An algorithm for computing minimal bidirectional linear recurrence relations

Citation: SALAGEAN, A.M., 2009. An algorithm for computing minimal bidirectional linear recurrence relations. IEEE Transactions on Information Theory, 55 (10), pp.4695-4700.

An Algorithm for Computing Minimal Bidirectional Linear Recurrence Relations

Ana Sălăgean

***Abstract*— We consider the problem of computing a linear recurrence relation (or equivalently a Linear Feedback Shift Register) of minimum order for a finite sequence over a field, with the additional requirement that not only the highest but also the lowest coefficient of the recurrence is nonzero. Such a recurrence relation can then be used to generate the sequence in both directions (increasing or decreasing order of indices), so we call it bidirectional. If the field is finite, a sequence is periodic if and only if it admits a bidirectional linear recurrence relation. For solving the above problem we propose an algorithm similar to the Berlekamp-Massey algorithm and prove its correctness. We describe the set of all solutions to this problem and show that if a sequence admits more than one linear recurrence relation then it admits a bidirectional one. We also prove some properties regarding the bidirectionality of the recurrences of the prefixes of the sequence.**

Keywords: Berlekamp-Massey algorithm, linear recurrence relation, minimal characteristic polynomial.

I. Introduction

The well-known Berlekamp-Massey algorithm ([1], [2]) computes a linear recurrence relation (or equivalently a Linear Feedback Shift Register, or a characteristic polynomial) of minimum order which generates a given finite sequence $s = s_0, s_2, \ldots s_{n-1}$.

In a linear recurrence relation, the coefficient of the highest term has to be equal to 1 (or some non-zero element). The coefficient of the lowest term is usually allowed to be arbitrary, including zero. If this coefficient happens to be non-zero, then we can actually use the linear recurrence relation in both directions: to generate the terms of the sequence in increasing order of their index, but also in decreasing order. We call such a linear recurrence "bidirectional".

In this paper we consider a similar problem to the one solved by the Berlekamp-Massey algorithm, but we restrict our search to bidirectional linear recurrence relations. Namely, given a finite sequence s our algorithm will compute a bidirectional linear recurrence for s of minimum order (note the order will be greater or equal to the minimum order of unrestricted linear recurrence relations).

The linear recurrence relation computed by the Berlekamp-Massey algorithm will in some cases be bidirectional. When it is not, our algorithm will compute a bidirectional one, by suitably combining some of the recurrences computed previously in the algorithm. Moreover, our algorithm also has the property that all intermediate minimal polynomials $C^{(i)}$ for $s_0, \ldots, s_{i-1}$ with $i = 1, \ldots, n$ are bidirectional whenever possible, i.e. whenever there exists a bidirectional minimal polynomial among the (unrestricted) minimal polynomials for $s_0, \ldots, s_{i-1}$. We prove an interesting property regarding the pattern of bidirectionality/non-bidirectionality of this sequence of intermediate minimal polynomials (see Theorem 3.4 and Corollary 3.5). We also describe the set of all bidirectional linear recurrence relations of minimal order for s and show that if s admits more than one linear recurrence relation then it admits a bidirectional one (see Corollary 3.6 and Theorem 3.7).

We dedicate the rest of this introductory section to further discussing the motivation of the problem we considered.

Recall that an infinite sequence $s^\infty = s_0, s_1, \ldots$ is called ultimately periodic if there are N, k such that $s_{i+N} = s_i$ for all $i \geq k$, and is called periodic if the above holds with $k = 0$. Given a linear recurrence of minimal order for the periodic part $s_k, s_{k+1}, \ldots$, the linear recurrence of minimal order of s^∞ is obtained by artificially increasing the order of the recurrence by introducing k extra coefficients, all equal to 0, in order to accommodate the terms in the pre-period $s_0, s_1, \ldots, s_{k-1}$, which do not "fit" the linear recurrence of the periodic part. (Equivalently, if $C(X)$ is a minimal polynomial for the periodic part of the sequence, then $X^k C(X)$ is a minimal polynomial for the whole sequence.) Hence, as far as linear recurrence relations are concerned, there is no connection between the pre-periodic part and the periodic part of s^∞. We could indeed describe the sequence s^∞ as two separate, unrelated entities: a finite sequence (the pre-periodic part) and an infinite periodic sequence. We can then concentrate on the linear recurrence relations of the periodic sequence.

