The Frobenius Problem in a Free Monoid

by

Zhi Xu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2009

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Given positive integers $c_1, c_2, \ldots, c_k$ with $\gcd(c_1, c_2, \ldots, c_k) = 1$, the Frobenius problem (FP) is to compute the largest integer $g(c_1, c_2, \ldots, c_k)$ that *cannot* be written as a non-negative integer linear combination of $c_1, c_2, \ldots, c_k$. The Frobenius problem in a free monoid (FPFM) is a non-commutative generalization of the Frobenius problem. Given words $x_1, x_2, \ldots, x_k$ such that there are only finitely many words that *cannot* be written as concatenations of words in $\{ x_1, x_2, \ldots, x_k \}$, the FPFM is to find the longest such words. Unlike the FP, where the upper bound $g(c_1, c_2, \ldots, c_k) \leq \max_{1 \leq i \leq k} c_i^2$ is quadratic, the upper bound on the length of the longest words in the FPFM can be exponential in certain measures and some of the exponential upper bounds are tight. For the 2FPFM, where the given words over Σ are of only two distinct lengths m and n with $1 < m < n$, the length of the longest omitted words is $\leq g(m, m\,|\Sigma|^{n-m} + n - m)$.

In Chapter 1, I give the definition of the FP in integers and summarize some of the interesting properties of the FP. In Chapter 2, I give the definition of the FPFM and discuss some general properties of the FPFM. Then I mainly focus on the 2FPFM. I discuss the 2FPFM from different points of view and present two equivalent problems, one of which is about combinatorics on words and the other is about the word graph. In Chapter 3, I discuss some variations on the FPFM and related problems, including input in other forms, bases with constant size, the case of infinite words, the case of concatenation with overlap, and the generalization of the local postage-stamp problem in a free monoid. In Chapter 4, I present the construction of some essential examples to complement the theory of the 2FPFM discussed in Chapter 2. The theory and examples of the 2FPFM are the main contribution of the thesis. In Chapter 5, I discuss the algorithms for and computational complexity of the FPFM and related problems. In the last chapter, I summarize the main results and list some open problems.

Part of my work in the thesis has appeared in the papers [83, 84, 157].

Acknowledgements

First, I wish to thank my supervisor Jeffrey O. Shallit, who is a great scholar, a nice friend of mine, and a good teacher who introduced to me the non-commutative Frobenius problem that finally became the topic of this thesis. I owe him a great deal for his valuable suggestions in our regular discussions and for his patience in reading my poorly-written manuscripts and in improving my presentation skills. I cannot find enough words to express my gratitude for his help in all aspects in my PhD life.

I also wish to thank my temporary supervisor professor Richard Trefler and all my friends in the WatForm group for their kindness that made my first year in Waterloo an enjoyable memory.

I wish to thank professor Janusz Brzozowski, professor Jonathan Buss, professor Ondřej Lhoták, professor Ming Li, all my friends in the Algorithm and Complexity group, and all the people who have ever helped me during my stay at University of Waterloo. Without their favors, my progress in the PhD program could not have proceeded so smoothly.

I wish to thank Dr. Narad Rampersad for reading my thesis thoroughly and providing helpful comments.

I wish to thank the members of my examining committee, professor Richard Cleve, professor Ming Li, professor Kai Salomaa, and professor Edlyn Teske for their precious time in reviewing my thesis, providing comments, and attending my defense.

I particularly want to thank my parents and my sister for their endless love and selfless support. Without that, I could not have gone so far in my career.

The research in the thesis was partly supported by David R. Cheriton Graduate Scholarships, the International Doctoral Student Award from University of Waterloo, and financial support from the David R. Cheriton School of Computer Science at the University of Waterloo.

Dedication

献给我的父亲母亲

(To my parents)

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In §1.1, I will first give some basic definitions and notation in number theory, and then give the definition of the original Frobenius problem. In §1.2, I will give a brief introduction to some of the basic results about the Frobenius problem in the literature. The discussion in the first two sections will take place in the domain of non-negative integers. Then, in §1.3, we will see how some results from the study of the Frobenius problem can be applied to analyzing and solving several problems in other research areas. In §1.4, we will see how the Frobenius problem can be generalized to the setting of a free monoid, which composes the main topic of this thesis. At the end, in §1.5, I will outline the organization of the thesis.

1.1 Introduction to the Frobenius problem

1.1.1 Number theory preliminaries

Let $\mathbb{N} = \{ 0, 1, 2, \ldots \}$ denote the set of all non-negative integers, let $\mathbb{Z} = \{ 0, \pm 1, \ldots \}$ denote the set of all integers, and let $q\mathbb{Z} = \{ 0, \pm q, \pm 2q, \ldots \}$ denote the set of all integral multiples of the number q. We say q *divides* p if p is an integral multiple of q and write $q \mid p$.

Let $| x |$ denote the absolute value of the number x. For any integers p and q, $q \neq 0$, the integer p can be uniquely represented as $p = sq + r$, where $s, r \in \mathbb{Z}$ and $0 \leq r < | q |$.

Let $\gcd(x_1, \ldots, x_k)$ denote the *greatest common divisor* and let $\operatorname{lcm}(x_1, \ldots, x_k)$ denote the *least common multiple* of integers $x_1, \ldots, x_k$. The operation gcd (respectively, lcm) satisfies the associative law $\gcd(a, \gcd(b, c)) = \gcd(\gcd(a, b), c)$, and the distributive law $a \gcd(b, c) = \gcd(ab, ac)$.

Lemma 1.1.1. *Let $x_1, x_2, \ldots, x_k \in \mathbb{Z}$ with $\gcd(x_1, x_2, \ldots, x_k) = d$. Then there exist $c_1, c_2, \ldots, c_k \in \mathbb{Z}$ such that $c_1 x_1 + c_2 x_2 + \cdots + c_k x_k = d$.*

Corollary 1.1.2. *Let $x_1, x_2, \ldots, x_k \in \mathbb{Z}$ with $\gcd(x_1, x_2, \ldots, x_k) = d$. Then $d = 1$ if and only if there exist $c_1, c_2, \ldots, c_k \in \mathbb{Z}$ such that*

$$c_1x_1 + c_2x_2 + \cdots + c_kx_k = 1. \tag{1.1}$$

We say p is *congruent* to q *modulo* k if k divides $p-q$ and write $p \equiv q \pmod{k}$. The modulo-k congruence relation is an equivalence relation. Each equivalence class in this case is called a *residue class* and the set of all modulo-k residue classes is denoted by $\mathbb{Z}_k = \mathbb{Z}/k\mathbb{Z}$. The minimal non-negative integer that is congruent to p modulo k is denoted by $p \bmod k$.

Lemma 1.1.3. *If $\gcd(p, q) = 1$, then for each integer r the equation*

$$px \equiv r \pmod{q} \tag{1.2}$$

has one unique solution x up to congruence $\pmod{q}$.

An integer $p > 1$ is called a *prime* if p has no positive divisor except 1 and p. The only integers that divide 1 are ± 1 and we do not call 1 a prime.

Euler's function $\phi(n)$ gives for each positive integer n the number of those positive integers m satisfying $0 < m \leq n$ and $\gcd(n, m) = 1$. Note that $\phi(1) = 1$.

1.1.2 Integer Frobenius problem

Given k positive integers $x_1, x_2, \ldots, x_k$, we say that an integer n can be written as a *non-negative integer linear combination* of $x_1, x_2, \ldots, x_k$, if there exist non-negative integers $c_1, c_2, \ldots, c_k$ such that

$$n = c_1x_1 + c_2x_2 + \cdots + c_kx_k. \tag{1.3}$$

We call this sequence of positive integers $x_1, x_2, \ldots, x_k$ a *basis*. Without loss of generality, we always assume $0 < x_1 \leq x_2 \leq \cdots \leq x_k$ without explicit explanation in all of the following chapters when we mention a basis (of integers). We use

$$\langle x_1, x_2, \ldots, x_k \rangle = \left\{ c_1x_1 + c_2x_2 + \cdots + c_kx_k : c_1, c_2, \ldots, c_k \in \mathbb{N} \right\} \tag{1.4}$$

to represent the set of all integers that can be written as non-negative integer linear combinations of $x_1, x_2, \ldots, x_k$, and say it is *generated* by the basis $x_1, x_2, \ldots, x_k$.

Deciding whether $\mathbb{N} \setminus \langle x_1, x_2, \ldots, x_k \rangle$ is finite

It is a folklore result that if $\gcd(x_1, x_2, \ldots, x_k) = 1$, then there are only finitely many non-negative integers that are not in $\langle x_1, x_2, \ldots, x_k \rangle$. This property has been used in additive number theory [142] and in probability theory [46, p. 336].

Lemma 1.1.4. *Let $x_1, x_2, \ldots, x_k$ be positive integers with* $\gcd(x_1, x_2, \ldots, x_k) = 1$. *There exists an integer N such that $n \in \langle x_1, x_2, \ldots, x_k \rangle$ for all integers $n > N$.*

Proof. (Taken from Feller's textbook [46, p. 336]) Since $\gcd(x_1, x_2, \ldots, x_k) = 1$, by Lemma 1.1.1, there exist integers $c_1, c_2, \ldots, c_k$ such that $c_1x_1 + \cdots + c_kx_k = 1$. Let $q = \sum_{i=1}^{k} x_i$ and

$$N = q^2 \max_{1 \le i \le k} |\, c_i \,|\,. \tag{1.5}$$

Now we prove that for all integers $n > N$, $n \in \langle x_1, x_2, \ldots, x_k \rangle$. Each integer $n > N$ can be uniquely represented as $n = N + pq + r$, where $p, r \in \mathbb{N}$, $0 \le r < q$. Then

$$n = q^2 \max_{1 \le i \le k} |\, c_i \,| + pq + r \sum_{i=1}^{k} c_i x_i = \sum_{i=1}^{k} (q \max_{1 \le i \le k} |\, c_i \,| + p + rc_i) x_i, \tag{1.6}$$

where $q \max_{1 \le i \le k} |\, c_i \,| + p + rc_i > 0$ for all $1 \le i \le k$. Hence $n \in \langle x_1, x_2, \ldots, x_k \rangle$. □

Furthermore, if there are only finitely many non-negative integers that are not in $\langle x_1, x_2, \ldots, x_k \rangle$, then $\gcd(x_1, x_2, \ldots, x_k) = 1$. Otherwise, there are infinitely many non-negative integers not in $\langle x_1, x_2, \ldots, x_k \rangle$. Suppose $\gcd(x_1, x_2, \ldots, x_k) = d > 1$. Then any integer of the form $nd+1$ is not in $\langle x_1, x_2, \ldots, x_k \rangle$ for all $n \in \mathbb{N}$. Therefore, the following theorem holds.

Theorem 1.1.5. *Let $x_1, x_2, \ldots, x_k$ be positive integers. There are only finitely many non-negative integers that are not in $\langle x_1, x_2, \ldots, x_k \rangle$ if and only if*

$$\gcd(x_1, x_2, \ldots, x_k) = 1. \tag{1.7}$$

Definition of the (integer) Frobenius problem

If the number of non-negative integers that are not in $\langle x_1, x_2, \ldots, x_k \rangle$ is finite, one natural question is, what is the largest integer among those integers? The *Frobenius problem* (FP) is as follows:

Problem 1.1.6 (*Frobenius problem*)**.** *Given k positive integers $x_1, x_2, \ldots, x_k$ such that* $\gcd(x_1, x_2, \ldots, x_k) = 1$, *what is the largest integer that* cannot *be represented as a non-negative integer linear combination of $x_1, x_2, \ldots, x_k$?*

The largest integer that is not in $\langle x_1, x_2, \ldots, x_k \rangle$ is called the *Frobenius number* of $x_1, x_2, \ldots, x_k$, and it is usually denoted by $g(x_1, x_2, \ldots, x_k)$.

The Frobenius problem is an old problem, and when it first appeared is not completely clear. Ferdinand G. Frobenius (1849–1917) had discussed the problem occasionally in his lectures in the late 1800's (according to Brauer [17]), although he did not publish anything on the problem (but obviously he knew this problem [49]).

Table 1.1: An integer Frobenius problem example — $g(3,5) = 7$

$1 \notin \langle 3,5 \rangle$,	$2 \notin \langle 3,5 \rangle$,	$3 = 1 \cdot 3 + 0 \cdot 5$,
$4 \notin \langle 3,5 \rangle$,	$5 = 0 \cdot 3 + 1 \cdot 5$,	$6 = 2 \cdot 3 + 0 \cdot 5$,
$7 \notin \langle 3,5 \rangle$,	$8 = 1 \cdot 3 + 1 \cdot 5$,	$9 = 3 \cdot 3 + 0 \cdot 5$,
$10 = 0 \cdot 3 + 2 \cdot 5$,	$11 = 2 \cdot 3 + 1 \cdot 5$,	$12 = 4 \cdot 3 + 0 \cdot 5$,
$13 = 1 \cdot 3 + 2 \cdot 5$,	$14 = 3 \cdot 3 + 1 \cdot 5$,	$15 = 5 \cdot 3 + 0 \cdot 5$,
	... ...	

Example 1.1.7. Let $k = 2, x_1 = 3, x_2 = 5$. Then the first few integers generated by the basis x_1, x_2 are shown in Table 1.1. Any larger integer that is not shown in Table 1.1 can be written as a sum of a positive multiple of 3 plus one of the integers $10, 11, 12$, and thus is in $\langle 3,5 \rangle$. So $g(3,5) = 7$.

There are also other names used for the Frobenius problem in the literature, such as the *money-changing problem* (or the *money-changing problem of Frobenius*, or the *coin-exchange problem of Frobenius*) [172, 60, 137], the *coin problem* (or the *coin problem of Frobenius*) [149, 143, 163, 60], the *Chicken McNuggets problem* [168], and the *Diophantine problem of Frobenius* (or the *linear Diophantine problem of Frobenius* and other combinations of the nouns such as the *Diophantine Frobenius problem*) [137, 126, 35].

Problem 1.1.8 (*Chicken McNuggets problem*)**.** *Chicken McNuggets™ are available in packs of either 6, 9, or 20 nuggets at McDonald's®. What is the largest number of Chicken McNuggets™ that one cannot purchase without throwing any away?*

Solution. (Taken from Vardi's book [168, pp. 233–234]) The answer is

$$g(6,9,20) = 43. \tag{1.8}$$

To see this, first of all, 43 is not in $\langle 6,9,20 \rangle$. Suppose $43 = 6a + 9b + 20c$ for some non-negative integers a, b, c. Then $c \leq 2$ and $43 \equiv 6a + 9b + 20c \pmod 3$, which implies $c \equiv 2 \pmod 3$. So $c = 2$ and thus $6a + 9b = 3$, which has no non-negative integer solution. Now all of the integers from 44 to 49 are in $\langle 6,9,20 \rangle$ as shown below:

$$\begin{aligned} 44 &= 1 \cdot 6 + 2 \cdot 9 + 1 \cdot 20, & 45 &= 0 \cdot 6 + 5 \cdot 9 + 0 \cdot 20, \\ 46 &= 1 \cdot 6 + 0 \cdot 9 + 2 \cdot 20, & 47 &= 0 \cdot 6 + 3 \cdot 9 + 1 \cdot 20, \\ 48 &= 8 \cdot 6 + 0 \cdot 9 + 0 \cdot 20, & 49 &= 0 \cdot 6 + 1 \cdot 9 + 2 \cdot 20. \end{aligned}$$

Since every integer $n \geq 44$ can be written as $n = 6k + m$, where $k \in \mathbb{N}$ and $m \in \{ 44, 45, 46, 47, 48, 49 \}$, all integers $n \geq 44$ are in $\langle 6,9,20 \rangle$. Therefore, 43 is the largest integer that is not in $\langle 6,9,20 \rangle$. In fact, one can check that the only positive integers that are not in $\langle 6,9,20 \rangle$ are 1, 2, 3, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 22, 23, 25, 28, 31, 34, 37, and 43. □

1.2 Research on the FP

Because of Theorem 1.1.5, one can efficiently decide whether $g(x_1, x_2, \ldots, x_k)$ exists for arbitrary positive integers $x_1, x_2, \ldots, x_k$ by computing $\gcd(x_1, x_2, \ldots, x_k)$. To compute $g(x_1, x_2, \ldots, x_k)$, however, is generally a more difficult task.

1.2.1 Formulae for the Frobenius number

Cases $k = 2$, $k = 3$, and $k = 4$

There exist simple formulae for the FP for a small fixed k. When $k = 2$, it is folklore that $g(x_1, x_2) = x_1x_2 - x_1 - x_2$, which is sometimes cited as due to Sylvester [165], although he did not actually provide this equation (for the available literature, see [4]).

Theorem 1.2.1. *Let x_1, x_2 be two positive integers with $\gcd(x_1, x_2) = 1$. Then*

$$g(x_1, x_2) = x_1x_2 - x_1 - x_2. \tag{1.9}$$

Proof. (Rewritten from a proof of Nijenhuis and Wilf [117]) Let $N = x_2(x_1-1)-x_1$, and let $n > N$ be an integer. By Lemma 1.1.3, the equation

$$x_2x \equiv n \pmod{x_1} \tag{1.10}$$

has a unique solution $x = c_2$ such that $0 \leq x < x_1$. Then $x_1 \mid n - x_2c_2$. So we can write $n - x_2c_2 = x_1c_1$ for some integer c_1. Then

$$x_1c_1 = n - x_2c_2 > N - x_2(x_1 - 1) = -x_1, \tag{1.11}$$

which implies $c_1 \geq 0$. Therefore, $n = x_1c_1 + x_2c_2 \in \langle x_1, x_2\rangle$ for any integer $n > N$.

Suppose $N = x_2(x_1 - 1) - x_1 \in \langle x_1, x_2\rangle$. Let $N = c_1x_1 + c_2x_2$, where $c_1, c_2 \in \mathbb{N}$. Then

$$-x_2 \equiv c_2x_2 \pmod{x_1}. \tag{1.12}$$

Since $\gcd(x_1, x_2) = 1$, it follows that $c_2 \equiv -1 \pmod{x_1}$. So $c_2 \geq x_1 - 1$. On the other hand, since $c_2x_2 \leq N < x_2(x_1 - 1)$, we have $c_2 < x_1 - 1$. No such c_2 exists. Hence N is not in $\langle x_1, x_2\rangle$.

Therefore, $g(x_1, x_2) = N = x_1x_2 - x_1 - x_2$. □

Example 1.2.2. To compute the Frobenius number $g(3, 5)$ in Example 1.1.7, one can apply Formula (1.9) and get

$$g(3, 5) = 3 \cdot 5 - 3 - 5 = 7. \tag{1.13}$$

Although when $k = 2$ there is a simple formula to compute the Frobenius number easily, the Frobenius problem becomes much harder for any fixed $k \geq 3$. So far, when $k \geq 4$, there is no simple formula known for the Frobenius number except for various special cases of $x_1, x_2, \ldots, x_k$.

In 1957, Roberts [135, 136] showed that for integers $x, b \geq 2$, where $b \mid x+1$,

$$g(x, x+1, x+b) = \begin{cases} \left(\frac{x+1}{b}\right) x + (b-3)x - 1, & \text{if } x \geq b^2 - 5b + 3; \\ \left(\frac{x+1}{b}\right) (x+b) + (b-4)x - (b+1), & \text{if } x \geq b^2 - 4b + 2, \end{cases} \tag{1.14}$$

(for a generalized form of Eq. (1.14), see Goldberg [55]). Let x_1, x_2, x_3 be positive integers with $\gcd(x_1, x_2, x_3) = 1$. In 1960, Johnson [80] (also see the work of Qiu and Niu [125]) showed that (as an immediate result from Lemma 1.2.6) when $x_3 \geq \frac{x_1}{d}\frac{x_2}{d} - \frac{x_1}{d} - \frac{x_2}{d}$, where $d = \gcd(x_1, x_2)$, the Frobenius number of x_1, x_2, x_3 is

$$g(x_1, x_2, x_3) = \frac{1}{d} x_1 x_2 - x_1 - x_2 + (d-1)x_3. \tag{1.15}$$

In 1962, Brauer and Shockley [19] showed that if $x_1 \mid x_2 + x_3$, then

$$g(x_1, x_2, x_3) = \max\left\{ x_2 \left\lfloor \frac{x_1 x_3}{x_2 + x_3} \right\rfloor - x_1,\ x_3 \left\lfloor \frac{x_1 x_2}{x_2 + x_3} \right\rfloor - x_1 \right\}. \tag{1.16}$$

Suppose none of the x_1, x_2, x_3 can be written as a non-negative integer linear combination of the others, and integers s, t, q, r are determined by $x_3 \equiv s x_2 \pmod{x_1}$, $1 < s < x_1$, $x_3 = s x_2 - t x_1$, $t > 0$, $x_1 = qs + r$, $0 < r < s$, and $x_2 \geq t(q+1)$. Then, in 1977, Selmer [149] (also see Hofmeister [68] and Byrnes [25]) showed that

$$g(x_1, x_2, x_3) = \max\{ (s-1)x_2 + (q-1)x_3, (r-1)x_2 + q x_3 \} - x_1, \tag{1.17}$$

which was later generalized by Hujter and Vizvári [75] (their result is complicated and is omitted here). In 1987, Hujter [74] proved that for any integer $x > 2$,

$$g(x^2, x^2+1, x^2+x) = 2x^3 - 2x^2 - 1. \tag{1.18}$$

In 2006, Fel [45, Eq. (153)] showed that for any sufficiently large integer x,

$$g(2x+1, 2x+3, 4x+3) = 2x^2 + 3x - 1. \tag{1.19}$$

For arbitrary positive integers x_1, x_2, x_3 with $\gcd(x_1, x_2, x_3) = 1$, there is a formula for $g(x_1, x_2, x_3)$ in terms of other constants. Let L_1, L_2, L_3 be the smallest positive integers that satisfy

$$L_1 x_1 = c_{12} x_2 + c_{13} x_3, \quad L_2 x_2 = c_{21} x_1 + c_{23} x_3, \quad L_3 x_3 = c_{31} x_1 + c_{32} x_2, \tag{1.20}$$

for some non-negative integers $c_{ij}, 1 \le i, j \le 3$. Then Denham [37] in 2003 and Ramírez-Alfonsín [130] in 2005 showed by using Hilbert series that for arbitrary positive integers x_1, x_2, x_3 with $\gcd(x_1, x_2, x_3) = 1$, the Frobenius number satisfies

$$g(x_1, x_2, x_3) + \sum_{n=1}^{3} x_n = \begin{cases} \max\{ L_i x_i + c_{jk} x_k, L_j x_j + c_{ik} x_k \}, & \text{if } c_{ij} > 0 \text{ for all } i, j; \\ L_j x_j + L_i x_i, & \text{if some } c_{ij} = 0, \end{cases} \tag{1.21}$$

A restricted version of Eq. (1.21) was found by Johnson [80] in 1960. Johnson also showed that for arbitrary x_1, x_2, x_3, the numbers L_1, L_2, L_3 can be calculated in polynomial time [80].

For $k = 4$ the FP becomes even harder than the case $k = 3$. In 1964, Dulmage and Mendelsohn [39] showed that for any positive integer x,

$$\begin{aligned} g(x, x+1, x+2, x+4) &= (x+1)\left\lfloor \tfrac{x}{4} \right\rfloor + \left\lfloor \tfrac{x+1}{4} \right\rfloor + 2\left\lfloor \tfrac{x+2}{4} \right\rfloor - 1; \\ g(x, x+1, x+2, x+5) &= x\left\lfloor \tfrac{x+1}{5} \right\rfloor + \left\lfloor \tfrac{x}{5} \right\rfloor + \left\lfloor \tfrac{x+1}{5} \right\rfloor + \left\lfloor \tfrac{x+2}{5} \right\rfloor + 2\left\lfloor \tfrac{x+3}{5} \right\rfloor - 1; \\ g(x, x+1, x+2, x+6) &= x\left\lfloor \tfrac{x}{6} \right\rfloor + 2\left\lfloor \tfrac{x}{6} \right\rfloor + 2\left\lfloor \tfrac{x+1}{6} \right\rfloor + 5\left\lfloor \tfrac{x+2}{6} \right\rfloor + \left\lfloor \tfrac{x+3}{6} \right\rfloor \\ &\quad + \left\lfloor \tfrac{x+4}{6} \right\rfloor + \left\lfloor \tfrac{x+5}{6} \right\rfloor - 1. \end{aligned} \tag{1.22}$$

In 1975, Byrnes [26] examined some special cases for $k = 4, 5$ (the formulae are too complicated and are omitted here).

Non-existence theorem of Curtis

In 1990, Curtis [34] proved that no simple formula for the Frobenius number exists when $k = 3$. Here by "simple formula" we mean a finite set of polynomials.

Theorem 1.2.3. *[34] Let*

$$A = \left\{ (x_1, x_2, x_3) \in \mathbb{N}^3 : x_1 < x_2 < x_3,\ x_1, x_2 \text{ are primes},\ \gcd(x_1, x_2, x_3) = 1 \right\}. \tag{1.23}$$

Then there is no non-zero polynomial $f \in \mathbb{C}[\![z_1, z_2, z_3, z_4]\!]$ such that

$$f\big(x_1, x_2, x_3, g(x_1, x_2, x_3)\big) = 0 \tag{1.24}$$

for every $(x_1, x_2, x_3) \in A$.

Corollary 1.2.4. *[34] There is no finite set of polynomials $f_1, f_2, \ldots, f_k$ in three variables such that for each choice of x_1, x_2, x_3 with $\gcd(x_1, x_2, x_3) = 1$ there exists some f_i computing the Frobenius number $g(x_1, x_2, x_3)$.*

Proof. Proof by contradiction. Let $f(z_1, z_2, z_3, z_4) = \prod_{i=1}^{k} \big(f_i(z_1, z_2, z_3) - z_4\big)$. Then f satisfies (1.24). □

Every basis x_1, x_2, x_3 can be padded with some dummy integers $x_4, \ldots, x_k$ that are already in $\langle x_1, x_2, x_3 \rangle$, giving a basis of an arbitrary size k. The Frobenius number of the new basis remains the same as the old:

$$g(x_1, x_2, x_3, x_4, \ldots, x_k) = g(x_1, x_2, x_3). \tag{1.25}$$

Since there are simple formulae providing upper bounds for $g(x_1, x_2, x_3)$ (see §1.2.2), the dummy integers $x_4, \ldots, x_k$ can be given explicitly by polynomials, and thus Corollary 1.2.4 also eliminates the possibility of the existence of a simple formula for the Frobenius number for any fixed $k \geq 3$.

Implicit formulae

There are several implicit formulae for the Frobenius number, which either depend on other characteristics or are recursive. Some of them are very useful in the sense that good bounds or algorithms can be derived from them.

For example, in 1962, Brauer and Shockley gave the following lemma [19, 68].

Lemma 1.2.5. *[19] Let $x_1, x_2, \ldots, x_k$ be positive integers. Given integers y and l, $0 \leq l < y$, let t_l $(= t_l(y))$ denote the smallest integer such that $t_l \in \langle x_1, x_2, \ldots, x_k \rangle$ and $t_l \equiv l \pmod{y}$. Then for every non-zero choice of $y \in \langle x_1, x_2, \ldots, x_k \rangle$,*

$$g(x_1, x_2, \ldots, x_k) = \max_{0 \leq l \leq y-1} t_l - y. \tag{1.26}$$

In 1960, Johnson [80] gave the following lemma for $k = 3$, which was then generalized by Brauer and Shockley [19].

Lemma 1.2.6. *[80, 19] Let $x_1, \ldots, x_k$ be positive integers with $\gcd(x_1, \ldots, x_k) = 1$. Then*

$$g(x_1, x_2, \ldots, x_k) = dg\Big(\frac{x_1}{d}, \frac{x_2}{d}, \ldots, \frac{x_{k-1}}{d}, x_k\Big) + (d-1)x_k, \tag{1.27}$$

where $d = \gcd(x_1, x_2, \ldots, x_{k-1})$.

Example 1.2.7. One can compute $g(6, 9, 20)$ in the Chicken McNuggets problem by either of the following procedures.

(a) By Lemma 1.2.6, we have

$$g(6, 9, 20) = 3g(2, 3, 20) + 2 \cdot 20 = 3g(2, 3) + 40 = 3 + 40 = 43, \tag{1.28}$$

(b) By Lemma 1.2.5, we have $g(6, 9, 20) = 63 - 20 = 43$, where

$$63 = \max\{\, 0, 21, 42, 63, 24, 45, 6, 27, 48, 9, 30, 51, 12, 33, 54, 15, 36, 57, 18, 39 \,\}. \tag{1.29}$$

1.2.2 Bounds on the Frobenius number

Since finding a general formula to exactly compute the Frobenius number for a basis is infeasible (see §1.2.5), finding formulae for good upper bounds and lower bounds on the Frobenius number becomes another interesting aspect of the FP.

A quadratic upper bound $g(x_1, x_2, \ldots, x_k) = O(x_k^2)$ for the basis $x_1, x_2, \ldots, x_k$ follows immediately from the following theorem, which was proved by Schur in his lectures in 1935 (according to Brauer [17]).

Theorem 1.2.8. *Let $x_1, x_2, \ldots, x_k$ be a basis with $\gcd(x_1, x_2, \ldots, x_k) = 1$. Then*

$$g(x_1, x_2, \ldots, x_k) \le x_1 x_k - x_1 - x_k. \tag{1.30}$$

Proof. (Rewritten from a proof of Brauer [17]) Proof by induction. For $k = 2$, $g(x_1, x_2) = x_1 x_2 - x_1 - x_2$ follows from Theorem 1.2.1. Now we assume (1.30) is true for all bases of size $k - 1$. Let $\gcd(x_1, x_3, \ldots, x_k) = d$. Then $\gcd(x_2, d) = 1$, and the equation

$$g(x_1, x_2, \ldots, x_k) \equiv x_2 x \pmod{d} \tag{1.31}$$

has a unique solution $0 \le x = c_2 < d$. Furthermore, $g(x_1, x_2, \ldots, x_k) - x_2 c_2$ is not in $\langle x_1, x_3, \ldots, x_k \rangle$. Since $\gcd(\frac{x_1}{d}, \frac{x_3}{d}, \ldots, \frac{x_k}{d}) = 1$, by the induction hypothesis, we have

$$\begin{aligned}
&g(x_1, x_2, \ldots, x_k) - (x_1 x_k - x_1 - x_k) \\
&= x_2 c_2 + d \cdot \frac{1}{d}\big(g(x_1, x_2, \ldots, x_k) - x_2 c_2\big) - (x_1 x_k - x_1 - x_k) && (1.32)\\
&\le x_2 c_2 + d g\Big(\frac{x_1}{d}, \frac{x_3}{d}, \ldots, \frac{x_k}{d}\Big) - (x_1 x_k - x_1 - x_k) && (1.33)\\
&\le x_2 c_2 + d\Big(\frac{x_1 x_k}{d^2} - \frac{x_1}{d} - \frac{x_k}{d}\Big) - (x_1 x_k - x_1 - x_k) && (1.34)\\
&= \big(d x_2 c_2 - x_1 x_k (d-1)\big)/d && (1.35)\\
&\le 0. && (1.36)
\end{aligned}$$

Therefore, $g(x_1, x_2, \ldots, x_k) \le x_1 x_k - x_1 - x_k$. □

The asymptotic upper bound $g(x_1, x_2, \ldots, x_k) = O(x_k^2)$ is tight for fixed k (= 2). When $k = 2$, by Theorem 1.2.1, we know $g(x, x+1) = x^2 - x - 1 = \Theta(x^2)$.

In 1942, Brauer [17] gave another upper bound as in the following theorem, which is the best possible upper bound under certain conditions [17, 18, 117].

Theorem 1.2.9. *[17] Let $x_1, \ldots, x_k$ be positive integers with $\gcd(x_1, \ldots, x_k) = 1$, and let $d_i = \gcd(x_1, x_2, \ldots, x_i)$ for $1 \le i \le k$. Then*

$$g(x_1, x_2, \ldots, x_k) \le x_2 \frac{d_1}{d_2} + x_3 \frac{d_2}{d_3} + \cdots + x_k \frac{d_{k-1}}{d_k} - \sum_{i=1}^{k} x_i. \tag{1.37}$$

There are other upper bounds for the Frobenius number in the literature. Let $x_1, x_2, \ldots, x_k$ be a basis with $\gcd(x_1, x_2, \ldots, x_k) = 1$, where $x_i \neq x_j$ for $i \neq j$. In 1972, Erdös and Graham [42] showed that

$$g(x_1, x_2, \ldots, x_k) \leq 2x_{k-1} \left\lfloor \frac{x_k}{k} \right\rfloor - x_k, \tag{1.38}$$

which is tight for $k = 2$. In 1977, a similar tight bound was found by Selmer [149] for the case $x_1 \geq k$, as follows:

$$g(x_1, x_2, \ldots, x_k) \leq 2x_k \left\lfloor \frac{x_1}{k} \right\rfloor - x_1. \tag{1.39}$$

In 1975, Vitek [169] showed another bound for $k \geq 3$ (also see Lewin's work [97])

$$g(x_1, x_2, \ldots, x_k) < \left\lfloor \frac{(x_2 - 1)(x_k - 2)}{2} \right\rfloor. \tag{1.40}$$

In 1990, Dixmier [38] proved a conjecture of Erdös and Graham [43, p. 86] that

$$g(x_1, x_2, \ldots, x_k) \leq \frac{x_k^2}{k-1} = O\Big(\frac{x_k^2}{k}\Big) \tag{1.41}$$

is tight in the sense that for any integer n and $2 \leq k < n$,

$$\left\lfloor \frac{n-2}{k-1} \right\rfloor (n-k+1) - 1 \leq \max_{1 \leq x_1, x_2, \ldots, x_k \leq n} g(x_1, x_2, \ldots, x_k) \leq \left(\left\lceil \frac{n-1}{k-1} \right\rceil - 1 \right) n - 1. \tag{1.42}$$

There are works on upper bounds for the Frobenius number in terms of other measures. For example, in 1992, Chrząstowski-Wachtel (according to the paper [126]) obtained the following upper bound:

$$g(x_1, x_2, \ldots, x_k) \leq (k-1) \operatorname{lcm}(x_1, x_2, \ldots, x_k). \tag{1.43}$$

There are also upper bounds for fixed k $(= 3)$. In 1957, Robert [136] showed that for $0 < a < b$, $\gcd(a, b) = 1$, and $m \geq 2$:

$$g(m, m+a, m+b) \leq m\Big(b - 2 + \left\lfloor \frac{m}{b} \right\rfloor\Big) + ab - a - b. \tag{1.44}$$

In 2002, Beck, Diaz and Robins [8] showed that

$$g(x_1, x_2, x_3) \leq \frac{1}{2}\Big(\sqrt{x_1 x_2 x_3 (x_1 + x_2 + x_3)} - x_1 - x_2 - x_3\Big). \tag{1.45}$$

There are some results on lower bounds for the Frobenius number. Let $x_1, \ldots, x_k$ be a basis with $\gcd(x_1, \ldots, x_k) = 1$. A trivial lower bound is

$$g(x_1, x_2, \ldots, x_k) \geq x_1 - 1 = \Omega(x_1), \tag{1.46}$$

which can be achieved by the example of consecutive integers in (1.72) on page 16. In 1994, Davison [35] gave the following tight lower bound for $k = 3$ (also see Hujter's work [74]):

$$g(x_1, x_2, x_3) \geq \sqrt{3x_1x_2x_3} - x_1 - x_2 - x_3. \tag{1.47}$$

In 1982, Hujter [73] gave the following lower bound for any arbitrary k (also see Killingbergtrø's work [85]):

$$g(x_1, x_2, \ldots, x_k) \geq \Big(\frac{k-1}{k}\Big)\big((k-1)!x_1x_2\cdots x_k\big)^{\frac{1}{k-1}} - \sum_{i=1}^{k} x_i = \Omega\Big(kx_1^{\frac{k}{k-1}}\Big). \tag{1.48}$$

1.2.3 Variations on the FP

Many variations on the FP are of interest. As usual, let $x_1, x_2, \ldots, x_k$ be positive integers with $\gcd(x_1, x_2, \ldots, x_k) = 1$ and $\langle x_1, x_2, \ldots, x_k\rangle$ be the set of all integers that can be represented as non-negative integer linear combinations of the given integers $x_1, x_2, \ldots, x_k$. Instead of asking for the Frobenius number $g(x_1, x_2, \ldots, x_k)$, the largest integer that is not in $\langle x_1, x_2, \ldots, x_k\rangle$, there are several other interesting problems.

One such problem is to find the minimal integer $n \in \langle x_1, x_2, \ldots, x_k\rangle$ such that $m \in \langle x_1, x_2, \ldots, x_k\rangle$ for all $m \geq n$. Obviously, $n = g(x_1, x_2, \ldots, x_k) + 1$. This integer n is called the *conductor* of $x_1, x_2, \ldots, x_k$. Sometimes a formula involving the conductor has a simpler form (or proof) than the equivalent formula involving the Frobenius number. For example, while the Frobenius number of integers x_1, x_2 is $x_1x_2 - x_1 - x_2$, the conductor of x_1, x_2 is simply $(x_1 - 1)(x_2 - 1)$.

Another variation on the FP is to find the number of all positive integers that are not in $\langle x_1, x_2, \ldots, x_k\rangle$.

Number $h(x_1, x_2, \ldots, x_k)$ of positive integers not in $\langle x_1, x_2, \ldots, x_k\rangle$

In 1882, James J. Sylvester (1814–1897) discussed the total number of those positive integers that are not in $\langle x_1, x_2, \ldots, x_k\rangle$ [164, p. 134], and later this problem appeared in the *Educational Times* as a recreational question asking for a formula for $k = 2$, which was then answered by Curran Sharp [165]. Let $h(x_1, x_2, \ldots, x_k)$ denote the number of positive integers that are not in $\langle x_1, x_2, \ldots, x_k\rangle$.

Theorem 1.2.10. *[164, 165] Let x_1, x_2 be positive integers with $\gcd(x_1, x_2) = 1$. Then the number of positive integers that are not in $\langle x_1, x_2\rangle$ is*

$$h(x_1, x_2) = \frac{1}{2}(x_1 - 1)(x_2 - 1). \tag{1.49}$$

Proof. (Rewritten from Sharp's solution [165]) Consider the coefficient of the term z^n, $0 \le n \le 2x_1x_2$, in the expansion of the following product:

$$f(z) = (1 + z^{x_1} + z^{2x_1} + \cdots + z^{x_1x_2})(1 + z^{x_2} + z^{2x_2} + \cdots + z^{x_2x_1}). \tag{1.50}$$

Suppose $c_1x_1 + c_2x_2 = n = c_1'x_1 + c_2'x_2$, $0 \le c_1, c_1' \le x_2$ and $0 \le c_2, c_2' \le x_1$. Then

$$(c_1 - c_1')x_1 = (c_2' - c_2)x_2. \tag{1.51}$$

We have either $c_1 = c_1', c_2 = c_2'$ or $c_1 = x_2, c_2' = x_1, c_1' = c_2' = 0$. So each coefficient of a non-zero term is 1 except that of $z^{x_1x_2}$, which is 2. By symmetry, the coefficient of the term z^n is equal to the coefficient of the term $z^{2x_1x_2-n}$. Let $z = 1$. Then the number of non-zero terms between z^0 and $z^{x_1x_2}$ is

$$\frac{1}{2}\big(f(1) - 2\big) + 1 = \frac{1}{2}(x_1 + 1)(x_2 + 1). \tag{1.52}$$

In addition, each non-zero term between z^0 and $z^{x_1x_2}$ corresponds to an integer in $\langle x_1, x_2\rangle$, and the converse is also true. By Theorem 1.2.1, we have

$$g(x_1, x_2) = x_1x_2 - x_1 - x_2 < x_1x_2. \tag{1.53}$$

So the number of zero terms between z^0 and $z^{x_1x_2}$ is exactly $h(x_1, x_2)$, which is

$$h(x_1, x_2) = (x_1x_2 + 1) - \frac{1}{2}(x_1 + 1)(x_2 + 1) = \frac{1}{2}(x_1 - 1)(x_2 - 1). \tag{1.54}$$

□

Example 1.2.11. In Example 1.1.7 we saw that there are four integers $1, 2, 4, 7$ in total that are not in $\langle 3, 5\rangle$. To compute $h(3, 5)$, we apply Formula (1.49) and get

$$h(3, 5) = \frac{1}{2}(3 - 1)(5 - 1) = 4. \tag{1.55}$$

Nijenhuis and Wilf [117] compared the two values $g(x_1, \ldots, x_k)$, the Frobenius number, and $h(x_1, \ldots, x_k)$, the number of positive integers not in $\langle x_1, \ldots, x_k\rangle$, and gave a tight lower bound on $h(x_1, \ldots, x_k)$ in terms of $g(x_1, \ldots, x_k)$. The conditions under which that lower bound is achieved were also discussed [117, 88].

Theorem 1.2.12. *[117] Let $x_1, \ldots, x_k$ be positive integers with* $\gcd(x_1, \ldots, x_k) = 1$. *Then*

$$\frac{1}{2}\big(g(x_1, x_2, \ldots, x_k) + 1\big) \le h(x_1, x_2, \ldots, x_k) \le g(x_1, x_2, \ldots, x_k). \tag{1.56}$$

Both upper and lower bounds given in (1.56) are tight. When the basis $x_1, \ldots, x_k$ is $p, p + 1, p + 2, \ldots, 2p - 1$, the equality of the upper bound in (1.56) is achieved, since in that case $h(x_1, x_2, \ldots, x_k) = g(x_1, x_2, \ldots, x_k) = p - 1$. When $k = 2$, the equality of the lower bound in (1.56) is achieved as follows:

$$\frac{1}{2}\big(g(x_1, x_2) + 1\big) = \frac{1}{2}(x_1x_2 - x_1 - x_2 + 1) = \frac{1}{2}(x_1 - 1)(x_2 - 1) = h(x_1, x_2). \tag{1.57}$$

Since $g(x_1,\ldots,x_k)=O(x_k^2)$ and $g(x_1,\ldots,x_k)=\Omega(x_1)$, then we know

$$h(x_1,\ldots,x_k)=O(x_k^2),\quad \text{and}\quad h(x_1,\ldots,x_k)=\Omega(x_1). \tag{1.58}$$

Killingbergtrø [85] studied $h(x_1,x_2,\ldots,x_k)$ by using a geometric method and presented the following lower bound:

$$h(x_1,x_2,\ldots,x_k)\ge \Big\lfloor \Big(\frac{k-1}{k}\Big)\big((k-1)!x_1x_2\cdots x_k\big)^{\frac{1}{k-1}}-\sum_{i=1}^{k}x_k-1\Big\rfloor. \tag{1.59}$$

Sum of positive integers not in $\langle x_1,x_2,\ldots,x_k\rangle$

A further question is to ask what is the sum of all positive integers that are not in $\langle x_1,x_2,\ldots,x_k\rangle$. It is straightforward that the sum is bounded by

$$\sum_{n=1}^{h(x_1,\ldots,x_k)} n\le \sum_{\left\{\,n\in\mathbb{N}\,:\,n\notin\langle x_1,x_2,\ldots,x_k\rangle\,\right\}} n\le \sum_{\substack{n=g(x_1,\ldots,x_k)\\ -h(x_1,\ldots,x_k)+1}}^{g(x_1,\ldots,x_k)} n=O(x_k^4). \tag{1.60}$$

In 1993, Brown and Shiue [21, 66] discovered the following simple formula for $k=2$:

$$\sum_{\left\{\,n\in\mathbb{N}\,:\,n\notin\langle x_1,x_2\rangle\,\right\}} n=\frac{1}{12}(x_1-1)(x_2-1)(2x_1x_2-x_1-x_2-1). \tag{1.61}$$

Rödseth [139] generalized Eq. (1.61) to any arbitrary power $m\ge 1$ as follows:

$$\begin{aligned}&\sum_{\left\{\,n\in\mathbb{N}\,:\,n\notin\langle x_1,x_2\rangle\,\right\}} n^{m-1}\\ &\quad=\frac{1}{m(m+1)}\sum_{i=0}^{m}\sum_{j=0}^{m-i}\binom{m+1}{i}\binom{m+1-i}{j}\mathsf{B}_i\mathsf{B}_j x_1^{m-j}x_2^{m-i}-\frac{1}{m}\mathsf{B}_m,\end{aligned} \tag{1.62}$$

where the B's are the Bernoulli numbers, which are defined by the following expansion

$$\frac{x}{e^x-1}=\mathsf{B}_0+\frac{\mathsf{B}_1}{1!}x+\frac{\mathsf{B}_2}{2!}x^2+\frac{\mathsf{B}_3}{3!}x^3+\cdots. \tag{1.63}$$

The denumerant $d(n;x_1,x_2,\ldots,x_k)$ — the number of partitions

For an integer $n\in\langle x_1,x_2,\ldots,x_k\rangle$, the expression of n as a non-negative integer linear combination of $x_1,x_2,\ldots,x_k$ may not be unique. One question to ask is: in how many different ways can m be written as $m=c_1x_1+c_2x_2+\cdots c_kx_k$? A more general problem arises when the basis $x_1,x_2,\ldots$ is infinite, for example, $\mathbb{N}$.

