On the complexity of finding shortest linear recurrences.

G. H. Norton, Dept. Mathematics, University of Queensland, Brisbane 4072.

April 26, 2002

Abstract

The Berlekamp-Massey Algorithm finds shortest linear-feedback shift register which generates a given sequence over a finite field. We show that it requires at most $2\lfloor n^2/4 \rfloor - n + 1$ multiplications for a sequence of length n. This improves a result of Gustavson by $\lfloor (3n-1)/2 \rfloor$.

A shortest linear recurrence can be obtained using Algorithm MR, due to the author. This algorithm is algebraic and independent of any particular application. We derive a monic version of it and show that at most $2\lfloor n^2/4 \rfloor - n$ multiplications are required. However since the monic Algorithm MR has a simpler control structure and admits parallelism, it is a practical improvement on the Berlekamp-Massey Algorithm.

Both analyses follow directly from a tight upper bound on the sum of the linear complexities of a sequence of length n: $\sum_{i=1}^{n-1} L_i \leq \lfloor n^2/4 \rfloor$, which may be of independent interest.

MR classification: 11B37, 12-04, 12Y05.

Keywords: linear recurrence, linear complexity, Berlekamp-Massey Algorithm, minimal realization.

1 Introduction

1.1 The Berlekamp-Massey Algorithm

Applications of finding a shortest linear recurrence or linear-feedback shift register (LFSR) via the Berlekamp-Massey (or LFSR synthesis) Algorithm [5, p. 124] are manifold in Coding Theory, Cryptography, Mathematical Systems Theory and Algebraic Computation. For a survey, see [9, Introduction]. We begin by discussing the worst-case complexity of this algorithm.

Recall that for $n \geq 1$, the linear complexity L_n of a finite sequence $(S_0, \ldots, S_{n-1})$ over a field is the least non-negative integer such that $(S_0, \ldots, S_{n-1})$ are the first n terms of an order L_n linear recurrence (or the first n outputs of an LFSR of shortest length L_n, [5]); by convention, $L_0 = 0$. Our analysis is based on an algorithm-free lemma that $\sum_{i=1}^{n-1} L_i \leq B_n \stackrel{\text{defn}}{=} \lfloor n^2/4 \rfloor$. Here the sequence is over any domain. Our inequality is tight and may be of independent interest. We note that in [2, Proof of Lemma 2.1], it was claimed that $\sum_{i=1}^{2t-1} L_i \leq t^2$.

We deduce that the Berlekamp-Massey algorithm requires at most $2B_n - n + 1$ multiplications for a sequence of length n. Our result improves the bound $n(n+1)/2$ of [4] (which was claimed to be tight) by $\lfloor (3n-1)/2 \rfloor$. Some additional remarks concerning [4] appear in the Appendix. We include an elementary inductive proof of the distribution of fixed-length sequences of prescribed linear complexity over a finite field, proved in [4] using the Berlekamp-Massey Algorithm.

1.2 Algorithm MR

Next we discuss the worst-case complexity of finding a shortest linear recurrence via Algorithm MR of [7, 9], where we developed a theory of shortest linear recurrences for finite sequences over a commutative integral domain R from first principles. (We viewed linear recurring sequences as torsion elements in the standard $R[X]$-module of Laurent series in X^{-1}, $R((X^{-1}))$; a shortest linear recurrence of the sequence then corresponds to a generator of its annihilator ideal in $R[X]$. The standard $R[X]$-module of Laurent *polynomials* $R[X^{-1}, X]$ then formed the basis of our approach to *finite* sequences. Thus our algebraic approach is independent of LFRS's and any particular application.)

The linear complexity L_n is now the degree of any non-zero 'minimal polynomial' $\mu \in R[X]$ of the given finite sequence (and we no longer require the convention $L_0 = 0$). For a precise statement of the equivalence of shortest LFRS's and minimal polynomials, see [9, Proposition 2.1, Corollary 2.3]. Algorithm MR actually computes a 'minimal realization' $(\mu, \beta) \in R[X] \times XR[X]$ of the sequence; the original sequence coincides with the initial coefficients of $\beta/\mu \in R[[X^{-1}]]$ when μ is monic.

We summarise the theory underlying Algorithm MR in Section 3.2; a simple counting argument ([7, Proposition 3.23]) showed that it is $\mathcal{O}(n^2)$. However, using our linear complexity bound, we deduce that Algorithm MR computes a minimal polynomial in at most $3B_n + 2n - 7$ R-multiplications, Proposition 3.5.

When $R = \mathbb{F}$ is a field, we derive a version of Algorithm MR which computes a *monic* minimal polynomial at each iteration, in such a way that the multiplications always involve the polynomials of smaller degree. This monic algorithm requires at most $2B_n - n$ multiplications.

We also show that computing β requires at most $B_n - n + 1$ more $\mathbb{F}$-multiplications. We remark that β could also be obtained by simply multiplying μ and the 'generating function' of the sequence in $\mathbb{F}[X^{-1}, X]$, for at most $n\deg(\mu) \leq n^2 \sim 2B_n$ multiplications. However, from a circuitry viewpoint, Algorithm MR computes β using the same control logic as for μ, differing only in the initialization. Thus μ and β could be computed in parallel, and so using Algorithm MR to compute β (without additional $\mathbb{F}[X^{-1}, X]$-multiplication circuitry) seems preferable.

