GENERATING FUNCTIONS AND THEIR APPLICATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BEGÜL BİLGİN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CRYPTOGRAPHY

AUGUST 2010

Approval of the thesis:

GENERATING FUNCTIONS AND THEIR APPLICATIONS

submitted by **BEGÜL BİLGİN** in partial fulfillment of the requirements for the degree of **Master of Science in Department of Cryptography, Middle East Technical University** by,

Prof. Dr. Ersan Akyıldız _____
Director, Graduate School of **Applied Mathematics**

Prof. Dr. Ferruh Özbudak _____
Head of Department, **Cryptography**

Assoc. Prof. Dr. Ali Doğanaksoy _____
Supervisor, **Department of Mathematics**

Examining Committee Members:

Prof. Dr. Ersan Akyıldız _____
Department of Mathematics, METU

Assoc. Prof. Dr. Ali Doğanaksoy _____
Department of Mathematics, METU

Dr. Muhiddin Uğuz _____
Department of Mathematics, METU

Dr. Nurdan Saran _____
Department of Computer Engineering, Çankaya University

Assist. Prof. Dr. Zülfükar Saygı _____
Department of Mathematics, TOBB ETU

Date: _____

*

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: BEGÜL BİLGİN

Signature :

ABSTRACT

GENERATING FUNCTIONS AND THEIR APPLICATIONS

Bilgin, Begül
M.Sc., Department of Cryptography
Supervisor : Assoc. Prof. Dr. Ali Doğanaksoy

August 2010, 54 pages

Generating functions are important tools that are used in many areas of mathematics and especially statistics. Besides analyzing the general structure of sequences and their asymptotic behavior; these functions, which can be roughly thought as the transformation of sequences into functions, are also used efficiently to solve combinatorial problems.

In this thesis, the effects of the transformations of generating functions on their corresponding sequences and the effects of the change in sequences on the generating functions are examined. With these knowledge, the generating functions for the resulting sequence of some combinatorial problems such as number of partitions, number of involutions, Fibonacci numbers and Catalan numbers are found. Moreover, some mathematical identities are proved by using generating functions.

The sequences are the bases of especially symmetric key cryptosystems in cryptography. It is seen that by using generating functions, linear complexities and periods of sequences generated by constant coefficient linear homogeneous recursions, which are used in linear feedback shift register (LFSR) based stream ciphers, can be calculated. Hence studying generating functions leads to have a better understanding in them. Therefore, besides combinatorial

problems, such recursions are also examined and the results are used to observe the linear complexity and the period of LFSR's combined in different ways to generate "better" system of stream cipher.

Keywords: generating functions, linear complexity, period, stream cipher, combinatoric, LFSR

ÖZ

ÜRETEÇ FONKSIYONLAR VE UYGULAMALARI

Bilgin, Begül
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Assoc. Prof. Dr.Ali Doğanaksoy

Ağustos 2010, 54 sayfa

Üreteç fonksiyonlar matematiğin ve bilhassa istatistiğin bir çok sahasında kullanılan önemli araçlardır. Kabaca, dizilerin fonksiyonlar cinsinden ifadesi olarak düşünebileceğimiz bu fonksiyonlar, dizilerin genel yapılarının ve asimtotik davranışlarının incelenmesinin yanısıra kombinatorik problemlerin çözümünde de etkinlikle kullanılan önemli araçlardır.

Bu tezde, üreteç fonksiyonlara uygulanan dönüşümlerin, karşı gelen diziler üstündeki etkileri ve dizilerdeki değişimlerin de üreteç fonksiyonlar üzerindeki etkileri incelenmiştir. Bu bilgiler ışığında tam sayıların parçalanış sayıları, involusyon sayıları, Fibonacci sayıları ve Catalan sayılarının bulunması gibi bazı temel kombinatorik problemlerin sonucu olan dizilerin üreteç fonksiyonları elde edilmiştir. Bunun yanısıra, bazı matematiksel özellikler üreteç fonksiyonlar kullanılarak ispatlanmıştır.

Diziler, kriptografi alanında, özellikle simetrik anahtarlı kriptosistemlerin yapı taşlarıdır. Doğrusal geribeslemeli ötemeli yazdırgaç (DGÖY) tabanlı akan şifrelerin teorik altyapısını teşkil eden sabit katsayılı doğrusal homojen indirgeme bağıntıları ile tanımlanan dizilerin doğrusal karmaşıklığı ve periyodlarının, üreteç fonksiyonlar kullanılarak hesaplanabileceği görülmüştür.

Üreteç fonksiyonları incelemek bu konuların daha iyi anlaşılmasını sağlamaktadır. Bu nedenle, kombinatorik problemlerin yanısıra, bu tip bağıntılar incelenmiş ve elde edilen sonuçlar "daha iyi" akan şifreler tasarlamak amacıyla birden çok DGÖY'nin birleştirilmesiyle oluşan sistemlerin incelenmesinde kullanılmıştır.

Anahtar Kelimeler: üreteç fonksiyonlar, doğrusal karmaşıklık, periyod, akan şifre, kombinatorik, DGÖY

To the two most important men in my life

ACKNOWLEDGMENTS

I owe my deepest gratitude to my supervisor, Assoc. Prof. Dr. Ali Doğanaksoy, for his guidance not only throughout this thesis but also in my whole study in METU. He taught too much about life and mathematics with his jokes and anecdotes.

I am very grateful to Prof. Dr. Turgut Önder who encouraged me to study cryptography.

I would like to thank my family, my uncle and my grandmother for always supporting and trusting in me.

It is a pleasure to thank to my friends in this department who made the study so rich and fun.

Last but not least, very special thanks to Gökhan Öztürk for believing in me and sharing his love with me.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Generating functions are used in a wide area from statistics to mechanics, from combinatorics to cryptology, especially in topics involving sequences. They help us

- to find the explicit formula for the general term of a sequence,
- to find recurrence relations,
- to compute averages and some other statistical properties,
- to find asymptotic formulas,
- to prove some mathematical identities

and more. They can roughly be thought as the transformations of sequences into functions.

To understand generating functions and how they are used better, we consider the problem of determining the number of ways to pay a fixed amount of money, say 6 TL, in terms of 2 and 3 TL coins. This counting problem can be solved easily by hand for small payments and less variety of coins, actually for 6 TL there exists two different ways. However it gets harder to count as the money to be paid and the variety of coins increases. This is why there is a need for a generic method rather then counting by hand.

Let us represent paying a TL as x^a and b TL as x^b. This representation may seem unusual but this is not without purpose. Actually paying $a + b$ TL can be symbolized as $x^{a+b} = x^a x^b$ and na TL as $x^a x^a x^a \dots x^a = (x^a)^n = x^{na}$ which is convenient for our choice of representation. Moreover paying no money can be represented as $(x^a)^0 = 1$ without any confusion. With this knowledge, we represent paying one 3 TL coin as x^3 and one 2 TL coin as x^2 to solve the

problem. Then the representation of paying a total of 6 TL can be seen in different ways like paying three 2 TL coins $\left(x^2\right)^3$ or paying two 3 TL coins $\left(x^3\right)^2$ both of which are equal to x^6. Recall that we wish to find a general solution, so let us consider all possible payments that can be done with only 2 TL coins, namely 0, 2, 4, 6, 8, 10, ... TL. This can be represented as

$$\left(x^2\right)^0 + \left(x^2\right)^1 + \left(x^2\right)^2 + \left(x^2\right)^3 + \cdots = 1 + x^2 + x^4 + x^6 + x^8 + \cdots = \sum_{n=0}^{\infty} x^{2n} \,. \tag{1.1}$$

In a similar way representation of all possible payments with only 3 TL coins is

$$\left(x^3\right)^0 + \left(x^3\right)^1 + \left(x^3\right)^2 + \left(x^3\right)^3 + \cdots = 1 + x^3 + x^6 + x^9 + x^{12} + \cdots = \sum_{n=0}^{\infty} x^{3n} \,. \tag{1.2}$$

When we multiply these two representations, the resulting function

$$1 + x^2 + x^3 + x^4 + x^5 + 2x^6 + x^7 + 2x^8 + 2x^9 + \cdots \tag{1.3}$$

indicates all possible ways of payments using 2 TL and 3 TL coins. Moreover, by looking at the coefficient of x^6 in this representation we again see that paying 6 TL can be done in two different ways. We also observe that the element x^1 does not exist in the product; as expected, since 1 TL can not be paid using 2 or 3 TL coins. To generalize, the coefficient of x^n say a_n represents the number of ways of all possible payments of n TL. The corresponding representation of the sequence a_n which is $\sum_{n=0}^{\infty} a_n x^n$ is called the generating function of the sequence.

For the sake of simplicity, the above functions are sometimes thought as power series and displayed in their corresponding closed forms. The variable x is thought to be in the radius of converge of the power series which allows to perform operations in their closed form. So one can write the generating function (1.1) as $\dfrac{1}{1-x^2}$ and the function (1.2) as $\dfrac{1}{1-x^3}$ so the product (1.3) of these functions is $\dfrac{1}{(1-x^2)(1-x^3)} = \dfrac{1}{1-x^2-x^3+x^6}$.

CHAPTER 2

OPERATIONS

Definition 2.0.1 *The ordinary generating function of a sequence* $\{a_n\} = \{a_0, a_1, a_2, \ldots\}$ *is the formal power series*

$$A(x) = a_0 + a_1 x + a_2 x^2 + \cdots = \sum a_i x^i \, .$$

The above definition is not the only way to form a generating function. Actually another formation will be studied in section 3.5, however the ordinary generating functions are the most commonly used type. Hence, from this point forward they will be mentioned simply as generating functions.

Note that, whether the sequence is finite or infinite, the formal power series in Definition 2.0.1 can be formed.

Example 2.0.1 *The finite sequence* $S_n = \left\{\binom{n}{0}, \binom{n}{1}, \binom{n}{2}, \cdots, \binom{n}{n}\right\}$ *has the generating function*

$$g(x) = \binom{n}{0} + \binom{n}{1}x + \binom{n}{2}x^2 + \cdots + \binom{n}{n}x^n = (1+x)^n \, .$$

It is seen from the closed form of the sequence that the problem of finding the generating function of S_n *can be thought as deciding whether to take an element or not which refers to* x^1 *or* x^0 *respectively from a set of n elements.*

Example 2.0.2 *The infinite sequence* $\{1\}_{n=0}^{\infty} = \{1, 1, 1, 1, \ldots\}$ *has the generating function*

$$g\,(x) = 1 + x + x^2 + x^3 + \cdots = \frac{1}{1-x} \, .$$

It is not straight forward to obtain the closed form of the generating function of a sequence immediately, however it may be useful in some applications. To find the closed form of the generating function of a sequence, one may manipulate some sequences, closed forms of whose generating functions are already known, using certain operations until he reaches the closed form of the desired function. In a similar way, one can also manipulate the generating functions of sequences, observing how these operations affect sequences until he reaches the sequence he wanted. We now examine some common operations.

2.1 Operations on the Sequences

2.1.1 Scalar Multiplication

In this first operation, scalar multiplication, every element of the sequence is multiplied by a scalar. Assume that $\{a_n\}$ with generating function $A(x) = \sum_{i=0}^{\infty} a_i x^i$ is taken and every term is multiplied with λ. Then the generating function of $\{\lambda a_n\}$ becomes

$$\begin{aligned} \sum_{i=0}^{\infty} \lambda a_i x^i &= \lambda \sum_{i=0}^{\infty} a_i x^i \\ &= \lambda A(x) \ . \end{aligned}$$

2.1.2 Shifting and Truncation

Shifting (right) a sequence by t terms means to pad the sequence with t 0's from left, without loosing any terms. To be more clear, after such a shifting, the sequence $\{a_0, a_1, a_2, \ldots\}$ takes the form $\{\underbrace{0, 0, 0, \ldots 0}_{t-terms}, a_0, a_1, a_2, \ldots\}$ whose generating function is

$$\begin{aligned} \widehat{A(x)} &= 0 + 0x + 0x^2 + \cdots + 0x^{t-1} + a_0 x^t + a_1 x^{t+1} + \cdots \\ &= x^t \left(a_0 + a_1 x + a_2 x^2 + \cdots \right) \\ &= x^t A(x) \ . \end{aligned}$$

However, in truncation (shifting left), the first t terms are lost. To be more explicit, the se-

quence $\{a_0,\ a_1,\ a_2,\ \ldots\}$ becomes $\{a_t, a_{t+1}, a_{t+2}, \ldots\}$ with the generating function

$$\begin{aligned}\widetilde{A(x)} &= a_t + a_{t+1}x + a_{t+2}x^2 + \cdots \\ &= \frac{A(x) - \left(a_0 + a_1 x + \cdots + a_{t-1}x^{t-1}\right)}{x^t}.\end{aligned}$$

2.2 Operations on the Functions

2.2.1 Changing the Variable

Obviously, there are countless ways of changing the variable of a generating function. We find two specific ways particularly important and mention them here.

One of the methods is to multiply the original variable x with a constant, say λ and to get another generating function with the variable λx which corresponds to another sequence.

Example 2.2.1 *Take the generating function $A(x)$ which represents $\{a_n\}$ and the constant $\lambda = -1$. After changing the variable by constant multiplication, the new generating function turns out to be*

$$A(-x) = a_0 - a_1 x + a_2 x^2 - a_3 x^3 + \cdots$$

which corresponds to the sequence $\{(-1)^n a_n\}$.

To generalize, the function $A(x)$ becomes $A(\lambda x) = a_0 + a_1\lambda x + a_2\lambda^2 x^2 + \cdots$ with the sequence $\left\{a_0,\ a_1\lambda,\ a_2\lambda^2,\ \ldots\right\} = \{\lambda^n a_n\}$ after changing the variable by constant multiplication.

The second change is done by taking a constant power of the variable. Again taking λ as the exponent, the function is obtained with the change of variable $x \mapsto x^\lambda$.

Example 2.2.2 *Take the constant $\lambda = 2$ for a simple example and the function $A(x)$. When the constant power of the variable is taken, the generating function will yield to*

$$A\left(x^2\right) = a_0 + a_1 x^2 + a_2 x^4 + a_3 x^6 + \cdots .$$

The new function belongs to the sequence $\{a_0, 0, a_1, 0, a_2, 0, a_3, 0, \ldots\}$ since odd degreed terms are missing.

In a similar way, when $\lambda = 3$, the function becomes

$$A\left(x^3\right) = a_0 + a_1x^3 + a_2x^6 + a_3x^9 + \cdots$$

which is the generating function of the sequence $\{a_0,\ 0,\ 0,\ a_1,\ 0,\ 0,\ a_2,\ 0,\ 0,\ a_3,\ 0,\ 0,\ \ldots\}$.

It is seen that when the λ^{th} $(\lambda \in \mathbb{Z}^+)$ power of the variable is taken, $\lambda - 1$ zeros arises in between each pair of terms of the sequence.

2.2.2 Differentiation and Integration

As mentioned in section (1), one can take the corresponding power series representation of a generating function assuming the variable in the radius of convergence. This means that one can apply differentiation and integration operations over generating functions without any problem as defined below.

$$\begin{aligned} D_x\left(A\left(x\right)\right) &= D_x\left(a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots\right) \\ &= a_1 + 2a_2x + 3a_3x^2 + 4a_4x^3 + \cdots . \end{aligned} \tag{2.1}$$

$$\begin{aligned} \int_0^x A\left(t\right)dt &= \int_0^x \left(a_0 + a_1t + a_2t^2 + a_3t^3 + \cdots\right)dt \\ &= a_0x + \frac{a_1}{2}x^2 + \frac{a_2}{3}x^3 + \frac{a_3}{4}x^4 + \cdots . \end{aligned} \tag{2.2}$$

The meaning of these operations may be understood better with the following example.