The general partition problem for the basis $\mathbb{N}$ is an old problem. Leonhard P. Euler (1707–1783) established an entire theory in the late 1700's [44] (100 years earlier than Frobenius's work!) by means of generating functions. In order to study the partition problem for the basis $\mathbb{N}$, he considered the series

$$1+\sum_{n=1}^{\infty} p(n)z^n = \prod_{n=1}^{\infty} \frac{1}{1-z^n} = (1+z+z^2+\cdots)(1+z^2+z^4+\cdots)(1+z^3+z^6+\cdots)\cdots. \tag{1.64}$$

Hardy and Ramanujan [61, p. 79] proved the following asymptotic formula for $p(n)$:

$$p(n) \sim \frac{1}{4\sqrt{3}n} e^{\pi\sqrt{\frac{2n}{3}}}. \tag{1.65}$$

Sylvester [164] defined the *denumerant* $d(n; x_1, x_2, \ldots, x_k)$ by the number of different non-negative integer linear combinations of n in the basis $x_1, x_2, \ldots, x_k$ and studied the series

$$1 + \sum_{n=1}^{\infty} d(n; x_1, x_2, \ldots, x_k)z^n = \frac{1}{(1-z^{x_1})(1-z^{x_2})\cdots(1-z^{x_k})}, \tag{1.66}$$

where to decide whether $d(n; x_1, x_2, \ldots, x_k) > 0$ in general is equivalent to the well-studied integer knapsack problem. A useful variation on Eq. (1.66) was discovered by Özlük and Sertöz [152] and was used to obtain Eq. (1.61).

Assuming $\gcd(x_1, \ldots, x_k) = 1$, Schur [147] gave the following asymptotic estimate of denumerants as $n \to \infty$,

$$d(n; x_1, x_2, \ldots, x_k) \sim \frac{n^{k-1}}{x_1 x_2 \cdots x_k (k-1)!}. \tag{1.67}$$

For $k = 2$, there is an exact formula for denumerants, which has been rediscovered several times [123, 151, 167, 8, 20], namely,

$$d(n; x_1, x_2) = \frac{n + c_1 x_1 + c_2 x_2}{x_1 x_2} - 1, \tag{1.68}$$

where c_1, c_2 are defined for each n such that

$$c_1 x_1 \equiv -n \pmod{x_2}, \quad 1 \le c_1 \le x_2, \quad c_2 x_2 \equiv -n \pmod{x_1}, \quad 1 \le c_2 \le x_1. \tag{1.69}$$

Positive integer linear combinations

The FP is about the set $\langle x_1, \ldots, x_k \rangle$ of all non-negative integer linear combinations of $x_1, \ldots, x_k$. What if we consider the *positive integer linear combination* of $x_1, \ldots, x_k$ instead? Let $f(x_1, \ldots, x_k)$ denote the largest integer that cannot be written as a positive integer linear combination of $x_1, \ldots, x_k$. Then as mentioned by Brauer [17] the two problems are strongly related.

Theorem 1.2.13. *[17]* $f(x_1, x_2, \ldots, x_k) = g(x_1, x_2, \ldots, x_k) + x_1 + x_2 + \cdots + x_k$.

Sometimes a formula involving $f(x_1, x_2, \ldots, x_k)$ has a simpler form or an easier proof than the equivalent one involving $g(x_1, x_2, \ldots, x_k)$. For example, the formulae $f(x_1, x_2, \ldots, x_k) = df\left(\frac{x_1}{d}, \frac{x_2}{d}, \ldots, \frac{x_{k-1}}{d}, x_k\right)$ and $f(x_1, x_2) = x_1x_2$ are shorter than their counterparts $g(x_1, x_2, \ldots, x_k) = dg\left(\frac{x_1}{d}, \frac{x_2}{d}, \ldots, \frac{x_{k-1}}{d}, x_k\right) + (d-1)x_k$ and $g(x_1, x_2) = x_1x_2 - x_1 - x_2$.

Extending the basis

Let $x_1, x_2, \ldots, x_k$ be a basis with $\gcd(x_1, x_2, \ldots, x_k) = 1$. When we include a new integer x_{k+1}, the extended basis satisfies $\langle x_1, \ldots, x_k\rangle \subseteq \langle x_1, \ldots, x_k, x_{k+1}\rangle$. So, by definition, the Frobenius number of the new basis cannot increase:

$$g(x_1, x_2, \ldots, x_k, x_{k+1}) \leq g(x_1, x_2, \ldots, x_k). \tag{1.70}$$

The equality in (1.70) is achieved when $x_{k+1} \in \langle x_1, x_2, \ldots, x_k\rangle$. If we require that $x_{k+1} \notin \langle x_1, x_2, \ldots, x_k\rangle$, then one question is: can the equality in (1.70) be attained?

In 1970, Mendelsohn [111] showed that for $k = 2$ the equality in (1.70) cannot be attained. By applying Lemma 1.2.6, we know for $k = 2$ the equality in (1.70) can be attained only when $x_3 \in \langle x_1, x_2\rangle$.

Theorem 1.2.14. *[111] Let* $x_1, x_2, x_3 \in \mathbb{N}$. *If* $\gcd(x_1, x_2) = 1$, $x_3 \notin \langle x_1, x_2\rangle$, *then*

$$g(x_1, x_2, x_3) < g(x_1, x_2). \tag{1.71}$$

In 1977, Selmer [149] also studied the problem of extending a basis. By considering an arithmetic sequence, he showed a more general result on this problem.

Theorem 1.2.15. *[149] For any integer* $l \geq 2$, *there exist an integer* k *and a basis* $x_1, x_2, \ldots, x_k$ *with* $\gcd(x_1, x_2, \ldots, x_k) = 1$ *such that when extending with positive integers* x_{k+1}, x_{k+2}, $\ldots$, x_{k+l}, *where* $x_{k+i} \notin \langle x_1, \ldots, x_k, \ldots, x_{k+i-1}\rangle$ *for* $1 \leq i \leq l$, *the Frobenius number of the basis does not change:* $g(x_1, x_2, \ldots, x_k, \ldots, x_{k+l}) = g(x_1, x_2, \ldots, x_k)$.

Coverage of Frobenius numbers

Finally, one question about the FP is whether every positive integer is a Frobenius number for some basis. A recent paper of Rosales, García-Sánchez and García-García [143] showed that every positive integer is a Frobenius number for some basis $\{ x_1, x_2, x_3 \}$.

Theorem 1.2.16. *[143] Let* n *be a positive integer. Then there exist three positive integers* x_1, x_2, x_3 *such that* $g(x_1, x_2, x_3) = n$.

Let $k \geq 3$ be a fixed integer. For any positive integer n, there are integers $x_1, x_2, x_3 \in \mathbb{N}$ such that $g(x_1, x_2, x_3) = n$. By padding with additional integers $x_4, \ldots, x_k \in \langle x_1, x_2, x_3 \rangle$, we know that $g(x_1, x_2, \ldots, x_k) = g(x_1, x_2, x_3) = n$. So any positive integer n is also the Frobenius number of k positive integers for $k \geq 3$. Furthermore, $k = 3$ is the least possible.

Proposition 1.2.17. *[143] There exist infinitely many positive integers that are not the Frobenius number of any pair of positive integers.*

Proof. If $\gcd(x_1, x_2) = 1$, then x_1, x_2 cannot both be even. So, the Frobenius number of two positive integers $g(x_1, x_2) = x_1x_2 - x_1 - x_2$ is always odd. □

1.2.4 The FP in special sequential bases

There are also studies on the FP in special cases, where the integers in a basis belong to a special type of sequence.

Basis is an arithmetic sequence

The sequence of k consecutive positive integers that are in the same residue class, $x, x + d, x + 2d, \ldots, x + (k - 1)d$, is called an *arithmetic sequence*. A special case of the FP is when the integers in a basis constitute an arithmetic sequence. In particular, one can easily verify that

$$g(x, x + 1, x + 2, \ldots, 2x - 1) = x - 1. \tag{1.72}$$

In 1942, Brauer (and Schur) [17] found the following formula for the Frobenius number of k consecutive integers:

$$g\big(x, x + 1, x + 2, \ldots, x + k - 1\big) = \left\lfloor \frac{x - 2}{k - 1} \right\rfloor x + x - 1, \tag{1.73}$$

which was generalized by Roberts [135] in 1956 (also see the simpler proof of Bateman [6]) as follows:

$$g\big(x, x + d, x + 2d, \ldots, x + (k - 1)d\big) = \left\lfloor \frac{x - 2}{k - 1} \right\rfloor x + dx - d. \tag{1.74}$$

Basis is an almost arithmetic sequence

A sequence $x_1, x_2, \ldots, x_k$ is called an *almost arithmetic sequence* if excluding either x_1 or x_k, the remaining part is an arithmetic sequence. The Frobenius number of almost arithmetic sequences was introduced by Lewin [99, 98]. In 1977, Selmer [149]

gave the following formula (also see Rödseth's work [138]) for the Frobenius number of integers constituting an almost arithmetic sequence:

$$g\big(x, hx+d, hx+2d, \ldots, hx+(k-1)d\big) = \left\lfloor \frac{x-2}{k-1} \right\rfloor hx + (d+h-1)x - d. \quad (1.75)$$

In 1979, Rödseth [138] found the formula (also see Shao's work [158]):

$$\begin{aligned} &g\big(x, x+d, \ldots, x+(k-2)d, x+Kd\big) \\ &\qquad = (x+Kd)\alpha - d + \max\left\{ x\left\lfloor \frac{\beta-2}{k-2} \right\rfloor + d\beta, x\left\lfloor \frac{K-2}{k-2} \right\rfloor - x \right\}, \quad (1.76) \end{aligned}$$

where $\gcd(x,d) = 1, K \geq k-1$, and $x = \alpha K + \beta, 0 \leq \beta < K$ such that $\beta = 0$ or $\alpha + d \geq \lfloor \frac{K-\beta-1}{k-2} \rfloor$.

Basis is a sequence of other type

In 1977, Selmer [149] considered sequences where the differences of terms constitute geometric sequences (also see Hofmeister [68] and Goldberg [55] for similar discussion), and gave the following formula:

$$g(x, x+1, x+2, x+2^2, \ldots, x+2^{k-2}) = (x+1)\Big(\frac{x}{2^{k-2}}\Big) + \sum_{i=0}^{k-3} 2^i \left\lfloor \frac{x+2^i}{2^{k-2}} \right\rfloor + (k-4)x - 1. \quad (1.77)$$

In 1982, Hujter [73] considered the same type of sequence and gave the following formula:

$$g(x^{k-1}, x^{k-1}+1, x^{k-1}+x, \ldots, x^{k-1}+x^{k-2}) = (k-1)(x-1)x^{k-1} - 1, \quad (1.78)$$

In 1987, Boros [16] considered a similar type of sequence, in which the differences of consecutive terms except the first term constitute geometric sequences.

In 2007, Einstein, Lichtblau, Strzebonski and Wagon [40] presented a method to produce formulae for the Frobenius number of a *quadratic sequence* of small length. For example, for $x \geq 2$,

$$g(9x, 9x+1, 9x+4, 9x+9) = 9x^2 + 18x - 2. \quad (1.79)$$

In 2008, Ong and Ponomarenko [119] discussed the Frobenius number of a *geometric sequence*, and showed that for $\gcd(x,y) = 1$,

$$g(x^k, x^{k-1}y, x^{k-2}y^2, \ldots, y^k) = \frac{xy(y^k - x^k) - (y^{k+1} - x^{k+1})}{y-x}. \quad (1.80)$$

1.2.5 Algorithm for and computational complexity of the FP

There are several known algorithms to compute the Frobenius number for a small fixed k. For $k = 3$, let x_1, x_2, x_3 be a basis with $\gcd(x_1, x_2, x_3) = 1$ and $x_i \neq x_j$ for $i \neq j$. In 1960, Johnson [80] outlined an algorithm idea to compute $g(x_1, x_2, x_3)$ without providing details. In 1962, Brauer and Shockley [19] presented a similar method that costs $O(x_1 + \log x_2)$ time to compute $g(x_1, x_2, x_3)$. In 1978, Selmer and Beyer [150] proposed an algorithm using simple continued fractions to compute $g(x_1, x_2, x_3)$; it is not easy to implement. Later, Rödseth [137] simplified their method and developed an algorithm that runs in $O(x_1 + \log x_2)$ time in the worst case, but runs faster in practice. In 1994, Davison [35] developed an algorithm that runs in $O(\log x_2)$ time in the worst case, based on modifications of the algorithms of Selmer-Beyer and Rödseth. In 2000, Killingbergtrø [85] developed an algorithm that converts the FP for $k \leq 4$ into a geometric problem of cube-figures, which can be solved in polynomial time for $k = 3$; see Owens [121] for a similar approach. So far, the algorithm given by Greenberg [59] in 1988 is the fastest algorithm to compute $g(x_1, x_2, x_3)$ (according to the experiments in [9]), which runs in $O(\log x_1)$ time in the worst case.

Input: a basis $x_1, x_2, \ldots, x_k$.
Output: the Frobenius number $g(x_1, x_2, \ldots, x_k)$.

```
1  form a circle of x_k lights as 0, 1, ..., x_k − 1, with only 0 turned on ;
2  repeat
3  |  sweeping each light ;
4  |  recording the number s(n) of visiting for each light n ;
5  |  if any of the k lights n − x_i is on for 1 ≤ i ≤ k then
6  |  |  turn n on (or keep n on if it is already on) ;
7  |  end
8  until there are x_1 consecutive lights, all of which are on ;
9  let r be the last visited light that is off ;
10 g(x_1, x_2, ..., x_k) ← r + (s(r) − 1)x_k ;
```

Figure 1.1: Wilf's algorithm to compute $g(x_1, x_2, \ldots, x_k)$

There are also several general algorithms for the FP for an arbitrary fixed k. For example, in 1964, Heap and Lynn [63, 64] developed an algorithm by converting the FP into the problem of computing the index of primitivity of a matrix, which runs in $O(x_k^3 \log^2 x_k)$ time and uses $O(x_k^3)$ space. In 1978, Wilf [172] (also see Huang's work [72] in 1981) gave an algorithm, which runs in $O(kx_k^2)$ time and uses $O(x_k)$ space; see Figure 1.1. In 1980, Greenberg [58] developed an algorithm by similar approach; the correctness of both Wilf's and Greenberg's algorithms is based on Lemma 1.2.5. In the same year, Nijenhuis [118] developed an algorithm to compute the Frobenius number by finding a minimal path in a certain weighted

graph, which runs in $O(x_1(k+\log x_1))$ time. In 1992, Kannan [81, 82], by studying a geometric problem about covering radius, presented an algorithm to compute the Frobenius number for any fixed k that runs in time bounded by a polynomial in $\log x_k$. Kannan's algorithm, however, runs in doubly-exponential time in the dimension k and is not easy to implement. In 1993, Scarf and Shallcross [146] developed an algorithm that converts the FP into the geometric problem of finding a maximal lattice-free body. In 2005, Beihoffer, Hendry, Nijenhuis and Wagon [9] showed a fast algorithm that can handle cases where $k = 10, x_1 = 10^7$.

In 1996, Ramírez-Alfonsín [129] proved that in the general case of an arbitrary k the FP is NP-hard by giving a Turing reduction from the *integer knapsack problem* (IKP), which is NP-complete [103].

Problem 1.2.18 (*IKP*)**.** *Given $k+1$ positive integers $x_1, x_2, \ldots, x_k$ and t, do there exist non-negative integers $c_1, c_2, \ldots, c_k$ such that $\sum_{i=1}^{k} c_i x_i = t$?*

Theorem 1.2.19. *[129] The FP is NP-hard under Turing reductions.*

Theorem 1.2.20. *These problems are NP-hard under Karp reductions:*

(a) Given integers $x_1, x_2, \ldots, x_k$ and n, compute $d(n; x_1, x_2, \ldots, x_k)$;

(b) Given integers $x_1, x_2, \ldots, x_k$, compute $h(x_1, x_2, \ldots, x_k)$.

Proof. (a) Let $x_1, x_2, \ldots, x_k$ and t be an arbitrary instance of IKP. Then

$$d(t; x_1, x_2, \ldots, x_k) > 0 \tag{1.81}$$

if and only if $t \in \langle x_1, x_2, \ldots, x_k \rangle$, which means the IKP has a solution.

(b) Let $x_1, x_2, \ldots, x_k$ and t be an arbitrary instance of IKP. Then

$$h(x_1, x_2, \ldots, x_k) = h(x_1, x_2, \ldots, x_k, t) \tag{1.82}$$

if and only if $t \in \langle x_1, x_2, \ldots, x_k \rangle$, which means the IKP has a solution. □

The NP-hardness result for $h(x_1, x_2, \ldots, x_k)$ is due to Pawel Gawrychowski [52].

1.3 Applications and related problems of the FP

1.3.1 Sylver coinage

Game 1.3.1 (*Sylver coinage*)**.** *In this game the players name numbers in turn. The number to be named must be a positive integer that has not yet been named. In addition, no one is allowed to name a number which is a multiple of a named number, nor to name a number which can be made up by adding together multiples of named numbers. The winner is the player who names the last number except 1, and the loser is the player who names 1. When 1 is named, the game is over since no other integer can be named.*

Sylver coinage was invented by Conway [11, Chap. 18] and the game cannot go on forever due to Sylvester's result (Theorem 1.2.10).

Theorem 1.3.2. *The Sylver coinage game will always stop after a finite number of turns.*

So far, there is no efficient algorithm to produce a winning strategy from an arbitrary opening of a game with named integers being $x_1, x_2, \ldots, x_k$. Some cases with particular openings have been studied [11, 161].

1.3.2 Quadratic residues

Let p be a positive integer. An integer n, $0 < n < p$, is called a *quadratic residue* of p if the equation $x^2 \equiv n \pmod{p}$ has a solution. Otherwise n, $0 < n < p$, is called a *quadratic non-residue*. It is well-known that when $p \geq 3$ is a prime then there are exactly $(p-1)/2$ quadratic residues and quadratic non-residues, respectively (for example, see the textbook of Hardy and Wright [62, pp. 67–69]).

In 2007, Spivey [163] found a connection between quadratic residues and those positive integers that are not in $\langle x_1, x_2 \rangle$ as described in the following theorem.

Theorem 1.3.3. *[163] Let $x_1, x_2 \geq 3$ be two primes. Then the squares of integers that are not in $\langle x_1, x_2 \rangle$ modulo x_1 (respectively, x_2) consist of $x_2 - 1$ (respectively, $x_1 - 1$) copies of each of the quadratic residues of x_1 (respectively, x_2).*

Example 1.3.4. As we saw in Example 1.1.7, the only positive integers not in $\langle 3, 5 \rangle$ are $1, 2, 4, 7$. The set of quadratic residues of 3 is $\{1\}$, and the set of quadratic residues of 5 is $\{1, 4\}$. Their relation is illustrated in Table 1.2: there are $5 - 1 = 4$ copies for each quadratic residue of 3, and $3 - 1 = 2$ copies for each quadratic residue of 5.

Table 1.2: Relation between quadratic residues of $3, 5$ and $x \in \mathbb{N} \setminus \langle 3, 5 \rangle$

x	1	2	4	7
x^2	1	4	16	49
$x^2 \bmod 3$	1	1	1	1
$x^2 \bmod 5$	1	4	1	4

1.3.3 Local postage-stamp problem

The *local postage-stamp problem* (LPSP) was introduced by Rohrbach [141, 140].

Problem 1.3.5 (*Local postage-stamp problem*)**.** *Given an integer $h \geq 1$ and $k+1$ positive integers $x_1, x_2, \ldots, x_k$, where $x_1 = 1$, what is the smallest positive integer that* cannot *be represented in the form $c_1x_1 + c_2x_2 + \cdots + c_kx_k$, where all the c_i are non-negative integers for $1 \leq i \leq k$ and $c_1 + c_2 + \cdots + c_k \leq h$?*

The name of the local postage-stamp problem comes from the following informal description of the problem: if the available denominations of stamps are $x_1, x_2, \ldots, x_k$, then what is the smallest amount of postage that *cannot* fit on an envelope of size h?

We denote by $\langle x_1, x_2, \ldots, x_k \rangle_h$ the set of all integers that can be represented as non-negative integer linear combinations $c_1x_1 + c_2x_2 + \cdots + c_kx_k$, where $c_1 + c_2 + \cdots + c_k \leq h$, and denote by $N_h(x_1, x_2, \ldots, x_k)$ the smallest positive integer that is not in $\langle x_1, x_2, \ldots, x_k \rangle_h$. It is folklore that

$$N_h(x_1, x_2) = (h + 3 - x_2)x_2 - 1. \tag{1.83}$$

There are a few papers on the LPSP and its variations, such as the global postage-stamp problem (refer to Guy's book [60, pp. 123–126]). The FP is related to the LPSP. For example, Shallit [153] proved LPSP to be NP-hard under Turing reductions by reducing from the FP, which is NP-hard.

Theorem 1.3.6. *[153] The LPSP is NP-hard under Turing reductions.*

The proof relies on the fact that given positive integers $b_1 < b_2 < \cdots < b_k$ with $\gcd(b_1, b_2, \ldots, b_k) = 1$, one can construct in polynomial time integers $h, a_1 = 1$, $a_2, \ldots, a_k, a_{k+1}, a_{k+2}$ such that $g(b_1, b_2, \ldots, b_k) = ha_{k+2} - N_h(a_1, a_2, \ldots, a_{k+2})$.

1.3.4 $(g_0, g_1, \ldots, g_k)$-tree

A tree is called a $(g_0, g_1, \ldots, g_k)$-tree if the number of children of every internal node belongs to $\{ g_0, g_1, \ldots, g_k \}$ and all leaves are of the same height. An integer n is called $(g_0, g_1, \ldots, g_k)$-realizable if there exists a $(g_0, g_1, \ldots, g_k)$-tree with n leaves. The $(g_0, g_1, \ldots, g_k)$-tree is a useful data structure; see the textbook [33, p. 300, p. 439] for examples of a $(3, 4)$-tree and a $(3, 4, 5)$-tree.

Theorem 1.3.7. *[120, 92] All positive integers except finitely many of them are $(g_0, g_1, \ldots, g_k)$-realizable if and only if* $\gcd(g_k - g_0, g_k - g_1, \ldots, g_k - g_{k-1}) = 1$.

The relationship between the $(g_0, g_1, \ldots, g_k)$-tree and the FP is the following: the largest integer that is not $(g_0, g_1, \ldots, g_k)$-realizable is $\geq g(g_0-1, g_1-1, \ldots, g_k-1)-1$.

1.3.5 Shellsort algorithm

An application of the FP appears in the analysis of the Shellsort algorithm [76, 148]. The *Shellsort* sorting algorithm was invented by Shell [159] in 1959; see Figure 1.2. Depending on the chosen increments $h_t, \ldots, h_2, h_1 = 1$, the computational complexity of the Shellsort algorithm varies. It is well known that for $t > 1$ the running time of the Shellsort algorithm with Shell's original setting $h = \ldots, 2^j, \ldots, 4, 2, 1$ is $O(N^2)$ in the worst case, where N is the number of elements to be sorted. The

Input: an array $a[1], a[2], \ldots, a[N]$ to be sorted
Output: sorted array a

```
1 foreach increment h in { h_t, ..., h_2, h_1 = 1 } do
2 |   foreach arithmetic progression index sub-array of increment h do
3 |   |   sort ..., a[i-2h], a[i-h], a[i], ... using insertion sort
4 |   end
5 end
```

Figure 1.2: Shellsort algorithm

works of Ajtai, Komlós and Szemerédi [3] and Leighton [93] on sorting network theory showed the possibility of the existence of increments that can make Shellsort run in $O(N \log N)$ time on average, but no such increment has been found yet. So far, the best increments were given by Pratt [124]. With Pratt's increments the Shellsort algorithm runs in $O(N(\log N)^2)$ time.

In 1985, by studying the FP, Incerpi and Sedgewick [76] showed increments that can make Shellsort run in $O(N^{1+\frac{\varepsilon}{\sqrt{\log N}}})$ time for any $\varepsilon > 0$. Their analysis was based on the following lemma.

Lemma 1.3.8. *[76] The number of steps required to sort with the increment h_j after a list of size N is already sorted with increments $h_t, \ldots, h_{j+2}, h_{j+1}$ in Shellsort is*

$$cost_j \leq N n_{h_j}(h_{j+1}, h_{j+2}, \ldots, h_t) = O\left(\frac{1}{h_j} N g(h_{j+1}, h_{j+2}, \ldots, h_t)\right), \tag{1.84}$$

where $n_{h_j}(h_{j+1}, \ldots, h_t)$ is the number of multiples of h_j that are not in $\langle h_{j+1}, \ldots, h_t \rangle$ and $g(h_{j+1}, \ldots, h_t)$ is the Frobenius number of $h_{j+1}, \ldots, h_t$.

For example, for either Hibbard's increments $1, 3, 7, \ldots, 2^k - 1$, or Papernov-Stasevich's increments $1, 3, 5, \ldots, 2^k+1$, or Knuth's increments $1, 4, 13, \ldots, \frac{1}{2}(3^k-1)$, the running time of Shellsort based on these increments are upper bounded by

$$\sum_{1 \leq j \leq t} \text{cost}_j = \sum_{1 \leq j \leq t_0} O(N h_j) + \sum_{t_0 \leq j \leq t} O\left(\frac{N^2}{h_j}\right) = O\left(N^{\frac{3}{2}}\right) + O\left(N^{\frac{3}{2}}\right) = O\left(N^{\frac{3}{2}}\right), \tag{1.85}$$

where t_0 is an index such that $h_{t_0} = \Theta(N^{\frac{1}{2}})$. The upper bound on the first sum in (1.85) comes from Lemma 1.3.8. The upper bound on the second sum in (1.85) comes from the analysis of insertion sort.

There are also papers on the analysis of the complexity of Shellsort using different methods (for example, see Jiang, Li and Vitányi [79]).

1.3.6 Tiling problem

The tiling problem is a family of problems, and the history of the tiling problem can be traced back thousands of years ago to the early Greeks. The general tiling

problem is to use small pieces of certain types of tessellations, called *tiles*, with fixed shapes and possibly painted colors, to mosaic a large area. Some problems allow the tiles to be rotated, while other problems do not allow rotating tiles. The FP is also related to a tiling problem [86].

Problem 1.3.9 (Problem B-3 from the 1991 William Lowell Putnam Examination)**.** *[86] "Does there exist a real number L such that, if m and n are integers greater than L, then an $m \times n$ rectangle may be expressed as a union of 4×6 and 5×7 rectangles, any two of which intersect at most along their boundaries?"*

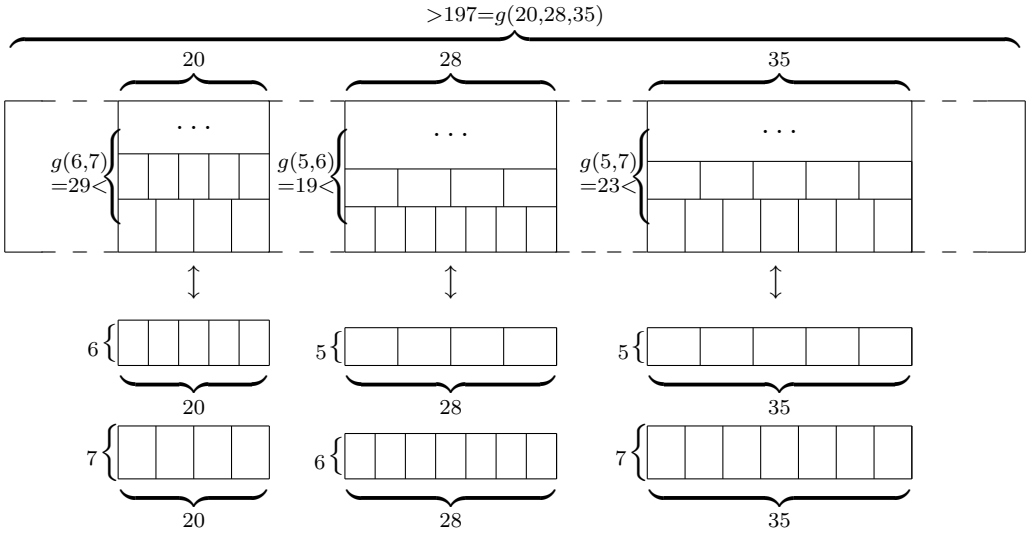


Figure 1.3: Illustration to the solution to Problem B-3

Solution. Here rotation of tiles is allowed. First, the 20×6 and 20×7 rectangles can be tiled by five 4×6 tiles and four 5×7 tiles, respectively. Since $g(6,7) = 29$, every $20 \times n$ rectangle can be tiled for any $n > 29$. Then the 28×5 and 28×6 rectangles can be tiled by four 5×7 tiles and seven 4×6 tiles respectively. So every $28 \times n$ rectangle can be tiled for any $n > g(5,6) = 19$. Finally, the 35×5 and 35×7 rectangles can be tiled by five and seven 5×7 tiles respectively and thus every $35 \times n$ rectangle can be tiled for any $n > g(5,7) = 23$. Since $\gcd(20, 28, 35) = 1$ and $g(20, 28, 35) = 4g(5, 7, 35) + 3 \cdot 35 = 4g(5, 7) + 3 \cdot 35 = 4 \cdot 23 + 3 \cdot 35 = 197$, any $m \times n$ rectangle with $m, n > \max\{29, 19, 23, 197\} = 197$ can be tiled by first dividing the $m \times n$ rectangle into several $20 \times n, 28 \times n, 35 \times n$ blocks and then tiling each block by the given 4×6 and 5×7 tiles. So, $L = 197$ is one solution. □

In 2002, Narayan and Schwenk [115] proved that for $m \geq n \geq 28$ the only $m \times n$ rectangles that cannot be tiled with the 4×6 and 5×7 tiles are the rectangles of sizes 31×29, 33×32, and 33×33. So the smallest L is 33.

The FP is related to many other problems, and has applications in various fields, including but not limited to combinatorics, number theory, and algorithms. In fact,

a recent book by Ramírez-Alfonsín [130] lists over 400 references on the FP, related problems (including most of the problems mentioned here), and applications. To mention a few that we do not consider in the thesis, applications of the FP appear in the theory of Petri nets [166, 31, 173], in the upper bound for the partition of a vector space [65], in the structure of monomial curves [112], in the study of algebraic geometric codes [69], in generating random vectors [171], in DNA sequencing [13, 14], and in image description [133].

1.4 Shallit's non-commutative generalization

1.4.1 Generalizations of the FP

There are several generalizations of the FP in algebraic structures other than the integers. For example, Skupień [162] in 1993 studied the following generalization of the FP in a numerical semigroup (also see Hofmeister [67]):

Problem 1.4.1 (*Modular generalization, FP*)**.** *[162] Let* $x_1, x_2, \ldots, x_k, m \in \mathbb{N}$*. We define* $N_j(m; x_1, x_2, \ldots, x_k)$ *to be the maximum* $p \in \mathbb{N}$ *such that* p cannot *be written as* $p = \sum_{i=1}^{k} c_i x_i$*, where* $c_i \in \mathbb{N}$ *for* $1 \leq i \leq k$ *and* $\sum_{i=1}^{k} c_i \equiv j \pmod{m}$*. Then what is the largest* $N_j(m; x_1, x_2, \ldots, x_k)$ *for* $0 \leq j \leq m-1$*?*

If we define $K(m; x_1, x_2, \ldots, x_k) = \max_{0 \leq j \leq k-1} N_j(m; x_1, x_2, \ldots, x_k)$, then Problem 1.4.1 is a generalization of the Frobenius number in the sense that

$$K(1; x_1, x_2, \ldots, x_k) = g(x_1, x_2, \ldots, x_k). \tag{1.86}$$

A multi-dimensional version of the FP about integer vectors has been studied independently (Knight [87]; Vizvári [170]). Instead of integers in $\mathbb{N}$, the multi-dimensional FP considers integer vectors in $\mathbb{N}^m$.

Problem 1.4.2 (*Multi-dimensional generalization, FP*)**.** *[170] Let* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \in \mathbb{N}^m$ *be* k *integer vectors and* $\langle \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \rangle$ *be the set of all vectors that can be written as a non-negative integer linear combination of* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$*. What is the vector* $\boldsymbol{v}$ *with the smallest Euclidean distance from the origin such that every integral vector in* $\boldsymbol{v} + \left\{ c_1 \boldsymbol{x}_1 + \cdots + c_k \boldsymbol{x}_k : 0 \leq c_1, \ldots, c_k \in \mathbb{R} \right\}$ *is in* $\langle \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \rangle$*?*

If $m = 1$, then $\left\{ c_1 x_1 + c_2 x_2 + \cdots + c_k x_k : 0 \leq c_1, c_2, \ldots, c_k \in \mathbb{R} \right\}$ is just a ray $\left\{ n \in \mathbb{R} : n \geq 0 \right\}$ and Problem 1.4.2 becomes the original FP.

There are also papers on a continuous version of the FP, where positive real numbers with certain conditions are considered (Lev [96]; Lenstra and Pomerance [95]).

In this thesis, we will focus on generalizations of the FP in a free monoid. Instead of considering the non-negative integer linear combination of integers, we will consider the concatenation of words. The concatenation operation does not satisfy the commutative law in general while integer addition does. Due to the lack of commutative law, the FP in a free monoid exhibits different and interesting properties.

1.4.2 Free monoids and the FP

Let S and T be two sets. As usual, $S \cup T$, $S \cap T$, and $S \setminus T$ denote the set union, set intersection, and set difference of S and T, respectively. We use both $\overline{S}$ and S^{-} to denote the complement of S, use $\mathcal{P}(S)$ to denote the powerset of S, and use both $|\, S \,|$ and $\#S$ to denote the cardinality of the set S. The symbol $\emptyset$ denotes the empty set $\{\ \}$. If S is a subset of T, we write $S \subseteq T$, and write $S \subsetneq T$ in case $S \subseteq T$ and $S \neq T$.

A *binary operation* in a set S is a mapping from the cartesian product $S \times S$ into S as $(a, b) \mapsto c$, where a, b, c are in S and we denote c by $a \cdot b$, which is written as ab for short, or by $a + b$ in some cases. A binary operation $\cdot$ is *associative* if $(ab)c = a(bc)$ and *commutative* if $ab = ba$.

A *semigroup* is an algebra $(S, \cdot)$, where S is a set and $\cdot$ is an associative binary operation in S. For brevity, when we call a set S a semigroup, we mean the semigroup $(S, \cdot)$, where the binary operation $\cdot$ is implicitly known.

An element e of S is an *identity* (for the binary operation $\cdot$) if $ea = ae = a$ for every a in S. If an identity exists, then it must be unique.

A semigroup is called a *monoid* if it has an identity.

Let Σ be a set called an *alphabet*, where each element is called a *letter*. In this thesis, without loss of generality, we always assume the letters in Σ are linearly ordered and written as $\mathtt{0}, \mathtt{1}, \mathtt{2}, \ldots (, \mathtt{z})$ where $\mathtt{z}$ is the largest letter in Σ if Σ is finite. For a letter $a \in \Sigma$, we write Σ_a to refer to the set $\Sigma \setminus \{\, a \,\}$. In this thesis, we write arbitrary alphabets as Σ, Δ, ... and we write arbitrary letters as a, b, c, d, ...

A (finite) *word* (or *string*) w over the alphabet Σ is a finite sequence of letters written as $a_1 a_2 \cdots a_n$, and the *empty word* is written as ϵ. The *length* of a word is defined by $|\, a_1 a_2 \cdots a_n \,| = n$, and $|\, \epsilon \,| = 0$. Let $u = a_1 a_2 \cdots a_n$ and $v = b_1 b_2 \cdots b_m$ be two words. We say words u and v are equal if $m = n$ and $a_i = b_i$ for $1 \leq i \leq n$. In this thesis, we write arbitrary words as u, v, w, x, y, z, ...

We write Σ^{+} to refer to the set of all (finite) non-empty words over Σ, write Σ^{*} to refer to the set of all (finite) words over Σ, and write Σ^{k} to refer to the set of all words of length k. The *concatenation* $\cdot$ of two words is defined as follows:

$$x_1 x_2 \cdots x_n \cdot y_1 y_2 \cdots y_m = x_1 x_2 \cdots x_n y_1 y_2 \cdots y_m. \tag{1.87}$$

Then $(\Sigma^{+}, \cdot)$ is a semigroup, which is called a *free semigroup*, and $(\Sigma^{*}, \cdot)$ is a monoid, which is called a *free monoid*.

For the word $w = a_1 a_2 \cdots a_n$, we define

$$w[i..j] = \begin{cases} a_i a_{i+1} \cdots a_j, & \text{if } 1 \leq i \leq j \leq n; \\ \epsilon, & \text{otherwise.} \end{cases} \tag{1.88}$$

Any word $w[1..i]$ for $0 \leq i \leq |\, w \,|$ is called a *prefix* of w and any word $w[i..\,|\, w \,|]$ for $1 \leq i \leq |\, w \,| + 1$ is called a *suffix* of w. We say a prefix (suffix) u of w is *proper* if

$u \neq w$. We write $\mathrm{Pref}(w)$ and $\mathrm{Suff}(w)$ to refer to the set of all prefixes and suffixes of w, respectively. A word u is a *conjugate* of a word v if $u = v[i+1..\,|\,v\,|]v[1..i]$ for some $1 \leq i \leq |\,v\,|$ and in this case we write $u \sim v$. The conjugacy relation $\sim$ is an equivalence relation.

Let Σ^*, Δ^* be two free monoids. A *morphism* $h : \Sigma^* \to \Delta^*$ is a mapping such that $h(uv) = h(u)h(v)$ for all $u, v \in \Sigma^*$. Then h is uniquely determined by $h' : \Sigma \to \Delta^*$ where $h'(a) = h(a)$ for all $a \in \Sigma$. Usually, when we define a morphism h, we only write down h' for brevity. For a morphism $h : \Sigma^* \to \Delta^*$, the *inverse morphism* $h^{-1} : \mathcal{P}(\Delta^*) \to \mathcal{P}(\Sigma^*)$ is defined by $h^{-1}(L) = \{\, w : h(w) \in L \,\}$. Two free monoids Σ^*, Δ^* are *isomorphic* if there are two morphisms $f : \Sigma^* \to \Delta^*, g : \Delta^* \to \Sigma^*$ such that $g(f(u)) = u$ and $f(g(v)) = v$ for all $u \in \Sigma^*$ and $v \in \Delta^*$.

The free monoid over an alphabet Σ is one of the major research objects in formal language theory and combinatorics [101, 90, 160], and results from the research on the free monoid have various applications in many domains, such as communication theory [50, 10], algebraic linguistics [30, 53], DNA sequence computing [132, 106], and text compression [176]. Due to the wide usage of the free monoid, the generalization of the Frobenius problem in a free monoid is an interesting topic.

Example 1.4.3. Let $\Sigma = \{\, \mathtt{0}, \mathtt{1} \,\}$ be the binary alphabet. Then the set of all finite binary words is $\Sigma^* = \{\, \epsilon, \mathtt{0}, \mathtt{1}, \mathtt{00}, \mathtt{01}, \mathtt{10}, \mathtt{11}, \ldots \,\}$ and, together with the concatenation operation, comprise a free monoid.

Example 1.4.4. Let $\Sigma = \{\, \mathtt{0} \,\}$ be the unary alphabet. Then the set of all finite unary words is $\Sigma^* = \{\, \epsilon, \mathtt{0}, \mathtt{00}, \mathtt{000}, \ldots \,\}$ and, together with the concatenation operation, comprise a free monoid. In the free monoid over a unary alphabet, the commutative law is satisfied.

The non-negative integers in $\mathbb{N}$ with addition constitute a monoid, which is isomorphic to the free monoid over the unary alphabet $\{\, \mathtt{0} \,\}$, specified by the morphism $f(y) = \mathtt{0}^y$ and $g(\mathtt{0}^y) = y$. It is, however, special in the sense that the addition satisfies the commutative law, which is in general not satisfied in a free monoid.

Let Σ be an alphabet. Any subset of Σ^* is called a *language* (in Σ^*). Let S, T be two languages. We denote by $S \cdot T$, or ST for short, the language

$$ST = \{\, xy : x \in S, y \in T \,\}. \tag{1.89}$$

We denote by S^k the language $\{\, x_1x_2\cdots x_k : x_1, x_2, \ldots, x_k \in S \,\}$, denote by S^+ the *concatenation closure* of S, which is $\{\, x_1x_2\cdots x_k : x_1, x_2, \ldots, x_k \in S, k \geq 1 \,\}$, and denote by S^* the language $S^+ \cup \{\, \epsilon \,\}$.

The *reverse* of a word $w = a_1a_2a_3\cdots a_k$ is $w^R = a_ka_{k-1}\cdots a_2a_1$, and the *reverse* of a language S is $S^R = \{\, w^R : w \in S \,\}$. A *palindrome* w is a word such that $w^R = w$.

A language S in Σ^* is called *finite* if it consists of finitely many words. If S is finite, we let $\mathrm{llw}\,(S)$ denote the length of the longest word(s) in S:

$$\mathrm{llw}\,(S) = \max_{w \in S} |\,w\,|\,. \tag{1.90}$$

We define $\mathrm{llw}(\emptyset) = -1$ and define $\mathrm{llw}(S) = \infty$ when S is infinite. A language S is called *co-finite* (in Σ^*) if its complement $\overline{S} = \Sigma^* \setminus S$ is finite.

Shallit asked the following question: *given k words $x_1, x_2, \ldots, x_k$ such that $\{ x_1, x_2, \ldots, x_k \}^*$ is co-finite, then what is (the length of) the longest word that is not in the language $\{ x_1, x_2, \ldots, x_k \}^*$?* This problem is the *Frobenius problem in a free monoid*, and the goal of this thesis is to try to answer Shallit's question from different points of view.

The Frobenius problem in a free monoid is a generalization of the Frobenius problem in the sense of the following theorem, which follows from Theorem 1.1.4.

Theorem 1.4.5. *[83, 84] Let $\Sigma = \{ 0 \}$ and $S = \{ 0^{x_1}, 0^{x_2}, \ldots, 0^{x_k} \}$. Then S^* is co-finite if and only if $\gcd(x_1, x_2, \ldots, x_k) = 1$. Furthermore, if S^* is co-finite, then the longest word in $\overline{S^*}$ is $0^{g(x_1, x_2, \ldots, x_k)}$ and the number of words in $\overline{S^*}$ is $h(x_1, x_2, \ldots, x_k)$.*

As shown in Theorem 1.1.4, the condition $\gcd(x_1, x_2, \ldots, x_k) = 1$ is necessary and sufficient for the existence of the Frobenius number, $g(x_1, x_2, \ldots, x_k)$, for given positive integers $x_1, x_2, \ldots, x_k$. For the Frobenius problem in a free monoid, however, there is no such simple condition. In fact, as we will see in Chapter 5, CO-FINITENESS OF STAR OF STAR-FREE REGULAR EXPRESSION, a strongly related decision problem, is NP-hard. A brief comparison of the original Frobenius problem and the Frobenius problem in a free monoid is given in Table 1.3.

Table 1.3: Comparison of the original FP and the FPFM

the original FP	the FP in a free monoid
$x_1, \ldots, x_k$ are integers	$x_1, \ldots, x_k$ are words
non-negative integer linear combination	concatenation of words
$\langle x_1, \ldots, x_k \rangle$	$\{ x_1, x_2, \ldots, x_k \}^*$
$\gcd(x_1, \ldots, x_k) = 1$	the language is co-finite
the Frobenius number	(length of) the longest word(s) omitted
$g(x_1, \ldots, x_k)$	$\mathrm{llw}\left(\overline{\{ x_1, x_2, \ldots, x_k \}^*}\right)$

Our generalization is non-commutative because in general the concatenation of words does not satisfy the commutative law. In other words, while the non-negative integers $\mathbb{N}$ with addition and multiplication comprise two commutative (abelian) monoids $(\mathbb{N}, +, 0)$ and $(\mathbb{N}, \times, 1)$, a free monoid $(\Sigma^*, \cdot, \epsilon)$ is not a commutative (abelian) monoid in general.

To avoid ambiguity, we call the original Frobenius problem (FP) the integer Frobenius problem, while we call the new generalization the Frobenius problem in a free monoid (FPFM).

1.5 Organization of the thesis

In this thesis, I will discuss generalized forms of the Frobenius problem in the universe of words, including Shallit's non-commutative Frobenius problem and other variations.