1.3 Conclusions

Compared to the Berlekamp-Massey Algorithm, Algorithm MR has a simpler derivation, it is structured, valid over any integral domain, admits parallelism and generalizes readily to finite chain rings, [8] and to multiple sequences, [10].

We have shown that the monic Algorithm MR and the Berlekamp-Massey Algorithm have virtually identical worst-case complexity. Algorithm MR also has storage and logic-circuit advantages over the Berlekamp-Massey Algorithm. Thus we have made both conceptual and implementation gains by being mathematical rather than application-oriented, with no loss in run-time efficiency.

Average-case analyses of Algorithm MR for sequences of integers (perhaps adapting the techniques of [3]) and of the monic Algorithm MR for sequences over a finite field would be interesting.

2 The linear complexity bound B_n

Linear complexity is usually computed algorithmically, but to emphasize that our bound is independent of any particular algorithm, we use the following definition suggested by [5, Theorem 2]:

DEFINITION 2.1 *Let D be a domain and $d_1, \ldots, d_{n-1} \in D$. Put $L_{-1} = L_0 = 0$. For $0 \leq i \leq n-2$ we inductively define $L_{i+1} \in \{0, 1, 2, \ldots\}$ by*

$$L_{i+1} = \begin{cases} L_i & \text{if } d_{i+1} = 0 \\ \max\{L_i, i + 1 - L_i\} & \text{if } d_{i+1} \neq 0. \end{cases}$$

Now $\sum_{i=1}^{n-1} L_i \leq \lfloor n^2/4 \rfloor$ is trivially satisfied if $L_i \leq \lfloor (i+1)/2 \rfloor$ for $0 \leq i \leq n-1$ since it is easy to check that $\sum_{i=1}^{n-1} \lfloor (i+1)/2 \rfloor = \lfloor n^2/4 \rfloor$. However, $L_i \leq \lfloor (i+1)/2 \rfloor$ does not hold in general: consider a sequence of length $n \geq 2$ which is zero except for its last term for example.

On the other hand, if $(S_0, \ldots, S_{n-1})$ is such that $d_1, \ldots, d_{n-1}$ are non-zero, then $L_i = \lfloor (i+1)/2 \rfloor$ for $0 \leq i \leq n-1$, so $\sum_{i=1}^{n-1} L_i = \lfloor n^2/4 \rfloor$ in this case.

LEMMA 2.2 *We have $\sum_{i=1}^{n-1} L_i \leq B_n$, where $B_n \stackrel{\text{defn}}{=} \lfloor n^2/4 \rfloor$.*

PROOF. It is enough to show that

$$\sum_{j=1}^{n-1} L_j \leq \sum_{j=1}^{n-1} \lfloor (j+1)/2 \rfloor. \tag{1}$$

It is convenient to call j *stable* if it is odd, $L_j = \lfloor (j+1)/2 \rfloor$ and $\sum_{i=1}^{j} L_i \leq \sum_{i=1}^{j} \lfloor (i+1)/2 \rfloor$. Clearly -1 is stable, so suppose inductively that $s \stackrel{\text{defn}}{=} 2u - 1 \geq -1$ is stable, so that $L_s = u$. Then $L_{s+1} = u$ independently of d_{s+1}. If $d_{s+2} \neq 0$ then $L_{s+2} = u + 1 = \lfloor (s+2+1)/2 \rfloor$ and

we can replace s by $s+2$. Hence we can assume that $d_{s+2} = 0$, so that $L_{s+2} = u$ and for some $t \geq 2$, $L_{s+1} = \cdots = L_{s+t} = u$. If $n-1 = s+t$, the result clearly holds. Otherwise, we can assume that $L_{s+t+1} \neq u$. Then $d_{s+t+1} \neq 0$ and $L_{s+t+1} = u+t$. Assume for the moment that $n-1 \geq s+2t$. Since $u+t = \max\{u+t, s+2t-(u+t)\}$, we have $L_{s+t+1} = \cdots = L_{s+2t} = u+t$ and $L_{s+2t} = \lfloor (s+2t+1)/2 \rfloor$. Now $\sum_{j=s+1}^{s+2t} L_j = tu + t(u+t)$ while

$$\begin{aligned} \sum_{j=s+1}^{s+2t} \lfloor (j+1)/2 \rfloor &= \sum_{k=0}^{t-1} (\lfloor (s+t-k+1)/2 \rfloor + \lfloor (s+t+1+k+1)/2 \rfloor) \\ &= \sum_{k=0}^{t-1} ((s+t-k+1)/2 + (s+t+1+k+1)/2 - 1/2) \qquad (2) \\ &= t(2u+t) \end{aligned}$$

where the equality (2) follows since $s+t \pm k$ have the same parity, so that the two numerators have opposite parity. Thus $s+2t$ is stable. By induction there is a maximal stable j. If $n-1$ is stable, we are done. Otherwise, if $n-1 = s+t+1+m$ for $0 \leq m < t-1$, then the pairs with indices $s+t-k, s+t+1+k$ for $k = 0, \ldots, m$ still satisfy (2), while each of the remaining terms $L_j = u$ for $j = s+1, \ldots, s+t-m-1$ is less or equal to the corresponding $\lfloor (j+1)/2 \rfloor$. $\bowtie$

3 Worst-case analyses

3.1 The Berlekamp-Massey Algorithm

First we rewrite this algorithm slightly:

ALGORITHM 3.1 *([5, p. 124])*

Input: $n \geq 1$, sequence $(S_0, \ldots, S_{n-1})$ over $\mathbb{F}$.
Output: L_n and connection polynomial $c = 1 + c_1 X + \cdots + c_r X^r \in \mathbb{F}[X]$ of $(S_0, \ldots, S_{n-1})$, $r \leq L_n$.
Auxiliaries: c', $t \in \mathbb{F}[X]$, $d, d' \in \mathbb{F} \setminus \{0\}$, integer e.