Example 2.2.3 *Consider the generating function* $g\left(x\right) = \dfrac{1}{1-x}$ *which represents the sequence* $\{1,\ 1,\ 1,\ \ldots\}$. *Then*

$$D_x\left(g\left(x\right)\right) = 1 + 2x + 3x^2 + 4x^3 + 5x^4 + \cdots = \frac{1}{\left(1-x\right)^2}$$

and

$$\int_0^x g\left(t\right)dt = x + \frac{1}{2}x^2 + \frac{1}{3}x^3 + \frac{1}{4}x^5 + \cdots = -ln\left(1-x\right)$$

correspond to the sequences $\{1,\ 2,\ 3,\ \ldots\}$ *and* $\{1,\ \frac{1}{2},\ \frac{1}{3},\ \ldots\}$ *respectively.*

2.2.3 Summation

In some operations more than one function is used unlike what is done so far in this section. One such operation is summation. Take $B(x)$ and $\{b_n\} = \{b_0,\ b_1,\ b_2,\ \ldots\}$ in addition to $A(x)$ and $\{a_n\}$. As $A(x)$ and $B(x)$ are added,

$$\sum_{i=0}^{\infty} a_i x^i + \sum_{i=0}^{\infty} b_i x^i = \sum_{i=0}^{\infty} (a_i + b_i) x^i$$

which is the generating function of $\{(a+b)_n\}$, is taken as a result. So it is seen that the i^{th} term of the sequence, that corresponds to the sum of two functions, is the sum of the i^{th} terms of those sequences.

2.2.4 Convolution

Although multiplying two functions is as natural as adding them, as it is seen below, general term of the resulting sequence of multiplication is not the product of i^{th} terms of the sequences. In fact

$$\begin{aligned} A(x)B(x) &= a_0\left(b_0 + b_1x + b_2x^2 + \cdots\right) + a_1\left(b_0 + b_1x + b_2x^2 + \cdots\right)x \\ &\quad + a_2\left(b_0 + b_1x + b_2x^2 + \cdots\right)x^2 + \cdots \\ &= a_0b_0 + (a_0b_1 + a_1b_0)\,x + (a_0b_2 + a_1b_1 + a_2b_0)\,x^2 + \cdots \\ &= \sum_{k=0}^{\infty}\left(\sum_{i+j=k} a_ib_j\right)x^k\,. \end{aligned}$$

This is why the resulting sequence $\{a_0b_0,\ a_0b_2 + a_1b_0,\ a_0b_2 + a_1b_1 + a_2b_0,\ \ldots\}$ has the special name *convolution* of $\{a_n\}$ and $\{b_n\}$.

Example 2.2.4 *Take $A(x)$ which may correspond to any sequence $\{a_n\}$ and $B(x) = \dfrac{1}{1-x}$ which belongs to the constant sequence $\{1\}$.*

$$\begin{aligned} A(x)\,\frac{1}{1-x} &= a_0 + (a_0 + a_1)\,x + (a_0 + a_1 + a_2)\,x^2 + \cdots \\ &= \sum_{k=0}^{\infty}\left(\sum_{i=0}^{k} a_i\right)x^k \end{aligned}$$

As it is seen the resulting function above is actually the generating function of the sequence of partial sums. So if $A(x)$ is also taken to be $\frac{1}{1-x}$, the multiplication becomes $\frac{1}{(1-x)^2} = 1 + 2x + 3x^2 + \cdots$ which matches with what was found in Example 2.2.3.

2.2.5 Composition

Take $A(x)$ and $B(x)$ once more to form the composed function

$$\begin{aligned} A\left(B\left(x\right)\right) &= a_0 + a_1\left(b_0 + b_1x + b_2x^2 + b_3x^3 + \cdots\right) + a_2\left(b_0 + b_1x + b_2x^2 + \cdots\right)^2 + \cdots \\ &= \left(a_0 + a_1b_0 + a_2b_0^2 + a_3b_0^3 + \cdots\right) + \cdots . \end{aligned}$$

It is seen that the coefficients have infinite sums unless $b_0 = 0$. If $B(0) = 0$, i.e., $b_0 = 0$,

$$A\left(B\left(x\right)\right) = a_0 + a_1b_1x + \left(a_1b_2 + a_2b_1^2\right)x^2 + \left(a_1b_3 + 2a_2b_1b_2 + a_3b_1^3\right)x^3 + \cdots$$

is found with a finite sum in every coefficient. To sum up, if a function without the 0^{th} coefficient is taken as $B(x)$, the composition can be done without any problem. Some examples to this operation will be studied in the following chapters.

2.3 Blending

This operation is a combination of some operations that is done so far. Assume that the generating function $g\left(x\right)$ of the sequence $S = \{a_0, b_0, a_1, b_1, a_2, b_2, \ldots\}$ is needed and the generating functions of the sequences $\{a_n\}$ and $\{b_n\}$ as $A\left(x\right)$ and $B\left(x\right)$ are already known. With some simple operations one can see

$$\begin{aligned} g\left(x\right) &= a_0 + b_0x + a_1x^2 + b_1x^3 + a_2x^4 + b_2x^5 + \cdots \\ &= \left(a_0 + a_1x^2 + a_2x^2 + \cdots\right) + x\left(b_0 + b_1x^2 + \cdots\right) \\ &= A\left(x^2\right) + xB\left(x^2\right) . \end{aligned}$$

It can be easily seen that actually the sequences $\{a_0, 0, a_1, 0, a_2, 0, \ldots\}$ and $\{0, b_0, 0, b_1, \ldots\}$, which is the one shifted right version of $\{b_0, 0, b_10, b_2, \ldots\}$, were added. To extend a little bit, if both of the sequences were taken as $\{a_n\}$ the blending operation would give $\{a_0, a_0, a_1, \ldots\}$ with the corresponding generating function $(1 + x)A(x^2)$. Now it is obvious that a 3-blending $\{a_0, b_0, c_0, a_1, b_1, c_1, \ldots\}$ with $\{c_n\}$ and its generating function $C(x)$ besides $\{a_n\}$ and $\{b_n\}$, can be seen as $A(x^3) + xB(x^3) + x^2C(x^3)$. With this fashion one can blend two or more sequences however he wants.

2.4 Decimation

This is the last operation that will be mentioned in this thesis. A decimation, sometimes referred to as a d-decimation is taking every d^{th} term of a sequence. Assume that the closed form of the generating function of the 2-decimation of the sequence $\{a_n\}$, in other words the sequence $S = \{a_0, a_2, a_{4,}a_6, \ldots\}$, is wanted. It is already known from the Example 2.2.1 that $A(-x)$ belongs to the sequence $\{a_0, -a_1, a_2, -a_3, \ldots\}$, then

$$\begin{aligned}\widetilde{g(x)} &= A(x) + A(-x) \\ &= (a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots) + (a_0 - a_1x + a_2x^2 - a_3x^3 + \cdots) \\ &= 2\left(a_0 + a_2x^2 + a_4x^4 + \cdots\right)\end{aligned}$$

corresponds the sequence $S' = \{2a_0, 0, 2a_2, 0, 2a_4, 0, \ldots\}$. As it is seen if every x^2 in $\widetilde{g(x)}$ is replaced with x, namely the variable is changed from x to $\sqrt{x}$, the zeros in S' will be discarded. So

$$g(x) = \frac{A\left(\sqrt{x}\right) + A\left(\sqrt{-x}\right)}{2}$$

is the generating function of S .

What about 3-decimation? Let 1, w and w^2 be the 3^{rd} roots of unity. The sum of the functions

$$A(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + \cdots$$

$$A(wx) = a_0 + a_1wx + a_2w^2x^2 + a_3w^3x^3 + \cdots$$

$$A\left(w^2x\right) = a_0 + a_1w^2x + a_2w^4x^2 + a_3w^6x^3 + \cdots$$

is

$$\begin{aligned}\widetilde{g(x)}(x) &= A(x) + A(wx) + A\left(w^2x\right) \\ &= 3a_0 + a_1\left(1 + w + w^2\right)x + a_2\left(1 + wa^2 + w^4\right)x^2 + a_3\left(1 + w^3 + w^6\right)x^3 + \cdots .\end{aligned}$$

Since 1, w and w^2 are the 3^{rd} roots of unity, $w^3 = 1$, $1 + w + w^2 = 0$, but $1 + w^3 + w^{2^3} = 3$. This is why

$$\widetilde{g(x)} = 3\left(a_0 + a_3x^3 + a_6x^6 + \cdots\right).$$

Hence the function for the 3-decimated sequence becomes

$$g(x) = \frac{1}{3}\left(A\left(\sqrt[3]{x}\right) + A\left(\sqrt[3]{wx}\right) + A\left(\sqrt[3]{w^2x}\right)\right) .$$

To generalize, to find the d-decimation of a sequence starting with the 0^{th} term, one needs to use the d^{th} roots of unity $e^{\frac{2n\pi i}{d}}$, $n = 0, 1, \ldots, d-1$; $d \in \mathbb{Z}$ and to generate the functions

$$f\left(e^{\frac{2n\pi i}{d}} x^{\frac{1}{d}}\right) = \sum_{k=0}^{\infty} a_k e^{(\frac{2n\pi i}{d})k} x^{\frac{k}{d}}, \forall n = 0, 1 \ldots, d-1$$

then to sum them up to get

$$\sum_{n=0}^{d-1} f\left(e^{\frac{2n\pi i}{d}} x^{\frac{1}{d}}\right) = \sum_{k=0}^{\infty} a_k \left(\sum_{n=0}^{d-1} \left(e^{\frac{2n\pi i}{d}}\right)^k x^{\frac{k}{d}}\right).$$

Since

$$\begin{aligned} \sum_{n=0}^{d-1} \left(e^{\frac{2n\pi i}{d}}\right)^k &= \begin{cases} d & d|k \\ 0 & d \nmid k \end{cases}, \\ \sum_{n=0}^{d-1} f\left(e^{\frac{2n\pi i}{d}} x^{\frac{1}{d}}\right) &= da_0 + da_d x + da_{2d} x^2 + \cdots . \end{aligned}$$

Hence the generating function for the d-decimation becomes

$$\frac{1}{d} \sum_{n=0}^{d-1} f\left(e^{\frac{2n\pi i}{d}} x^{\frac{1}{d}}\right).$$

Example 2.4.1 *Consider the sequence* $S = \{a_0, a_1, 0, a_2, a_3, 0, \ldots\}$ *where a zero is inserted following every two consecutive terms. If the sequence* $\{a_n\}$ *has the generating function* $A(x)$*, then the 2-decimated sequences* $\{a_0, a_2, a_4, \ldots\}$ *and* $\{a_1, a_3, a_5, \ldots\}$ *have the generating functions* $B(x) = \frac{A(\sqrt{x})+A(\sqrt{-x})}{2}$ *and* $C(x) = \frac{A(\sqrt{x})-A(\sqrt{-x})}{2}$ *as mentioned before. It is seen that if the sequences* $\{a_0, 0, 0, a_2, 0, 0, a_4, 0, 0, \ldots\}$ *and* $\{0, a_1, 0, 0, a_3, 0, 0, a_5, 0, 0, \ldots\}$ *are added the desired sequence* S *is obtained. Moreover it is known that changing the variable is enough to put zeros between every term. Therefore the generating function* $B(x^3) + xC(x^3)$ *represents the sequence S.*

Example 2.4.2 *As opposed to what is done in the previous example, one could also need the generating function for the sequence* $S = \{a_0, a_1, a_3, a_4, a_6, a_7, \ldots\}$ *in which every third term is dismissed from the sequence* $\{a_n\}$ *with generating function* $A(x)$*. It is seen easily that this time the desired sequence is simply the sum of* $\{a_0, 0, a_3, 0, a_6, 0, \ldots\}$ *and* $\{0, a_1, 0, a_4, 0, a_7, \ldots\}$*. As it is seen they are the 3-decimated versions of the sequences* $\{a_n\}$ *and* $\{a_{n+1}\}$ *with 0's between terms. Therefore, if* $\{a_0, a_3, a_6, \ldots\}$ *and* $\{a_1, a_4, a_7, \ldots\}$ *have generating sequences* $B(x)$ *and* $C(x)$ *respectively than the desired function for* S *is* $B(x^2) + xC(x^2)$*.*

Example 2.4.3 *The final problem of this chapter is finding the generating function* $g(x)$ *of the sequence* $S = \{a_0,\ b_1,\ a_2,\ b_3,\ a_4,\ b_5\}$ *assuming that* $A(x)$ *and* $B(x)$ *are known. First the two decimated versions of the sequences needs to be taken and between each term zeros must be inserted to get* $\{a_0,\ 0,\ a_2,\ 0,\ a_4,\ 0,\ \ldots\}$ *and* $\{0, b_1, 0, b_3, 0, b_5, 0, \ldots\}$ *then S will simply be the sum of these sequences. So*

$$g(x) = \frac{A(x) + A(-x)}{2} + \frac{B(x) - B(-x)}{2}.$$

CHAPTER 3

Combinatorial Problems

Combinatorics is one of the topics that uses generating functions to solve the problems occasionally. In this chapter we will give some examples of the usage of generating functions in combinatorics besides the problem we have solved in Chapter 1.

3.1 Distributing n Candies to r Children

In this section, it is always assumed that the candies are indistinguishable. Before studying the question on distributing candies under a specific rule, we first consider the problem of finding the number of distributing n candies to r children without any restriction using generating functions. That means that every child can have any number of candies, of course up to n since the number of candies are finite, including none, in which case the generating function for a child becomes

$$1 + x + x^2 + x^3 + \cdots + x^n = \frac{1 - x^{n+1}}{1 - x} .$$

The order of the candies a child gets does not matter since the candies are indistinguishable. That is why distributing t candies can be done in only 1 way for a child, therefore the coefficients are all 1. Moreover since there are r children, the r^{th} power of this generating function should be taken to get

$$\left(\frac{1 - x^{n+1}}{1 - x}\right)^r .$$

At this point, looking at the coefficient of x^n in the above function is enough to solve the question, however, one can make a trick and solve the same question in another way and get the coefficient exactly. Observe that even if the generating function for a child is taken as $\dfrac{1}{1 - x}$, without any upper restriction, the overall generating function still gives the same

result since only the coefficient of x^n is important. In that case the overall generating function becomes $\frac{1}{(1-x)^r} = \sum_{k=0}^{\infty} \binom{r+k-1}{k} x^k$ with $\binom{r+n-1}{n}$ as the n^{th} coefficient.

As another version of this problem assume that every child gets at least t candies. In that case the generating function for distribution of candies for one child becomes

$$x^t + x^{t+1} + \cdots = \frac{x^t}{1-x}$$

which gives

$$\left(\frac{x^t}{1-x}\right)^r = x^{tr} \frac{1}{(1-x)^r} = x^{tr} \sum_{k=0}^{\infty} \binom{r+k-1}{k} x^k$$

for r children. Again one should look at the coefficient of x^n to solve the question. Hence, it is necessary to find the coefficient of x^{n-tr} in $\sum_{k=0}^{\infty} \binom{r+k-1}{k} x^k$ which is $\binom{r+n-tr-1}{n-tr}$. This result also matches with giving t candies to every children first then distributing the remaining $n-tr$ candies.

One other thing that can be done is to give at most t candies to each of the children, not to make them sick. Again there are n candies to distribute and this time $n < tr$ is taken not to face with any problem. Then the distribution function for a child becomes

$$1 + x + x^2 + \cdots + x^t = \frac{1-x^{t+1}}{1-x} .$$

This is why the answer to the question is

$$\begin{aligned} \frac{\left(1-x^{t+1}\right)^r}{(1-x)^r} &= \left(1-x^{t+1}\right)^r \sum_{k=0}^{\infty} \binom{r+k-1}{k} x^k \\ &= \sum_{i=0}^{r} (-1)^i \binom{r}{i} x^{i(t+1)} \sum_{k=0}^{\infty} \binom{r+k-1}{k} x^k \\ &= \sum_{i=0}^{r} \sum_{k=0}^{\infty} (-1)^i \binom{r}{i} \binom{r+k-1}{k} x^{k+i(t+1)} \end{aligned}$$

with $\sum_{i=0}^{r} (-1)^i \binom{r}{i} \binom{r+n-i(t+1)-1}{r-1}$ being the coefficient of x^n .

Here is one last illustrating example to combine everything mentioned so far in this section. In this one there are 4 children from whom the first child gets at least 3 candies, the second one gets at most 5 candies, the third one gets even number of candies and the last one takes either 4 or 5 candies. The question is to find the number ways to distribute 17 candies for those children. To solve the question one should first look at the generating functions of taking candies for every child .