In Chapter 2, I will give the definition of the Frobenius problem in a free monoid (FPFM) and discuss some general properties of the FPFM. Then I will mainly focus on a particular subproblem, the 2FPFM, where the words in the basis are of only two distinct lengths. I will discuss the 2FPFM from different points of view. I will present two equivalent problems and finally give the complete spectrum of the solutions of the 2FPFM. A word graph, as the generalization of the famous de Bruijn graph, is introduced in order to discuss the 2FPFM. At the end of Chapter 2, I will discuss the Frobenius problem in a free monoid in case where the lengths of words in the basis constitute a special sequence.

In Chapter 3, I will discuss some variations on the FPFM and related problems. The first variation is that the concatenation of words are taken in a fixed order. Then I will discuss the FPFM with the input and output being specified by other means (for example, deterministic finite automata — DFAs, nondeterministic finite automata — NFAs, regular expressions). Some other variations on the FPFM with different aspects instead of the length of the longest omitted words are: the total number of omitted words, numbers that can be solutions to instances of the FPFM, bases with small sizes, and the number of different factorizations. I will also discuss the co-finiteness of sets of infinite words (right-infinite, left-infinite, bi-infinite), concatenation with overlap, co-slender languages and other settings. At the end, I will discuss a generalization of the local postage-stamp problem in a free monoid.

In Chapter 4, I will present the construction of some essential examples to complement the theory of the 2FPFM discussed in Chapter 2. The examples include a description of bases achieving exponentially-long omitted words, and corresponding examples with input being specified by NFAs, regular expressions, and deterministic pushdown automata (DPDAs). The examples also include a description of bases achieving doubly-exponentially many omitted words. At the end, I will give some experimental statistics.

In Chapter 5, I will discuss the algorithm for and computational complexity of the FPFM and related problems. I will present two exponential-time algorithms for the general FPFM and one polynomial-time algorithm for the 2FPFM. At the end, I will show that a particular decision problem related to the FPFM is NP-hard and in PSPACE.

In the last chapter, I will summarize the main results in this thesis and list some open problems.

Part of my work in this thesis has appeared in papers [83, 84, 157].

Chapter 2

General theory of the FPFM

In §2.1 I will give the definition of the Frobenius problem in a free monoid (FPFM) and some general properties of the FPFM. In §2.2, I will discuss different ways of describing the problem and different measures that we can use to bound the solution. In §2.3, I will show bounds on the longest words that could be the solutions to the FPFM. In the remaining sections, except the last one, I will focus on a particular type of the FPFM, called the 2FPFM. In §2.4, I will give the definition of the 2FPFM. In §2.5, I will discuss the 2FPFM from the view of combinatorics on words and give an equivalent problem. In §2.6, I will use some concepts from graph theory and define the word graph for the 2FPFM to present another equivalent form of the 2FPFM. In §2.6, by generalizing the de Bruijn graph, I will also give the complete spectrum of the lengths of words that could be the solutions to the 2FPFM. Finally, in §2.7, I will tackle the FPFM in some other special cases.

The main theorems in this chapter are 2.4.12, 2.5.3, 2.6.6, and 2.6.19; these are my contribution to the FPFM.

2.1 The Frobenius problem in a free monoid

We will start from the formal definition of the *Frobenius problem in a free monoid* (FPFM), which is the main topic of this thesis, and then exhibit some examples and general properties of the FPFM.

2.1.1 Definition of the FPFM

Problem 2.1.1 (*Frobenius problem in a free monoid*)**.** *Let Σ be a (finite) alphabet. Given k non-empty words $x_1, x_2, \ldots, x_k \in \Sigma^*$ such that there are only finitely many words that* cannot *be written as concatenations of words in $\{ x_1, x_2, \ldots, x_k \}$, then what is the longest such word(s)?*

If a word w can be written as a concatenation of non-empty words as

$$w = x'_1 x'_2 \cdots x'_l, \tag{2.1}$$

where $l \geq 0$ and all $x'_i \in \{ x_1, x_2, \ldots, x_k \}$ for $1 \leq i \leq l$, then we call (2.1) a *factorization* of w (into $x_1, x_2, \ldots, x_k$) and each of the x'_i a *factor* of w. In particular, we say the empty word ϵ can be factorized into a concatenation of 0 words from any basis. The set S^* can be written as

$$S^* = \{ x'_1 x'_2 \cdots x'_l : x'_1, x'_2, \ldots, x'_l \in S, l \geq 0 \}, \tag{2.2}$$

which contains all words that can be factorized into words in S, and we say that S^* is *generated* by the *basis* $x_1, x_2, \ldots, x_k$, where $S = \{ x_1, x_2, \ldots, x_k \}$. Unlike the case of integers, the longest word that is not in S^*, if any, may not be unique.

Example 2.1.2. For any arbitrary alphabet Σ, let $S_1 = \Sigma^3 \cup \Sigma^5$. Then S_1^* is co-finite and $\overline{S_1^*} = \Sigma^1 \cup \Sigma^2 \cup \Sigma^4 \cup \Sigma^7$. The longest words not in S_1^* are Σ^7 and $\operatorname{llw}\left(\overline{S_1^*}\right) = 7 = g(3, 5)$.

The following propositions are about co-finiteness, some of which are based on results from the study of the integer FP.

In the definition of the FPFM, the number of the given words is finite. In fact, it does not matter whether the number of given words is finite or not, in the sense that one can always choose a finite subset of them to get a new basis, and the two bases generate the same co-finite language, as the following proposition says.

Proposition 2.1.3. *Let S be a set of words over Σ. If S^* is co-finite, then there exists a finite subset $T \subseteq S$ such that $T^* = S^*$.*

Proof. If $S^* = \Sigma^*$, then we can take $T = \Sigma$. Otherwise, $S^* \neq \Sigma^*$. Since S^* is co-finite, let $l = \operatorname{llw}\left(\overline{S^*}\right)$. We now consider the finite set of words

$$U = \Sigma^{l+1} \cup \Sigma^{l+2} \cup \cdots \cup \Sigma^{2l+1}. \tag{2.3}$$

By Formula (1.72), $g(l+1, l+2, \ldots, 2l+1) = l$. So any word of length $l' > l$ can be factorized into words in U according to the partition of l' into $l+1, l+2, \ldots, 2l+1$. Hence U^* is co-finite and $\operatorname{llw}\left(\overline{U^*}\right) = l = \operatorname{llw}\left(\overline{S^*}\right)$. Furthermore, all words in U can be factorized into words in S. Let T_1 be the set of all words that appear in a factorization, into the basis S, of each word in U, and let T_2 be the set of all words of lengths $\leq l$ in S. Then $T = T_1 \cup T_2$ is a finite subset of S, and thus $T^* \subseteq S^*$. Every word of length $> l$ is in S^* and is also in T_1^*. A word of length $\leq l$ is in S^* if and only if it can be factorized into words in T_2. So $S^* \subseteq (T_1^* \cup T_2^*) \subseteq T^*$. Therefore, $T^* = S^*$. □

The following proposition is particularly useful to decide if a language generated by a basis is co-finite. To show that S generates a co-finite language, one only needs to check that S^* covers all words of some lengths x_1, x_2 with $\gcd(x_1, x_2) = 1$. In particular, one can choose x_1, x_2 to be two consecutive integers.

Proposition 2.1.4. *Let $S \subseteq \Sigma^*$. Then S^* is co-finite if and only if there exist two positive integers y_1, y_2 such that $\gcd(y_1, y_2) = 1$ and $\Sigma^{y_1} \cup \Sigma^{y_2} \subseteq S^*$. Furthermore, $\mathrm{llw}\left(\overline{S^*}\right) \leq g(y_1, y_2)$.*

Proof. $\Rightarrow$: If S^* is co-finite, let $y_1, y_2 > \max\left\{1, \mathrm{llw}\left(\overline{S^*}\right)\right\}$ be two distinct primes. Then $T = (\Sigma^{y_1} \cup \Sigma^{y_2}) \subseteq S^*$ and $\gcd(y_1, y_2) = 1$.

$\Leftarrow$: Let y_1, y_2 be two positive integers such that $\gcd(y_1, y_2) = 1$ and $T = \Sigma^{y_1} \cup \Sigma^{y_2} \subseteq S^*$. Then any word of length $l > g(y_1, y_2)$ can be factorized into words in T according to the partition of l into y_1, y_2, and thus T^* is co-finite. Since $T^* \subseteq (S^*)^* = S^*$, we know that S^* is also co-finite.

Furthermore, $\mathrm{llw}\left(\overline{S^*}\right) \leq \mathrm{llw}\left(\overline{T^*}\right) = g(y_1, y_2)$. □

For any fixed $k \geq 2$, a result similar to Proposition 2.1.4 also holds: namely, S^* is co-finite if and only if S^* includes all words of some lengths $y_1, y_2, \ldots, y_k$ with $\gcd(y_1, y_2, \ldots, y_k) = 1$, and furthermore $\mathrm{llw}\left(\overline{S^*}\right) \leq g(y_1, y_2, \ldots, y_k)$.

There exist non-trivial instances of the FPFM. Here are several examples.

Example 2.1.5. For the binary alphabet, let $S_2 = \{\, \mathtt{1}, \mathtt{00}, \mathtt{01}, \mathtt{10}, \mathtt{000}, \mathtt{010} \,\}$. Then S_2^* is co-finite and factorizations of words of lengths ≤ 3 are given below:

	(ϵ)	$\mathtt{0}$	$(\mathtt{1})$	$(\mathtt{00})$	$(\mathtt{01})$	$(\mathtt{10})$	$(\mathtt{1})(\mathtt{1})$
$(\mathtt{000})$	$(\mathtt{00})(\mathtt{1})$	$(\mathtt{010})$	$(\mathtt{01})(\mathtt{1})$	$(\mathtt{1})(\mathtt{00})$	$(\mathtt{1})(\mathtt{01})$	$(\mathtt{1})(\mathtt{10})$	$(\mathtt{1})(\mathtt{1})(\mathtt{1})$

Since $\gcd(2, 3) = 1$, all words of lengths $\geq 4 > g(2, 3) = 1$ are in S_2^*. Therefore, the only word not in S_2^* is $\mathtt{0}$.

Example 2.1.6. For the binary alphabet, let

$$S_3 = \{\, \mathtt{0}, \mathtt{11}, \mathtt{001}, \mathtt{010}, \mathtt{100}, \mathtt{101}, \mathtt{111}, \mathtt{1001}, \mathtt{1011}, \mathtt{1101}, \mathtt{10001}, \mathtt{10101} \,\}. \tag{2.4}$$

Then S_3^* is co-finite and factorizations of words of lengths ≤ 5 are given below:

	(ϵ)	$(\mathtt{0})$	$\mathtt{1}$	$(\mathtt{0})^2$	$\mathtt{01}$	$\mathtt{10}$	$(\mathtt{11})$
$(\mathtt{0})^3$	$(\mathtt{001})$	$(\mathtt{010})$	$(\mathtt{0})(\mathtt{11})$	$(\mathtt{100})$	$(\mathtt{101})$	$(\mathtt{11})(\mathtt{0})$	$(\mathtt{111})$
$(\mathtt{0})^4$	$(\mathtt{0})(\mathtt{001})$	$(\mathtt{0})(\mathtt{010})$	$(\mathtt{0})^2(\mathtt{11})$	$(\mathtt{0})(\mathtt{100})$	$(\mathtt{0})(\mathtt{101})$	$(\mathtt{0})(\mathtt{11})(\mathtt{0})$	$(\mathtt{0})(\mathtt{111})$
$(\mathtt{100})(\mathtt{0})$	$(\mathtt{1001})$	$(\mathtt{101})(\mathtt{0})$	$(\mathtt{1011})$	$(\mathtt{11})(\mathtt{0})^2$	$(\mathtt{1101})$	$(\mathtt{111})(\mathtt{0})$	$(\mathtt{11})^2$
$(\mathtt{0})^5$	$(\mathtt{0})^2(\mathtt{001})$	$(\mathtt{0})^2(\mathtt{010})$	$(\mathtt{0})^3(\mathtt{11})$	$(\mathtt{0})^2(\mathtt{100})$	$(\mathtt{0})^2(\mathtt{101})$	$(\mathtt{0})^2(\mathtt{11})(\mathtt{0})$	$(\mathtt{0})^2(\mathtt{111})$
$(\mathtt{0})(\mathtt{100})(\mathtt{0})$	$(\mathtt{0})(\mathtt{1001})$	$(\mathtt{0})(\mathtt{101})(\mathtt{0})$	$(\mathtt{0})(\mathtt{1011})$	$(\mathtt{0})(\mathtt{11})(\mathtt{0})^2$	$(\mathtt{0})(\mathtt{1101})$	$(\mathtt{0})(\mathtt{111})(\mathtt{0})$	$(\mathtt{0})(\mathtt{11})^2$
$(\mathtt{100})(\mathtt{0})^2$	$(\mathtt{10001})$	$(\mathtt{1001})(\mathtt{0})$	$(\mathtt{100})(\mathtt{11})$	$(\mathtt{101})(\mathtt{0})^2$	$(\mathtt{10101})$	$(\mathtt{1011})(\mathtt{0})$	$(\mathtt{101})(\mathtt{11})$
$(\mathtt{11})(\mathtt{0})^3$	$(\mathtt{11})(\mathtt{001})$	$(\mathtt{11})(\mathtt{010})$	$(\mathtt{11})(\mathtt{0})(\mathtt{11})$	$(\mathtt{11})(\mathtt{100})$	$(\mathtt{11})(\mathtt{101})$	$(\mathtt{11})^2(\mathtt{0})$	$(\mathtt{11})(\mathtt{111})$

Since $\gcd(3, 4, 5) = 1$, all words of lengths $\geq 6 > g(3, 4, 5) = 2$ are in S_3^*. Therefore, the only words not in S_3^* are in $\{\, \mathtt{1}, \mathtt{01}, \mathtt{10} \,\}$ and $\mathtt{01}, \mathtt{10}$ are the longest two.

Proposition 2.1.7. *Let $x_1, \ldots, x_k$ be k words over Σ. Then $L = \{\, x_1, \ldots, x_k \,\}^*$ is not co-finite if and only if there are words u, v, w such that $v \neq \epsilon$ and $uv^*w \subseteq \overline{L}$.*

Proof. Trivially, if such u, v, w exist, then by definition L is not co-finite. Now, we suppose L is not co-finite. Then $\overline{L}$ contains arbitrarily long words. The finite language $\{ x_1, x_2, \ldots, x_k \}$ is regular. Since the class of regular languages is closed under Kleene-star $*$ and complement, $\overline{L} = \overline{\{ x_1, x_2, \ldots, x_k \}^*}$ is also regular. So such u, v, w exist by the pumping lemma (for example, see the textbook [71]). □

Corollary 2.1.8. *Let $x_1, x_2, \ldots, x_k$ be k words over Σ such that $L = \{ x_1, \ldots, x_k \}^*$ is co-finite, and let $n = \max_{1 \le i \le k} | x_i |$. Then, for any letters $b_1, b_2, \ldots, b_n \in \Sigma$ and any word $u \in \Sigma^*$, we have $\{ ub_1,\ ub_1b_2,\ ub_1b_2b_3,\ \ldots,\ ub_1b_2 \cdots b_n \} \cap L \neq \emptyset$.*

Proof. Consider the language $L' = ub_1b_2 \cdots b_n\Sigma^*$. Since L is co-finite, $L' \cap L \neq \emptyset$. Let $w \in L' \cap L$, and let $w = v_1v_2 \cdots v_t$ be a factorization of w into words x's. Then there is some r such that $| v_1 \cdots v_r | \le | u | < | v_1 \cdots v_rv_{r+1} |$. Since $| v_{r+1} | \le n$, by comparing lengths, we have $v_1 \cdots v_{r+1} \in \{ ub_1,\ ub_1b_2,\ ub_1b_2b_3,\ \ldots,\ ub_1b_2 \cdots b_n \}$. So $\{ ub_1,\ ub_1b_2,\ ub_1b_2b_3,\ \ldots,\ ub_1b_2 \cdots b_n \} \cap L \neq \emptyset$. □

2.1.2 Relations of the FPFM and the Frobenius number

In the following part of this section, we will see some propositions about co-finiteness in the FPFM, and some propositions that illustrate the relation between the length of the longest omitted words in the FPFM and the Frobenius number. Our first proposition concerns restricting a basis to a smaller alphabet.

Proposition 2.1.9. *Let S be a set of words over Σ such that S^* is co-finite in Σ^*, and let $\Delta \subseteq \Sigma$. Then $T = S \cap \Delta^*$ is a set of words over Δ, and T generates a language that is co-finite in Δ^*. Furthermore,* $\mathrm{llw}(\Delta^* \setminus T^*) \le \mathrm{llw}(\Sigma^* \setminus S^*)$.

Proof. Every word w in $S^* \cap \Delta^*$ can be written as a factorization into the basis S and all factors are in Δ^*, so w is in $(S \cap \Delta^*)^*$ and thus $S^* \cap \Delta^* \subseteq (S \cap \Delta^*)^*$. On the other hand, $T^* = (S \cap \Delta^*)^* \subseteq S^* \cap \Delta^*$, so $T^* = S^* \cap \Delta^*$. Then $\Delta^* \setminus T^* = \Delta^* \setminus S^* \subseteq \Sigma^* \setminus S^*$, and thus $\mathrm{llw}(\Delta^* \setminus T^*) \le \mathrm{llw}(\Sigma^* \setminus S^*)$. □

Example 2.1.10. The basis S_2 over the binary alphabet in Example 2.1.5 contains two bases $\{ \mathtt{1} \}$ and $\{ \mathtt{00}, \mathtt{000} \}$ for the unary alphabet. The longest word not in S_2^* is $\mathtt{0}$ (of length 1), while $g(1) = -1, g(2, 3) = 1$. The basis S_3 over the binary alphabet in Example 2.1.6 contains two bases $\{ \mathtt{0} \}$ and $\{ \mathtt{11}, \mathtt{111} \}$ for the unary alphabet. The longest words not in S_3^* are $\mathtt{01}$ and $\mathtt{10}$ (of length 2), while $g(1) = -1, g(2, 3) = 1$.

Proposition 2.1.11. *Let $f : \Sigma_1^* \to \Sigma_2^*$ be a morphism such that $f(\Sigma_1^*)$ is co-finite in Σ_2^*. If $L \subseteq \Sigma_1^*$ is co-finite, then $f(L) \subseteq \Sigma_2^*$ is also co-finite. Furthermore, if $f(\Sigma_1^*) = \Sigma_2^*$, then* $\mathrm{llw}(f(\Sigma_1^* \setminus L)) \ge \mathrm{llw}(\Sigma_2^* \setminus f(L))$.

Proof. Since $A \setminus C \subseteq A \setminus B \cup B \setminus C$ for arbitrary A, B, C, then

$$\Sigma_2^* \setminus f(L) \subseteq \big(\Sigma_2^* \setminus f(\Sigma_1^*)\big) \cup \big(f(\Sigma_1^*) \setminus f(L)\big). \tag{2.5}$$

Since $f(A) \setminus f(B) \subseteq f(A \setminus B)$ for arbitrary A, B, then $f(\Sigma_1^*) \setminus f(L) \subseteq f(\Sigma_1^* \setminus L)$, and thus $\Sigma_2^* \setminus f(L)$ is finite. This proves the first assertion. If $f(\Sigma_1^*) = \Sigma_2^*$, then $\Sigma_2^* \setminus f(\Sigma_1^*) = \emptyset$, and so $\Sigma_2^* \setminus f(L) \subseteq f(\Sigma_1^* \setminus L)$. □

If we define a morphism of words by omitting particular letters, then the image of a co-finite language is also co-finite. This corollary follows immediately from Proposition 2.1.11.

Corollary 2.1.12. *Let $\Delta \subseteq \Sigma$ and define the morphism $|_\Delta : \Sigma^* \to \Delta^*$ by*

$$a|_\Delta = \begin{cases} a, & \text{if } a \in \Delta; \\ \epsilon, & \text{if } a \notin \Delta. \end{cases} \tag{2.6}$$

If $\{ x_1, \ldots, x_k \}^*$ is co-finite in Σ^*, then so is $\{ x_1|_\Delta, \ldots, x_k|_\Delta \}^*$ in Δ^*. Furthermore, $\operatorname{llw}\left(\{ w|_\Delta : w \in \Sigma^* \setminus \{ x_1, \ldots, x_k \}^* \} \right) \geq \operatorname{llw}(\Delta^* \setminus \{ x_1|_\Delta, \ldots, x_k|_\Delta \}^*)$.

Example 2.1.13. In Example 2.1.5, $S_2|_{\{0\}} = \{ 00, 0, 000 \}$ and $S_2|_{\{1\}} = \{ 1 \}$. The longest word not in S_2^* is 0 (of length 1), while $g(1, 2, 3) = -1, g(1) = -1$. In Example 2.1.6, $S_3|_{\{0\}} = \{ 0, 00, 000 \}$ and $S_3|_{\{1\}} = \{ 11, 1, 111 \}$. The longest words not in S_3^* are 01 and 10 (of length 2), while $g(1, 2, 3) = -1$. Here ϵ in every basis is omitted.

If a set of words generates a co-finite language, then the set of lengths of those words generates a co-finite set in $\mathbb{N}$. This can be viewed as another corollary of Proposition 2.1.11, but I will present a complete proof here for further discussion.

Proposition 2.1.14. *Let $x_1, \ldots, x_k$ be k words over Σ. If $L = \{ x_1, \ldots, x_k \}^*$ is co-finite, then* $\gcd(|x_1|, \ldots, |x_k|) = 1$. *Furthermore,* $\operatorname{llw}\left(\overline{L}\right) \geq g(|x_1|, \ldots, |x_k|)$.

Proof. One can verify that the mapping $f(w) = |w|$ is a morphism from $(\Sigma^*, \cdot)$ to $(\mathbb{N}, +)$. Hence, if $w \in \{ x_1, x_2, \ldots, x_k \}^*$, then $|w| \in \langle |x_1|, |x_2|, \ldots, |x_k| \rangle$. Assume $L = \{ x_1, x_2, \ldots, x_k \}^*$ is co-finite. Then there are only finitely many non-negative integers that are not in $\langle |x_1|, |x_2|, \ldots, |x_k| \rangle$. By Theorem 1.1.5, it follows that $\gcd(|x_1|, |x_2|, \ldots, |x_k|) = 1$. Any word of length $g(|x_1|, |x_2|, \ldots, |x_k|)$ is not in L, so $\operatorname{llw}\left(\overline{L}\right) \geq g(|x_1|, |x_2|, \ldots, |x_k|)$. □

From the proof, we know that the set of lengths of words in a language generated by a basis S is exactly the set generated by the lengths of words in S as follows:

$$\{ |w| : w \in \{ x_1, x_2, \ldots, x_k \}^* \} = \langle |x_1|, |x_2|, \ldots, |x_k| \rangle, \tag{2.7}$$

while in general a word not in a language generated by a basis S may be of length that is in the set generated by the lengths of words in S, which means

$$\{ |w| : w \notin \{ x_1, x_2, \ldots, x_k \}^* \} \supseteq \mathbb{N} \setminus \langle |x_1|, |x_2|, \ldots, |x_k| \rangle. \tag{2.8}$$

Proposition 2.1.15. *Let S be a set of words over Σ such that S^* is co-finite. Then there exists a finite sequence of integers $y_1, y_2, \ldots, y_{k'}$ such that* $\gcd(y_1, \ldots, y_{k'}) = 1$ *and the set of lengths of words not in S^* is exactly the set $\mathbb{N} \setminus \langle y_1, y_2, \ldots, y_{k'} \rangle$. In other words,* $\operatorname{llw}\left(\overline{S^*}\right) = g(y_1, y_2, \ldots, y_{k'})$.

Proof. Let $l = \operatorname{llw}\left(\overline{S^*}\right)$. We define $U_1 = \left\{ n \in \mathbb{N} : n < l, \Sigma^n \subseteq S^* \right\}$ and $U_2 = \{ l+1, l+2, \ldots, 2l+1 \}$. Let $U = U_1 \cup U_2$, and let $y_1, y_2, \ldots, y_{k'}$ be the integers in U. Then $\bigcup_{i \in U} \Sigma^i \subseteq S^*$ and the set of lengths of words not in S^* is exactly $\mathbb{N} \setminus \langle y_1, \ldots, y_{k'} \rangle$. Furthermore, $\gcd(y_1, \ldots, y_{k'}) = \gcd(l+1, l+2, \ldots, 2l+1) = 1$. □

From the proof, we also know that there may exist a word in S^* that is of a length not in $\langle y_1, y_2, \ldots, y_{k'} \rangle$, which means

$$\left\{ \, |\, w \,| : w \in S^* \right\} \supseteq \langle y_1, y_2, \ldots, y_{k'} \rangle, \tag{2.9}$$

while the set of lengths of words not in S^* is exactly $\mathbb{N} \setminus \langle y_1, y_2, \ldots, y_k \rangle$ as follows:

$$\left\{ \, |\, w \,| : w \notin S^* \right\} = \mathbb{N} \setminus \langle y_1, y_2, \ldots, y_{k'} \rangle. \tag{2.10}$$

Comparing Eqs. (2.9) and (2.10) with Eqs. (2.7) and (2.8), we know that the two sets of integers $\{ y_1, y_2, \ldots, y_{k'} \}$ and $\{ |\, x_1 \,|, |\, x_2 \,|, \ldots, |\, x_k \,| \}$ are not identical in general, nor there is any obvious relation between them. In fact, we will see in §2.5 that $y_{k'}$ can be exponential in $|\, x_k \,|$. The sequence $y_1, y_2, \ldots, y_{k'}$ is not unique.

Example 2.1.16. Characteristics of Examples 2.1.2, 2.1.5, and 2.1.6, with their constant sequences $y_1, y_2, \ldots, y_{k'}$ specified in Proposition 2.1.15, are in Table 2.1. We saw that Eqs. (2.7)–(2.10) hold and only in the example with basis S_1 all the equalities in Eqs. (2.8)–(2.9) are attained at the same time.

Table 2.1: Characteristics of the FPFM with bases S_1, S_2, and S_3

$S : x_1, \ldots, x_k$	$S_1 : \Sigma^3 \cup \Sigma^5$	$S_2 : {}^{1,0^2,01,10,}_{0^3,010}$	$S_3 : {}^{0,1^2,001,010,100,101,1^3,}_{10^21,1011,1101,10^31,(10)^21}$
$\|\, x_1 \,|, \ldots, |\, x_k \,|$	$\{ 3, 5 \}$	$\{ 1, 2, 3 \}$	$\{ 1, 2, 3, 4, 5 \}$
$y_1, \ldots, y_{k'}$	$\{ 3, 5 \}$	$\{ 2, 3 \}$	$\{ 3, 4, 5 \}$
$\left\{ \, \|\, w \,| : w \in S^* \right\}$	$\{ 0, 3, 5, 6, i \geq 8 \}$	$\{ i \geq 0 \}$	$\{ i \geq 0 \}$
$\left\{ \, \|\, w \,| : w \notin S^* \right\}$	$\{ 1, 2, 4, 7 \}$	$\{ 1 \}$	$\{ 1, 2 \}$
$\langle |\, x_1 \,|, \ldots, |\, x_k \,| \rangle$	$\{ 0, 3, 5, 6, i \geq 8 \}$	$\{ i \geq 0 \}$	$\{ i \geq 0 \}$
$\langle |\, x_1 \,|, \ldots, |\, x_k \,| \rangle^-$	$\{ 1, 2, 4, 7 \}$	$\emptyset$	$\emptyset$
$\langle y_1, \ldots, y_{k'} \rangle$	$\{ 0, 3, 5, 6, i \geq 8 \}$	$\{ 0, i \geq 2 \}$	$\{ 0, i \geq 3 \}$
$\langle y_1, \ldots, y_{k'} \rangle^-$	$\{ 1, 2, 4, 7 \}$	$\{ 1 \}$	$\{ 1, 2 \}$

2.1.3 Twins proposition in the FPFM

The following proposition is more powerful than it first appears, and later, based on this proposition, I will present a complete solution to the 2FPFM. I prefer to call it *twins* proposition in the Frobenius problem in a free monoid, because it shows that if S^* is co-finite, then every word in S must appear in groups (pairs, triples, etc.).

Proposition 2.1.17 (Twins proposition)**.** *[83, 84] Let S be a set of non-empty words over Σ such that S^* is co-finite and $S^* \neq \Sigma^*$. Then for each $x \in S$, there exists $y \in S$ such that y is a proper prefix of x, or vice versa. Similarly, for each $x \in S$, there exists $z \in S$ such that z is a proper suffix of x, or vice versa.*

Proof. Let $x \in S$. Since $S^* \neq \Sigma^*$, there is a word $v \notin S^*$. Consider the language x^*v. Since S^* is co-finite, $x^*v \cap S^* \neq \emptyset$. Let i be the smallest integer such that $x^iv \in S^*$. Since $v \notin S^*$, we have $i \geq 1$. Let

$$x^iv = y_1y_2 \cdots y_j \tag{2.11}$$

be a factorization of x^iv in the basis S. Then $y_1 \neq x$, for otherwise $x^{i-1}v = y_2 \cdots y_j \in S^*$, which contradicts the minimality of i. If $|\, x \,| < |\, y_1 \,|$, then x is a proper prefix of y_1, while otherwise y_1 is a proper prefix of x. Here $y_1 \in S$.

Consider the set S^R that contains the reverse of every word in S. Then $(S^R)^* \neq \Sigma^*$ is co-finite. By applying the result about prefixes on S^R, we see that the analogous result about suffixes holds. □

Example 2.1.18. The bases S_2 and S_3 in Examples 2.1.5 and 2.1.6 are grouped according to the prefix relation as follows, where every pair of words that satisfy the prefix relation is connected by a line:

S_2 : `1`—`10`, `00`—`000`, `01`—`010`;

S_3 : `001`—`0`—`010`, `111`—`11`—`1101`, `1001`—`100`—`10001`, `1011`—`101`—`10101`.

2.2 Various measures for the FPFM

Let Σ be an alphabet, which can be unary or of larger size. Given as input k words $x_1, x_2, \ldots, x_k$, which satisfy certain conditions, we can apply the Kleene-star operator to the language $S = \{\, x_1, x_2, \ldots, x_k \,\}$ and determine some characteristics of S^*. The FPFM is a special setting of the procedure described here.

If Σ is a unary alphabet: one particular case of the procedure is to let the condition be $\gcd(|\, x_1 \,|, |\, x_2 \,|, \ldots, |\, x_k \,|) = 1$ and let the output measure be $\mathrm{llw}\left(\overline{S^*}\right)$. Then we have a problem equivalent to the integer FP, which we know is NP-hard, and there is an upper bound on the output as follows: (see Eq. (1.41))

$$\mathrm{llw}\left(\overline{S^*}\right) = g(|\, x_1 \,|, |\, x_2 \,|, \ldots, |\, x_k \,|) \leq \frac{\max_{1 \leq i \leq k} |\, x_i \,|^2}{k-1}, \tag{2.12}$$

where $\max_{1\le i\le k} |\, x_i \,|$ is one possible measure of the input. There are also other measures of the input such as $\operatorname{lcm}(|\, x_1 \,|, \ldots, |\, x_k \,|)$, and we know (see Eq. (1.43))

$$\operatorname{llw}\left(\overline{S^*}\right) = g(|\, x_1 \,|, \ldots, |\, x_k \,|) \le (k-1)\operatorname{lcm}(|\, x_1 \,|, \ldots, |\, x_k \,|). \tag{2.13}$$

Another case of the procedure is, over an arbitrary alphabet Σ, to let the condition be that S^* is co-finite and let the output measure be $\operatorname{llw}\left(\overline{S^*}\right)$. Then we have the FPFM. In order to give an upper bound on the output, there are different measures of the input that can be chosen.

2.2.1 Measures of the input

In the general procedure described above, some choices of measures of the input are listed in Table 2.2. But by no means is the table exhaustive. Here the *state complexity* of a language L is the number of states in the minimal DFA that accepts L, and the *nondeterministic state complexity* of L is the minimal number of states in an NFA that accepts L. Here the *alphabetic length* of a regular expression is the number of alphabet symbols of that regular expression.

Table 2.2: Measures of a list of words $x_1, x_2, \ldots, x_k$ as the input

$\kappa = k$,	the number of distinct words in the input;
$\nu = \max_{1\le i\le k} \lvert\, x_i \,\rvert$,	the length of the longest words in the input;
$\mu = \sum_{1\le i\le k} \lvert\, x_i \,\rvert$,	the total number of symbols in the input;
$g(\lvert\, x_1 \,\rvert, \lvert\, x_2 \,\rvert, \ldots, \lvert\, x_k \,\rvert)$,	the Frobenius number of lengths of input words;
$\operatorname{lcm}(\lvert\, x_1 \,\rvert, \ldots, \lvert\, x_k \,\rvert)$,	the least common multiple of lengths of input words;
$\operatorname{sc}(S)$,	the state complexity of S;
$\operatorname{nsc}(S)$,	the nondeterministic state complexity of S;
$\operatorname{alph}(S)$,	the minimal alphabetic length of a regular expression for S.

There are more measures of the input in a free monoid than in the integers. Over the unary alphabet (or integers), obviously some of the measures are related, as we have

$$1 \le \kappa \le \nu \le \mu, \quad \text{and} \quad \mu = O\left(\nu^2\right). \tag{2.14}$$

But over a larger alphabet, both κ and μ can be exponentially large in ν when the input consists of all words of two lengths m, n with $\gcd(m, n) = 1$ as shown in Example 2.1.2, and thus they are different measures in general. Some of the measures are bounded in others such as

$$g(|\, x_1 \,|, \ldots, |\, x_k \,|) \le (\kappa - 1)\operatorname{lcm}(|\, x_1 \,|, \ldots, |\, x_k \,|), \quad g(|\, x_1 \,|, \ldots, |\, x_k \,|) \le \frac{\nu^2}{\kappa - 1},$$
$$\mu \le \kappa\nu, \qquad \operatorname{sc}(S) \le 2^{\operatorname{nsc}(S)}, \qquad \operatorname{nsc}(S) \le \operatorname{alph}(S). \tag{2.15}$$

2.2.2 Measures of the output

Depending on the output measure, the procedure described at the beginning of this section can become an entirely different problem. For example, as we saw in the integer FP, one can ask for the Frobenius number $g(x_1, x_2, \ldots, x_k)$, or the number $h(x_1, x_2, \ldots, x_k)$ of positive integers not in $\langle x_1, x_2, \ldots, x_k \rangle$, or the sum of positive integers not in $\langle x_1, x_2, \ldots, x_k \rangle$, or the denumerant $d(n; x_1, x_2, \ldots, x_k)$. In the FPFM, we ask, as output, for the longest words not in S^*. There are different measures of the longest words not in S^*, such as the length of such words, the number of such words, the total number of symbols in all such words (which is the product of the previous two measures). Furthermore, instead of considering the longest words not in S^*, other output measures are also possible. Table 2.3 lists several candidates, some of which lead to variations on the FPFM and will be discussed in Chapter 3.

Table 2.3: Measures of the longest omitted words and other characteristics

$\mathcal{L} = \operatorname{llw}((S^*)^-) = \max_{w \in \Sigma^* \setminus S^*} \lvert w \rvert$,	the length of the longest words not in S^*;
$\mathcal{I} = \#\{ w \notin S^* : \lvert w \rvert = \mathcal{L} \}$,	the number of the longest words not in S^*;
$\mathcal{IL} = \mathcal{L} \cdot \mathcal{I}$,	the total number of symbols in all the longest words not in S^*;
$\mathcal{M} = \#\overline{S^*}$,	the number of words not in S^*;
$\mathcal{W} = \sum_{w \in \overline{S^*}} \lvert w \rvert$,	the total number of symbols in all the words not in S^*;
$\mathcal{S} = \operatorname{sc}(S^*)$,	the state complexity of S^*;
$\mathcal{N} = \operatorname{nsc}(S^*)$,	the nondeterministic state complexity of S^*;
$\mathcal{R} = \operatorname{alph}(S^*)$,	the minimal alphabetic length of a regular expression for S^*;
$\mathcal{D}(w)$,	the number of different factorizations of w in the basis S.

The output in the case of a free monoid differs from that in the case of integers and there are more measures of the output. The measures in the case of the unary alphabet are specified by the subscript $\wr$. Over the unary alphabet (or integers), $\mathcal{I}_\wr$, if any, is always 1 and thus $\mathcal{IL}_\wr = \mathcal{L}_\wr$. Furthermore, as immediate consequences of results on the integer FP (see Formula (1.56)), the inequalities

$$\frac{\mathcal{L}_\wr}{2} \leq \mathcal{M}_\wr \leq \mathcal{L}_\wr \quad \text{and} \quad \mathcal{W}_\wr = O(\mathcal{L}_\wr^{\,2}) \tag{2.16}$$

hold. But over a larger alphabet, all of the quantities $\mathcal{I}, \mathcal{IL}, \mathcal{M}, \mathcal{W}$ may be exponential in $\mathcal{L}$, and in §2.5 I will show that they are exponential in some cases.

2.2.3 Constraints on the problem

In order to consider $\operatorname{llw}\left(\overline{S^*}\right)$, it is necessary that S^* be co-finite. When our measure of the output changes, there can be more or fewer conditions. For example, even when S^* is not co-finite, we can still study the state complexity of S^*. Furthermore,

imposing certain constraints can simplify the problem and leads to interesting subproblems. As we saw in the integer FP, when $\kappa = 2$ nearly all output measures can be expressed by a simple formula. Table 2.4 shows various possible conditions one could impose.

Table 2.4: Conditions to be satisfied and additional constraints

on the alphabet:	the alphabet Σ is unary;
on the generated language:	$S^* = \{ x_1, x_2, \ldots, x_k \}^*$ is co-finite;
on the generated language:	$S^* = \{ x_1, x_2, \ldots, x_k \}^*$ is co-slender[a];
on the number of words:	$\kappa = 2$;
on the number of words:	κ is fixed;
on the lengths of words:	$x_1, x_2, \ldots, x_k$ are of only two lengths;
on the lengths of words:	$x_1, x_2, \ldots, x_k$ are of only a fixed number of lengths;
on the input words:	$x_1, x_2, \ldots, x_k$ satisfy certain patterns.

[a] See §3.4.3 for the definition and discussion of the co-slender language.

These conditions are not totally independent. For example, over the unary alphabet, as soon as each input word is distinct, a constraint on the number κ of the input words is the same as the constraint on the number of the lengths of the input words. Over larger alphabets, however, the two conditions are not equivalent in general. In addition, S^* is always co-slender over a unary alphabet, so when we impose the latter condition the former condition is automatically satisfied.

Clearly not every combination of the measures of the input, the measures of the output, and imposed conditions leads to a feasible and interesting question. In addition, some combinations result in trivial problems and some result in a well-studied problem. Nevertheless, the one appearing in the definition of the FPFM is not such a combination. Some of the measures were given by Shallit and some of their combinations are discussed in our papers [83, 84] with Shallit and Kao.

2.3 Bounds on the longest omitted words

One aspect of the generalized FP is to find a good upper bound or lower bound on the length of the longest omitted words, in some measure. Since there is no simple polynomial formula for the FP, a general simple formula for the FPFM is also unlikely. Hence the study of upper and lower bounds is a more realistic goal.

Let $S = \{ x_1, x_2, \ldots, x_k \}$ be the input.

Lower bound on the length of the longest omitted words

A lower bound on $\mathrm{llw}\left(\overline{S^*}\right)$ follows immediately from Proposition 2.1.14 as follows:

$$\mathcal{L} = \mathrm{llw}\left(\overline{S^*}\right) \geq g(|\, x_1 \,|, |\, x_2 \,|, \ldots, |\, x_k \,|). \tag{2.17}$$

The lower bound in (2.17) is tight, since the equality in (2.17) can be attained when the set S consists of all words of some particular lengths as shown in Example 2.1.2, where the gcd of those lengths is 1. The lower bound in (2.17), however, is not very useful in practice.

Upper bound on the length of the longest omitted words

To discuss the upper bound on $\mathrm{llw}\left(\overline{S^*}\right)$, we can apply the following proposition and obtain an upper bound from the corresponding result on state complexity.

Proposition 2.3.1. *Let S be a set of words over Σ and M be a DFA of n states accepting the language S^*. If S^* is co-finite, then* $\mathrm{llw}\left(\overline{S^*}\right) < n$.

Proof. Proof by contradiction. Assume w is not in S^* and $|\,w\,| \geq n$. Now we consider all states that M visited when M rejects w. There are $|\,w\,| + 1$ of them, which is $\geq n + 1$. Then there must be one state that M visited at least twice. Suppose $\delta(q_0, u) = \delta(q_0, uv)$, where $v \neq \epsilon$, and $w = uvz$. Then none of the words in uv^*z is in S^*, which contradicts the fact that S^* is co-finite. □

Table 2.5: The length of longest words and the size of computing models

	in $\mathrm{sc}(S)$	in $\mathrm{nsc}(S)$	in $\mathrm{alph}(S)$
$\mathrm{llw}\,(S)$	linear or ∞	linear or ∞	linear or ∞
$\mathrm{llw}\left(\overline{S}\right)$	linear or ∞	exponential or ∞	exponential or ∞
$\mathrm{llw}\left(\overline{S^*}\right)$	exponential or ∞	exponential or ∞	exponential or ∞
$\mathrm{llw}\,(S^*)$	-1 or ∞	-1 or ∞	-1 or ∞

More generally, the relations between the length of the longest words and state complexity (respectively, nondeterministic state complexity, alphabetic length of regular expressions) are given in Table 2.5, where S is given by a DFA (respectively, NFA, regular expression) and ∞ represents the case in which the corresponding language is not finite.

So the two measures of $\mathcal{L}$ and $\mathcal{S}$ are related as follows:

$$\mathcal{L} = \mathrm{llw}\left(\overline{S^*}\right) < \mathcal{S} = \mathrm{sc}(S^*). \tag{2.18}$$

From Eqs. (3.4) and (3.6) for state complexity (which will be discussed in §3.2.1), we have the following upper bounds on $\mathcal{L}$. Let the input be a finite set of words S over Σ. Let $\kappa = |\,S\,|$, $\nu = \max_{1 \leq i \leq \kappa} |x_i|$, $\mu = \sum_{1 \leq i \leq \kappa} |x_i|$. Then

$$\mathcal{L} < 2^{\mathrm{sc}(S)-3} + 2^{\mathrm{sc}(S)-4} = O(2^{\mathrm{sc}(S)}), \quad \text{if } \mathrm{sc}(S) \geq 4; \tag{2.19}$$

$$\mathcal{L} < 2^{\mu-\kappa+1} = O(2^{\mu}); \tag{2.20}$$

$$\mathcal{L} < 2^{\mathrm{nsc}(S)-1} = O(2^{\mathrm{nsc}(S)}) = O(2^{\mathrm{alph}(S)}). \tag{2.21}$$

In particular, by Theorem 3.2.5, Shallit gave the following result in our papers [83, 84] with Shallit and Kao.

Corollary 2.3.2. *[83, 84] Let $S = \{ x_1, x_2, \ldots, x_k \}$. Suppose $|\, x_i \,| \leq \nu$ for all $1 \leq i \leq k$, and S^* is co-finite. Then*

$$\mathcal{L} = \mathrm{llw}\left(\overline{S^*}\right) < \frac{2}{2\,|\Sigma| - 1}(2^{\nu}\,|\Sigma|^{\nu} - 1) = |\Sigma|^{O(\nu)}\,. \tag{2.22}$$

One exciting aspect of the bound in Corollary 2.3.2 is that it is tight. A proof of tightness of those bounds given above cannot be obtained easily from the study of the state complexity, since none of the corresponding constructions, as discussed in §3.2.1, generates a co-finite language. I will present in Chapter 4 examples to show the upper bound in (2.22) to be asymptotically tight, based on the study of a subproblem of the FPFM, the 2FPFM, where the words $x_1, x_2, \ldots, x_k$ are of only two distinct lengths.

Corollary 2.3.2 showed that $\mathcal{L} = \mathrm{llw}\left(\overline{\{ x_1, \ldots, x_k \}^*}\right)$ is exponentially bounded in the measure $\nu = \max_{1 \leq i \leq k} |x_i|$, while we know that in the FP (or equivalently over the unary alphabet) the tight upper bound for the Frobenius number

$$g(x_1, x_2, \ldots, x_k) = O\left(\frac{n^2}{k}\right) \tag{2.23}$$

is quadratic in $n = \max_{1 \leq i \leq k} x_i$. In other measures, such as $\mu = \sum_{1 \leq i \leq k} |\, x_i \,|$ (respectively, $m = \sum_{1 \leq i \leq k} x_i$ for integers), the known upper bounds are also quite different. While most of our upper bounds for the FPFM are exponential, the upper bounds for the FP are only polynomial. We also observed this property in some variations on the Frobenius problem, where a bound can even be doubly-exponential and can be achieved in the case of a free monoid when the corresponding bound in the case of integers is only quadratic. Intuitively, this can be explained by the fact that in the case of integers, the commutative law holds and most of the input measures are polynomially bounded in each other, while in a free monoid there is no commutative law in general and there are measures that are exponential in others.

