Regularity-preserving letter selections

Armando B. Matos

LIACC, Universidade do Porto

Rua do Campo Alegre 823, 4150 Porto, Portugal

1 Introduction and definitions

Seiferas and McNaughton gave in [SM76] a complete characterization of the family of regularity-preserving prefix removals of regular languages; see also references to previous work in that paper. We generalize these results by studying what kind of algorithms for letter selection preserve regularity.

In Section 2 we characterize subword selection methods based only on the word length. In Section 3 the regularity-preserving property for some special selection algorithms is proved; in particular we show that all ultimately periodic selection algorithms are regularity-preserving. In Section 5 we study sets that may destroy the regularity of a language, that is, sets that are not regularity-preserving. Finally in Section 7 we present the main conclusions of this work and mention some open problems.

Note added in 2006 In [BLC$^+$06] the authors have essentially solved the letter selection problem (also called the "filtering" problem).

1.1 Definitions and notation

The language recognized by a finite automaton A will be denoted by $\mathcal{L}(A)$ and the language represented by the regular expression E by $\mathcal{L}(E)$. We identify a regular expression with the language that it denotes. If Σ is a (finite) alphabet, the set of all semi-infinite words with letters in Σ is denoted by Σ^ω. A (finite) word w of Σ^ω is identified with the a mapping

$$w : \mathcal{N} \to \Sigma$$

where $w(n)$ denotes the nth letter of w; the first letter corresponds to index 0. Let x be a possibly infinite word. We denote by $\mathrm{pref}(x)$ the language of all the finite prefixes of x (including ε).

Let α be some algorithm mapping words into either words or into the special symbol $\perp$ ("undefined"). Notice that the corresponding computation always terminates.

$$\alpha : \Sigma^\star \to \Sigma^\star \cup \{\perp\}$$

This mapping is extended to a function $\alpha : \mathcal{P}(\Sigma^\star) \to \mathcal{P}(\Sigma^\star)$ as follows: let L be some language; $\alpha(L)$ is defined as

$$\alpha(L) = \{\alpha(x) \mid x \in L \wedge \alpha(x) \neq \perp\}$$

The algorithm A is said to *preserve regularity* (or to be *regularity-preserving*) if $\alpha(L)$ is regular whenever L is regular.

A set $A \subseteq \mathcal{N}$ is *ultimately periodic* or *u.p.* if it is finite or if there is a positive integer p such that, for all sufficiently large n

$$n \in A \text{ iff } n + p \in A$$

2 Algorithms for selecting subwords

In this section we consider several methods for selection subwords of a given word.

Definition 1 (Proportional and exact proportional selections) *Let q and r be integers with $q \geq 1$ and $0 \leq r < q$.*

- *The* proportional p_r^q *selection of the word $a_0 a_1 \cdots a_n$ is the word whose successive letters are a_{qi+r} for $i = 0, 1, \ldots, \lfloor \frac{n-r}{q} \rfloor$.*
- *The* exact proportional e_r^q *selection of the word $a_0, a_1, \cdots, a_n$ where $n = kj + r$ for some $j \geq 1$, is the word whose successive letters are a_{ki+r} for $i = 0, 1, \ldots, \frac{n-r}{q}$.*

[**Example**] We have

$$p_0^2(\underline{a}b\underline{a}c\underline{a}c\underline{a}b) = aaaa$$
$$p_0^2(\underline{a}b\underline{a}c\underline{a}c\underline{a}) = aaaa$$
$$e_0^3(\underline{a}bb\underline{a}bc\underline{a}bc) = aaa$$
$$e_2^3(abb\underline{a}bc\underline{a}bc\underline{c}cc) = bcc$$
$$e_0^2(abacaca) = \perp \quad \textit{(because } 8 - 0 = 8 \textit{ is not divisible by 2)}$$

[**End**]

A more general selection method is the following

Definition 2 (Selection by index sets) *Let S be a recursive set of integers and let x be the word $a_0, a_1, \cdots, a_n$. The selection x_S of x by S is the (in general noncontiguous) subword of x formed by the letters having indices in S.*

[**Example**] We have

$$aa\underline{ba}cc\underline{b}_{\{2,3,6,12,100\}} = bab$$

[**End**]

[**Example**] The proportional selection method is also a selection by an index set: for every word w we have

$$p_r^q(w) = w_{\{qi+r \mid i \in \mathcal{N}\}}$$

[**End**]

Although in this work we are mainly interested in selection by index sets, we now characterize the "algorithmic method", a very general selection method. Consider an algorithm α that satisfies the following conditions.