If an infinite sequence is periodic, its minimal linear recurrence relation will be bidirectional. For finite fields the converse is also true, i.e. an infinite sequence which has a bidirectional linear recurrence relation will always be periodic (Recall that for any polynomial f over a finite field, such that $X \nmid f$, there is an integer N, called the order of f, such that $f | X^N - 1$, see for example [3]). Moreover, for infinite sequences over finite fields the following three notions are equivalent: recurrent, linear recurrent, ultimately periodic.

One of the important applications of the Berlekamp-Massey algorithm is in the cryptanalysis of stream ciphers. The keystream in a stream cipher is usually an infinite sequence s^∞ defined by some recurrence relation over a finite field. s^∞ is therefore a linear recurrent sequence and also ultimately periodic. If an attacker obtained access to a finite sequence s consisting of a number of successive terms of s^∞ (say via a known plaintext attack) they can apply the Berlekamp-Massey algorithm to determine a linear recurrence relation for s, and hope that this linear recurrence relation generates s^∞ and so the cipher would be broken. If s contains some elements of the pre-periodic part of s^∞, these elements will not help an attack based on linear recurrence relations. Therefore it makes sense to assume that s is as useful to the attacker as possible i.e. it only contains elements of the periodic part s^p of s^∞. If moreover it contains a number of elements equal to twice the linear complexity of s^p, then the linear recurrence obtained by Berlekamp-Massey algorithm for s is guaranteed to generate s^p. When fewer terms are known, this might not be the case, so in order to improve the chances of determining the correct linear recurrence for s^p we should restrict our search to bidirectional linear recurrences.

The author is with the Department of Computer Science, Loughborough University, UK (e-mail A.M.Salagean@lboro.ac.uk). A preliminary version of this work was presented at the IEEE International Symposium on Information Theory 2008, Toronto, Canada.

As bidirectional linear recurrences can generate the sequence in either direction, the relationship between a sequence s and its reverse sequence $\tilde{s}$ is of interest. If s is periodic, the linear complexities of s and $\tilde{s}$ are the same, and their minimal polynomials are reciprocals of each other. For finite sequences, this would be the case if we restricted to bidirectional linear recurrences, but otherwise the linear complexities of the two sequences are no longer necessarily equal (for example the sequence 0001 has linear complexity 4 and any polynomial of degree 4 is a minimal polynomial, whereas the reverse sequence 1000 has linear complexity 1 and minimal polynomial X). At first sight, for cryptanalysis we should therefore use whichever of the sequences has lower complexity, as that would make the attack easier (like in the case of p-adic complexity, see [4]). However, a closer look reveals that the lower complexity always stems from a non-bidirectional minimal polynomial (see Theorem 2.6) which, as we saw, would not be useful if our finite sequence is part of an infinite periodic sequence.

II. Background

Definition 2.1: Given an infinite sequence $s = s_0, s_1, \ldots$ (or a finite sequence $s = s_0, s_1, \ldots, s_{n-1}$) with elements in a field K, we say that s is a linear recurrent sequence if it satisfies a relation of the form

$$c_L s_j + c_{L-1} s_{j-1} + \ldots + c_1 s_{j-L+1} + c_0 s_{j-L} = 0$$

for all $j = L, L+1, \ldots$ (or for all $j = L, L+1, \ldots n-1$, respectively), where $c_0, c_1, \ldots, c_{L-1}, c_L \in K$ are constants and $c_L \neq 0$. The equation above is an *homogeneous linear recurrence relation of order* L and is associated with the *characteristic polynomial* $C(X) = c_L X^L + c_{L-1} X^{L-1} + \ldots + c_1 X + c_0$. If L is minimal for the given sequence, we call L the *linear complexity* of s, denoted $L(s)$, the recurrence relation is called a *minimal recurrence relation* and the characteristic polynomial is called a *minimal polynomial*.

We normally concentrate on monic characteristic polynomials, since any characteristic polynomial can be written as a monic characteristic polynomial multiplied by a non-zero constant. Similarly for minimal polynomials.

