On the minimal realizations of a finite sequence. *

GRAHAM NORTON
Centre for Communications Research, University of Bristol, England.

July 19, 2001

Abstract

We develop a theory of minimal realizations of a finite sequence over an integral domain R, from first principles. Our notion of a minimal realization is closely related to that of a linear recurring sequence and of a partial realization (as in Mathematical Systems Theory). From this theory, we derive Algorithm MR which computes a minimal realization of a sequence of L elements using at most $L(5L+1)/2$ R–multiplications. We also characterize all minimal realizations of a given sequence in terms of the computed minimal realization.

This algorithm computes the linear complexity of an R sequence, solves non–singular linear systems over R (extending Wiedemann's method), computes the minimal polynomial of an R–matrix, transfer/growth functions and symbolic Padé approximations. There are also a number of applications to Coding Theory.

We thus provide a common framework for solving some well–known problems in Systems Theory, Symbolic/Algebraic Computation and Coding Theory.

AMS subject classifications (1991). Primary: 13P05, 68Q40; secondary: 93B20, 94A55, 94B35.

1 Introduction

Minimal polynomials occur frequently in Coding Theory and Cryptography. Likewise for partial *realizations* in Systems Theory. Both are instances of rational approximation of a finite sequence in $\mathbb{F}_q[[X]]$ and $\mathbb{R}[[X^{-1}]]$ respectively, where $\mathbb{F}_q$ denotes a field with q elements and $\mathbb{R}$ denotes the reals. In this paper, we develop a theory of minimal realizations of a finite sequence from first principles, where the elements of the sequence are from an integral domain R. Our resulting iterative minimal realization algorithm for sequences over R (Algorithm 4.6 — MR) is division–free and requires at most $5L(L+1)/2$ R–multiplications for a sequence of length L. If R is a field $\mathbb{F}$, Algorithm MR performs rational approximation in $\mathbb{F}[[X^{-1}]]$.

*Research supported by U.K. Science and Engineering Research Grant GR/H15141. Current addresses: Dept. Mathematics, University of Queensland, Brisbane 4072, ghn@maths.uq.edu.au.

This work continues the study of sequences over R via R–Laurent series in negative powers begun in Norton (1994). Our approach is to relate minimal realization to the theory of linear recurring sequences (lrs) over R, starting from a basic identity (Proposition 2.2) in $R((X^{-1}))$. (We note that $\mathbb{F}((X^{-1}))$ was used (with different indexing) to study $\mathbb{F}$–sequences in Niederreiter (1988).) We relate minimal realization to finding a recurrence of least degree satisfied by a finite sequence, a problem which has been studied by many authors, most notably Massey (1969). Massey reinterpreted the error–locator of Berlekamp's solution of the "key equation" (Berkekamp (1968)) as a connection polynomial of a shortest–length shift–register (in $\mathbb{F}[[X]]$, with $L \geq 0$) and was thus able to decode BCH codes. See also Trench (1964). A number of authors have also suggested ways of justifying the Berlekamp–Massey (BM) algorithm (Dornstetter (1987), Dai & Wan (1988), Ferrand (1988), Fitzpatrick & Norton (1995), Imamura & Yoshida (1987), Jonckheere & Ma (1989), Lidl & Niederreiter (1983), Mills (1975), Niederreiter (1988), Welch & Scholtz (1979), Zierler (1968)). See also Camion (1989).

We may consider the preceding algorithms as being either *intrinsic* (which do not assume anything about the sequence or its length) or *extrinsic* (which do). The BM algorithm and our minimal realization algorithm are intrinsic. Extrinsic algorithms typically use continued fractions or Hankel matrices and require $L \geq 2\delta$ terms, where δ is the degree of some recurrence satisfied by the first L terms. In Fitzpatrick & Norton (1995), we discussed lrs over a factorial domain and showed that the XPRS algorithm (which generalizes the Extended Euclidean algorithm) could be applied to compute the minimal polynomial — when "enough" terms are known. That approach is therefore also extrinsic.

Extensions of the BM algorithm to more general partial realization problems have appeared (Dickinson *et al.* (1974), Sain (1975)). Kalman has solved a parametrized partial realization problem by using Hankel matrices and by relating parametrized intermediate minimal realizations to continued fractions / the Euclidean algorithm in $\mathbb{F}[X]$ (Kalman (1979)). See also Chapter 10 of Kalman *et al.* (1969). Minimal realizations over domains are also discussed in Rouchaleau & Sontag (1979).

The basic theory, constructions and applications of minimal polynomials are given in Section 3. We can now compute the minimal polynomial (using Algorithm MR — 3.19 or 4.2) of an lrs over any domain without polynomial remainder sequence (PRS) constants (*cf.* Fitzpatrick & Norton (1995)). We have verified that the complexity of the first L prime numbers is $\lceil L/2 \rceil$, $L \leq 25, L \neq 7, 8$ and conjecture that the complexity of the first L prime numbers is $\lceil L/2 \rceil$ for L suitably large. We extend the method of Wiedemann (1986) to solve non–singular linear systems over R and to compute the minimal polynomial of an R–matrix.

The minimal polynomial constructions and algorithms are extended to minimal realizations in Section 4. We recompute the example of Kalman (1979) by applying Algorithm MR in $R = \mathbb{Z}[\xi, \eta]$ and compute the transfer function of a convolutional code or a trellis code without using PRS constants (*cf.* Fitzpatrick & Norton (1991), Chan & Norton (1995)). An application of

Algorithm MR to computing Padé approximants appears in Sheppard (1994). Some applications of Algorithm MR to Coding Theory are being written up separately in Norton (1995). We conclude by characterizing all minimal realizations of a given sequence in terms of the one computed by Algorithm MR.

Some of the results of this paper were presented at the 10^{th} British Colloquium on Theoretical Computer Science (Bristol, April 1994) and at Eurocodes '94 (Côte d'Or, France, October 1994).

2 Realizations and Annihilators

In general, we use lower case Roman letters for elements, Greek letters for functions, and short names for sets. $\mathbb{N} = \{0, 1, 2, \ldots\}$ and we use standard notation to describe intervals in the integers $\mathbb{Z}$; thus $\mathbb{N} = [0, \infty)$.

R denotes an (integral) domain with 1. The letters f, f', g will *always* denote $f, f', g \in R[X]$; δf denotes the degree of f; $\delta 0 = -\infty$. We write $f|g$ if g is a multiple of f.

We will often work in $R((X^{-1}))$ i.e. with Laurent series in X^{-1} over R, which contains $R[X]$. In addition, $R[X]$ acts on $R((X^{-1}))$ by multiplication in such a way that $R[X]$ is the standard $R[X]$–module.

For $F \in R((X^{-1})) \setminus \{0\}$, λF denotes its leading coefficient and the *support* of F is $\mathrm{Supp}(F) = \{a \in \mathbb{Z} : F_a \neq 0\}$, where F_a denotes a coefficient of F; $\mathrm{Supp}(0) = \emptyset$.

As in Niederreiter (1988), we use the (exponential) valuation on $R((X^{-1}))$, which extends the degree function on $R[X]$; for convenience, we also denote it by δ. Thus for $F \in R((X^{-1}))\setminus\{0\}$, $\delta F = \max \mathrm{Supp}(F)$. The following well–known properties of δ on $R((X^{-1}))$ will be used without further ado: $\delta FG = \delta F + \delta G$, $\delta(F + G) \leq \max\{\delta F, \delta G\}$ and $\delta(F + G) = \max\{\delta F, \delta G\}$ if $\delta F \neq \delta G$.

We denote by $S^1(R)^-$ the set of functions $-\mathbb{N} \to R$ i.e. the set of R–sequences indexed by $-\mathbb{N}$, and the value $s(a) \in R$ is written s_a. If addition and scalar product are defined componentwise, $S^1(R)^-$ becomes a unitary R–module. The *generating function* of $s \in S^1(R)^-$ is $\Gamma(s) = \sum_{a \leq 0} s_a X^a \in R[[X^{-1}]]$.

DEFINITION 2.1 *We call* $[1, \delta f]$ *the* border *of* $(-\infty, 0]$ *in* $(-\infty, \delta f]$ *and*

$$\beta(f, s)(X) = \sum_{1 \leq a \leq \delta f} (f\Gamma(s))_a X^a$$

the border polynomial *of* f *and* s.

There is an action of $R[X]$ on $S^1(R)^-$: if $f = \sum_{a=0}^{\delta f} f_a X^a$ then

$$(f \circ s)_b = \sum_{a=0}^{\delta f} f_a s_{b-a} \text{ where } b \leq 0.$$

This makes $S^1(R)^-$ into an $R[X]$–module. Recall that f is a *characteristic polynomial* of s if $f \circ s = 0$, $\mathrm{Ann}(s) = \{f : f \circ s = 0\}$ is the *annihilator ideal* of s, and s is *a linear recurring sequence (lrs)* if $\mathrm{Ann}(s) \neq (0)$. Recall also that if $f \in \mathrm{Ann}(s)$ and $\delta = \delta f \geq 1$, then $\lambda f s_b = -\sum_{a=0}^{\delta-1} f_a s_{b+\delta-a}$ for all $b \leq -\delta$. It was shown in Theorem 4.4 of Fitzpatrick & Norton (1995), Theorem 4.4 of Norton (1994) that if s is an lrs over a factorial domain R, then $\mathrm{Ann}(s)$ has a *primitive generator* $\gamma(s)$, unique up to a unit of R.

The following basic (and easily proved) identity relating terms defined above is the main reason for introducing the action $\circ$:

PROPOSITION 2.2 *In* $R((X^{-1}))$, $f\Gamma(s) = \beta(f, s) + \Gamma(f \circ s)$.

This coherence of $R[X]$–modules avoids special arguments involving the order of Laurent or power series, avoids the usual characterization of $f \in \mathrm{Ann}(s)$ in terms of the reciprocal of f and its degree, and simplifies many of our results (and their proofs). Clearly s is an lrs iff $f\Gamma(s) = \beta(f, s)$ for some non–zero f iff $\Gamma(s)$ is the rational function $\beta(f, s)/f$ for some non–zero f; see also Lemmas 1 and 2, p39 of Niederreiter (1988) when R is a field.

Proposition 2.2 is actually the case $n = 1$ of the decomposition formula for n–dimensional sequences of Norton (1994), Section 3. In the general case $n \geq 1$, $f\Gamma(s)$ has 2^n summands and the border polynomial of Definition 2.1 is written as $\beta_0(f, s)$. See Norton (1994) for details.

We now develop some properties of the border polynomial to be used below.

PROPOSITION 2.3 *(a) For* $r \in R$, $\beta(r, s) = 0$

(b) if $\delta f \geq 1$, δf *terms of* s *are needed to compute* $\beta(f, s)$, *and* $\delta\beta(f, s) \leq \delta f$

(c) β *is linear in each argument*

(d) For $d \geq 0$, $\beta(X^d, s) = \sum_{0 \leq a \leq d-1} s_{-a} X^{d-a}$

(e) $\beta(fg, s) = f\beta(g, s) + \beta(f, g \circ s)$

(f) If $s \neq 0$ *and* $f \in Ann(s) \setminus \{0\}$, *then* $\beta(f, s) \neq 0$.