$L := 0$; $c := 1$; $e := 1$; $c' := 1$; $d' := 1$;
for $i := 0$ to $n-1$ do
  $d := \sum_{j=0}^{L} c_j \cdot S_{i-j}$;
  if $(d = 0)$ $e := e+1$;
  else if $(2L > i)\{$ $c := c - (d/d') \cdot X^e c'$; $e := e+1;\}$
    else $\{L := i+1-L$; $t := c$; $c := c - (d/d') \cdot X^e c'$; $c' := t$; $d' := d$; $e := 1;\}$
return L, c.

PROPOSITION 3.2 *If $(S_0, \ldots, S_{n-1})$ is a sequence of elements from a field $\mathbb{F}$, the Berlekamp-Massey Algorithm requires at most $2B_n - n + 1$ multiplications.*

PROOF. For $0 \leq i \leq n-1$, we write L_{i+1} and $c^{(i+1)}$ for the $(i+1)^{st}$ connection polynomial formed at the end of iteration i from $c^{(i)}$ and $c'^{(i)}$. Computing the discrepancy requires at most $\sum_{i=0}^{n-1} L_i \leq B_n$ multiplications. For additional multiplications to accrue, we need $\deg(c'^{(i)}) > 0$ for $i \geq j > 0$. Now $c^{(0)} = 1 = c'^{(0)}$ and $e > 0$, so $c'^{(i)}$ always has constant term 1. Also, $\deg(c^{(i)}) \leq L_i$ and $\deg(c'^{(i)}) > 0$ implies that $\deg(c'^{(i)}) < \deg(c^{(i)})$. So at most $\sum_{i=j}^{n-1}(\deg(c'^{(i)}) \leq \sum_{i=j}^{n-1}(\deg(c^{(i)}) - 1)$ additional multiplications can occur, for some j. Examination of the initial iterations shows that the minimal value of j is 3, with $\deg(c^{(1)}) = \deg(c^{(2)}) = 1$ and $\deg(c^{(3)}) = 2$ (and $c'^{(3)} = 1 - S_0X$). This gives at most an additional

$$\sum_{i=3}^{n-1}(\deg(c^{(i)}) - 1) = \sum_{i=0}^{n-1}\deg(c^{(i)}) - \sum_{i=0}^{2}\deg(c^{(i)}) - n + 3 \leq B_n - n + 1.$$

$\bowtie$

3.2 Algorithm MR

We recall some of the key notions which underly the main constructive result of [9], Theorem 7.3.

Let R be a commutative integral domain with $1 \neq 0$. A finite sequence over R is written $(S_0, \ldots, S_m)$ with $m \leq 0$ (so the length of the sequence is $1-m$). Then $(S_0, \ldots, S_m)$ satisfies a linear recurrence of order d if for some monic $f \in R[X]$ with $\deg(f) = d$, $S_j = -f_{d-1}S_{j+1} + \cdots + f_0 S_{j+d}$ for $m \leq j \leq -d$ (and $d \geq 1 - m$ is allowed).

With $S = S_0 + \cdots + S_m X^m \in R[X^{-1}]$, this is equivalent to $(f \cdot S)_j = 0$ for $m + \deg(f) \leq i \leq 0$ for some monic f, where $\cdot$ denotes multiplication in $R[X^{-1}, X]$. We call such an f an annihilator of $(S_0, \ldots, S_m)$, without insisting that f be monic. Clearly f extends to an annihilator of $(S_0, \ldots, S_{m-1})$ if and only if the obstruction $\mathcal{O}f \stackrel{\text{defn}}{=} (f \cdot S')_{m-1+\deg(f)} \in R$ vanishes, where $S' = S + S_{m-1}X^{m-1}$. Two annihilators of $(S_0, \ldots, S_m)$ were constructed inductively from first principles in [9, Section 4] using commutators $[\mathcal{O}f, S_0], [\mathcal{O}f, \mathcal{O}g] \in R$ for certain annihilators f, g.

For any $f \in R[X]$ the polynomial $\beta(f, S) = \sum_{i=1}^{\deg(f)}(f \cdot S)_i X^i \in XR[X]$ plays a useful role in developing the theory. Clearly f is an annihilator of $(S_0, \ldots, S_m)$ if and only if $f \cdot S = \beta(f, S)$. Further, if f is monic, the first $1-m$ coefficients of $\beta(f, S)/f \in R[[X^{-1}]]$ coincide with the original sequence (see [9, Proposition 2.12]).