Note that in the literature there are two different ways of associating a polynomial to a recurrence relation, the one in the definition above, and the *feedback polynomial (or connection polynomial)* $c_0 X^L + c_1 X^{L-1} + \ldots + c_{L-1} X + c_L$, i.e. with the coefficients appearing in the reverse order. Given one of the polynomials one can easily obtain the other using reciprocals, for example if $\widetilde{C}(X)$ is the feedback polynomial and L the linear complexity, then the characteristic polynomial can be computed as $C(X) = X^L \widetilde{C}(X^{-1})$. We prefer to use characteristic polynomials as the degree of the minimal polynomial equals the linear complexity of the sequence and the characteristic polynomials of an infinite sequence form an ideal. The Berlekamp-Massey algorithm as presented in Massey's paper [2] uses feedback polynomials, so we will reformulate it as Algorithm 2.2 using characteristic polynomials.

As in [2], we describe the meaning and some of the properties of the variables used in the algorithm, to help its understanding and further development. At the beginning of the **for** loop, $C(X)$ will be a minimal polynomial for the current initial segment of the sequence, $s_0, s_1, \ldots, s_{N-1}$; d is the discrepancy, i.e. the difference between the actual value of s_N and the value that we would obtain for s_N using the linear recurrence given by C; $B(X)$ will be the last value taken by $C(X)$ of degree strictly smaller than the degree of the current $C(X)$; b and m will be the value of the discrepancy d and of the index N at that point. The degree of C satisfies the relation $\deg(C) = m + 1 - \deg(B)$ (see equation (13) in [2]), which justifies our comments in the algorithm. The formulae for updating $C(X)$ can be derived from the ones in the original algorithm as follows: denote by $C^{(N)}, B^{(N)}, m^{(N)}, v^{(N)}$ the value of C, B, m, v at the beginning of the **for** loop, before the updates (1) or (2) take place in Algorithm 2.2. $\widetilde{C}^{(N)}, \widetilde{B}^{(N)}$ are similarly defined for the original Berlekamp-Massey algorithm. For (2) we have in the original algorithm

$$\widetilde{C}^{(N+1)}(X) \leftarrow \widetilde{C}^{(N)}(X) - \frac{d}{b} X^{N-m^{(N)}} \widetilde{B}^{(N)}(X).$$

Substituting X^{-1} for X and multiplying by $X^{\deg(\widetilde{C}^{(N+1)})}$ we obtain therefore $X^{\deg(C^{(N+1)})} \widetilde{C}^{(N+1)}(X^{-1}) \leftarrow X^{\deg(C^{(N+1)})} \widetilde{C}^{(N)}(X^{-1}) - \frac{d}{b} X^{\deg(C^{(N+1)}) - (N - m^{(N)})} \widetilde{B}^{(N)}(X^{-1})$ but since in this case $\deg(C^{(N+1)}) = N + 1 - \deg(C^{(N)}) = N + 1 - (m^{(N)} + 1 - \deg(B^{(N)}))$,we obtain (2). Formula (1) can be similarly derived.

Algorithm 2.2: (Berlekamp-Massey Algorithm)
Input: $s = s_0, s_1, \ldots, s_{n-1}$ a sequence over a field K
Output: A monic minimal polynomial $C(X)$ for s
begin
$C(X) \leftarrow 1$
$B(X) \leftarrow 1$; $b \leftarrow 1$; $m \leftarrow -1$
for $N = 0$ **to** $n-1$ **do**
  $d \leftarrow \sum_{i=0}^{\deg(C)} c_i s_{i+N-\deg(C)}$
  if $d \neq 0$ **then**
    $v \leftarrow N - m - (\deg(C) - \deg(B))$
    if $2\deg(C) > N$ **then** (Comment: in this case $v \leq 0$)
(1) $C(X) \leftarrow C(X) - \frac{d}{b} X^{-v} B(X)$
    else (Comment: in this case $v > 0$)
      $T(X) \leftarrow C(X)$
(2) $C(X) \leftarrow X^v C(X) - \frac{d}{b} B(X)$
      $B(X) \leftarrow T(X)$
      $b \leftarrow d$; $m \leftarrow N$
    endif
  endif
endfor
return$(C(X))$
end

Example 2.3: We give an example of running the Berlekamp-Massey algorithm on the binary sequence $s = 0110010101101$. Table I records the values of the variables $N, m, B(X), C(X), d$ at the beginning of each run of the **for** loop. Being in the binary case, $b = 1$ throughout. The minimal polynomial for s computed by the Berlekamp-Massey