PROOF. Parts (a) to (d) are straightforward. It suffices to prove part (e) for $f = X^d$ and $g = X^e$ from the linearity of β. The result is then a simple application of (d). To prove part (f), if $f \in \mathrm{Ann}(s) \setminus \{0\}$ and $\beta(f, s) = 0$, then $f\Gamma(s) = 0$. Since $R((X^{-1}))$ is a domain, $\Gamma(s) = 0$ i.e. $s = 0$. □

REMARK 2.4 *Our border polynomial corresponds to the* polynome initial *of Definition 1.3.5 of Ferrand (1988), defined using Newton's divided differences. Propositions 2.2, 2.3 can be proved using the* polynome initial, *but this requires some "summationology" and a product formula for divided differences.*

We have seen that the generating function of an lrs is always a certain rational function, and are interested in to what extent the "generating function" of a *finite* sequence can be "realized".

$(s|L)$ will always denote a sequence of L elements of R, with $L \geq 1$ and indexed by $0, -1, \ldots, -L+1$. We call $\Gamma(s|L) = \sum_{a=-L+1}^{0} s_a X^a$ the *generating function* of $(s|L)$.

DEFINITION 2.5 *We say that* (f,g) is a realization of $(s|L)$ *or that* (f,g) realizes $(s|L)$ *if* $f, g \in R[X], f \neq 0, Supp(g) \subseteq [1, \delta f]$ *and* $\delta(f\Gamma(s|L) - g) \leq -L + \delta f$.

It is easy to see that if $f \neq 0$, then (f,g) realizes $(s|L)$ iff there is an $h \in R[X]$ such that $\delta h \leq \delta f - 1$ and the order of $f^*\Gamma(s)(X) - h$ is at least L, where $f^* = X^{\delta f} f(X^{-1})$ denotes the reciprocal of f. Over a field $\mathbb{F}$, realization is equivalent to rational approximation in $\mathbb{F}[[X^{-1}]]$.

From Proposition 2.2, we can always write

$$f\Gamma(s|L) - \beta(f,s) = \Gamma(f \circ s) + pX^{-L+\delta f} = \sum_{i=-L+\delta f+1}^{0} (f \circ s)_i X^i + qX^{-L+\delta f}$$

for some $p, q \in R[[X^{-1}]]$. So if $\delta f \geq L$, the sum vanishes and *any* $(f, \beta(f,s))$ realizes $(s|L)$. We therefore concentrate on realizations (f,g) with $\delta f \leq L-1$. It also follows from Proposition 2.2 that if s is an lrs and $f \in \mathrm{Ann}(s)$, then $(f, \beta(f,s))$ realizes $(s|L)$ for any $L \geq 1$.

PROPOSITION 2.6 *If* (f,g) *realizes* $(s|L)$ *and* $\delta f \leq L$, *then* $g = \beta(f,s)$.

PROOF. By Proposition 2.2, $\beta(f,s) + \Gamma(f \circ s) = g + p(X^{-1})X^{-L+\delta f}$ for some $p \in R[[X]]$. Let $h = \beta(f,s) - g$. Then $X|h$ and $\delta h \leq 0$, so that $h = 0$ and $g = \beta(f,s)$. □

We now define the "annihilators of $(s|L)$" and relate them to realizations of $(s|L)$.

DEFINITION 2.7 *If* $\delta f \leq L-1$, *we set* $f \circ (s|L) = (f \circ s | L - \delta f)$. *We say that* f *is a* characteristic polynomial *for* $(s|L)$, *and write* $f \in Ann(s|L)$, *if either (a)* $\delta f \geq L$ *or (b)* $\delta f \leq L-1$ *and* $f \circ (s|L) = 0$.

Note that if $\delta f \leq L-1$, then $L - \delta f \geq 1$, so that $(f \circ s | L - \delta f)$ makes sense; further, $f \circ (s|L) = 0$ means that $(f \circ s)_a = 0$ for $-(L-1-\delta f) \leq a \leq 0$.

PROPOSITION 2.8 *Let* $L \geq 2$ *and* $f = \sum_{a=0}^{\delta} f_a X^a$, *where* $1 \leq \delta = \delta f \leq L-1$. *Then the following are equivalent:*

(a) $f \in Ann(s|L)$

(b) λf $s_b = -\sum_{a=0}^{\delta-1} f_a s_{b+\delta-a}$ *for* $-(L-1) \leq b \leq -\delta$

(c) $\delta\Gamma(f \circ s) \leq -L + \delta f$

(d) $(f, \beta(f,s))$ *realizes* $(s|L)$

(e) for some $g \in R[X]$, (f,g) *realizes* $(s|L)$.

PROOF. (a) $\Longleftrightarrow$ (b): $f \in \mathrm{Ann}(s|L) \Longleftrightarrow \lambda f\ s_{b-\delta} = -\sum_{a=0}^{\delta-1} f_a s_{b-a}$ for $-(L-1-\delta) \le b \le 0 \Longleftrightarrow \lambda f s_b = -\sum_{a=0}^{\delta-1} f_a s_{b+\delta-a}$ for $-(L-1) \le b \le -\delta$. (a) $\Longleftrightarrow$ (c) is obvious; (c) $\Longleftrightarrow$ (d) follows easily from $f\Gamma(s|L) - \beta(f,s) = \Gamma(f \circ s) - f \cdot (\sum_{i \le -L} s_i X^i)$ and (d) $\Longleftarrow$ (e) was shown in Proposition 2.6. □

It follows that realizing $(s|L)$ can be broken down into two steps (a) finding an annihilating polynomial f with $\delta f \le L-1$ (if one exists) or taking f to be any polynomial of degree L and (b) computing its border polynomial $\beta(f,s)$.

PROPOSITION 2.9 *(a) If $f \in Ann(s|L)$ and $g \in R[X]$, then $gf \in Ann(s|L)$*

(b) $Ann(s|L+1) \subseteq Ann(s|L)$

(c) if $f \in Ann(s|L)$ and $\delta f \le L$, then $f \in Ann(s|L+1) \Longleftrightarrow (f \circ s)_{-(L-\delta f)} = 0$

PROOF. (a) Firstly, $((gf) \circ s)_a = (g \circ (f \circ s))_a = \sum_{b=0}^{\delta g} g_b (f \circ s)_{a-b}$. Suppose $-(L-1-\delta(gf)) \le a \le 0$. Then for $0 \le b \le \delta g$, $-(L-1-\delta f) + b \le -(L-1-\delta(gf)) \le a$ and so $-(L-1-\delta f) \le a-b \le 0$ and $((gf) \circ s)_a = 0$.

(b) Let $f \in \mathrm{Ann}(s|L+1)$. If $\delta f \ge L$, then $f \in \mathrm{Ann}(s|L)$; otherwise if $(f \circ s|L+1-\delta f) = 0$, then $(f \circ s|L-\delta f) = 0$. Thus $\mathrm{Ann}(s|L+1) \subseteq \mathrm{Ann}(s|L)$.

(c) Omitted. □

We now come to a key concept (*cf.* Massey (1969)):

DEFINITION 2.10 *Let $f \in Ann(s|L)$. We define the* discrepancy $\Delta(s|L+1)(f)$ *by*

$$\begin{aligned} \Delta(s|L+1)(f) &= (f \circ s)_{-(L-\delta f)} && \text{if } \delta f \le L \\ \Delta(s|L+1)(f) &= 0 && \text{otherwise.} \end{aligned}$$

It is clear that if $f \in \mathrm{Ann}(s|L)$, then $f \in \mathrm{Ann}(s|L+1) \Longleftrightarrow \Delta(s|L+1)(f) = 0$; $\Delta(s|L+1)(f)$ is the obstruction to $f \in \mathrm{Ann}(s|L+1)$.

We end this section with a simple application of the theory so far (*cf.* Justification 3.6.3(c), p40 of Ferrand (1988)): $(s|L)$ is a *geometric sequence* if $L \ge 2, s_0 \ne 0$ and $s_a = r^{-a} s_0$ for $-(L-1) \le a \le -1$ and some $r \in R \setminus \{0\}$ (the common ratio).

THEOREM 2.11 *Let $L \ge 2$, $s_0, r \in R \setminus \{0\}$, and let $\Delta = \Delta(s|L+1)(X-r)$.*

(a) $(s|L)$ is a geometric sequence with common ratio $r \Longleftrightarrow X - r \in Ann(s|L)$

(b) $s_0 X^{L-1}(X-r) - \Delta \in Ann(s|L+1)$

(c) $\beta(s_0 X^{L-1}(X-r) - \Delta, s) = s_0^2 X^{L-1}$

(d) If $(s|L)$ is geometric but $(s|L+1)$ is not and $f \in Ann(s|L+1)$ then $\delta f \geq L$.

PROOF. Parts (a)–(c) are straightforward.

(d) If $(s|L+1)$ is not a geometric sequence with common ratio r, then $\Delta \neq 0$, and if $f \in \mathrm{Ann}(s|L+1)$, then $\delta = \delta f \geq 1$ since $s_0 \neq 0$. Suppose that $\delta \leq L-1$. We will show that $\Delta = 0$, for a contradiction. Since $(s|L)$ is geometric, $s_{-b} = r^b s_0$ for $1 \leq b \leq \delta$ and

$$f(r)s_0 = (\sum_{b=0}^{\delta} f_b r^b)s_0 = \sum_{b=0}^{\delta} f_b s_{-b} = (f \circ s)_0 = 0.$$

Then $f(r) = 0$ because $s_0 \neq 0$. Let $\lambda = \lambda f$. Now

$$\begin{aligned}
0 &= (f \circ s)_{-L+\delta} = \sum_{a=0}^{\delta-1} f_a s_{-L+\delta-a} + \lambda s_{-L} = \sum_{a=0}^{\delta-1} f_a r^{L-\delta+a} s_0 + \lambda s_{-L} \\
&\langle \text{ since } -(L-1) \leq -(L-\delta) - a \leq -1 \rangle \\
&= r^{L-\delta}(f(r) - \lambda r^{\delta})s_0 + \lambda s_{-L} \\
&= \lambda(s_{-L} - r^L s_0) \ \langle \text{since } f(r) = 0 \rangle \\
&= \lambda(s_{-L} - r s_{-L+1}) \ \langle \text{since } r^{L-1} s_0 = s_{L+1} \rangle \\
&= \lambda\Delta
\end{aligned}$$

which is impossible. □

3 Minimal polynomials

3.1 The constructions

We define the notions of minimal realization, minimal polynomial and $\kappa(s|L)$, the complexity of $(s|L)$. We then prove an important inequality for $\kappa(s|L)$ and apply it to our first construction of minimal polynomials. The second construction uses the notion of the antecedent $\alpha(s|L)$ of $\kappa(s|L)$. Combining these two constructions will yield an iterative algorithm to compute a minimal polynomial of $(s|L)$.

DEFINITION 3.1 *A realization (f, g) of $(s|L)$ is called* minimal *if $\delta f = \min\{\delta f' : (f', g') \text{ realizes } (s|L)\}$. The minimal value of this degree is called the* complexity $\kappa(s|L)$. *$Min(s|L)$ denotes the set of minimal polynomials of $(s|L)$.*

Since the theory is essentially known for fields, the connection between the minimal polynomials of a sequence over a domain R and its minimal polynomials over the fraction field of R is of interest. This is summarised in the following proposition, the easy proof of which is omitted.