1. Given a word x, the algorithm tests if some condition $p(n)$ depending on $n = |x|$ is satisfied. If it is, the output is the (non-necessarily contiguous) subword of x defined below. If not, the output is $\perp$. In this case we say that $a(x)$ is undefined (a non-standard use of the word "undefined" because the computation terminates).
   When we write "$a(x) = y$" we mean that the condition is satisfied and that the subword selected is y.

2. The selected letters depend only on the length of $|x|$ and not on the individual letters of x. Moreover, we assume that the output of such algorithms is a set of indices $\{i_1, i_2, \cdots, i_k\}$ where every indice is ≥ 0 and $\leq |x| - 1$ and. Assume that $a_1 \leq a_2 \leq \cdots \leq a_k$. If $x = a_0 a_1 \ldots a_n$, we say that the algorithm selects the subword $a_{i_1} a_{i_2} \cdots a_{i_k}$. For instance, if the subword selected from *aabcbccc* is *bb*, then the same algorithm applied the word *bbaacbbb* (which has the same length) must produce the word "*ac*".

We now formalize this method of selecting sub-words.

Definition 3 (Algorithmic selection) *Consider a predicate $p : \mathcal{N} \to \{\mathtt{F}, \mathtt{T}\}$ and a function*

$$s : n \to \mathcal{P}([0..n-1])$$

We say that, if $p(|w|)$ is true, $[p, s]$ selects the the subword of w formed by the sequence of letters of w with indices $s(|w|)$ (by the same order).

These algorithms are partial (in the sense explained above) functions from $\Sigma^\star$ to $\Sigma^\star$. They can be extended to (total) functions mapping languages into languages.

Definition 4 *Let α be a selection algorithm and let L be a language. We define $\alpha(L)$ as the language*

$$\alpha(L) = \{y \mid \exists x \in L,\, \alpha(x) = y,\, \alpha(x) \neq \perp\}$$

Notice that, if no word in L satisfies the condition, $\alpha(L) = \emptyset$.

All the following methods are selection algorithms.

- The "first half" algorithm of [SM76]

$$fh(a_1a_2\cdots a_n) = \begin{cases} a_1a_2\cdots a_{n/2} & \text{if } n \text{ is even} \\ \text{undefined} & \text{if } n \text{ is odd} \end{cases}$$

- The proportional and exact selections as defined in Definition1. As an example we characterize an exact proportional selection with $q = 2$, $r = 1$ by a selection algorithm.

```
function  e_1^2 (x) /* where x = a_0 a_1 ··· a_{n-1} */
if n is odd and n ≥ 2
    i ← 1;
    while i ≤ n − 1
        output i;
        i ← i + 2
else
    output ⊥;
```

- Selections by recursive index sets (see Definition 2).

3 Some index sets that preserve regularity

In this section we show that for certain families of sets, the language L_S (see definition 2 is regular whenever L is regular. The more general result is Theorem 5.

We begin with the selection method e_0^2. Recall that, if L is a language, then

$$e_0^2 = \{a_0a_2a_4\cdots a_{n-2} \mid a_0a_1a_2\cdots a_{n-1} \in L\}$$

We now show that the function e_0^2 is regularity-preserving.

Theorem 1 (e_0^2 preserves regularity) *If L is regular then $e_0^2(L)$ is also regular.*

Proof. If $\varepsilon \in L$, we can write $L = \{\varepsilon\} \cup L'$ where L' is regular and $\varepsilon \notin L'$. As $e_0^2(L) = e_0^2(L')$ we consider only languages not containing ε. Let $A = (S, s_0, F, \Sigma, \delta)$ be a (non-deterministic) finite automaton that recognizes L where $\varepsilon \notin L$, and suppose then that $\varepsilon \notin L$. We define an automaton $A' = (S, s_0, F, \Sigma, \delta')$ and prove that it recognizes $e_0^2(L)$. The transition relation δ' is defined by

$$(s_1, a, s_3) \in \delta' \;\; \Leftrightarrow \;\; \exists s_2 \in S,\; b \in \Sigma \;\; (s_1, a, s_2) \in \delta \;\wedge\; (s_2, b, s_3) \in \delta$$

The states s_1, s_2 and s_3 are not necessarily distinct.