TABLE I

RUNNING THE BM ALGORITHM ON THE SEQUENCE 0110010101101

N	m	B	C	d
0	-1	1	1	
1	-1	1	1	0
2	1	1	X^2+1	1
3	1	1	X^2+X+1	1
4	1	1	X^2+X+1	0
5	4	X^2+X+1	X^3+X^2+X+1	1
6	4	X^2+X+1	X^3+X^2+X+1	0
7	6	X^3+X^2+X+1	X^4+X^3+1	1
8	6	X^3+X^2+X+1	X^4+X^2+X	1
9	8	X^4+X^2+X	X^5+X+1	1
10	8	X^4+X^2+X	X^5+X+1	0
11	8	X^4+X^2+X	X^5+X+1	0
12	11	X^5+X+1	$X^7+X^4+X^3+X$	1
	11	X^5+X+1	$X^7+X^4+X^3+X$	0

algorithm is $X^7+X^4+X^3+X$.

While Algorithm 2.2 computes one minimal polynomial for the given sequence, one can in fact compute all minimal polynomials using the following result from [2].

Theorem 2.4: [2, Thorem 3] At the end of Algorithm 2.2, if $2L(s) \leq n$, then $C(X)$ is the unique monic minimal polynomial of s. If $2L(s) > n$ then the set of all monic minimal polynomials of the sequence is given by

$$\{C(X)+Q(X)B(X) | \deg(Q) \leq \deg(C)-\deg(B)-(n-m)\}$$

Next we will define bidirectional linear recurrence relations.

Definition 2.5: (Bidirectional linear recurrence relation) A linear recurrence relation as defined in Definition 2.1 is called *bidirectional* if $c_0 \neq 0$. A characteristic polynomial associated to a bidirectional linear recurrence relation will be called bidirectional characteristic polynomial. A bidirectional characteristic polynomial which has minimal degree among all bidirectional characteristic polynomials of the sequence s will be called a *minimal bidirectional characteristic polynomial* (note that it may not be a minimal polynomial of s).

Hence a characteristic polynomial is called bidirectional if it is not divisible by X or equivalently, if its constant term is non-zero. If an infinite sequence admits a bidirectional characteristic polynomial (in particular, if the sequence is periodic), then the minimal polynomial of the sequence, being a factor of any characteristic polynomial, will also be bidirectional.

Note that if we have a bidirectional linear recurrence for a finite or infinite sequence s, then we can recover the whole sequence if we are given any $L = L(s)$ successive terms of the sequence, $s_i, s_{i+1}, \ldots, s_{i+L-1}$. We can compute both the next terms using the formula $s_j = -\frac{1}{c_L}(c_{L-1}s_{j-1} + \ldots + c_1 s_{j-L+1} + c_0 s_{j-L})$ for $j \geq i+L$ as well as the previous terms, using $s_j = -\frac{1}{c_0}(c_1 s_{j+1} + \ldots + c_L s_{j+L})$ for $0 \leq j < i$. If the recurrence is not bidirectional and k is the smallest index for which $c_k \neq 0$, then the terms $s_0, s_1, \ldots s_{k-1}$ cannot be recovered given arbitrary L successive terms of the sequence.

In other words, if s admits a bidirectional minimal polynomial f, then its reverse sequence $\tilde{s}$ admits the reciprocal of f as characteristic polynomial and $L(s) \geq L(\tilde{s})$. When both s and $\tilde{s}$ admit bidirectional minimal polynomials (as is the case when s is periodic), $L(s) = L(\tilde{s})$. For finite sequences this is not always the case (see example at the end of Section I) and the situation is characterised below:

Theorem 2.6: Given a finite sequence $s = s_0, s_1 \ldots, s_{n-1}$ and its reverse sequence $\tilde{s} = s_{n-1}, \ldots, s_1, s_0$ at least one of s and $\tilde{s}$ will admit a bidirectional minimal polynomial. We have $L(s) > L(\tilde{s})$ iff $\tilde{s}$ does not admit a bidirectional minimal polynomial.