1^{st} child : $x^3 + x^4 + \cdots = \frac{x^3}{1-x}$

2^{nd} child : $1 + x + \cdots + x^5 = \frac{1-x^6}{1-x}$

3^{rd} child : $1 + x^2 + x^4 + \cdots = \frac{1}{1-x^2}$

4^{th} child : $x^4 + x^5 = x^4(1 + x)$

Then the generating function for the whole distribution becomes

$$\frac{x^3}{1-x}\frac{1-x^6}{1-x}\frac{1}{1-x^2}x^4(1+x) = \frac{x^7\left(1-x^6\right)}{(1-x)^3}.$$

Recall from Example 2.2.4 that multiplying a generating function with $\frac{1}{1-x}$ means taking partial sums of the sequence and the generating function $\frac{1}{(1-x)^2}$ represents the sequence $\{n+1\}_{n=0}^{\infty}$. Therefore the generating function for the distribution can be viewed as

$$\begin{aligned}
\frac{x^7\left(1-x^6\right)}{(1-x)^3} &= x^7\frac{1}{(1-x)^2}\frac{1}{1-x} - x^{13}\frac{1}{(1-x)^2}\frac{1}{1-x} \\
&= \sum_{k=0}^{\infty}\sum_{i=0}^{k}(i+1)x^{k+7} - \sum_{k=0}^{\infty}\sum_{i=0}^{k}(i+1)x^{k+13} \qquad (3.1)\\
&= \sum_{k=0}^{\infty}\frac{(k+1)(k+2)}{2}x^{k+7} - \sum_{k=0}^{\infty}\frac{(k+1)(k+2)}{2}x^{k+13} \qquad (3.2)\\
&= \sum_{k=0}^{\infty}\binom{k+2}{2}x^{k+7} - \sum_{k=0}^{\infty}\binom{k+2}{2}x^{k+13} \qquad (3.3)
\end{aligned}$$

in which the coefficient of x^{17} which is equal to $\binom{12}{2} - \binom{6}{2}$ is the answer.

3.2 Fibonacci's Rabbits

In the year 1202, Fibonacci (Leonardo of Pisa) investigated the breeding of rabbits in ideal circumstances. If a suitable environment is created, a pair of rabbit can give birth to a new pair every month and a new born grows enough to be bred after one month. The Fibonacci question is to find the number of pairs after n months starting with only one new born pair and assuming that every mother give birth to exactly two children, a male and a female and none of the rabbits die. Since the rabbits in hand at the beginning are newly born they can not breed in the first month however, they can mate at the end of that month so that in the second

month the female gives birth to a new pair. In the third month, the original female gives birth to a new pair again but her children can not however they mate at the and of that month. Observing more deeply, it is seen that at the end of every month, the rabbits in hand 2 months ago give birth to a new pair and the ones in the previous months stays in the population. If the number of rabbit pairs in the n^{th} month is f_n, referring to Fibonacci number,

$$f_n = f_{n-1} + f_{n-2} \, .$$

Think every Fibonacci number as an element of the Fibonacci sequence $\{f_n\} = \{f_0, f_1, \ldots\}$ with generating function $f(x)$ and initial terms $f_0 = f_1 = 1$. Then the sum of $\{f_{n-1}\} = \{0, f_0, f_1, f_2, \ldots\}$ and $\{f_{n-2}\} = \{0, 0, f_0, f_1, f_2, \ldots\}$ with generating functions $xf(x)$ and $x^2 f(x)$ respectively gives $\{f_n\}$ except for the first term. So

$$\begin{aligned} f(x) &= xf(x) + x^2 f(x) + 1 \\ \left(1 - x - x^2\right) f(x) &= 1 \\ f(x) &= \frac{1}{1 - x - x^2} \, . \end{aligned}$$

At this moment, to find the coefficient of x^n it is enough to see the generating function as

$$\begin{aligned} f(x) &= \frac{1}{(1 - \alpha_1 x)(1 - \alpha_2 x)} \\ &= \frac{A}{1 - \alpha_1 x} + \frac{B}{1 - \alpha_2 x} \\ &= \frac{1}{\sqrt{5}} \left(\frac{1}{1 - \alpha_1 x} - \frac{1}{1 - \alpha_2 x} \right) \\ &= \frac{1}{\sqrt{5}} \left(\sum_{i=0}^{\infty} \alpha_1^i x^i - \sum_{i=0}^{\infty} \alpha_2^i x^i \right) \end{aligned}$$

where $\alpha_{1,2} = \frac{1 \pm \sqrt{5}}{2}$. Therefore the coefficient of x^n is $f_n = \frac{1}{\sqrt{5}} \left((\alpha_1)^n - (\alpha_2)^n \right)$.

Looking at the sequence $\{1, 1, 2, 3, 5, 8, 13, 21, 34, 55, \ldots\}$ this function represents, it is seen that every term is actually sum of the previous and the second previous element.

3.3 Regions in the Plane

It is countable with ease that one, two or three lines on a plane divide the plane into two, four or seven regions respectively. What if the number of lines were somewhat higher? In this section, the generating function for the number of regions p_n on a plane that n different lines

create will be found. Of course, it is assumed that lines are in general position [1]. To start with consider a line l_1. When another line l_2 is drawn, it intersects l_1 and the intersection point separates it into two pieces where each piece divides one of the previous regions into two. If this is continued with another line l_3, the two intersection points of l_3 with l_1 and l_2 divide it into three pieces where every piece again divides one of the previous regions into two. As this goes on it is seen that l_n is divided into n pieces by the intersection points with the existing lines where each piece divides a region into two. Therefore the below recursion is taken.

$$p_n = p_{n-1} + n$$

If $p(x) = \sum_{i=0}^{\infty} p_i x^i$, then

$$p(x) = xp(x) + \frac{x}{(1-x)^2} + 1$$

since function for $\{n_n\}$ is $\frac{x}{(1-x)^2}$ as shown in Example 2.2.3. This implies that

$$\begin{aligned} p(x)(1-x) &= \frac{x}{(1-x)^2} + 1 \\ p(x) &= \frac{x}{(1-x)^3} + \frac{1}{1-x}\,. \end{aligned}$$

Hence with an operation similar to what was done in equation 3.1, it is seen that $p_n = \binom{n+1}{2} + 1$.

3.4 Partitions

A partition of a positive integer n is a set of positive integers which sum to n. For example for the integer 4, the partitions are $\{1,1,1,1\}, \{1,1,2\}, \{2,2\}, \{1,3\}, \{4\}$. If the number of partitions for an integer n is denoted as p_n then $p_4 = 5$. In this section, the question is to find the number of all partitions of a number n without counting every possible sets, namely using generating functions. For every element k in the partition, the corresponding factor in the generating function is

$$1 + x^k + x^{2k} + x^{3k} + \cdots = \frac{1}{1-x^k}\,.$$

Moreover, in a partition, it is obvious that the greatest number that can be used is n. Hence the generating function for p_n becomes

$$p(x) = \frac{1}{(1-x)(1-x^2)(1-x^3)\cdots(1-x^n)} = \prod_{k=1}^{n} \frac{1}{1-x^k}. \tag{3.4}$$

[1] No two lines are parallel or coincident

Unlike the previous problems, in this problem the exact value of p_n can not be extracted by looking at the generating function of the sequence $\{p_n\}$. However as mentioned in Chapter 1, the generating functions can be used to prove mathematical identities. Namely, the following lemma can be proved by using the above generating function.

Lemma 3.4.1 *The number of partitions with unequal elements in the set is equal to the number of partitions where each element in the set is an odd integer.*

Proof. Let u_n be the number of partitions of n with unequal elements. That means every integer can be seen at most once in the set. Therefore, the representation of seeing an integer k is $1 + x^k$ which makes the generating function for unequal partitions

$$\begin{aligned} u(x) &= (1+x)\left(1+x^2\right)\left(1+x^3\right)\cdots(1+x^n) \\ &= \prod_{k=1}^{n} 1+x^k \,. \end{aligned} \tag{3.5}$$

In a similar fashion, let o_n be the number of partitions with only odd numbers in the set. Unlike to the equation 3.4, this time only the integers $1, 3, 5, \ldots$ can be used which makes the generating function for odd partitions

$$\begin{aligned} o(x) &= \frac{1}{(1-x)(1-x^3)(1-x^5)\cdots} \\ &= \prod_{k=1}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{1-x^k} \,. \end{aligned} \tag{3.6}$$

So if the equivalence of the equations (3.5) and (3.6) are shown, the proof will be done. It is easily seen that

$$\begin{aligned} u(x) &= (1+x)\left(1+x^2\right)\left(1+x^3\right)\cdots(1+x^n) \\ &= \frac{1-x^2}{1-x}\frac{1-x^4}{1-x^2}\frac{1-x^6}{1-x^3}\cdots\frac{1-x^{2n}}{1-x^n} \\ &= \frac{1}{(1-x)(1-x^3)(1-x^5)\cdots} = o(x) \,. \end{aligned}$$

■

3.5 Permutations

Until this point, in each problem the order of choice was not important unlike permutations. It is known that the ordinary generating function for $P(n,k)$ which stands for the number of

k-permutations on a set of n elements is

$$\begin{aligned} P(x) &= P(n,0) + P(n,1)\,x + P(n,2)\,x^2 + \cdots + P(n,n)\,x^n \\ &= \sum_{k=0}^{n} P(n,k)\,x^k \\ &= \sum_{k=0}^{n} \frac{n!}{(n-k)!} x^k \end{aligned}$$

for a fixed n. Moreover the relation between $P(n,k)$ and $C(n,k)$ which stands for choosing k-elements from a set of n- elements is also known to be $P(n,k) = C(n,k)\,k!$ since every ordering count as a different permutation. As it is done in Example 2.0.1

$$C(x) = (1+x)^n$$

which immediately implies that

$$P(n,0)\,\frac{x^0}{0!} + P(n,1)\,\frac{x^1}{1!} + P(n,2)\,\frac{x^2}{2!} + \cdots + P(n,n)\,\frac{x^n}{n!} = (1+x)^n \ .$$

Therefore, in the expansion of $(1+x)^n$, the coefficient of $\dfrac{x^k}{k!}$ gives the number $P(n,k)$. This type of generating function, namely the generating function $\displaystyle\sum_{k=0}^{\infty} a_k \frac{x^k}{k!}$ of the sequence $\{a_n\}$ is called the **exponential generating function** for that sequence. This name makes perfect sense since for $\{a_n\} = \{1\}$ the generating function is

$$1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots = e^x.$$

Example 3.5.1 *To have a better understanding consider the number of ways to generate 5 letter words with or without meaning from the word "GENERATING". While generating the words a special attention for the letters E, N and G needs to be paid since there are two of them. To be more specific one should not count the word GENRE two times as the E's change places with each other. Except for E, N and G the representation of taking a letter is simply* $(1+x)$*, either take it or leave it. But for E,N and G, the representation becomes* $1 + x + \frac{x^2}{2!}$*. Therefore the coefficient of* $\frac{x^5}{5!}$ *in the product* $\left(1 + x + \frac{x^2}{2!}\right)^3 (1+x)^4$ *gives the number of all possible 5-letter words.*

The operations with exponential generating functions are similar to the ordinary generating functions. However there are some special features of them that one needs to pay attention.

Let $E(x)$ be the exponential generating function for the sequence $\{a_n\}$. After differentiation

$$D_x(E(x)) = a_1 + a_2 x + a_3 \frac{x^2}{2!} + a_4 \frac{x^3}{3!} + \cdots \tag{3.7}$$

belongs to the sequence $\{a_{n+1}\}$ and after integration, which works in the opposite way the exponential generating function

$$\int_0^x E(t)\, dt = a_0 x + a_1 \frac{x^2}{2!} + a_2 \frac{x^3}{3!} + \cdots \tag{3.8}$$

corresponds to $\{a_{n-1}\}$. In other words the operations shifting and truncating are seen as integration and differentiation respectively.

Another thing to point out is the multiplication of two exponential generating functions. Take the exponential generating functions $E(x)$ and $F(x)$ which correspond to $\{a_n\}$ and $\{b_n\}$ respectively. Then

$$\begin{aligned}
E(x)F(x) &= \sum_{i=0}^{\infty} a_i \frac{x^i}{i!} \sum_{j=0}^{\infty} b_j \frac{x^j}{j!} \\
&= \sum_{j,i\geqslant 0} \frac{a_j b_i}{j!i!} x^{j+i} \\
&= \sum_{k=0}^{\infty} x^k \sum_{i+j=k} \frac{a_j b_i}{j!i!} \\
&= \sum_{k=0}^{\infty} \frac{x^k}{k!} \sum_{i+j=k} k! \frac{a_j b_i}{j!i!} \\
&= \sum_{k=0}^{\infty} \frac{x^k}{k!} \sum_{i=0}^{k} \binom{k}{i} a_i b_{k-i}
\end{aligned}$$

which implies that the coefficient of $\frac{x^n}{n!}$ is

$$\sum_{k=0}^{n} \binom{n}{k} a_k b_{n-k} \,. \tag{3.9}$$

3.6 Functions

Since high school, it is taught that the number of functions that can be formed from a set X with n elements to another set Y with m elements is m^n. This can be easily seen by choosing an element from a set of m elements for every different $x_i \in X$. Now the same problem will be

solved by using generating functions. Every $y_i \in Y$ can be the image of one or more x_i values. Also since the function is not necessarily onto, y_i may not be an image of any x_i . This gives the representation

$$1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + \cdots = e^x$$

for a single y_i, therefore the overall generating function becomes $(e^x)^m = e^{xm} = \sum_{k=0}^{\infty} m^k \frac{x^k}{k!}$ with m^n as the coefficient of $\frac{x^n}{n!}$.

Another similar question is finding the number of onto functions from X to Y. This time, the only difference is that every y_i must be an image of some x_i so the function for an element in Y becomes

$$x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + \cdots = e^x - 1.$$

That gives the generating function

$$\begin{aligned}
(e^x - 1)^m &= \sum_{k=0}^{m} \binom{m}{k} e^{xk} (-1)^{m-k} \\
&= \sum_{k=0}^{m} \sum_{i=0}^{\infty} \binom{m}{k} \frac{(xk)^i}{i!} (-1)^{m-k} \\
&= \sum_{i=0}^{\infty} \left(\sum_{k=0}^{m} (-1)^{m-k} \binom{m}{k} k^i \right) \frac{x^i}{i!}
\end{aligned}$$

as the overall generating function, with $\sum_{k=0}^{\infty} (-1)^{m-k} \binom{m}{k} k^i$ being the number of onto functions.