A non-zero annihilator of $(S_0, \ldots, S_m)$ is minimal if its degree is minimal. We write $\mu^{(m)}$ for a typical minimal polynomial of $(S_0, \ldots, S_m)$ and κ_m for $\deg(\mu^{(m)})$; in the notation of Section 1, $\kappa_m = L_{1-m}$. For example, if $(S_0, \ldots, S_m) = (0, \ldots, 0, 1)$, we can take $\mu^{(m)} = X^{1-m}$.

The inductive proof that the second construction yields a minimal polynomial required the key notion of the antecedent α_{m+1} of κ_{m+1}, defined when $\kappa_0 < \kappa_{m+1}$ (so $m + 1 < 0$); α_{m+1} is the smallest value of i, $m + 1 < i \leq 0$ such that $\kappa_i < \kappa_{m+1}$, [9, Definition 5.2]. (We think of $\mu^{(\alpha_{m+1})}$ as a 'best previous minimal polynomial' with $\deg(\mu^{(\alpha_{m+1})}) < \deg(\mu_{m+1})$.) The minimality of

the constructions was proved in [9, Proposition 5.1, 5.3], which have the following immediate consequence

COROLLARY 3.3 *(Cf. [5, Theorem 2]) If* $\mathcal{O}\mu^{(m+1)} \neq 0$ *for some* $\mu^{(m+1)}$ *then*

$$\deg(\mu^{(m)}) = \max\{\deg(\mu^{(m+1)}), 1 - m - \deg(\mu^{(m+1)})\}.$$

Motivated by Mathematical Systems Theory, we called $(f, \beta(f, S))$ a realization of $(S_0, \ldots, S_m)$; if further $\deg(f)$ is minimal, we call $(f, \beta(f, S))$ a minimal realization (MR) of $(S_0, \ldots, S_m)$. It follows that minimal realization is closely related to rational approximation with minimal-degree denominator. We write an MR of $(S_0, \ldots, S_m)$ as $\overline{\mu}^{(m)} = (\mu^{(m)}, \beta^{(m)})$. It was shown that $\beta^{(m)}$ could also be constructed inductively in [9, Section 6].

Finally, in [9, Theorem 7.3], the two inductive MR constructions were simplified and combined by

(i) supressing α_{m+1} and using $\mu^{(\alpha_{m+1})}$ instead;

(ii) subsuming the case $m + 1 \leq 0$ and $\kappa_{m+1} = \kappa_0$ by defining $\overline{\mu}^{(\alpha_{m+1})}$ to be $(0, -X)$ if $S_0 = 0$ and (1,0) otherwise; see [9, Definition 7.2];

(iii) using the variable $d_{m+1} \stackrel{\text{defn}}{=} 2\kappa_{m+1} + m - 1$, which satisfies $d_{m+1} < 0$ if and only if $\kappa_m = 1 - m - \kappa_{m+1} > \kappa_{m+1}$ if and only if $\overline{\mu}^{(\alpha_m)}$ must be updated (by $\overline{\mu}^{(m+1)}$).

This combined construction directly suggested the body of Algorithm MR (writing current values $\overline{\mu}^{(\alpha_{m+1})}$, $\mathcal{O}\mu^{(\alpha_{m+1})}$ and d_{m+1} as $\overline{\mu}'$, $\mathcal{O}'$ and d respectively) as well as how to initialize the algorithm, [9, Section 7.2]. (We have negated $\overline{\mu}$ so that $\mu^{(m)} = X^{1-m}$ if $(S_0, \ldots, S_m) = (0, \ldots, 0, 1)$.)

ALGORITHM 3.4 *([7, 9, Algorithm MR])*

Input: $m \leq 0$, R an integral domain, $(S_0, \ldots, S_m) \in R$.
Output: $\overline{\mu}$, an MR for $(S_0, \ldots, S_m)$.
Auxiliaries: MR's $\overline{\mu}'$, $\overline{t} \in R[X] \times XR[X]$, $\mathcal{O}, \mathcal{O}' \in R \setminus \{0\}$, integer d.

$\overline{\mu} := (1, 0)$; $\overline{\mu}' := (0, -X)$; $\mathcal{O}' := 1$; $d := -1$;
for $i := 0$ downto m do
  { $\mathcal{O} := \sum_{j=0}^{\deg(\mu)} \mu_j \cdot S_{i+\deg(\mu)-j}$; /* *compute* $\mathcal{O}$ */
    if $(\mathcal{O} \neq 0)$ { if $(d < 0)$ $\{d := -d; \text{swap}(\overline{\mu}, \overline{\mu}'); \text{swap}(\mathcal{O}, \mathcal{O}');\}$ /* *update* $\overline{\mu}', \mathcal{O}'$ */
      $\overline{\mu} := \mathcal{O} \cdot X^d \overline{\mu}' - \mathcal{O}' \cdot \overline{\mu};\}$ /* *update* $\overline{\mu}$ */
    $d := d - 1;\}$ /* *update* d */
return $\overline{\mu}$.

PROPOSITION 3.5 *Let* $(S_0, \ldots, S_m)$ *be a sequence of elements from a domain* R *and* $n = 1 - m$. *The number of* R*-multiplications required to compute a minimal polynomial is at most* $3B_n + 2n - 7$ *and to compute an MR is at most* $5B_n + n - 6$.