Suppose that A accepts the word $a_0a_1a_2\cdots a_{n-1}$ and that n is even. The accepting path is represented in Figure 1. Then it is easy to see that A' accepts the word $a_0a_2a_4\cdots a_{n-2}$; in fact, by definition of A', we see that all the transitions (s_0, a_0, s_2), (s_2, a_2, s_4),..., (s_{n-2}, a_{n-2}, s_n) are possible in A' – that is, belong to δ'. We see that $e_0^2(L) \subseteq \mathcal{L}(A')$.

Conversely suppose that A' accepts a word $a_0a_2a_4\cdots a_{n-2}$ (the letter indices are obviously arbitrary; for notational convenience we use even numbers as indices). By construction of A', there are in A states s_1, s_3,..., s_n, letters a_1, a_3,..., a_{n-1} and transitions

$$(s_0, a_0, s_1),\ (s_1, a_1, s_2),\ (s_2, a_2, s_3),\ (s_3, a_3, s_4),\ \cdots,\ (s_{n-2}, a_{n-2}, s_{n-1}),\ (s_{n-1}, a_{n-1}, s_n)$$

We conclude that $a_0a_1a_2\cdots a_{n-1} \in L$, so that $\mathcal{L}(A') \subseteq e_0^2(L)$. Then $\mathcal{L}(A') = e_0^2(L)$. The language $e_0^2(L)$, being recognized by a finite automaton, is regular. $\diamond$

Let us now consider the function e_1^2, that is, the subword selection $a_1a_3\cdots a_{n-2}$.

Theorem 2 (e_1^2 preserves regularity) *If L is regular then $e_1^2(L)$ is also regular.*

Proof. The language $e_1^2(L)$ depends only on the words of L whose length is *odd* and ≥ 3. Supposing that $\varepsilon \notin L$ (the case $\varepsilon \in L$ can be handled as in the proof of Theorem 1) the language L can be represented by (where the a_i are the first letters of words in L)

$$L = a_1L_1 \cup a_2L_2 \cup \cdots \cup a_kL_k$$

A word x having a length that is both *odd* and at least 3 belongs to L iff it has the form $x = a_iy$ where $1 \leq i \leq k$ and y is a word of L_i with an even length ≥ 2. That is,

$$e_1^2(L) = a_1e_0^2(L_1) \cup a_2e_0^2(L_2) \cup \cdots \cup a_ke_0^2(L_k)$$

Using Theorem 1 and the fact that the class of regular languages is closed for union and that aL is regular for every regular language L and $a \in \Sigma$, we see that $e_1^2(L)$ is regular. $\diamond$

The following theorem generalizes theorems 1 and 2. The proof is an easy generalization of the corresponding proofs.

Theorem 3 (e_r^q preserves regularity) *Let q and r be integers with $q \geq 1$ and $0 \leq r < q$. If L is regular, then $e_r^q(L)$ is regular.*

To extend this result for proportional selections we need the following lemma.

Lemma 1 (Padding preserves regularity) *Let q be a positive integer and a a letter. Define $pad_a^q(L)$ as the language obtained by putting at the end of every word of L a minimum number of a's so the the length becomes a multiple of q.*

$$pad_a^q(L) = \{xa^r \mid x \in L,\, 0 \leq r < q,\, |x| + r = 0 \mod q\}$$

If L is regular, $pad_a^q(L)$ is regular.

Proof. Let A be an automaton with transitions δ that recognizes L. We define an automaton A' with transitions δ' that recognizes $pad_a^q(L)$. For each state s_i of A, there are q states in A' denoted by $s_{i,j}$ for $0 \leq j < q$ that keep track of the length modulus q of the word read so far; let us denote this length by j. To every transition $(s_i, a, s_k) \in \delta$ there are q transitions $(s_{i,j}, a, s_{k+1(\mathtt{mod} q)}) \in \delta'$

If s_i is a final state in A there are also new states and transitions attached to each $s_{i,j}$ with $j \neq 0$ in A' as follows

$$s_{i,j} \xrightarrow{a} s_{i,j,j+1} \xrightarrow{a} s_{i,j,j+2} \xrightarrow{a} \cdots \xrightarrow{a_{q-1}} s_{i,j,q-1} \xrightarrow{a} s_{i,j,q}$$

Of these states only $s_{i,j,q}$ is final. If $j = 0$ there are no new states added at this stage and $s_{i,j}$ is final in A'. (The total number of states in A' is $nq + fq(1 + \cdots + q - 1)$ where n and f denote respectively the number of states and the number of final states of A). Clearly A' will accept exactly the words of $pad_a^q(L)$. ◇

This lemma can be easily extended for other forms of padding; we can for instance replace a^r by the prefix with length r of a some fixed word w.