3.7 Parentheses and Catalan Numbers

In this problem, one wants to count all possible ways of arranging n pairs of parentheses such that every arrangement is valid, meaning that the number of closing parentheses at a point on the row is always less then or equal to the number of opening ones. i.e: ())(is not a valid arrangement whereas ()(()()) is. If the number of possible ways with n pairs is denoted by c_n, then $c_3 = 5$; ((())) , (() ()) , (()) () , () (()) and () () (). Notice that in the first and second examples, the first parentheses is not closed until the end of the arrangement. This type of an arrangement is called *primitive* of order n. Then the third example is actually the concatenation of a primitive arrangement of order 2 and of order 1. So it can be said that an n-arrangement is generated by either using one primitive of order n or concatenating a primitive of order $k < n$ and some other arrangement. If the number of primitive arrangements with n

elements are denoted by p_n, then $c_n = \sum_{i=1}^{n} p_i c_{n-i}$, but what about p_n? It is seen that as one deletes the first and the last parenthesis - a pair- from a primitive arrangement, what remains are arrangements of order $n-1$ which may or may not be primitive. So $p_n = c_{n-1}$ and so $c_n = \sum_{i=1}^{n} c_{i-1} c_{n-i}$. Let $C(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + \cdots$ be the generating function for all arrangements, where we set $c_0 = 1$, then the generating function for $\{c_{n-1}\}$ is $xC(x)$ as we did in section (2.1.2). As a result

$$\begin{aligned} C(x) &= xC(x)C(x) + 1 \\ C(x) &= \frac{1 \pm \sqrt{1-4x}}{2x}. \end{aligned}$$

It can be seen that if the + sign is taken as the result, $\lim_{x\to\infty} C(x)$ does not exists. That is why the result should be the one with the $-$ sign. Hence, the generating function for the number of arranging parentheses is

$$C(x) = \frac{1 - \sqrt{1-4x}}{2x}.$$

Getting c_n from this function is not as straight forward as the previous ones. For that reason one needs to think $C(x)$ as $\frac{1}{2x}(1 - f(x))$, and get the power series representation for $f(x) = \sqrt{(1-4x)}$ first. To find that observe $f(x)$ as it is differentiated a few times to see how it behaves.

$$\begin{aligned} f'(x) &= -\frac{1}{2} 4 (1-4x)^{-\frac{1}{2}} \\ f''(x) &= -\frac{1}{2}\frac{1}{2} 4^2 (1-4x)^{-\frac{3}{2}} \\ f'''(x) &= -\frac{1}{2}\frac{1}{2}\frac{3}{2} 4^3 (1-4x)^{-\frac{5}{2}} \\ &\vdots \\ f^{(k)}(x) &= -\frac{1.3.5.7.9\cdots(2k-3)}{2^k} 4^k (1-4x)^{-\frac{2k-1}{2}} \end{aligned}$$

Since $f^{(k)}(0) = -\frac{(2k-2)!}{2^{k-1}(k-1)!} 2^k$,

$$\begin{aligned} f(x) &= 1 - \sum_{k=1}^{\infty} 2 \frac{(2k-2)!}{(k-1)!} \frac{x^k}{k!} \\ &= 1 - \sum_{k=1}^{\infty} \frac{2}{k} \binom{2k-2}{k-1} x^k. \end{aligned}$$

Then,

$$\begin{aligned} C(x) &= \frac{1}{2x}(1 - f(x)) \\ &= \frac{1}{2x}\sum_{k=1}^{\infty}\frac{2}{k}\binom{2k-2}{k-1}x^k \\ &= \sum_{k=1}^{\infty}\frac{1}{k}\binom{2k-2}{k-1}x^{k-1} \\ &= \sum_{k=0}^{\infty}\frac{1}{k+1}\binom{2k}{k}x^k. \end{aligned}$$

Now, it is seen that $c_n = \frac{1}{n+1}\binom{2n}{n}$. The sequence $\{c_n\}$ plays a very important role in combinatorics and it is used to solve lots of other problems. Named after Eugene Charles Catalan, the numbers in $\{c_n\}$ are called Catalan numbers.

3.8 Derangements

In this question, there is a distracted secretary, who is supposed to put n letters in n envelopes. Unfortunately the secretary mixes the letters and envelopes so they do not go to the places they are supposed to. The question is how many ways are there for the secretary to put the letters in the envelopes so that none of the letters go to its correct place. This question actually refers to a permutation in which none of the elements in the set go to its original place which is called a derangement. Let the set of derangements with n elements be denoted by D_n, then the union of all possible derangements, namely D_{n-k}, for all possible k values gives every single permutation of those n elements. While creating the set D_{n-k}, k elements are chosen to be fixed, meaning k letters to be put into the correct envelopes, then the rest is mixed such that none of them gets into its original envelope. So if d_n is the number of elements in D_n,

$$n! = \sum_{k=0}^{n}\binom{n}{k}d_k.$$

Since only the exponential generating function for $n!$ is known, taking the exponential generating function for both sides will be appropriate. It is seen from the equation (3.9) that the right hand side of this equation becomes $e^x D(x)$ if $D(x)$ is thought to be the generating function of d_n. So the above equation becomes

$$\begin{aligned} \frac{1}{1-x} &= e^x D(x) \\ D(x) &= \frac{e^{-x}}{1-x}. \end{aligned}$$

As always, getting the coefficient of x^n in the above equation will give the answer.

3.9 Involutions

An involution of a set S is a permutation of that set in which there exists no cycles of length greater then 2. In other words all permutations are their own inverses. For example for $S = \{1, 2, 3\}$, the number of involutions are $i_3 = 4$ which are $(1\,2\,3), (2\,1\,3), (1\,3\,2), (3\,2\,1)$. There are two different ways an involution with n elements can be formed. The first way is to put the element n to the n^{th} position, and distribute the remaining $n-1$ elements so that they form an involution. The second way is to put the element n not to its original position but some j^{th} where $j < n$. In that case, j is in the n^{th} position since every cycle has length at most 2. This implies that, the remaining $n-2$ elements, except n and j, must form an involution. Therefore the recursion for the number of involutions is

$$i_n = i_{n-1} + (n-1)\, i_{n-2} \,.$$

What about the generating function? As it is done for all ordered sets, an exponential generating function, say $I(x)$ is considered. Then as shown in the equation (3.8), the generating function for i_{n-1} and $(n-1)\, i_{n-2}$ is found to be $\int_0^x I(t)dt$ and $x\int_0^x I(t)dt$ respectively. Therefore

$$\begin{aligned} I(x) &= \int_0^x I(t)dt + x\int_0^x I(t)dt \\ &= (1+x)\int_0^x I(t)dt \\ \frac{I(x)}{1+x} &= \int_0^x I(t)dt \end{aligned}$$

and when both sides are differentiated

$$\begin{aligned} \frac{D_x(I(x))\,(1+x) - I(x)}{(1+x)^2} &= I(x) \\ D_x(I(x)) - I(x)\left(\frac{1}{1+x} + 1 + x\right) &= 0 \end{aligned}$$

is obtained. Solving this differential equation gives

$$\begin{aligned} I(x) &= \frac{1}{1+x} e^{-\left(x+\frac{x^2}{2}\right)} \\ &= \frac{1}{1+x}\left(\sum_{k=0}^{\infty}(-1)^k \frac{x^k}{k!}\right)\left(\sum_{l=0}^{\infty}(-1)^l \frac{x^{2l}}{2^l l!}\right) \\ &= \frac{1}{1+x}\sum_{k,l\geqslant 0}(-1)^k (-1)^l \frac{1}{2^l k! l!} x^{k+2l} \\ &= \sum_{m=0}^{\infty}\frac{x^m}{m!}\sum_{k+2l=m} m!\,(-1)^{k+l}\frac{1}{2^l k! l!} \\ &= \sum_{m=0}^{\infty}\frac{x^m}{m!}\sum_{l=0}^{\lfloor \frac{m}{2}\rfloor} m!\,(-1)^{m-l}\frac{1}{2^l l!\,(m-2l)!} \end{aligned}$$

where $i_n = \sum_{l=0}^{\lfloor \frac{n}{2}\rfloor} n!\,(-1)^{n-l}\frac{1}{2^l l!\,(n-2l)!}$.

3.10 Graphs

Throughout this section, simple and undirected graphs will be considered [2].

A labeled graph L with n vertices is a graph where every vertex $v \in V(L)$ is numbered differently. So two graphs L_1 and L_2 are the same if there is a 1-1 map $f : V(L_1) \to V(L_2)$ such that v_1 and v_2 are adjoint in L_1 if and only if $f(v_1)$ and $f(v_2)$ are adjoint in L_2. The number l_p of labeled graphs with n vertices and p edges for a fixed n can be thought as either getting an edge from the set of $\binom{n}{2}$ possible edges or not. So that the generating function becomes

$$L_p(x) = (1+x)^{\binom{n}{2}} = \sum_{k=0}^{n}\binom{\binom{n}{2}}{k}x^k.$$

Therefore the number of labeled graphs with n vertices is

$$L = L_p(1) = 2^{\binom{n}{2}}. \tag{3.10}$$

A connected graph is a graph in which every two vertices is joined by a path. To count the number of connected labeled graphs, one also needs to know what a rooted graph and a component is. A component of a graph is a maximal, connected sub-graph and a rooted graph is a graph where there is a vertex, called the root, which is distinguished from other vertices.

[2] A graph that has no loop (an edge which joins a vertex to itself) and no more than one edge between any two different vertices

Hence the number of rooted labeled graphs with p edges is pl_p. Furthermore, from these explanations, it is seen that the number of rooted labeled graphs is $\sum_{k=0}^{p} k\binom{p}{k} c_k l_{p-k}$ where c_k is the number of connected graphs with k vertices. This is true, since for every rooted graph, there is a component with k vertices which has the root in it. Knowing all these, the number of connected labeled graphs c_n can be found by subtracting the number of disconnected graphs -graphs with more then one component- from all possible labeled graphs

$$c_n = 2^{\binom{n}{2}} - \frac{1}{p}\sum_{k=0}^{p-1} k\binom{p}{k} 2^{\frac{p-k}{2}} c_k \, .$$

Now, let $C(x) = \sum_{k=1}^{\infty} c_k \frac{x^k}{k!}$ and $L(x) = \sum_{k=1}^{\infty} l_k \frac{x^k}{k!}$ be the exponential generating function for labeled connected graphs and labeled graphs respectively. It is desired to find a relationship between these two. For that reason again the idea above will be used. That is the union of k connected labeled graphs for all k gives us all labeled graphs. i.e. labeled graphs are composed of k connected labeled graphs. The result is

$$L(x) = \sum_{k=1}^{\infty} \frac{C^k(x)}{k!},$$

which gives us the relation between labeled and connected labeled graphs.

$$1 + L(x) = e^{G(x)} \tag{3.11}$$

The same idea can also be used to find the generating function of labeled Eulerian graphs which are connected even graphs [3]. To find the number of labeled Eulerian graphs, first the number of labeled even graphs w_n with n vertices will be found. It can be seen easily that $w_n = l_{n-1}$. The reason for that is, taking a labeled graph of order $n-1$, then connecting all the odd vertices to a new vertex to make them even gives us a labeled even graph of order n and this relation is 1-1. So $w_n = 2^{\binom{n-1}{2}}$ as in equation (3.10). Hence the function becomes

$$W(x) = \sum_{i=0}^{\infty} 2^{\binom{i-1}{2}} \frac{x^i}{i!}.$$

From that and the equation (3.11) the function $E(x)$ for Eulerian graphs will be

$$1 + W(x) = e^{E(x)}$$

$$\begin{aligned} E(x) &= ln(1 + W(x)) \\ &= x + \frac{x^3}{3!} + \frac{3x^4}{4!} + \frac{38x^5}{5!} + \cdots \end{aligned}$$

which gives $E_n = 2^{\binom{n-1}{2}} - \frac{1}{n}\sum_{k=0}^{n-1} k\binom{n}{k} 2^{\binom{n-k-1}{2}} E_k$.

[3] An even graph is a graph where each of its vertices have even degree.

3.11 Radioactive Particles

In this section, we will define two recursions that are related to each other. Let there be two particles, say α and β. At each minute, the particle α dissolves into 5 α-particles and 3 β-particles and the particle β dissolves into 2 α-particles and 4 β-particles. Then the question is given α_0 and β_0 how many α and β-particles are there after 30 minutes. So the recursions are

$$\alpha_n = 5\alpha_{n-1} + 2\beta_{n-1}$$
$$\beta_n = 3\alpha_{n-1} + 4\beta_{n-1}.$$

Let $f(x)$ and $g(x)$ be the generating functions for the sequences $\{\alpha_n\}$ and $\{\beta_n\}$ respectively. Since $xf(x)$ is the function for $\{0, \alpha_0, \alpha_1, \alpha_2 \ldots, \}$ and $xg(x)$ is the function for $\{0, \beta_0, \beta_1, \ldots, \}$

$$f(x) = 5xf(x) + 2xg(x) + \alpha_0$$
$$g(x) = 3xf(x) + 4xg(x) + \beta_0$$

which give

$$f(x) = \frac{\alpha_0 + 2xg(x)}{1 - 5x}$$
$$g(x) = \frac{\beta_0 + 3xf(x)}{1 - 4x}\,.$$

After solving them together we get

$$\begin{aligned} f(x) &= \frac{\alpha_0 + (-4\alpha_0 + 2\beta_0)x}{1 - 9x + 14x^2} \\ &= \frac{1}{5}\left(\frac{3\alpha_0 + 2\beta_0}{1 - 7x} + \frac{2\alpha_0 - 2\beta_0}{1 - 2x}\right) \\ g(x) &= \frac{\beta_0 + (-5\beta_0 + 3\alpha_0)x}{1 - 9x + 14x^2} \\ &= \frac{1}{5}\left(\frac{3\alpha_0 + 2\beta_0}{1 - 7x} + \frac{3\beta_0 - 3\alpha_0}{1 - 2x}\right). \end{aligned}$$

So the coefficients of x^{30} in both of the equations, which are

$$\alpha_{30} = \frac{1}{5}((3\alpha_0 + 2\beta_0)7^{30} + (2\alpha_0 - 2\beta_0)2^{30})$$
$$\beta_{30} = \frac{1}{5}((3\alpha_0 + 2\beta_0)7^{30} + (3\beta_0 - 3\alpha_0)2^{30})\,,$$

give the number of α and β-particles after 30 minutes.

CHAPTER 4

Constant Coefficient Linear Homogeneous Recursions

Definition 4.0.1 *A recurrence relation for a sequence* $\{a_n\}_{n=0}^{\infty}$ *is an expression that relates* a_n *to the previous terms of the sequence for all* $n \geq n_0$ *for some positive integer* n_0*. The terms* $a_0, a_1, \ldots, a_{n_0}$*, called the initial terms, should be defined explicitly.*

A recurrence relation represents different sequences as it is initialized differently. The set of all sequences a recurrence relation represents is called the solution set and two recursions are said to be equivalent if they have the same solution set.

If the recursion is of the form $a_n = F(a_{n-1}, a_{n-2}, \ldots, a_{n-d})$ for some fixed $1 \leq d \leq n_0$, then d is called the degree of the recursion.

To make it more clear, the recursion in section 3.3 for the regions of the plane has degree 1; the recursions for the Fibonacci sequence in section 3.2 and for the involutions in section 3.9 have degree 2. However the recursion for Catalan numbers described in section 3.7 does not have a degree since to express c_n all previous c_i's where $0 \leq i \leq n-1$ are used so 'd' changes as n changes.

A recursion does not have a degree does not necessarily mean that all of its equivalent recursions also do not have one. Take the recursion for partial sums which is $a_n = \sum_{i=0}^{n-1} a_i$, that does not have a degree; however, the recursion $a_n = 2a_{n-1}$ which also has the same solution set has degree 1.

The degree of a recursion is not the only distinctive property it has. Until now, we have dealed with different kinds of recursions. The recursion for parenthesis in section 3.7 has a non-linear coefficient $c_{i-1}c_{n-i}$, the one for involutions in section 3.9 has a not constant

coefficient $(n-1)\, i_{n-2}$ and the one for regions in the plane in section 3.3 has the coefficient n which makes it non homogeneous. However the recursion for the Fibonacci numbers satisfies everything that is expected from a constant coefficient linear homogeneous recursion.

A recurrence relation of the form

$$a_{n+k} = c_{k-1}a_{n+k-1} + c_{k-2}a_{n+k-2} + \cdots + c_0 a_n \tag{4.1}$$

is called a constant coefficient linear homogeneous recursion of order k provided that $c_0 \neq 0$. Such a relation is proper if given the first k values, the rest of the sequence is formed in terms of the given elements. Sometimes, instead of k initial values $m > k$ initial values may be given so that a_i, $0 \leq i \leq m$ do not satisfy the recursion but the rest do. Such recursions are called improper. From now on, we will study constant coefficient linear homogeneous proper recurrence relations unless stated otherwise.

