

# Linear Recurrence Polynomial Rings (Draft)

Joseph M. Shunia

August 2023

## Abstract

This paper introduces the Linear Recurrence Polynomial Ring (LRPR): an algebraic structure that reveals a deep connection between linear recurrence relations and polynomial rings. By defining a ring  $R_f$  with a mapping  $x^d \rightarrow f(x)$ , where  $d$  is the degree of  $f(x)$ , we unveil a powerful framework for working with linear recurrence relations. The structure allows for efficient computation of linear recurrence values, their binomial and multinomial transforms, partial sums, convolutions, and more, all achievable in  $O(\log n)$  time. The methodology extends to handle signed integers and non-uniform sequences, opening new avenues in mathematical and computational exploration. Applications and implications in various mathematical and computational fields are discussed, unveiling a unifying theory with broad potential impact.

## 1 Introduction

### 1.1 Background

Linear recurrence relations are utilized in various fields of mathematics, computer science, physics, and engineering. They describe sequences where each term is defined based on its preceding terms, following a fixed rule. Traditional methods to solve or analyze linear recurrence relations often rely on characteristic polynomials, matrix methods, or generating functions, requiring complex computations and a deep understanding of linear algebra or calculus.

### 1.2 Novel Polynomial Ring Structure

In this paper, we introduce a novel approach to computing the values of linear recurrence relations using a specific polynomial ring structure. Given a linear recurrence relation of degree  $d$ , we construct a polynomial  $f(x)$  corresponding to the coefficients of the relation. Then, we define a unique ring structure where:

$$x^d = f(x)$$

This equation encapsulates the linear recurrence behavior within an algebraic framework, allowing for a more intuitive and unified representation of recurrence relations.

## 2 Definition of Linear Recurrence Polynomial Ring (LRPR)

### 2.1 Linear Recurrence Polynomial Ring (LRPR)

A Linear Recurrence Polynomial Ring (LRPR) over a field  $K$  with respect to a given polynomial  $f(x) \in K[x]$  of degree  $d$  is a ring structure defined as follows:

Given a polynomial  $f(x) = C_d x^d + C_{d-1} x^{d-1} + \dots + C_1 x + C_0$ , we define a ring  $R_f$  where multiplication is carried out under the usual polynomial multiplication rules, except for the relation:

$$x^d \rightarrow f(x) \text{ in } R_f$$

That is, whenever the power  $x^d$  is encountered in a product, it is replaced by the defining polynomial  $f(x)$ .

When we exponentiate  $x$  within the ring  $R_f$ , the coefficients of the resulting polynomials correspond to the values of the recurrence relation according to the defining polynomial and initial starting values. This exponentiation mirrors the iterative process of the recurrence relation, generating the sequence elements directly from the ring structure.

### 2.2 Introduction of Ideal and Custom Mapping

To represent linear recurrence relations, we introduce a specific structure within the polynomial ring. Given a polynomial  $f(x)$  representing the linear recurrence relation, we form the ideal generated by  $f(x)$ :

$$I = \langle x^d - f(x) \rangle$$

where  $d$  is the degree of  $f(x)$ . The Linear Recurrence Polynomial Ring (LRPR) is formed by taking the quotient ring with respect to this ideal, denoted by  $R_f = F[x]/I$ . Here, the subscript  $f$  signifies that the ring is constructed based on the polynomial  $f(x)$ .

### 2.3 Construction of LRPR

Given a linear recurrence relation of degree  $d$  with characteristic coefficients  $C_0, C_1, \dots, C_{d-1}$ , we construct the corresponding polynomial  $f(x)$ :

$$f(x) = C_0 + C_1 x + \dots + C_{d-1} x^{d-1}$$

The Linear Recurrence Polynomial Ring (LRPR) associated with this polynomial is denoted by  $R_f$ .

## 2.4 Defining Relation and Exponentiation in LRPR

Within the ring  $R_f$ , the defining relation is expressed as:  $x^d \rightarrow f(x)$  in  $R_f$ . Here, the arrow  $\rightarrow$  signifies the specific relationship within the LRPR that encapsulates the recurrence behavior.

When we exponentiate  $x$  within the ring  $R_f$ , the coefficients of the resulting polynomials correspond to the values of the recurrence relation according to the defining polynomial and initial starting values. This exponentiation mirrors the iterative process of the recurrence relation, generating the sequence elements directly from the ring structure.

## 2.5 Example: LRPR for the Fibonacci Sequence

For the Fibonacci sequence, with  $f(x) = 1 + x$ , the corresponding LRPR is denoted by  $R_f$ , and the defining relation is:

$$x^2 \rightarrow 1 + x \text{ in } R_f$$

Exponentiation within this ring reflects the Fibonacci sequence according to the given recurrence relation and initial values  $(0, 1)$ . We will demonstrate how to compute  $x^n$  in the ring  $R_f$  for various values of  $n$ .

- $n = 1$  : In this case:

$$x^1 = x$$

- $n = 2$  : Using the relation  $x^2 = x + 1$ , we have:

$$x^2 = 1 + x$$

- $n = 3$  : Expanding further:

$$\begin{aligned} x^3 &= x \cdot (x^2) \\ &= x \cdot (1 + x) \\ &= x + x^2 \\ &= x + (x + 1) \\ &= 2x + 1 \end{aligned}$$

- $n = 4$  : Continuing:

$$\begin{aligned} x^4 &= x \cdot (2x + 1) \\ &= 2x^2 + x \\ &= 2(x + 1) + x \\ &= 3x + 2 \end{aligned}$$

• ...