PROPOSITION 3.2 *Let $(s|L)$ be a sequence over R, let R' denote the fraction field of R, and let $(s'|L)$ be $(s|L)$ considered as a sequence over R'. Then*

(a) $Ann(s|L) \subseteq Ann(s'|L)$

(b) If $f \in Ann(s'|L)$ and $r \in R$ clears denominators in f, then $rf \in Ann(s|L)$

(c) $Min(s|L) \subseteq Min(s'|L)$

(d) If $f \in Min(s'|L)$ and $r \in R$ clears denominators in f, then $rf \in Min(s|L)$.

The next result implies that, once we can compute a minimal realization of a sequence over R, we can compute a minimal realization of a sequence over the fraction field of R using computations in R only (thus avoiding gcd computations in the fraction field for example):

PROPOSITION 3.3 *Let R' be the fraction field of R and let $(s'|L)$ be a sequence over R'. For $0 \leq i \leq L-1$, define $s_i = ds'_i$ where $d \in R$ clears denominators in $s'_0, \ldots, s'_{-L+1}$. If $(f, \beta(f,s))$ is a minimal realization of $(s|L)$ over R, then $(f, \beta(f,s'))$ is a minimal realization of $(s'|L)$ over R'.*

PROOF. Let $(f, \beta(f,s))$ be a minimal realization of $(s|L)$. By the linearity of $\Gamma(\ |L)$ and $\beta(f,\ )$,

$$d\,(f\Gamma(s'|L) - \beta(f,s')) = (f\Gamma(s|L) - \beta(f,s)) = p(X^{-1})X^{-L+\delta f}$$

for some $p \in R[[X]]$. Thus $(f, \beta(f,s'))$ realizes $(s'|L)$. Hence if $g \in \mathrm{Min}(s'|L)$, then $\delta f \geq \delta g$. Also, if $r \in R$ clears denominators in g, then $rg \in \mathrm{Ann}(s|L)$ and so $\delta g = \delta(rg) \geq \delta f$, as required. □

We now return to sequences over R.

EXAMPLE 3.4 *If $L \geq 2$ and $(s|L)$ is a geometric sequence but $(s|L+1)$ is not, $s_0X^{L-1}(X-r) - \Delta(s|L+1)(X-r) \in Min(s|L+1)$ by Theorem 2.11.*

We continue with some elementary properties of $\kappa(s|L)$.

PROPOSITION 3.5 *(a) $\kappa(s|L) \leq \kappa(s|L+1)$*

(b) $\kappa(s|L) \leq L$

(c) Let $L \geq 1, (s|L) = 0$ and $s_{-L} \neq 0$. If $f \in Ann(s|L+1)$ then $\delta f \geq L+1$.

PROOF.

(a): $\mathrm{Ann}(s|L+1) \subseteq \mathrm{Ann}(s|L)$ so that if $f \in \mathrm{Ann}(s|L+1)$, then $\kappa(s|L) \leq \delta f$. Thus $\kappa(s|L) \leq \kappa(s|L+1)$.

(b): $L \leq \delta f \Rightarrow f \in \mathrm{Ann}(s|L)$.

(c): If $f \in \mathrm{Ann}(s|L+1)$ then $\delta = \delta f \geq 1$ since R is a domain and $s_{-L} \neq 0$. If $\delta f \leq L$ then $0 \leq a \leq \delta - 1$ implies $-(L-1) \leq -(L-\delta) - a \leq 0$ and so

$$0 = (f \circ s)_{-(L-\delta)} = \sum_{a=0}^{\delta-1} f_a s_{-(L-\delta)-a} + \lambda f s_{-L} = \lambda f \ s_{-L}$$

which is a contradiction. □

If $\kappa(s|L) = 0$ or $\kappa(s|L) = L$, we trivially have $\mathrm{Min}(s|L) = R \setminus \{0\}$ or $\mathrm{Min}(s|L) = \{f : \delta f = L\}$.

PROPOSITION 3.6 *Let $\mu \in Min(s|L)$. If $\Delta(s|L+1)(\mu) = 0$, then $\mu \in Min(s|L+1)$ and $\kappa(s|L+1) = \kappa(s|L)$.*

PROOF. We have $\mu \in \mathrm{Ann}(s|L+1)$ and so $\delta\mu = \kappa(s|L) \leq \kappa(s|L+1) \leq \delta\mu$. □

The following key result was proved for fields in Massey (1969), where it was noted that the result was true for commutative rings ([*loc. cit.*, VII Remarks]). The reader may easily verify that the result fails if there are divisors of zero; for completeness, we prove the result for commutative domains.

THEOREM 3.7 *Let $L \geq 2$. If $f \in Ann(s|L) \setminus Ann(s|L+1)$ and $f' \in Ann(s|L+1)$, $\delta f' \geq 1$, then $\delta f' \geq L + 1 - \delta f$.*

PROOF. If $\delta f = 0$, the result follows from Proposition 3.5(c). Suppose now that $\delta f \geq 1$. We will show that $f' \in \mathrm{Ann}(s|L+1)$ and $\delta f' \leq L - \delta f$ leads to a contradiction. Set $\delta = \delta f$, $\delta' = \delta f'$. Since $\delta + \delta' \leq L$, $\delta \leq L - 1$ and $\delta' \leq L - 1$. Hence by Proposition 2.8, we can write

$$\lambda f \ s_a = -\sum_{b=0}^{\delta-1} f_b s_{a+\delta-b} \text{ for } -(L-1) \leq a \leq -\delta$$

and

$$\lambda f' \ s_a = -\sum_{b=0}^{\delta'-1} f'_b s_{a+\delta'-b} \text{ for } -L \leq a \leq -\delta'.$$

Since $f \notin \mathrm{Ann}(s|L+1)$ and $\delta \leq L - 1$, $\Delta = \Delta(s|L+1)(f) \neq 0$. Letting $\lambda = \lambda f$, $\lambda' = \lambda f'$:

$$\begin{aligned}
\lambda'\Delta &= \lambda' \ (f \circ s)_{-L+\delta} = \lambda' \left(\sum_{a=0}^{\delta-1} f_a \ s_{-L+\delta-a} + \lambda s_{-L} \right) \\
&= \sum_{a=0}^{\delta-1} f_a (\lambda' s_{-L+\delta-a}) + \lambda'\lambda s_{-L} \\
&= -\sum_{a=0}^{\delta-1} f_a \left(\sum_{b=0}^{\delta'-1} f'_b \ s_{(-L+\delta-a)+\delta'-b} \right) + \lambda'\lambda s_{-L} \\
&\langle \quad \text{since } -L < -L + \delta - a \leq -L + \delta \leq -\delta' \rangle
\end{aligned}$$

$$
\begin{aligned}
&= -\sum_{b=0}^{\delta'-1} f'_b \left(\sum_{a=0}^{\delta-1} f_a \; s_{(-L+\delta'-b)+\delta-a} \right) + \lambda'\lambda s_{-L} \\
&= \sum_{b=0}^{\delta'-1} f'_b(\lambda s_{-L+\delta'-b}) + \lambda'\lambda s_{-L} \\
&\langle \quad \text{since } -(L-1) \le -L+\delta'-b \le -L+\delta' \le -\delta \rangle \\
&= \lambda \sum_{b=0}^{\delta'-1} f'_b \; s_{-L+\delta'-b} + \lambda'\lambda s_{-L} = -\lambda\lambda' s_{-L} + \lambda'\lambda s_{-L} = 0
\end{aligned}
$$

since $-L \le -\delta'$, which is the required contradiction. □

We can now prove the first main result for constructing minimal polynomials. Henceforth, for $\mu_L \in Min(s|L)$, we write $\Delta = \Delta(s|L+1)(\mu_L)$.

THEOREM 3.8 *Suppose that* $\kappa(s|L) = \kappa(s|1)$. *If* $\Delta \neq 0$ *for some* $\mu_L \in Min(s|L)$, *let*

$$
\begin{aligned}
&\mu_{L+1} = s_0 X^{L-1} \mu_L - \Delta \;\; \textit{if } s_0 \neq 0 \\
&\mu_{L+1} = f_{L+1} \qquad\qquad \textit{if } s_0 = 0
\end{aligned}
$$

where $f_{L+1} \in R[X]$ *is any polynomial of degree* $L+1$. *Then* $\mu_{L+1} \in Min(s|L+1)$ *and* $\kappa(s|L+1) = L+1-\kappa(s|L) = \max\{\delta\mu_L, L+1-\delta\mu_L\}$.

PROOF. Suppose first that $s_0 = 0$. Then $\kappa(s|L) = \kappa(s|1) = 0$, $\mu_L = r$ for some $r \in R \setminus \{0\}$, $s_a = 0$ for $-(L-1) \le a \le 0$, and $\Delta = r s_{-L}$, so that $s_{-L} \neq 0$. By Proposition 3.5(c), $\kappa(s|L+1) = L+1$ and $\kappa(s|L+1) = L+1-\kappa(s|L) = \max\{\delta\mu_L, L+1-\delta\mu_L\}$ in this case.

Suppose now that $s_0 \neq 0$. Then $\kappa(s|L) = 1$ and clearly $\delta\mu_{L+1} = L$. As in Theorem 2.11(b), $\mu_{L+1} \in \mathrm{Ann}(s|L+1)$. If $L = 1$, then $X \in \mathrm{Min}(s|1), \Delta = s_{-1}$, $\mu_2 = s_0 X - s_{-1} \in \mathrm{Min}(s|2)$ and $\kappa(s|2) = \kappa(s|1) = 1 = L+1-\kappa(s|1)$. If $L \ge 2$, Theorem 3.7 implies that $\kappa(s|L+1) \ge L+1-\delta\mu_L = L = \delta\mu_{L+1} \ge \kappa(s|L+1)$. Thus $\mu_{L+1} \in \mathrm{Min}(s|L+1)$ and $\kappa(s|L+1) = L = L+1-\kappa(s|L) = \max\{\delta\mu_L, L+1-\delta\mu_L\}$. □

EXAMPLE 3.9 $(s|3) = 1, 1, 2$ *has* $\mu_1 = X, \mu_2 = X-1, \mu_3 = X(X-1)-1 = X^2-X-1$. *Here, only three terms suffice to find a minimal polynomial; Algorithm MINPOL of Fitzpatrick & Norton (1995) required an upper bound* δ *(viz.* 2*) on the degree of some characteristic polynomial, and* 2δ *terms.*

We now know how to construct minimal polynomials when either (a) $\Delta = 0$ or (b) $\Delta \neq 0$ and $\kappa(s|1) = \kappa(s|L)$. The following definition (suggested by Lemma 6, p217 of Sakata (1990)) is very useful for simplifying the remaining case (c): "$\Delta \neq 0$ and $\kappa(s|1) < \kappa(s|L)$".

DEFINITION 3.10 *Suppose that* $L \geq 2,\ 1 \leq M \leq L-1,\ f \in Ann(s|M) \setminus Ann(s|M+1)$ *and* $g \in Ann(s|L) \setminus Ann(s|L+1)$. *Let* $\Delta_f = \Delta(s|M+1)(f)$ *and* $\Delta_g = \Delta(s|L+1)(g)$ *be the respective (non–zero) discrepancies, and set* $\delta = \max\{\delta g, L - M + \delta f\}$. *We define*

$$[g, f] = \Delta_f\ X^{\delta - \delta g} g - \Delta_g\ X^{\delta - L + M - \delta f} f.$$

It is easy to see that $[g, f]$ is a well–defined polynomial with $\delta[g, f] = \delta \leq L$; $[\ ,\ ]$ was suggested by the commutator $[\Delta_g, \Delta_f] = \Delta_g \Delta_f - \Delta_f \Delta_g$ which appears naturally in the proof of:

PROPOSITION 3.11 *Let* $L \geq 2$. *If* $1 \leq M \leq L-1,\ f \in Ann(s|M),\ g \in Ann(s|L)$ *and* $[g, f]$ *is defined, then* $[g, f] \in Ann(s|L+1)$.