Bounds on the number, and total symbols, of the longest omitted words

While there is, if any, only one Frobenius number, there can be more than one longest word as a solution to the FPFM. Obviously, the number of the longest words not in S^* could be exponential, since we have

$$\mathcal{I} = \#\{\, w \notin S^* : |\, w \,| = \mathcal{L} \,\} \leq |\Sigma|^{\mathcal{L}} \tag{2.24}$$

The equality in (2.24) can be achieved when the set S consists of all words of lengths $y_1, y_2, \ldots, y_k$ such that $\gcd(y_1, y_2, \ldots, y_k) = 1$, as shown in Example 2.1.2.

In some cases of $x_1, x_2, \ldots, x_k$, the longest word not in $\{ x_1, x_2, \ldots, x_k \}$ is unique, for example over the unary alphabet, while in some other cases, as shown, the cardinality of the longest words can be exponential. So in the entire discussion

of this thesis, without further explanation, we do not differentiate between the singular and plural form of the concept of longest omitted word(s).

Since $\mathcal{IL} = \mathcal{I} \cdot \mathcal{L}$, it follows trivially that

$$\mathcal{L} \leq \mathcal{IL} \leq |\Sigma|^{\mathcal{L}} \mathcal{L}, \tag{2.25}$$

and both equalities can be achieved.

2.4 The FPFM for two lengths — the 2FPFM

In this section, we will discuss a certain natural subproblem of the FPFM. As we saw in Chapter 1, when k, the number of given integers, is fixed and small, the solution to the integer FP becomes feasible. In particular, when $k = 2$, there are formulae for most of the variations on the integer FP. What non-trivial constraint can we impose on the FPFM in order to simplify it? We will examine the case where the input words are of the same length, the case where the input consists of two distinct words, and the case where the input words are of two distinct lengths.

Proposition 2.4.1. *Let $S \subseteq \Sigma^n$ for some integer n. Then S^* is co-finite if and only if $S = \Sigma$. Furthermore, when S^* is co-finite, then $S^* = \Sigma^*$.*

Proof. If $n \neq 1$, then $n \nmid kn + 1$ for all positive integers k. So $(\Sigma^n)^*\Sigma \cap S^* = \emptyset$, which contradicts the fact that S^* is co-finite. Hence $n = 1$ and thus $S = \Sigma$. □

Proposition 2.4.2. *Let x_1, x_2 be two words over Σ. Then $\{ x_1, x_2 \}^*$ is co-finite if and only if either Σ is unary and $\gcd(x_1, x_2) = 1$ or Σ is binary and $\{ x_1, x_2 \} = \Sigma$.*

Proof. Without loss of generality, let $\Sigma = \{ \mathtt{0}, \mathtt{1}, \dots \}$.

$\Leftarrow$: One can verify that, in either case, $\{ x_1, x_2 \}^*$ is co-finite.

$\Rightarrow$: By Proposition 2.1.9, $T = \{ x_1, x_2 \} \cap \{ \mathtt{0} \}^*$ generates a co-finite language in $\{ \mathtt{0} \}^*$ and thus $T \neq \emptyset$. If both $x_1, x_2 \in \{ \mathtt{0} \}^*$, then by the co-finiteness of $\{ x_1, x_2 \}^*$ in $\{ \mathtt{0} \}^*$, we have $\gcd(x_1, x_2) = 1$ and there cannot be any other letter in Σ. If only one of the two words, say x_1, is in $\{ \mathtt{0} \}^*$, then $x_1 = \mathtt{0}$; and similarly by considering the letter $\mathtt{1}$, one can prove that $x_2 = \mathtt{1}$, and there cannot be a third letter in Σ. □

So the two special cases of the FPFM, where the words $x_1, x_2, \dots, x_k$ are of the same length, and where the number of distinct words $\kappa = 2$, can only have trivial solutions. We denote the FPFM with bases consisting solely of words of the same length by *1FPFM*. As we saw in Proposition 2.4.1, for the 1FPFM the generated language is co-finite if and only if the basis is the entire alphabet. A slightly-improved subproblem of the FPFM is that the words $x_1, x_2, \dots, x_k$ are of only two distinct lengths m and n, and we denote by 2FPFM the *Frobenius problem in a free monoid with bases composed of words of two distinct lengths*. By Proposition 2.1.14, we know that $\{ x_1, x_2, \dots, x_k \}^*$ is co-finite only if $\gcd(m, n) = 1$.

2.4.1 Definition of the 2FPFM

We now formally define the 2FPFM.

Problem 2.4.3 (*2FPFM*)**.** *Let Σ be a (finite) alphabet. Given k non-empty words $x_1, x_2, \ldots, x_k \in \Sigma^m \cup \Sigma^n$, where m, n are two distinct positive integers and $m < n$, such that there are only finitely many words that* cannot *be written as concatenations of words in $\{ x_1, x_2, \ldots, x_k \}$, then what is the longest such word(s)?*

In the remaining part of this chapter except the last section, our discussion will focus on the 2FPFM. To begin the discussion, we will first see one trivial type of the 2FPFM.

Proposition 2.4.4. *Let $S \subseteq \Sigma \cup \Sigma^n$. Then S^* is co-finite if and only if $\Sigma \subseteq S$. Furthermore, when S^* is co-finite, then $S^* = \Sigma^*$.*

Proof. $\Leftarrow$: If $\Sigma \subseteq S$, then $S^* = \Sigma^*$ is co-finite.

$\Rightarrow$: Let $\Delta = \Sigma \setminus S$ and $T = S \cap \Delta^*$. Then by Proposition 2.1.9, T generates a co-finite language in Δ^*. Furthermore none of the letters in Δ is in S, so none is in T, and thus the words in T can only be of length n. Then, by Proposition 2.4.1, $T = \Delta$, a contradiction. Therefore, $\Sigma \setminus S = \emptyset$ and thus $\Sigma \subseteq S$. □

Example 2.1.2 on page 30, where $S = \Sigma^3 \cup \Sigma^5$, is in fact an instance of the 2FPFM. Here is another instance of the 2FPFM.

Example 2.4.5. Let $\Sigma = \{ \mathtt{0}, \mathtt{1} \}, m = 3, n = 5$ and $S_4 = \Sigma^3 \cup \Sigma^5 \setminus \{ \mathtt{00001} \}$. Then S_4^* is co-finite and there are in total 222 binary words in $\overline{S_4^*}$, as shown in Table 2.6, where the longest words are

$$\mathtt{0000100000001},\ 0000100100001,\ 0000101000001,\ 0000101100001,$$
$$\mathtt{0000110000001},\ 0000110100001,\ 0000111000001,\ 0000111100001,$$

each of which is of length 13.

Given any set S of words of lengths m and n, then $\mathrm{llw}\left(\overline{S^*}\right)$ is always a Frobenius number of m and some l, where l characterizes the structure of the basis S, and l can be calculated in polynomial time in the measure $\mu = \sum_{w \in S} |\, w \,|$ as the following sections will show.

2.4.2 The First and Second Lemmas of the 2FPFM

I will devote this subsection to two fundamental lemmas about the 2FPFM, which will be used to illustrate the upper bound for the 2FPFM, and from which some results will be derived in later chapters. They are named *the First Lemma of the 2FPFM* and *the Second Lemma of the 2FPFM*, because almost the entire theory of the 2FPFM is based on them. Furthermore, they characterize those sets S such that S^* is co-finite, when S is a set containing words of no more than two distinct lengths.

Table 2.6: All the words in $\{0,1\}^* \setminus (\{0,1\}^3 \cup \{0,1\}^5 \setminus \{00001\})^*$

1	[1]0	57	[7]0100001	113	[7]1011001	169	[10]0000110000
2	[1]1	58	[7]0100010	114	[7]1011010	170	[10]0000110001
3	[2]00	59	[7]0100011	115	[7]1011011	171	[10]0000110010
4	[2]01	60	[7]0100100	116	[7]1011100	172	[10]0000110011
5	[2]10	61	[7]0100101	117	[7]1011101	173	[10]0000110100
6	[2]11	62	[7]0100110	118	[7]1011110	174	[10]0000110101
7	[4]0000	63	[7]0100111	119	[7]1011111	175	[10]0000110110
8	[4]0001	64	[7]0101000	120	[7]1100000	176	[10]0000110111
9	[4]0010	65	[7]0101001	121	[7]1100001	177	[10]0000111000
10	[4]0011	66	[7]0101010	122	[7]1100010	178	[10]0000111001
11	[4]0100	67	[7]0101011	123	[7]1100011	179	[10]0000111010
12	[4]0101	68	[7]0101100	124	[7]1100100	180	[10]0000111011
13	[4]0110	69	[7]0101101	125	[7]1100101	181	[10]0000111100
14	[4]0111	70	[7]0101110	126	[7]1100110	182	[10]0000111101
15	[4]1000	71	[7]0101111	127	[7]1100111	183	[10]0000111110
16	[4]1001	72	[7]0110000	128	[7]1101000	184	[10]0000111111
17	[4]1010	73	[7]0110001	129	[7]1101001	185	[10]0001000001
18	[4]1011	74	[7]0110010	130	[7]1101010	186	[10]0001100001
19	[4]1100	75	[7]0110011	131	[7]1101011	187	[10]0010000001
20	[4]1101	76	[7]0110100	132	[7]1101100	188	[10]0010100001
21	[4]1110	77	[7]0110101	133	[7]1101101	189	[10]0011000001
22	[4]1111	78	[7]0110110	134	[7]1101110	190	[10]0011100001
23	[5]00001	79	[7]0110111	135	[7]1101111	191	[10]0100000001
24	[7]0000000	80	[7]0111000	136	[7]1110000	192	[10]0100100001
25	[7]0000001	81	[7]0111001	137	[7]1110001	193	[10]0101000001
26	[7]0000010	82	[7]0111010	138	[7]1110010	194	[10]0101100001
27	[7]0000011	83	[7]0111011	139	[7]1110011	195	[10]0110000001
28	[7]0000100	84	[7]0111100	140	[7]1110100	196	[10]0110100001
29	[7]0000101	85	[7]0111101	141	[7]1110101	197	[10]0111000001
30	[7]0000110	86	[7]0111110	142	[7]1110110	198	[10]0111100001
31	[7]0000111	87	[7]0111111	143	[7]1110111	199	[10]1000000001
32	[7]0001000	88	[7]1000000	144	[7]1111000	200	[10]1000100001
33	[7]0001001	89	[7]1000001	145	[7]1111001	201	[10]1001000001
34	[7]0001010	90	[7]1000010	146	[7]1111010	202	[10]1001100001
35	[7]0001011	91	[7]1000011	147	[7]1111011	203	[10]1010000001
36	[7]0001100	92	[7]1000100	148	[7]1111100	204	[10]1010100001
37	[7]0001101	93	[7]1000101	149	[7]1111101	205	[10]1011000001
38	[7]0001110	94	[7]1000110	150	[7]1111110	206	[10]1011100001
39	[7]0001111	95	[7]1000111	151	[7]1111111	207	[10]1100000001
40	[7]0010000	96	[7]1001000	152	[10]0000000001	208	[10]1100100001
41	[7]0010001	97	[7]1001001	153	[10]0000100000	209	[10]1101000001
42	[7]0010010	98	[7]1001010	154	[10]0000100001	210	[10]1101100001
43	[7]0010011	99	[7]1001011	155	[10]0000100010	211	[10]1110000001
44	[7]0010100	100	[7]1001100	156	[10]0000100011	212	[10]1110100001
45	[7]0010101	101	[7]1001101	157	[10]0000100100	213	[10]1111000001
46	[7]0010110	102	[7]1001110	158	[10]0000100101	214	[10]1111100001
47	[7]0010111	103	[7]1001111	159	[10]0000100110	215	[13]0000100000001
48	[7]0011000	104	[7]1010000	160	[10]0000100111	216	[13]0000100100001
49	[7]0011001	105	[7]1010001	161	[10]0000101000	217	[13]0000101000001
50	[7]0011010	106	[7]1010010	162	[10]0000101001	218	[13]0000101100001
51	[7]0011011	107	[7]1010011	163	[10]0000101010	219	[13]0000110000001
52	[7]0011100	108	[7]1010100	164	[10]0000101011	220	[13]0000110100001
53	[7]0011101	109	[7]1010101	165	[10]0000101100	221	[13]0000111000001
54	[7]0011110	110	[7]1010110	166	[10]0000101101	222	[13]0000111100001
55	[7]0011111	111	[7]1010111	167	[10]0000101110		
56	[7]0100000	112	[7]1011000	168	[10]0000101111		

Lemma 2.4.6 (*The First Lemma of the 2FPFM*)**.** *[83, 84] Let S be a set of words of lengths m and n, where $0 < m < n$, over the alphabet Σ. If S^* is co-finite, then $\Sigma^m \subseteq S$.*

Proof. If $S^* = \Sigma^*$, then $\Sigma \subseteq S$, since no single letter can be written as the concatenation of more than one word. Hence $m = 1$ and $\Sigma^m = \Sigma \subseteq S$.

Now we assume $S^* \neq \Sigma^*$. Let $x \in \Sigma^m$. Consider the language $x\Sigma^*$. Since S^* is co-finite, $x\Sigma^* \cap S^* \neq \emptyset$. We choose v such that $xv \in S^*$. Then there is a factorization of xv of the form

$$xv = y_1 y_2 \cdots y_j, \tag{2.26}$$

where all the y_i are in S for $1 \leq i \leq j$ and thus each y_i is of length either m or n. If $y_1 \in \Sigma^m$, then by comparing lengths, we have $x = y_1$, and thus $x \in S$. Otherwise $y_1 \in \Sigma^n$. By the twins proposition, there exists $z \in S$ such that y_1 is a proper prefix of z, or vice versa. But since S contains words of only lengths m and n, and $y_1 \in \Sigma^n$, we must have $z \in \Sigma^m$, and z is a proper prefix of y_1. Then by comparing lengths, we have $x = z$, and so $x \in S$. □

Proposition 2.4.4 can be viewed as a corollary of the First Lemma of the 2FPFM. Before giving the Second Lemma of the 2FPFM, we will first prove a weaker version of that lemma, by restricting the lengths m and n.

Lemma 2.4.7 (*The Second Lemma of the 2FPFM, weaker version*)**.** *[83, 84] Let S be a set of words of lengths m and n over Σ, where $0 < m < n < 2m$. If S^* is co-finite, then $\Sigma^l \subseteq S^*$, where $l = m\,|\Sigma|^{n-m} + n - m$.*

Proof. Proof by contradiction. Let x be a word of length l that is not in S^*. Then we can write x uniquely as

$$x = y_0 z_0 y_1 z_1 \cdots y_{|\Sigma|^{n-m}-1} z_{|\Sigma|^{n-m}-1} y_{|\Sigma|^{n-m}}, \tag{2.27}$$

where $y_i \in \Sigma^{n-m}$ for $0 \leq i \leq |\Sigma|^{n-m}$, and $z_i \in \Sigma^{2m-n}$ for $0 \leq i < |\Sigma|^{n-m}$.

Now suppose that $y_i z_i y_{i+1} \in S$ for some i with $0 \leq i < |\Sigma|^{n-m}$. Then we can write

$$x = \left(\prod_{0 \leq j < i} (y_j z_j) \right) (y_i z_i y_{i+1}) \left(\prod_{i+1 \leq k \leq |\Sigma|^{n-m}} (z_k y_k) \right). \tag{2.28}$$

Note that each factor $y_j z_j$ and $z_k y_k$ is of length m, which, by the First Lemma of the 2FPFM, is in S. Hence all terms in the factorization (2.28) are in S and thus $x \in S^*$, a contradiction. It follows that

$$y_i z_i y_{i+1} \notin S \quad \text{for all } i \text{ with } 0 \leq i < |\Sigma|^{n-m}. \tag{2.29}$$

Now the factorization of x in Eq. (2.27) uses $|\Sigma|^{n-m}+1$ words y_i, and there are only $|\Sigma|^{n-m}$ distinct words of length $n-m$. So, by the pigeonhole principle, we have $y_p = y_q$ for some $0 \leq p < q \leq |\Sigma|^{n-m}$. Now define

$$u = y_0 z_0 \cdots y_{p-1} z_{p-1}, \quad v = y_p z_p \cdots y_{q-1} z_{q-1}, \quad w = y_q z_q \cdots y_{|\Sigma|^{n-m}}. \tag{2.30}$$

Then $x = uvw$ and $v \neq \epsilon$. Consider the language uv^*w. Since S^* is co-finite, $uv^*w \cap S^* \neq \emptyset$. There exists a $k \geq 0$ such that $uv^k w \in S^*$.

Now let $uv^k w = x_1 x_2 \cdots x_j$ be a factorization into words in S. Then x_1 is a word of length either m or n. If $|\, x_1 \,| = n$, then comparing lengths gives $x_1 = y_0 z_0 y_1$. But by Eq. (2.29) we know $y_0 z_1 y_1 \notin S$. So $|x_1| = m$, and comparing lengths gives $x_1 = y_0 z_0$. By similar reasoning we see that $x_2 = y_1 z_1$, and so on. Hence finally $x_j = y_{|\Sigma|^{n-m}-1} z_{|\Sigma|^{n-m}-1} y_{|\Sigma|^{n-m}} \in S$. But this again contradicts (2.29).

Therefore, our assumption that $x \notin S^*$ must be false, and so $x \in S^*$. Since x was arbitrary, this proves the result. □

Example 2.4.8. Let Σ be a binary alphabet. If S contains words of lengths $n-1$ and n, and S^* is co-finite, then all words of length

$$l_1 = 2(n-1)+1 = 2n-1 = (n-1)+n \tag{2.31}$$

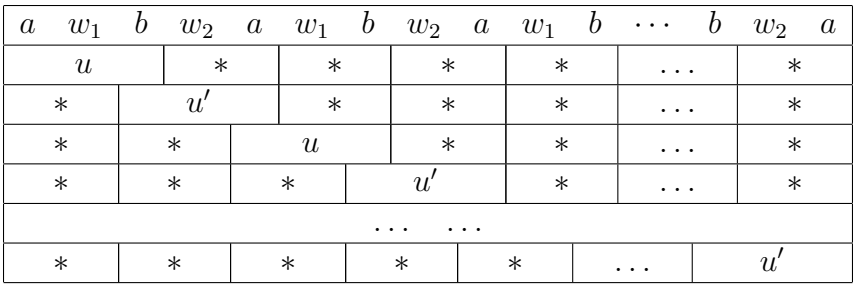
can be factorized as concatenations of words in S. To see this, let $w = aw_1bw_2c = uv = vu'$, where $a, b, c \in \Sigma$, $w_1, w_2 \in \Sigma^{n-2}$, $u, u' \in \Sigma^n$ and $v, v' \in \Sigma^{n-1}$. All possible factorizations of w are as follows:

a	w_1	b	w_2	c
u			v	
v'			u'	

If $b = c$, by the co-finiteness of S^*, we have $aw_1b(w_2b)^* \cap S^* \neq \emptyset$. The possible factorizations of a word in $aw_1b(w_2b)^*$ are shown below by considering the first factor of length n, any of which can be used to factorize w.

a	w_1	b	w_2	b	w_2	b	$\cdots$	b	w_2	b
u			$*$		$*$		$\ldots$		$*$	
$*$		u'			$*$		$\ldots$		$*$	
$*$		$*$		u'			$\ldots$		$*$	
$\ldots$		$\ldots$								
$*$		$*$		$*$		$\ldots$		u'		

The case $a = b$ is similar to $b = c$. If $a = c$, by the co-finiteness of S^*, then $a(w_1bw_2a)^* \cap S^* \neq \emptyset$. The possible factorizations of a word in $a(w_1bw_2a)^*$ are shown below by considering the first factor of length n, any of which can be used to factorize w.

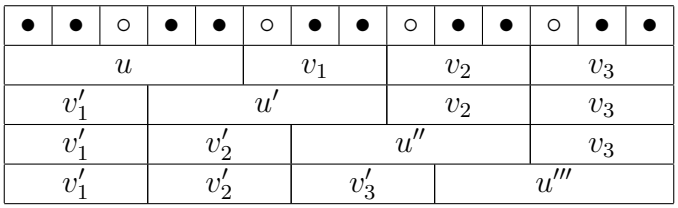


In all cases, w can be factorized over the basis S.

Example 2.4.9. Let Σ be a binary alphabet. If S contains words of lengths 3 and 5, and S^* is co-finite, then all words of length of

$$l_2 = 2^{5-3} \cdot 3 + (5-3) = 4 \cdot 3 + 2 = 14 \tag{2.32}$$

can be factorized. To see this, let w be a word of length 14. Then all possible factorizations of w are as follows (•'s and ∘'s are only labels, and the letters may not be the same)



Since there are 5 ••'s, by the pigeonhole principle, at least two of them are identical. By similar discussion, if S^* is co-finite, then w is in S^*.

Proposition 2.4.10. *Let S be a set that contains all words of length m and possibly some words of length n over Σ, where $0 < m < n$, and let $x = a_1 a_2 \cdots a_l$ be a word of length l, where $l = n + jm$ for some $j \geq 0$. Then $x \in S^*$ if and only if at least one of the words $y_i = a_{im+1} a_{im+2} \cdots a_{im+n}$, for $0 \leq i \leq j$, is in S.*

Proof. If $m \mid n$, then for all x of length l, since $\Sigma^m \subseteq S$, it follows that x and all the y_i are in S for $0 \leq i \leq j$. The result holds. Now, we assume $m \nmid n$.

Suppose $x \in S^*$. Let $x = w_0 w_1 \cdots w_h$ be a factorization of x into the words in S. Since $m \nmid n$, at least one of the w's is of length n. Let w_k be the first such factor. By comparing lengths, $w_k = a_{km+1} a_{km+2} \cdots a_{km+n} = y_k$ is in S, and thus y_k is the desired word. This proves one direction.

For the converse, suppose $y_k = a_{km+1} a_{km+2} \cdots a_{km+n} \in S$ for some k with $0 \leq k \leq j$. Then we can write

$$x = \Big(\prod_{0 \leq i \leq k-1} (a_{im+1} a_{im+2} \cdots a_{im+m}) \Big) y_k \Big(\prod_{k \leq i \leq j-1} (a_{im+n+1} a_{im+n+2} \cdots a_{im+n+m}) \Big). \tag{2.33}$$

Since $\Sigma^m \subseteq S$, each term in the factorization (2.33) is in S. Hence $x \in S^*$. □

Lemma 2.4.11 (*The Second Lemma of the 2FPFM*)**.** *Let S be a set of words of lengths m and n, where $0 < m < n$, over the alphabet Σ. If S^* is co-finite, then $\Sigma^l \subseteq S^*$, where $l = m\,|\Sigma|^{n-m} + n - m$.*

Proof. Let $x = a_1 a_2 \cdots a_l$ be a word of length l that is not in S^*. We define

$$y_i = a_{(i-1)m+1} a_{(i-1)m+2} \cdots a_{(i-1)m+n}, \qquad \text{for } 1 \le i \le |\Sigma|^{n-m}; \tag{2.34}$$

$$z_i = a_{im+1} a_{im+2} \cdots a_{im+n-m}, \qquad \text{for } 0 \le i \le |\Sigma|^{n-m}. \tag{2.35}$$

Their positions are illustrated in Figure 2.1. (When $m < n < 2m$, no z overlaps with other z's. In generally, arbitrarily many y's and z's can overlap together.)

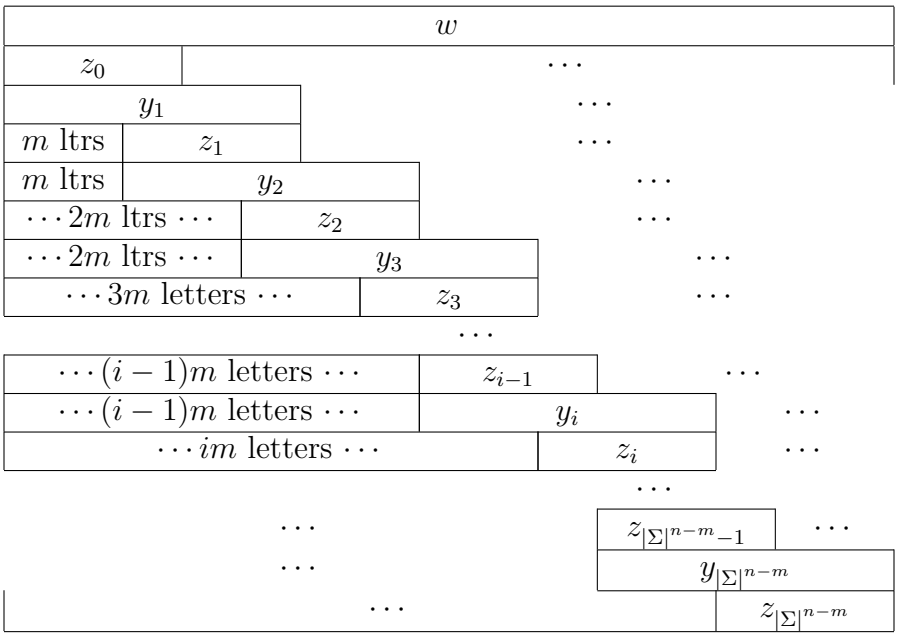


Figure 2.1: Position of factors in the proof of the Second Lemma of the 2FPFM

Since S^* is co-finite, by the First Lemma of the 2FPFM, $\Sigma^m \subseteq S$. Then, by Proposition 2.4.10, none of the y_i for $1 \le i \le |\Sigma|^{n-m}$ is in S.

There are only $|\Sigma|^{n-m}$ distinct words of length $n-m$ over Σ. By the pigeonhole principle, we have

$$z_p = z_q, \qquad \text{for some } 0 \le p < q \le |\Sigma|^{n-m}. \tag{2.36}$$

Now we define

$$u = a_1 a_2 \cdots a_{pm}, \quad v = a_{pm+1} a_{pm+2} \cdots a_{qm}, \quad w = a_{qm+1} a_{qm+2} \cdots a_l. \tag{2.37}$$

Then $x = uvw$ and $v \neq \epsilon$. Since S^* is co-finite, $uv^*w \cap S^*$ is not empty. Let k be the smallest positive exponent such that $uv^k w \in S^*$. Since $x = uvw \notin S$, we have $k \ge 2$. Now let

$$uv^k w = x_1 x_2 \cdots x_j \tag{2.38}$$

be a factorization into elements of S. If $|\,x_1\,| = n$, then comparing lengths gives $x_1 = y_1 \notin S$, a contradiction. So $|\,x_1\,| = m$. By similar reasoning we see that $|x_2| = m$, and so on. Notice that $x = uvw$ and uv^kw agree on the first $|uv|+(n-m)$ letters, since $z_q = z_p$. So all factors from x_1 to x_q are of length m. We can write

$$u = x_1x_2\cdots x_p, \quad v = x_{p+1}x_{p+2}\cdots x_q. \tag{2.39}$$

By removing the leftmost copy of v from uv^kw, the new word

$$uv^{k-1}w = x_1x_2\cdots x_px_{q+1}x_{q+2}\cdots x_j \tag{2.40}$$

is also in S^*, where $k-1 \geq 1$. This contradicts the minimality of k. □

2.4.3 Bounds on the longest omitted words for the 2FPFM

Let $S = \{\, x_1, x_2, \ldots, x_k \,\}$ be a set of words over the alphabet Σ such that $S \subseteq \Sigma^m \cup \Sigma^n$, where $0 < m < n$, and S^* is co-finite. The lower bound for the FPFM on $\mathrm{llw}\left(\overline{S^*}\right)$ given by Proposition 2.1.14, namely,

$$\mathcal{L} = \mathrm{llw}\left(\overline{S^*}\right) \geq g(m,n), \tag{2.41}$$

still holds and it is tight for the 2FPFM as well. For any two positive integers m, n with $\gcd(m,n) = 1$, let

$$S = \Sigma^m \cup \Sigma^n. \tag{2.42}$$

Then S^* is co-finite and $\mathrm{llw}\left(\overline{(\Sigma^m \cup \Sigma^n)^*}\right) = g(m,n)$. We will see in §2.5 that for all other cases $\mathcal{L}$ is strictly greater than the Frobenius number $g(m,n)$.

Furthermore, the exponential bound in the FPFM on the number of the longest words not in S^*, namely

$$\mathcal{I} = \#\{\, w \notin S^* : |\,w\,| = \mathcal{L} \,\} \leq |\Sigma|^{\mathcal{L}}, \tag{2.43}$$

can also be achieved for the 2FPFM by the same example when the set S contains all words of lengths m and n. In this case, the equality on the right hand side of the bound

$$\mathcal{L} \leq \mathcal{I}\mathcal{L} \leq |\Sigma|^{\mathcal{L}}\,\mathcal{L}, \tag{2.44}$$

can be achieved as well. In the 2FPFM, there are also examples of bases S such that each S^* is co-finite and the longest word not in S^* is unique, in which case equality on the left-hand side in (2.44) can be achieved.

Now I will provide an upper bound on the length of the longest words not in S^*, where S contains words of at most two distinct lengths. The upper bound is derived from the First and the Second Lemmas of the 2FPFM. A weaker version of Theorem 2.4.12, by restricting $m < n < 2m$, appeared in our papers [83, 84].

Theorem 2.4.12. *Let S be a set of words of lengths m and n, where $0 < m < n$, over the alphabet Σ. If S^* is co-finite, then*

$$\operatorname{llw}\left(\overline{S^*}\right) \leq g(m,l) = ml - m - l, \tag{2.45}$$

where $l = m|\Sigma|^{n-m} + n - m$.

Proof. If S^* is co-finite, by Proposition 2.1.14, $\gcd(m,n) = 1$. By the First Lemma of the 2FPFM, we have $\Sigma^m \subseteq S \subseteq S^*$; by the Second Lemma of the 2FPFM, we have $\Sigma^l \subseteq S^*$, where $l = m|\Sigma|^{n-m} + n - m$. Since S^* contains all words of lengths m and l, and $\gcd(m,l) = 1$, by Proposition 2.1.4, the length of the longest words not in S^* is $\leq g(m,l) = ml - m - l$. □

Corollary 2.4.13. *Let S be a set of words of lengths m and n, where $0 < m < n$ and $\gcd(m,n) = 1$, over the alphabet Σ. Then S^* is co-finite if and only if $\Sigma^m \subseteq S$ and $\Sigma^l \subseteq S^*$, where $l = m|\Sigma|^{n-m} + n - m$.*

Proof. If S^* is co-finite, for the same reason as in the proof of Theorem 2.4.12, we get $\Sigma^m \subseteq S$ and $\Sigma^l \subseteq S^*$. On the other hand, if $\Sigma^m \subseteq S$ and $\Sigma^l \subseteq S^*$, then since $\gcd(m,l) = 1$, by Proposition 2.1.4, S^* is co-finite. □

2.5 Combinatorics on words in the 2FPFM

We will discuss the 2FPFM in terms of combinatorics on words, and provide an equivalent condition on co-finiteness, which is one of the supporting results for a theorem, called the spectrum theorem, appearing in §2.6.4.

2.5.1 Boosting the length of omitted words

First we need one technical lemma, which allows us, from one word of a particular length not in S^*, to construct a longer word that is also not in S^*. (All longest omitted words can be constructed by this lemma; see Corollary 2.5.5.) A weaker version of Lemma 2.5.1 (with the restriction $m < n < 2m$) appeared in our papers [83, 84].

Lemma 2.5.1. *Suppose $S \subseteq \Sigma^m \cup \Sigma^n, 0 < m < n$, and S^* is co-finite. Let T be a set of words that are not in S^*, where each word is of length $\equiv n \pmod m$. Then*

$$S^* \cap (T\Sigma^m)^{i-1}T = \emptyset \tag{2.46}$$

for $1 \leq i \leq m-1$.

Proof. Since S^* is co-finite, by Proposition 2.1.14, then $\gcd(m,n) = 1$. Define

$$L_i = S^* \cap (T\Sigma^m)^{i-1}T, \quad \text{for } 1 \leq i < m. \tag{2.47}$$

Now we will prove that $S^* \cap L_i = \emptyset$ by induction on i, where $1 \le i < m$.

The base case is $i = 1$. In this case, $S^* \cap L_1 = S^* \cap T = \emptyset$.

Now we suppose $S^* \cap L_i = \emptyset$ for some i, where $1 \le i \le m-2$, and we prove that $S^* \cap L_{i+1} = \emptyset$.

First we show that $S^* \cap \Sigma^k L_i = \emptyset$ for all $0 \le k \le n$ such that $k \equiv n \pmod m$. Assume $uw \in S^*$ for some $u \in \Sigma^k, w \in L_i$. Then there is a factorization

$$uw = y_1 y_2 \cdots y_t, \tag{2.48}$$

where all the y's are in S. Consider the length of uw, which is

$$|\,uw\,| = |\,u\,| + |\,w\,| \equiv n + ni = n(i+1) \pmod m. \tag{2.49}$$

<table>
<tr><td colspan="2">u</td><td colspan="5">w</td></tr>
<tr><td>y_1</td><td>y_2</td><td colspan="2">$\cdots$</td><td>y_{r-1}</td><td colspan="2">y_r</td><td>$y_{r+1} \cdots y_t$</td></tr>
<tr><td colspan="2">u</td><td>z_1</td><td>z_2</td><td colspan="2">$\cdots$</td><td>$z_{r'-1}$</td><td>$z_{r'}$</td><td>$y_{r+1} \cdots y_t$</td></tr>
</table>

Since $2 \le i+1 \le m-1$, m does not divide $i+1$, and thus $i+1 \not\equiv 0 \pmod m$. Then, since $\gcd(n, m) = 1$, it follows that $n(i+1) \not\equiv 0 \pmod m$, and thus m does not divide $|\,uw\,|$. Hence at least one of the y's is of length n. Let r be the smallest index such that $|\,y_r\,| = n$. Consider the word $y_1 \cdots y_{r-1} y_r$, which is of length $n + m(r-1)$. Since $|\,u\,| < n$ and $u \equiv n \pmod m$, by Eq. (2.48), comparing lengths gives $y_1 \cdots y_{r-1} y_r \in u(\Sigma^m)^*$. By the First Lemma of the 2FPFM, all words of length m are in S^* and thus we can write $y_1 \cdots y_{r-1} y_r = u z_1 \cdots z_{r'}$, where all the z's are of length m and in S. Then

$$uw = y_1 \cdots y_{r-1} y_r \; y_{r+1} \cdots y_t = u z_1 \cdots z_{r'} \; y_{r+1} \cdots y_t, \tag{2.50}$$

which, by canceling u from the left on both sides, gives a factorization of w, and contradicts the induction hypothesis $S^* \cap L_i = \emptyset$.

<table>
<tr><td colspan="4">τ</td><td>α</td><td colspan="2">ω</td></tr>
<tr><td>g_1</td><td>$\cdots$</td><td colspan="2">$g_{r'}$</td><td>$\cdots$</td><td>α</td><td colspan="2">$\cdots$</td></tr>
<tr><td>g_1</td><td colspan="2">$\cdots$</td><td colspan="2">g_{j+1}</td><td>α</td><td colspan="2">$\cdots$</td></tr>
<tr><td>g_1</td><td colspan="3">$\cdots$</td><td colspan="2">g_{j+2}</td><td colspan="2">$\cdots$</td></tr>
<tr><td>g_1</td><td colspan="4">$\cdots$</td><td colspan="2">$g_{r'}$</td><td>$\cdots$</td></tr>
<tr><td>g_1</td><td colspan="5">$\cdots$</td><td colspan="2">$g_{r'}$</td><td>$\cdots$</td></tr>
<tr><td>g_1</td><td></td><td></td><td colspan="3">$\cdots$</td><td></td><td>$\cdots$</td></tr>
</table>

Figure 2.2: Position of factors in the proof of the boosting lemma

Now we prove that $S^* \cap L_{i+1} = \emptyset$. Otherwise, since $L_{i+1} = T\Sigma^m L_i$, there exist words $\tau \in T, \alpha \in \Sigma^m$ and $\omega \in L_i$ such that $\tau\alpha\omega \in S^*$. Then $|\tau| = n + jm$ for some integer j, and there is a factorization

$$\tau\alpha\omega = g_1 g_2 \cdots g_{t'}, \tag{2.51}$$

where all the g's are in S. Consider the first g of length n in the factorization, if any, say $g_{r'}$. Now we consider several cases.

1. If $r' \leq j+1$, then by comparing lengths, we have

$$\tau = g_1 \cdots g_{r'-1} g_{r'} \ \cdots = g_1 \cdots g_{r'-1} g_{r'} \ g'_1 \cdots g'_{j-r'+1} \tag{2.52}$$

for some g''s of length m, which shows $\tau \in S^*$ and contradicts $S^* \ \cap \ T = \emptyset$.

2. If $r' = j+2$, then by comparing lengths, we have

$$\tau\alpha = g_1 g_2 \cdots g_{j+2}, \quad \text{and} \quad \omega = a_{j+3} \cdots a_r \in L_i, \tag{2.53}$$

which shows $\omega \in S^*$ and contradicts the induction hypothesis $S^* \ \cap \ L_i = \emptyset$.

3. If $j+3 \leq r' \leq j + \lceil \frac{n}{m} \rceil$,

$$\omega = \cdots \ g_{r'} g_{r'+1} \cdots g_{t'} = g'_1 g'_2 \cdots g'_{r'-j-2} \ g_{r'+1} \cdots g_{t'}, \tag{2.54}$$

for some g''s of length m, which again contradicts our induction hypothesis $S^* \ \cap \ L_i = \emptyset$.

4. Finally, we consider the case where $r' > j + \lceil \frac{n}{m} \rceil$ or none of the g's is of length n. In either situation, the first $j + \lceil \frac{n}{m} \rceil$ g's are all of length m and thus

$$g_{j+\lceil \frac{n}{m} \rceil + 1} \cdots g_{t'} \in \Sigma^{n \bmod m} L_i, \tag{2.55}$$

which contradicts the result $S^* \cap \Sigma^k L_i = \emptyset$ for all $0 \leq k \leq n$ such that $k \equiv n \pmod m$.

In every case, there is a contradiction. So $S^* \ \cap \ L_{i+1} = \emptyset$, and the lemma is proved. □

2.5.2 The structure of omitted words

We also need another technical lemma, which specifies the structure of all omitted words. Any word that is not in S^* must be either of a specific length or can be expressed in a factorization, where each factor is of a specific length and none of the factors is in S^*.

Lemma 2.5.2. *Suppose $S \subseteq \Sigma^m \cup \Sigma^n, 0 < m < n$, and S^* is co-finite. Let w be a word that is not in S^* and the length of w is in $\langle m, n \rangle$. Then w can be written in the form*

$$w = \tau_1 u_1 \tau_2 u_2 \cdots \tau_t, \tag{2.56}$$

where all the u's are of length m, all the τ's are of lengths in $\{ n-m, n, n+m, \ldots \}$, none of the τ's is in S^*, and t is the smallest positive integer such that $tn \equiv |\, w \,| \pmod m$.

Proof. Since S^* is co-finite, by the First Lemma of the 2FPFM, all words of length m are in S. So, if a word w is not in S^*, its length cannot be a multiple of m. Furthermore, any integer in $\langle m, n\rangle$ is of the form $pm + qn$, where $p, q \geq 0$. Hence we can write $|\, w \,| = rm + tn$, where $1 \leq t < m$, $r \geq 0$.

We now prove the lemma by induction on the length of w. The base case is $|\, w \,| = n$. In this case, w is already of the form $w = \tau_1 u_1 \tau_2 u_2 \cdots \tau_t$, where $\tau_1 = w$ and $t = 1$. Now we assume the result is true for all cases with word length less than $|\, w \,|$, and we prove it is true for w.

Let y_i be the prefix of w of length $n + im$, and z_i be the suffix of w such that $w = y_i z_i$. Consider the prefix sequence $y_0, y_1, \ldots, y_r$.

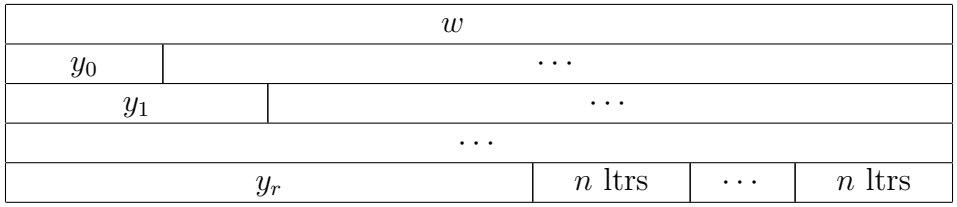


Figure 2.3: Position of factors in the proof of the structure lemma

If $y_0 = w[1..n] \in S^*$, then $z_0 = w[n+1..\,|\, w \,|] \notin S^*$. Otherwise $w = y_0 z_0$ is in S^* and this contradicts $w \notin S^*$. In addition, the length of z_0 is $rm + (t-1)n$, which is in $\langle m, n\rangle$. By the induction hypothesis, $z_0 = \tau_1 u_1 \cdots \tau_{t-1}$ for some τ's and u's that satisfy Eq. (2.56) for z_0. On the other hand, S^* is co-finite, so $\gcd(m, n) = 1$, and thus none of the words of length $n-m$ is in S^*. In particular, the prefix $y_0[1..n-m]$ is not in S^*. Then the following factorization

$$w = y_0[1..n-m]\ y_0[n-m+1..n]\ \tau_1\ u_1\ \cdots\ \tau_{t-1} \tag{2.57}$$

is the required form in Eq. (2.56) for w.

If $y_r = w[1..n+rm] \notin S^*$, then we can write $w = y_r v_1 v_2 \cdots v_{n-1}$, where each of the v's is of length n. Then one can verify that the form

$$w = y_r\ v_1[1..m]\ v_1[m+1..n]\ \cdots\ v_{n-1}[1..m]\ v_{n-1}[m+1..n] \tag{2.58}$$

is what is required.

Now we assume $y_0 \notin S^*$ and $y_r \in S^*$. Then there must be an integer i such that $0 \leq i < r$, $y_i = w[1..n+im] \notin S^*$, and $y_{i+1} = w[1..n+(i+1)m] \in S^*$. Since $w = y_{i+1} z_{i+1} \notin S^*$, it follows that $z_{i+1} \notin S^*$. The length of z_{i+1} is $(r-i-1)m + (t-1)n$, which is in $\langle m, n\rangle$. By the induction hypothesis, $z_0 = \tau_1 u_1 \cdots \tau_{t-1}$ for some τ's and u's that satisfy Eq. (2.56) for z_0. Then w can be written as

$$w = y_i\ w[n+(i-1)m+1..n+im]\ \tau_1\ u_1\ \cdots\ \tau_{t-1}. \tag{2.59}$$

Therefore, the lemma is proved. □

2.5.3 An equivalent condition on co-finiteness

Let S be a set of words of two lengths m and n, where $m < n$ and $\gcd(m, n) = 1$. As we observed in Corollary 2.4.13, the basis S generates a co-finite language if and only if all words of length m are in S and all words of length $l = m\,|\Sigma|^{n-m} + n - m$ can be factorized into elements of S. Now I will give an equivalent condition, under which S generates a co-finite language. Furthermore, the instance of the 2FPFM specified by S can be solved accordingly. If $m = 1$, by Proposition 2.4.4, S^* is co-finite if and only if $\Sigma \subseteq S$, and when S^* is co-finite, $S^* = \Sigma^*$. So now we assume $m > 1$.

Theorem 2.5.3. *Let S be a set of words of lengths m and n over the alphabet Σ, where $1 < m < n$ and $\gcd(m, n) = 1$. Then S^* is co-finite if and only if S contains all words of length m and there exists an integer l such that $l \equiv n \pmod m$, $\Sigma^{l-m} \setminus S^* \neq \emptyset$, and $\Sigma^l \setminus S^* = \emptyset$. Furthermore, when S^* is co-finite, this l is unique and $\mathrm{llw}\left(\overline{S^*}\right) = g(m, l)$.*

Proof. $\Rightarrow$: Suppose S^* is co-finite. By the First Lemma of the 2FPFM, S contains all words of length m. Now we prove the existence of l. No word of length $n-m$ can have any factor of length n. Furthermore, $\gcd(m, n) = 1$, so $\Sigma^{n-m} \setminus S^* \neq \emptyset$. On the other hand, by the Second Lemma of the 2FPFM, there exists $l' = m|\Sigma|^{n-m}+n-m$ such that $\Sigma^{l'} \setminus S^* = \emptyset$. Then consider the sequence

$$\Sigma^{n-m} \setminus S^*,\ \Sigma^n \setminus S^*,\ \Sigma^{n+m} \setminus S^*,\ \cdots,\ \Sigma^{l'} \setminus S^*. \tag{2.60}$$

There must be two consecutive terms such that the former is non-empty and the latter is empty. Therefore, there exists l between n and l' such that $l \equiv n \pmod m$, $\Sigma^{l-m} \setminus S^* \neq \emptyset$, and $\Sigma^l \setminus S^* = \emptyset$.

