$-problem.*

It follows from the last example that the subshift V_2^5 of all $(5,2)$-valid words is not of finite type. Nevertheless, every subshift V_k^ℓ must be a union of subshifts of finite type, which can be constructed from so-called (ℓ,k)-generating sets.

Definition 8. *For given positive integers ℓ and k, a subset G of $\mathcal{A}^{\ell+1}$ is called* (ℓ,k)-generating set *if the following conditions are satisfied:*

1. *for all $v^{(1)},\dots,v^{(k)} \in G$, it holds $\mathrm{C}_k^\ell(v^{(1)},\dots,v^{(k)})$;*
2. *it is maximal possible (i.e., it cannot contain any other word from $\mathcal{A}^{\ell+1}$).*

Observation 3. *Let us take a word from an (ℓ,k)-generating set G. If we remove the last or the first letter and concatenate the letter - to the beginning or to the end of the word, we obtain again a word from G. Therefore, every (ℓ,k)-generating set G must contain, e.g., the word $\text{-}^{\ell+1}$.*

Every generating set G fully determines a subshift of finite type, we will denote it by $S(G)$. This subshift contains all bi-infinite words $\mathbf{u}$ such that $\mathcal{L}_{\ell+1}(\mathbf{u}) \subseteq G$.

Definition 9. *Consider a seed Q and an (ℓ,k)-generating set G. By $S(G)$, we denote the subshift S_X of finite type given by $X = \mathcal{A}^{\ell+1}\backslash G$. We say that a seed Q is* generated *by G if $Q \in \mathcal{L}(S(G))$.*

In other words, a seed Q satisfying $|Q| \geq \ell+1$ is generated by G if $\mathcal{L}_{\ell+1}(Q) \subseteq G$. A seed Q such that $|Q| < \ell+1$ is generated by G if $\exists w \in G\big(Q \in \mathcal{L}(w)\big)$. We can also observe that every (ℓ,k)-valid word is generated by some (ℓ,k)-generating set.

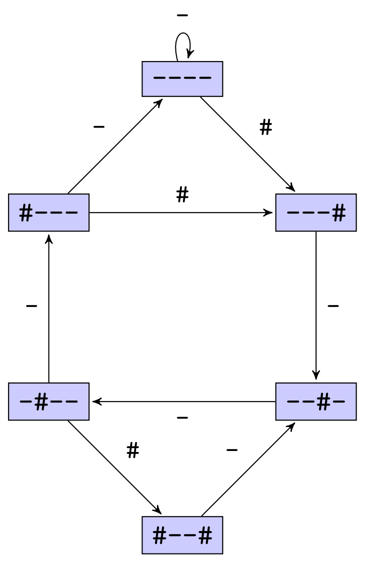
Observation 4. *For every (ℓ,k)-valid bi-infinite word $\mathbf{u}$, there exists an (ℓ,k)-generating set G such that $\mathbf{u} \in S(G)$.*

Example 4. *Continue with the setting from Example 2. Consider the only one $(3,2)$-generating set $G = \{$#--#, #---, -#--, --#-, ---#, ----$\}$. Since $S(G)$ is of finite type, it follows from theory of symbolic dynamics that there exists a strongly connected labeled graph H such that $S(G)$ coincide with labels of all bi-infinite paths in H (for details, see [14]). This graph also determines a finite automaton recognizing the set $\mathcal{L}(S(G))$, i.e., the set of labels of finite paths in H. Such automaton can be created from a de-Bruijn graph. However, it would not be minimal as it is shown in Figure 2.*

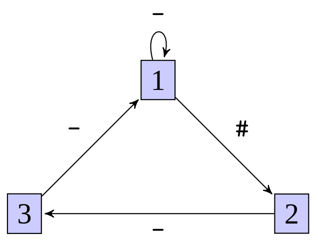
Theorem 2. *Let k and ℓ be positive integers. The set* Seed_k^ℓ *is a regular language.*

Proof. There can be only finite number of (ℓ,k)-generating sets; denote them $G_1,\dots,G_d$. It follows from Observation 4 that $S(G_1)\cup\ldots\cup S(G_d) = \mathrm{V}_k^\ell$ and, from Lemma 1, we know that $\mathcal{L}(\mathrm{V}_k^\ell) = \mathrm{Seed}_k^\ell$.