PROOF. Put $h = [g, f],\ \delta = \delta h$ and $\epsilon = L - M + \delta f$. For $a \leq 0$,

$$(h \circ s)_a = \Delta_f (g \circ s)_{a - \delta + \delta g} - \Delta_g (f \circ s)_{a - \delta + \epsilon}$$

If $-(L - 1 - \delta) \leq a \leq 0$, then $(h \circ s)_a = 0$ since $-(L - 1 - \delta g) \leq a - \delta + \delta g \leq -\delta + \delta g \leq 0$ and $-(M - 1 - \delta f) \leq a - \delta + \epsilon \leq -\delta + \epsilon \leq 0$. If $a = -(L - \delta)$, then $a - \delta + \delta g = -(L - \delta g)$ and $a - \delta + \epsilon = -(M - \delta f)$, so that $(h \circ s)_{-(L-\delta)} = -[\Delta_g, \Delta_f] = 0$ and so $h \in \mathrm{Ann}(s|L+1)$. □

In case (c), we are interested in constructing a $[g, f] \in \mathrm{Min}(s|L+1)$. For given g, it is natural to try to find an $M < L$ and an $f \in \mathrm{Ann}(s|M)$ with $-M + \delta f$ minimal.

DEFINITION 3.12 *Let* $L \geq 2$. *If* $\kappa(s|1) < \kappa(s|L)$, *we define* $\alpha(s|L)$, *the* antecedent *of* $\kappa(s|L)$, *by*

$$\alpha(s|L) = \max\{i : 1 \leq i \leq L-1\ \&\ \kappa(s|i) < \kappa(s|L)\}$$

and if $\mu_\alpha \in Min(s|\alpha(s|L))$, *we set* $\Delta_\alpha = \Delta(s|\alpha(L)+1)(\mu_\alpha)$.

Thus $\alpha(s|L)$ is the last "position" where $\kappa(s|\alpha) < \kappa(s|L)$; $\alpha(s|L)$ is not defined if $\kappa(s|1) = \ldots = \kappa(s|L)$. The next result shows how to construct a minimal polynomial in our remaining case (c):

THEOREM 3.13 *Let* $L \geq 2,\ \kappa(s|L) > \kappa(s|1)$ *and* $\alpha = \alpha(s|L)$. *If* $\mu_\alpha \in Min(s|\alpha) \setminus Ann(s|\alpha+1)$ *and* $\mu_L \in Min(s|L) \setminus Ann(s|L+1)$, *then* $\delta\mu_L = \alpha + 1 - \delta\mu_\alpha,\ [\mu_L, \mu_\alpha] \in Min(s|L+1)$ *and* $\delta[\mu_L, \mu_\alpha] = \max\{\delta\mu_L, L + 1 - \delta\mu_L\}$.

PROOF. We already know that $[\mu_L, \mu_\alpha] \in \mathrm{Ann}(s|L+1)$ from Proposition 3.11. We prove the theorem by induction on L, along with the additional hypothesis that for $1 \leq i \leq L-1,\ \kappa(s|i) <$

$\kappa(s|i+1) \Rightarrow \kappa(s|i+1) = i+1-\kappa(s|i)$. (If $L = 2, i = 1$ and $\kappa(s|2)$ is always equal to $2 - \kappa(s|1)$ by Theorem 3.8.) Suppose then that $L \geq 2$ and inductively that for $1 \leq i \leq L-1, \kappa(s|i) < \kappa(s|i+1) \Rightarrow \kappa(s|i+1) = i+1-\kappa(s|i)$.

Put $\kappa = \kappa(s|L)$, $f = \mu_\alpha$, $g = \mu_L$ and $h = [g, f]$. Since $1 \leq \alpha \leq L-1, \delta\mu_L = \kappa = \kappa(s|\alpha+1) = \alpha + 1 - \delta\mu_\alpha$ and so $-\alpha + \delta f = 1 - \kappa$. So $\delta h = \max\{\kappa, L+1-\kappa\}$. Since $\delta g = \kappa > \kappa(s|\alpha) \geq 0$ and $g \in \mathrm{Ann}(s|L) \setminus \mathrm{Ann}(s|L+1)$, Theorem 3.7 implies that $\delta h \geq \kappa(s|L+1) \geq L+1-\delta g = L+1-\kappa$. If $\kappa \geq L+1-\kappa$, then $\kappa = \delta h \geq \kappa(s|L+1) \geq \kappa$. If $\kappa \leq L+1-\kappa$, then $L+1-\kappa = \delta h \geq \kappa(s|L+1) \geq L+1-\kappa$ by Theorem 3.7. In either case, $\delta h = \kappa(s|L+1)$ i.e. $h \in \mathrm{Min}(s|L+1)$.

To complete the inductive step, we need to show that $\kappa(s|L) < \kappa(s|L+1) \Rightarrow \kappa(s|L+1) = L+1-\kappa(s|L)$. But if $\delta\mu_L = \kappa(s|L) < \kappa(s|L+1)$, then $\delta h = L+1-\kappa$. □

COROLLARY 3.14 *For $L \geq 2$, $\kappa(s|L+1) = \max\{\kappa(s|L), L+1-\kappa(s|L)\}$.*

PROOF. This is immediate from Theorems 3.8, 3.13. □

REMARK 3.15 *Analogues of Example 3.4 (which is implied by Theorem 3.8) and of Proposition 3.5 form part of the long, complex inductive proof of the "Massey–Berlekamp" algorithm in Section 3.6 of Ferrand (1988). Theorems 2.11, 3.8 and 3.13 were first distilled from this long proof (which begins with $L = 0$) and then generalized to R. Our proof of Theorem 3.13 is however closer to Massey's in spirit and does not use continued fractions (as in Ferrand (1988)) to prove that either $\kappa(s|L+1) = \kappa(s|L)$ or $\kappa(s|L+1) = L+1-\kappa(s|L)$.*

Our next goal is to show how Theorems 3.8 and 3.13 may be combined to yield an (iterative) algorithm to compute a minimal polynomial for $(s|L)$. We first show how the minimal polynomials of Theorem 3.8 may also be written as $\mu_{L+1} = [\mu_L, \mu_\alpha]$ using appropriate choices for α and μ_α.

DEFINITION 3.16 *If $\kappa(s|L) = \kappa(s|1)$, we define the antecedent $\alpha = \alpha(s|L)$ of $\kappa(s|L)$, μ_α and Δ_α as in the following table:*

	α	μ_α	Δ_α
$s_0 = 0$	*-1*	*0*	*1*
$s_0 \neq 0$	*0*	*1*	s_0

If $g \in Ann(s|L) \setminus Ann(s|L+1)$, we define $[g, 0] = X^{L+1-\delta g} g$.

EXAMPLE 3.17 *In Example 3.4, $\alpha = 0, \mu_L = X - r, \mu_\alpha = 1$ and the cited minimal polynomial is $[\mu_L, \mu_\alpha]$.*

PROPOSITION 3.18 *Let* $\kappa(s|L) = \kappa(s|1), \alpha = \alpha(s|L)$. *Then*

(i) $\alpha = \max\{i| -1 \leq i \leq L-1 \ \& \ \kappa(s|i) < \kappa(s|L)\}$ *and if* $\mu_\alpha \neq 0$ *then* $\delta\mu_L = \alpha + 1 - \delta\mu_\alpha$.

(ii) $\mu_{L+1} = [\mu_L, \mu_\alpha] \in Min(s|L+1)$ *and* $\kappa(s|L+1) = \max\{\delta\mu_L, L + 1 - \delta\mu_L\}$.

PROOF.

(i) Let $\alpha' = \max\{i| -1 \leq i \leq L-1 \ \& \ \kappa(s|i) < \kappa(s|L)\}$. We have $\kappa(s|1) = 0 \Longleftrightarrow s_0 = 0 \Longleftrightarrow \alpha = -1$, and $\kappa(s|L) = 0 \Rightarrow \alpha' = -1$. If $\kappa(s|L) = 1$, then $\alpha = 0 = \alpha'$. If $\delta\mu_\alpha \neq 0$, then $s_0 \neq 0$ and $\alpha + 1 - \delta\mu_\alpha = 1 = \delta\mu_1 = \delta\mu_L$.

(ii) If $\alpha = -1$, then $s_0 = 0, \ \mu_L = 1, \ \kappa(s|L) = 0$, and $\mu_{L+1} = X^{L+1} = [\mu_L, \mu_\alpha]$. If $\alpha = 0, \ \mu_\alpha = 1$ and $\mu_{L+1} = s_0 X^{L-1}\mu_L - \Delta(s|L+1)(\mu_L) = [\mu_L, 1]$. Thus part (ii) follows from Theorem 3.8. □

We have now justified the following:

ALGORITHM 3.19 *(MP — Minimal polynomial)*

Input: A domain $R, \ L \geq 1, \ s_0, \ldots, s_{-L+1} \in R$.

Output: $\mu \in Min(s|L)$.

$\alpha := -1; \ \mu_\alpha := 0; \ \Delta_\alpha := 1; \mu_0 := 1;$

for $j := 1$ *to* L *do begin*
    $\Delta_j := \Delta(s|j)(\mu_{j-1});$
    if $\Delta_j = 0$ *then* $\mu_j := \mu_{j-1}$
        else begin $\mu_j := [\mu_{j-1}, \ \mu_\alpha];$
            $\alpha := \alpha(s|j);$ *end;*
    end;

return μ_L.

EXAMPLE 3.20 *Consider the truncated Fibonacci sequence* $(s|5) = 0, 1, 1, 2, 3$. *We give the table which corresponds to applying Algorithm MP:*

j	α	Δ_α	μ_α	Δ	μ
1	*-1*	*1*	*0*	*0*	*1*
2	*-1*	*1*	*0*	*1*	X^2
3	*1*	*1*	*1*	*1*	$X^2 - X$
4	*1*	*1*	*1*	*1*	$X^2 - X - 1$
5	*1*	*1*	*1*	*0*	$X^2 - X - 1$

EXAMPLE 3.21 *Algorithm MP showed that the complexity of the first L prime numbers is equal to* $\lceil L/2 \rceil$*, if* $L \leq 25, L \neq 7, 8$*. (We obtained* $\mu_6 = \mu_7 = X^3 - 2X^2 - 3X + 6$ *and* $\mu_8 = X^5 - 2X^4 - 3X^3 + 12X^2 + 6X - 24$*.)*

The following therefore seems reasonable:

CONJECTURE 3.22 *The complexity of the first L prime numbers is* $\lceil L/2 \rceil$ *for L suitably large.*

We now give an upper bound for the complexity of Algorithm MP.