$\Leftarrow$: Suppose such an integer l exists. Since $\Sigma^m \subseteq S$, $\Sigma^l \subseteq S^*$, and $\gcd(l, m) = \gcd(n, m) = 1$, by Proposition 2.1.4, it follows that S^* is co-finite.

Now, suppose S^* is co-finite, and l is one integer satisfying the condition. Then

$$\mathrm{llw}\left(\overline{S^*}\right) \leq g(m, l). \tag{2.61}$$

Let $T = \Sigma^{l-m} \setminus S^*$ and $L = (T\Sigma^m)^{m-2}T$. All words in T are of length $\equiv n \pmod m$. Applying Lemma 2.5.1, we get that $L \cap S^* = \emptyset$, where all words in L are of length

$$(l - m + m)(m - 2) + (l - m) = lm - l - m = g(m, l). \tag{2.62}$$

So the bound in (2.61) can be actually achieved and $\mathrm{llw}\left(\overline{S^*}\right) = g(m, l)$.

To prove uniqueness, suppose there are two such l_1, l_2. Then the length of the longest words not in S^* is $g(m, l_1) = g(m, l_2)$, and thus $ml_1 - m - l_1 = ml_2 - m - l_2$. So $(m - 1)(l_1 - l_2) = 0$. Since $m > 1$, we have $l_1 = l_2$. Hence, l is unique. □

Example 2.5.4. For the basis $S_4 = \Sigma^3 \cup \Sigma^5 \setminus \{\,\texttt{00001}\,\}$ in Example 2.4.5 on page 42, the unique l is 8 as $\Sigma^5 \setminus S_4^* = \{\,\texttt{00001}\,\} \neq \emptyset$ and $\Sigma^8 \setminus S_4^* = \emptyset$. So S_4^* is co-finite, and $\mathrm{llw}\left(\overline{S_4^*}\right) = g(3, 8) = 13$.

Theorem 2.5.3 provides an equivalent condition for co-finiteness in the 2FPFM and converts the 2FPFM into a simpler problem. Instead of finding the longest words not in S^*, one can find the longest words not in S^* with lengths $\equiv n \pmod m$ and then construct a set of longest words not in S^*. Suppose S is the basis for an instance of the 2FPFM. Then S^* is co-finite if and only if for some l all of the sets $\Sigma^{n-m}\backslash S^*, \Sigma^n\backslash S^*, \ldots, \Sigma^{l-m}\backslash S^*$ are non-empty and all of the sets $\Sigma^l\backslash S^*, \Sigma^{l+m}\backslash S^*, \ldots$ are empty. A set of longest words not in S^* can be constructed accordingly, if such an l exists. In fact, this constructed set contains precisely all the longest words not in S^*.

Corollary 2.5.5. *Let S be a set of words of lengths m and n over the alphabet Σ, where $1 < m < n$, and S^* is co-finite, where l is the unique integer such that $l \equiv n \pmod m$, $T = \Sigma^{l-m} \setminus S^* \neq \emptyset$, and $\Sigma^l \setminus S^* = \emptyset$. Then the set of the longest words not in S^* is*

$$(T\Sigma^m)^{m-2}T. \tag{2.63}$$

Proof. By the proof of Theorem 2.5.3, none of the words in $(T\Sigma^m)^{m-2}T$ is in S^* and they are the longest such words, which are of length $g(m,l)$. Furthermore, all of the sets $\Sigma^{n-m} \setminus S^*, \Sigma^n \setminus S^*, \ldots, \Sigma^{l-m} \setminus S^*$ are non-empty, and all of the sets $\Sigma^l \setminus S^*, \Sigma^{l+m} \setminus S^*, \ldots$ are empty.

Let w be any longest word not in S^*. Then w is of length $g(m,l) = ml - m - l$. The smallest positive integer t such that

$$tn \equiv g(m,l) \pmod m \tag{2.64}$$

is $t = m-1$, since by the co-finiteness of S^*, $\gcd(m,n) = 1$. By Lemma 2.5.2, there are words $u_1, u_2, \ldots, u_{m-2}$ of length m, and $\tau_1, \tau_2, \ldots, \tau_{m-1}$ of length $\equiv n \pmod m$ such that

$$w = \tau_1 u_1 \tau_2 u_2 \cdots \tau_{m-1}, \tag{2.65}$$

and none of the τ's is in S^*. Then the length of each τ's is $\leq l - m$ and thus the length of w is

$$\leq (l-m)(m-1) + m(m-2) = lm - l - m = g(m,l), \tag{2.66}$$

where the equality can be attained only when all the τ's are of length $l-m$. Hence all the τ's are in T, and thus w is in $(T\Sigma^m)^{m-2}T$. □

Example 2.5.6. For the basis $S_4 = \Sigma^3 \cup \Sigma^5 \setminus \{\,\texttt{00001}\,\}$, the unique integer l is 8. Since $\Sigma^5 \setminus S^* = \{\,\texttt{00001}\,\}$, the longest words not in S_4^* are $\texttt{00001}\Sigma^3\texttt{00001}$.

Corollary 2.5.5 shows that the 2FPFM is equivalent to the following problem.

Problem 2.5.7 (Equivalent statement of the 2FPFM, the 1st)**.** *Let Σ be a (finite) alphabet and let S be a set of non-empty words of lengths m and n, where m, n are two positive integers and $1 < m < n$, such that $\Sigma^m \subseteq S$ and there exists an integer l' such that $l' \equiv n \pmod m, \Sigma^{l'} \subseteq S^*$. Find the set T of words of some length l such that $l \equiv n \pmod m$, $\Sigma^l \setminus S^* = \emptyset$ and $T = \Sigma^{l-m} \setminus S^*$.*

Theorem 2.5.3 shows an equivalence between the 2FPFM and a combinatorics problem considering only words of particular lengths. The longest omitted words for the former problem can be expressed by words of a particular length in the latter problem. In fact, all words not in a generated co-finite language in the former problem can be expressed by words of particular lengths in the latter problem, as the following corollary says.

Corollary 2.5.8. *Let S be a set of words of lengths m and n over the alphabet Σ, where $1 < m < n$, and S^* is co-finite, where l is the unique integer such that $l \equiv n \pmod{m}$, $\Sigma^{l-m} \setminus S^* \neq \emptyset$, and $\Sigma^l \setminus S^* = \emptyset$. Then the set of words not in S^* is*

$$\left(\bigcup_{j \notin \langle m,n \rangle} \Sigma^j \right) \bigcup \left(\bigcup_{i=0}^{m-2} (T\Sigma^m)^i T \right), \tag{2.67}$$

where $T = \Sigma^{n-m} \cup \Sigma^n \cup \Sigma^{n+m} \cup \cdots \cup \Sigma^{l-m} \setminus S^*$.

Proof. For any $j \notin \langle m, n \rangle$, by Eq. (2.8) on page 33, none of the words of length j is in S^*. On the other hand, none of the words in T is in S^*. Furthermore, each word in T is of length $\equiv n \pmod{m}$, and so by Lemma 2.5.1, none of the words in $\bigcup_{i=0}^{m-2} (T\Sigma^m)^i T$ is in S^*.

Let w be a word that is not in S^*. If the length of w is not in $\langle m, n \rangle$, then w is in $\bigcup_{j \notin \langle m,n \rangle} \Sigma^j$. Otherwise, by Lemma 2.5.2, there are words $u_1, u_2, \ldots, u_{t-1}$ of length m, and $\tau_1, \tau_2, \ldots, \tau_t$ of lengths in $\{ n-m, n, n+m, n+2m, \ldots \}$ such that

$$w = \tau_1 u_1 \tau_2 u_2 \cdots \tau_t, \tag{2.68}$$

where none of the τ's is in S^* and $t \leq m-1$. From the proof of Theorem 2.5.3, we know that all of the sets $\Sigma^{n-m} \setminus S^*, \Sigma^n \setminus S^*, \ldots, \Sigma^{l-m} \setminus S^*$ are non-empty, and all of the sets $\Sigma^l \setminus S^*, \Sigma^{l+m} \setminus S^*, \ldots$ are empty. So all the τ's are in T and thus w is in $\bigcup_{i=0}^{m-2} (T\Sigma^m)^i T$.

Therefore, the complement of S^* is $\left(\bigcup_{j \notin \langle m,n \rangle} \Sigma^j \right) \bigcup \left(\bigcup_{i=0}^{m-2} (T\Sigma^m)^i T \right)$. □

Example 2.5.9. For the basis $S_4 = \Sigma^3 \cup \Sigma^5 \setminus \{ \texttt{00001} \}$, the unique l is 8 and $T = \{ \texttt{00}, \texttt{01}, \texttt{10}, \texttt{11}, \texttt{00001} \}$. Then all words not in S_4^* are

$$\Sigma \cup \Sigma^2 \cup \Sigma^4 \cup \Sigma^7 \cup T \cup T\Sigma^3 T. \tag{2.69}$$

Let S be a set of words of two lengths m, n with $1 < m < n$ such that S^* is co-finite. Then by Eq. (2.7) on page 33, the set of lengths of words in S^* is

$$\langle m, n \rangle. \tag{2.70}$$

Furthermore, let $l = n + jm$ be the unique integer satisfying the condition in Theorem 2.5.3 for S. Then l can be exponential in n since

$$0 \leq j \leq |\Sigma|^{n-m} - 1. \tag{2.71}$$

Define $T_i = \Sigma^{n+im} \setminus S^*$ for $-1 \le i \le j-1$, and $T = T_{-1} \cup T_0 \cup T_1 \cup \cdots \cup T_{j-1}$. Then none of the T's is empty. By calculating the lengths of words in (2.67), we see that the set of lengths of words not in S^* is

$$\big(\mathbb{N} \setminus \langle m, n\rangle\big) \ \cup \ \big\{\, pn + qm \in \mathbb{N} : 1 \le p \le m-1, -1 \le q \le pj - 1 \,\big\}, \tag{2.72}$$

which satisfies Eq. (2.8) as a superset of $\mathbb{N} \setminus \langle m, n\rangle$. The two terms of the expression (2.72) may not necessarily be disjoint, since the two subsets in the language given in (2.67) can share common words. There is a simple formula for the set of lengths of words not in S^*, which is obtained by noticing m, l are the y's in Proposition 2.1.15 on page 34. In other words, the integers m, l satisfy Eqs. (2.9) and (2.10) on page 34.

Corollary 2.5.10. *Let S be a set of words of lengths m and n over the alphabet Σ, where $1 < m < n$, and S^* is co-finite, where l is the integer such that $l \equiv n \pmod m$, $\Sigma^{l-m} \setminus S^* \neq \emptyset$, and $\Sigma^l \setminus S^* = \emptyset$. Then the set of lengths of words not in S^* is*

$$\mathbb{N} \setminus \langle m, l\rangle. \tag{2.73}$$

Proof. Since $\langle m, l\rangle \subseteq \langle m, n\rangle \subseteq \mathbb{N}$, we have

$$\mathbb{N} \setminus \langle m, l\rangle = \big(\mathbb{N} \setminus \langle m, n\rangle\big) \ \cup \ \big(\langle m, n\rangle \setminus \langle m, l\rangle\big), \tag{2.74}$$

Let $l = n + jm$. Then $j \ge 0$. Any integer s can be uniquely written in the form

$$s = pn + qm, \tag{2.75}$$

where $0 \le p \le m-1$. Then $s \in \langle m, n\rangle$ if and only if $q \ge 0$. We can also write $s = p(n+jm) + (q-pj)m$, and by the same reason, $s \in \langle m, l\rangle$ if and only if $q \ge pj$. Hence it follows that

$$\langle m, n\rangle \setminus \langle m, l\rangle = \big\{\, pn + qm \in \mathbb{N} : 0 \le p \le m-1, 0 \le q \le pj - 1 \,\big\} \tag{2.76}$$

$$= \big\{\, pn + qm \in \mathbb{N} : 1 \le p \le m-1, 0 \le q \le pj - 1 \,\big\}, \tag{2.77}$$

and $\big\{\, pn + qm \in \mathbb{N} : 1 \le p \le m-1, q = -1 \,\big\} \subseteq \mathbb{N} \setminus \langle m, n\rangle$. So

$$\mathbb{N} \setminus \langle m, l\rangle = \big(\mathbb{N} \setminus \langle m, n\rangle\big) \cup \big\{\, pn + qm \in \mathbb{N} : 1 \le p \le m-1, -1 \le q \le pj - 1 \,\big\}, \tag{2.78}$$

which is exactly the set of lengths in the language (2.67). By Corollary 2.5.8, the language (2.67) is $\overline{S^*}$. □

Example 2.5.11. For the basis $S_4 = \Sigma^3 \ \cup \ \Sigma^5 \setminus \{\, \texttt{00001} \,\}$, the unique l is 8. Then the lengths of all words not in S_4^* are in $\mathbb{N} \setminus \langle 3, 8\rangle = \{\, 1, 2, 4, 5, 7, 10, 13 \,\}$.

Each term in the right-hand part in the language given in (2.67) is distinct, since their words have different lengths. Then the number $\mathcal{M}$ of words not in S^* is bounded by

$$\mathcal{M} \ge \sum_{i=0}^{m-2} (|\,T\,| \cdot |\Sigma|^m)^i \,|\,T\,| = \frac{|\,T\,|^m \,|\Sigma|^{m(m-1)} - |\,T\,|}{|\,T\,| \cdot |\Sigma|^m - 1} = \Theta\big(\,|\,T\,|^{m-1} \,|\Sigma|^{m(m-2)}\,\big). \tag{2.79}$$

By Corollary 2.5.5, the number of the longest words not in S^* is

$$\mathcal{I} = (|\,T\,| \cdot |\Sigma|^m)^{m-2}\,|\,T\,| = |\,T\,|^{m-1}\,|\Sigma|^{m(m-2)}\,, \tag{2.80}$$

and the total number of symbols of the longest words not in S^* is

$$\mathcal{IL} = g(m, n+km)\,|\,T\,|^{m-1}\,|\Sigma|^{m(m-2)}\,. \tag{2.81}$$

2.6 The de Bruijn graph in the 2FPFM

In this section, I will give a set $\varkappa(m,n)$ of integers for each pair of positive integers m and n with $\gcd(m,n) = 1$ as a spectrum for the 2FPFM. For any set S consisting of words of lengths m and n, if S^* is co-finite, then $\mathrm{llw}\left(\overline{S^*}\right) \in \varkappa(m,n)$. In order to describe and prove the result about the spectrum $\varkappa(m,n)$, we need a graphical concept and some properties of a particular type of directed graph, which can be viewed as a generalization of the de Bruijn graph. All graphs in this section are directed graphs.

2.6.1 Word graphs of the 2FPFM

Definitions in graph theory

First of all, we will review some basic concepts in graph theory. A *directed graph* G, or *digraph* for short, is a triple (V, A, ψ) consisting of a nonempty set V of *vertices*, a set A of *arcs* (also called *edges* in some literatures), and an *incidence function*

$$\psi : A \to V \times V. \tag{2.82}$$

When $\psi(a) = (u, v)$, arc a is said to join u to v, vertex u is the *tail*, and vertex v is the *head*. A directed graph is *strict* if it has no loop and it has at most one arc from one vertex to another. In the literature, "directed graph" sometimes refers to what

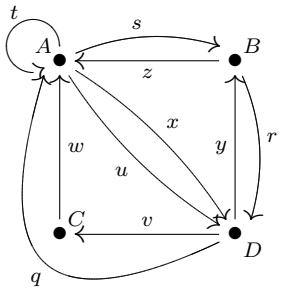


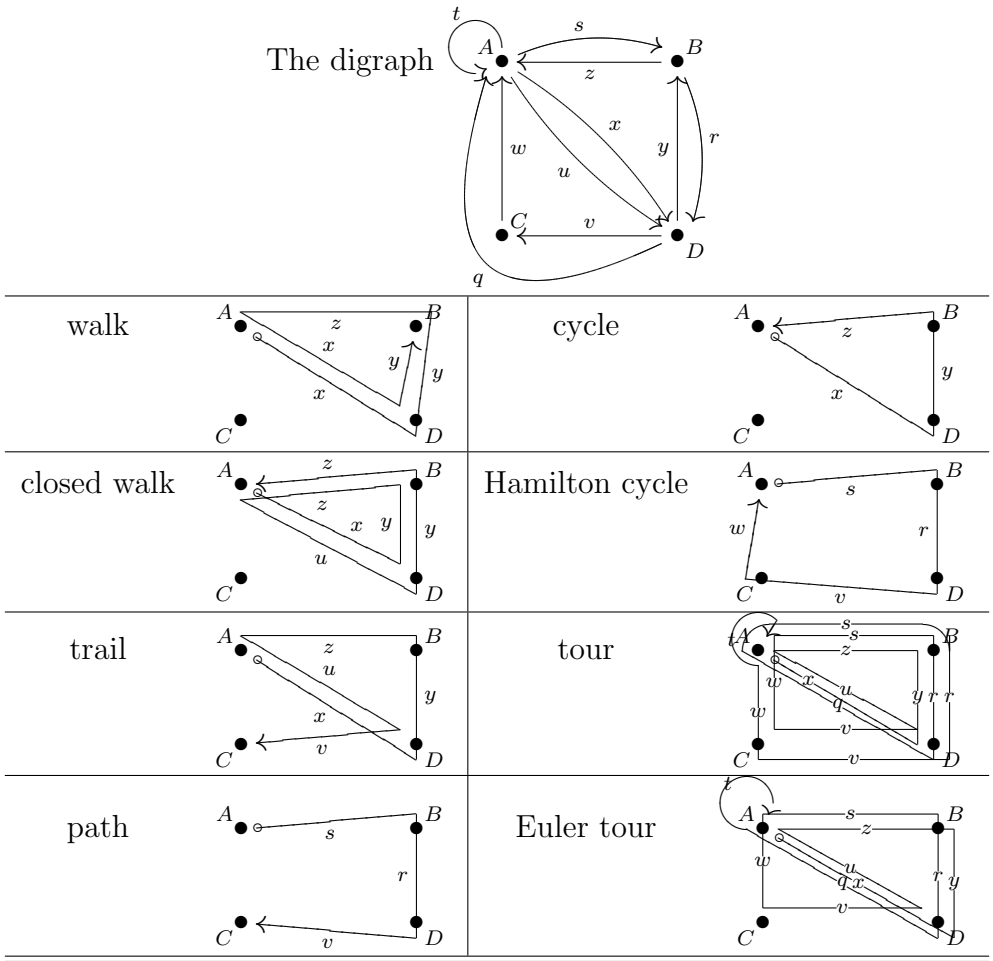
Figure 2.4: An example of a directed graph

we have called the strict directed graph here. Figure 2.4 is an example of a directed graph, where $\{ A, B, C, D \}$ is the set of vertices and $\{ q, r, s, t, u, v, w, x, y, z \}$ is the set of arcs. The incidence function can be easily understood by the arrows in the diagram. For example, $\psi(s) = (A, B)$.

Let $G_1 = (V_1, A_1, \psi_1)$ and $G_2 = (V_2, A_2, \psi_2)$ be two digraphs. Digraph G_1 is a *subgraph* of digraph G_2 if $V_1 \subseteq V_2$, $A_1 \subseteq A_2$ and ψ_1 is the restriction of ψ_2 to A_1. Furthermore, when $V_1 = V_2$, the subgraph G_1 is called a *spanning subgraph* of G_2. Digraph G_1 is *isomorphic* to digraph G_2 if there are bijections $\zeta : V_1 \to V_2$ and $\xi : A_1 \to A_2$ such that $\psi_2\big(\xi(a)\big) = \big(\zeta(v), \zeta(v')\big)$ for all $a \in A_1$, where $\psi_1(a) = (v, v')$.

A *walk* in G is a finite nonempty sequence $W = v_0, a_1, v_1, a_2, v_2, \cdots, a_k, v_k$, or simply $v_0, v_1, v_2, \cdots, v_k$, or $a_1, a_2, \cdots, a_k$, where v's are vertices and a's are arcs such that each a_i joins v_{i-1} to v_i, and k is the length of W. Without confusion, commas between terms are omitted in order to save space. In Figure 2.4, $AxDyBzAxDyB$ is a walk. In particular, any sequence consisting of a single vertex is treated as a walk of length 0. A walk is called a *trail* if each arc is distinct (such as $AxDyBzAuDvC$), and is called a *path* if each vertex is distinct (such as $AsBrDvC$), and called

Table 2.7: Different types of walks in a digraph



closed if the origin and terminus are the same (such as $AxDyBzAuDyBzA$). A closed trail is called a *cycle* if the origin and internal vertices are distinct (such as $AxDyBzA$). A *Hamilton cycle* is a cycle which contains every vertex (such as $AsBrDvCwA$). A *tour* is a closed walk which traverses each arc at least once (such as $AxDyBzAuDvCwAsBrDqAsBrDvCwAtA$). In the definitions given here, a tour is not a cycle in general, since a cycle cannot contain a vertex twice except at the ends, but a tour can. An *Euler tour* is a tour which traverses each arc exactly once (such as $AxDyBzAuDvCwAsBrDqAtA$).

Two vertices u and v are said to be *connected* if each can be reached from the other by a directed path. A graph is *connected* if every pair of vertices is connected. The *indegree* $d^-(v)$ of a vertex v is the number of arcs with head v; and the *outdegree* $d^+(v)$ of a vertex v is the number of arcs with tail v.

All above definitions in graph theory are taken from the textbook of Bondy and Murty [15], and the reader may refer to it for more details.

There is a classic theorem on Euler tours as follows.

Theorem 2.6.1. *A nonempty connected digraph G contains an Euler tour if and only if $d^+(v) = d^-(v)$ for each vertex v.*

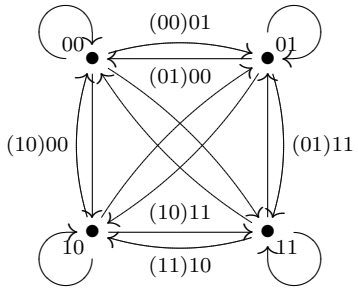
The definition of the word graph

In order to study the 2FPFM, I will introduce a concept of a word graph for a set of words of two distinct lengths. Let $S = \{ x_1, x_2, \ldots, x_k \}$ be a set of words of lengths m and n over an alphabet Σ, where $0 < m < n$ (m, n not necessarily co-prime). The *word graph* $G_S^{(m,n)}$ for S is a directed graph $(\Sigma^{n-m}, \Sigma^n \setminus S, \psi)$, where ψ is defined on each word $w \in \Sigma^n \setminus S$ by $\psi(w) = (u, v)$, where $u = w[1..n-m]$ and $v = w[m+1..n]$. The superscript (m, n) is omitted if it is clear from the context. In addition, a *labeling function* $\varphi : \Sigma^n \setminus S \to \Sigma^m$ on the arcs of a word graph is defined by $\varphi(w) = w[n-m+1..n]$. In other words, if $\psi(w) = (u, v)$ and $\varphi(w) = w'$, then $w = uw' = w''v$, where $w'' = w[1..m]$. In the graph diagram of a word graph, we label each arc w by $(u)w'$ instead of the arc w for explicitness, where $w' = \varphi(w), \psi(w) = (u, v), w = uw'$.

Example 2.6.2. The word graph for the basis $S_1 = \Sigma^3 \cup \Sigma^5$ in Example 2.1.2 on page 30 is a digraph with no arcs:

Example 2.6.3. The word graph for the basis $S_4 = \Sigma^3 \cup \Sigma^5 \setminus \{ \texttt{00001} \}$ in Example 2.4.5 on page 42 is the digraph depicted below:

Example 2.6.4. The word graph for Σ^2 is the digraph depicted below, where Σ^2 is viewed as a set of words of lengths 2 and 4: (some labels are omitted for clarity)



2.6.2 Another equivalent condition on co-finiteness

The following lemma provides a general connection between a set S of words of two lengths and its word graph. Any word not in S^* that is of a particular length is associated with a path in the word graph $G_S^{(m,n)}$, even when S^* is not co-finite. Hence the word graph $G_S^{(m,n)}$ for a given set of words $S \subseteq \Sigma^m \cup \Sigma^n$ is a useful characteristic of S when considering the words not in S^*.

Lemma 2.6.5. *Suppose* $\Sigma^m \subseteq S \subseteq \Sigma^m \cup \Sigma^n$, *where* $1 < m < n$ *and* $\gcd(m,n) = 1$. *Let* $j \geq -1$ *be an integer, and* $l = n + jm$. *Then there is a word* x *of length* l *that is not in* S^* *if and only if there is a walk of length* $j+1$ *in the word graph* $G_S^{(m,n)}$.

Proof. Let $j \geq -1$ be an integer, $l = n + jm$, and $x = a_1 a_2 \cdots a_l$ be a word of length l that is not in S^*. Define

$$y_i = a_{im+1} a_{im+2} \cdots a_{im+n}, \qquad \text{for } 0 \leq i \leq j; \tag{2.83}$$

$$z_i = a_{im+1} a_{im+2} \cdots a_{im+n-m}, \qquad \text{for } 0 \leq i \leq j+1. \tag{2.84}$$

Then, by Proposition 2.4.10 on page 46, none of the y's is in S^*. By comparing lengths, we can write

$$y_i = z_i \; a_{im+n-m+1} \cdots a_{im+n} = a_{im+1} \cdots a_{im+m} \; z_{i+1} \tag{2.85}$$

for each y_i, and thus y_i is an arc in the word graph $G_S^{(m,n)}$. Hence there is a walk $z_0 y_0 z_1 y_1 \cdots y_j z_{j+1}$ of length $j+1$ in the word graph $G_S^{(m,n)}$.

Let $x_0 w_1 x_1 w_2 \cdots w_{j+1} x_{j+1}$ be a walk of length $j+1$ in the word graph $G_S^{(m,n)}$. Then, by definition, none of the words w_i is in S^* for $1 \leq i \leq j+1$, and $w_i = x_{i-1} u_i = v_i x_i$ for u_i and v_i of length m, where $u_i = \varphi(w_i)$ is the labeling function. Consider the word

$$x = x_0 u_1 u_2 \cdots u_{j+1} = v_1 v_2 \cdots v_{j+1} x_{j+1}. \tag{2.86}$$

By comparing lengths, it can be checked that for this word x, the factors y_i in Proposition 2.4.10 are exactly the w_i defined here. Since none of the w_i is in S^*, x is not in S^*. □

Now I will give another equivalent condition describing when S generates a co-finite language. Furthermore, the instance of the 2FPFM specified by S can be solved accordingly. If $m = 1$, by Proposition 2.4.4, S^* can only be trivially co-finite, which means $S^* = \Sigma^*$. Assume $m > 1$. By Theorem 2.5.3 and Corollaries 2.5.5 and 2.5.8, the solution to an instance of the 2FPFM depends on the omitted words of particular lengths, which are exactly those lengths specified in Lemma 2.6.5. Therefore, based on the equivalent problem in combinatorics on words and the connection between the word graph and words of particular lengths, another equivalent problem to the 2FPFM in graph theory is as follows.

Theorem 2.6.6. *Let S be a set of words of lengths m and n over the alphabet Σ, where $1 < m < n$ and $\gcd(m, n) = 1$. Then S^* is co-finite if and only if S contains all words of length m and there is no cycle in the word graph $G_S^{(m,n)}$. Furthermore, when S^* is co-finite, then $\operatorname{llw}\left(\overline{S^*}\right) = g(m, l)$, where $l = n + jm$ and j is the length of the longest path in $G_S^{(m,n)}$.*

Proof. Suppose S^* is co-finite. By the First Lemma of the 2FPFM, all words of length m are in S. If there is a cycle in $G_S^{(m,n)}$, then there are arbitrarily long walks in $G_S^{(m,n)}$, and by Lemma 2.6.5, there are arbitrarily long words that are not in S^*, which contradicts the co-finiteness of S^*. Hence there is no cycle in $G_S^{(m,n)}$.

If there is no cycle in $G_S^{(m,n)}$, let j be the length of the longest path and let $l = n + jm$. Then by Lemma 2.6.5, $\Sigma^{l-m} \setminus S^* \neq \emptyset$, and $\Sigma^l \setminus S^* = \emptyset$. In addition, all words of length m are in S, so by Theorem 2.5.3, S^* is co-finite. Furthermore, also by Theorem 2.5.3, the length of the longest words not in S^* is $g(m, l)$. □

Now we extend the definition of the *labeling function* $\varphi(a)$ from arcs to any walk, where an arc is viewed as a walk of length 1. Let $x_0w_1x_1w_2 \cdots w_jx_j$ be a walk in the word graph $G_S^{(m,n)}$. Then, by definition, each w_i can be written as $w_i = x_{i-1}u_i = v_ix_i$ for some u_i and v_i of length m, where $u_i = \varphi(w_i)$. Define

$$\varphi(x_0w_1x_1w_2 \cdots w_jx_j) = \varphi(w_1)\varphi(w_2) \cdots \varphi(w_j) = u_1u_2 \cdots u_j. \tag{2.87}$$

By the proof of Lemma 2.6.5, for any walk, the word $x_0\varphi(x_0w_1x_1w_2 \cdots w_jx_j)$ is not in S^*.

Corollary 2.6.7. *Let S be a set of words of lengths m and n over the alphabet Σ, where $1 < m < n$, and S^* is co-finite, where l is the length of the longest path in the word graph $G_S^{(m,n)}$. Then the set of the longest words not in S^* is*

$$(T\Sigma^m)^{m-2}T, \tag{2.88}$$

where $T = \left\{ x_0\varphi(x_0 \cdots x_l) : x_0 \cdots x_l \text{ is a path in } G_S^{(m,n)} \text{ of length } l \right\}$.

Proof. Follows directly from Lemma 2.6.5 and Corollary 2.5.5. □

Corollary 2.6.7 shows that the 2FPFM is equivalent to the following problem.

Problem 2.6.8 (Equivalent statement of the 2FPFM, the 2nd)**.** *Let Σ be a (finite) alphabet and let S be a set of non-empty words of lengths m and n, where m, n are two positive integers and $1 < m < n$, such that there is no cycle in the word graph $G_S^{(m,n)}$. Find the longest paths in the word graph $G_S^{(m,n)}$.*

Corollary 2.6.9. *Let S be a set of words of lengths m and n over the alphabet Σ, where $1 < m < n$, and S^* is co-finite. Then the set of words not in S^* is*

$$\left(\bigcup_{j \notin \langle m,n \rangle} \Sigma^j \right) \bigcup \left(\bigcup_{i=0}^{m-2} (T\Sigma^m)^i T \right), \tag{2.89}$$

where $T = \left\{ x_0\varphi(x_0 \cdots x_l) : x_0 \cdots x_l \text{ is a path in } G_S^{(m,n)} \text{ of length } \geq 0 \right\}$.

Proof. Follows directly from Lemma 2.6.5 and Corollary 2.5.8. □

Corollary 2.6.10. *Let S be a set of words of lengths m and n over the alphabet Σ, where $1 < m < n$, and S^* is co-finite, where l is the length of the longest path in the word graph $G_S^{(m,n)}$. Then the set of lengths of words not in S^* is*

$$\mathbb{N} \setminus \langle m, n + ml \rangle. \tag{2.90}$$

Proof. Follows directly from Lemma 2.6.5 and Corollary 2.5.10. □

Example 2.6.11. The word graph for the basis $S_4 = \Sigma^3 \cup \Sigma^5 \setminus \{ \texttt{00001} \}$ in Example 2.4.5 on page 42 is

which contains only paths $(\texttt{00}), (\texttt{01}), (\texttt{10}), (\texttt{11})$ of length 0 and $(\texttt{00}, \texttt{00001}, \texttt{01})$ of length 1, and has no cycle. So S_4^* is co-finite. Let $T = \{ \texttt{00}, \texttt{01}, \texttt{10}, \texttt{11}, \texttt{00001} \}$. Then the longest words not in S_4^* are $\texttt{00001}\Sigma^3\texttt{00001}$ and all words not in S_4^* are

$$\Sigma \cup \Sigma^2 \cup \Sigma^4 \cup \Sigma^7 \cup T \cup T\Sigma^3 T. \tag{2.91}$$

2.6.3 The de Bruijn graph and a generalization

The de Bruijn word and the de Bruijn graph

We call a word w a k-ary *de Bruijn word* of order n if each k-ary word of length n appears in w as a factor exactly once. For example, two of the binary de Bruijn words of order 3 are 0001011100 and 0001110100. It is easy to see that a k-ary de Bruijn word of order n consists of $k^n + n - 1$ letters. Sometimes, a de Bruijn word is represented in a circular form by clockwise arranging the first k^n letters in

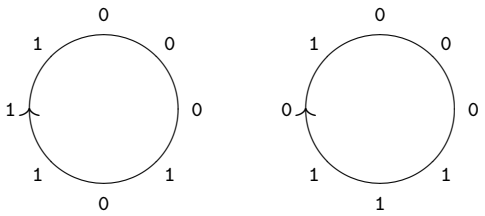


Figure 2.5: Binary de Bruijn words of order 3

a circle. Figure 2.5 shows the two circular forms of the binary de Bruijn words of order 3. Here we will use the linear form of the de Bruijn word.

In 1946, de Bruijn [22] showed that the number of such binary words of order n is $2^{2^{n-1}-n}$ by introducing a digraph, which was independently produced by Good [57] in the same year. In general, the number of distinct k-ary de Bruijn words of length n is $(k!)^{k^{n-1}}k^{-n}$ [1, 36]. The digraph de Bruijn introduced is usually called the *de Bruijn graph* or *de Bruijn-Good graph*. The de Bruijn graph and de Bruijn words are strongly related, since each Euler tour in a de Bruijn graph produces a de Bruijn word. The binary de Bruijn graph of order 2 is depicted in Figure 2.6.

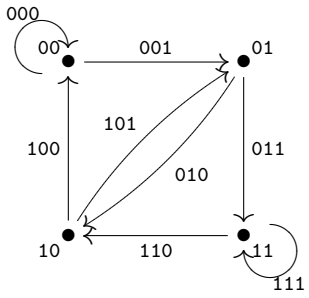


Figure 2.6: Binary de Bruijn graph of order 2

More formally, a k-ary *de Bruijn graph* of order n is a digraph $(\Sigma^n, \Sigma^{n+1}, \psi)$, where $\Sigma = \{\, 0, 1, \ldots, k-1 \,\}$ and $\psi(ua) = (u, v)$ for $v = u[2..n]a$, $u, v \in \Sigma^n$, $a \in \Sigma$. In a de Bruijn graph, the indegree and outdegree of each vertex is exactly $|\Sigma|$.

The study of the de Bruijn word dates back at least to the late 1800's. In 1894, de Rivière [134] raised the question of the number of de Bruijn words, which was then answered by Flye Sainte-Marie [47], in the same journal *L'Intermédiaire des Mathématiciens*. The concept of de Bruijn words is also connected to some other concepts, such as Lyndon words and necklaces [144] and feedback shift registers [56]. All of the results can be used to efficiently generate a de Bruijn word; for example, see Fredricksen [48] or Ralston [128] for a good survey on the topic of generating de Bruijn words.

The generalized de Bruijn graph

Let m, n be two distinct positive integers, where $m < n$, and Σ be an alphabet. The set of words Σ^m can be viewed as of lengths m and n, although it contains no word of length n. The special word graph $G_{\Sigma^m}^{(m,n)}$ is denoted by

$$\Gamma(m, n) = G_{\Sigma^m}^{(m,n)}. \tag{2.92}$$

The digraph $\Gamma(m, n)$ is a generalization of the de Bruijn graph in the sense of the following theorem.

Theorem 2.6.12. *The digraph $\Gamma(1, n)$ defined over an alphabet of size k is isomorphic to the k-ary de Bruijn graph of order $n - 1$.*

Proof. Let Σ be the alphabet of $\Gamma(1, n)$ and let $\Delta = \{\, 0, 1, \ldots, k-1 \,\}$. Then, by definition, $\Gamma(1, n) = (\Sigma^{n-1}, \Sigma^n, \psi)$ and the k-ary de-Bruijn graph of order $n - 1$ is $(\Delta^{n-1}, \Delta^n, \psi')$. Since $|\Sigma| = |\,\Delta\,| = k$, let $\eta : \Sigma^* \to \Delta^*$ be a bijection. Let ζ and ξ be the restrictions of η on Σ^{n-1} and Σ^n respectively, and let $w \in \Sigma^n$ be an arbitrary arc of $\Gamma(1, n)$. Then by the definition of the word graph, $\psi(w) = (u, v)$, where $u = w[1..n-1]$ and $v = w[2..n]$. By the definition of the de Bruijn graph, $\psi'\big(\xi(w)\big) = \psi'(w') = (u', v')$, where $w' = u'a$ and $v' = u'[2..n-1]a$ for some $a \in \Delta$. By comparing lengths, we have $u' = w'[1..n-1]$ and $v' = w'[2..n]$. So $\psi'\big(\xi(w)\big) = \big(\zeta(u), \zeta(v)\big)$, and thus $(\Sigma^{n-1}, \Sigma^n, \psi)$ is isomorphic to $(\Delta^{n-1}, \Delta^n, \psi')$. □

We call $\Gamma(m, n)$ a *generalized de Bruijn graph*. By the definition of the word graph, $G_{\Sigma^m}^{(m,n)} = G_{\emptyset}^{(m,n)} = G_T^{(m,n)}$ for any $T \subseteq \Sigma^m$. When S is co-finite, S must contain all words of length m, so in the general discussion on the 2FPFM by the graph approach, we are only interested in those S (and $G_S^{(m,n)}$) for which $\Sigma^m \subseteq S$. We always assume $\Sigma^m \subseteq S$ without further explanation in the remaining of this section.

Table 2.8: Comparison of the de Bruijn graph and the word graph $\Gamma(m, n)$

	word graph $\Gamma(m, n) = G_{\Sigma^m}^{(m,n)}$	de Bruijn graph of order k
vertices	Σ^{n-m}	Σ^k
arcs	Σ^n	Σ^{k+1}
$\psi(w) = (u, v)$	$w \in u\Sigma^m \ \cap \ \Sigma^m v$	$w \in u\Sigma \ \cap \ \Sigma v$

Proposition 2.6.13. *Let m, n be two integers, $0 < m < n$, and let G be a digraph. The following two conditions are equivalent.*

(a) There exists a set S of words of lengths m and n such that S contains all words of length m and its word graph $G_S^{(m,n)} = G$;

(b) The digraph G is a spanning subgraph of $\Gamma(m, n)$.

Proof. From the definitions, the result follows straightforwardly. □

By Theorem 2.6.6 and corresponding Corollaries 2.6.7 and 2.6.9, we know that each acyclic spanning subgraph of $\Gamma(m,n)$ is a word graph that corresponds to a basis S of words of lengths m,n such that S^* is co-finite, and vice versa. The words not in S^* can be expressed by a regular expression in terms of the labeling of paths in the corresponding word graph, and the length of the longest omitted word can be expressed exactly in terms of the length of the longest paths in the corresponding word graph. Based on the study of the generalized de Bruijn graph, I will show below some general properties of the 2FPFM. For example, the maximum acyclic subgraph of $\Gamma(m,n)$ can be used to construct an S with the minimum $\kappa = |\,S\,|$ such that S^* is co-finite; and the longest (simple) path in $\Gamma(m,n)$ can be used to construct an S with the maximum $\mathcal{L} = \mathrm{llw}\left(\overline{\mathcal{S}^*}\right)$ among all instance of the 2FPFM with respect to the integers m,n.

Now we consider the general properties of the generalized de Bruijn graph $\Gamma(m,n)$. First we need a concept from graph theory. The *arc graph* of a digraph $G = (V,A,\psi)$ is a digraph $\tilde{G} = (A,B,\varphi)$ that has the arcs of G as vertices, and the set B of arcs contains a, where $\varphi(a) = (u,v)$, if there are $x,y,z \in V$ such that $\psi(u) = (x,y)$ and $\psi(v) = (y,z)$. Then the arc graph $\tilde{G}$ has a Hamilton cycle if and only if G has an Euler tour as in the following proposition.

Proposition 2.6.14. *Let $\tilde{G} = (A,B,\varphi)$ be the arc graph of $G = (V,A,\psi)$. Then $\tilde{G}$ has a Hamilton cycle if and only if G has an Euler tour.*

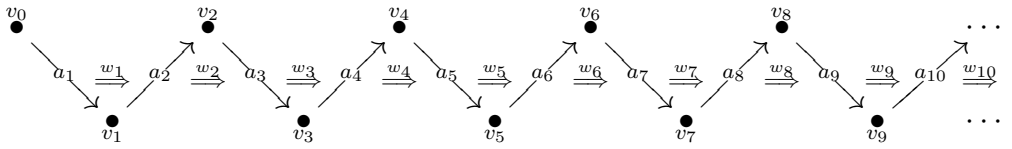


Figure 2.7: $\tilde{G}$ has a Hamilton cycle if and only if G has an Euler tour

Proof. $\Leftarrow$: Suppose G has an Euler tour given by

$$W = v_0 a_1 v_1 a_2 v_2 \cdots a_{k-1} v_{k-1} a_k v_0 \tag{2.93}$$

that covers each arc in A exactly once and thus each a_i is distinct for $1 \le i \le k$. By definition, between each pair a_i, a_{i+1} of arcs in A (and a_k, a_1 at the very end) as vertices in $\tilde{G}$, there is an arc w_i such that $\varphi(w_i) = (a_i, a_{i+1})$. Then there is a closed walk in $\tilde{G}$

$$\tilde{W} = a_1 w_1 a_2 w_2 \cdots w_{k-1} a_k w_k a_1, \tag{2.94}$$

where each vertex encountered is distinct, except that the origin is the same as the terminus. Then $\tilde{W}$ is a cycle. Since W is an Euler tour, $\tilde{W}$ encounters each vertex exactly once, and thus $\tilde{W}$ is a Hamilton cycle.

$\Rightarrow$: Suppose $\tilde{G}$ has a Hamilton cycle given by

$$\tilde{W} = a_1 w_1 a_2 w_2 \cdots w_{k-1} a_k w_k a_1. \tag{2.95}$$

By the definitions, each a_i and w_i is distinct for $1 \leq i \leq k$ and $\tilde{W}$ covers each vertex in A exactly once. For each w_i, there are $a_i, b_i, a_{i+1}, b_{i+1} \in V$ such that $\psi(a_i) = (v_i, u_i)$, $\psi(a_{i+1}) = (v_{i+1}, u_{i+1})$, and $u_i = v_{i+1}$. In addition, by $\varphi(w_k) = (a_k, a_1)$, we have $u_{k+1} = v_1$. Consider the walk in G given by

$$W = v_1 a_1 v_2 a_2 \cdots v_k a_k v_1. \tag{2.96}$$

Then W is a closed walk. Since $\tilde{W}$ is a Hamilton cycle, W encounters each arc exactly once, and thus W is an Euler tour. □

The following result is generalized from a property of the de Bruijn graph. Zhang and Lin [175] showed in 1987 that the de Bruijn graph of order $k+1$ is the arc graph of the de Bruijn graph of order k. Lemma 2.6.15 is a generalization of this result, namely, that the generalized de Bruijn graph $\Gamma(m, n+m)$ is the arc graph of the generalized de Bruijn graph $\Gamma(m, n)$.

Lemma 2.6.15. *$\Gamma(m, n+m)$ is the arc graph of $\Gamma(m, n)$.*

Proof. Let Σ be the alphabet. By definition, we have

$$\Gamma(m, n) = (\Sigma^{n-m}, \Sigma^n, \psi), \quad \Gamma(m, n+m) = (\Sigma^n, \Sigma^{n+m}, \varphi). \tag{2.97}$$

The vertices of $\Gamma(m, n+m)$ are exactly those arcs of $\Gamma(m, n)$. Now, we consider the arcs of $\Gamma(m, n+m)$. Let a, b be two arcs in $\Gamma(m, n)$ such that $\psi(a) = (x, y)$, $\psi(b) = (y, z)$, $a = xu$, $b = yv$, where u, v are of length m. Then the word

$$xuv \in \Sigma^{n+m}, \tag{2.98}$$

as an arc in $\Gamma(m, n+m)$, is the one that corresponds to a, b in the definition of the arc graph of $\Gamma(m, n)$. On the other hand, each word $w \in \Sigma^{n+m}$ as an arc in $\Gamma(m, n+m)$ joins $w[1..n]$ to $w[m..n+m]$, where the two vertices share a common part $w[m..n]$. In other words, as two arcs in G,

$$\psi(w[1..n]) = (w[1..n-m], w[m..n]), \tag{2.99}$$

$$\psi(w[m..n+m]) = (w[m..n], w[2m..n+m]), \tag{2.100}$$

they share a common vertex that satisfies the condition in the definition of the arc graph of $\Gamma(m, n)$. Therefore, $\Gamma(m, n+m)$ is the arc graph of $\Gamma(m, n)$. □

2.6.4 Spectrum theorem for the 2FPFM

Let Σ be the alphabet and m, n be two integers that $0 < m < n$. We know that each word graph for a set of words of lengths m, n is a spanning subgraph of the generalized de Bruijn graph $\Gamma(m, n)$, and from any spanning subgraph G of $\Gamma(m, n)$, we can construct a set S of words of lengths m, n such that $G_S^{(m,n)} = G$. Furthermore, the language S^* generated by S is co-finite if and only if the word graph $G_S^{(m,n)}$ contains no cycle. Then by the equivalence of the 2FPFM and the word graph in Theorem 2.6.6, the bound in Theorem 2.4.12 is achievable if and only if there is a path of length $|\Sigma|^{n-m} - 1$ in an acyclic spanning subgraph of $\Gamma(m, n)$, that is to say there is a Hamilton path in $\Gamma(m, n)$, since there are $|\Sigma|^{n-m}$ vertices in $\Gamma(m, n)$.