For every $i \in \{1,\dots,d\}$, the set $S(G_i)$ is a subshift of finite type, so every set $\mathcal{L}(S(G_i))$ is a regular language. Since the set Seed_k^ℓ is a union of finitely many regular languages, it is a regular language. □



(a) A graph created as a de-Bruijn graph from the set of vertices G.



(b) The previous graph after minimization.

Figure 2: Labeled graphs H for the subshift $S(G)$ in Example 4.

5 Application for seed design

In this section, we describe how to design seeds with knowledge of an (ℓ,k)-generating set. Then we show how to search $(\ell,1)$ and $(\ell,2)$-generating sets.

5.1 Seed design using generating sets

Let us have an (ℓ,k)-generating set G and let us consider a task of designing a seed Q of length s, which would solve the $(\ell+s,k)$-problem. If $s \leq \ell+1$, we can take an arbitrary factor of length s of any word from G.

If $s > \ell+1$, we need to construct the seed in $s-\ell$ steps by extending letter by letter. In the first step, we take an arbitrary word $w \in G$ and set $Q := w$. In every other step, we take any word w from G such that the last ℓ letters of Q are equal to ℓ first letters of w and concatenate the last letter of w to Q. Existence of such word w is guaranteed since we can use at least the letter - in every step.

5.2 Generating sets for $k = 1$

As a simple consequence of Corollary 1, we get a full characterization of all seeds solving $(m,1)$-problems ([2, Theorem 5]).

Proposition 2. $\mathrm{Seed}_1^\ell = \{Q \in \mathcal{A}^* \mid \#^{\ell+1}$ *is not a factor of* $Q\}$

Proof. Denote $\mathbf{v} = \cdots\text{--}|\text{-}^\ell Q\text{--}\cdots$ and $\ell = m - |Q|$. Then from Corollary 1 follows that: Q solves the $(m,1)$-problem $\iff \forall i \in \mathbb{Z}\left((\sigma^i(\mathbf{v}))[0,\ell] \neq \#^{\ell+1}\right) \iff Q$ does not contain $\#^{\ell+1}$. ☐

Thus, for every positive ℓ, the only $(\ell,1)$-generating set is $\mathcal{A}^{\ell+1} \setminus \{\#^{\ell+1}\}$, i.e., the set of all words of length $\ell+1$ except $\#^{\ell+1}$.

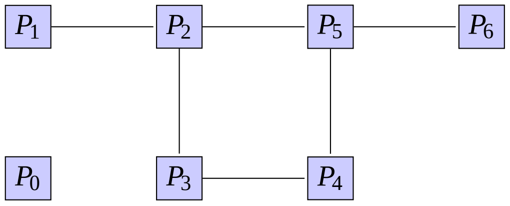


Figure 3: The simplified graph of sets of equivalent seeds for $(5,2)$-generating sets search in Example 6.

5.3 Generating sets for $k = 2$

Let $k = 2$ and ℓ be an arbitrary fixed positive integer. We can derive all $(\ell,2)$-generating sets using graph theoretical methods by transformation to independent sets search. Let $V := \{w^{(1)},\dots,w^{(q)}\}$ denote the set of all seeds of length $\ell+1$ solving the $(2\ell+1,2)$-problem. Consider a graph R given by the adjacency matrix $(M_R)_{i,j} = \begin{cases} 0 & \text{if } \mathrm{C}_2^{\ell}(w^{(i)},w^{(j)}), \\ 1 & \text{otherwise}. \end{cases}$

Then the generating sets are "maximal" independent sets (maximal with respect to inclusion) in the graph R. We require maximality here since it is already required by the second property in Definition 8.

We can partially simplify the graph R. We say that two vertices v and w in this graph are equivalent if $\forall x \in V \left(\mathrm{C}_k^{\ell}(x,v) \iff \mathrm{C}_k^{\ell}(x,w)\right)$. Then we can put all equivalent vertices into one vertex, i.e., every vertex will contain a set of words instead of only one word. The step with searching "maximal" independent sets stays unchanged.