PROPOSITION 3.23 *Algorithm MP requires at most* $L(3L+1)/2$ *R–multiplications.*

PROOF. The discrepancy $\Delta(s|j)(\mu_j)$ requires at most $\delta\mu_{j-1}+1$ multiplications and (when $\mu_\alpha \neq 0$), computing $[\mu_{j-1}, \mu_\alpha]$ requires at most $(\delta\mu_{j-1}+1)+(\delta\mu_\alpha+1)$ multiplications. Now $\delta\mu_{j-1} \leq j-1$ and $\delta\mu_\alpha \leq \alpha$ by Proposition 3.5 and Definition 3.16, and $\alpha \leq j-2$ by definition. Thus the j^{th} iteration requires at most $3j-1$ multiplications and summing over j yields the stated upper bound. □

3.2 Uniqueness

We show that $\mathrm{Min}(s|L)$ is essentially a singleton if $1 \leq \kappa(s|L) \leq L/2$. Items 3.24, 3.25, 3.27 are based on corresponding results for sequences over fields in Ferrand (1988), but simplified and proved afresh using the basic identity (Proposition 2.2) in $R((X^{-1}))$.

The analogue of the following result for lrs was proved in Lemma 4.2 of Norton (1994).

COROLLARY 3.24 *Let R be factorial. If* $f \in Min(s|L)$ *is primitive,* $\gcd(f, \beta(f,s)/X) = 1$.

PROOF. Without loss of generality, we can assume that s is non–zero and $\delta f \geq 1$. Let $d = \gcd(f, \beta(f,s)/X)$. We have $f\Gamma(s|L) - X\beta(f,s)/X = pX^{-L+\delta f}$ for some $p \in R[[X^{-1}]]$ and d divides the left–hand side. Now $\delta(pX^{-L+\delta f}/d) \leq -L + \delta(f/d)$, so by Proposition 2.8, $f/d \in \mathrm{Ann}(s|L)$. Finally, $\delta(f/d) \leq \delta f$, so that $\delta d = 0$ (otherwise f is not minimal) and since f is primitive, $d = 1$. □

COROLLARY 3.25 *Let* $f, g \in Ann(s|L)$ *where* $\delta f \geq 1$, $\delta g \geq 1$. *If* $\delta f + \delta g \leq L$, *then* $f\beta(g,s) = g\beta(f,s)$.

PROOF. We have $g(f\Gamma(s|L) - \beta(f,s)) = gpX^{-L+\delta f}$ and $f(g\Gamma(s|L) - \beta(g,s)) = fqX^{-L+\delta g}$ for some $p, q \in R[[X^{-1}]]$. Thus

$$g\beta(f,s) - f\beta(g,s) = -gpX^{-L+\delta f} + fqX^{-L+\delta f}$$

where the right–hand side has degree at most $-L+\delta f+\delta g \leq 0$ and the left–hand side is divisible by X. Thus the left–hand side is zero, as required. □

The next result generalizes Proposition 2.3(f) to finite sequences:

LEMMA 3.26 *If* $(s|L) \neq 0$, $f \in Ann(s|L)$ *and* $1 \leq \delta f \leq L-1$, *then* $\beta(f,s) \neq 0$.

PROOF. If $f \in \mathrm{Ann}(s|L), \delta f \geq 1$ and $\beta(f,s)=0$, then by Proposition 2.8, $f\Gamma(s|L) = pX^{-L+\delta f}$ for some $p \in R[[X^{-1}]]$. Hence $\delta f + \delta\Gamma(s|L) \leq -L+\delta f$, i.e. $\delta\Gamma(s|L) \leq -L$ and so $(s|L)=0$. □

We now turn to the determination of $\mathrm{Min}(s|L)$ when $1 \leq \kappa(s|L) \leq L-1$. When R is a field, the following uniqueness result also follows from Theorem 1, p42 of Niederreiter (1988).

COROLLARY 3.27 *Let* R *be factorial. If* $1 \leq \kappa(s|L) \leq L/2$ *and* $f \in Min(s|L)$ *is primitive, then* f *is unique (up to a unit of* R*).*

PROOF. Let $f,g \in \mathrm{Min}(s|L)$. Then $1 \leq \delta g = \delta f$ and $\delta f + \delta g \leq L$, so by Corollary 3.25, $f\beta(g,s)/X = g\beta(f,s)/X$, where $\beta(f,s)$ (and hence $\beta(g,s)$) is non–zero by Lemma 3.26. Since $\gcd(f, \beta(f,s)/X) = 1$ by Corollary 3.24, $f|g$. Similarly, $g|f$. □

3.3 Some applications to lrs

Recall that if R is factorial and s is an lrs, then $\gamma(s)$ denotes a primitive generator of $\mathrm{Ann}(s)$. By Corollary 3.27, if s is a non–zero lrs over a factorial R, then $1 \leq \kappa(s|\delta\gamma(s)) \leq \kappa(s|2\delta\gamma(s)) \leq \delta\gamma(s)$ and so $\mathrm{Min}(s|2\delta\gamma(s))$ consists of the associates of some polynomial.

COROLLARY 3.28 *Let* R *be factorial and let* s *be a non–zero lrs over* R. *Then* $\gamma(s) \in Min(s|2\delta\gamma(s))$.

PROOF. Let $\gamma = \gamma(s)$, let $L = 2\delta\gamma \geq 2$ and let $\mu \in \mathrm{Min}(s|L)$ be primitive. Since s is non–zero, $\delta\mu \geq 1$. Clearly $\gamma \in \mathrm{Ann}(s|L)$, so that $\delta\mu \leq \delta\gamma$ and $\delta\mu + \delta\gamma \leq 2\delta\gamma$. By Corollary 3.25, $\gamma\beta(\mu,s) = \mu\beta(\gamma,s)$ where $\beta(\gamma,s)$ (and hence $\beta(\mu,s)$) is non–zero by Proposition 2.3(f). Now γ is relatively prime to $\beta(\gamma,s)/X$ by Lemma 4.2 of Norton (1994) and so $\gamma|\mu$. Similarly, $\mu|\gamma$ by Corollary 3.24, which yields the result. □

It follows that Algorithm MP can also be used to compute the minimal polynomial of an lrs (without using polynomial remainder sequence constants as in Fitzpatrick & Norton (1995)).

The following result is implicit in Wiedemann (1986) for lrs over any field. We give a proof for factorial domains.

PROPOSITION 3.29 *If* R *is factorial,* $lcm(f, \gamma(s)) = f\gamma(f \circ s)$, *up to a unit of* R.

PROOF. Let $g = \gamma(s)$ and $d = \gcd(f, g)$. Since $g \circ (f \circ s) = (gf) \circ s = f \circ (g \circ s) = 0, f \circ s$ is an lrs and $h = \gamma(f \circ s)$ is also well–defined. Now $(fh) \circ s = (hf) \circ s = h \circ (f \circ s) = 0$, so $g|(fh)$ and hence $\mathrm{lcm}(f, g)|(fh)$. Also, $(g/d) \circ (f \circ s) = (gf/d) \circ s = f/d \circ (g \circ s) = 0$. Therefore $h|(g/d)$ and $(fh)|\mathrm{lcm}(f, g)$. □

Corollary 3.28 and Proposition 3.29 now justify the following algorithm, which generalizes Algorithm 2, p25 of Wiedemann (1986) to factorial domains:

ALGORITHM 3.30 *(lcm of lrs generators)*

Input: Factorial domain R, $L \geq 1, N \geq 2$, $s_0^{(k)}, \ldots, s_{-2L+1}^{(k)} \in R$, *with* $\delta\gamma(s^{(k)}) \leq L$ *for* $1 \leq k \leq N$.

Output: $lcm(\gamma(s^{(1)}), \ldots, \gamma(s^{(N)}))$, *up to a unit of* R.

$g := 1$;

for $k := 0$ *to* $N - 1$ *do* $g_{k+1} = g_k \gamma(g_k \circ s^{(k+1)})$; /* *use Algorithm MP* */

return g_N.

As in Wiedemann (1986), we may exit from the above loop if $\delta g_j = L$ for some j since by Proposition 3.29, $g_k|g_{k+1}$ and $\delta g_k \leq L$ for $1 \leq k \leq N - 1$.

DEFINITION 3.31 *Let* M *be an* $(N \times N)$*–matrix over* R *and let* b *be a non–zero* N*–vector of* R *elements. For* $1 \leq k \leq N$, *let* $w^{(k)} = w(M, b)^{(k)} \in S^1(R)^-$ *be defined by* $w_i^{(k)} = (M^{-i}b)_k$, *for* $i \leq 0$.

Thus if $d = \delta f, (f \circ w^{(k)})_0 = \sum_{i=0}^{d} f_i w_{-i}^{(k)} = \sum_{i=0}^{d} f_i (M^i b)_k$.

PROPOSITION 3.32 *For* $1 \leq k \leq N$, $w(M, b)^{(k)}$ *has a characteristic polynomial of degree at most* N.

PROOF. The Cayley–Hamilton Theorem is valid over R (see Theorem 3.1, p561 of Lang (1993)) and so the minimal polynomial, f say, of M has degree at most N. A simple check shows that $f \in \mathrm{Ann}(w^{(k)})$ for $1 \leq k \leq N$. □

PROPOSITION 3.33 *Let* $\gamma = \sum_{i=0}^{d} \gamma_i X^i \in \bigcap_{k=1}^{N} Ann(w^{(k)})$. *If* $a = -\sum_{i=1}^{d} \gamma_i M^{i-1} b$, *then* $Ma = \gamma_0 b$. *In particular, if* γ_0 *is a unit of* R, $M\gamma_0^{-1} a = b$.

PROOF. For $1 \leq k \leq N, (Ma - \gamma_0\, b)_k = -\sum_{i=1}^{d} \gamma_i (M^i b)_k - \gamma_0 b_k = -\sum_{i=0}^{d} \gamma_i (M^i b)_k - (\gamma \circ (w^{(k)})_0 = 0$ and $Ma = \gamma_0 b$. □

The prime candidate for γ in Proposition 3.33 is of course $\gamma = \mathrm{lcm}(\gamma(s^{(1)}), \ldots, \gamma(s^{(N)}))$ which generates $\bigcap_{k=1}^{N} \mathrm{Ann}(w^{(k)})$, and which we may compute using the $(w^{(k)}|2N)$ by Algorithm 3.30 and Proposition 3.32.

ALGORITHM 3.34 *Input: Factorial R, $(N \times N)$–matrix M and N–vector b over R.*

Output: $\gamma_0 \in R$ and a such that $Ma = \gamma_0 b$.

1. *Compute $(w^{(k)}|2N)$.*
2. *Compute $\gamma = lcm(\gamma(s^{(1)}), \ldots, \gamma(s^{(N)}))$ via Algorithm 3.30.*
3. *Return $-\sum_{i=1}^{d} \gamma_i M^{i-1} b$.*

Wiedemann's method of solving $Mx = b$ is an application of Algorithm 3.34 (using the BM algorithm instead of Algorithm MP in Algorithm 3.30) to a *non–singular* matrix with entries from a field $\mathbb{F}$. (In this case, $\gamma_0 \neq 0$ since $b \neq 0$ and so γ_0 has an inverse in $\mathbb{F}$.) The "Fundamental Iterative Algorithm" of Feng & Tzeng (1991) may also be used instead of Algorithm 2 of Wiedemann (1986).

A useful special case of Algorithm 3.34 is the case $R = A[z]$ where A is a factorial domain and M is non–singular. In this case we obtain the solution to $Mx = b$ as a vector of rational functions a/γ_0 (*cf.* Guiver (1985), Lotti (1992)).

Algorithm 3.30 may also be used to find the minimal polynomial of matrix.