Euler tours and Hamilton cycles in $\Gamma(m, n)$

Each ordinary de Bruijn graph has Euler tours and Hamilton cycles [22, 57, 175]. I will show that this result can also be generalized for the word graph $\Gamma(m, n)$.

Theorem 2.6.16. *For any integers m and n, where $0 < m < n$, there is an Euler tour in the generalized de Bruijn graph $\Gamma(m, n)$.*

Proof. In order to prove that there is an Euler tour in $\Gamma(m, n) = (\Sigma^{n-m}, \Sigma^n, \psi)$, by Theorem 2.6.1, we only need to show the following two conditions hold:

1. For any vertex v, the indegree $d^-(v)$ is equal to the outdegree $d^+(v)$;
2. For any ordered pair of vertices u, v, there is a directed path from u to v.

By definition, for each vertex $v \in \Sigma^{n-m}$, the set of arcs with head v is $v\Sigma^m$, and thus $d^+(v) = |\Sigma|^m$. Similarly, the set of arcs with tail v is $\Sigma^m v$, and thus $d^-(v) = |\Sigma|^m = d^+(v)$

For any two vertices $u, v \in \Sigma^{n-m}$, define $\tau = u0^{m\lceil \frac{n}{m} \rceil - n}v$. Then we have $|\,\tau\,| \equiv n \pmod m$ and we can write

$$\begin{aligned}\tau = uw_1w_2w_3\cdots w_{\lceil \frac{n}{m} \rceil - 1} = w_1'u_1w_2w_3\cdots w_{\lceil \frac{n}{m} \rceil - 1} = w_1'w_2'u_2w_3\cdots w_{\lceil \frac{n}{m} \rceil - 1} \\ = \cdots = w_1'w_2'\cdots w_{\lceil \frac{n}{m} \rceil - 2}'u_{\lceil \frac{n}{m} \rceil - 2}w_{\lceil \frac{n}{m} \rceil - 1} = w_1'w_2'\cdots w_{\lceil \frac{n}{m} \rceil - 1}'v, \quad (2.101)\end{aligned}$$

where all the w_i are of length m and all the u_i are of length n. Then there is a directed path from u to v given by

$$u, uw_1, u_1, u_1w_2, u_2, \ldots, u_{\lceil \frac{n}{m} \rceil - 2}, u_{\lceil \frac{n}{m} \rceil - 2}w_{\lceil \frac{n}{m} \rceil - 1}, v. \quad (2.102)$$

Therefore, there is an Euler tour in the generalized de Bruijn graph $\Gamma(m, n)$. □

Lemma 2.6.17. *For any integers m and n, where $0 < m < n \leq 2m$, the generalized de Bruijn graph $\Gamma(m, n)$ has a Hamilton cycle.*

Proof. First we prove that for any pair of vertices u, v, there is an arc that joins u and v in each direction. By definition, $\Gamma(m, n) = (\Sigma^{n-m}, \Sigma^n, \psi)$. For any two vertices $u, v \in \Sigma^{n-m}$, the arc $u0^{2m-n}v \in \Sigma^n$ has u as the head and v as the tail. Now one can find a Hamilton cycle in $\Gamma(m, n)$ as follows. First find an arbitrary permutation of all vertices. Since there is an arc that joins each pair of vertices, the permutation of all vertices can be converted into a closed walk, which is then a Hamilton cycle. □

Theorem 2.6.18. *For any integers m and n, where $0 < m < n$, there is a Hamilton cycle in the generalized de Bruijn graph $\Gamma(m, n)$.*

Proof. If $n \leq 2m$, by Lemma 2.6.17, $\Gamma(m, n)$ has a Hamilton cycle. Now we assume $n > 2m$. Then the notation $\Gamma(m, n - m)$ is meaningful, and by Lemma 2.6.15, $\Gamma(m, n)$ is the arc graph of $\Gamma(m, n-m)$. From Theorem 2.6.16, we know that there is an Euler tour in $\Gamma(m, n - m)$. Then by Proposition 2.6.14, there is a Hamilton cycle in $\Gamma(m, n)$. □

The spectrum theorem

We are now ready to state the spectrum theorem, which gives a set of integers $\varkappa(m, n)$ for the 2FPFM corresponding to each choice of m and n that covers all possible lengths of longest omitted words. Recall that $g(x_1, x_2) = x_1x_2 - x_1 - x_2$ is the Frobenius number of x_1, x_2.

Theorem 2.6.19 (*Spectrum theorem*)**.** *Let m, n be integers such that $1 < m < n$ and $\gcd(m, n) = 1$. For any basis $U \subseteq \Sigma^m \cup \Sigma^n$ such that U^* is co-finite, then $\operatorname{llw}\left(\overline{U^*}\right) \in \varkappa(m, n)$, where*

$$\varkappa(m, n) = \left\{ g(m, l) : l = n, n + m, \ldots, n + im, \ldots, n + (|\Sigma|^{n-m} - 1)m \right\}. \quad (2.103)$$

Furthermore, there is a set S of words of lengths m, n for each number $g(m, l)$ in $\varkappa(m, n)$, such that $\operatorname{llw}\left(\overline{S^*}\right) = g(m, l)$.

Proof. Let S be a set of words of lengths m, n such that S^* is co-finite. By the equivalence between the 2FPFM and combinatorics on words of particular lengths as in Theorem 2.5.3, the longest words not in S^* are of length $g(m, l)$, where l is the unique integer that $l \equiv n \pmod{m}$, $\Sigma^{l-m} \setminus S^* \neq \emptyset$, and $\Sigma^l \setminus S^* = \emptyset$. By the Second Lemma of the 2FPFM, all words of length $m|\Sigma|^{n-m} + n - m$ are in S^*. In addition, none of the words of length $n - m$ is in S^* for $\gcd(m, n) = 1$. So $n \leq l \leq n + (|\Sigma|^{n-m} - 1)m$, and $\operatorname{llw}\left(\overline{S^*}\right) \in \varkappa(m, n)$.

By Theorem 2.6.18, there is a Hamilton cycle in the generalized de Bruijn graph $\Gamma(m, n)$. So there is an acyclic spanning subgraph G_j of $\Gamma(m, n)$, where the longest

path is of length j for each $j = 0, 1, 2, \ldots, |\Sigma|^{n-m} - 1$. That subgraph G_j is just the graph consisting all vertices of $\Gamma(m, n)$ and only the arcs that lie in a single path of length j. Let S_j be the corresponding set of words such that $G_{S_j}^{(m,n)} = G_j$. Then by the equivalence between the 2FPFM and word graph as in Theorem 2.6.6, S_j^* is co-finite and llw $\left(\overline{S_j^*}\right) = g(m, n + jm)$ for $0 \leq j \leq |\Sigma|^{n-m} - 1$. □

As shown in Theorem 2.6.19, there are exactly $|\Sigma|^{n-m}$ integers that can be the lengths of the longest words as solutions to the 2FPFM over the alphabet Σ with lengths of words in the basis being m, n. Furthermore, the $|\Sigma|^{n-m}$ integers in $\varkappa(m, n)$ constitute an arithmetic progression with common difference $m^2 - m$. With fixed m and alphabet size ≥ 2, the largest integer in $\varkappa(m, n)$ is

$$\mathcal{L} = g(m, |\Sigma|^{n-m} + n - m) = \Theta(|\Sigma|^{\nu}), \tag{2.104}$$

where $\nu = \max\{m, n\} = n$.

Example 2.6.20. Consider small integers m, n such that $0 < m < n < 10$ and $\gcd(m, n) = 1$. Then the possible lengths of the longest words not in a co-finite language generated by words of lengths m, n are in Table 2.9.

Table 2.9: Spectrum $\varkappa(m, n)$ of length of longest words not in S^* in the 2FPFM

m, n	$\|\Sigma\| = 1$	$\|\Sigma\| = 2$	$\|\Sigma\| = 3$	$\|\Sigma\| = 4$
$1, *$	$\{-1^a\}$	$\{-1\}$	$\{-1\}$	$\{-1\}$
$2, 3$	$\{1\}$	$\{1, 3\}$	$\{1, 3, 5\}$	$\{1, 3, 5, 7\}$
$2, 5$	$\{3\}$	$\{3, 5, 7, \ldots, 17\}_8{}^b$	$\{3, 5, 7, \ldots, 55\}_{27}$	$\{3, 5, 7, \ldots, 129\}_{64}$
$2, 7$	$\{5\}$	$\{5, 7, 9, \ldots, 67\}_{32}$	$\{5, 7, 9, \ldots, 489\}_{243}$	$\{5, 7, 9, 2051\}_{1024}$
$2, 9$	$\{7\}$	$\{7, 9, 11, \ldots, 261\}_{128}$	$\{7, 9, 11, \ldots, 4379\}_{2187}$	$\{7, 9, 11, \ldots, 32773\}_{16384}$
$3, 4$	$\{5\}$	$\{5, 11\}$	$\{5, 11, 17\}$	$\{5, 11, 17, 23\}$
$3, 5$	$\{7\}$	$\{7, 13, 19, 25\}$	$\{7, 13, 19, \ldots, 55\}_9$	$\{7, 13, 19, \ldots, 97\}_{16}$
$3, 7$	$\{11\}$	$\{11, 17, 23, \ldots, 101\}_{16}$	$\{11, 17, 23, \ldots, 491\}_{81}$	$\{11, 17, 23, \ldots, 1541\}_{256}$
$3, 8$	$\{13\}$	$\{13, 19, 25, \ldots, 199\}_{32}$	$\{13, 19, 25, \ldots, 1465\}_{243}$	$\{13, 19, 25, \ldots, 6151\}_{1024}$
$4, 5$	$\{11\}$	$\{11, 23\}$	$\{11, 23, 35\}$	$\{11, 23, 35, 47\}$
$4, 7$	$\{17\}$	$\{17, 29, 41, \ldots, 101\}_8$	$\{17, 29, 41, \ldots, 329\}_{27}$	$\{17, 29, 41, \ldots, 773\}_{64}$
$4, 9$	$\{23\}$	$\{23, 35, 47, \ldots, 395\}_{32}$	$\{23, 35, 47, \ldots, 2927\}_{243}$	$\{23, 35, 47, \ldots, 12299\}_{1024}$
$5, 6$	$\{19\}$	$\{19, 39\}$	$\{19, 39, 59\}$	$\{19, 39, 59, 79\}$
$5, 7$	$\{23\}$	$\{23, 43, 63, 83\}$	$\{23, 43, 63, \ldots, 183\}_9$	$\{23, 43, 64, \ldots, 323\}_{16}$
$5, 8$	$\{27\}$	$\{27, 47, 67, \ldots, 167\}_8$	$\{27, 47, 67, \ldots, 547\}_{27}$	$\{27, 47, 67, \ldots, 1287\}_{64}$
$5, 9$	$\{31\}$	$\{31, 51, 71, \ldots, 331\}_{16}$	$\{31, 51, 71, \ldots, 1631\}_{81}$	$\{31, 51, 71, \ldots, 5131\}_{256}$
$6, 7$	$\{29\}$	$\{29, 59\}$	$\{29, 59, 89\}$	$\{29, 59, 89, 119\}$
$7, 8$	$\{41\}$	$\{41, 83\}$	$\{41, 83, 125\}$	$\{41, 83, 125, 167\}$
$7, 9$	$\{47\}$	$\{47, 89, 131, 173\}$	$\{47, 89, 131, \ldots, 383\}_9$	$\{47, 89, 131, \ldots, 677\}$
$8, 9$	$\{55\}$	$\{55, 111\}$	$\{55, 111, 167\}$	$\{55, 111, 167, 223\}$

[a] The length of longest words not in S^* is -1 means all words are in S^*.

[b] The subscript k in $\{\ldots\}_k$ means the set contains k elements.

Given any set S of words of lengths m and n, the length llw $\left(\overline{S^*}\right)$ is the Frobenius number of m and some l, where l characterizes the structure of the basis and can be calculated in polynomial time in the measure μ, the total number of symbols in the input. The algorithm will be discussed in Chapter 5.

2.6.5 Bounds on the size of the basis in the 2FPFM

Let S be a set of words of lengths m and n over the alphabet Σ such that S^* is co-finite. Now we consider the number of words in S. By the First Lemma of the 2FPFM, S must contain all words of length m and thus $|\, S \cap \Sigma^m \,| = |\Sigma|^m$. The set Σ^m does not generate a co-finite language in general except when $m = 1$, so S must contain words of length n. Since $\Sigma^m \cup \Sigma^n$ generates a co-finite language, S may contain all words of length n and thus the upper bound $|\, S \cap \Sigma^n \,| \leq |\Sigma|^n$ is tight. By Proposition 2.1.17 (the twins proposition), $|\, S \cap \Sigma^n \,| \geq |\Sigma|^m$. Here I will give a better lower bound for $|\, S \cap \Sigma^n \,|$.

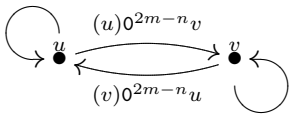
Maximum acyclic spanning subgraph of $\Gamma(m, n)$

Each set S of words of two lengths m, n with $\gcd(m, n) = 1$ corresponds to a word graph $G_S^{(m,n)}$, which is a spanning subgraph of $\Gamma(m, n)$. By Theorem 2.6.6, S^* is co-finite if and only $G_S^{(m,n)}$ contains no cycle and $\Sigma^m \subseteq S$. Since the arcs of $G_S^{(m,n)}$ are in $\Sigma^n \setminus S$, the maximum acyclic spanning subgraph of $\Gamma(m, n)$ is the word graph of the basis of the least size among all such S that S consists of words of lengths m, n and S^* is co-finite.

Lemma 2.6.21. *For any two integers m and n, where $0 < m < n \leq 2m$, let $G = (\Sigma^{n-m}, A, \psi)$ be an acyclic spanning subgraph of the generalized de Bruijn graph $\Gamma(m, n)$. Then*

$$|\, A \,| \leq \frac{|\Sigma|^n - |\Sigma|^m}{2}. \tag{2.105}$$

Proof. By definition, $\Gamma(m, n) = (\Sigma^{n-m}, \Sigma^n, \psi)$. For each vertex u, there are $|\Sigma|^{2m-n}$ loops in $\Gamma(m, n)$ given by $u\Sigma^{2m-n}u$. So there are in total $|\Sigma|^{2m-n} \cdot |\Sigma|^{n-m} = |\Sigma|^m$ loops, which cannot be in A. Now we consider the remaining $|\Sigma|^n - |\Sigma|^m$ arcs.



By comparing lengths, any arc from u to v can be written as uwv for some w of length $2m - n$. Then there is another arc from v to u given by vwu, which forms a cycle given by u, uwv, v, vwu, u. So A contains at most half of the remaining arcs. Hence $|\, A \,| \leq (|\Sigma|^n - |\Sigma|^m)/2$. □

When $1 < m < n < 2m$ and $\gcd(m, n) = 1$, there is a family of examples of acyclic spanning subgraphs of $\Gamma(m, n)$ such that the number of arcs is

$$\frac{|\Sigma|^n - |\Sigma|^m}{2} = \Theta(|\Sigma|^n), \tag{2.106}$$

which corresponds to the sets of words of lengths m, n of least size that generate co-finite languages. The following theorem gives a bound on $|\, A \,|$.

Theorem 2.6.22. *For any integers m and n, where $0 < m < n$, let (Σ^{n-m}, A, ψ) be an acyclic spanning subgraph of the generalized de Bruijn graph $\Gamma(m, n)$. Then*

$$|\, A \,| < \left(1 - \frac{1}{n}\right) |\Sigma|^n . \tag{2.107}$$

Proof. Define the function $\delta : \Sigma^n \to \Sigma^n$, where $\delta(w) = w[m+1..n]w[1..m]$. Starting from each arc $w \in \Sigma^n$, we consider the following walk in $\Gamma(m, n)$:

$$w[1..n-m],\ w,\ w[m+1..n],\ \delta(w),\ \delta(w)[m+1..n],\ \delta(\delta(w)),\ \delta(\delta(w))[m+1..n],\ \ldots, \tag{2.108}$$

where all arcs are conjugates. Now we define a relation $\rightarrow$ on arcs, where $u \rightarrow v$ if starting from the arc u, the walk described in (2.108) eventually visits v.

The relation $\rightarrow$ is an equivalence relation. The reflexive property follows from the definition straightforwardly. Suppose $u \rightarrow v$ and $v \rightarrow w$. Then there are two walks $u[1..n-m], u, u[m+1..n], \delta(u), \ldots, v, v[m+1..n]$ and $v[1..n-m], v, v[m+1..n], \delta(v), \ldots, w, w[m+1..n]$. There is a walk starting from the arc u and visiting w as follows: $u[1..n-m], u, u[m+1..n], \delta(u), \ldots, v, v[m+1..n], \delta(v), \ldots, w, w[m+1..n]$. So $u \rightarrow w$ and thus the transitive property holds. It remains to show the symmetric property. By definition, $v = \delta^i(u)$ for some $i \in \mathbb{N}$. Since for each word of length n, there are at most n conjugates, the walk starting from v must visit some arc t twice, where $t = \delta^j(v) = \delta^l(v)$ for some $j < l$,$j, l \in \mathbb{N}$. The function δ is invertible. So $u = \delta^{-i}(v)$ and $v = \delta^{l-j}(v)$. Then $u = \delta^{(l-j)i-i}(v)$, where $(l-j)i - i \geq 0$. Hence u is in the walk starting from v, and thus $v \rightarrow u$. The symmetric property holds.

Since all arcs in the same walk are conjugates, the relation $\rightarrow$ is a refinement of the conjugates, and thus the number of the $\rightarrow$ equivalence classes is at least as large as the number of necklaces[1] of order n, which is

$$\frac{1}{n} \sum_{d|n} \phi(d)\, |\Sigma|^{\frac{n}{d}} > \frac{|\Sigma|^n}{n}. \tag{2.109}$$

Any acyclic subgraph of $\Gamma(m, n)$ cannot contains an entire $\rightarrow$ equivalence class. So for each of the $\rightarrow$ equivalence classes, there is at least an arc not in the subgraph, and thus $|\, A \,| < \left(1 - \frac{1}{n}\right) |\Sigma|^n$. □

Bounds on the basis size

Theorem 2.6.23. *Let S be a set of words of lengths m and n, where $1 < m < n$, over the alphabet Σ. If S^* is co-finite, then S contains $|\Sigma|^m$ words of length m and more than $\frac{1}{n} |\Sigma|^n$ words of length n.*

[1]The number of necklaces is a classic result in combinatorics, which was discussed by M. E. Jablonshi and M. Moreau in 1892. Refer to MacMahon's paper [104] for more details.

Proof. If S^* is co-finite, by the First Lemma of the 2FPFM, S must contain all words of length m. By Theorem 2.6.6, the word graph $G_S^{(m,n)}$ has no cycle, and by Theorem 2.6.22, $G_S^{(m,n)}$ has at most $\left(1 - \frac{1}{n}\right)|\Sigma|^n$ arcs. So, by definition, S has at least $\frac{1}{n}|\Sigma|^n$ words of length n. □

In order to be co-finite, the set S must contain sufficiently many words of both lengths m and n. Let κ be the number of words in S. Then by Theorem 2.6.23,

$$|\Sigma|^m + \frac{1}{n}|\Sigma|^n < \kappa \leq |\Sigma|^m + |\Sigma|^n . \tag{2.110}$$

This suggests that the running time of an algorithm to decide the co-finiteness of S^* might be exponential in $\nu = \mathrm{llw}(S)$. There is input with exponentially many words such that the input generates a co-finite language but changing any input word will result in a basis that does not generate a co-finite language. So in order to determine co-finiteness for that input, an algorithm must read all input words, which is exponential in $\nu = n$. Another issue raised by this observation is that one can discuss variations on the FPFM where the input, instead of in the form of finitely many words $x_1, x_2, \ldots, x_k$, is given in a compact way, such as a regular expression or an NFA. We will discuss some of those variations in Chapter 3. In particular, in the 2FPFM, when S^* is co-finite, using the words in $\Sigma^m \cup \Sigma^n \setminus S$ to describe S is more concise.

If $1 < m < n < 2m$, by similar reasoning and Lemma 2.6.21, the bound becomes

$$\frac{3}{2}|\Sigma|^m + \frac{1}{2}|\Sigma|^n \leq \kappa \leq |\Sigma|^m + |\Sigma|^n , \tag{2.111}$$

where both the upper bound and the lower bound can be achieved, respectively.

2.7 The FPFM with basis of special sequential lengths

Since there are simple formulae for special cases of the FP where the input satisfies a certain pattern, this suggests the question of the existence of simple formulae or algorithms for the FPFM where input words satisfy a certain pattern. So far, however, we have not found such formulae, although some of the lemmas satisfied in the case of the 2FPFM are also satisfied in some cases with input words satisfying certain patterns.

Proposition 2.7.1. *Let $S, T \subseteq \Sigma^*$, and suppose S^* is co-finite.*

(a) Then $(S \cup T)^*$ is also co-finite, and $\mathrm{llw}\left(\overline{S^*}\right) \geq \mathrm{llw}\left(\overline{(S \cup T)^*}\right)$.

(b) Let U be the set of factors of words in T such that $T \subseteq U^*$. Then $((S \setminus T) \cup U)^*$ is co-finite, and $\mathrm{llw}\left(\overline{S^*}\right) \geq \mathrm{llw}\left(\overline{((S \setminus T) \cup U)^*}\right)$.

Table 2.10: All bases $S \subseteq \Sigma^3 \cup \Sigma^4$ for $\Sigma = \{ \mathtt{0}, \mathtt{1} \}$ with S^* co-finite.

$\Sigma^3 \cup \Sigma^4 \setminus S$	one $w \notin S^*$	$\{ \lvert w \rvert : w \notin S^* \}$	$\lvert \Sigma^* \setminus S^* \rvert$
$\{ \mathtt{0001}, \mathtt{0011}, \mathtt{0101}, \mathtt{0111} \}$	00010000001	$\{ 1, 2, 4, 5, 8, 11 \}$	282
$\{ \mathtt{0001}, \mathtt{0011}, \mathtt{0101} \}$	00010000001	$\{ 1, 2, 4, 5, 8, 11 \}$	200
$\{ \mathtt{0001}, \mathtt{0011}, \mathtt{0111} \}$	00010000001	$\{ 1, 2, 4, 5, 8, 11 \}$	200
$\{ \mathtt{0001}, \mathtt{0011} \}$	00010000001	$\{ 1, 2, 4, 5, 8, 11 \}$	132
$\{ \mathtt{0001}, \mathtt{0101}, \mathtt{0111} \}$	00010000001	$\{ 1, 2, 4, 5, 8, 11 \}$	200
$\{ \mathtt{0001}, \mathtt{0101} \}$	00010000001	$\{ 1, 2, 4, 5, 8, 11 \}$	132
$\{ \mathtt{0001}, \mathtt{0111} \}$	00010000001	$\{ 1, 2, 4, 5, 8, 11 \}$	132
$\{ \mathtt{0001} \}$	00010000001	$\{ 1, 2, 4, 5, 8, 11 \}$	78
$\{ \mathtt{0011}, \mathtt{0101}, \mathtt{0111} \}$	00110000011	$\{ 1, 2, 4, 5, 8, 11 \}$	200
$\{ \mathtt{0011}, \mathtt{0101} \}$	00110000011	$\{ 1, 2, 4, 5, 8, 11 \}$	132
$\{ \mathtt{0011}, \mathtt{0111} \}$	00110000011	$\{ 1, 2, 4, 5, 8, 11 \}$	132
$\{ \mathtt{0011} \}$	00110000011	$\{ 1, 2, 4, 5, 8, 11 \}$	78
$\{ \mathtt{0101}, \mathtt{0111} \}$	01010000101	$\{ 1, 2, 4, 5, 8, 11 \}$	132
$\{ \mathtt{0101} \}$	01010000101	$\{ 1, 2, 4, 5, 8, 11 \}$	78
$\{ \mathtt{0111} \}$	01110000111	$\{ 1, 2, 4, 5, 8, 11 \}$	78
$\{ \mathtt{1000}, \mathtt{1010}, \mathtt{1100}, \mathtt{1110} \}$	10000001000	$\{ 1, 2, 4, 5, 8, 11 \}$	282
$\{ \mathtt{1000}, \mathtt{1010}, \mathtt{1100} \}$	10000001000	$\{ 1, 2, 4, 5, 8, 11 \}$	200
$\{ \mathtt{1000}, \mathtt{1010}, \mathtt{1110} \}$	10000001000	$\{ 1, 2, 4, 5, 8, 11 \}$	200
$\{ \mathtt{1000}, \mathtt{1010}, \}$	10000001000	$\{ 1, 2, 4, 5, 8, 11 \}$	132
$\{ \mathtt{1000}, \mathtt{1100}, \mathtt{1110} \}$	10000001000	$\{ 1, 2, 4, 5, 8, 11 \}$	200
$\{ \mathtt{1000}, \mathtt{1100} \}$	10000001000	$\{ 1, 2, 4, 5, 8, 11 \}$	132
$\{ \mathtt{1000}, \mathtt{1110} \}$	10000001000	$\{ 1, 2, 4, 5, 8, 11 \}$	132
$\{ \mathtt{1000} \}$	10000001000	$\{ 1, 2, 4, 5, 8, 11 \}$	78
$\{ \mathtt{1010}, \mathtt{1100}, \mathtt{1110} \}$	10100001010	$\{ 1, 2, 4, 5, 8, 11 \}$	200
$\{ \mathtt{1010}, \mathtt{1100} \}$	10100001010	$\{ 1, 2, 4, 5, 8, 11 \}$	132
$\{ \mathtt{1010}, \mathtt{1110} \}$	10100001010	$\{ 1, 2, 4, 5, 8, 11 \}$	132
$\{ \mathtt{1010} \}$	10100001010	$\{ 1, 2, 4, 5, 8, 11 \}$	78
$\{ \mathtt{1100}, \mathtt{1110} \}$	11000001100	$\{ 1, 2, 4, 5, 8, 11 \}$	132
$\{ \mathtt{1100} \}$	11000001100	$\{ 1, 2, 4, 5, 8, 11 \}$	78
$\{ \mathtt{1110} \}$	11100001110	$\{ 1, 2, 4, 5, 8, 11 \}$	78
$\emptyset$	00000	$\{ 1, 2, 5 \}$	38

Proof. (a) $S^* \subseteq (S \cup T)^*$, so $(S \cup T)^*$ is also co-finite and the result is straightforward.

(b) $S^* \subseteq (S \cup U)^* = ((S \setminus T) \cup U)^*$; then the result is straightforward. □

The sequence $m, 2m, 3m, \ldots, (k-1)m, n$

As we saw, the First Lemma of the 2FPFM is crucial for the 2FPFM. But for a set S of words of lengths $m, 2m, 3m, \ldots, (k-1)m, n$ over the alphabet Σ, where $(k-1)m < n$, S need not contain all words of length m in order to generate a co-finite language. For example, we can choose even multiples of m and odd n, such as the particular case $2m, n$, and still obtain a basis S consisting of words of those lengths that generates a co-finite language, such as $S = \Sigma^{2m} \cup \Sigma^n$ for $\gcd(2m, n) = 1$, where S contains no word of length m.

There are also bases for which none of $\Sigma^{im} \setminus S$, $\Sigma^{im} \cap S$ is empty for $1 \leq i < k$, and S^* is still co-finite. Let $k, m, n \geq 3$ be integers such that $\gcd(km - m, n) = 1$, and let $T = \Sigma^{(k-1)m} \cup \Sigma^n$. From arbitrary words $x_1, x_2, \ldots, x_{k-2}$ of length $(k-1)m$ and $x_i \neq x_j$ for $i \neq j$, we construct

$$S = (T \setminus \{ x_1, x_2, \ldots, x_{k-2} \}) \cup \bigcup_{1 \leq i \leq k-2} \{ x_i[1..im], x_i[im+1..(k-1)m] \}. \quad (2.112)$$

By Proposition 2.7.1, S also generates a co-finite language, and $T^* \subseteq S^*$. The basis S contains exactly two words of each of the lengths $m, 2m, \ldots, (k-2)m$.

If we require that S contains all words of length m, then S^* contains all words of lengths that are multiples of m and the problem becomes the 2FPFM.

Proposition 2.7.2. *Let S be a set of words of lengths $m, 2m, 3m, \ldots, (k-1)m, n$ such that S^* is co-finite. Then* $\operatorname{llw}\left(\overline{S^*}\right) \geq g(m, l)$*, where $l = n + jm$ and j is the length of the longest path in $G_S^{(m,n)}$.*

Proof. Let $U = \left\{ w[im+1..im+m] : w \in S \setminus \Sigma^n, |\, w \,| = hm, 0 \leq i < h \right\}$, and let $T = S \cap \Sigma^n$, where Σ is the alphabet. If S^* is co-finite, by Proposition 2.7.1, $(U \cup T)^*$ is also co-finite and $\operatorname{llw}\left(\overline{S^*}\right) \geq \operatorname{llw}\left(\overline{(U \cup T)^*}\right)$. Since $U \cup T$ contains only words of lengths m, n, we have $U = \Sigma^m$ and $\operatorname{llw}\left(\overline{(\Sigma^m \cup T)^*}\right) = g(m, l)$, where $l = n + jm$ and j is the length of the longest path in $G_{U \cup T}^{(m,n)} = G_S^{(m,n)}$. □

The sequence $m, n, 2n, 3n, \ldots, (k-1)n$

Lemma 2.7.3. *Let S be a set of words of lengths $m, n, 2n, 3n, \ldots, (k-1)n$, where $0 < m < n$, over the alphabet Σ. If S^* is co-finite, then $\Sigma^m \subseteq S$.*

Proof. If S^* is co-finite, by Proposition 2.1.14, $\gcd\big(m, n, 2n, 3n, \ldots, (k-1)n\big) = 1$, and thus $\gcd(m, n) = 1$. Let $w \in \Sigma^m$. Now we consider the language

$$w(\Sigma^{n-m}w)^*. \tag{2.113}$$

Since S^* is co-finite, $T = w(\Sigma^{n-m}w)^* \cap S^* \neq \emptyset$. Suppose one of the words in T is

$$\tau = wu_1w\cdots u_l = x_1x_2\cdots x_j, \tag{2.114}$$

where all the u_i are of length $n-m$ and all the x_i are in S. Then $|\,\tau\,| = (l-1)n+m$. Since $\gcd(m, n) = 1$, at least one x_i is of length m. By comparing lengths, the first x_i of length m in the factorization (2.114) is equal to w. Hence w is in S and by the arbitrary choice of w, we see that S^* contains all words of length m. □

Lemma 2.7.3 shows that a basis must contain all words of length m in the case that the basis consists of words of lengths of this particular sequence. Lemma 2.7.3 becomes the First Lemma of the 2FPFM in the case where $k = 2$, and thus generalizes the latter.

Let $T = \big\{\, w[in{+}1..in{+}n] : w \in S\backslash\Sigma^m, |\,w\,| = hn, 0 \le i < h \,\big\}$, and $U = S \cap \Sigma^m$. If S^* is co-finite, then by Proposition 2.7.1, $(U \cup T)^*$ is also co-finite. Since $U \cup T$ contains only words of lengths m, n, the set U contains all words of length m (which gives another proof for Lemma 2.7.3) and $\mathrm{llw}\left(\overline{(U \cup T)^*}\right) = g(m, l)$, where $l = n + jm$ and j is the length of the longest path in $G^{(m,n)}_{U \cup T}$. So we can obtain a lower bound $\mathrm{llw}\left(\overline{S^*}\right) \ge \mathrm{llw}\left(\overline{(U \cup T)^*}\right) = g(m, n + jm)$, where j is the length of the longest path in $G^{(m,n)}_{U \cup T}$.

The sequence $m, m + d, m + 2d, \ldots, m + (k-1)d$

In this case, a result analogous to the First Lemma of the 2FPFM does not hold in general. For example, let $S = \Sigma^{m+(k-2)d} \cup \Sigma^{m+(k-1)d}$ for $\gcd(m, m + d, \ldots, m + (k-1)d) = 1$. Since $\gcd(m + (k-2)d, m + (k-1)d) = \gcd(m, d) = (m, m + d, \ldots, m + (k-1)d) = 1$, the language S^* is co-finite. But S contains no word of the lengths $m, m + d, \ldots, m + (k-3)d$.

As we saw in the previous two sequences $m, 2m, 3m, 4m, \ldots, (k-1)m, n$ and $m, n, 2n, 3n, \ldots, (k-1)n$, we can convert an instance of the FPFM with lengths being those sequences into an instance of the 2FPFM, and obtain a lower bound for the longest omitted words. The case of $m, m + d, m + 2d, \ldots, m + (k-1)d$, however, cannot be converted into the 2FPFM easily.

The sequence $m, n, n + m, n + 2m, \ldots, n + (k-2)m$

Since $\gcd(n + m, n) = \gcd(m, n) = 1$, in this case we can construct a basis S consisting of words of lengths $n + im, n + (i+1)m, \ldots, n + (k-2)m$ that generate

a co-finite language. Then a basis may not necessarily contain all words of length m in order to generate a co-finite language. In other words, there is no analogous result to the First Lemma of the 2FPFM. But if the basis contains all words of length m, then a similar result to the Second Lemma of the 2FPFM holds.

Lemma 2.7.4. *Let S be a set of words of lengths $m, n, n+m, n+2m, \ldots, n+(k-2)m$, where $k-2 \leq m < n$, over the alphabet Σ. If $\Sigma^m \subseteq S$ and S^* is co-finite, then $\Sigma^l \subseteq S^*$, where $l = m\,|\Sigma|^{n+(k-3)m} + n + (k-3)m$.*

Proof. Proof by contradiction. Let $x = a_1 a_2 \cdots a_l$ be a word of length l that is not in S^* and $r = |\Sigma|^{n+(k-3)m} + (k-3)$. Then

$$x[im+1..(i+j)m+n] \notin S \tag{2.115}$$

for $0 \leq i \leq r, 0 \leq j \leq k-2, (i+j)m+n \leq l$. Otherwise, $x =$

$$\Big(\prod_{0\leq f<i} x[fm+1..fm+m]\Big)x[im+1..(i+j)m+n]\Big(\prod_{i+j\leq g<r} x[gm+n+1..(g+1)m+n]\Big) \tag{2.116}$$

is a factorization of x into elements of S, since $\Sigma^m \subseteq S$. We define

$$z_i = a_{im+1}a_{im+2}\cdots a_{im+n+(k-3)m}, \quad \text{for } 0 \leq i \leq |\Sigma|^{n+(k-3)m}. \tag{2.117}$$

There are only $|\Sigma|^{n+(k-3)m}$ distinct words of length $n+(k-3)m$ over Σ, but there are $|\Sigma|^{n+(k-3)m}+1$ factors z_i. By the pigeonhole principle, we have $z_p = z_q$, for some $0 \leq p < q \leq |\Sigma|^{n+(k-3)m}$. Now we define

$$u = a_1 a_2 \cdots a_{pm}, \quad v = a_{pm+1}a_{pm+2}\cdots a_{qm}, \quad w = a_{qm+1}a_{qm+2}\cdots a_l. \tag{2.118}$$

Then $x = uvw$ and $v \neq \epsilon$. Since S^* is co-finite, $uv^*w \cap S^*$ is not empty. Let λ be the smallest positive exponent such that $uv^\lambda w \in S^*$. Since $x = uvw \notin S$, we have $\lambda \geq 2$. Now let $uv^\lambda w = x_1 x_2 \cdots x_j$ be a factorization into elements of S. Since $z_q = z_p$, the words $x = uvw$ and $uv^\lambda w$ agree on the first $|uv| + (n+(k-3)m)$ letters. So all factors from x_1 to x_q are of length m. We can write

$$u = x_1 x_2 \cdots x_p, \quad v = x_{p+1}x_{p+2}\cdots x_q. \tag{2.119}$$

By removing the leftmost copy of v from $uv^\lambda w$, the new word

$$uv^{\lambda-1}w = x_1 x_2 \cdots x_p x_{q+1}x_{q+2}\cdots x_j \tag{2.120}$$

is also in S^*, where $\lambda - 1 \geq 1$. This contradicts the minimality of λ. □

Similarly to the bound we obtained for the 2FPFM, when S contains all words of length m and S^* is co-finite, we have

$$\operatorname{llw}\big(\overline{S^*}\big) \leq g(m,l) = g\big(m, m\,|\Sigma|^{n+(k-3)m} + n + (k-3)m\big). \tag{2.121}$$

The upper bound is tight. One can consider the 2FPFM with lengths m and $n+(k-2)m$, which can be viewed as of lengths $m, n, n+m, n+2m, \ldots, n+(k-2)m$. The example that achieves the upper bound in the 2FPFM also attains the equality in the upper bound in (2.121) for the FPFM with sequential lengths.

Chapter 3

Variations on the FPFM and related problems

In this chapter, I will examine variations on the FPFM, some of which can also be viewed as a generalization of the Frobenius problem, and discuss problems related to the FPFM. In §3.1, we will see the variation where the concatenation of words is taken in a fixed order. In §3.2, we will see variations on the FPFM where the input and output are specified by other forms, including deterministic finite automata (DFAs), nondeterministic finite automata (NFAs), regular expressions, deterministic pushdown automata (DPDAs), linear bounded automata (LBAs), and context-sensitive grammars (CSGs). In §3.3, we will see some variations and related problems of the FPFM with different points of view instead of the length of the longest omitted words ($\mathcal{L} = \operatorname{llw}\left(\overline{S^*}\right)$). These problems include

(a) the number of (symbols in) omitted words ($\mathcal{M}$ and $\mathcal{W}$, §3.3.1);

(b) the number of words (and symbols) in a basis that generates a co-finite language (κ, §3.3.2 and §3.3.3);

(c) what words (and integers) can be (the length of) the solution to instances of the FPFM (§3.3.2);

(d) the number of different factorizations of a word w ($\mathcal{D}(w)$, §3.3.3).

In §3.4, I will examine co-finiteness in different settings, including

(i) infinite words (right-infinite Σ^ω, left-infinite ${}^\omega\Sigma$, bi-infinite ${}^\omega\Sigma^\omega$, §3.4.1);

(ii) concatenation with overlap ($\flat$, $\natural$, $\sharp$, §3.4.2);

and discuss co-slender languages (§3.4.3). At the end, in §3.5, I will examine a generalization of the local postage-stamp problem in a free monoid.

The new results are mainly in §3.3, §3.4, and §3.5. My most important results in this chapter include Theorems 3.3.4, 3.4.24, 3.5.9, and a series of propositions in §3.4 concerning co-finiteness.

3.1 Concatenation of words with fixed order

Given k words $x_1, x_2, \ldots, x_k$, not necessarily distinct, we can consider two analogues of the non-negative integer linear combination. One is to consider the language $\{ x_1, x_2, \ldots, x_k \}^*$, which is the FPFM. The other is to consider the language $x_1^* x_2^* \cdots x_k^*$, which perhaps appears closer to the non-negative integer linear combination in the integer FP, which is based on the set

$$\langle x_1, x_2, \ldots, x_k \rangle = \big\{ c_1 x_1 + c_2 x_2 + \cdots + c_k x_k : c_1, c_2, \ldots, c_k \in \mathbb{N} \big\}. \tag{3.1}$$

Problem 3.1.1 (*the FPFM with fixed word order*)**.** *Let Σ be an alphabet. Given k non-empty words $x_1, x_2, \ldots, x_k \in \Sigma^*$ such that there are only finitely many words that are not in the language $x_1^* x_2^* \cdots x_k^*$, then what is the length of the longest such word(s)?*

Shallit asked the FPFM with fixed word order and proved the following theorem in our papers [83, 84] with Shallit and Kao.

Theorem 3.1.2. *[83, 84] Let $x_1, x_2, \ldots, x_k \in \Sigma^+$. Then $L = x_1^* x_2^* \cdots x_k^*$ is co-finite if and only if $|\Sigma| = 1$ and* $\gcd\big(|x_1|, |x_2|, \ldots, |x_k| \big) = 1$.

So the FPFM with fixed word order for the language $x_1^* x_2^* \cdots x_k^*$ is only meaningful over the unary alphabet. When the alphabet is unary, in fact, the two languages $x_1^* x_2^* \cdots x_k^*$ and $\{ x_1, x_2, \ldots, x_k \}^*$ are identical. Hence the FPFM with fixed word order can also be viewed as an equivalent form of the FP in the setting of a free monoid in the same sense that the FPFM over the unary alphabet is an equivalent form of the FP. The language over a unary alphabet is co-finite if and only if $\gcd\big(|x_1|, |x_2|, \ldots, |x_k| \big) = 1$, and the length of the longest omitted words is the Frobenius number $\mathrm{llw}\big(\overline{x_1^* x_2^* \cdots x_k^*}\big) = g\big(|x_1|, |x_2|, \ldots, |x_k| \big)$.

3.2 Variations with different measures

In Chapter 2, various measures of the input and various measures of the output were discussed. In this section, the input words $x_1, x_2, \ldots, x_k$ of the FPFM will be in a compact form other than simply enumerating them. For example, we will use an automaton or a grammar to describe the words $x_1, x_2, \ldots, x_k$. Similarly, the generated language $\{ x_1, x_2, \ldots, x_k \}^*$ can also be described succinctly.

3.2.1 State complexity of the generated language

In this subsection, we will see another variation on the FPFM. The results illustrated in this subsection on state complexity are mainly due to Shallit and Kao. Let $x_1, x_2, \ldots, x_k$ be k words over Σ. For the language $L = x_1^* x_2^* \cdots x_k^*$ or alternatively

$L = \{ x_1, x_2, \ldots, x_k \}^*$, instead of considering $\mathcal{L} = \mathrm{llw}\left(\overline{L}\right)$, the length of the longest words not in L, we now consider $\mathcal{S} = \mathrm{sc}(L)$, the state complexity, and $\mathcal{N} = \mathrm{nsc}(L)$, the nondeterministic state complexity, of L.

Here we do not require L to be co-finite. Discussing $\mathrm{sc}(L)$ and $\mathrm{nsc}(L)$ is sensible when L is regular, which is always true for the two languages obtained by applying Kleene-star to a finite language.

Problem 3.2.1 (*State complexity of star of a finite language*)**.** *Given k words*

$$x_1, x_2, \ldots, x_k, \tag{3.2}$$

what is the state complexity and nondeterministic state complexity of the languages $x_1^* x_2^* \cdots x_k^*$ and $\{ x_1, x_2, \ldots, x_k \}^*$, respectively?

The state complexity of the star of a finite language can be viewed as one generalized form of the Frobenius problem. Theorem 3.2.2 will show that in the unary case the state complexity is strongly related to the Frobenius number of the lengths of given words.

Over a unary alphabet

Let $\Sigma = \{ \mathtt{0} \}$ be the unary alphabet. Then the two languages $x_1^* x_2^* \cdots x_k^*$ and $\{ x_1, x_2, \ldots, x_k \}^*$ are identical. For convenience we use x to represent both the integer x and the unary word of length x.

Theorem 3.2.2. *[83, 84] Let $x_1, x_2, \ldots, x_k$ be k words over the unary alphabet $\{ \mathtt{0} \}$ with $\gcd(x_1, x_2, \ldots, x_k) = 1$. Then $\mathrm{sc}\left(\{ x_1, x_2, \ldots, x_k \}^* \right) = g(x_1, x_2, \ldots, x_k) + 2$.*

Corollary 3.2.3. *[83, 84] Let $x_1, x_2, \ldots, x_k$ be k words over the unary alphabet $\{ \mathtt{0} \}$ and $d = \gcd(x_1, \ldots, x_k)$. Then $\mathrm{sc}\left(\{ x_1, x_2, \ldots, x_k \}^* \right) = d\, g(\frac{x_1}{d}, \frac{x_2}{d}, \ldots, \frac{x_k}{d}) + (d+1)$.*

By the results on the integer FP, it follows that $\mathrm{sc}\left(\{ x_1, x_2, \ldots, x_k \}^* \right) = O\left(\frac{x_k^2}{k}\right)$, which is tight.