Example 5. *Let $k = 2$. For every $\ell \in \{1,\dots,4\}$, all seeds solving the $(\ell+1,2)$-problem are mutually compatible, which means that there exists a unique $(\ell,2)$-generating set. We list them out in the following table.*

ℓ	G
1	{--}
2	{#--, -#-, --#, ---}
3	{#--#, #---, -#--, --#-, ---#, ---}
4	{##---, -##--, --##-, ---##, #-#--, -#-#-, --#-#, #--#-, -#--#, #---#, #----, -#---, --#--, ---#-, ----#, -----}

Example 6. *Let $k = 2$ and $\ell = 5$. We find the graph R by the procedure above. After its simplification, we obtain the graph in Figure 3, where*

$P_0 = \{$------; -----#, ----#-, ---#--, --#---, -#----, #-----;
----##, ---##-, --##--, -##---, ##----;
---#-#, --#-#-, -#-#--, #-#---;
--#--#, -#--#-, #--#--; -#---#, #---#-;
#---##; ##---#; #----#$\}$,

$P_1 = \{$#--#-#$\}$, $P_2 = \{$--##-#, -##-#-, ##-#--$\}$,

$P_3 = \{$-##--#, ##--#-$\}$, $P_4 = \{$-#--##, #--##-$\}$,

$P_5 = \{$--#-##, -#-##-, #-##--$\}$, $P_6 = \{$#-#--#$\}$.

By finding "maximal" independent sets in the graph in Figure 3, we get all $(5,2)$*-generating sets:*

$$G_1 = P_0 \cup P_1 \cup P_3 \cup P_5, \qquad G_2 = P_0 \cup P_1 \cup P_3 \cup P_6,$$
$$G_3 = P_0 \cup P_2 \cup P_4 \cup P_6, \qquad G_4 = P_0 \cup P_1 \cup P_4 \cup P_6.$$

To conclude the section, let us remark that a similar derivation can be done using hypergraphs also for $k > 2$.

6 Conclusion

In this paper, we have studied lossless seeds from the perspective of symbolic dynamics. We have concentrated on the seed margin ℓ defined as a difference of the length m of compared strings and the length of a seed. We have derived asymptotic behavior of ℓ for optimal seeds (Proposition 1), which must satisfy $\ell \in \Theta(m^{\frac{k}{k+1}}) = \Theta(m - w(m))$. We have shown another criterion for errors detection by seeds (Theorem 1). From this criterion we have proved that lossless seeds coincide with languages of certain sofic subshifts, therefore, they are recognized by finite automata (Theorem 2). We have presented that these subshifts are fully given by the number of allowed errors k and the seed margin ℓ and that they can be further decomposed into subshifts of finite type.

These facts explain why periodically repeated patterns often appear in lossless seeds. This is caused by the fact that these patterns correspond to cycles in recognizing automata (which correspond to seeds for cyclic (m,k)-problems in [11]). Nevertheless, it remains unclear what is the upper bound on the length of cycles to obtain at least some optimal seeds. In the case case $k = 2$, it was conjectured in [2, Conjecture 1] that it is sufficient to consider patterns having length at most $\ell + 1$ to obtain some of optimal seeds.

Acknowledgements. The author is supported by the ABS4NGS grant of the French government (program *Investissement d'Avenir*). He is grateful to Gregory Kucherov and Karel Klouda for helpful ideas. He also thanks three anonymous referees for valuable comments.

References

[1] Karel Břinda (2013): *Laser method on-line*. Available at `http://brinda.cz/laser-method/`.

[2] Karel Břinda (2013): *Lossless seeds for approximate string matching*. Master's thesis, FNSPE Czech Technical University in Prague, Czech Republic. Available at `http://brinda.cz/publications/diplomka.pdf`.

[3] Stefan Burkhardt & Juha Kärkkäinen (2001): *Better Filtering with Gapped q-Grams*. In: *Proceedings of the 12th Symposium on Combinatorial Pattern Matching (CPM), Lecture Notes in Computer Science* 2089, Springer, pp. 73–85, doi:10.1007/3-540-48194-X_6.

[4] Stefan Burkhardt & Juha Kärkkäinen (2002): *Better filtering with gapped q-grams*. *Fundamenta Informaticae* 56(1-2), pp. 51–70.

[5] Yangho Chen, Tate Souaiaia & Ting Chen (2009): *PerM: efficient mapping of short sequencing reads with periodic full sensitive spaced seeds*. *Bioinformatics* 25(19), pp. 2514–2521, doi:10.1093/bioinformatics/btp486.

[6] Lavinia Egidi & Giovanni Manzini (2011): *Spaced Seeds Design Using Perfect Rulers*. In: *Proceedings of the 18th International Symposium on String Processing and Information Retrieval (SPIRE), Pisa (Italy), Lecture Notes in Computer Science* 7024, Springer, pp. 32–43, doi:10.1007/978-3-642-24583-1_5.