PROPOSITION 3.35 *Let M be an $(N \times N)$–matrix over R and for $1 \leq k \leq N^2$, let $(s^{(k)}|2N)$ be defined by $s^{(k)}_{-i} = M^i_k$ for $0 \leq i \leq 2N-1$. Then the minimal polynomial μ of M is $\gamma = lcm(\gamma(s^{(1)}), \ldots, \gamma(s^{(N^2)}))$, up to a unit of R.*

PROOF. It is clear that $\mu \in \mathrm{Ann}(s^{(k)}|2N)$ for $1 \leq k \leq N^2$ and so $\gamma|\mu$. On the other hand, $\gamma \circ (s^{(k)}|2N) = 0$ for $1 \leq k \leq N^2$ and so $\gamma(M) = 0$. □

Since it seems likely that there is an $O(L \log L)$ version of Algorithm MP (as in Blahut (1983)) we will not consider fast methods of matrix multiplication to reduce the complexity of Algorithm 3.30 here.

4 Minimal realization

4.1 An iterative algorithm

It is clear that we obtain a minimal realization algorithm by simply returning $\beta(\mu_L, s)$ in Algorithm MP. Now it is easy to check (using Proposition 2.3(d) and the linearity of $\beta(\ , s)$) that computation of $\beta(f, s)$ requires at most $\delta f(\delta f + 1)/2$ R–multiplications. Since $\delta\mu_L \leq L$, computing a minimal realization in this way incurs at most $L(L+1)/2$ additional R–multiplications.

In this section we show how to extend Algorithm MP to compute $\beta(\mu_L, s)$ *iteratively.* This is done by first expanding $[\ ,\ ]$ and then expressing $\beta([\ ,\ ], s)$ in terms of the border polynomials of each argument. We will see that computing $\beta(\mu_L, s)$ in this way requires at most L^2 more

R–multiplications than Algorithm MP, so that this iterative method of computing $\beta(\mu_L, s)$ is potentially slightly less efficient. However, the iterative version brings out the underlying similarity of computing μ_L and $\beta(\mu_L, s)$ (*cf.* Dornstetter (1987), Appendix), which could be exploited in a hardware implementation. Further, the form of Algorithm MR shows that μ_L and $\beta(\mu_L, s)$ could even be computed simultaneously.

PROPOSITION 4.1 *If* $m = m(L) = 2\delta\mu_L - L - 1$ *and* $\Delta = \Delta(s|L+1)(\mu_L)$, *then*

$$\begin{aligned} [\mu_L, \mu_\alpha] &= \Delta_\alpha \mu_L - \Delta X^m \mu_\alpha \\ \alpha(s|L+1) &= \alpha(s|L) \\ m(L+1) &= m(L) - 1 \qquad \textit{if } m \geq 0 \end{aligned}$$

$$\begin{aligned} [\mu_L, \mu_\alpha] &= \Delta_\alpha X^{-m} \mu_L - \Delta \mu_\alpha \\ \alpha(s|L+1) &= L \\ m(L+1) &= -m(L) - 1 \qquad \textit{otherwise.} \end{aligned}$$

PROOF. Put $\alpha = \alpha(s|L)$. If $\alpha = -1$, then $\mu_\alpha = 0, \mu_L = \mu_1 = 1$ and $m = -L - 1 < 0$. Thus $[\mu_L, \mu_\alpha] = X^{L+1}$ and $\alpha(s|L+1) = L$, as stated. For $\alpha \geq 0$, $\delta[\mu_L, \mu_\alpha] = \max\{\delta\mu_L, L + 1 - \delta\mu_L\}$ by Theorem 3.13 and Proposition 3.18, and $L+1-\delta\mu_L = L-\alpha+\delta\mu_\alpha$, so $\delta-\delta\mu_L = \max\{0, -m\}$ and $\delta - L + \alpha - \delta\mu_\alpha = \max\{0, m\}$. Now $\delta\mu_{L+1} = \delta\mu_L \iff \delta\mu_L \geq L+1-\delta\mu_L \iff m(L) \geq 0$, and so the form of $[\mu_L, \mu_\alpha]$ follows from Definitions 3.10 and 3.16. Also, $\alpha(s|L+1) = \alpha(s|L) \iff m(L) \geq 0$ and if $m(L) < 0$, $\alpha(s|L+1) = L$ by definition of $\alpha(s|L+1)$. We omit the verification of $m(L+1)$.□

It follows that μ_α, Δ_α are to be updated (to μ_L, Δ respectively) only when $m(L) < 0$. Suppressing indices (but retaining the μ_α, Δ_α notation) and factoring out the update of μ usings swaps yields the following algorithm (where for $j = 1$ and $\Delta = s_0 \neq 0$, we force μ to change from 1 to X by initializing m to -1).

ALGORITHM 4.2 *(MP— Minimal polynomial — expanded version).*

Input: A domain R, $L \geq 1$, $s_0, \ldots, s_{-L+1} \in R$.

Output: $\mu \in Min(s|L)$.

$\mu_\alpha := 0$; $\mu := 1$;
$\Delta_\alpha := 1$; $m := -1$;

for $j := 1$ *to* L *do begin*
    $\Delta := \Delta(s|j)(\mu)$;
    if $\Delta \neq 0$ *then begin if* $m < 0$ *then begin* $m := -m$;

Swap $\{\mu, -\mu_\alpha\}$;

Swap $\{\Delta, \Delta_\alpha\}$;*end;*

$\mu := \Delta_\alpha\ \mu - \Delta\ X^m\ \mu_\alpha$;*end;*

$m := m - 1$;*end;*

return μ.

We may of course dispense with the negation. It is easy to see that if R is a field then adding the statement "$\mu := \mu/\Delta_\alpha$;" makes each μ monic. The interested reader may readily reformulate Proposition 4.1 in terms of loop invariants for Algorithm 4.2 using the j^{th} iterates μ_j, α_j etc. We note that in Ferrand (1988), the analogue of μ_α is initialised to 1 and j ranges from 0 to $L-1$.

EXAMPLE 4.3 *Algorithm 4.2 applied to* $(s|5) = 1,0,1,0,0$ *in GF(2) (Massey (1969), Example p125) is:*

j	Δ_α	μ_α	Δ	m	μ
1	*1*	*0*	*1*	*0*	X
2	*1*	*1*	*0*	*-1*	X
3	*1*	*1*	*1*	*0*	X^2+1
4	*1*	X	*0*	*-1*	X^2+1
5	*1*	X	*1*	*0*	X^3

Observe that Massey obtains 1 as the "connection polynomial", which is the reciprocal of X^3. We remark that Algorithm 4.2 computes the minimal polynomial of the example on p441 of Lidl & Niederreiter (1983) using only 5 terms.

Our iterative minimal realization algorithm rests on the next two results which show that β behaves very neatly with regard to the [,] construction:

PROPOSITION 4.4 *If* $1 \le M \le L-1$, $f \in Ann(s|M)$, $g \in Ann(s|L)$ *and* $[g,f]$ *is defined (as in Definition 3.10), then*

$$\beta([g,f],s) = \Delta_f X^{\delta-\delta g}\beta(g,s) - \Delta_g X^{\delta-L+M-\delta f}\beta(f,s)$$

where $\delta = \max\{\delta g, L-M+\delta f\}$.

PROOF. Put $\epsilon = L - M + \delta f$.

(i) If $\delta = \delta g$, then $\delta - \epsilon \ge 0$ and

$$\beta([g,f],s) \quad = \quad \beta(\Delta_f\ g - \Delta_g X^{\delta-\epsilon}\ f, s) = \Delta_f\ \beta(g,s) - \Delta_g \beta(X^{\delta-\epsilon}\ f, s)$$

$$\begin{aligned} &\langle \quad \text{since } \beta \text{ is linear} \rangle \\ &= \quad \Delta_f\ \beta(g,s) - \Delta_g(X^{\delta-\epsilon}\beta(f,s) + \beta(X^{\delta-\epsilon}, f\circ s)) \end{aligned}$$

by Proposition 2.3(e).

We show that $\beta(X^{\delta-\epsilon}, f\circ s) = 0$. If $\delta-\epsilon = 0$, then $\beta(1,s) = 0$. If $\delta-\epsilon \geq 1$, then

$$\beta(X^{\delta-\epsilon}, f\circ s) = \sum_{a=0}^{\delta-\epsilon-1} (f\circ s)_{-a} X^{\delta-\epsilon-a}.$$

For $\delta = \delta g \leq L-1,\ -(M-1-\delta f) \leq -(\delta-\epsilon-1)$ and so $(f\circ s)_{-a} = 0$ for $0 \leq a \leq \delta-\epsilon-1$. Thus $\beta(X^{\delta-\epsilon}, f\circ s) = 0$ for all $\delta \geq \epsilon$, and $\beta([g,f],s)$ is as required.

(ii) If $\delta = \epsilon$, then $\delta - \delta g \geq 0$ and

$$\beta([g,f],s) = \beta(\Delta_f X^{\delta-\delta g} g - \Delta_g f, s) = \Delta_f \beta(X^{\delta-\delta g} g, s) - \Delta_g \beta(f,s).$$

From Proposition 2.3(e),

$$\beta(X^{\delta-\delta g}\ g, s) = X^{\delta-\delta g}\beta(g,s) + \beta(X^{\delta-\delta g},\ g\circ s).$$

Now $\beta(X^{\delta-\delta g}, g\circ s) = \sum_{a=0}^{\delta-\delta g-1}(g\circ s)_{-a}X^{\delta-\delta g-a}$. Since $\delta \leq L$ implies $-(L-1-\delta g) \leq -(\delta-1-\delta g), (g\circ s)_{-a} = 0$ for $0 \leq a \leq \delta-\delta g-1$ and $\beta([g,f],s)$ is as stated. □

The initialization "$\beta(\mu_{-1},s) = -X$" in Algorithm MR below is required by the following result, which will complete our justification of the iterative minimal realization algorithm. (Addition and polynomial multiplication of minimal realizations will be by component.)

THEOREM 4.5 *Define* $\beta(\mu_{-1},s) = -X$ *and* $m = 2\delta\mu_L - L - 1$. *Then*

$$\beta([\mu_L,\mu_\alpha],s) = \Delta_\alpha\beta(\mu_L,s) - \Delta X^m\beta(\mu_\alpha,s) \text{ if } m \geq 0$$

and

$$\beta([\mu_L,\mu_\alpha],s) = \Delta_\alpha X^{-m}\beta(\mu_L,s) - \Delta\beta(\mu_\alpha,s) \text{ if } m \leq 0.$$

PROOF. For $\alpha \geq 1$, the result follows from Propositions 4.1 and 4.4.

If $\alpha = -1$, then $\mu_\alpha = 0,\ \Delta_\alpha = 1$ and $\mu_L = \mu_1 = 1$, so that $m = -L-1 < 0$ and

$$\beta([\mu_L,\mu_\alpha],s) = \beta(X^{L+1},s) = Xs_{-L} = -\Delta\beta(\mu_{-1},s).$$

If $\alpha = 0$ then $\mu_\alpha = 1, \beta(\mu_\alpha,s) = 0,\ \Delta_\alpha = s_0 \neq 0$ and $\mu_L = \mu_1 = X$. Thus $m = -L+1$ and

$$\beta([\mu_L,\mu_\alpha],s) = \beta(s_0\ X^{L-1}\mu_L - \Delta, s) = s_0\ X^{L-1}\beta(\mu_L,s)$$

by Proposition 2.3. □

ALGORITHM 4.6 *(MR — Iterative minimal realization).*

Input: A domain R, $L \geq 1$, $s_0, \ldots, s_{-L+1} \in R$.