Proof: Let $X^a f$ and $X^b g$ be minimal polynomials for s and $\tilde{s}$, respectively, with $X \nmid f$ and $X \nmid g$. Assume for a contradiction that neither s nor $\tilde{s}$ admit a bidirectional minimal polynomial, so $a \geq 1$ and $b \geq 1$. Assume $\deg(X^a f) \geq \deg(X^b g)$ (if not, reverse the roles of s and $\tilde{s}$). One can then verify that $X^a f + X^{\deg(g)} g(1/X)$ is a bidirectional minimal polynomial for s. ∎

III. ALGORITHM

Our aim is to compute a monic minimal bidirectional characteristic polynomial for a given finite sequence s. We could start by applying the Berlekamp-Massey algorithm, and the minimal polynomial thus obtained will sometimes happen to be bidirectional. When it is not, if the minimal polynomial is not unique then we can check whether the set given in Theorem 2.4 contains an alternative bidirectional one. We will prove that this is always the case. When the monic minimal polynomial obtained by the Berlekamp-Massey algorithm is unique but not bidirectional, then the minimal bidirectional characteristic polynomials must have a higher degree than the minimal polynomial and we show how they can be computed from the current minimal polynomial and the previous one. Rather than adjusting only the final output from the Berlekamp-Massey algorithm, our algorithm will, additionally, make sure that the intermediate values of the minimal polynomial $C(X)$ are bidirectional whenever possible. This is achieved by modifying the update formula (2) of Algorithm 2.2, justified by the following more general result.

Theorem 3.1: Algorithm 2.2 (Berlekamp-Massey algorithm) remains correct when the update formulae (1) and (2) are replaced by

$$(1') \quad C(X) \leftarrow C(X) - \frac{d}{b} X^{-v} B(X) + Q(X)B(X)$$

$$(2') \quad C(X) \leftarrow X^v C(X) - \frac{d}{b} B(X) + R(X)C(X)$$

respectively, where $Q(X), R(X)$ are arbitrary with $\deg(Q) < -v$ and $\deg(R) < v$ respectively. When $v = 0$, then $2\deg(C) = N+1$ and $Q(X) = 0$, so (1) and (1') coincide and produce the only monic minimal polynomial for s.

Proof: The correctness proof of the Berlekamp-Massey algorithm (see [2]) only depends on $C^{(N+1)}$ being a monic minimal polynomial for $s_0, \ldots, s_N$ at each step of the algorithm, and not on its particular value where several minimal polynomials exist. We know $\deg(C^{(N)}) = m^{(N)} + 1 - \deg(B^{(N)})$. When $v^{(N)} = 0$ we deduce $\deg(C^{(N)}) = N - m^{(N)} + \deg(B^{(N)})$. Adding the two formulae for $\deg(C^{(N)})$, we obtain $2\deg(C^{(N)}) = 2\deg(C^{(N+1)}) = N+1$. By Theorem 2.4, $C^{(N+1)}(X)$ is in this case the only monic minimal polynomial for $s_0, \ldots, s_N$. In any other cases, $C^{(N+1)}(X)$ is not unique; the set of all minimal polynomials is given by Theorem 2.4 as $C^{(N+1)}(X) + Q(X)B^{(N+1)}(X)$ with $\deg(Q) \leq \deg(C^{(N+1)}) - \deg(B^{(N+1)}) - (N+1-m^{(N+1)}) = -v^{(N+1)}$.

Algorithm 3.2: (Minimal bidirectional characteristic polynomial computation)
Input: $s = s_0, s_1, \ldots, s_{n-1}$ a finite sequence over a field K
Output: $C(X)$ and $D(X)$, where
$C(X)$ is a monic minimal polynomial for s, bidirectional if possible
$D(X)$ is a monic minimal bidirectional characteristic polynomial for s

begin
$C(X) \leftarrow 1$;
$B(X) \leftarrow 1$; $b \leftarrow 1$; $m \leftarrow -1$
for $N = 0$ **to** $n-1$ **do**
  $d \leftarrow \sum_{i=0}^{\deg(C)} c_i s_{i+N-\deg(C)}$
  if $d \neq 0$ **then**
    $v \leftarrow N - m - (\deg(C) - \deg(B))$
    if $2\deg(C) > N$ **then** (Comment: in this case $v \leq 0$)
(1) $C(X) \leftarrow C(X) - \frac{d}{b} X^{-v} B(X)$
    else (Comment: in this case $v > 0$)
      $T(X) \leftarrow C(X)$
      if $b_0 = 0$ and $c_0 \neq 0$ **then**
(2a) $C(X) \leftarrow X^v C(X) - \frac{d}{b} B(X) + C(X)$
      else
(2b) $C(X) \leftarrow X^v C(X) - \frac{d}{b} B(X)$
      endif
      $B(X) \leftarrow T(X)$
      $b \leftarrow d$; $m \leftarrow N$
    endif
  endif
endfor
if $c_0 \neq 0$ **then** $D(X) \leftarrow C(X)$
else
  $D(X) \leftarrow X^{n-m-(\deg(C)-\deg(B))} C(X) + B(X)$
end
return$(C(X), D(X))$
end