Theorem 4.0.1 *A sequence satisfies a constant coefficient linear homogeneous recursion if and only if its generating function is a rational function.*

Proof. Assume that $\{a_n\}$, with generating function $A(x) = \sum_{i=0}^{\infty} a_i x^i$, satisfies $a_{n+k} = c_{k-1}a_{n+k-1} + c_{k-2}a_{n+k-2} + \cdots + c_0 a_n$, $c_0 \neq 0$. To prove this theorem, the idea that was used in section 3.2 to find the generating function of Fibonacci numbers will be used.

$$\begin{aligned}
\{a_n\} &\implies a_0\ a_1\ a_2\ a_3\ a_4 \cdots a_{k-1}\ a_k\ \cdots \\
\{a_{n+1}\} &\implies 0\ \ a_0\ a_1\ a_2\ a_3 \cdots a_{k-2}\ a_{k-1}\ \cdots \\
\{a_{n+2}\} &\implies 0\ \ 0\ \ a_0\ a_1\ a_2 \cdots a_{k-3}\ a_{k-2}\ \cdots \\
\{a_{n+3}\} &\implies 0\ \ 0\ \ 0\ \ a_0\ a_1 \cdots a_{k-4}\ a_{k-3}\ \cdots \\
&\ \ \vdots \\
\{a_{n+k-1}\} &\implies 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ \cdots\ \ 0\ \ a_0\ \cdots
\end{aligned}$$

After multiplying with the proper coefficients the above sequences are added, the appropriate changes are done to get the first $k-1$ elements properly and as a result $\{a_n\}$ is obtained. Hence,

$$\begin{aligned}
A(x) \;=\; & c_{k-1}xA(x) + c_{k-2}x^2A(x) + \cdots + c_0x^kA(x) \\
& + a_0 + (a_1 - c_{k-1}a_0)\,x + (a_2 - c_{k-1}a_1 - c_{k-2}a_0)\,x^2 + \cdots \\
& + (a_{k-1} - c_{k-1}a_{k-2} - \cdots - c_1a_0)\,x^{k-1}.
\end{aligned}$$

This implies that

$$\begin{aligned} A(x)\left(1-\sum_{j=1}^{k} c_{k-j}x^{j}\right) &= a_0+\sum_{i=1}^{k-1}\left(a_i-\sum_{j=1}^{i} c_{k-j}a_{i-j}\right)x^{i} \\ A(x) &= \frac{a_0+\sum_{i=1}^{k-1}\left(a_i-\sum_{j=1}^{i} c_{k-j}a_{i-j}\right)x^{i}}{1-\sum_{j=1}^{k} c_{k-j}x^{j}}. \end{aligned} \tag{4.2}$$

If the notation $A(x)=\frac{\overline{p(x)}}{\overline{q(x)}}$ is used, then $\overline{q(x)}$ is a polynomial of degree k that depends only on the coefficients of the recursion and $\overline{p(x)}$ is a polynomial with degree not exceeding $k-1$, moreover $\overline{p(x)}$ depends on the initial terms.

The other way of the if and only if statement can be proved by simply reversing the above operations. ■

Although the above proof assumes that the recursion is proper; the same idea also applies for improper recursions. Say $m+k$ elements are given instead of k where the first $m+k$ elements do not satisfy the recursion but the rest do. Such a sequence can be generated from a proper sequence by shifting elements by m terms in order to feed the sequence with the first improper m elements. In that case

$$\begin{aligned} \widetilde{A(x)} &= a_0+a_1x+\cdots+a_{m-1}x^{m-1}+x^{m}\frac{a_0+\sum_{i=1}^{k-1}\left(a_i-\sum_{j=1}^{i} c_{k-j}a_{i-j}\right)x^{i}}{1-\sum_{j=1}^{k} c_{k-j}x^{j}} \\ &= \frac{\left(a_0+a_1x+\cdots+a_{m-1}x^{m-1}\right)\overline{q(x)}+x^{m}\overline{p(x)}}{\overline{q(x)}} \\ &= \frac{\overline{p'(x)}}{\overline{q(x)}} \end{aligned}$$

where $\overline{p'(x)}$ has degree $m+k$ depending on all the initial conditions. Throughout this thesis as $\overline{p(x)}$ and $\overline{q(x)}$ are mentioned, they will be considered as they appear in the equation (4.2).

Even though it is proved that $\overline{q(x)}$ is of degree k , one must be aware of the fact that, for some initial values, $\overline{p(x)}$ and $\overline{q(x)}$ might have non-trivial common factors in which case, the generating function becomes $A(x)=\frac{p(x)}{q(x)}$ with $gcd(p(x),q(x))=1$ and $deg(q(x))<deg\left(\overline{q(x)}\right)$. From this point forward, when the notations $p(x)$ and $q(x)$ is used, it means that they do not have any non-trivial common factor.

Definition 4.0.2 *The linear complexity of a sequence is the smallest degree of the denominator of its generating function.*

Therefore the linear complexity is the degree of $q(x)$, not $\overline{q(x)}$ in general.

Example 4.0.1 *Consider the recurrence relation $a_{n+2} = 3a_{n+1} - 2a_n$ where the generating function is*

$$A(x) = \frac{a_0 + (a_1 - 3a_0)x}{1 - 3x + 2x^2} .$$

If the initial terms are $a_0 = 1$ and $a_1 = 2$,

$$A(x) = \frac{1 - x}{1 - 3x + 2x^2} = \frac{1}{1 - 2x}$$

which shows that actually, the sequence is $\{1, 2, 4, \cdots\}$ with the recursion $a_{n+1} = 2a_n$ which has linear complexity 1. However if the initial terms are taken as $a_0 = 1$ and $a_1 = 3$, then the function is

$$A(x) = \frac{1}{1 - 3x + 2x^2}$$

which represents the sequence $\{1, 3, 7, 15, \cdots\}$ with linear complexity 2 since $gcd(1 - 5x, 1 - 3x + 2x^2) = 1$.

4.1 General Term

Definition 4.1.1 *The "general term" of a sequence $\{a_n\}$ is an explicit expression that represents a_n in terms of n.*

Given the general term of a sequence, one can calculate the n^{th} term of that sequence, without knowing the previous terms which makes the general term valuable. However, most of the time, even though the recursion can be derived obviously from the given problem, the general term can not. In that case, the problem becomes finding the general term of the sequence from the recursion. If it is not found, one has to extract all the previous terms to find the n^{th} term. This problem is as complex as the problem of finding the solution for a given differential equation in the continuous case. Fortunately the method for finding the general term of a constant coefficient linear homogeneous recursion is well known. Actually that method is used in finding the general term f_n for the Fibonacci numbers in section 3.2 and the general term p_n for the number of regions in a plane in section 3.3.

Observation 4.1.1 *A constant coefficient linear homogeneous recursion has $\overline{q(x)} = 1 - \sum_{j=1}^{k} c_{k-j}x^j$, therefore $\overline{q(0)} = 1$. This implies that 0 is not a root of $\overline{q(x)}$ hence $x \nmid \overline{q(x)}$.*

Theorem 4.1.1 *Let the sequence* $\{a_n\} \in \mathbb{C}^{\infty}$, $(a_i \in \mathbb{C}, 1 \leq i \leq k)$ *have the generating function* $A(x) = \frac{p(x)}{q(x)}$ *where* $deg(q(x)) = k$ *and let* $(1-\alpha_1 x), \ldots, (1-\alpha_k x)$, $(\alpha_i \in \mathbb{C}\backslash\{0\})$ *be the distinct factors of* $q(x) \in \mathbb{C}$. *Then the general term is* $a_n = C_1\alpha_1^n + C_2\alpha_2^n + \cdots + C_k\alpha_k^n$ *where* $C_i \in \mathbb{C}$.

Proof. $q(x)$ having k distinct factors implies that

$$\begin{aligned} A(x) &= \frac{p(x)}{(1-\alpha_1 x)(1-\alpha_2 x)\cdots(1-\alpha_k x)} \\ &= \frac{C_1}{(1-\alpha_1 x)} + \frac{C_2}{(1-\alpha_2 x)} + \cdots + \frac{C_k}{(1-\alpha_k x)} \\ &= C_1 \sum_{i=0}^{\infty} \alpha_1^i x^i + C_2 \sum_{i=0}^{\infty} \alpha_2^i x^i + \cdots + C_k \sum_{i=0}^{\infty} \alpha_k^i x^i. \end{aligned}$$

Therefore the general term is found as $a_n = C_1\alpha_1^n + C_2\alpha_2^n + \cdots + C_k\alpha_k^n$. ■

Theorem 4.1.2 *Let the sequence* $\{a_n\} \in \mathbb{R}^{\infty}$, $(a_i \in \mathbb{R},\ 1 \leq i \leq k)$ *have the generating function* $A(x) = \frac{p(x)}{q(x)}$ *where* $deg(q(x)) = k$ *and* $q(x)$ *has no multiple roots, i.e.* $(1-\alpha_i x)$, $(\alpha_i \in \mathbb{C}\ \{0\})$, *are all different factors of* $q(x)$ *in* $\mathbb{C}$. *Then the general term is* $a_n = C_1\alpha_1^n + C_2\alpha_2^n + \cdots + C_k\alpha_k^n$ *for some* $C_i \in \mathbb{C}$.

Proof. As in the proof of the previous theorem $q(x)$ having k distinct roots in $\mathbb{C}$ implies that

$$\begin{aligned} A(x) &= \frac{p(x)}{(1-\alpha_1 x)(1-\alpha_2 x)\cdots(1-\alpha_k x)} \\ &= \frac{C_1}{(1-\alpha_1 x)} + \frac{C_2}{(1-\alpha_2 x)} + \cdots + \frac{C_k}{(1-\alpha_k x)} . \end{aligned}$$

Moreover, in this case, if $(1-\alpha_k x)$ is a factor of $q(x)$ such that $\alpha_k \notin \mathbb{R}$, then $(1-\overline{\alpha_k} x)$ is also a factor. So $A(x)$ can be written as

$$\frac{p(x)}{q(x)} = \frac{A}{1-\alpha_k x} + \frac{B}{1-\overline{\alpha_k} x} + \frac{\widehat{p(x)}}{\widehat{q(x)}}$$

Then

$$p(x) = A(1-\overline{\alpha_k} x)\widehat{q(x)} + B(1-\alpha_k x)\widehat{q(x)} + \widehat{p(x)}(1-\overline{\alpha_k} x)(1-\alpha_k x) .$$

When $x = \frac{1}{\alpha_k}$, $p\left(\frac{1}{\alpha_k}\right) = A\left(1-\frac{\overline{\alpha_k}}{\alpha_k}\right)\widehat{q\left(\frac{1}{\alpha_k}\right)}$ which implies

$$A = \frac{p\left(\frac{1}{\alpha_k}\right)\alpha_k}{\widehat{q\left(\frac{1}{\alpha_k}\right)}(\alpha_k - \overline{\alpha_k})} \tag{4.3}$$

and when $x = \frac{1}{\overline{\alpha_k}}$, $p\left(\frac{1}{\overline{\alpha_k}}\right) = B\left(1 - \frac{\alpha_k}{\overline{\alpha_k}}\right)\widehat{q\left(\frac{1}{\overline{\alpha_k}}\right)}$ which implies

$$B = \frac{p\left(\frac{1}{\overline{\alpha_k}}\right)\overline{\alpha_k}}{\widehat{q\left(\frac{1}{\overline{\alpha_k}}\right)}(\overline{\alpha_k} - \alpha_k)} \,. \tag{4.4}$$

From the equations (4.3) and (4.4), one can see that the complex conjugate of A is

$$\begin{aligned} \overline{A} &= \frac{\overline{p\left(\frac{1}{\alpha_k}\right)}\overline{\alpha_k}}{\overline{\widehat{q\left(\frac{1}{\alpha_k}\right)}(\alpha_k - \overline{\alpha_k})}} \\ &= \frac{p\left(\frac{1}{\overline{\alpha_k}}\right)\overline{\alpha_k}}{\widehat{q\left(\frac{1}{\overline{\alpha_k}}\right)}(\overline{\alpha_k} - \alpha_k)} = B \,. \end{aligned}$$

Therefore the summand corresponding to $(1 - \alpha_k x)$ and $(1 - \overline{\alpha_k} x)$ in the general term is

$$\begin{aligned} A\,(\alpha_k)^n + \overline{A}\,(\overline{\alpha_k})^n &= A\,(x + iy)^n + \overline{A}\,(x - iy)^n \\ &= (m + in)\sum_{i=0}^{n}\binom{n}{i}x^n\,(iy)^{n-i} + (m - in)\sum_{i=0}^{n}(-1)^n\binom{n}{i}x^n\,(iy)^{n-i} \\ &= m\left(\sum_{i=0}^{n}\binom{n}{i}x^n\,(iy)^{n-i} + \sum_{i=0}^{n}(-1)^n\binom{n}{i}x^n\,(iy)^{n-i}\right) \\ &\quad + in\left(\sum_{i=0}^{n}\binom{n}{i}x^n\,(iy)^{n-i} + \sum_{i=0}^{n}(-1)^{n+1}\binom{n}{i}x^n\,(iy)^{n-i}\right) \\ &= mM + in\,(iN) \qquad M, N \in \mathbb{R} \\ &= mM - nN \end{aligned}$$

which is a real number.

It's seen that even if $q\,(x)$ is factorized in $\mathbb{C}$, the general term always gives a real number because if $\frac{1}{\alpha_k}$ is a root so is $\frac{1}{\overline{\alpha_k}}$. So even the terms of the sequence are in $\mathbb{R}$, the factors can be found in $\mathbb{C}$ which is the smallest complete field containing $\mathbb{R}$. ■

Note that the above theorems do not cover multiple factors in the denominator.

Example 4.1.1 *Take the function* $g\,(x) = \frac{x}{1-4x+4x^2}$ *which has multiple factor* $(1 - 2x)^2$ *in the denominator. Then*

$$\begin{aligned} g\,(x) &= \frac{x}{(1 - 2x)^2} \\ &= -\frac{1}{2}\left(\frac{1}{1 - 2x} - \frac{1}{(1 - 2x)^2}\right) \\ &= \frac{1}{2}(\sum_{i=0}^{\infty} i2^i x^i - \sum_{i=0}^{\infty} 2^i x^i) \end{aligned}$$

which gives the general term $g_n = \frac{1}{2}(n2^n - 2^n) = n2^{n-1} - 2^{n-1}$.

This idea can be generalized as in the following theorem.

Theorem 4.1.3 *Let* $\{a_n\} \in \mathbb{C}^\infty$ *have the generating function* $A(x) = \frac{p(x)}{q(x)}$ *where* $deg(q(x)) = k$ *and* $(1 - \alpha_i x)$ *is a* $k-$*multiple factor of* $q(x)$ *in* $\mathbb{C}$. *Then the general term has summand* $\sum_{t=0}^{k-1} C_t n^t \alpha_i^n$ *corresponding to that factor.*

Proof. If $q(x)$ has a factor $(1 - \alpha_i x)^k$, then the corresponding summand in $A(x)$ for that factor becomes

$$\begin{aligned}
&\frac{B_{i,1}}{(1-\alpha_i x)} + \frac{B_{i,2}}{(1-\alpha_i x)^2} + \cdots + \frac{B_{i,k}}{(1-\alpha_i x)^k} \\
&= B_{i,1}\frac{1}{(1-\alpha_i x)} + B_{i,2}\frac{1}{\alpha_i}D_x\left(\frac{1}{(1-\alpha_i x)}\right) + \cdots + B_{i,k}\frac{1}{\alpha_i^{k-1}}D_x^{k-1}\left(\frac{1}{(1-\alpha_i x)}\right) \\
&= C_{i,1}\sum_{t=0}^{\infty}\alpha_i^t x^t + C_{i,2}\sum_{t=0}^{\infty} t\alpha_i^t x^t + \cdots + C_{i,k}\sum_{t=0}^{\infty} t^{k-1}\alpha_i^t x^t \\
&= \sum_{j=1}^{k} C_{i,j}\sum_{t=0}^{\infty} t^{j-1}\alpha_i^t x^t
\end{aligned}$$

which implies that the general term has the summand $\sum_{t=0}^{k-1} C_t n^t \alpha_i^n$. ■

It's seen easily that the same idea also applies for sequences in $\mathbb{R}$.

Until this point all the work is done in infinite fields. However, sometimes a desire to work in finite fields may occur. If that is the case, examples given below may be helpful to understand the basic idea of the operations done in finite fields.