PROOF. For $m \le i \le 0$, $\overline{\mu}^{(i)}$ is obtained using $\mathcal{O}_i = \mathcal{O}\mu^{(i+1)}$, d_{i+1}, $\overline{\mu}'^{(i+1)}$, $\mathcal{O}'_{i+1}$ and $\overline{\mu}^{(i+1)}$. Computing $\mathcal{O}_i$ requires at most $\sum_{i=0}^{m}(\kappa_{i+1}+1) = \sum_{i=0}^{n-1} L_i + n \le B_n + n$ multiplications. To maximise the number of updating multiplications, we want $\mathcal{O}_i \neq 0$ for $m \le i \le 0$. Initially, we have $\overline{\mu}^{(0)} = (X, S_0X)$, $\overline{\mu}'^{(0)} = (1,0)$ and $d_0 = 0$; then $\overline{\mu}^{(-1)} = (-S_0X + S_{-1}, -S_0^2X)$ and $d_{-1} = -1$. Next, $\mathcal{O}_{-2} = S_{-1}^2 - S_0S_{-2}$, $\overline{\mu}^{(-2)} = \mathcal{O}'_{-1}X\overline{\mu}^{(-1)} - \mathcal{O}_{-2}(1,0)$ and $\overline{\mu}'^{(-2)} = \overline{\mu}^{(-1)}$.

For computing minimal polynomials, $\overline{\mu}^{(0)}$ and $\overline{\mu}^{(-1)}$ require no multiplications and $\mu^{(-2)}$ at most 2. For $m \le i \le -3$, $1 \le \deg(\mu'^{(i+1)}) \le \deg(\mu^{(i+1)}) - 1$, giving at most $\sum_{i=j}^{m}(\deg(\mu'^{(i+1)}) + \deg(\mu^{(i+1)}) + 2)$ more multiplications. In all therefore, there are at most

$$2 + \sum_{i=-3}^{m}(2\kappa_{i+1}+1) = 2\left(\sum_{i=0}^{m}\kappa_{i+1} - \sum_{i=0}^{-2}\kappa_{i+1}\right) - m \le 2B_n + n - 7.$$

For computing an MR, $\beta^{(-1)} = S_0^2X$ and $\beta^{(-2)} = \mathcal{O}'_{-1}X\beta^{(-1)}$ each require at most one multiplication. In general, $\deg(\beta^{(i+1)}) \le \deg(\mu^{(i+1)})$ and $X|\beta^{(i+1)}$. Similarly for $\beta'^{(i+1)}$. Hence this updating requires at most another

$$2 + \sum_{i=-3}^{m}\left(\deg(\beta'^{(i+1)}) + \deg(\beta^{(i+1)})\right) \le 2 + \sum_{i=-3}^{m}(2\kappa_{i+1}-1) \le 2B_n - n + 1$$

for a total of at most $5B_n + n - 6$ multiplications. ⋈

4 A monic Algorithm MR

When $R = GF(2)$, Algorithm MR clearly computes a monic minimal polynomial, the multiplications reduce to additions and the updating becomes:

$$\text{if } \left(\sum_{j=0}^{\deg(\mu)} \mu_j \cdot S_{i+\deg(\mu)-j} \neq 0\right) \{ \text{ if } (d<0)\ \{d := -d; \text{swap}(\overline{\mu}, \overline{\mu}');\}\ \overline{\mu} := \overline{\mu} + X^d\overline{\mu}';\}.$$

We now show how to compute a monic minimal polynomial when R is a field $\mathbb{F}$ in such a way that the polynomials μ', β' (of strictly smaller degree) are multiplied by an element of $\mathbb{F}$.

THEOREM 4.1 *Let $(S_0, \ldots, S_m)$ be a sequence over a field $\mathbb{F}$. Put*

$$\begin{array}{llll} \overline{\mu}^{(0)} = (1,0), & \overline{\mu}^{(\alpha_0)} = (0,-X) \text{ and } \mathcal{O}\mu^{(\alpha_0)} = 1 & \text{if } S_0 = 0 \\ \overline{\mu}^{(0)} = (X, S_0X), & \overline{\mu}^{(\alpha_0)} = (1,0) \quad \text{and } \mathcal{O}\mu^{(\alpha_0)} = S_0 & \text{if } S_0 \neq 0. \end{array}$$

For $m \le i \le -1$ let $\overline{\mu}^{(i)} \in \mathbb{F}[X]$ be defined inductively by $\overline{\mu}^{(i)} = \overline{\mu}^{(i+1)}$ if $\mathcal{O}\mu^{(i+1)} = 0$ and

$$\overline{\mu}^{(i)} = \begin{cases} \overline{\mu}^{(i+1)} - q_i \cdot X^{d_{i+1}}\overline{\mu}^{(\alpha_{i+1})} & \text{if } d_{i+1} \ge 0 \\ X^{-d_{i+1}}\overline{\mu}^{(i+1)} - q_i \cdot \overline{\mu}^{(\alpha_{i+1})} & \text{if } d_{i+1} < 0 \end{cases}$$

otherwise, where $q_i = \mathcal{O}\mu^{(i+1)}/\mathcal{O}\mu^{(\alpha_{i+1})}$. *Then* $\overline{\mu}^{(i)}$ *is a monic MR and*

$$\overline{\mu}^{(\alpha_i)} = \begin{cases} \overline{\mu}^{(\alpha_{i+1})} & \text{if } d_{i+1} \geq 0 \\ \overline{\mu}^{(i+1)} & \text{if } d_{i+1} < 0 \end{cases} \qquad d_i = \begin{cases} d_{i+1} - 1 & \text{if } \mathcal{O}\mu^{(i+1)} = 0 \\ |d_{i+1}| - 1 & \text{otherwise.} \end{cases}$$