When formula (1) was applied we have $v^{(N+1)} = v^{(N)} + 1$, so (1') follows easily. When (2) was applied, $\deg(C^{(N+1)}) = N + 1 - \deg(C^{(N)})$, $B^{(N+1)} = C^{(N)}$ and $m^{(N+1)} = N$, so the polynomials above become $C^{(N+1)}(X) + Q(X)C^{(N)}(X)$ and $-v^{(N+1)} = v^{(N)} - 1$ so (2') follows. ∎

The algorithm we propose is given as Algorithm 3.2. The variables have the same meaning as in Algorithm 2.2.

Example 3.3: Table II shows the values of the variables $N, m, B(X), C(X), d$ at the beginning of each run of the for loop during Algorithm 3.2 for the same sequence as in Example 2.3. Comparing this with the run of the Berlekamp-Massey Algorithm described in Table I, we see that the algorithms work identically until $N = 11$ is reached, at which point the case $b_0 = 0$ and $c_0 \neq 0$ occurs, causing Algorithm 3.2 to apply a different update formula for $C(X)$, producing a bidirectional polynomial, $X^7+X^5+X^4+X^3+1$, unlike the Berlekamp-Massey algorithm.

Note that all the intermediate values of $C(X)$ in this example are bidirectional, except for the one at $N = 8$, i.e. for the sequence $s' = 01100101$ consisting of the first 8 terms of s. $X^4 + X^2 + X$ is the unique monic minimal polynomial of s', so it is impossible to find a bidirectional one of same degree. If the algorithm had s' as input, then at the end of the algorithm, since $C(X)$ is not bidirectional, $D(X) = XC(X)+B(X) = X^5+X+1$, would be computed as a minimal bidirectional characteristic polynomial for s'.

TABLE II
RUNNING ALGORITHM 3.2 ON THE SEQUENCE 0110010101101

N	m	B	C	d
0	-1	1	1	
1	-1	1	1	0
2	1	1	X^2+1	1
3	1	1	X^2+X+1	1
4	1	1	X^2+X+1	0
5	4	X^2+X+1	X^3+X^2+X+1	1
6	4	X^2+X+1	X^3+X^2+X+1	0
7	6	X^3+X^2+X+1	X^4+X^3+1	1
8	6	X^3+X^2+X+1	X^4+X^2+X	1
9	8	X^4+X^2+X	X^5+X+1	1
10	8	X^4+X^2+X	X^5+X+1	0
11	8	X^4+X^2+X	X^5+X+1	0
12	11	X^5+X+1	$X^7+X^5+X^4+X^3+1$	1
	11	X^5+X+1	$X^7+X^5+X^4+X^3+1$	0

It is clear that the modified algorithm terminates and has the same computational complexity as the Berlekamp-Massey algorithm, namely $\mathcal{O}(n^2)$ (more precisely, $\mathcal{O}(nL(s))$). As before, we denote by $C^{(N)}, B^{(N)}$ etc. the values of the variables at the beginning of the **for** loop in Algorithm 3.2 (before the updates (1), (2a) or (2b)). For the correctness of the algorithm, it can be easily verified using Theorem 3.1 that $C^{(N)}(X)$ is a minimal polynomial for $s_0, \ldots, s_{N-1}$, for $N = 1, \ldots n$. To show that these polynomials are also bidirectional whenever possible, we will need the following technical result, which is also of interest in its own right.

Theorem 3.4: Let s be a finite sequence so that the value of $C(X)$ at some point in Algorithm 3.2 is non-bidirectional. Let i be such that $C^{(i)}(X)$ is non-bidirectional and $C^{(i-1)}(X) \neq C^{(i)}(X)$. Let k be maximal such that $\deg(C^{(k)}) < \deg(C^{(i)})$ and let t be minimal such that $C^{(i)}(X) = C^{(i+1)}(X) = \ldots = C^{(t-1)}(X) \neq C^{(t)}(X)$. Then the following are true:
(i) $C^{(i)}(X)$ is the unique minimal polynomial for $s_0, s_1, \ldots, s_{i-1}$
(ii) i is even, $\deg(C^{(i)}) = \deg(C^{(i-1)}) = i/2$ and $\deg(C^{(t)}) > \deg(C^{(i)})$
(iii) $C^{(k)}(X), C^{(k+1)}(X), \ldots, C^{(i-1)}(X)$, as well as $C^{(t)}(X)$ are all bidirectional.
(iv) All the $C^{(j)}(X)$ with $\deg(C^{(j)}) = \deg(C^{(k)})$, or $\deg(C^{(j)}) = \deg(C^{(t)})$ are also bidirectional.