Example 4.1.2 *Take the recursion* $r_{n+2} = r_{n+1} + 2r_n$ *in* $GF(5)$ *with initial conditions* $r_0 = 1$, $r_1 = 1$. *The generating function for the recursion is*

$$\begin{aligned}
A(x) &= \frac{1}{1 - x - 2x^2} \\
&= \frac{1}{1 + 4x + 3x^2} \\
&= \frac{c_1}{1+x} + \frac{c_2}{1+3x} \\
&= \frac{2}{1+x} + \frac{4}{1+3x}
\end{aligned}$$

hence the general term is $2 \cdot 4^n + 4 \cdot 2^n$ *in* $GF(5)$.

Example 4.1.3 *Take* $r_{n+3} = r_{n+1} + 2r_n$ *in* $GF(3)$ *with initial conditions* $r_0 = 1$, $r_1 = 1$, $r_2 = 1$. *The function for the recursion is* $A(x) = \frac{1}{1-x^2-2x^3} = \frac{1}{1+2x^2+x^3}$. *But in this case* $q(x)$ *is irreducible in* $GF(3)$. *As it is done on the infinite case, again one should go to the smallest complete field, which contains* $1 + 2x^2 + x^3$ *reducible, that is the splitting field* $GF(3)/1 + 2x^2 + x^3$. *In that case if* $1 - \alpha x$ *is a factor then* $1 - \alpha^3 x = 1 - (2 + \alpha)x$ *and* $1 - \alpha^9 x = 1 - (1 + \alpha)x$ *are also factors of* $1 + 2x^2 + x^3$. *So the generating function for the recursion can be represented as*

$$\begin{aligned} A(x) &= \frac{1}{1 - x^2 - 2x^3} \\ &= \frac{1}{1 + 2x^2 + x^3} \\ &= \frac{A}{1 - \alpha x} + \frac{B}{1 - (2 + \alpha)x} + \frac{C}{1 - (1 + \alpha)x} \\ &= \frac{\alpha}{1 - \alpha x} + \frac{2 + 2\alpha}{1 - (2 + \alpha)x} + \frac{2}{1 - (1 + \alpha)x} \end{aligned}$$

therefore the general term is $r_n = \alpha^{n+1} + (2 + 2\alpha)^{n+1} + 2(1 + \alpha)^n$.

4.2 Linear Complexity

Linear complexity of a recursion becomes important frequently in different topics. That is why we will go deeper and see how the complexity changes as different operations are applied to functions or sequences. During this section we will always deal with proper sequences. If as a result of an operation we get an improper one, we will take the linear complexity of the truncated proper sequence as the linear complexity of the generated sequence.

It is obvious that if every term of a sequence is multiplied by a constant, $\deg(q(x))$ and hence the linear complexity does not change. Also shifting the sequence by n, which actually corresponds to multiplying the corresponding generating function with x^n, does not change the complexity since 0 is not a root of $q(x)$[1]. But if n zeros are inserted between every pair of successive terms of a sequence namely if the variable of the corresponding function is changed from x to x^n, then the complexity becomes n times the original one since $\deg(q(x^n)) = n \deg(q(x))$.

[1] We must note that multiplying $p(x)$ with x^n may cause $\deg(p(x))$ become greater than $\deg(q(x))$ in which case we might need to specify more initial values, but we take the complexity of the ultimate sequence.

Theorem 4.2.1 *Let $\{a_n\}$ and $\{b_n\}$ be two different sequences with linear complexities l and k. Then the sequence $\{c_n\} = \{a_n + b_n\}$ has linear complexity not larger than $k + l$.*

Proof. Let $A(x) = \frac{p(x)}{q(x)}$ and $B(x) = \frac{p'(x)}{q'(x)}$ be generating functions of $\{a_n\}$and $\{b_n\}$ respectively. Then the generating function for $\{(a+b)_n\}$ is

$$\begin{aligned} C(x) &= \frac{p(x)}{q(x)} + \frac{p'(x)}{q'(x)} \\ &= \frac{p(x)\,q'(x) + p'(x)\,q(x)}{q(x)\,q'(x)} \\ &= \frac{P(x)}{Q(x)}. \end{aligned}$$

Notice that $P(x)$ and $Q(x)$ may have common factor, therefore

$$deg(Q(x)) \leq deg(q(x)) + deg(q'(x)) = k + l.$$

■

Theorem 4.2.2 *If two sequences with linear complexities k and l are convolved, the generated sequence will have linear complexity at most $k + l$.*

Proof. Let $A(x) = \frac{p(x)}{q(x)}$ and $B(x) = \frac{p'(x)}{q'(x)}$ be generating functions for $\{a_n\}$ and $\{b_n\}$ respectively. As the sequences are convolved, the generating function becomes

$$C(x) = \frac{p(x)\,p'(x)}{q(x)\,q'(x)} = \frac{P(x)}{Q(x)}$$

which implies that

$$deg(Q(x)) \leq k + l\,.$$

Equality occurs only when $gcd(p(x), q'(x)) = 1$ and $gcd(p'(x), q(x)) = 1$. ■

After this point one can see the affect of blending and regular decimation easily.

Theorem 4.2.3 *If two sequences with linear complexities k and l are blended, then the newly generated sequence will have a linear complexity not higher then $2(k + l)$.*

Proof. Remember from section 2.3 that when two sequences with generating functions $A(x) = \frac{p(x)}{q(x)}$ and $B(x) = \frac{p'(x)}{q'(x)}$ are blended the generating function of the resulting sequence

becomes

$$\begin{aligned} A\left(x^2\right)+xB\left(x^2\right) &= \frac{p\left(x^2\right)}{q\left(x^2\right)}+x\frac{p'\left(x^2\right)}{q'\left(x^2\right)} \\ &= \frac{p\left(x^2\right)q'\left(x^2\right)+xp'\left(x^2\right)q\left(x^2\right)}{q\left(x^2\right)q'\left(x^2\right)} \\ &= \frac{P\left(x\right)}{Q\left(x\right)}. \end{aligned}$$

In that case $deg\left(Q\left(x\right)\right) \leq 2\left(k+l\right)$ where $deg\left(q\left(x\right)\right) = l$ and $deg\left(q'\left(x\right)\right) = k$. ■

Theorem 4.2.4 *Taking the d-decimation ($d \in \mathbb{Z}^+$)of a sequence does not increase its linear complexity.*

Proof. When the d-decimation of a sequence with the function $A\left(x\right) = \frac{p(x)}{q(x)}$ is taken as in section 2.4

$$\widetilde{A\left(x\right)} = \frac{1}{d}\sum_{n=0}^{d-1} A\left(e^{\frac{2n\Pi i}{d}}x^{\frac{1}{d}}\right)$$

is obtained as the result which is the summation of d different functions with degree $\frac{1}{d}deg\left(A\left(x\right)\right)$ that gives $Q\left(x\right) \leq d\frac{1}{d}deg\left(q\left(x\right)\right) = deg\left(q\left(x\right)\right)$. ■

Example 4.2.1 *Remember the Example 2.4.1 where a zero is inserted after every two terms of the sequence $\{a_n\}$ to generate the sequence $S = \{a_0,\ a_1,\ 0,\ ,a_2,\ a_3,\ 0,\ \ldots\}$. The sequence S has generating function $D(x) = B(x^3) + xC(x^3)$ where $B(x)$ and $C(x)$ corresponds to the generating functions of the two decimated sequences. When Theorem 4.2.4 is considered it is seen that the linear complexity of the 2-decimated sequences has the same linear complexity as $\{a_n\}$ say l. Therefore S has linear complexity at most $3l$ since the variable is changed from x to x^3 in functions B and C.*

Example 4.2.2 *The sequence $S = \{a_0, a_1, a_3, a_4, a_6, a_7, \ldots\}$ generated from the sequence $\{a_n\}$ with linear complexity l in Example 2.4.2 has linear complexity $2l$ since it has the generating function $B(x^2)+xC(x^2)$ where the functions B and C corresponds to the decimated sequences.*

Lemma 4.2.1 *The linear complexity of the sequence $\{c_n\} = \{a_nb_n\}$, which is generated from the sequences $\{a_n\}$ and $\{b_n\}$ with linear complexities k and l has linear complexity not exceeding kl.*

Proof. Let the general terms of the given sequences be $a_n = \sum_{i=1}^{k} c_i\alpha_i^n$ and $b_n = \sum_{i=1}^{l} d_i\beta_i^n$. Then the general term of $\{c_n\}$ is

$$\begin{aligned} c_n &= a_n b_n \\ &= \sum_{i=1}^{k} c_i\alpha_i^n \sum_{i=1}^{l} d_i\beta_i^n \\ &= \sum_{t=1}^{kl} e_t\gamma_t \end{aligned}$$

where each $\gamma_t \in \{\alpha_i\beta_j | i = 1, \ldots, k; j = 1, \ldots, l\}$ and e_t are constants. Observe that there are at most kl different γ_t s which implies that the linear complexity of the sequence is not larger than kl . ■

Theorem 4.2.5 *If a sequence $\{b_n\}$ is generated by combining k sequences $\{a_n^1\}, \{a_n^2\}, \ldots, \{a_n^k\}$ with respective linear complexities $l_1, l_2, \ldots, l_k$ with a polynomial function f, i.e. $b_n = f(a_n^1, a_n^2, \ldots, a_n^k)$, then the linear complexity of $\{b_n\}$ is at most $g(l_1, l_2, \ldots, l_k)$ where g is obtained from the polynomial f by replacing every non-zero coefficients with 1.*

Proof. Proof follows from Theorem 4.2.1 and the above lemma immediately. ■

Example 4.2.3 *Combine three sequences $\{a_n\}, \{b_n\}$ and $\{c_n\}$ with complexities k, l and m respectively with a polynomial to get the new sequence $\{d_n\} = \{a_n + 2a_nb_n + b_nc_n\}$. The linear complexities of $\{a_nb_n\}$ and $\{b_nc_n\}$ are at most kl and lm respectively and even if every term of $\{a_nb_n\}$ is multiplied with the constant 2 to get $\{2a_nb_n\}$ the linear complexity does not change therefore the overall complexity becomes at most $k + kl + lm$.*

Lemma 4.2.2 *The linear complexity of the sequence $\{b_n\} = \{a_na_{n-t}\}$ where $0 \leq t \leq n$ has linear complexity at most $\binom{k}{2}$ where k is the linear complexity of $\{a_n\}$.*

Proof. Let $a_n = \sum_{i=1}^{k} c_i\alpha_i^n$ be the general term of $\{a_n\}$. Then the general term for $\{a_{n-t}\}$ is $\sum_{i=1}^{k} d_i\alpha_i^n$ since it has the same recurrence relation with different initial conditions. Hence the

general term for $\{b_n\}$ becomes

$$\begin{aligned} b_n &= a_n a_{n-t} \\ &= \sum_{i=1}^{k} c_i \alpha_i^n \sum_{i=1}^{k} d_i \alpha_i^n \\ &= \sum_{l=1}^{\binom{k}{2}} e_l \gamma_l \end{aligned}$$

where all $\gamma_l \in \{\alpha_i \alpha_j | i = 1, \ldots, k; j = 1, \ldots, k\}$ and e_l are constants. Observe that there can be at most $\binom{k}{2}$ different γ_k s so the linear complexity can not exceed $\binom{k}{2}$. ■

Theorem 4.2.6 *Given a sequence $\{a_n\}$ with linear complexity k, the sequence $\{b_n\}$ generated by polynomial f over the terms of $\{a_n\}$, i.e $b_n = f(a_n, a_{n-1}, \ldots, a_{n-i})$, has linear complexity at most $\sum_{j=0}^{d} \binom{k}{j}$ where d is the degree of the polynomial f.*

Proof. Proof follows from Theorem 4.2.1 and the above lemmas immediately. ■

To be more precise, the linear complexity of $\{b_n\}$ in the above theorem is $\binom{k}{d_1} + \binom{k}{d_2} + \cdots + \binom{k}{d_j}$ where j is the number of different degreed summands in f and d_j s are the corresponding degrees.

Example 4.2.4 *Let $\{a_n\} = \sum_{i=1}^{l} c_i \alpha_i^n$ have linear complexity l and $\{b_n\} = \{a_n + 2a_{n-1}a_{n-2} + a_n a_{n-2}\}$. The linear complexity of $\{a_{n-1}a_{n-2}\}$ is less then or equal to $\binom{l}{2}$ because the set $\{\gamma_t\}$ where $\gamma_t \in \{\alpha_i \alpha_j | i, j = 1, \ldots, l\}$ has at most $\binom{l}{2}$ elements. As in the previous example shifting and multiplying with a constant does not change the linear complexity so the sequences $\{2a_{n-1}a_{n-2}\}$ and $\{a_n a_{n-2}\}$ have linear complexities $\binom{l}{2}$ moreover, they have the same set of γ_ts. So, the overall linear complexity of $\{b_n\}$ becomes at most $l + \binom{l}{2}$.*

Lemma 4.2.3 *Let $\{a_n\}$ and $\{b_n\}$ be two sequences formed in a way that they both affect each other linearly, i.e $a_n = f(a_{n-1}, \ldots, a_{n-k}) + f'(b_{n-1}, \ldots, b_{n-k'})$ and $b_n = g'(a_{n-1}, \ldots, a_{n-l'}) + g(b_{n-1}, \ldots, b_{n-l})$ where f, f', g, g' are all linear functions with complexities k, k', l, l' respectively, then the linear complexities of both of these sequences are at most $max(k + l, l' + k')$ and moreover they actually represent the same recurrence relation.*

Proof. Let $\{a_n\}$ and $\{b_n\}$ have generating functions $A(x)$ and $B(x)$ respectively which give

$$A(x) = p_1(x)A(x) + p_1'(x)B(x) + \alpha_0 + \alpha_1 x + \cdots + \alpha_{K-1}x^{K-1}$$
$$B(x) = p_2'(x)A(x) + p_2(x)B(x) + \beta_0 + \beta_1 x + \cdots + \beta_{L-1}x^{L-1}$$

where $K = max(k, k')$, $L = max(l, l')$ and p_1 is formed from the function f such that if ca_{n-t} is a summand in f then there exist the coefficient cx^t for the corresponding sequence in the function p_1 . In a similar way p_1', p_2 and p_2' are generated from the functions g, f' and g' respectively. Moreover α_i and β_j are chosen so that the initial conditions hold. Then

$$(1 - p_1(x))A(x) = p_1'(x)B(x) + \alpha_0 + \alpha_1 x + \cdots + \alpha_{K-1}x^{K-1}$$
$$(1 - p_2(x))B(x) = p_2'(x)A(x) + \beta_0 + \beta_1 x + \cdots + \beta_{L-1}x^{L-1}$$

and solving them together gives

$$\begin{aligned}(1 - p_1(x))A(x) &= p_1'(x)\frac{p_2'(x)A(x) + \beta_0 + \cdots + \beta_{L-1}x^{L-1}}{(1 - p_2(x))} + \alpha_0 + \alpha_1 x + \cdots + \alpha_{K-1}x^{K-1} \\ A(x) &= \frac{\beta_0 + \beta_1 x + \cdots + \beta_{L-1}x^{L-1} + (1 - p_2(x))(\alpha_0 + \alpha_1 x + \cdots + \alpha_{K-1}x^{K-1})}{(1 - p_1(x))(1 - p_2(x)) - p_2'(x)p_1'(x)}\end{aligned}$$

and

$$\begin{aligned}(1 - p_2(x))B(x) &= p_2'(x)\frac{p_1'(x)B(x) + \alpha_0 + \cdots + \alpha_{K-1}x^{K-1}}{(1 - p_1(x))} + \beta_0 + \beta_1 x + \cdots + \beta_{L-1}x^{L-1} \\ B(x) &= \frac{\alpha_0 + \alpha_1 x + \cdots + \alpha_{K-1}x^{K-1} + (1 - p_1(x))(\beta_0 + \beta_1 x + \cdots + \beta_{L-1}x^{L-1})}{(1 - p_1(x))(1 - p_2(x)) - p_2'(x)p_1'(x)}.\end{aligned}$$

As it is seen both of the generating functions have the same denominator so the same recurrence relation, moreover the linear complexities are at most $max(k+l, k'+l')$ since $p_1(x)p_2(x)$ has degree $k + l$ and $p_1'(x)p_2'(x)$ has degree $k' + l'$. ■

Theorem 4.2.7 *If k sequences $\{a_n^1\}, \{a_n^2\}, \ldots, \{a_n^k\}$ are generated such that all of them affect each other linearly, i.e $a_n^i = f_1^i(a_{n-1}^1, \ldots, a_{n-l_1^i}^1) + f_2^i(a_{n-1}^2, \ldots, a_{n-l_2^i}^2) + \cdots + f_k^i(a_{n-1_k}^k, \ldots, a_{n-l_k^i}^k)$ where f_j^i s are all linear $i, j = 1, \ldots, k$, then the linear complexity of any of them can not exceed $max\sum_{j=0}^{k} l_j^{i_j}$ where $1 \le i_j \le k$ are all different.*

Proof. Follows from the previous lemma immediately. ■

An example to such sequences can be seen in the radioactive particles question in section 3.11.