PROOF. It was noted in [9] that the initial values are natural. The proof that $\overline{\mu}^{(i)}$ is as claimed is by induction on i. Assume that $\mu^{(j)}$ is monic for $i+1 \leq j \leq 0$ and that $\mathcal{O}\mu^{(i+1)} \neq 0$. Since a non-zero multiple of an MR is clearly an MR, [9, Theorem 7.3](i) implies that $\overline{\mu}^{(i)}$ is an MR and we need only show that $\overline{\mu}^{(i)}$ is monic.

We consider two cases: $\kappa_{i+1} = \kappa_0$ and $\kappa_{i+1} > \kappa_0$. If $\kappa_0 = \kappa_{i+1}$, then as in the proof of *loc. cit.*, Theorem 7.3(i), $\mu^{(i)}$ is precisely the minimal polynomial of [9, Proposition 5.3] made monic, and we are done.

Suppose now that $\kappa_{i+1} > \kappa_0$, so that $\mu^{(\alpha_{i+1})} \neq 0$. Put $a = \alpha_{i+1}$. If $d_{i+1} < 0$ then $\mu^{(i)}$ is monic since $\mu^{(i+1)}$ is monic and $-d_{i+1} + \deg(\mu^{(i+1)}) > \deg(\mu^{(i+1)}) > \deg(\mu^{(a)})$. For $d_{i+1} \geq 0$, we show that $\deg(\mu^{(i+1)}) > d_{i+1} + \deg(\mu^{(a)})$, which will imply that $\mu^{(i)}$ is monic.

By construction, we have $i+1 < a \leq 0$ and $\deg(\mu^{(i+1)}) = \deg(\mu^{(a-1)}) > \deg(\mu^{(a)})$. Now $\deg(\mu^{(a-1)}) = 1 - (a-1) - \deg(\mu^{(a)})$ by Corollary 3.3. Thus $\deg(\mu^{(i+1)}) = 2 - a - \deg(\mu^{(a)})$ and $d_{i+1} = 2\deg(\mu^{(i+1)}) + i - 1 = 4 - 2a - 2\deg(\mu^{(a)}) + i - 1$. Hence if $\deg(\mu^{(i+1)}) \leq d_{i+1} + \deg(\mu^{(a)})$, then $2 - a - \deg(\mu^{(a)}) \leq 3 - 2a - \deg(\mu^{(a)}) + i$ and $a \leq i + 1$, for a contradiction. The remainder of the proof is the same as part (ii) of *loc. cit.*, Theorem 7.3. ⋈

As in *loc. cit.*, Section 7.2, Theorem 4.1 yields

ALGORITHM 4.2 *(monic Algorithm MR)* (cf. [1, p. 184], [5, p. 124].)

Input: $m \leq 0$, sequence $(S_0, \ldots, S_m)$ over $\mathbb{F}$.
Output: $\overline{\mu}$, a monic MR for $(S_0, \ldots, S_m)$.
Auxiliaries: MR's $\overline{\mu}'$, $\overline{t} \in \mathbb{F}[X] \times X\mathbb{F}[X]$, $\mathcal{O}, \mathcal{O}' \in \mathbb{F} \setminus \{0\}$, integer d.

$\overline{\mu} := (1, 0)$; $d := -1$; $\overline{\mu}' := (0, -X)$; $\mathcal{O}' := 1$;
for $i := 0$ downto m do
  $\mathcal{O} := \sum_{j=0}^{\deg(\mu)} \mu_j \cdot S_{i+\deg(\mu)-j}$;
  if $(\mathcal{O} \neq 0)$ { if $(d \geq 0)$ $\overline{\mu} := \overline{\mu} - (\mathcal{O}/\mathcal{O}') \cdot X^d \overline{\mu}'$;
    else $\{d := -d$; $\overline{t} := \overline{\mu}$; $\overline{\mu} := X^d \overline{\mu} - (\mathcal{O}/\mathcal{O}') \cdot \overline{\mu}'$; $\overline{\mu}' := \overline{t}$; $\mathcal{O}' := \mathcal{O}$;$\}\}$
  $d := d - 1$;$\}$
return $\overline{\mu}$.

We show the algorithm at work when $\mathcal{O}_i \neq 0$ for $i = 0, -1, -2$. Writing $q_i = \mathcal{O}_i/\mathcal{O}'_{i+1}$, we have $\mathcal{O}_0 = S_0$, $\overline{\mu}^{(0)} = (X, q_0 X)$, $\overline{\mu}'^{(0)} = (1, 0)$, $\mathcal{O}'_0 = S_0$ and $d_0 = 0$. Next, $\mathcal{O}_{-1} = S_{-1}$, $\overline{\mu}^{(-1)} = (X - q_{-1}, q_0 X)$, $\overline{\mu}'^{(-1)} = (1, 0)$ and $d_{-1} = -1$. On the third iteration, $\mathcal{O}_{-2} = S_{-2} - S_{-1}^2/S_0$,

$\overline{\mu}^{(-2)} = (X^2 - q_{-1}X - q_{-2}, q_0X^2)$, $\overline{\mu}'^{(-2)} = (X - q_{-1}, q_0X)$ and $d_{-2} = 0$. Note that the only multiplication is to compute $\mathcal{O}_{-2}$.