Proof: Let us examine how $B^{(N+1)}(X)$ and $C^{(N+1)}(X)$ become bidirectional or non-bidirectional, depending on whether $B^{(N)}(X)$ and $C^{(N)}(X)$ are bidirectional or not. We will denote by 11, 10, 01 and 00 the four possible "states" of the algorithm, where the first binary digit denotes whether $B^{(N)}(X)$ is bidirectional (denoted by 1) or not (denoted by 0) and the second binary digit denotes whether $C^{(N)}(X)$ is bidirectional or not (so in the binary case the label of the state coincides with $b_0^{(N)}, c_0^{(N)}$). Note that after the initialisation steps the algorithm is in State 11. Whenever the discrepancy $d = 0$, the algorithm stays in the same state it was in.

State 11. If the update formula (1) is applied, then $B^{(N+1)}(X) = B^{(N)}(X)$ and we have two subcases for $C^{(N+1)}$: if the additional conditions $v^{(N)} = 0$ and $c_0^{(N)} = \frac{d}{b} b_0^{(N)}$ are satisfied, then $C^{(N+1)}(X)$ is non-bidirectional so we move to state 10, otherwise $C^{(N+1)}(X)$ is bidirectional so we stay in state 11. Moreover, in the first subcase we know from Theorem 3.1 that $C^{(N+1)}(X)$ is the unique minimal polynomial of $s_0, s_1, \ldots, s_N$. Also in the first subcase, note that by the time of the next update of $C(X)$ the value of N would have increased while m stayed unchanged, so $v > 0$ and therefore formula (1) will not be applicable.

If the update formulae (2a) or (2b) have to be applied, case 11 means that (2b) will be chosen, and it can be verified that $C^{(N+1)}(X)$ is bidirectional. $B^{(N+1)}(X) = C^{(N)}(X)$ and is therefore bidirectional, so we stay in State 11.

State 10 We saw above that when we arrive in this state formula (1) will not be applicable, so $\deg(C^{(N+1)}) > \deg(C^{(N)})$. We can also see that formula (2b) will be chosen and computing the constant term in this formula we see that $C^{(N+1)}(X)$ is bidirectional. $B^{(N+1)}(X) = C^{(N)}(X)$ and is therefore non-bidirectional, so we move to State 01.

State 01 If update formula (1) is applied, then $C^{(N+1)}(X)$ stays bidirectional and $B(X)$ stays unchanged, so we stay in State 01. If the update formulae (2a) or (2b) have to be applied, the fact that we are in case 01 means that (2a) will be chosen and $C^{(N+1)}(X)$ stays bidirectional. $B^{(N+1)}(X) = C^{(N)}(X)$ and is therefore bidirectional, so we move to State 11.

State 00 From the cases above we saw that the algorithm never reaches State 00, so we need not consider it.

Following the progress of the algorithm through the different states, statements (i)-(iii) of the theorem follow. For (iv), if we assume there is a non-bidirectional $C^{(j)}(X)$, then applying (i)-(iii) for j instead of i we obtain a contradiction. ■

The Theorem above can be generalised by not requiring the intermediate minimal polynomials to be computed by a particular algorithm:

Corollary 3.5: Theorem 3.4 holds whenever each $C^{(N)}(X)$ for $N = 1, \ldots, n$ is a minimal polynomial for the sequence $s_0, s_1, \ldots, s_{N-1}$ chosen so that it is bidirectional if a bidirectional minimal polynomial exists.

A graphical illustration of the result of Theorem 3.4 and Corollary 3.5 can be given as follows. Recall that the *linear complexity profile* of a finite sequence $s_0, s_1, \ldots, s_{n-1}$ is the set $\{(i, L_i) | i = 0, \ldots, n\}$ where L_i denotes the linear complexity of the prefix $s_0, s_1, \ldots, s_{i-1}$. If we represent the points in this set as a graph we obtain a "staircase" shape. Let us label these points, for short, as "bidirectional" and "non-bidirectional" according to whether the corresponding prefix does or does not admit at least one bidirectional polynomial among its minimal polynomials. The results above tell us that "non-bidirectional" points (if they exist at all) will only appear starting at the intersection of a step with the line $2L = i$ and continuing to the end of that step. Moreover if "non-bidirectional" points appear on a particular step, then we are guaranteed that they do not appear on the step above it and the step below below it. See Figure 1 for an example.