State complexity of $\{ x_1, x_2, \ldots, x_k \}^*$

Over a larger alphabet, the two languages $x_1^* x_2^* \cdots x_k^*$ and $\{ x_1, x_2, \ldots, x_k \}^*$ are not necessarily identical. First, we consider the language $\{ x_1, x_2, \ldots, x_k \}^*$.

In 1994, Yu, Zhuang, and Salomaa [174] showed that if S can be accepted by a DFA of n states, then there exists a DFA of at most $2^{n-1} + 2^{n-2}$ states that accepts S^*, and that bound is tight when $n \geq 2$ (for the latter result, also see Maslov [107]). In the notation of the measures defined in §2.2, this translates into

$$\mathcal{S} = \mathrm{sc}(S^*) \leq 2^{\mathrm{sc}(S)-1} + 2^{\mathrm{sc}(S)-2}. \tag{3.3}$$

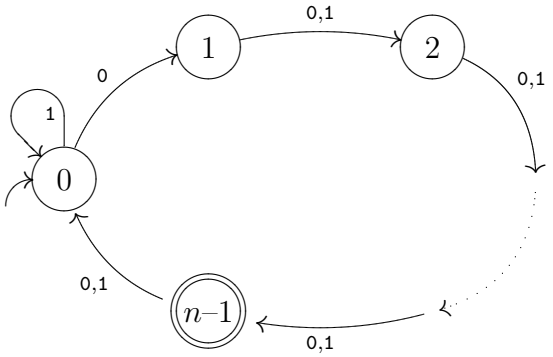


Figure 3.1: Example for the bound $2^{n-1} + 2^{n-2}$ on star operator

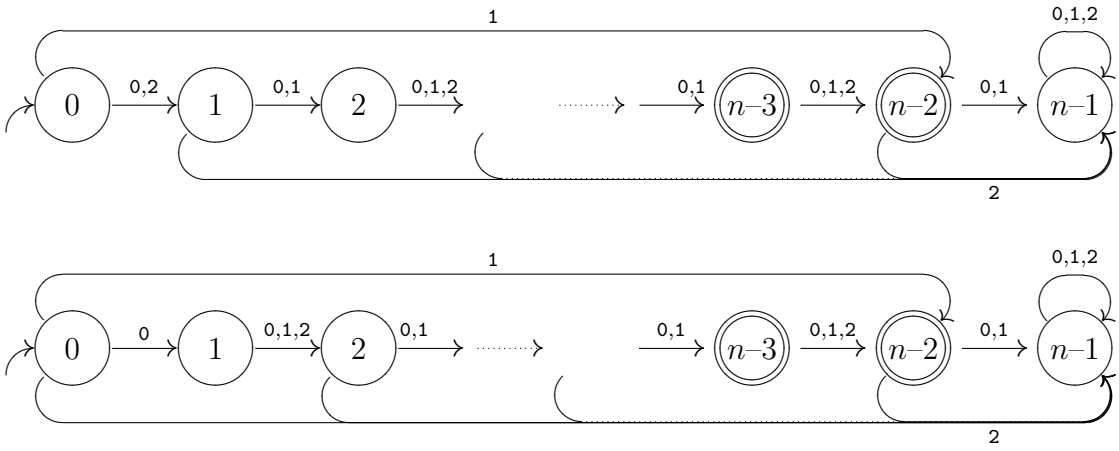


Figure 3.2: Examples for the bound $2^{n-3}+2^{n-4}$ on star operator of finite languages

If S is a finite language, Câmpeanu, Culik, Salomaa, and Yu [29, 28] showed that $2^{n-3} + 2^{n-4}$ states is enough for $n \geq 4$ and that bound is tight when the alphabet size is ≥ 3. That result can be translated into

$$\mathcal{S} = \mathrm{sc}(S^*) \leq 2^{\mathrm{sc}(S)-3} + 2^{\mathrm{sc}(S)-4}. \tag{3.4}$$

In 2003, Holzer and Kutrib [70] examined the nondeterministic counterpart of this problem and showed that if S can be accepted by an NFA of n states, $n \geq 3$, then there exists an NFA of at most $n + 1$ states accepting S^* and that bound is tight. Furthermore, when S is finite and $n \geq 2$, then an NFA of $n-1$ states suffices and that bound is also tight. It is equivalent to say respectively

$$\mathcal{N} = \mathrm{nsc}(S^*) \leq \mathrm{nsc}(S) + 1, \quad \text{and} \quad \mathcal{N} = \mathrm{nsc}(S^*) \leq \mathrm{nsc}(S) - 1. \tag{3.5}$$

By the subset construction, we have

$$\mathcal{S} = \mathrm{sc}(S^*) \le 2^{\mathrm{nsc}(S)+1}, \quad \text{and} \quad \mathcal{S} = \mathrm{sc}(S^*) \le 2^{\mathrm{nsc}(S)-1}. \tag{3.6}$$

Câmpeanu and Ho [27] gave in 2004 tight bounds for the state complexity of the Kleene-star of a finite language in the input measure of the length of the longest words in the input.

Shallit gave the following bounds in our papers [83, 84] with Shallit and Kao.

Proposition 3.2.4. *[83, 84] Let $x_1, x_2, \ldots, x_k$ be k words over the alphabet Σ as the input, and let $\mu = \sum_{1 \le i \le k} |x_i|$ be the total number of symbols in the input.*

(a) $\mathcal{N} = \mathrm{nsc}\left(\{ x_1, x_2, \ldots, x_k \}^* \right) \le \mu - k + 1 = O(\mu)$.

(b) $\mathcal{S} = \mathrm{sc}\left(\{ x_1, x_2, \ldots, x_k \}^* \right) \le 2^{\mu-k+1} = O(2^{\mu})$.

(c) If no x_i is a prefix of any other x_j, then $\mathcal{S} = \mathrm{sc}\left(\{ x_1, x_2, \ldots, x_k \}^* \right) \le \mu - k + 2$.

Proof. (a) We can construct an NFA as in Figure 3.3, where each cycle of states

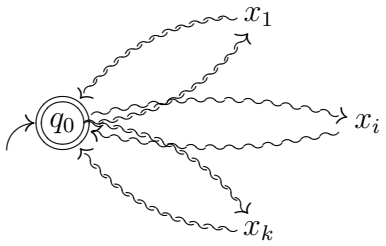


Figure 3.3: An NFA accepting $\{ x_1, x_2, \ldots, x_k \}^*$

starts from a common initial state q_0, accepts one of the words $x_1, x_2, \ldots, x_k$, and returns to q_0. The NFA has at most $\mu - k + 1$ states.

(b) Change the NFA in part (a) into a DFA by applying the subset construction.

(c) Add one "dead" state that absorbs all unspecified transitions in the NFA given in part (a), and combine those cycles that present words with a common prefix by using a shared path of states to read that prefix. If no x_i is a prefix of any other x_j, then by doing so, the NFA in part (a) becomes a DFA. □

Now we consider the upper bound on $\mathrm{sc}(S^*)$ in the measure $\nu = \max_{w \in S} |\, w \,|$.

Theorem 3.2.5. *[83, 84] Let $x_1, x_2, \ldots, x_k$ be k words over the alphabet Σ as the input, and $\nu = \max_{1 \le i \le k} |\, x_i \,|$ be the length of the longest words. Then*

$$\mathcal{S} = \mathrm{sc}\left(\{ x_1, x_2, \ldots, x_k \}^* \right) \le \frac{2}{2\,|\Sigma| - 1}(2^{\nu}\,|\Sigma|^{\nu} - 1) = O(2^{\nu}\,|\Sigma|^{\nu}). \tag{3.7}$$

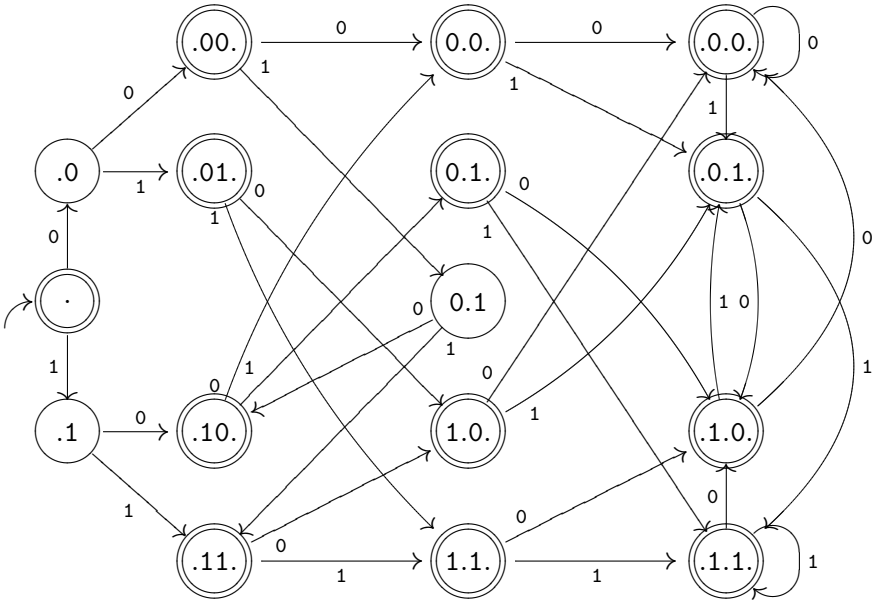


Figure 3.4: The DFA accepts $(\Sigma^2 \cup \Sigma^3 \setminus \{001\})^*$

Proof. Let $S = \{ x_1, x_2, \ldots, x_k \}$. The idea is to create a DFA $M = (Q, \Sigma, \delta, q_0, F)$ accepting $\{ x_1, x_2, \ldots, x_k \}^*$, which keeps track of the last $\nu - 1$ symbols scanned, together with a set of ν signals, which represent whether a factorization of the input word into words in the basis S could occur at the position after each of the last ν symbols scanned. More formally,

$$Q = \left\{ [w, T] : w \in \Sigma^*,\ |\, w \,| < \nu,\ T \subseteq \{ 0, 1, \ldots, |\, w \,| \} \right\} \tag{3.8}$$

$$q_0 = [\epsilon, \{ 0 \}], \tag{3.9}$$

$$F = \left\{ [x, T] : 0 \in T \right\},\ \text{and} \tag{3.10}$$

$$\delta([x, T], a) = [xa, U],\ \text{if } |\, x \,| < \nu - 1, \text{ where} \tag{3.11}$$

$$U = \begin{cases} \left\{ 0, \nu + 1 : \nu \in T \right\}, & \text{if } x[\nu - i.. \,|\, x \,|]a \in S \text{ for some } i \in T; \\ \left\{ \nu + 1 : \nu \in T \right\}, & \text{otherwise;} \end{cases}$$

$$\delta([x, T], a) = [x[2 \ldots |\, x \,|]a, U],\ \text{if } |\, x \,| = \nu - 1, \text{ where} \tag{3.12}$$

$$U = \begin{cases} \left\{ 0, \nu + 1 : \nu \in T \right\} \setminus \{ \nu \}, & \text{if } x[\nu - i.. \,|\, x \,|]a \in S \text{ for some } i \in T; \\ \left\{ \nu + 1 : \nu \in t \right\} \setminus \{ \nu \}, & \text{otherwise.} \end{cases}$$

Then $L(M) = S^*$ and the number of states of M is

$$\sum_{0 \le i < n} |\Sigma|^i \, 2^{i+1} = \frac{2}{2\,|\Sigma| - 1}(2^\nu \,|\Sigma|^\nu - 1) = O(2^\nu \,|\Sigma|^\nu). \tag{3.13}$$

□

Shallit gave in our papers [83, 84] with Shallit and Kao a construction of a family of finite languages L over the binary alphabet such that $\mathrm{sc}(L^*)$ is exponential for

each L, where the number of words in L is linear in the measure $\mathrm{nsc}(L)$ and also in $\nu = \max_{w \in L} |\, w \,|$.

Let $t \geq 2$ be an integer, and define words as follows:

$$y = \mathtt{0}1^{t-1}\mathtt{0}, \qquad x_i = 1^{t-i-1}\mathtt{0}1^{i+1}, \quad 0 \leq i \leq t-2\ . \tag{3.14}$$

Let $S_t = \{\, \mathtt{0}, x_0, x_1, \ldots, x_{t-2}, y \,\}$. For example,

$$S_6 := \{\, \mathtt{0}, \mathtt{1111101}, \mathtt{1111011}, \mathtt{1110111}, \mathtt{1101111}, \mathtt{1011111}, \mathtt{0111110} \,\}\,. \tag{3.15}$$

Theorem 3.2.6. *[83, 84] S_t^* has state complexity $3t2^{t-2} + 2^{t-1}$.*

Corollary 3.2.7. *[83, 84] As usual, let κ be the number of words, ν be the length of the longest words, and μ be the total number of symbols in the finite language as input. There exists a family of word sequences $x_1, x_2, \ldots, x_\kappa$, where $\nu = \kappa$, such that the state complexity $\mathcal{S} = \mathrm{sc}\left(\{\, x_1, x_2, \ldots, x_\kappa \,\}^*\right)$ satisfies*

$$\mathcal{S} = 2^{\Theta(\kappa)}, \quad \mathcal{S} = 2^{\Theta(\nu)}, \quad \textit{and} \quad \mathcal{S} = 2^{\Theta(\sqrt{\mu})}. \tag{3.16}$$

State complexity of $x_1^* x_2^* \cdots x_k^*$

Now we will examine the state complexity of $x_1^* x_2^* \cdots x_k^*$. To avoid confusion, the state complexity and nondeterministic state complexity of $x_1^* x_2^* \cdots x_k^*$, as output measures, are written as $\mathcal{S}'$ and $\mathcal{N}'$ respectively.

Proposition 3.2.8. *Let $x_1, x_2, \ldots, x_k$ be k words over the alphabet Σ as the input, and let $\mu = \sum_{1 \leq i \leq k} |x_i|$ be the total number of symbols in the input.*

(a) $\mathcal{N}' = \mathrm{nsc}(x_1^* x_2^* \cdots x_k^*) \leq \mu = O(\mu)$.

(b) $\mathcal{S}' = \mathrm{sc}(x_1^* x_2^* \cdots x_k^*) \leq 2^\mu = O(2^\mu)$.

(c) If no x_i is a prefix of any other x_j, then $\mathcal{S} = \mathrm{sc}(x_1^* x_2^* \cdots x_k^*) \leq \mu + 1$.

Proof. (a) We can construct an NFA-ϵ as in Figure 3.5, where each cycle of states

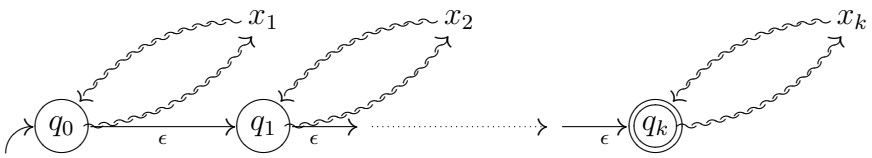


Figure 3.5: An NFA accepting $x_1^* x_2^* \cdots x_k^*$

accepts one x_i^*, and q_k is the only final state. Converting the NFA-ϵ into an NFA, the number of states will not change, which is μ.

(b) Change the NFA in part (a) into a DFA by applying the subset construction.

(c) Add a "dead" state absorbing all unspecified transitions in the NFA-ϵ given in part (a), and omit all ϵ transitions and let δ be the function for the remaining transitions. If no x_i is a prefix of any other x_j, then there is a common prefix $v_{ij} \in \Sigma^*$ for each pair of words x_i and x_j such that $x_i = v_{ij}ay_i, x_j = v_{ij}by_j$, where $a \neq b$, $a, b \in \Sigma$. Without loss of generality, we assume $i < j$. Then add a new transition from the state $\delta(q_i, v)$ to $\delta(q_j, vb)$ labeled by b. By doing so, the NFA-ϵ becomes a DFA. □

Using similar ideas from the results on $\{ x_1, x_2, \ldots, x_k \}^*$, Kao created an example achieving exponential state complexity $\mathcal{S}' = \mathrm{sc}(x_1^* x_2^* \cdots x_k^*)$, in our papers [83, 84] with Shallit and Kao.

Theorem 3.2.9. *[83, 84] As before, define* $y = \mathtt{0}\mathtt{1}^{t-1}\mathtt{0}, x_i = \mathtt{1}^{t-i-1}\mathtt{0}\mathtt{1}^{i+1}, 0 \leq i \leq t-2$. *Let* $L = (\mathtt{0}^* x_0^* x_1^* \cdots x_{t-2}^* y^*)^e$ *where* $e = (t+1)(t-2)/2 + 2t$. *Then* $\mathrm{sc}(L) \geq 2^{t-2}$.

Corollary 3.2.10. *[83, 84] As usual, let* κ *be the number of words,* ν *be the length of the longest words, and* μ *be the total number of symbols in the finite language as input. There exists a family of word sequences* $x_1, x_2, \ldots, x_\kappa$, *where* $\kappa = \Theta(\nu^2)$, *such that the state complexity* $\mathcal{S}' = \mathrm{sc}(x_1^* x_2^* \cdots x_\kappa^*)$ *satisfies*

$$\mathcal{S}' = 2^{\Omega(\sqrt{\kappa})}, \quad \mathcal{S}' = 2^{\Omega(\nu)}, \quad \textit{and} \quad \mathcal{S}' = 2^{\Omega(\sqrt[4]{\mu})}. \tag{3.17}$$

State complexity for two words

Kao and Shallit also discussed the state complexity in the case $\kappa = 2$ in our papers [83, 84]. Here $g(a, b)$ is the Frobenius number of a and b.

Theorem 3.2.11. *[83, 84] Let* $x_1, x_2 \in \Sigma^+$. *Then*

$$\mathrm{sc}(\{ x_1, x_2 \}^*) \leq \begin{cases} |x_1| + |x_2|, & \textit{if } x_1 x_2 \neq x_2 x_1; \\ d \cdot g\left(|x_1|/d, |x_2|/d\right) + (d+2), & \textit{if } x_1 x_2 = x_2 x_1, \end{cases} \tag{3.18}$$

where $d = \gcd(|x_1|, |x_2|)$. *Furthermore, this bound is tight.*

Theorem 3.2.12. *[83, 84] Let* $x_1, x_2 \in \Sigma^+$. *Then*

$$\mathrm{sc}(x_1^* x_2^*) \leq \begin{cases} |x_1| + 2|x_2|, & \textit{if } x_1 x_2 \neq x_2 x_1; \\ d \cdot g\left(|x_1|/d, |x_2|/d\right) + (d+2), & \textit{if } x_1 x_2 = x_2 x_1, \end{cases} \tag{3.19}$$

where $d = \gcd(|x_1|, |x_2|)$.

In this section, we discussed the state complexity $\mathcal{S} = \mathrm{sc}(L)$ (or $\mathcal{S}' = \mathrm{sc}(L')$) and nondeterministic state complexity $\mathcal{N} = \mathrm{nsc}(L)$ (or $\mathcal{N}' = \mathrm{nsc}(L')$) of a generated language $L = \{ x_1, x_2, \ldots, x_k \}^*$ (or with fixed order as $L' = x_1^* x_2^* \cdots x_k^*$).

Since the FPFM is about the words not in L, a natural question is to discuss the state complexity $\bar{\mathcal{S}}$ (or $\bar{\mathcal{S}}'$) and nondeterministic state complexity $\bar{\mathcal{N}}$ (or $\bar{\mathcal{N}}'$) of the complement of L (or L'). Since the complement operation does not change the number of states of a DFA, we have

$$\bar{\mathcal{S}} = \mathcal{S}, \qquad \bar{\mathcal{S}}' = \mathcal{S}', \tag{3.20}$$

and the bounds on the state complexity of L (or L') are also bounds on the state complexity of the complement of L (or L'). Every DFA is also an NFA, so any upper bound on the state complexity of L (or L') is also an upper bound on the nondeterministic state complexity of the complement of L (or L') given by

$$\bar{\mathcal{N}} \leq \mathcal{S}, \qquad \bar{\mathcal{N}}' \leq \mathcal{S}'. \tag{3.21}$$

3.2.2 Input in other forms

Representation by complement

As we saw in the 2FPFM regarding words of lengths m and n, the number of words must be exponential in n in order to generate a co-finite language, and each of the bases S is essentially in the form

$$S = \Sigma^m \ \cup \ \Sigma^n \setminus T \tag{3.22}$$

for some T. So instead of enumerating all words in S, one easy way to represent the basis S in an instance of the 2FPFM is to list the words in T instead and give the two integers m, n (since the information about m is missing from the set T). More generally, if S is a set of words of lengths $c_1, c_2, \ldots, c_k$, then instead of enumerating S, the complement defined by

$$T = (\Sigma^{c_1} \ \cup \ \Sigma^{c_2} \ \cup \ \cdots \ \cup \ \Sigma^{c_k}) \setminus S \tag{3.23}$$

may be of smaller size, and thus the basis can be represented by T and the integers $c_1, c_2, \ldots, c_k$. Then it is possible that T may be more succinctly described than S and provides an entirely different family of measures. In this way, we can reduce the input size by giving the complement of the basis in some set.

Problem 3.2.13 (the FPFM with complement input)**.** *Given a set S of words of length $c_1, c_2, \ldots, c_k$ over Σ such that $L = ((\Sigma^{c_1} \ \cup \ \Sigma^{c_2} \ \cup \ \cdots \ \cup \ \Sigma^{c_k}) \setminus S)^*$ is co-finite, then what is the longest word(s) not in L?*

In general, $\mathrm{llw}\left(\overline{S^*}\right)$ is not bounded in $|\,S\,|$, which will be discussed in a later section as shown in Theorem 3.3.4. In the 2FPFM, however, it is bounded. For a basis S of words of lengths m, n that generates a co-finite language, we have $\mathcal{L} = \mathrm{llw}\left(\overline{S^*}\right) \leq g(m, m\,|\Sigma|^{n-m} + n - m)$ and $|\,S\,| > |\Sigma|^m + |\Sigma|^n / n$. Hence there is a trivial linear upper bound on $\mathcal{L}$ in the size of S as

$$\mathcal{L} < g\left(m, mn\frac{|\,S\,|}{|\Sigma|^m} - mn + n - m\right), \tag{3.24}$$

which is not tight. There is a tight linear bound on $\mathcal{L}$ in the size of $T = \Sigma^m \cup \Sigma^n \setminus S$ as

$$\mathcal{L} \leq g\left(m, n + m \min\left\{\, |\Sigma|^{n-m} - 1, |\, T \,| \,\right\}\right), \tag{3.25}$$

which is an immediate result from the following proposition.

Proposition 3.2.14. *Let S be a set of words of lengths m and n over the alphabet Σ, where $1 < m < n$, and S^* is co-finite. Then* $\mathrm{llw}\left(\overline{S^*}\right) \in \varkappa$, *where*

$$\varkappa = \left\{\, g(m,l) : l = n, n+m, \ldots, n+im, \ldots, n+m \min\left\{\, |\Sigma|^{n-m} - 1, |\, T \,| \,\right\} \,\right\}, \tag{3.26}$$

and $T = \Sigma^m \ \cup \ \Sigma^n \setminus S$.

Proof. By the equivalence of the 2FPFM and word graph, the longest words not in S^* is of length $g(m,l)$, where l is the longest path in the word graph $G_S^{(m,n)}$. Since each arc in $G_S^{(m,n)}$ corresponds to a word of length n that is not in S, the size of T is $\geq l$. On the other hand, there are in total $|\Sigma|^{n-m}$ vertices in $G_S^{(m,n)}$. So $l \leq |\Sigma|^{n-m} - 1$. Otherwise there is a cycle in $G_S^{(m,n)}$ and S^* cannot be co-finite. □

Representation by regular expressions

There are also other compact ways to describe the basis, for example, by regular expressions. We know that the length of a regular expression for the finite language $S = \{\, x_1, x_2, \ldots, x_k \,\}$ is bounded as follows:

$$\mathrm{alph}(S) \leq \mu = \sum_{1 \leq i \leq k} |\, x_i \,|, \tag{3.27}$$

and thus a regular expression is at least as succinct as simply enumerating the finite language S, where $\mathrm{alph}(S)$ is the smallest number of alphabetic symbols in a regular expression for S. Although there is no standard definition for the length of a regular expression, Ellul, Krawetz, Shallit, and Wang [41] showed in 2004 that some common definitions on the length of a regular expression are bounded in each other only up to a constant factor.

In 1980, Leiss [94] gave an algorithm (also see Glushkov [54]), by which an NFA with $\leq \mathrm{alph}(S) + 1$ states can be built to accept S. In other words,

$$\mathrm{nsc}(S) \leq \mathrm{alph}(S) + 1, \quad \mathrm{sc}(S) \leq 2^{\mathrm{alph}(S)} + 1, \tag{3.28}$$

and thus an NFA is at least as succinct as a regular expression in the sense of representing the basis S.

In addition, if S is finite, we know that $\mathrm{llw}\,(S)$ must be bounded in the number of alphabetic symbols (or the number of states) of a regular expression (or a DFA or an NFA) for S, and thus for the finite language $S = \{\, x_1, x_2, \ldots, x_k \,\}$,

$$\nu = \max_{1 \leq i \leq k} |\, x_i \,| \leq \mathrm{alph}(S), \ \nu = \max_{1 \leq i \leq k} |\, x_i \,| < \mathrm{nsc}(S), \ \nu = \max_{1 \leq i \leq k} |\, x_i \,| < \mathrm{sc}(S). \tag{3.29}$$

Therefore, some of the bounds in previous chapters in other measures can easily be converted into bounds in $\mathrm{alph}(S)$. For example, by $\mathcal{L} < 2^{\mathrm{nsc}(S)-1}$, it follows

$$\mathcal{L} = \mathrm{llw}\left(\overline{S^*}\right) < 2^{\mathrm{alph}(S)}. \tag{3.30}$$

Representation by automata

The case of input being represented by DFAs and NFAs are discussed in §3.2.1. In Chapter 4, I will show how to construct a family of examples in the 2FPFM, in which $\mathrm{llw}\left(\overline{S^*}\right)$ is exponential in $\mathrm{llw}\,(S)$. In other words, when the alphabet is of size ≥ 2, the following bound

$$\mathcal{L} = O\big(\,|\Sigma|^{(1+\log_{|\Sigma|}2)\nu}\,\big) \tag{3.31}$$

is nearly tight in the sense there are examples such that

$$\breve{\mathcal{L}} = \Theta(|\Sigma|^{\nu}), \tag{3.32}$$

where the "breve" sign means the bound is for some particular family of examples. Based on the example, Shallit showed that an NFA with $O(\nu^2)$ states and a regular expression of length $O(\nu^2 \log \nu)$ can be constructed such that the longest words as solutions to the instance of the 2FPFM can be exponential. In other words,

$$\mathcal{L} = O\big(2^{\mathrm{nsc}(S)}\big), \quad \breve{\mathcal{L}} = \Theta\big(\,|\Sigma|^{\sqrt{\mathrm{nsc}(S)}}\,\big), \quad \mathcal{L} = O\big(2^{\mathrm{alph}(S)}\big), \quad \breve{\mathcal{L}} = \Theta\big(\,|\Sigma|^{\sqrt[2+\varepsilon]{\mathrm{alph}(S)}}\,\big). \tag{3.33}$$

Details will be given in Chapter 4.

Now we consider pushdown automata (PDA). A deterministic pushdown automata (DPDA) can be effectively constructed to accept each of the bases in my exponential examples in the 2FPFM, and thus we know that the upper bound on $\mathcal{L}$ must be at least exponential in the size of the DPDA accepting the basis. Nevertheless, so far, an upper bound has not been obtained yet, nor do we know whether it is bounded. For the same reason, the upper bound on $\mathcal{L}$ in the measure of the size of the PDA accepting the basis is at least exponential.

Now we consider linear bounded automata (LBA). For input being represented by a LBA (or more generally a TM), $\mathcal{L}$ is unbounded in the size of the automaton. In order to show that, we need an uncomputable function.

Problem 3.2.15 (*Busy beaver problem*)**.** *[127] Let M be a deterministic Turing machine of $n+1$ states with a single doubly-infinite tape, where the tape alphabet is $\{\,\mathtt{1}, \mathtt{B}\,\}$ and $\mathtt{B}$ is the blank symbol. Initially, the tape is completely blank. At each step, the tape head of M must move either right or left. There is a single halting state that has no out-going transitions. Then what is the maximum number of $\mathtt{1}$'s on the tape when M halts, say $\Sigma(n)$, and what is the maximum number of moves M made when M halts, say $S(n)$?*

The busy beaver problem was introduced by Rado in 1962. Neither the function $\Sigma(n)$ nor the function $S(n)$ are computable functions [127].

Theorem 3.2.16. *There is a family of LBAs such that for each M, the language $L = L(M)$ generates a co-finite language $L^* = \big(L(M)\big)^*$ in Σ^* and* $\operatorname{llw}\big(\overline{L^*}\big)$ *is not bounded by any computable function in the number of states of M.*

Proof. Construct the LBA M_k as follows: on input x, it simulates a k-state busy beaver Turing machine N_k on a second track for $k \geq 2$. After each move of the N_k, the machine M_k then erases a letter of the input. Before N_k halts, if at any point the input tape becomes full of blank symbols, then we reject. If the simulated N_k halts, then compare the number of remaining letters of the input and the number of blanks on the input tape. If the former is less than the latter, then accept; otherwise, reject. By the construction, at the i-th step, the simulated busy beaver TM can move its head at most to the right and to the left i squares away. So, if M_k accepts an input, then the simulation of the k-state busy beaver TM will not move past the endmarkers of the LBA.

By the construction, the machine M_k has $k + O(1)$ states and accepts all words w of length $S(k) \leq |\, w \,| < 2S(k)$, where S is Rado's uncomputable function given above. Since $S(k) \geq S(2) = 6$ for $k \geq 2$ (see [100]), the basis $L(M_k)$ generates a co-finite language, where the longest words not in the generated language are of length $S(k) - 1$ by Eq. (1.72) and $S(k) - 1$ is not bounded by any computable function. □

Representation by grammar

For input represented by a CSG (or more generally an unrestricted grammar), the length of the longest words not in a generated language is unbounded by any computable function in the number of symbols in the productions of the grammar. This result is an immediate corollary of Theorem 3.2.16.

Corollary 3.2.17. *There is a family of CSGs such that for each G, the language $L = L(G)$ generates a co-finite language L^* in Σ^* and* $\operatorname{llw}\big(\overline{L^*}\big)$ *is unbounded by any computable function in the number of symbols in the productions of G.*

Proof. By the equivalence of CSGs and LBAs [91, 89], there is a CSG G that represents each of the languages $L(M_k) \setminus \{\, \epsilon \,\} = L(M_k)$, where M_k is the machine constructed in the proof of Theorem 3.2.16 and the number of symbols in the productions of G is linear in the number of transition plus the number of states of M_k. By Theorem 3.2.16, the length of the longest words not in $L(G)^*$ is unbounded by any computable function in the number of states of M_k, while the number of symbols in the productions of G is quadratic in the number of states of M_k. □

The construction using the busy beaver TM in the previous discussion is due to Shallit [154].

3.2.3 Output in other forms

As usual, let S be a set of words over the alphabet Σ. The cases of output being represented by DFAs and NFAs are discussed in §3.2.1. As shown in Table 2.5 on page 39, some of the measures are bounded in each other.

Let $\mathcal{S}$ be the number of states in the minimal DFA accepting S^*, $\mathcal{N}$ be the minimal number of states of an NFA accepting S^*, $\mathcal{R}$ be the minimal number of symbols in a regular expression for S^*, and $\bar{\mathcal{S}}, \bar{\mathcal{N}}, \bar{\mathcal{R}}$ be those for the complement of S^* respectively. Then their relations are

$$\mathcal{N} \leq \mathcal{S} \leq 2^{\mathcal{N}}, \qquad \bar{\mathcal{N}} \leq \bar{\mathcal{S}} \leq 2^{\bar{\mathcal{N}}}, \tag{3.34}$$

$$\mathcal{N} - 1 \leq \mathcal{R} \leq |\Sigma|\, \mathcal{N} 4^{\mathcal{N}}, \qquad \bar{\mathcal{N}} - 1 \leq \bar{\mathcal{R}} \leq |\Sigma|\, \bar{\mathcal{N}} 4^{\bar{\mathcal{N}}}, \tag{3.35}$$

where the upper bound for regular expressions follows from the McNaughton-Yamada algorithm [110].

Suppose S^* is co-finite, and let $\mathcal{L}$ be the length of the longest words not in S^*. By Proposition 2.3.1, a relation between $\mathcal{L}$ and the state complexity of S^* (or complement of S^*) is given by

$$\mathcal{L} < \mathcal{S} = \bar{\mathcal{S}}. \tag{3.36}$$

Proposition 3.2.18. *Let S be a set of words over Σ such that S^* is co-finite., and L be the complement of S^*.*

(a) $\operatorname{llw}(L) \leq \operatorname{nsc}(L)$;

(b) $\operatorname{llw}(L) \leq \operatorname{alph}(L)$.

Proof. Let M be any NFA accepting L and R be any regular expression for L.

(1) Similar to the proof of Proposition 2.3.1. We prove the result by contradiction. Assume $w \in \overline{S^*} = L(M)$ and $|\, w \,| \geq n$, the number of states in the NFA M. Then M accepts w and M visits $n+1$ states to accept w. There must be one state that M visits at least twice. Suppose $\delta(q_0, u) = \delta(q_0, uv)$, where $v \neq \epsilon$, and $w = uvz$. Then M accepts all words uv^*z, which are not in S^*. Then S^* cannot be co-finite.

(2) We prove instead that if $L(R)$ is finite, then $\operatorname{llw}(L(R)) \leq \operatorname{alph}(R)$, by induction on the structure of a regular expression R. If R is ϵ, then $\operatorname{llw}(L(R)) = 0 \leq \operatorname{alph}(R)$. If R is a single letter, then $\operatorname{llw}(L(R)) = 1 \leq \operatorname{alph}(R)$. If $R = R_1 + R_2$, by induction, $\operatorname{llw}(L(R)) = \max\{\operatorname{llw}(R_1), \operatorname{llw}(R_2)\} \leq \max\{\operatorname{alph}(R_1), \operatorname{alph}(R_2)\} \leq \operatorname{alph}(R_1 + R_2)$. If $R = R_1R_2$, then by induction $\operatorname{llw}(L(R)) = \operatorname{llw}(R_1) + \operatorname{llw}(R_2) \leq \operatorname{alph}(R_1) + \operatorname{alph}(R_2) = \operatorname{alph}(R_1R_2)$. If $R = R_1^*$, then since $L(R)$ is finite, $L(R_1)$ contains only ϵ. In this case, $\operatorname{llw}(L(R)) = 0 \leq \operatorname{alph}(R_1)$. Therefore, $\operatorname{llw}(L(R)) \leq \operatorname{alph}(R)$, and thus if S^* is co-finite, then $\operatorname{llw}\left(\overline{S^*}\right) \leq \operatorname{alph}(S^*)$. □

Then we have additional bounds on $\mathcal{L} = \mathrm{llw}\left(\overline{S^*}\right)$, in the nondeterministic state complexity of S^* (and complement of S^*) and in the minimal alphabetic length of a regular expression for S^* (and complement of S^*) as follows:

$$\mathcal{L} < \bar{\mathcal{N}}, \qquad \mathcal{L} \leq \bar{\mathcal{R}}, \qquad \mathcal{L} < 2^{\mathcal{N}}, \qquad \mathcal{L} < 2^{(\mathcal{R}+1)}, \tag{3.37}$$

where the exponential bounds follow from $\mathcal{L} < \mathcal{S}$.

As we saw, some measures of the input are more succinct than others in the sense of considering the bound on $\mathcal{L}$. For example, LBAs and more powerful automata, CSGs and more powerful grammars are more succinct, since $\mathcal{L}$ is unbounded in those measures. For DPDAs, we know that at least there are examples where $\mathcal{L}$ is exponential in the size of the DPDA accepting the basis. For NFAs and regular expressions, the bound on $\mathcal{L}$ is exponential, and there are examples to achieve an exponential $\mathcal{L}$. In the same sense, the number of distinct words in the input, $\kappa = |\, S \,|$, is succinct, since $\mathcal{L}$ is unbounded in κ. The length of the longest words in the input, $\nu = \max_{w \in S} |\, w \,|$, is almost as succinct as NFAs, since there is a tight exponential bound on $\mathcal{L}$ in ν. For DFAs and the total number of symbols in the input, $\mu = \sum_{w \in S} |\, w \,|$, there is an exponential bound on $\mathcal{L}$, but we do not yet know whether it is tight.

For different measures of the generated language S^*, most of them provide an good upper bound on $\mathcal{L}$, except nondeterministic state complexity $\mathcal{N} = \mathrm{nsc}(S^*)$ and the minimal alphabetic length of the regular expression $\mathcal{R} = \mathrm{alph}(S^*)$.

3.3 Variations with different aspects

Let $x_1, x_2, \ldots, x_k$ be a basis. Instead of asking for the longest words not in $L = \{\, x_1, x_2, \ldots, x_k \,\}^*$, there are other topics related to L and factorizations of words into $x_1, x_2, \ldots, x_k$. For example, what is the number of all words not in L? If L is co-finite, what is the possible size of k? Can every word be a solution to an instance of the FPFM with some $x_1, x_2, \ldots, x_k$? For a word w, in how many different ways can w be factorized into a concatenation of $x_1, x_2, \ldots, x_k$?

3.3.1 Number of words and number of symbols

Let S be a set of words such that S^* is co-finite. The properties of the longest words not in S^* are discussed in Chapter 2. There are two other topics, one of which concerns all words not in S^* and the other concerns all words in S. The number of these words and total number of symbols contained in those words will be discussed in this section.

All words not in a generated co-finite language

Let $x_1, x_2, \ldots, x_k$ be k integers. Recall that $\langle x_1, x_2, \ldots, x_k \rangle$ is the set of all non-negative integers that can be written as non-negative integer linear combinations of the given integers $x_1, x_2, \ldots, x_k$. As we saw in the integer FP, the number of all positive integers not in $\langle x_1, x_2, \ldots, x_k \rangle$, denoted by $h(x_1, x_2, \ldots, x_k)$, is an interesting topic and the relation to the Frobenius number is given by Eq. (1.56) as follows:

$$\frac{1}{2}\big(g(x_1, x_2, \ldots, x_k) + 1\big) \leq h(x_1, x_2, \ldots, x_k) \leq g(x_1, x_2, \ldots, x_k). \tag{3.38}$$

Now let $\{ x_1, x_2, \ldots, x_k \}$ be k words. The FPFM, as discussed in Chapter 2, is to ask for the longest words that are not in $\{ x_1, \ldots, x_k \}^*$. The analog of $h(x_1, x_2, \ldots, x_k)$ in the setting of a free monoid is the following problem, which also exhibits different properties from its integer counterpart as the FPFM does.

Problem 3.3.1 (*the FPFM, all words*)**.** *Given k non-empty words $x_1, x_2, \ldots, x_k \in \Sigma^*$ such that there are only finitely many words that cannot be written as concatenations of words in $\{ x_1, x_2, \ldots, x_k \}$, then what are those words?*

Let $S = \{ x_1, x_2, \ldots, x_k \}$ such that S^* is co-finite. Two measures can be used to describe those words not in S^*. One is $\mathcal{M} = \big| \overline{S^*} \big|$, the number of words not in S^*, and the other is $\mathcal{W} = \sum_{w \in \overline{S^*}} | w |$, the total number of symbols of words not in S^*. They are generalizations of the corresponding problems for integers, or over a unary alphabet

$$\mathcal{M}_\wr = h(| x_1 |, | x_2 |, \ldots, | x_k |) = O(\nu^2), \tag{3.39}$$

$$\mathcal{W}_\wr = \sum \big\{ i \in \mathbb{N} : i \notin \langle | x_1 |, | x_2 |, \ldots, | x_k | \rangle \big\} = O(\nu^4), \tag{3.40}$$

where $\nu = \max_{1 \leq i \leq k} | x_i |$ is the length of the longest x's.

For larger alphabets Σ, $\mathcal{M}$ and $\mathcal{W}$ are exponentially bounded in $\mathcal{L} = \mathrm{llw}\big(\overline{S^*}\big)$. Since $\overline{S^*} \subseteq \Sigma \cup \Sigma^2 \cup \cdots \cup \Sigma^{\mathcal{L}}$, it follows that

$$\mathcal{M} \leq \sum_{1 \leq i \leq \mathcal{L}} |\Sigma|^i = \frac{|\Sigma|^{\mathcal{L}+1} - |\Sigma|}{|\Sigma| - 1} = O(|\Sigma|^{\mathcal{L}}), \tag{3.41}$$

$$\mathcal{W} \leq \sum_{1 \leq i \leq \mathcal{L}} i\,|\Sigma|^i = \frac{\mathcal{L}\,|\Sigma|^{\mathcal{L}+2} - (\mathcal{L}+1)\,|\Sigma|^{\mathcal{L}+1} + |\Sigma|}{(|\Sigma| - 1)^2} = O(\mathcal{L}\,|\Sigma|^{\mathcal{L}}), \tag{3.42}$$

both of which can be achieved. Since $\mathcal{L} = O(|\Sigma|^\nu)$, where $\nu = \max_{1 \leq i \leq k} | x_i |$, then

$$\mathcal{M} = |\Sigma|^{O(|\Sigma|^\nu)}, \qquad \mathcal{W} = |\Sigma|^{O(|\Sigma|^\nu)}, \tag{3.43}$$

both of which are doubly-exponential in ν. More precisely, the following corollary is a direct result of Corollary 2.3.2.

Corollary 3.3.2. *[83, 84] Let $S = \{ x_1, x_2, \ldots, x_k \}$. Suppose $| x_i | \leq \nu$ for all i, and S^* is co-finite. Then the number of words not in S^* is*

$$\mathcal{M} \leq \frac{|\Sigma|^q - 1}{|\Sigma| - 1}, \tag{3.44}$$

where $q = \frac{2}{2|\Sigma|-1}(2^\nu |\Sigma|^\nu - 1)$.

I will show in the next chapter that there are examples in which both the number $\mathcal{M}$ of words not in a generated co-finite language, and the total number $\mathcal{W}$ of symbols of such words, can be doubly-exponential in ν, the length of the longest words in the given basis.

Number of words and symbols in the basis

Instead of considering the generated language, we can also consider the basis.

Problem 3.3.3 (*the FPFM, the basis*)**.** *Given a set S of non-empty words of lengths $c_1, c_2, \ldots, c_k$ over the alphabet Σ such that there are only finitely many words that cannot be written as concatenations of words in S, then what is the size of S?*

As usual, let $\nu = \max \{ c_1, c_2, \ldots, c_k \}$ be the longest length.

Over the unary alphabet, then $| S | \leq \nu$ and the problem becomes less interesting. Over the unary alphabet, either $k = 1$ and $| S | \geq 1$ or $k \geq 2$ and $| S | \geq 2$ holds, and the bounds are tight since there are two integers c_1, c_2 such that $\gcd(c_1, c_2) = 1$.

Over a larger alphabet, $| S |$ can be exponential in ν. Over larger alphabets, obviously there is an upper bound on the size of S, even when S^* is not co-finite, given by

$$| S | \leq |\Sigma|^{c_1} + |\Sigma|^{c_2} + \cdots + |\Sigma|^{c_k} = O(|\Sigma|^n), \qquad \sum_{w \in S} | w | = O(n |\Sigma|^n), \tag{3.45}$$

both of which are tight. When $k = 1$, by Proposition 2.4.1, we have $c_1 = 1$ and the equation

$$| S | = \sum_{w \in S} | w | = |\Sigma| \tag{3.46}$$

holds. When $k = 2$, the problem is discussed in Chapter 2 in the 2FPFM, the Frobenius problem in a free monoid with bases composed of words of two distinct lengths. As we saw in the 2FPFM, if S is a set of words of lengths c_1, c_2 and S^* is co-finite, then by Theorem 2.6.23,

$$| S | > |\Sigma|^{c_1} + \frac{1}{c_2} |\Sigma|^{c_2}, \qquad \sum_{w \in S} | w | > c_1 |\Sigma|^{c_1} + |\Sigma|^{c_2}, \tag{3.47}$$

where we suppose $c_1 < c_2$. In other words, when the basis consists of words of two lengths, an exponentially large basis is required to generate a co-finite language.

For larger $k \geq 3$, is it necessary for a basis to be exponential in ν in order to generate a co-finite language? The answer is "no". There are examples such that the size of each basis is linear in ν and each basis still generates a co-finite language.