Theorem 4 (p_r^q preserves regularity) *Let q and r be integers with $q \geq 1$ and $0 \leq r < q$. If L is regular, then $p_r^q(L)$ is regular.*

Proof. If $r > 0$, we can write the language L as

$$L = F \cup x_0 L_0 \cup x_1 L_1 \cup \cdots \cup x_k L_k$$

where F is finite, all elements of F have length $< k$ and, for $0 \leq i \leq k$, the words x_i have length r and the languages L_i are regular. We have

$$p_r^q(L) = x_0 p_0^q(L_0) \cup x_1 p_0^q(L_1) \cup \cdots \cup x_k p_0^q(L_k)$$

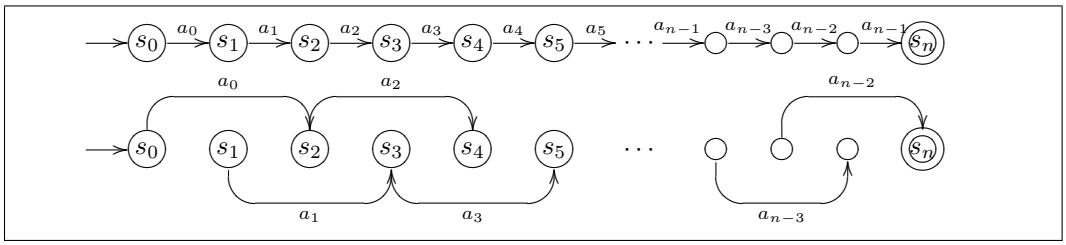

Figure 1: The automaton recognizing L (above) accepts the word $a_0a_1\cdots a_{n-1}$ (with $n \geq 2$ and even) iff the transformed automaton (below) recognizes the word $a_0a_2\cdots a_{n-2}$. The states are not necessarily distinct

So let us consider only the case $r = 0$. We will extend the language L so that the length of every word is a multiple of q. Let us first notice that, for any words x and y such that $|y| < q$ and $|x| + |y|$ is a multiple of q, we have

$$p_0^q(x) = e_0^q(xy) = p_0^q(xy)$$

A simple example of this observation can be seen in Figure 2.

Consider now the language $pad_a^q(L)$ which from Lemma 1 is regular. It follows that $p_0^q(L) = e_0^q(pad(L))$ which is also regular from Theorem 3. ◇

Now a more general result is easy to prove.

Theorem 5 (UP set selection preserves regularity) *Let S be an ultimately periodic set of integers and let L be a regular language. The set selection L_S is regular.*

Proof. Any ultimately periodic set A can be written as an union (see for instance [Mat94])

$$S = F \cup S_1 \cup S_2 \cup \cdots \cup S_k$$

where F is finite and each of the S_i has the form

$$S_i = \{c_i + p_i j \mid j \geq 0\}$$

where for each i with $1 \leq i \leq k$, c_i is an integer, p_i is a positive integer and $c_i < p_i$. Then we have

$$S = L_F \cup L_{S_1} \cup L_{S_2} \cup \cdots \cup L_{S_k}$$

The language L_F is regular because it is finite. For each $1 \leq i \leq k$, the language $L_{S_i} = p_{c_i}^{p_i}(L)$ is regular by Theorem 4. Thus L is regular. ◇

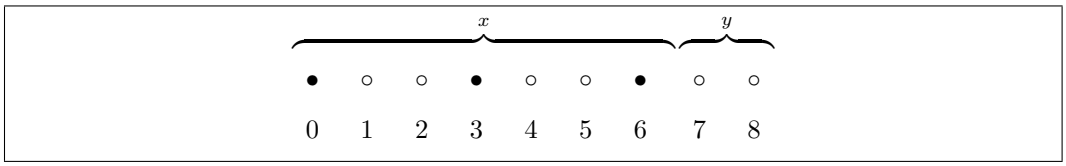

Figure 2: A proportional method with $q = 3$ and $r = 0$, selects the letters of x marked "•". The same letters are selected in the word xy (with length 9) by the exact proportional selection methods (with $q = 3$ and $r = 0$). Symbolically, $p_0^3(x) = e_0^3(xy) = p_0^3(xy)$.

4 Selection by index sets: some properties

In this Section we study some properties of the selection by index sets. These properties may turn out to be useful for the characterization of sets that preserve regularity. First let us state a collection of simple, easy to prove facts.