Output: A minimal realization $(\mu_L, \beta(\mu_L, s))$ *for* $(s|L)$.

$(\mu_\alpha, \beta_\alpha) := (0, -X)$; $(\mu, \beta) := (1, 0)$;
$\Delta_\alpha := 1$; $m := -1$;

for $j := 1$ *to* L *do begin*
$\Delta := \Delta(s|j)(\mu)$;
if $\Delta \neq 0$ *then begin if* $m < 0$ *then begin* $m := -m$;
$Swap\{(\mu, \beta), -(\mu_\alpha, \beta_\alpha)\}$;
$Swap\{\Delta, \Delta_\alpha\}$;*end;*
$(\mu, \beta) := \Delta_\alpha(\mu, \beta) - \Delta\ X^m(\mu_\alpha, \beta_\alpha)$;*end;*
$m := m - 1$;*end*;

return (μ, β).

EXAMPLE 4.7 *If we apply Algorithm MR to Example 4.3, we obtain* $\beta_1 = X = \beta_2$, $\beta_3 = X^2 = \beta_4$, $\beta_5 = X(X^2 + 1)$.

It is easy to check that the computation of the border polynomials requires at most L^2 more R–multiplications than in Proposition 3.23:

PROPOSITION 4.8 *Algorithm MR requires at most* $L(5L + 1)/2$ R*–multiplications.*

4.2 Some applications

We begin with Kalman's parametrized partial realization example:

EXAMPLE 4.9 *Algorithm MR with* $R = \mathbb{Z}[\xi, \eta]$ *and* $(s|6) = 1, 1, 1, 2, \xi, \eta$ *yields the minimal realization* $((x-1)\{x^2 + 3x + 7 + [\xi^2 - (x+4)\xi - \eta]\} - 1, x(x^2 + 3x + 7 + [\xi^2 - (x+4)\xi - \eta]))$. *This agrees with formula 3.2 of Kalman (1979), obtained from Algorithm 2.7,* loc. cit. *and the* $\mathbb{R}[x]$*–continued fraction formula p19,* loc. cit. *for all minimal realizations of* $1, 1, 1, 2$ *over the reals* $\mathbb{R}$.

In the next example, we no longer need to know that the sequence has a characteristic polynomial of degree 2:

EXAMPLE 4.10 *If we apply Algorithm MR to* $R = \mathbb{F}_2[y]$ *and* $(s|4) = y, 1, y + 1, y^2 + 1$, *we obtain* $\mu(X) = X(X + y + 1), \beta(X) = X(yX + y^2 + y + 1)$ *(*cf. *Example 5.3 of Fitzpatrick & Norton (1995)).*

It follows from Corollary 3.28 that Algorithm MR also computes the minimal realization of an lrs, provided enough terms are known. Algorithm MR may be used to find path enumerators/transfer functions:

EXAMPLE 4.11 *For the standard* $(2,1,2)$ *convolutional code (Example CC1, p200 of McEliece (1977)), we obtain* $\Gamma = Y^3Z/X^4(X - YZ(Y+1))$ *as the complete path enumerator using* $R = \mathbb{Z}[Y,Z]$ (cf. *p287,* loc. cit.*). For the* 2*–state trellis code of Chan & Norton (1995) with* $a,b,c,d \neq 0$*, the transfer function obtained is* $(aX - (ad-bc))/(X-d)$*, using* $R = \mathbb{Z}[a,b,c,d]$.

REMARK 4.12 *As expected, there are applications of Algorithms 4.2 and MR to Coding Theory (Norton (1995)). In fact, the key equation is more natural in* $\mathbb{F}_q[[X^{-1}]]$ *and so can be solved using Algorithm MR, giving new algorithms for decoding BCH and Reed–Solomon codes. This approach also extends to decoding classical Goppa codes. There is an extension of Algorithm 4.2 to finding a (simultaneous) minimal polynomial of several* $\mathbb{F}$*–sequences, which simplifies the "Fundamental Iterative Algorithm" of Feng & Tzeng (1991) and an extension of Algorithm MR to* $R = \mathbb{Z}/p^e\mathbb{Z}$ *which simplifies Sections 4,5 of Reeds & Sloane (1985).*

It seems likely that Algorithm MR will also apply to computing growth functions of groups (see Brazil (1993)), to the analysis of linear systems (see Chen (1983)) and to solving discrete–time Wiener–Hopf equations (see Sugiyama (1986)).

4.3 The set of minimal realizations

We characterize the minimal realizations of a finite sequence. This section was partly suggested by Lemma 3 and Theorem 1 of Niederreiter (1988), which use continued fractions in $\mathbb{F}((X^{-1}))$ to study the linear complexity of an infinite sequence and to characterize its minimal polynomials. See also Theorem 3, p124 of Massey (1969) and Proposition 3.4.7 of Ferrand (1988).

We begin with an easy result showing how to generate additional minimal realizations of $(s|L)$ from (μ_L, β_L). Addition and polynomial multiplication of realizations will be by component.

PROPOSITION 4.13 *Let* (μ_L,β_L), $(\mu_\alpha,\beta_\alpha)$ *be computed as in Algorithm MR, where* $\alpha = \alpha(s|L)$*. Then for any* $r \in R \setminus \{0\}$ *and* $c \in R[X]$, $\delta c \leq 2\delta\mu_L - L - 1$, $r(\mu_L,\beta_L) + c(\mu_\alpha,\beta_\alpha)$ *is a minimal realization of* $(s|L)$.

PROOF. Without loss of generality, we can assume that $\mu_\alpha \neq 0$. Let $\mu = r\mu_L + c\mu_\alpha$. Since $\alpha \leq L-1$ and $\delta\mu_L = \alpha + 1 - \delta\mu_\alpha$ by Theorem 3.13 and Proposition 3.18,

$$\delta c + \delta\mu_\alpha \leq (2\delta\mu_L - L - 1) + (\alpha + 1 - \delta\mu_L) = \delta\mu_L - L + \alpha \leq \delta\mu_L - 1$$

and so $\delta\mu = \delta\mu_L$. Also, $\beta(\mu,s) = r\beta_L + c\beta_\alpha$ by Proposition 2.3. A simple calculation now yields

$$\delta(\mu\Gamma(s|L) - \beta(\mu,s)) \leq \max\{-L + \delta\mu,\ \delta c - \alpha + \delta\mu_\alpha\} \leq -L + \delta\mu$$

since $\delta c \leq 2\delta\mu_L - L - 1$. Thus $(\mu, \beta(\mu, s))$ is a minimal realization of $(s|L)$. □

Our proof of the converse requires two lemmas:

LEMMA 4.14 *Let $(s|L)$ be a sequence over R and either (i) $f = 1$ or (ii) $1 \leq i \leq L-1$ and $f \in Ann(s|i)$. Put $\beta = \beta(f, s)$ and $\delta = \delta(f\Gamma(s|L) - \beta)$. Then either (i) $\delta \leq \delta f$ or (ii) $\delta \leq -i + \delta f$. If in addition (i) $L \geq 2$ and $f \notin Ann(s|1)$ or (ii) $i \leq L-2$ and $f \notin Ann(s|i+1)$, then either (i) $\delta = \delta f$ or (ii) $\delta = -i + \delta f$.*

PROOF. The case $f = 1$ is easy to check. Expanding $\Gamma(s|L)$ yields $\delta \leq -i + \delta f$. Writing

$$f\Gamma(s|L) - \beta = f\Gamma(s|i+1) - \beta + f\ (s_{-i-1}x^{-i-1} + \ldots + s_{-L+1}x^{-L+1})$$

shows we must have $\delta \geq -i + \delta f$, for otherwise $f \in \mathrm{Ann}(s|i+1)$. □

LEMMA 4.15 *Let F be a field, let $(s|L)$ be a sequence over F, with $\mu_0 = 1$ and $\mu_i \in Min(s|i)$ for $1 \leq i \leq L$. Then any $f \in F[X]$ with $\delta f \geq \delta\mu_L$ may be written as*

$$f = \sum_{i=0}^{L} c_i\mu_i$$

where $\delta c_i < \delta\mu_{i+1} - \delta\mu_i$ for $0 \leq i \leq L-1$ and $\delta c_L = \delta f - \delta\mu_L$. If in addition, (f, g) realizes $(s|L)$ and $\delta f \leq L$, then $g = \beta(f, s)$ and $(f, g) = \sum_{i=0}^{L} c_i(\mu_i, \beta_i)$.

PROOF. The first statement is an easy induction on $\delta f \geq 0$ and the form of g follows from Proposition 2.3(e) and Proposition 2.6. □

THEOREM 4.16 *Let $(s|L)$ be a sequence over a field F and let (μ_L, β_L), $(\mu_\alpha, \beta_\alpha)$ be computed as in Algorithm MR, where $\alpha = \alpha(s|L)$. Then any minimal realization (f, g) of $(s|L)$ satisfies*

$$(f, g) = c_L(\mu_L, \beta_L) + c_\alpha(\mu_\alpha, \beta_\alpha)$$

for some $c_L \in F \setminus \{0\}$, $c_\alpha \in F[X]$ where either $c_\alpha = 0$ or $\delta c_\alpha \leq 2\delta\mu_L - L - 1$. In particular, if $2\delta\mu_L \leq L$, then $(f, g) = c_L(\mu_L, \beta_L)$ for some $c_L \in F \setminus \{0\}$.

PROOF. We first dispense with the case $\alpha = -1$; here $s_0 = 0$ and $\mu_L = \mu_0 = 1$. If $f \in \mathrm{Min}(s|L)$, $f = c_L$ for some $c_L \in F \setminus \{0\}$ and $f = c_L\mu_L$. Now suppose that $\alpha \geq 0$. By Lemma 4.15 we have $f = \sum_{i=0}^{L} c_i\mu_i$ where $\delta c_i < \delta\mu_{i+1} - \delta\mu_i$ for $0 \leq i \leq L-1$ and $c_L \in F \setminus \{0\}$ since $\delta f = \delta\mu_L \geq 0$. Let $j = \min\{i : 0 \leq i \leq L \ \&\ c_i \neq 0\}$. If $j = L$, we are done.

We now show that if $0 \leq j \leq L-1$ then $j = \alpha$. Note first that by Theorem 3.13 and Proposition 3.18, $\alpha + 1 \leq i \leq L-1$ implies that $\delta\mu_i = \delta\mu_L$. So $c_i = 0$ for $\alpha + 1 \leq i \leq L-1$ and $j \leq \alpha$. Let us suppose that $j < \alpha$.

It follows easily from Proposition 2.3 that

$$f\Gamma(s|L) - \beta(f,s) = \sum_{i=j}^{L} c_i(\mu_i\Gamma(s|L) - \beta(\mu_i,s)),$$

which we use to prove that $\delta c_j - j + \delta\mu_j \leq -L + \delta f$.

Since $c_j \neq 0$, $\delta\mu_j < \delta\mu_{j+1}$ and $\delta\mu_{j+1} = j + 1 - \delta\mu_j$. This implies that $\delta c_j - j + \delta\mu_j \geq 1 - \delta\mu_{j+1}$. Lemma 4.14 yields $\delta(\mu_j\Gamma(s|L) - \beta(\mu_j,s)) = -j + \delta\mu_j$ and so $\delta(c_j(\mu_j\Gamma(s|L) - \beta(\mu_j,s))) \geq 1 - \delta\mu_{j+1}$.