As we continue to exponentiate, we find that the coefficients correspond to the Fibonacci sequence.

### 3 LRPR Binomial Transforms

#### 3.1 Computing the Binomial Transform

The binomial transform of a sequence can be computed within an LRPR by considering the polynomial  $(1 + P(x))^n$  and utilizing the exponentiation by squaring method, where  $P(x) = P(1) \rightarrow R_f$ .

Given a LRPR  $R_f$  representing a recurrence relation, the key step to compute its binomial transform is to work with the polynomial ring structure and perform the exponentiation  $(1 + P(x))^n$ . The coefficient of 1 plays a crucial role in this computation, as it fundamentally alters the behavior of the exponentiation and captures the essence of the binomial transform.

#### 3.2 Algorithm for Binomial Transform

1. **Define the LRPR:** Represent the recurrence relation using the LRPR  $R_f$ .
2. **Prepare the Polynomial:** Form the polynomial  $(1 + P(x))^n$  within the ring.
3. **Perform Exponentiation:** Use the exponentiation by squaring method to efficiently compute the required power of the polynomial.
4. **Extract the Coefficients:** The coefficients of the resulting polynomial correspond to the binomial transform of the original recurrence relation.

### 4 Multivariate LRPR

Given a set of equations  $\{a^{d_1} = f_1(a, b, \dots), b^{d_2} = f_2(a, b, \dots), \dots\}$ , we can define a multivariate Linear Recurrence Polynomial Ring (LRPR) by the following relations:

$$\begin{aligned} a^{d_1} &\rightarrow f_1(a, b, \dots), \\ b^{d_2} &\rightarrow f_2(a, b, \dots), \\ &\vdots \\ k^{d_k} &\rightarrow f_k(a, b, \dots). \end{aligned}$$

The relations define how the exponents of each variable map to particular expressions, allowing us to model complex systems of relations among variables.

## 5 Proofs

**Theorem 1.** *The LRPR defined by  $x^2 = x + 1$  and the initialization vector  $\mathbf{b} = (F_0, F_1)$  can be used to compute the Fibonacci sequence.*

*Proof.* We'll proceed by induction on  $n$ .

**Base Case ( $\mathbf{n} = 0$  and  $\mathbf{n} = 1$ ):** In the base case, we already have  $F_0 = 0$  and  $F_1 = 1$  in our initialization vector, so our LRPR definition correctly calculates the first two terms of the Fibonacci sequence.

**Inductive Step:** Assume that the LRPR correctly computes the Fibonacci numbers up to  $F_n$  (inductive hypothesis). We want to show that it also correctly computes  $F_{n+1}$ .

Using our LRPR, we have:

$$x^{n+2} = x \cdot x^{n+1} = x \cdot (x^n + x^{n-1}) = x \cdot F_n + x \cdot F_{n-1} = F_{n+1} + F_n = F_{n+1}$$

Here, we used the recurrence relation for the Fibonacci sequence  $F_{n+1} = F_n + F_{n-1}$  and our assumption that the LRPR correctly computes  $F_n$  and  $F_{n-1}$ . Thus, by induction, the LRPR correctly computes the Fibonacci numbers for all non-negative integers  $n$ .  $\square$

**Theorem 2.** *The Linear Recurrence Polynomial Ring (LRPR) defined by  $x^d = a_{d-1}x^{d-1} + a_{d-2}x^{d-2} + \dots + a_1x + a_0$  and an initialization vector  $\mathbf{b} = (r_0, r_1, \dots, r_{d-1})$  can be used to compute the terms of any linear recurrence relation with constant coefficients of order  $d$ , given by:*

$$r_n = a_{d-1}r_{n-1} + a_{d-2}r_{n-2} + \dots + a_1r_{n-(d-1)} + a_0r_{n-d}$$

*Proof.* **Base Case:**

The base case consists of the first  $d$  terms of the recurrence relation, which are given in the initialization vector  $\mathbf{b}$ , so our LRPR definition correctly calculates the initial terms of the recurrence.

**Inductive Step:**

Assume that the LRPR correctly computes the recurrence numbers up to  $r_n$  (inductive hypothesis). We want to show that it also correctly computes  $r_{n+1}$ . Using our LRPR, we have:

$$\begin{aligned} x^{n+d} &= x \cdot x^{n+d-1} \\ &= x \cdot (a_{d-1}x^{n+d-2} + \dots + a_1x^{n+1} + a_0x^n) \\ &= a_{d-1}x^{n+d-1} + \dots + a_1x^{n+2} + a_0x^{n+1} \\ &= r_{n+1} \end{aligned}$$

Here, we used the given recurrence relation and our assumption that the LRPR correctly computes  $r_n, r_{n-1}, \dots, r_{n-(d-1)}$ . Thus, by induction, the LRPR correctly computes the recurrence numbers for all non-negative integers  $n$ .  $\square$

## 6 Work In Progress

This paper is a work in progress. It is an attempt to collate and formally define mathematics for LRPR, which I discovered and have been researching for many years. This paper is merely an introduction to LRPR, which are incredibly versatile and have immense depth; we are merely scratching the surface here. Remaining work for this paper:

1. Describe initialization vectors and how to apply them. We may need new notation for these.
2. Explain how to perform convolutions and partial sums of sequences using LRPR.
3. Explain how to perform multinomial transforms of sequences using LRPR.
4. Explain how to perform sequence to sequence and other transforms using LRPR.
5. Add pseudo code for computational algorithms.
6. Analyze computational complexity
7. Add a table which maps LRPRs to well-known integer sequences on OEIS.
8. Add summary and future work sections.
9. Add citations and references section.
10. Refine presentation.