4.3 Periodicity

Definition 4.3.1 *A sequence $\{a_n\}$ is said to be periodic with $s \in \mathbb{N}$, if $a_{n+s} = a_n$ is satisfied for all $n \in \mathbb{N}$. Smallest such s is the period of the sequence.*

From the definition it can be said that if a sequence is periodic then it has the generating function $A(x) = \frac{\widetilde{p(x)}}{1-x^s}$. The numerator is denoted by a tilde to indicate that $gcd(p(x), 1-x^s)$ may or may not be 1.

Theorem 4.3.1 *Let $A(x) = \frac{p(x)}{q(x)}$ be the generating function of the sequence $\{a_n\}$. Then $\{a_n\}$ is periodic with s if and only if $q(x) | 1-x^s$.*

Observation 4.3.1 *Periodicity and the period of the sequence, represented by the same generating function, may change as the field changes.*

Example 4.3.1 *It is immediately seen that the sequence $\{1,0,1,0,\ldots\}$ with the function $f(x) = \frac{1}{1-x^2}$ has period 2 or the sequence $\{1,2,3,\ldots\}$ represented by $\frac{1}{(1-x)^2}$ does not have a period in $\mathbb{C}$ since $(1-x)^2 \nmid 1-x^s$ for any $s \in \mathbb{C}$. However, the period of the same generating function $\frac{1}{(1-x)^2} = \frac{1}{1-2x+x^2}$ in $GF(3)$ exists. Actually the corresponding sequence is $\{1,2,0,1,2,0\ldots\}$ which has period 3 since $1-2x+x^2 = 1+x+x^2 | 1-x^3 = (1-x)(1+x+x^2)$. Moreover the same generating function has period 5 in $GF(5)$ since $1-2x+x^2 = 1+3x+x^2 | 1-x^5 = (1-x)(1+3x+x^2)^2$.*

Theorem 4.3.2 *The general term of a periodic sequence $\{a_n\} \in \mathbb{F}$ of order s is $a_n = \sum_{j=1}^{s} C_j \alpha_j^n$ where α_j s are the s^{th} root of unities over $\mathbb{F}$.*

Proof. If $\{a_n\}$ has period s then its generating function is

$$\begin{aligned} A(x) &= \frac{\widetilde{p(x)}}{1-x^s} \\ &= \frac{\widetilde{p(x)}}{(1-\alpha_1 x)(1-\alpha_2 x)\ldots(1-\alpha_s x)} \\ &= \frac{C_1}{(1-\alpha_1 x)} + \frac{C_2}{(1-\alpha_2 x)} + \cdots + \frac{C_s}{(1-\alpha_s x)} \end{aligned}$$

Therefore the general term becomes $a_n = \sum_{j=1}^{s} C_j \alpha_j^n$ where α_j s are the s^{th} root of unities. ■

In the previous section we have observed how linear complexity changes when different operations are applied to sequences. What about the change of the period when the same operations are applied. The idea is very similar. Shifting and multiplying every term of a sequence with a constant does not change the period since it does not change the denominator of the corresponding generating function. If two sequences with periods m and l where $A(x) = \frac{p(x)}{q(x)}$ such that $q(x) \mid 1 - x^m$ and $B(x) = \frac{p'(x)}{q'(x)}$ such that $q(x) \mid 1 - x^l$ respectively are taken, both the sum and the product of the two functions give the denominator $q(x)\, q'(x) \mid (1 - x^m)\left(1 - x^l\right) \mid 1 - x^{lcm(l,m)}$.

Theorem 4.3.3 *Every sequence in a finite field which satisfies a recurrence relation of a certain degree is periodic.*

Proof. An element a_n in a sequence that satisfies a recurrence relation can be written as a function of the previous k elements. Assume that the field has m distinct elements. Then there exist m^k different k-tuples that an element can be generated from. So in the worst case, after m^k k-tuples a k-tuple repeats itself, thus the sequence is periodic. ■

CHAPTER 5

Linear Feedback Shift Registers

There is a need for random sequences in different areas including mathematics, statistics, computer engineering and cryptography. However, generating truly random sequences is a hard task. Even though, any sequence obtained as an output of an algorithm is not random, some particular algorithms can be used to obtain sequences that look like random which are called pseudo random sequences. It is expected from those sequences to possess some of the properties a random sequence possess. A common way to generate such sequences is feedback shift registers.

A feedback shift register, r-FSR, is a machine with r-registers, $x_1,\ x_2,\ \ldots,\ x_r$ that generates every element a_n from previous r elements with respect to a function $f(x_1,\ x_2,\ \ldots,\ x_r)$ and some initial elements $a_0,\ \ldots,\ a_{r-1}$. At the n^{th} clocking a_{n+1} is generated, a_is in the registers are shifted one stage and a_{n+1} is pulled. If output is needed then a_{n-r} is taken as output.

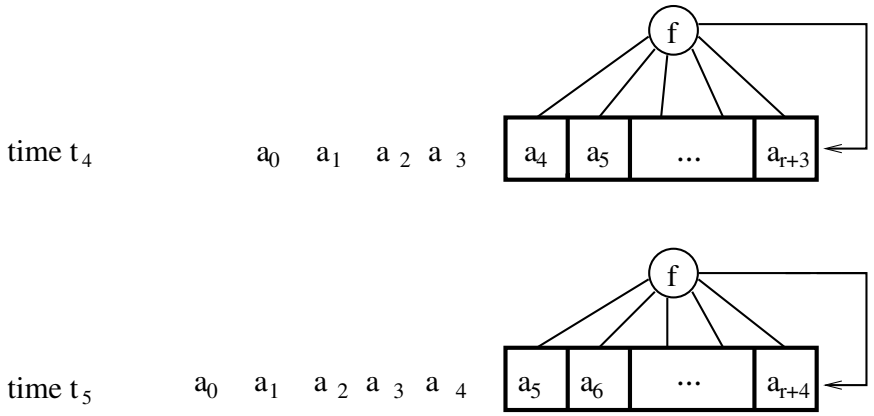

Figure 5.1: Diagram of a feedback shift register

The sequences generated by FSRs can be thought as the sequences that were studied in the previous sections and elements in the registers while initialization can be thought as the initial terms of those sequences. The functions and so the recursions, can be linear or non-linear with constant or non-constant coefficients. However, in this section only the FSRs with constant coefficient linear homogeneous recursions which are called linear feedback shift registers, LFSRs, will be studied.

Even though LFSRs can be defined in any field and generate sequences as described in previous sections, because of its importance and common use, they are examined on the fields with characteristic 2 in this section unless stated otherwise. Therefore recursions are of the form $a_{n+r} = c_{r-1}a_{n+r-1} \oplus c_{r-2}a_{n+r-2} \oplus \cdots \oplus c_0 a_n$ where $c_i \in \{0, 1\}$ and $\oplus$ is addition in the field $GF(2)$.

Definition 5.0.2 *Let D be the shift operator $D : a_n \to a_{n-1}$. Define as $D^0 = 1$, $D^1 = D$ and $D^n = D \circ D \circ \cdots \circ D$ for $n > 1$. It is evident that $D^r a_n = a_{n-r}$. Therefore any given recursion of the form $a_{n+r} + c_{r-1}a_{n+r-1} \oplus c_{r-2}a_{n+r-2} \oplus \cdots \oplus c_0 a_n = 0$ can be written as*

$$\begin{aligned} a_{n+r} \oplus c_{r-1}Da_{n+r} \oplus \cdots \oplus c_0 D^r a_{n+r} &= 0 \\ \left(c_0 D^r \oplus c_1 D^{r-1} \oplus \cdots \oplus c_{r-1}D^{r-1}\right) a_{n+r} &= 0 \\ L(D)\, a_{n+r} &= 0\,. \end{aligned}$$

$L(D)$ in the definition is called the connection polynomial and an LFSR is represented as $< r, L(D) >$.

Example 5.0.2 *Look at the recurrence relation $a_{n+5} = a_{n+3} \oplus a_{n+1} \oplus a_n$ and its corresponding LFSR with initial elements* $(1, 0, 1, 0, 0, 1, 1)$.

$$\begin{aligned} a_{n+5} \oplus a_{n+3} \oplus a_{n+1} \oplus a_n &= 0 \\ D^5 a_n \oplus D^3 a_n + Da_n + a_n &= 0 \\ (D^5 \oplus D^3 \oplus D \oplus 1)a_n &= 0 \end{aligned}$$

$D^5 \oplus D^3 \oplus D \oplus 1$ is the connection polynomial of the corresponding LFSR (figure (5.2)) which can be represented as $< 7, (D^5 \oplus D^3 \oplus D \oplus 1) >$.

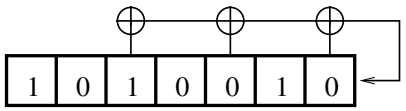

Figure 5.2: LFSR with $< 7, (D^5 \oplus D^3 \oplus D \oplus 1) >$

An important observation coming from the Example 5.0.2 is that the first two terms of the LFSR do not affect the rest of the sequence. Therefore if those two terms are truncated, the ultimate sequence will still be represented by the same connection polynomial. In such a case, representing the truncated LFSR as $< 5, (D^5 \oplus D^3 \oplus D \oplus 1) >$ or equivalently $L(D) = D^5 \oplus D^3 \oplus D \oplus 1$ is enough. From this point forward only ultimate sequences will be worked on as it is done in previous sections by assuming the length of the LFSR being equal to the degree of its connection polynomial unless stated otherwise.

Now consider the generating function of the sequence, derived from the LFSR in Example 5.0.2. It can be found as

$$\begin{aligned} A(x) &= \frac{1 \oplus x^2 \oplus x^5 \oplus x^6}{1 \oplus x \oplus x^3 \oplus x^5} \\ &= \frac{\widetilde{p(x)}}{L(x)} \end{aligned}$$

using the Theorem 4.2. Hence $\widetilde{q(x)} = L(x)$ and one can see easily that this is true for any LFSR.

Recall that if $L(x)$ is reducible then the period and the linear complexity of the sequence may change as the initial conditions change because for some initial terms $gcd(\widetilde{p(x)}, \widetilde{q(x)})$ may be different then 1. This is why the connection polynomial $L(D)$ and hence $\widetilde{q(x)}$ will be taken to be irreducible and they will be denoted by $q(x)$ without causing any confusion throughout this section. As a result, an r-LFSR with these properties has linear complexity r without any confusion.

Observation 5.0.2 *The all zero state* $(0)_r = (0, 0, \ldots, 0)$ *is an exceptional state. Observe that for* $(0)_r$ *to occur at a time t, the state at time* $t - 1$ *must be either* $(1, 0, 0, \ldots, 0)$ *or* $(0, 0, 0, \ldots, 0)$. *However, the connection polynomials are taken to be constant coefficient linear and homogeneous with degree r. That is why the state* $(1, 0, \ldots, 0)$ *is never followed by the all zero state. Therefore if at any time the state* $(0)_r$ *is observed, one can say without a doubt that the r-LFSR is actually initialized with* $(0)_r$.

Theorem 5.0.4 *The period of an LFSR with r-registers is at most* $2^r - 1$.

Proof. From Theorem 4.3.3, it is already known that an LFSR has finitely many states, at most 2^r. Moreover it is observed that the all zero state may not occur in an LFSR unless it is initialized with it. And if it is initialized so, the period will be simply 1. Therefore, for the period to be maximum, it should not be initialized with $(0)_r$ which gives $2^r - 1$ distinct states. That is why the period of an r-LFSR is at most $2^r - 1$. ■

Throughout this section, the initialization vector is always assumed to be different then $(0)_r$.

Lemma 5.0.1 *If* $\mathbb{F}$ *is a finite field with q elements and* $\mathbb{K}$ *is a subfield of* $\mathbb{F}$*, then the polynomial* $x^q - x$ *in* $\mathbb{K}[x]$ *factors in* $\mathbb{F}[x]$ *as*

$$x^q - x = \prod_{\alpha \in \mathbb{F}} (x - \alpha)$$

and $\mathbb{F}$ *is a splitting field of* $x^q - x$ *over* $\mathbb{K}$*.*

Lemma 5.0.1 shows that if an LFSR with generating function $A(x) = \frac{p(x)}{q(x)}$ has r-registers, $deg(q(x)) = r$, then $q(x) \in GF(2^r)[x]$, namely $q(x)|x^{2^r} - x$. To be more precise $q(x) \mid x^{2^r-1} - 1$ since $x \nmid q(x)$ as shown in the previous section. This also proves the Theorem 5.0.4.

Definition 5.0.3 *If the period of the generated sequence for an r-LFSR is maximal, i.e* $2^r - 1$*, then* $q(x)$ *is a primitive polynomial.*

The above definition also suits with the definition of a primitive polynomial in a finite field, where a primitive polynomial is a polynomial that powers of its root covers all nonzero elements in the generated field.

As a result of the above definition, the function $q(x)$ in the generating function is always taken as a primitive polynomial throughout this section to have the maximum period.

The properties of sequences generated by LFSRs become important in cryptography as it is mentioned in the beginning of this section. LFSRs are most widely used in stream cipher based crypto-systems in which randomness is very important. The generated sequences must satisfy the randomness postulates of Golomb which can be studied deeply from [5]. It is seen that if an LFSR has maximal length, then it satisfies all of these postulates.

Linear complexity is also an important issue. Using the Berlekamp-Massey algorithm[6], one can find the generating function of a sequence with linear complexity l in $O(l^2)$ time by knowing only $2l$ terms. For this search to be infeasible, in cryptography a linear complexity larger than 2^{80} is used nowadays. Also it is desired for the sequence to have a very high period, so that combined with linear complexity and randomness, the LFSR based stream cipher acts much like a Vernam Cipher. Moreover, to increase the security, instead of using one very large LFSR it is preferred to combine reasonably small LFSRs somehow and to achieve a high linear complexity and period. This is why we want to observe how complexity and period changes as more than one LFSRs are combined.

Get two LFSRs with lengths l_1 and l_2 generated by primitive connection polynomials then add the outputs of every clock to get a new sequence. Remember from Theorem 4.2.1 that the new sequence has linear complexity at most $l_1 + l_2$. Observe that initially from the two LFSRs, there are already $l_1 + l_2$ registers and with a proper connection polynomial one can get the maximal linear complexity easily, so it is seen that even though an additional operation is done, the linear complexity of the sequence generated by $l_1 + l_2$ registers did not increase. As oppose to this, when the outputs at each clock of the generated sequences are multiplied, a sequence with linear complexity of $l_1 l_2$ can be taken as a result if the connection polynomials are chosen properly as seen in the proof of lemma 4.2.1.