If now $j < i \leq \alpha$ and $c_i \neq 0$, then

$$\delta c_i - i + \delta\mu_i < \delta\mu_{i+1} - i = 1 - \delta\mu_i \leq 1 - \delta\mu_{j+1}.$$

Lemma 4.14 implies that $\delta(\mu_i\Gamma(s|L) - \beta(\mu_i,s)) \leq -i + \delta\mu_i$ and so $\delta(c_i(\mu_i\Gamma(s|L) - \beta(\mu_i,s))) \leq 1 - \delta\mu_{j+1}$. Since $f \in \mathrm{Ann}(s|L)$, we conclude that $\delta c_j - j + \delta\mu_j \leq -L + \delta\mu_L$.

Hence if $j < \alpha$, then

$$1 - \delta\mu_\alpha \leq 1 - \delta\mu_{j+1} = -j + \delta\mu_j \leq \delta c_j - j + \delta\mu_j \leq -L + \delta\mu_L$$

so $1 - \delta\mu_\alpha \leq -L + \delta\mu_L = -L + \alpha + 1 - \delta\mu_\alpha$, which contradicts $\alpha \leq L - 1$. Thus the only possible non–zero c_i is c_α. Also, $\delta c_\alpha - \alpha + \delta\mu_\alpha \leq -L + \delta\mu_L$ implies that $\delta c_\alpha \leq 2\delta\mu_L - L - 1$.

Finally, $\delta f = \delta\mu_L \leq L$, so that Lemma 4.15 and the first part imply that either $g = \beta(c_L,s) = 0 = c_L\beta_L$ (corresponding to the case $\alpha = -1$) or $g = c_L\mu_L + c_\alpha\mu_\alpha$. □

THEOREM 4.17 *Let $(s|L)$ be a sequence over R and let (μ_L,β_L), $(\mu_\alpha,\beta_\alpha)$ be computed as in Algorithm MR, where $\alpha = \alpha(s|L)$. Then (f,g) is a minimal realization of $(s|L)$ iff*

$$r(f,g) = c_L(\mu_L,\beta_L) + c_\alpha(\mu_\alpha,\beta_\alpha)$$

for some r, $c_L \in R \setminus \{0\}$, $c_\alpha \in R[X]$ where either $c_\alpha = 0$ or $\delta c_\alpha \leq 2\delta\mu_L - L - 1$. If in addition, R is factorial and μ_α is primitive, we may take $r = \lambda\mu_L$ and $c_L = \lambda f$.

PROOF. ⇐: By Proposition 4.13, $r(f,g)$ is a minimal realization of $(s|L)$. It is easy to see that since R is a domain, (f,g) is also a minimal realization of $(s|L)$.

⇒: Let R' be the fraction field of R and let $(s'|L)$ be $(s|L)$ considered as a sequence over R'. Now as in Proposition 3.2, (f,g) and (μ_L,β_L) are minimal realizations of $(s'|L)$ and so by the previous theorem,

$$(f,g) = c'_L(\mu_L,\beta_L) + c'_\alpha(\mu_\alpha,\beta_\alpha)$$

for some $c'_L \in R' \setminus \{0\}$, $c'_\alpha \in R'[X]$ where either $c'_\alpha = 0$ or $\delta c'_\alpha \leq 2\delta\mu_L - L - 1$. If we now let $r \in R \setminus \{0\}$ clear denominators in c'_L and c'_α, $c_L = rc'_L$ and $c_\alpha = rc'_\alpha$, then $r(f,g)$ has the stated form.

Finally, if R is factorial and μ_α is primitive, we have $r(f,g) = c_L(\mu_L, \beta_L) + c_\alpha(\mu_\alpha, \beta_\alpha)$ by the first part. Comparing leading terms of f and μ_L gives $r\lambda f = c_L \lambda\mu_L$ and so

$$r\lambda\mu_L \ f = r\lambda f \ \mu_L + \lambda\mu_L \ c_\alpha\mu_\alpha.$$

Since μ_α is primitive, $r|(\lambda\mu_L \ c_\alpha)$, which yields the result. □

REMARK 4.18 *It is easy to check that the previous theorem also yields Corollary 3.27 since* $2\delta\mu_L \leq L$ *implies* $c_\alpha = 0$. *In particular Conjecture 3.22 would imply that for large enough* L, *the sequence of the first* L *prime numbers has a unique (up to sign) minimal realization if* L *is even. On the other hand if* L *is odd, Proposition 4.13 would yield infinitely many minimal realizations* $(\mu_L, \beta_L) + n(\mu_{L-1}, \beta_{L-1})$ *where* (μ_L, β_L), (μ_{L-1}, β_{L-1}) *are computed using Algorithm MR, and* n *is any integer.*

Acknowledgements. The author gratefully acknowledges financial support from the UK Science and Engineering Research Council under grant GR/H15141, and is pleased to thank the anonymous referees for useful comments, suggestions and Niederreiter (1988), which improved this paper. Some of the algorithms of this paper were implemented in MAPLE by A.Au.

Notes added in proof: (i) An analogue of the Berlekamp–Massey algorithm for partial realization over a field appeared in J. Conan (1985): A recursive procedure for the solution of the minimal partial realization problem for scalar rational sequences, *Rev. Roumaine Math. Pures Appl.* **30**, 625–645. (ii) The author regrets this late publication of his research: an earlier version of this work was submitted to another journal in May 1992. It was withdrawn from that journal in September 1993 as no referees reports had yet been received by the editors.

References

Berlekamp, E. (1968). *Algebraic Coding Theory*. Mc–Graw Hill, New York.

Blahut, R. (1983). *Theory and Practice of Error Control Codes*. Addison–Wesley, Reading MA.

Brazil, M. (1993). Growth functions for some one–relator monoids. *Communications in Algebra* **21**, 3135–3146.

Camion, P. (1989). An iterative Euclidean algorithm. *Proceedings AAECC–5 (L.Huguet, A.Poli, eds.), Lecture Notes in Computer Science*, **356**, 88–128. Springer.

Chan, K.Y., Norton, G.H. (1995). A new algebraic algorithm for generating the transfer function of a trellis encoder. *IEEE Transactions on Communications* **43**, 1866–1867.

Chen, W.–K. (1983). *Linear Networks and Systems*. Wadsworth, Belmont California.

Dai, Z.D., Wan, Z.X. (1988). A relationship between the Berlekamp–Massey and the Euclidean algorithms for linear feedback shift register synthesis. *Acta. Math. Sinica (N.S.)* **4**, 55–63.

Dickinson, B.W., Morf, M., Kailath, T. (1974). A minimal realization algorithm for matrix sequences. *IEEE Transactions Automatic Control* **19**, 31–38.

Dornstetter, J.L. (1987). On the equivalence between Berlekamp's and Euclid's algorithms. *IEEE Trans. Information Theory* **33**, 428–431.

Feng, G.L. and Tzeng, K.K. (1991). A generalization of the Berlekamp–Massey algorithm for multisequence shift register sequence synthesis with applications to decoding cyclic codes. *IEEE Trans. Information Theory* **37**, 1274 – 1287.

Ferrand, D. (1988). *Suites Récurrentes*. IRMAR, Université de Rennes.

Fitzpatrick, P. and Norton, G.H. (1991). Linear recurring sequences and the path weight enumerator of a convolutional code. *Electronic Letters* **27:1**, 98–99.

Fitzpatrick, P. and Norton, G.H. (1995). The Berlekamp–Massey algorithm and linear recurring sequences over a factorial domain. *Applicable Algebra in Engineering, Communication and Computing* **6**, 309–323.

Guiver, J.P. (1985). The equation $Ax = b$ over the ring $\mathbb{C}[z, w]$. In *Multidimensional System Theory, (Ed. N.K. Bose)*, 233 – 244. D.Reidel Publishing Co.

Imamura, K. , Yoshida, W. (1987). A simple derivation of the Berlekamp–Massey algorithm and some applications. *IEEE Trans. Information Theory* **33**, 146–150.

Jonckheere, E., Ma, C. (1989). A simple Hankel interpretation of the Berlekamp–Massey algorithm. (1989). *Linear Algebra and its Applications* **125**, 65–76.

Kalman, R.E. (1979). On partial realizations, transfer functions and canonical forms. *Acta Polytech. Scand. Math. Comput. Sci.* **31**, 9–32.

Kalman, R.E., Farb, P.L., and Arbib, M.A. (1969). *Topics in Mathematical Systems Theory*. McGraw–Hill.

Lang, S. (1993) *Algebra (Third Edition, reprinted)*. Addison–Wesley.

Lidl, R., Niederreiter, H. (1983). *Finite Fields, Encyclopedia of Mathematics and its Applications* **20**. Addison–Wesley.

Lotti, G. (1992). Fast solution of linear systems with polynomial coefficients over the ring of integers. *J. Algorithms* **13**, 564–576.

Massey, J.L. (1969). Shift register synthesis and BCH decoding. *IEEE Trans. Information Theory* **15**, 122–127.

McEliece, R. (1977). *The Theory of Information and Coding, Encyclopedia of Mathematics and its Applications* **3**. Addison–Wesley, Reading, Mass.

Mills, W.H. (1975). Continued fractions and linear recurrences. *Mathematics of Computation* **29**, 173–180.

Niederreiter, H. (1988). Sequences with almost perfect linear complexity profile. In *Advances in Cyptology – EUROCRYPT '87. Lecture Notes in Computer Science* **304**, 37 – 51. Springer.

Norton, G.H. (1994). On n–dimensional sequences.I. *J. Symbolic Computation*. To appear.

Norton, G.H. (1995). Some decoding applications of minimal realization. *Proc. V^{th} I.M.A. Conference on Cryptography and Coding, Springer Lecture Notes in Computer Science.* To appear.

Reeds, J.A., Sloane, N.J.A. (1985). Shift–register synthesis (modulo m). *S.I.A.M. J. Computing* **14**, 505–513.

Rouchaleau, Y., Sontag, E.D. (1979). On the existence of minimal partial realizations of linear dynamical systems over Noetherian integral domains. *Journal of Computer and System Sciences* **18**, 65 – 75.

Sain, M.K. (1975). Minimal torsion spaces and the partial input/output problem. *Information and Control* **29**, 103 – 124.

Sakata, S. (1990). Extension of the Berlekamp–Massey algorithm to n–dimensions. *Information and Computation* **84**, 207–239.

Sheppard, N. (1994). Symbolic Computation of Padé approximants. Third Year Project, Dept. of Electrical Engineering, University of Bristol.

Sugiyama, Y. (1986). An algorithm for solving discrete–time Wiener–Hopf equations based on Euclid's algorithm. *IEEE Trans. Information Theory* **32**, 394 – 409.

Trench, W.F. (1964). An algorithm for the inversion of finite Toeplitz matrices. *J. S.I.A.M.* **12**, 515–522.

Welch, L.R. and Scholtz, R.A. (1979). Continued Fractions and Berlekamp's algorithm. *IEEE Trans. Information Theory* **25**, 19–27.

Wiedemann, D.G. (1986). Solving sparse linear equations over finite fields. *IEEE Trans. Information Theory* **32**, 54–62.

Zierler, N. (1968). Linear recurring sequences and error–correcting codes. In (H.B.Mann Ed.) *Error Correcting Codes*, 47–59. J. Wiley (New York).