[7] Lavinia Egidi & Giovanni Manzini (2013): *Better spaced seeds using quadratic residues*. *Journal of Computer and System Sciences* 79(7), pp. 1144–1155, doi:10.1016/j.jcss.2013.03.002.

[8] Lavinia Egidi & Giovanni Manzini (2014): *Design and analysis of periodic multiple seeds*. *Theoretical Computer Science* 522, pp. 62–76, doi:10.1016/j.tcs.2013.12.007.

[9] Lavinia Egidi & Giovanni Manzini (2014): *Spaced Seeds Design Using Perfect Rulers*. *Fundamenta Informaticae* 131(2), pp. 187–203, doi:10.3233/FI-2014-1009.

[10] Martin Farach-Colton, Gad M. Landau, Süleyman Cenk Sahinalp & Dekel Tsur (2007): *Optimal spaced seeds for faster approximate string matching*. *Journal of Computer and System Sciences* 73(7), pp. 1035–1044, doi:10.1016/j.jcss.2007.03.007.

[11] Gregory Kucherov, Laurent Noé & Mikhail A. Roytberg (2005): *Multiseed lossless filtration*. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 2(1), pp. 51–61, doi:10.1109/tcbb.2005.12.

[12] Heng Li & Nils Homer (2010): *A survey of sequence alignment algorithms for next-generation sequencing*. *Briefings in bioinformatics* 11(5), pp. 473–83, doi:10.1093/bib/bbq015.

[13] Hao Lin, Zefeng Zhang, Michael Q. Zhang, Bin Ma & Ming Li (2008): *ZOOM! Zillions Of Oligos Mapped*. *Bioinformatics* 24(21), pp. 2431–2437, doi:10.1093/bioinformatics/btn416.

[14] Douglas Lind & Brian Marcus (1995): *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, doi:10.1017/CBO9780511626302.

[15] M. Lothaire (1983): *Combinatorics on Words*. *Encyclopedia of Mathematics and its Applications* 17, Addison-Wesley Publishing Co., Reading, Mass., doi:10.1017/CBO9780511566097. Reprinted in the Cambridge University Press, Cambridge, UK, 1997.

[16] Bin Ma, John Tromp & Ming Li (2002): *PatternHunter: Faster and more sensitive homology search*. *Bioinformatics* 18(3), pp. 440–445, doi:10.1093/bioinformatics/18.3.440.

[17] Saul B Needleman & Christian D Wunsch (1970): *A general method applicable to the search for similarities in the amino acid sequence of two proteins*. *Journal of Molecular Biology* 48(3), pp. 443–453, doi:10.1016/0022-2836(70)90057-4.

[18] François Nicolas & Éric Rivals (2005): *Hardness of Optimal Spaced Seed Design*. In A. Apostolico, M. Crochemore & K. Park, editors: *Proceedings of the 16th Annual Symposium on Combinatorial Pattern Matching (CPM), Jeju Island (Korea)*, *Lecture Notes in Computer Science* 3537, Springer, pp. 144–155, doi:10.1007/b137128.

[19] François Nicolas & Éric Rivals (2008): *Hardness of Optimal Spaced Seed Design*. *Journal of Computer and System Sciences* 74(5), pp. 831–849, doi:10.1016/j.jcss.2007.10.001.

[20] Laurent Noé (2014): *Spaced seeds bibliography*. Available at `http://www.lifl.fr/~noe/spaced_seeds.html`.

[21] Laurent Noé & Gregory Kucherov (2005): *YASS: enhancing the sensitivity of DNA similarity search*. *Nucleic Acids Research* 33(suppl. 2), pp. W540–W543, doi:10.1093/nar/gki478.

[22] Paolo Ribeca (2012): *Short-Read Mapping*. In Naiara Rodríguez-Ezpeleta, Michael Hackenberg & Ana M. Aransay, editors: *Bioinformatics for High Throughput Sequencing*, Springer New York, pp. 107–125, doi:10.1007/978-1-4614-0782-9_7.

[23] Temple F. Smith & Michael S. Waterman (1981): *Identification of common molecular subsequences*. *Journal of molecular biology* 147(1), pp. 195–7, doi:10.1016/0022-2836(81)90087-5.