The reader may check that when $\mathbb{F} = GF(2)$, the updating of $\overline{\mu}$ and $\overline{\mu}'$ in the monic version reduces to that given at the beginning of this section.

PROPOSITION 4.3 *Let $(S_0, \ldots, S_m)$ be a sequence of elements from a field $\mathbb{F}$ and $n = 1 - m$. The number of $\mathbb{F}$-multiplications required to compute a monic minimal polynomial is at most $2B_n - n$ and to compute a monic MR is at most $3B_n - 2n$.*

PROOF. This is similar to the proof of Proposition 3.5. Computing $\mathcal{O}_i$ requires at most $\sum_{i=0}^{m} \kappa_{i+1} = \sum_{i=0}^{n-1} L_i \leq B_n$ multiplications since $\mu^{(i+1)}$ is monic. To maximise the multiplications arising during updating, we want $\mathcal{O}_i \neq 0$ for $m \leq i \leq 0$. None arise for $i = 0, -1, -2$ and for $i \leq -3$, $1 \leq \deg(\mu'^{(i+1)}) \leq \deg(\mu^{(i+1)}) - 1$, so computing $\mu^{(m)}$ requires at most $\sum_{i=-3}^{m}(\kappa_{i+1} - 1) \leq B_n - n$ multiplications. Similarly, computing $\beta^{(m)}$ requires at most $\sum_{i=-3}^{m} \deg(\mu'^{(i+1)}) \leq B_n - n$ multiplications. ⋈

Appendix: Three remarks on [4]

According to [4, p.209], the number of multiplications required by the Berlekamp-Massey algorithm for a sequence of length $n \geq 1$ over a field $\mathbb{F}$ is

$$\begin{cases} W_n = n(n+1)/2 & \text{at worst} \\ A_{q,n} = n(n+1)/2 - q^{-1}n^2/4 + \mathrm{o}(q^{-2}) & \text{on the average,} \end{cases}$$

where the average is for uniformly and independently distributed (u.i.d.) sequences and $|\mathbb{F}| = q$. Both results were proved using the Berlekamp-Massey algorithm.

1. It was claimed on [4, p. 209] that W_n is a tight upper bound, but $W_n = 2B_n - n + 1 + \lfloor (3n-1)/2 \rfloor$ and so it cannot be tight.

2. Since $A_{q,n} + \mathrm{o}(q^{-1}) = W_n = 2B_n - n + 1 + \lfloor (3n-1)/2 \rfloor$, we have $2B_n - n \geq A_{q,n} > 2B_n - n$. Thus the value of $A_{q,n}$ found in [4] is incorrect.

3. The proof of [4, Proposition 1] used the Berlekamp-Massey algorithm under the hypothesis that 'there is one formula for a sequence of length zero'. We now give an elementary inductive proof of [4, Proposition 1], independent of any particular algorithm and which avoids 'sequences of length zero'. Our inductive basis is simply that for $n = 1$, there is a unique sequence of linear complexity 0 and that there are $q - 1$ sequences of linear complexity 1. We then continue with Definition 2.1 and elementary properties of linear complexity.

Proposition Let $\mathbb{F}$ be a field with q elements and $n \geq 1$. If sequences of length n are u.i.d., then

$$\Pr[L_n = L] = \begin{cases} 0 & \text{if } L < 0 \\ q^{-n} & \text{if } L = 0 \\ q^{2L-n-1}(q-1) & \text{if } 1 \leq L \leq \lfloor n/2 \rfloor \\ q^{n-2L}(q-1) & \text{if } \lfloor n/2 \rfloor < L \leq n \\ 0 & \text{if } L > n. \end{cases}$$

PROOF. Clearly $\Pr[L_n = L] = 0$ for $L < 0$ or $L > n$. It is convenient to put $N(n, L) = |\{(S_0, \ldots, S_{n-1}) \ : L_n = L\}|$. We will show by induction on n that $N(n, L)$ coincides with the stated probabilities multiplied by q^n . Let $n = 1$. It is clear that $S_0 = 0$ is the unique sequence with $L_1 = 0$ and that there are $q - 1$ sequences S_0 of complexity 1, which agrees. Suppose inductively that the result is true for sequences of length $n - 1 \geq 1$. We consider three cases.

(a) $L = 0, n$. Let $L = 0$. Then clearly $N(n, L) \geq 1$. If $L_n(S_0, \ldots, S_{n-1}) = 0$ then $(S_0, \ldots, S_{n-2})$ is the all-zero sequence by the inductive hypothesis since $0 \leq L_{n-1} \leq L_n = L$ and so $N(n, L) = 1$. Suppose now that $L = n$. We show that $N(n, L) = q - 1$. If $(S_0, \ldots, S_{n-2})$ is the all-zero sequence and $d_n = S_{n-1} \neq 0$ then $L_n = n$, so $N(n, L) \geq q - 1$. Moreover, $L_{n-1} \leq n - 1$ and $n = L_n = \max\{L_{n-1}, n - L_{n-1}\}$ forces $L_{n-1} = 0$, so $(S_0, \ldots, S_{n-2})$ is the all-zero sequence. Thus $N(n, n) = q - 1$.