Lemma 5.0.2 *Take $\alpha_i \in GF(2^k)$ and $\beta_j \in GF(2^l)$. The products $\alpha_i\beta_j$ for all i,j are distinct if $gcd(k, l) = 1$.*

Theorem 5.0.5 *Taking two maximal length LFSRs with lengths l_1 and l_2 with the condition that $gcd(l_1, l_2) = 1$ and generating a sequence by multiplying the elements popped from the LFSRs gives a sequence with linear complexity at most $l_1 l_2$.*

Proof. Let the generated sequences from the two LFSRs L_1 and L_2 have general terms $\sum_{i=1}^{l_1} c_i\alpha_i^n$ and $\sum_{j=1}^{l_2} d_i\beta_i^n$ where each $1 - \alpha_i x$ and $1 - \beta_i x$ are factors of the connection polynomials of L_1 and L_2 respectively. It is known from the proof of lemma 4.2.1 that the generated sequence has general term $\sum e_t\gamma_t^n$ where each $\gamma_t \in \alpha_i\beta_j$ for all $i = 1, \ldots, l_1$ and $j = 1, \ldots, l_2$. If all γ_ts are distinct than the generated sequence will have linear complexity at most $l_1 l_2$. It is seen from the above lemma that the upper bound is reached when $gcd(l_1, l_2) = 1$ and initial terms are chosen properly. ■

Using the above idea one can combine some relatively short LFSRs and get a sequence with higher linear complexity.

5.1 Nonlinear Combiner

One technique to generate a random looking sequence is to use n different LFSRs working in parallel and to combine the outputs of them in every clock using a non-linear function f to increase the linear complexity.

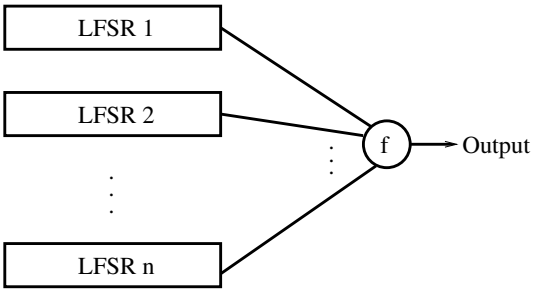

Figure 5.3: A non-linear combiner

The function f has n inputs say $x_1,\ x_2,\ \ldots,\ x_n$ which are the outputs of the LFSRs in every clock. The length of the LFSRs will be defined relatively prime with each other to get greater linear complexity as stated in Theorem 5.0.5.

Theorem 5.1.1 *The linear complexity of the output sequence of a non-linear combiner which uses n LFSRs with lengths* $l_1,\ l_2,\ \ldots,\ l_n$ *and a non-linear function* $f(x_1,\ x_2,\ \ldots,\ x_n) \in GF(2)$ *is at most* $f(l_1, l_2, \ldots, l_n) \in \mathbb{Z}^+$.

Proof. The proof follows from the Theorems 5.0.5 and 4.2.5 ■

Theorem 5.1.2 *If maximum length LFSRs are used in the combiner with pairwise relatively prime lengths* $l_1, l_2, \ldots, l_n$ *then the period of the generated sequence is* $(2^{l_1}-1)(2^{l_2}-1)\cdots(2^{l_n}-1)$.

5.1.1 Geffe Generator

Geffe Generator is an example of a non-linear combiner with three LFSRs say L_1, L_2 and L_3. In every clock all of the LFSRs are clocked, if the output of L_2 is 1 the output of L_1 is taken as the system output, otherwise the output of L_3 is taken. One can observe that this schedule can be described using the function

$$\begin{aligned} f(x_1, x_2, x_3) &= x_1x_2 \oplus (1 \oplus x_2)x_3 \\ &= x_1x_2 \oplus x_2x_3 \oplus x_3 \end{aligned}$$

where x_i is the output of L_i at a time. With the knowledge of f, one can easily say that the linear complexity of the generator is at most $l_1l_2 + l_2l_3 + l_3$ and the period is at most $(2^{l_1} - 1)(2^{l_2} - 1)(2^{l_3} - 1)$ where l_i is the length of L_i. Moreover, if l_is are pairwise relatively prime and the connection polynomials are primitive, then the maximum value is reached with the proper initialization.

5.1.2 Majority Generator

A majority generator is another example of a non-linear combiner. In this section the one with three LFSRs will be defined, however one can use any odd number of LFSRs without facing with any problems. Let the LFSRs be defined as in the Geffe Generator schedule with primitive connection polynomial. In every clock, all the LFSRs are clocked and the output of the system is taken as the majority of the outputs of the three LFSRs, i.e. If any two of them gives output 1, then the output of the system is 1, otherwise it is 0. As it is done in the previous section for Geffe Generator, once again the system can be described with a function, namely $f(x_1, x_2, x_3) = x_1x_2 \oplus x_2x_3 \oplus x_1x_3$. Using this function one can find the highest possible linear complexity as $l_1l_2 + l_2l_3 + l_1l_3$ which is higher then the Geffe generator; however the period is the same.

It may seem that as linear complexity of the system increases, the security also increases however, this may not be the case. While combining the LFSRs, one may loose the correlation immunity or balancedness properties of the sequence which may cause serious weaknesses. Different attacks like correlation attacks may be applied to such ciphers. The techniques to prevent those attacks is not in the scope of this thesis and will not be mentioned but we feel obligated to make this warning.

5.2 Nonlinear Filter

In some cases, instead of using more than one LFSRs, one may have a desire to use only one single LFSR but still increase the linear complexity. In such cases, a non-linear filter may be the answer.

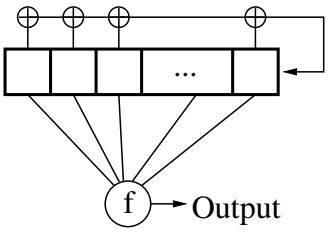

Figure 5.4: A non-linear Filter

Theorem 5.2.1 *The linear complexity of the sequence generated by a non-linear function from an l-LFSR is at most* $2^l - 1$.

Proof. Remember from Theorem 4.2.6 that the linear complexity of the sequence is at most $\sum_{i=1}^{d} \binom{d}{i}$ where d is the degree of the recursion; in this case it can be viewed as the number of registers multiplied in the filter. So the maximum value is reached when all the registers are used and there exist at least one term of all possible degrees in the recursion. ■

Example 5.2.1 *Let L be a maximum length LFSR with degree l. If the filtering function is* $f = x_1x_2 + x_5$ *then the linear complexity of the system becomes* $\binom{5}{2} + \binom{5}{1} = 15$ *as it is explained in lemma 4.2.2. Moreover even if the function is changed as* $f = x_1x_2 + x_2x_4 + x_5$ *the linear complexity does not change as in Example 4.2.4 and the maximum linear complexity is reached if the function is* $f = x_1x_2x_3x_4x_5 + x_1x_2x_4x_5 + x_3x_4x_5 + x_2x_4 + x_1$, *namely if there exist a term for every possible degree in the function.*

5.3 Some Other Possible Generators

The knowledge of filters and combiners yield to combine more then one LFSR in different ways. The figures (5.5) and (5.6) are some examples. In both of the examples the functions are linear functions. However one must be careful that the output of an LFSR affects the other

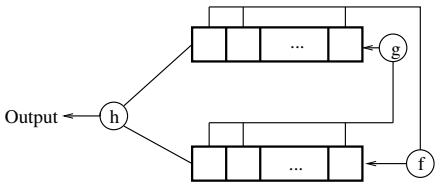

Figure 5.5: Combination of two LFSRs without filtering

one. With a similar idea that was used in the proof of lemma 4.2.3, the generating functions of the output sequences of the LFSRs just before the function h can be written as $A(x)$ and $B(x)$ respectively. Let the first LFSR have linear complexity l_1 after the f function and the other one have linear complexity l_2 right after the g function. Then the generating functions $A(x)$ and $B(x)$ for the sequences in figure (5.5) can be written easily as

$$\begin{aligned} A(x) &= \alpha_0 + \alpha_1 x + \cdots + \alpha_{r_1} x^{r_1-1} + p_1(x)B(x) \\ B(x) &= \beta_0 + \beta_1 x + \cdots + \beta_{r_2} x^{r_2-1} + p_2(x)A(x) \end{aligned}$$

where $p_1(x)$, $p_2(x)$ and α_is and β_is are formed as in lemma 4.2.3.

After solving these equations one gets

$$\begin{aligned} A(x) &= \alpha_0 + \alpha_1 x + \cdots + \alpha_{r_1} x^{r_1-1} + p_1(x)(\beta_0 + \beta_1 x + \cdots + \beta_{r_2} x^{r_2-1} + p_2(x)A(x)) \\ &= \frac{\alpha_0 + \alpha_1 x + \cdots + \alpha_{r_1} x^{r_1-1} + p_1(x)\beta_0 + p_1(x)\beta_1 x + \cdots + p_1(x)\beta_{r_2} x^{r_2-1}}{1 - p_1(x)p_2(x)} \end{aligned}$$

and

$$\begin{aligned} B(x) &= \beta_0 + \beta_1 x + \cdots + \beta_{r_2} x^{r_2-1} + p_2(x)(\alpha_0 + \alpha_1 x + \cdots + \alpha_{r_1} x^{r_1-1} + p_1(x)B(x)) \\ &= \frac{\beta_0 + \beta_1 x + \cdots + \beta_{r_2} x^{r_2-1} + p_2(x)\alpha_0 + p_2(x)\alpha_1 x + \cdots + p_2(x)\alpha_{r_1} x^{r_1-1}}{1 - p_1(x)p_2(x)}. \end{aligned}$$

It is seen that both of the sequences have linear complexity at most $l_1 + l_2$ that is the sum of the linear complexities of the original sequences generated by the LFSRs. Moreover, if the initial conditions are chosen properly, $\gcd(p(x), q(x))$ may be equal to one therefore the maximum bound will be reached. The rest is just an easy combining operation.

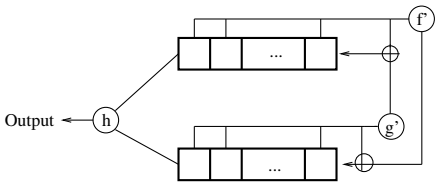

Figure 5.6: Combination of two LFSRs with filtering

Now, look at the structure in figure (5.6). It is seen from the lemma 4.2.3 that the linear complexity of the whole structure just before the h function is again l_1+l_2 which is an expected result since the maximum degrees of the linear functions f' and g' are l_1 and l_2 respectively which is the same as the degrees of the feedback functions.

Moreover, the period in both cases are the same which is at most $(2^{r_1}-1)(2^{r_2}-1)$ if they have primitive connection polynomials with $gcd(r_1, r_2) = 1$.

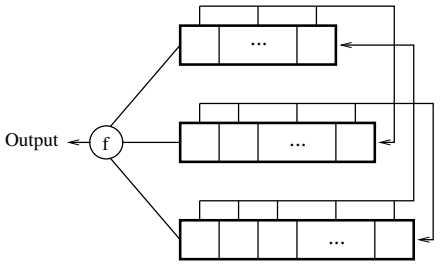

Figure 5.7: Combination of three LFSRs without filtering

Even though the above generators have two LFSRs, while finding the linear complexity and the period of generators that has more than two LFSRs the same idea is used. As an example look at the figure (5.7). When the same method to find the linear complexity of the generator as in figure (5.5) is applied, it is seen that the linear complexity of the system just before the combining operation is at most $l_1 + l_2 + l_3$ if l_i is the length of the LFSR L_i. Moreover if the connections polynomials are irreducible and the initial conditions are chosen properly, the upper bound is reached. In addition to that, if the polynomials are primitive and l_is are relatively prime, than the period of the system is $(2^{l_1}-1)(2^{l_2}-1)(2^{l_3}-1)$.

As a last example look at the figure where the first and the third LFSRs affect the second and the second affecting them. Let the LFSRs have linear complexity l_1, l_2 and l_3, then the maximum possible degrees are also the same for the linear functions f, g and h respectively. Therefore the maximum possible linear complexity is $l_1 + l_2 + l_3$. Moreover, assuming that

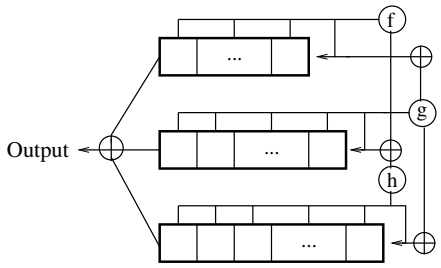

Figure 5.8: Combination of three LFSRs with filtering

the polynomials are primitive and l_is are relatively prime, than the period of the system is $(2^{l_1} - 1)(2^{l_2} - 1)(2^{l_3} - 1)$.

Even though the functions f, g and h here are said to be linear, they can also be a filtering operation in which case the linear complexity of the sequences after the filtering operations increase. Remember from Theorem 4.2.7 that the linear complexity is calculated by adding the maximum linear complexities of the sequences generated by the LFSR. So for example if f was a filtering operation where the sequence has linear complexity l'_1 after the operation, then the linear complexity of the system can be at most $l'_1 + l_2 + l_3$ by choosing proper initial conditions.

5.4 Clock Control Generators

In this type of generators, as opposed to what was done so far, the LFSRs in the system are not controlled by the same clock. This means that they may not be clocked at the same time. Generally the clocking of an LFSR is determined by the output of another LFSR in the system. In this thesis, two particularly important types of clock controlled generators, namely shrinking generator and alternating step generator, and their behaviors will be examined.

5.4.1 Shrinking generator

A shrinking generator is composed of two LFSRs L_1 and L_2 with lengths l_1 and l_2 respectively. Both of the LFSRs are clocked at the same time. This opposes how the clock control generators are defined but since a shrinking generator is taken as a clock control generator by the researchers studying on this subject, we take this subject in this chapter. In a shrink-

ing generator the output of L_2 is taken as the system output only when the output of L_1 is 1; otherwise the output of L_2 is discarded. Remember that the connection polynomial of the LFSRs are taken to be primitive, so that they will possess all the randomness postulates of Golomb. Therefore, the number of ones in a period is nearly half of the length of the period. Hence the period of the generator is $2^{l_1-1}(2^{l_2}-1)$. While finding an upper bound for the linear complexity $L(x)$ of the system, again the number of ones in the first LFSR is used. Since there are 2^{l_1-1} ones in L_1, by taking proper decimations for 2^{l_1-1} different starting points and inserting $2^{l_1-1}-1$ zeroes in between every consecutive terms of the decimated sequences, the decimated sequences will have linear complexity at most $l_2 2^{l_1-1}$. As we add those sequences after making proper changes, the output of the shrinking generator is gained therefore, $L(x) \leq l_2 2^{l_1-1}$.

In the previous formation of generators, finding the upper bound for the system was enough because, by looking at the denominator of the generating functions, the conditions for the linear complexity to reach the upper bound can be observed easily. However, in this type of generators, analyzing the generating functions is difficult therefore it is necessary to give a lower bound for the linear complexity. By using the properties of decimations and period, it can be seen that $l_2 2^{l_1-2} < L(x)$. Hence the linear complexity of the generator $L(x)$ satisfies $l_2 2^{l_1-2} < L(x) \leq l_2 2^{l_1-1}$.

5.4.2 Alternating Step Generator

Three LFSRs L_1, L_2 and L_3 with linear complexities l_1, l_2 and l_3 and output sequences $\{a_n\}$, $\{b_n\}$ and $\{c_n\}$ respectively are used for this type of generators. L_1 is used as the clock control and the others determine the output sequence. To be more clear, L_2 is clocked if the output of L_1 is 1, otherwise L_3 is clocked and the sum of the outputs of L_2 and L_3 at that time is taken. Therefore at time t, the output can be represented as $r_t = b_{T(t)} \oplus c_{t-T(t)-1}$ where $T(t) = (\sum_{i=0}^{t} a_i) - 1$. As opposed to shrinking generators, in alternating step generators the whole periods are used for every LFSR,that is why the period of the generator is $(2^{l_1}-1)(2^{l_2}-1)(2^{l_3}-1)$. Moreover it can be shown by using the properties of decimation that the linear complexity $L(x)$ satisfies $(L_2+L_3)2^{L_1-1} < L(x) \leq (L_2+L_3)2^{L_1}$.

REFERENCES

[1] Herbert S. Wilf. *generatingfunctionology*. Academic Press, Inc., 2 edition, 1994.

[2] Fred S. Roberts. *Applied combinatorics*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1984. ISBN 0-13-39313-4.

[3] F. Harary and E. M. Palmer. *Graphical Enumeration*. New York, Academic Press, 1973. ISBN 0123242452.

[4] Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1996. ISBN 0849385237.

[5] Solomon W. Golomb. *Shift Register Sequences*. Holden-Day, 1967.

[6] James L. Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, 15:122–127, 1969.