A consequence of Corollary 3.5 relates to Theorem 2.4:

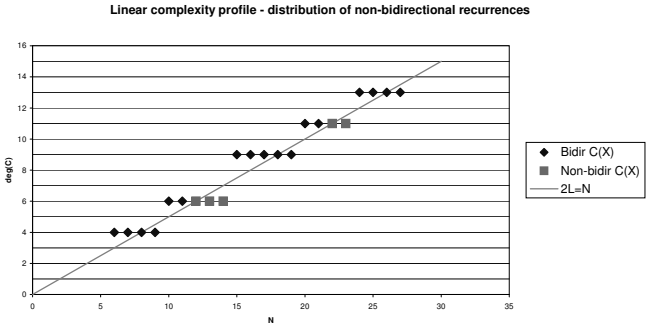

Fig. 1. Linear complexity profile

Corollary 3.6: If a finite sequence admits more than one minimal polynomial, then it admits at least one bidirectional minimal polynomial. In other words, the set of minimal polynomials described in Theorem 2.4 always contains at least one bidirectional polynomial.

We can in fact compute all minimal bidirectional characteristic polynomials, obtaining an analogue of Theorem 2.4:

Theorem 3.7: With the notations of Algorithm 3.2, the set of all monic minimal bidirectional characteristic polynomials for a finite sequence s is given by:

i. $\{C(X)\}$ if $2\deg(C) \leq n$ and $c_0 \neq 0$.

ii. $\{C(X) + Q(X)B(X) | \deg(Q) \leq \deg(C) - \deg(B) - (n - m)$, and if $b_0 \neq 0$ then $q_0 \neq -\frac{c_0}{b_0}\}$ if $2\deg(C) > n$ and $c_0 \neq 0$

iii. $\{Q(X)C(X) + uB(X) | Q(X)$ monic, $\deg(Q) = n - m - (\deg(C) - \deg(B)), u \in K^*\}$ if $2\deg(C) \leq n$ and $c_0 = 0$.

(Note that the case $2\deg(C) > n$ and $c_0 = 0$ never happens.)

Proof: The first two cases follow immediately from Theorem 2.4. We examine the third case. By Theorem 3.4, if $C(X)$ is non-bidirectional, then it is the unique minimal polynomial of s (and hence $2\deg(C) \leq n$). In general, if $C(X)$ is the unique minimal polynomial of a sequence s, one can verify that that the set of all monic characteristic polynomials of s of degree equal to $\deg(C) + i$ is $\{Q(X)C(X) | Q(X)$ monic, $\deg(Q) = i\}$, for $i = 0, 1, 2, \ldots, n - m - (\deg(C) - \deg(B)) - 1$. If $c_0 = 0$ then none of these polynomials is bidirectional. The lowest degree for which we obtain any other characteristic polynomials beside multiples of $C(X)$ is $m - n + \deg(B)$, and these polynomials are exactly the ones given in (iii) and they are bidirectional as $b_0 \neq 0$ by Theorem 3.4. ■

As a consequence of Theorems 3.1, 3.4 and 3.7 we have:

Theorem 3.8: Algorithm 3.2 is correct.

Acknowledgement The author would like to thank both reviewers for their helpful suggestions.

REFERENCES

[1] E. Berlekamp, *Algebraic Coding Theory*. McGraw Hill, 1968.

[2] J. Massey, "Shift register synthesis and BCH decoding," *IEEE Trans on Information Theory*, vol. 15, pp. 122–127, 1969.

[3] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications*. Cambridge university Press, 1994.

[4] H. Hu and D. Feng, "On the 2-adic complexity and the k-error 2-adic complexity of periodic binary sequences," *IEEE Trans. Information Theory.*, vol. 54, no. 2, pp. 874–883, 2008.

Ana Sălăgean is a Lecturer in Computer Science at Loughborough University. She graduated from University of Bucharest, Romania and obtained her PhD at RISC, J. Kepler University, Linz, Austria. She held positions at University of Bucharest, University of Bristol and Nottingham Trent University. Her main research interests are sequences and codes over rings.