Theorem 6 *For every languages L and M and set of integers S*

1. $L_\emptyset = \emptyset_S = \emptyset$
2. $(L \cup M)_S = L_S \cup M_S$

Observe that $L_{S\cup T} = L_S \cup L_T$ may be false. Consider for instance the language

$$L = \{(ab)^n \mid n \geq 0\}$$

and let S and T be respectively the set of even integers and the set of odd integers. We have

$$L = L_{\mathcal{N}} = L_{S\cup T} \neq a^\star + b^\star = L_S \cup L_T$$

Notice that for certain regular languages L and non-regularity preserving sets S it may happen that L_S is regular. An extreme example is the regular language $\Sigma^\star$. If S is infinite we always have $\Sigma_S^\star = \Sigma^\star$! To prove this, consider an arbitrary word $w = a_1 a_2 \dots a_k$ and let the set S be $\{n_1, n_2, \cdots\}$ with $n_1 < n_2 < \cdots$. The word w may be obtained by index selection with S in the following word

$$z = x_1 a_1 x_2 a_2 \cdots x_k a_k$$

where x_2, x_2,...,x_k have lengths respectively n_1, $n_2 - 1$, $n_3 - 2$,..., $n_k - k + 1$.

To prove that S is not regularity preserving, we must select some regular language L such that L_S is not regular.

5 Index sets that do not preserve regularity

To prove that the selection by a set S does not preserve regularity we only have to find some regular language L such that L_S is not regular. Let us begin with some examples. In the first we present a proof that a certain set is not regularity preserving.

[**Example**] Consider the language L denoted by the regular expression $(ab)^\star(\varepsilon + a)$. The words of this language are exactly the (finite) prefixes of the infinite word

$$ababababab\cdots$$

that is, $L = \{\varepsilon, a, ab, aba, \cdots\}$. The selection by the set S (to be defined below) results in the language L_S of the prefixes of the infinite word

$$abaabbaaabbbaaaabbbb\cdots$$

The language L_S is not regular. The selection is illustrated in the following diagram

a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b	$\cdots$
0	1	2		4	5		7	8		10		12	13		15		17	$\cdots$

The set S is the following where, for clarity, we have grouped its elements

$$S = \{\langle 0, 1\rangle, \langle 2, 4, 5, 7\rangle, \langle 8, 10, 12, 13, 15, 17\rangle, \langle 18, 20, 22, 24, 25, 27, 29, 31\rangle, \cdots\}$$

[**End**]

[**Example**] Consider the language $L = (abb)^\star$ and the set P of prime numbers. The first letters of $(abb)^\omega$ selected by P are illustrated below.

$$(ab\underline{b}a\underline{b}\underline{b}ab\underline{b}ab\underline{b}\underline{a}bab\underline{b}a\cdots)_{\{2,3,5,7,11,13,17,\cdots\}} = babbbbb\cdots$$

The corresponding infinite word is bab^ω, reflecting the fact that no prime greater than 3 is multiple of 3. We have $L_P = bab^\star$ [**End**]

Although in this example the selection by P preserves the regularity of the language, this is not true in general as the following example suggests.

[**Example**] Consider the language $L' = (aab)^\star$ and the set P of prime numbers. The first letters of $(aab)^\omega$ selected by P are

$$(aa\underline{b}a\underline{b}\underline{a}ba\underline{b}\underline{a}\underline{b}aa\underline{b}\underline{a}baab\underline{a}\underline{b}a\cdots)_{\{2,3,5,7,11,13,17,\cdots\}} = bababababbaababbbaabaabbaba\cdots$$

There is no obvious pattern and L_P does not seem to be regular. [**End**]

[**Example**] Let $S = \{3^n \mid n \geq 0\}$. We have

$$\begin{aligned} ((ab)^\omega)_S &= b^\star \\ ((abc)^\omega)_S &= ba^\star \\ ((abcd)^\omega)_S &= (bd)^\star \\ ((abcde)^\omega)_S &= (bdec)^\star \end{aligned}$$

In fact, for every word $x \neq \varepsilon$, $(x^\star)_S$ is regular. This does not prove that S is regularity-preserving. All possible forms of languages must be considered; for instance

$$(((abc)^\star + (cb)^\star + (abcb)^\star)^\star)$$

must also be regular. [**End**]

Theorem 5 states that, if L is a regular regular and S is an ultimately set of integers, then the language L_S is also regular. The following theorem, which is the most important result of this paper, states that the converse is also true.