(b) $1 \leq L \leq \lfloor n/2 \rfloor$. Suppose first that $2L \leq n - 1$. Then $L_{n-1} \leq L_n = L \leq \lfloor (n-1)/2 \rfloor$ and we can apply the inductive hypothesis to any $(S_0, \ldots, S_{n-2})$. If S_{n-1} is such that $d_n = 0$ for some $(S_0, \ldots, S_{n-2})$, then $1 \leq L = L_n = L_{n-1} \leq \lfloor (n-1)/2 \rfloor$, and we obtain $N(n-1, L) = q^{2L-1}(q-1)$ sequences in this way. We also have $L < n - L$, so L cannot result from some $(S_0, \ldots, S_{n-2})$ with $d_n \neq 0$. Thus $N(n, L) = N(n-1, L) = q^{2L-1}(q-1)$ as required.

Suppose now that $2L = n$. Then $L > \lfloor (n-1)/2 \rfloor$. If S_{n-1} is such that $L_n = L$ and $d_n = 0$, the inductive hypothesis yields $N(n-1, L) = q^{2(n-1-L)}(q-1)$ sequences. There are also $(q-1)N(n-1, L) = q^{2(n-1-L)}(q-1)^2$ sequences resulting from $d_n \neq 0$. Thus

$$N(n, L) = N(n-1, L) + (q-1)N(n-1, L),$$

and substituting the inductive values and $n = 2L$ yields the result.

(c) $\lfloor n/2 \rfloor < L \leq n$. Then $(n-1)/2 < L$ and $\max\{L, n - L\} = L$. If S_{n-1} is such that $d_n = 0$, then $\lfloor (n-1)/2 \rfloor < L = L_{n-1} \leq n - 1$ and we can apply the inductive hypothesis to $(S_0, \ldots, S_{n-2})$, giving $N(n-1, L) = q^{2(n-1-L)}(q-1)$ sequences. We also get a sequence of complexity L if $d_n \neq 0$ and either (i) $L_{n-1} = L$ or (ii) $L_{n-1} = n - L$. Since $\lfloor (n-1)/2 \rfloor < L = L_{n-1} \leq n - 1$, (i) gives $(q-1)N(n-1, L) = q^{2(n-1-L)}(q-1)^2$ sequences. For (ii), we have $1 \leq n - L \leq \lfloor (n-1)/2 \rfloor$ and so we obtain an additional $(q-1)N(n-1, n-L) = q^{2(n-L)-1}(q-1)^2$ sequences. Thus

$$N(n, L) = N(n-1, L) + (q-1)N(n-1, L) + (q-1)N(n-1, n-L)$$

and on substituting the inductive values, we easily get $N(n, L) = q^{2(n-L)}(q-1)$ as required.

For an equivalent result derived from the number of sequences $(S_0, \ldots, S_{n-1})$ over $\mathbb{F}$ with prescribed linear and jump complexity, see [6].

The author would like to thank Tim Blackmore for his interest and an anonymous referee for simplifying an earlier proof of Lemma 2.2.

We take this opportunity to correct several typographical errors in [9]:

p. 335, line 7: delete $\epsilon(g) + \deg(g) \leq -m$

p. 336, lines 2,3 should read $\mathcal{O}(X^2 - X) = ((X^2 - X) \circ \mathcal{F}'_{-3+2} = \mathcal{F}'_{-3} - \mathcal{F}'_{-2} = 1$

p. 336 line 13 $n < m$ should be $m \leq n$.

References

[1] E. R. Berlekamp. *Algebraic Coding Theory.* Series in Systems Science. McGraw Hill, New York-Toronto, 1968.

[2] P. Fitzpatrick and S. Jennings. Comparison of two algorithms for decoding alternant codes. *Applicable Algebra in Engineering, Communications and Computing*, 9:211–220, 1998.

[3] P. Flajolet and B. Vallée. Continued fractions, comparison algorithms and fine structure constants, in Constructive, Experimental and Nonlinear Analysis. *Proceedings Canadian Mathematical Society*, 27:53–82, 2000.

[4] F.G. Gustavson. Analysis of the Berlekamp-Massey linear feedback shift-register synthesis algorithm. *IBM J. Res. Dev.*, 20:204–212, 1976.

[5] J. L. Massey. Shift-register synthesis and BCH decoding. *IEEE Trans. Inform. Theory*, 15:122–127, 1969.

[6] H. Niederreiter. The linear complexity profile and the jump complexity of keystream sequences. *Lecture Notes in Computer Science*, 473:174–188, 1990.

[7] G. H. Norton. On the minimal realizations of a finite sequence. *J. Symbolic Computation*, 20:93–115, 1995.

[8] G. H. Norton. On minimal realization over a finite chain ring. *Designs, Codes and Cryptography*, 16:161–178, 1999.

[9] G. H. Norton. On shortest linear recurrences. *J. Symbolic Computation*, 27:323–347, 1999.

[10] G.H. Norton. Some decoding applications of minimal realization. In *Cryptography and Coding*, volume 1025, pages 53–62. Lecture Notes in Computer Science. Springer, 1995.