Theorem 7 *An index set preserves regularity if and only if it is ultimately periodic.*

We have only to prove the "only if" part: if S is such that L_S is regular whenever L is regular, then S is ultimately periodic.

6 Working Section

Here we establish a number of results that may help to prove Theorem 7. What we want to prove (or disprove) is the "only if" part of the theorem[1]:

Statement 1 *If an index set preserves regularity it is ultimately periodic.*

First let us see if a somewhat weaker condition – a set $S \subseteq \mathcal{N}$ preserves regularity for a certain class of regular languages – is enought to garantee that S is ultimately periodic.

6.1 Repeating a word infinitely

Lemma 2 *Let x and y be words of $\Sigma^\star$ where Σ is a (finite) alphabet. The language pref(yx^ω) of all the finite prefixes of yx^ω is regular.*

[1] A "statement" is a proposition that has not yet been proved. At this stage, Theorem 7 is in fact a "statement".

Proof. As an example from which the general proof easily follows, let us consider the particular words $y = \varepsilon$ and $x = abbb$. The language $\text{pref}(x^\omega)$ is

$$\text{pref}(x^\omega) = \{\varepsilon, a, ab, abb, abbb, abbba, abbbab, \cdots\}$$

which can be represented by the regular expression

$$(abbb)^\star(\varepsilon + a + ab + abb)$$

Thus $\text{pref}(x^\omega)$ is regular. $\diamond$

Statement 2 *Let S be some infinite set of integers. If, for any word x, there are words y and z such that $(x^\omega)_S = yz^\infty$, then S is ultimately periodic.*

Proof. [Direction $\Rightarrow$] Let $S = \{n_0, n_1, \cdots\}$ where $n_0 < n_1 < n_2 < \cdots$. After some order k the sequence n_k, n_{k+1},... is periodic, that is, there is some $p > 0$ such that, for $i \geq k$, $n_i \in S$ iff $n_{i+p} \in S$. It follows that after that order, the corresponding sequence $(x^\omega)_S$ is also periodic; this part of the sequence corresponds to z^ω. $\diamond$

In order to prove statement 2 we have only to show that the previous statement holds. This is because, if $(x^\omega)_S = yz^\infty$, the following language is regular

$$\text{pref}(x^\omega)_S$$

Statement 2 implies Theorem 7. Let us sumarize statement 2 as follows

$$\forall S \in \mathcal{N}\ [(\forall x \in \Sigma^\star\ \exists y, z \in \Sigma^\star \quad (x^\omega)_S = yz^\omega) \quad \Rightarrow \quad S \text{ is ultimately periodic}]$$

Let us try to prove this statement by contradiction. So we are going to look for a more positive charaterizion of sets that *are not* ultimately periodic.

6.2 When the set is not ultimately periodic

Let us denote a set S of integers by $\{n_0, n_1, \ldots\}$ with $n_0 < n_1 < n_2 < \cdots$. Let us call two integers n and m *discordant* (relative to S) if either $n \in S$ and $m \notin S$ or $n \notin S$ and $m \in S$. Notice that, if n and m are discordant, then one of them is equal to some a_i.

Lemma 3 *If S is a set of integers that is not ultimately periodic then, for each $p \geq 1$, there are infinitely many integers n such that n and m are discordant (relative to S).*

Proof. By contradiction. Suppose there is some p such that only finitely many pairs (n, m) are discordant. Then the set S ultimately periodic with period p. $\diamond$

Using Lemma 3 we can easily define interesting sequences of discordant pairs. For instance, if S is not ultimately periodic, there is a sequence of discordant pairs

$$(n_1^1, n_1^1 + 1), (n_2^1, n_2^1 + 1), (n_1^2, n_1^2 + 2), (n_3^1, n_3^1 + 1), (n_2^2, n_2^2 + 2), (n_1^3, n_1^3 + 3), \cdots$$

where $n_1^1 + 1 < n_2^1$, $n_2^1 + 1 < n_1^2$, $n_1^2 + 2 < n_3^1$,...

7 Conclusions and further work

References

[BLC+06] J. Berstel, L.Boasson, O. Carton, B. Pettazzoni, and J.-E. Pin. Operations preserving regular languages. *Theoretical Computer Science*, 354:405–420, 2006.

[Mat94] Armando B. Matos. Periodic sets of integers. *Theoretical Computer Science*, 28(1):577–693, June 1994.

[SM76] J. I. Seiferas and R. McNaughton. Regularity-preserving relations. *Theoretical Computer Science*, 2:147–154, 1976.