Kao gave in 2006 the following example, which shows the existence of a linear-size basis. Over the binary alphabet $\Sigma = \{ \mathtt{0}, \mathtt{1} \}$, let

$$U_k = \left\{ \mathtt{1}, \mathtt{00}, \mathtt{01}, \mathtt{000}, \mathtt{010}, \mathtt{0010}, \mathtt{0110}, \ldots, \mathtt{001}^{k-3}\mathtt{0}, \mathtt{01}^{k-2}\mathtt{0} \right\}, \tag{3.48}$$

and $V_k = U_k \cup \left\{ \mathtt{1}^{k-2}\mathtt{0}, \mathtt{1}^{k-1}\mathtt{0} \right\}$ for $k \geq 3$. Then V_k^* is co-finite, and one can verify by induction that

$$\Sigma^* \setminus V_k^* = \left\{ \mathtt{0}, \mathtt{10}, \mathtt{110}, \ldots, \mathtt{1}^{k-3}\mathtt{0} \right\}. \tag{3.49}$$

The basis V_k is minimal in the sense that any proper subset of V_k cannot generate a co-finite language in $\{ \mathtt{0}, \mathtt{1} \}^*$. The size of V_k is $2k+1$, which is linear in $\nu = \operatorname{llw}(V_k) = k$. In addition, the total number of symbols in V_k, the alphabetic length of a shortest regular expression for V_k, and the (nondeterministic) state complexity of V_k are all polynomial in ν.

I will show in the next section an even better example of a basis generating a co-finite language, where the size of the basis is constant in ν and thus the total number of symbols of all words in the basis is ν plus a constant.

3.3.2 Coverage of words as solutions

Given any word w over alphabet Σ, is there a basis S such that w is one of the longest words not in the generated language S^*, or is there a basis S such that w is the unique longest word not in the generated language S^*? For the integer FP, we saw in Theorem 1.2.16 that any positive integer n is a Frobenius number $n = g(x_1, x_2, x_3)$ for some positive integers x_1, x_2, x_3. So over the unary alphabet, any word can be the unique longest word not in a language generated by three words.

Over larger alphabets when $|\Sigma| \geq 2$, we first limit the number of words in the basis S by $|\, S \,| \leq k$ for some constant k. Then at least for every positive odd integer l, there is a basis S of size k such that S^* is co-finite and $l = \operatorname{llw}\left(\overline{S^*}\right)$.

Theorem 3.3.4. *Let Σ be an alphabet. There is an integer k such that every positive odd integer l is the length of the longest words not in a generated co-finite language S^* in Σ^*, where the basis S is of cardinality k.*

Proof. Let $k = |\Sigma|^2 + 2|\Sigma| - 1$. We define

$$S_l = \big(\Sigma \setminus \{ \mathtt{0} \}\big) \cup \Sigma^2 \cup \mathtt{0}\big(\Sigma \setminus \{ \mathtt{0} \}\big)\mathtt{0} \cup \left\{ \mathtt{0}^{l+2} \right\} \tag{3.50}$$

for odd positive integers l. Then S_l is of cardinality k. Since $\Sigma^2 \subseteq S_l$, any word of even length is in S_l^*. Consider a word $w = a_1 a_2 \cdots a_n$ of odd length. If $a_i \neq \mathtt{0}$ for some odd integer i, then

$$w = (a_1 a_2) \cdots (a_{i-2} a_{i-1}) a_i (a_{i+1} a_{i+2}) \cdots (a_{n-1} a_n) \in S_l^*, \tag{3.51}$$

since $\Sigma \setminus \{ \mathtt{0} \} \subseteq S_l$. Now we assume $a_i = \mathtt{0}$ for all odd integers $1 \leq i \leq n$. If $a_i \neq \mathtt{0}$ for some even integer i, then

$$w = (a_1 a_2) \cdots (a_{i-3} a_{i-2})(a_{i-1} a_i a_{i+1})(a_{i+2} a_{i+3}) \cdots (a_{n-1} a_n) \in S_l^*, \tag{3.52}$$

since $\mathtt{0}\big(\Sigma \setminus \{ \mathtt{0} \}\big)\mathtt{0} \subseteq S_l$. So all words not in S_l^* are powers of $\mathtt{0}$. Since

$$\mathbb{N} \setminus \langle 2, l+2 \rangle = \{ 1, 3, 5, \ldots, l \}, \tag{3.53}$$

then S_l^* is co-finite and $S_l^* = \Sigma^* \setminus \left\{ \mathtt{0}, \mathtt{0}^3, \mathtt{0}^5, \ldots, \mathtt{0}^l \right\}$, where $\mathrm{llw}\left(\overline{S_l^*}\right) = l$. □

Theorem 3.3.4 shows that using a fixed number of words, one can generate a co-finite language such that the longest word can be arbitrarily long. Hence the measure $\mathcal{L} = \mathrm{llw}\left(\overline{S^*}\right)$ is unbounded in the measure $|\, S \,|$. Theorem 3.3.4 also shows that, for a fixed alphabet, the size of a basis that generates a co-finite language can be a constant in $\nu = \max_{w \in S} |\, w \,|$, and thus the following bounds are tight for a fixed alphabet

$$|\, S \,| = O(|\Sigma|^\nu), \qquad |\, S \,| = \Omega(1). \tag{3.54}$$

The constructed S_l is composed of words of only four lengths and the total number of symbols in S_l is $\nu + O(1)$. That shows the exponential lower bounds for the 2FPFM, namely $|\, S \,| > |\Sigma|^m + |\Sigma|^n / n$ and $\sum_{w \in S} |\, w \,| > m\,|\Sigma|^m + |\Sigma|^n$, where m, n are the two lengths of words in the 2FPFM, cannot be generalized in the general FPFM.

Now, we limit the number of different lengths of words in the basis. The following proposition is a direct consequence of the integer FP.

Proposition 3.3.5. *For any word w over the alphabet Σ,*

(a) there is a basis S of words of three lengths such that S^* is co-finite and w is one of the longest words not in S^*;

(b) there is a basis S' of words of four lengths such that S^* is co-finite and w is the unique longest word not in S^*.

Proof. Let $l = |\, w \,|$. By Theorem 1.2.16, there are three integers c_1, c_2, c_3 such that $g(c_1, c_2, c_3) = l$. Let

$$S = \Sigma^{c_1} \cup \Sigma^{c_2} \cup \Sigma^{c_3}. \tag{3.55}$$

Then S^* is co-finite, and w is one of the longest words not in S^*, where S is composed of words of three lengths c_1, c_2, c_3. Furthermore, let

$$S' = \Sigma^{c_1} \cup \Sigma^{c_2} \cup \Sigma^{c_3} \cup \Sigma^l \setminus \{ w \}. \tag{3.56}$$

Then S'^* is co-finite, and w is the unique longest word not in S'^*, where S' is composed of words of four lengths c_1, c_2, c_3, l. □

Now we consider another type of problem: what fraction of bases generate co-finite languages? First, we consider sets consisting of words of lengths m and n, $0 < m < n < 2m$. By Lemma 2.6.21, we have $|\,\Sigma^n \setminus S\,| \geq 1/2(|\Sigma|^n - |\Sigma|^m)$ and there are bases that achieve equality, which will be presented in Chapter 4. Suppose S is one such basis. Then any set T such that $S \subseteq T \subseteq \Sigma^m \cup \Sigma^n$ also generates a co-finite language. There are in total $2^{\frac{1}{2}(|\Sigma|^n+|\Sigma|^m)}$ such T, while the number of sets consisting of words of lengths m and n is $2^{|\Sigma|^n+|\Sigma|^m}$. Now, we consider sets consisting of words of lengths $1, 2, \ldots, l$ for an odd integer l. By Theorem 3.3.4, the set $S_{l-2} = \big(\Sigma \setminus \{\,\mathtt{0}\,\}\big) \cup \Sigma^2 \cup \mathtt{0}\big(\Sigma \setminus \{\,\mathtt{0}\,\}\big)\mathtt{0} \cup \{\,\mathtt{0}^l\,\}$ generates a co-finite language. Then any set T such that $S_{l-2} \subseteq T \subseteq \bigcup_{i=1}^{l} \Sigma^i$ generates a co-finite language. Then at least

$$\frac{1}{2^{|\Sigma|^2+2|\Sigma|-1}} \tag{3.57}$$

of all subsets of $\bigcup_{i=1}^{l} \Sigma^i$ generate co-finite languages.

Now we consider extending a basis S by adding words into S without changing $\mathrm{llw}\big(\overline{S^*}\big)$. For the unary alphabet, this problem becomes an integer problem and is already studied (see §1.2.3). Now we discuss it over a larger alphabet.

Theorem 3.3.6. *For any alphabet Σ, $|\Sigma| \geq 2$, and any integer $p \in \mathbb{N}$, there are two sets S, T of words such that $|\,T\,| \geq p$, for any $w \in S \cup T$, $w \notin (S \cup T \setminus \{\,w\,\})^*$, and $\mathrm{llw}\left(\overline{(S \cup T)^*}\right) = \mathrm{llw}\left(\overline{S^*}\right)$.*

Proof. Assume $|\Sigma|^q > p$. Now we consider the 2FPFM with lengths $m = q+1, n = 2q+1$. In the word graph $\Gamma(m,n) = (\Sigma^{n-m}, \Sigma^n, \psi)$, there are $|\Sigma|^m - |\Sigma|$ arcs that join the vertex $\mathtt{0}^{n-m}$ to other vertices, namely $V = \mathtt{0}^{n-m}\Sigma(\Sigma^{n-m} \setminus \mathtt{0}^{n-m})$. Let $S = \Sigma^m \cup \Sigma^n \setminus V, S' = \Sigma^m \cup \Sigma^n \setminus \{\,\mathtt{0}^{n-1}\mathtt{1}\,\}$. Then the longest paths in both of the word graph $G_S^{(m,n)}, G_{S'}^{(m,n)}$ are of length 1, which implies $\mathrm{llw}\left(\overline{S^*}\right) = \mathrm{llw}\left(\overline{S'^*}\right) = g(m, n+m)$. Since $S' \subseteq \Sigma^m \cup \Sigma^n$ and $\gcd(m,n) = 1$, for any $w \in S'$, we have $w \notin (S' \setminus \{\,w\,\})^*$. Then the two sets S and $S' \setminus S$ satisfy the required conditions. □

3.3.3 The number of different factorizations

Let S be a set of words over an alphabet Σ such that S^* is co-finite. If a word w is in S^*, then by definition, w can be written as a concatenation of words in S, and this factorization may not necessarily be unique. Denote by $\mathcal{D}(w)$ the number of different factorizations of w over the given basis S. We say two factorizations of a word $w = x_1x_2\cdots x_p$ and $w = y_1y_2\cdots y_q$ are different if $p \neq q$ or $p = q$ and there exists i, $1 \leq i \leq p$, such that $x_i \neq y_i$.

Problem 3.3.7 (*Factorization of words*)**.** *Given k non-empty words $x_1, \ldots, x_k$ over Σ, and a word w that can be written as a concatenation of words in $\{\,x_1, \ldots, x_k\,\}$, then in how many different ways can w be written in that form?*

In the integer analogue, let $x_1, x_2, \ldots, x_k$ be k positive integers. The denumerant $d(n; x_1, x_2, \ldots, x_k)$ is the number of different ways that n can be written as a non-negative integer linear combination of $x_1, x_2, \ldots, x_k$ given by

$$n = c_1 x_1 + c_2 x_2 + \cdots + c_k x_k. \tag{3.58}$$

As we saw in Eq. (1.67), for fixed basis $x_1, x_2, \ldots, x_k$ and as $n \to \infty$, the asymptotic bound is

$$d(n; x_1, x_2, \ldots, x_k) = \Theta\Big(\frac{n^{k-1}}{x_1 x_2 \cdots x_k (k-1)!}\Big) = \Theta(n^{k-1}), \tag{3.59}$$

which is polynomial in n.

In the setting of a free monoid, even over a unary alphabet, the result on the number of different factorizations is different. This property is one of the few cases where the FPFM differs from the integer FP, *even when over the unary alphabet* $\{ \mathtt{0} \}$. For example, although $d(11; 3, 5) = 1$, as there is a unique partition of 11 in $\langle 3, 5 \rangle$ given by

$$11 = 2 \cdot 3 + 1 \cdot 5, \tag{3.60}$$

there are three distinct factorizations of $\mathtt{0}^{11}$ in $\{ \mathtt{0}^3, \mathtt{0}^5 \}^*$ given by

$$\mathtt{0}^{11} = \mathtt{0}^3\mathtt{0}^3\mathtt{0}^5 = \mathtt{0}^3\mathtt{0}^5\mathtt{0}^3 = \mathtt{0}^5\mathtt{0}^3\mathtt{0}^3. \tag{3.61}$$

The function $d(n; x_1, x_2, \ldots, x_k)$ is strongly related to the Euler partition problem of integers which can be viewed as $d(n; \mathbb{N})$. Similarly, the number $\mathcal{D}(w)$ of different factorizations of w in a given finite basis is obviously less than or equal to the number of different factorizations of w in the infinite basis Σ^*. Hence it follows that

$$\mathcal{D}(w) \leq 2^{|w|-1} = O(2^{|w|}), \tag{3.62}$$

which is tight. Let $S = \bigcup_{1 \leq i \leq l} \Sigma^l$ be the basis. Then for any word of length $\leq l$, the equality in the bound in (3.62) can be attained as $\mathcal{D}(w) = 2^{|w|-1}$.

There is a trivial lower bound

$$\mathcal{D}(w) \geq 1 \tag{3.63}$$

(or $\mathcal{D}(w) \geq 0$ if we do not ask w to be in the generated language S^*), and it is tight. Let w be one of the shortest words in the basis S. Then the factorization of w is unique and thus $\mathcal{D}(w) = 1$. In fact, in the trivial case $S = \Sigma$, each factorization is unique due to the freeness of the monoid, and thus $\mathcal{D}(w) = 1$ for all w.

In particular, in the 2FPFM with lengths being m, n, $m < n$, then the factorization, if any, of each word of length im for $0 < i < n$ or in for $0 < i < m$ must be unique, since im, for $0 < i < n$, has a unique expression as a linear combination in $\langle m, n \rangle$ given by $im = i \cdot m$, and any word of length im has a unique factorization, where each factor is of length m. The case of words of length in for $0 < i < m$ is similar. There is another situation where the factorization is unique, as given by the following proposition.

Proposition 3.3.8. *Let S be a set of words over Σ of lengths m and n, $m < n$, such that S^* is co-finite. If $w \in S^*$ with $|\, w \,| \equiv n \pmod{m}$, but either $w[1..\,|\, w \,| - m] \notin S^*$ or $w[m+1..\,|\, w \,|] \notin S^*$, then w has a unique factorization in S.*

Proof. Without loss of generality, we prove the prefix result. The suffix result is similar. Suppose $w \in S^*$ and $w[1..\,|\, w \,| - m] \notin S^*$.

Since S^* is co-finite, $\gcd(m, n) = 1$. Let $|\, w \,| = n + jm$. Then $j \geq 0$. If $j = 0$, then w has a unique factorization given by $w = w$. Now we assume $j > 0$. The prefix $w[1..n + (j-1)m]$ is not in S^*, so by Proposition 2.4.10 none of the words

$$w[im + 1..im + n] \tag{3.64}$$

is in S, for $0 \leq i \leq j - 1$. Since w is in S^*, there is a factorization of w given by

$$w = w_1 w_2 \cdots w_k. \tag{3.65}$$

Since $m \nmid |\, w \,| = n + jm$, at least one of the factors is of length n. Consider the first factor, say w_l, that is of length n. Then all the w_i, $i < l$, are of length m. Comparing lengths leads to $w_l = w[(l-1)m + 1..(l-1)m + n]$. Hence $l - 1 \geq j$, and thus $|\, w_1 w_2 \cdots w_l \,| = (l-1)m + n \geq jm + n$. But we know $|\, w \,| = n + jm$. So $k = l = j + 1$, and all factors are of length m except the last factor, which is of length n. Therefore, w has a unique factorization in S. □

3.4 General form of the Frobenius problem of words

Given $x_1, x_2, \ldots, x_k$ such that $\{\, x_1, x_2, \ldots, x_k \,\}^*$ is co-finite, the Frobenius problem in a free monoid is to find the longest word(s) not in the generated language.

One way to look at the generalized Frobenius problem is that by defining the appropriate concatenation of words, such that the concatenation-closure is well-defined, then the concatenation-closure of input words covers all words in some set, for example Σ^*, over an alphabet Σ, except a "small" number of words, for example, a finite number. When the concatenation is the normal concatenation, the set to be covered is Σ^*, and that "small" number is finite, it becomes the FPFM. Here we will see some other variations on the problem.

3.4.1 Infinite words

First, we will discuss some types of infinite words. The study on words of infinite length is essential in symbolic dynamics [12, 113] and logic with temporal properties [24, 114, 109] (refer to the book [122] for a comprehensive reading). Since each word is of infinite length, the length of the longest words is not interesting, and thus we will instead focus on the co-finiteness property.

An *infinite word* w is an infinite sequence of letters over an alphabet Σ

$$w = a_1 a_2 a_3 \cdots \tag{3.66}$$

and the set of all infinite words is denoted by Σ^ω. To avoid ambiguity, the words we have studied thus far are called finite words. The set of all finite words and infinite words over Σ is denoted by Σ^∞. Let w_1, w_2 be two infinite words and u be a finite word. Then we define $w_1 \cdot w_2 = w_1$, $w_1 \cdot u = w_1$, and $u \cdot w_1 = uw_1$.

An infinite word w is *periodic* if $w = u^\omega$ for some $u \in \Sigma^*$ and is *ultimately periodic* if $w = vu^\omega$ for some $u, v \in \Sigma^*$.

The set of all words that can be written as concatenations of words in S is, as usual, denoted by

$$S^* = \big\{\, x_1 x_2 \cdots x_l : x_1, x_2, \ldots, x_l \in S, l \geq 0 \,\big\}. \tag{3.67}$$

Analogously, we define the set of infinite concatenation of words in S as

$$S^\omega = \big\{\, x_1 x_2 \cdots : x_1, x_2, \ldots \in S \,\big\}, \tag{3.68}$$

and we write $S^\infty = S^* \cup S^\omega$.

The co-finiteness of infinite words over a unary alphabet is not interesting, since there is only one such word. In this section, we always assume the alphabet contains at least two letters without further explanation.

Proposition 3.4.1. *Let S be a set of words in Σ^∞ and $T = S \cap \Sigma^*$. Then S^∞ is co-finite in Σ^∞ if and only if T^* is co-finite in Σ^*. Furthermore, when T^* is co-finite in Σ^*, then T^∞ is co-finite in Σ^∞.*

Proof. $\Rightarrow$: If S^∞ is co-finite, then $S^\infty \cap \Sigma^*$ is co-finite in Σ^*. Since any infinite concatenation of nonempty words gives an infinite word, and any concatenation involving an infinite word gives an infinite word,

$$S^\infty \cap \Sigma^* = S^* \cap \Sigma^* = T^* \cap \Sigma^* \tag{3.69}$$

is co-finite in Σ^*. Hence T^* is co-finite in Σ^*.

Assume T^* is co-finite in Σ^*. We now prove T^∞ is co-finite in Σ^∞. Let w be an infinite word, and $l = \mathrm{llw}\,(\Sigma^* \setminus T^*)$. Write $w = u_1 w_1$, where $|\, u_1 \,| = l + 1$. Then $u_1 \in T^*$ and w_1 is an infinite word. Repeat this procedure for w_1, and so forth. Then $w = u_1 u_2 \cdots$ is in T^ω, where all the $u_i, i \geq 1$ are in T^*. Hence $\Sigma^\omega = T^\omega$. In addition, T^* is co-finite in Σ^*, so T^∞ is co-finite in $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$.

$\Leftarrow$: If T^* is co-finite in Σ^*, then we already proved T^∞ is co-finite in Σ^∞. Hence $S^\infty \supseteq T^\infty$ is also co-finite in Σ^∞. □

Therefore, any set S of words that generates a co-finite language in Σ^∞ includes as a subset a set of finite words that generates a co-finite language in Σ^*, and any set S' of finite words that generates a co-finite language in Σ^* itself generates a co-finite language in Σ^∞. In this sense, the behavior in Σ^∞ and the behavior in Σ^* are equivalent. Furthermore, if S generates a co-finite language in Σ^∞, then any word given by a concatenation of words in S that includes some infinite word must be an infinite word, and thus can be written as concatenation of words in $S \cap \Sigma^*$. So in fact $S^\infty = (S \cap \Sigma^*)^\infty$.

Right-infinite words

Now we consider Σ^ω. To avoid confusion with another concept, the left-infinite word, the type of infinite word discussed so far is also called a *right-infinite word*. Let S be a set of words in Σ^ω. Then by definition, $S^\omega = S$. Hence, any set of words in Σ^ω that generates a co-finite language in Σ^ω must itself be co-finite. So in order to generate a co-finite language in Σ^ω non-trivially, the basis must contain words in Σ^*. The following corollary follows directly from the proof of Proposition 3.4.1.

Corollary 3.4.2. *Let S be a set of words in Σ^*. If S^* is co-finite in Σ^*, then $S^\omega = \Sigma^\omega$.*

Shallit observed the property in the previous corollary and he gave an equivalent condition in Proposition 3.4.3 to check whether $S^\omega = \Sigma^\omega$ for a set S of words in Σ^*. Proposition 3.4.3 can be used to disprove the co-finiteness of S^* for a particular basis S of words in Σ^* in the FPFM. The converse of Corollary 3.4.2 is not true. For example, over the binary alphabet, let

$$S = \{\, \mathtt{00}, \mathtt{01}, \mathtt{1} \,\}. \tag{3.70}$$

Proposition 3.4.3 implies that $S^\omega = \Sigma^\omega$, but S^* is not co-finite in Σ^*.

Proposition 3.4.3. *Let S be a set of words in Σ^*. Then $S^\omega = \Sigma^\omega$ if and only if for any word w in Σ^ω there is a finite nonempty prefix of w that is in S.*

Proof. $\Rightarrow$: Suppose $S^\omega = \Sigma^\omega$. Then any word w in Σ^ω can be factorized into nonempty words in S, and the first factor is a nonempty finite prefix of w that is in S.

$\Leftarrow$: Let w be a word in Σ^ω. Then $w = u_1 w_1$, where u_1 is a nonempty word in S. Since w_1 is again in Σ^ω, $w_1 = u_2 w_2$, where u_2 is in S, and so on. Hence $w = u_1 u_2 \cdots$ can be factorized into words in S. So $S^\omega = \Sigma^\omega$. □

As we saw in Proposition 2.1.3, if a set S of words in Σ^* generates a co-finite language in Σ^*, then there is a finite subset of S that generates the same co-finite language in Σ^*. But for co-finite languages in Σ^ω, this is not true. Consider

$$S = \mathtt{0}^*\mathtt{1} + \mathtt{0}^\omega = \{\, \mathtt{1}, \mathtt{01}, \mathtt{001}, \mathtt{0001}, \ldots, \mathtt{00}\cdots \,\}. \tag{3.71}$$

Then S^ω contains all words in Σ^ω for $\Sigma = \{\, \mathtt{0}, \mathtt{1} \,\}$. For any word w in Σ^ω, if $w = u\mathtt{0}^\omega$, where u is a finite word that does not end with $\mathtt{0}$, then $w = u_1u_2\cdots u_k\mathtt{0}^\omega v^\omega$ is in S^ω, where $u = u_1u_2\cdots u_k$ is the factorization of u into $\{\, \mathtt{1}, \mathtt{01}, \mathtt{001}, \mathtt{0001}, \ldots \}$ and v is an arbitrary word in S. Otherwise, w can be factorized by the positions after each letter $\mathtt{1}$ and thus w is in S^ω. Now we consider a subset $T \subsetneq S$. If T does not contain $\mathtt{0}^i\mathtt{1}$ for some $i \geq 0$, then none of the words $\mathtt{0}^i\mathtt{1}\Sigma^\omega$ is in S^ω. If T does not contain $\mathtt{0}^\omega$, then none of the words $\Sigma^*\mathtt{0}^\omega$ is in S^ω. Hence S generates a co-finite language in Σ^ω, while no proper subset of S generates a co-finite language in Σ^ω.

Is there a basis S generating a co-finite language in Σ^ω such that $S^\omega \neq \Sigma^\omega$? The answer is "yes". Consider the following basis over the binary alphabet

$$S = \mathtt{0}^*\mathtt{1} + \mathtt{0}^*\mathtt{10}^\omega = \{\, \mathtt{1}, \mathtt{01}, \mathtt{001}, \mathtt{0001}, \ldots, \mathtt{100}\cdots, \mathtt{0100}\cdots, \mathtt{00100}\cdots, \ldots \}\,. \tag{3.72}$$

For any word w in Σ^ω that contains at least one $\mathtt{1}$, w can be factorized by the positions after each letter $\mathtt{1}$ (except the last), and thus w is in S^ω. The only word that cannot be so factorized is $\mathtt{0}^\omega$, and thus $S^\omega = \Sigma^\omega \setminus \{\, \mathtt{0}^\omega \,\}$ is non-trivially co-finite ($\neq \Sigma^\omega$).

Even when the basis contains only finitely many words in Σ^ω, the answer is still "yes". For example, let

$$T = \{\, \mathtt{1}, \mathtt{00}, \mathtt{011}, \mathtt{0100}, \mathtt{01011}, \mathtt{010100}, \ldots \}\,, \quad \text{and} \quad S = T \,\cup\, T\mathtt{0}, \tag{3.73}$$

where T contains all prefixes of $(\mathtt{01})^\omega$ with the last letter altered. Clearly, S contains no word in Σ^ω. We claim that S^ω is co-finite in Σ^ω and $S^\omega \neq \Sigma^\omega$. Let w be any word in Σ^ω except $(\mathtt{01})^\omega$. Then w has a prefix u_1 in T and can therefore be written as $w = u_1w_1$. Now there are two cases for w_1. If w_1 is not $(\mathtt{01})^\omega$, then we can perform the factorization again on w_1, and so forth. Otherwise, w_1 is $(\mathtt{01})^\omega$, then write $w = (u_1\mathtt{0})(\mathtt{10})^\omega$, which is a factorization of w into words in S. Hence S^ω is co-finite in Σ^ω, and $S^\omega = \Sigma^\omega \setminus \{\, (\mathtt{01})^\omega \,\}$.

When the basis contains only finitely many words in Σ^*, the answer is trivially "yes". Let S be any co-finite proper subset of Σ^ω. Then S contains no word in Σ^* and $S^\omega = S \neq \Sigma^\omega$ is non-trivially co-finite in Σ^ω.

When a basis is finite, however, the answer is "no" as in the following proposition.

Proposition 3.4.4. *Let S be a finite set of words in Σ^∞. Then S^ω is co-finite in Σ^ω if and only if $S^\omega = \Sigma^\omega$.*

Proof. $\Leftarrow$: It is straightforward. $\Rightarrow$: Suppose there is a word w in Σ^ω such that no finite nonempty prefix is in S. Since $S \,\cap\, \Sigma^*$ is finite, let u be a prefix of w that is longer than any word in S. Now consider the language $u\Sigma^\omega$. We claim that if w in $u\Sigma^\omega$ can be factorized into words in S then w is in S. To see this, consider the first factor in the factorization of u in S. It must be infinite, since any finite word in S is of length $< |\, u \,|$ and none of the prefixes of u is in S. But $S \,\cap\, \Sigma^\omega$

is finite, so there are infinitely many words in $u\Sigma^\omega$ that are not in S^ω, and thus S^ω cannot be co-finite. Therefore, every word w in Σ^ω has a finite prefix in S. By Proposition 3.4.3, we have $S^\omega = \Sigma^\omega$. □

By the preceding proof, we also know that for a finite set S of words, if $S^\omega \neq \Sigma^\omega$, then $\Sigma^\omega \setminus S^\omega$ is of uncountable cardinality.

Left-infinite words

For each finite word $w = a_1a_2\cdots a_l$, the reverse of w is $w^R = a_la_{l-1}\cdots a_1$. Similarly, a *left-infinite word* w^R is the reverse of an infinite word $w = a_1a_2a_3\cdots$ as

$$w^R = \cdots a_3a_2a_1 \tag{3.74}$$

and the set of all left-infinite words is denoted by ${}^\omega\Sigma$. The set of all finite words and left-infinite words is denoted by ${}^\infty\Sigma = \Sigma^* \cup {}^\omega\Sigma$. The concatenation of two finite words is defined as usual. Let w_1, w_2 be two left-infinite words and u be a finite word. Then define $w_1 \cdot w_2 = w_2, w_1 \cdot u = w_1u, u \cdot w_1 = w_1$. A left-infinite word w is *periodic* if $w = {}^\omega u$ for some $u \in \Sigma^*$ and is *ultimately periodic* if $w = {}^\omega uv$ for some $u, v \in \Sigma^*$. Analogously, we define the set of left-infinite concatenations of words in S as follows:

$${}^\omega S = \{\, \cdots x_2x_1 : x_1, x_2, \ldots \in S \,\}, \tag{3.75}$$

and we write ${}^\infty S = S^* \cup {}^\omega S$. Then by the previous discussion on right-infinite words, the following propositions hold analogously.

Proposition 3.4.5. *Let S be a set of words in ${}^\infty\Sigma$ and $T = S \cap \Sigma^*$. Then ${}^\infty S$ is co-finite in ${}^\infty\Sigma$ if and only if T^* is co-finite in Σ^*. Furthermore, when T^* is co-finite in Σ^*, ${}^\infty T$ is co-finite in ${}^\infty\Sigma$, and ${}^\omega T = {}^\omega\Sigma$.*

Proposition 3.4.6. *Let T be a set of words in Σ^*. Then ${}^\omega T = {}^\omega\Sigma$ if and only if for any word w in ${}^\omega\Sigma$ there is a nonempty finite suffix of w that is in T.*

Applying the reversal operator to each word in a basis in the examples for right-infinite words, we obtain examples for left-infinite words. Let

$$S = \mathtt{10}^* + {}^\omega\mathtt{0} = \{\, \mathtt{1}, \mathtt{10}, \mathtt{100}, \mathtt{1000}, \ldots, \ldots \mathtt{00} \,\}. \tag{3.76}$$

Then S generates a co-finite language in ${}^\omega\Sigma$, but no proper subset of S generates a co-finite language in ${}^\omega\Sigma$. So Proposition 2.1.3 does not hold for languages of left-infinite words. Let

$$T = \{\, \mathtt{1}, \mathtt{00}, \mathtt{110}, \mathtt{0010}, \mathtt{11010}, \mathtt{001010}, \ldots \,\}, \tag{3.77}$$

which contains all suffixes of ${}^\omega(\mathtt{10})$ with the first letter altered. Then $S = T \cup \mathtt{0}T$ generates a co-finite language in ${}^\omega\Sigma$ with ${}^\omega S = {}^\omega\Sigma \setminus \{\, {}^\omega(\mathtt{10}) \,\} \neq {}^\omega\Sigma$.

Proposition 3.4.7. *Let S be a finite set of words in ${}^\infty\Sigma$. If ${}^\omega S$ is co-finite in ${}^\omega\Sigma$, then ${}^\omega S = {}^\omega\Sigma$.*

Bi-infinite words

Another concept of words of infinite length is the bi-infinite word. A *bi-infinite word* w is the concatenation of a left-infinite word u with a right-infinite word v, for example,

$$w = u \cdot v = \cdots b_3 b_2 b_1 a_1 a_2 a_3 \cdots . \tag{3.78}$$

For an arbitrary left-infinite word u, an arbitrary finite word x, and an arbitrary right-infinite word v, we let $(ux)v = u(xv)$. In other word, for a bi-infinite word w, the factorization $w = u \cdot v$ is not necessarily unique. This kind of bi-infinite word is called an "unpointed" bi-infinite word [7]. The set of all bi-infinite words is denoted by ${}^{\omega}\Sigma^{\omega}$, and the set of all finite words and bi-infinite words is denoted by ${}^{\infty}\Sigma^{\infty}$. Since the concatenation of 2 bi-infinite words is not well-defined, the concatenation in the discussion on bi-infinite words is only performed on finite words. A bi-infinite word w is *periodic* if $w = {}^{\omega}uu^{\omega}$ for some $u \in \Sigma^*$ and is *ultimately periodic* if $w = {}^{\omega}uxv^{\omega}$ for some $u, x, v \in \Sigma^*$. We define the set of bi-infinite concatenations of words in a set S of finite words as

$${}^{\omega}S^{\omega} = \left\{ \cdots x_2 x_1 y_1 y_2 \cdots : x_1, x_2, \ldots, y_1, y_2, \ldots \in S \right\}, \tag{3.79}$$

and write ${}^{\infty}S^{\infty} = S^* \cup {}^{\omega}S^{\omega}$. Analogously, a basis that generates a co-finite language in Σ^* also generates a co-finite language in ${}^{\infty}\Sigma^{\infty}$ and ${}^{\omega}\Sigma^{\omega}$, respectively.

Proposition 3.4.8. *Let S be a set of words in Σ^*. If S^* is co-finite in Σ^*, then ${}^{\infty}S^{\infty}$ is co-finite in ${}^{\infty}\Sigma^{\infty}$, and ${}^{\omega}S^{\omega} = {}^{\omega}\Sigma^{\omega}$.*

Proof. Since S^* is co-finite in Σ^*, let $l = \operatorname{llw}(\Sigma^* \setminus S^*)$. Then $\Sigma^{l+1} \subseteq S^*$. Any bi-infinite word can be factorized into Σ^{l+1}, by grouping $l+1$ consecutive letters together, so ${}^{\omega}\Sigma^{\omega} \subseteq {}^{\omega}(\Sigma^{l+1})^{\omega} \subseteq {}^{\omega}(S^*)^{\omega} \subseteq {}^{\omega}S^{\omega}$. Hence ${}^{\omega}\Sigma^{\omega} = {}^{\omega}S^{\omega}$. Then the language ${}^{\infty}\Sigma^{\infty} \setminus {}^{\infty}S^{\infty} = \Sigma^* \setminus S^*$ is finite, and thus ${}^{\infty}S^{\infty}$ is co-finite in ${}^{\infty}\Sigma^{\infty}$. □

The co-finiteness of generated sets of bi-infinite words is related to the co-finiteness of generated sets of right-infinite words and left-infinite words. First we need a technical lemma.

Lemma 3.4.9. *Let S be a set of words in Σ^*.*

(a) If S^{ω} is co-finite in Σ^{ω}, and w is a periodic word not in S^{ω}, then there is a suffix u of w such that $u \in S^{\omega}$.

(b) If ${}^{\omega}S$ is co-finite in ${}^{\omega}\Sigma$, and w is a left-periodic word not in ${}^{\omega}\Sigma$, then there is a prefix u of w such that $u \in {}^{\omega}S$.

Proof. We prove the result (a) for right-infinite words; the result (b) on left-infinite words is similar.

Suppose S^{ω} is co-finite in Σ^{ω}, and w is a periodic word not in S^{ω}. Consider the language $L = \Sigma^* w$. Since S^{ω} is co-finite in Σ^{ω}, we see that $L \cap S^{\omega} \neq \emptyset$. Let v be

a word in $L \cap S^\omega$ and $v = u_1 u_2 \cdots$ be the factorization of v into the elements of S. Then v is ultimately periodic and all of the u_i that are fully in the periodic part of v give a suffix of v which is in S^ω. This suffix of v is also a suffix of w. □

Proposition 3.4.10. *Let S be a set of words in Σ^*. If S^ω is co-finite in Σ^ω and ${}^\omega S$ is co-finite in ${}^\omega\Sigma$, then ${}^\omega S^\omega$ is co-finite in ${}^\omega\Sigma^\omega$.*

Proof. Let w be a word in ${}^\omega\Sigma^\omega$ that is not ultimately periodic in any direction. Then w can be written as $w = uv$ for some u in ${}^\omega\Sigma$ and v in Σ^ω. Since w is not ultimately periodic in any direction, the number of distinct factorizations $w = uv$ is infinite. Since there are only finitely many words in ${}^\omega\Sigma \setminus {}^\omega S$ and in $\Sigma^\omega \setminus S^\omega$, there must be some u, v such that $w = uv$ and $u \in {}^\omega S, v \in S^\omega$. So w is in ${}^\omega S^\omega$.

Let w be a word in ${}^\omega\Sigma^\omega$, which is ultimately periodic in some direction (or both directions), but not periodic. Without loss of generality, suppose $w = uv$ and v is periodic. By Lemma 3.4.9, there is a suffix v' of v such that v' is in S^ω. Then w can be written as $w = u'v'$, where the number of factorizations of w with distinct u' is infinite. Since there are only finitely many words in ${}^\omega\Sigma \setminus {}^\omega S$, there must be some u' such that $w = u'v'$ and $u' \in {}^\omega S$. Since $v' \in S^\omega$, w is in ${}^\omega S^\omega$.

Now, assume $w = {}^\omega(w')^\omega$ is a periodic word in ${}^\omega\Sigma^\omega$. Write $w = uv$ for some $u \in {}^\omega\Sigma$ and $v \in \Sigma^\omega$. Then both u, v are ultimately periodic and w is uniquely determined by u and by v. If w is not in ${}^\omega S^\omega$, then either u is not in ${}^\omega S$ or v is not in S^ω. Hence the number of w not in ${}^\omega S^\omega$ is less than or equal to the number of periodic and ultimately periodic words in $({}^\omega\Sigma \setminus {}^\omega S) \cup (\Sigma^\omega \setminus S^\omega)$, which is finite.

Therefore, the words, if any, in ${}^\omega\Sigma^\omega \setminus {}^\omega S^\omega$ must be periodic, and there are only finitely many such words. In other words, ${}^\omega S^\omega$ is co-finite in ${}^\omega\Sigma^\omega$. □

By Proposition 3.4.10, if a set S of finite words generates co-finite languages in both right-infinite words and left-infinite words, then S also generates a co-finite language in bi-infinite words, and those words that cannot be factorized must be periodic. The converse of Proposition 3.4.10 is not true. For example, let

$$S = \{ \mathtt{00}, \mathtt{10}, \mathtt{1} \}. \tag{3.80}$$

Then ${}^\omega S^\omega = {}^\omega\Sigma^\omega$ is co-finite in ${}^\omega\Sigma^\omega$ as $\Sigma^2 \subseteq S^*$, but S^ω is not co-finite in Σ^ω as $\mathtt{01}\Sigma^\omega \subseteq \Sigma^\omega \setminus S^\omega$. Nevertheless, the co-finiteness in both right-infinite words and left-infinite words does not ensure the co-finiteness in finite words. Let

$$S = \{ \mathtt{00}, \mathtt{01}, \mathtt{1}, \mathtt{10} \}. \tag{3.81}$$

Then $S = \{ \mathtt{00}, \mathtt{01}, \mathtt{1} \} \cup \{ \mathtt{00}, \mathtt{10}, \mathtt{1} \}$, where the first term generates a co-finite language in Σ^ω and the second term generates a co-finite language in ${}^\omega\Sigma$. So S generates co-finite languages in all three types of infinite words, but S does not generate a co-finite language in Σ^*, since $\mathtt{0}(\mathtt{00})^* \subseteq \Sigma^* \setminus S^*$.

To summarize, the strength of co-finiteness is as follows. The co-finiteness in finite words leads to the co-finiteness in all types of infinite words. The co-finiteness

in both single direction (right and left) infinite words leads to the co-finiteness in bi-infinite words.

In 2009, Shallit [154] showed that a finite set cannot generate a co-finite language in ${}^{\omega}S^{\omega}$ non-trivially. First, we need a proposition.

Proposition 3.4.11. *[154] Let S be a finite set of words in Σ^*. Then ${}^{\omega}S^{\omega} = {}^{\omega}\Sigma^{\omega}$ if and only if for any word w in Σ^* there are words $u, v \in \Sigma^*$ such that $uwv \in S^*$.*

Proof. $\Rightarrow$: Suppose ${}^{\omega}S^{\omega} = {}^{\omega}\Sigma^{\omega}$. Let w be a word in Σ^*. Now we consider the factorization ${}^{\omega}0w0^{\omega} = \cdots x_3x_2x_1y_1y_2y_3\cdots$ into words in S. Then there are u, v such that $uwv = x_j\cdots x_1y_1\cdots y_i \in S^*$ for some indices i, j.

$\Leftarrow$: Let $z = \cdots b_3b_2b_1a_0a_1a_2a_3\cdots$ be a bi-infinite word in ${}^{\omega}\Sigma^{\omega}$, where $a_i, b_i \in \Sigma$. Consider the following sequence of finite words

$$a_0,\ b_1a_0a_1,\ b_2b_1a_0a_1a_2,\ b_3b_2b_1a_0a_1a_2a_3,\ \ldots. \tag{3.82}$$

Then for each $i \geq 0$, there are u_i, v_i such that $w_i = u_ib_i\cdots b_1a_0a_1\cdots a_iv_i \in S^*$. Now we consider the middle factor that covers the letter a_0 in each of the factorizations of words in $W_0 = \left\{ w_i : i \geq 0 \right\}$ into words in S. Since S is finite, by the infinite pigeonhole principle there must be infinitely many words in W_0 such that their factorizations have the same middle term, say x_0. Let W_1 be the set of all such words. Now among those factorizations of words in W_1, we consider the factors to the left and to the right of x_0. Again, since S is finite, by the infinite pigeonhole principle, there must be infinitely many words in W_1 such that their factorizations have the same middle terms, say $x_1x_0y_1$. Let W_2 be the set of all such words. Then consider the factors to the left of x_1 and to the right of y_1. Continuing in this procedure, we construct a factorization for z as $z = \cdots x_2x_1x_0y_1y_2\cdots$. □

Proposition 3.4.12. *[154] Let S be a finite set of words in Σ^*. Then ${}^{\omega}S^{\omega}$ is co-finite in ${}^{\omega}\Sigma^{\omega}$ if and only if ${}^{\omega}S^{\omega} = {}^{\omega}\Sigma^{\omega}$.*

Proof. $\Leftarrow$: It is straightforward. $\Rightarrow$: Suppose ${}^{\omega}S^{\omega} \neq {}^{\omega}\Sigma^{\omega}$. By Proposition 3.4.11, there exists a word w in Σ^* such that no word uwv can be factorized into words in S for all $u, v \in \Sigma^*$. Then none of the words in ${}^{\omega}\Sigma w\Sigma^{\omega}$ can be factorized into words in S, which contradicts the co-finiteness of ${}^{\omega}S^{\omega}$. □

By the preceding proof, we know that for a finite set S of words, if ${}^{\omega}S^{\omega} \neq {}^{\omega}\Sigma^{\omega}$, then ${}^{\omega}\Sigma^{\omega} \setminus {}^{\omega}S^{\omega}$ is of uncountable cardinality.

3.4.2 Concatenation with overlap

In this section, we will see some alternative definitions of concatenation of words. We now return to words of finite length. Some concatenations do not satisfy the associative law and thus may not lead to a monoid. But as far as the concatenation-closure is well-defined, the discussion will be meaningful. To avoid ambiguity, we denote by $u \cdot v$ or uv for short the normal concatenation of words, and by other notation for each alternative.

Repeated deletion

Ito, Kari, Kincaid, and Seki [77] introduced a new binary operation on languages $L, R \subseteq \Sigma^*$ defined by

$$L \natural R = \{ xyz : xy \in L, yz \in R, y \neq \epsilon \}. \tag{3.83}$$

Proposition 3.4.13. *[77] The class of regular languages is closed under the operation of* $\natural$.

Proof. Define a morphism on words $h : (\Sigma \cup \Sigma')^* \to \Sigma^*$ by $h(a) = a$ and $h(a') = a$ for all $a \in \Sigma, a' \in \Sigma'$. Let h^{-1} be the inverse morphism of h. Then

$$L \natural R = h\big((h^{-1}(L) \cap \Sigma^*\Sigma'^+)\Sigma^* \cap \Sigma^*(h^{-1}(R) \cap \Sigma'^+\Sigma^*)\big). \tag{3.84}$$

Since the family of regular languages is closed under the operations of inverse morphism (for example, see Shallit's textbook [156]), intersection, concatenation, and morphism, it is also closed under $\natural$. □

Next we consider a variation on $\natural$ where the overlap can be empty. We define the *concatenation with overlap* of two words by

$$u \flat v = \{ u'w'v' : \exists w' \in \Sigma^* \text{ such that } u = u'w', v = w'v' \} \tag{3.85}$$

and define the *concatenation with overlap* of two languages L, R by

$$L \flat R = \{ w : w \in u \flat v, u \in L, v \in R \}. \tag{3.86}$$

We call $L \natural R$ the *concatenation with non-empty overlap* of L and R, and define $u \natural v = \{ u \} \natural \{ v \}$ for two words u, v. Then the relation between the two new operations is that

$$L \flat R = (L \natural R) \cup (L \cdot R), \tag{3.87}$$

where $L \cdot R$ is the normal concatenation of languages. The next proposition follows immediately.

Proposition 3.4.14. *The family of regular languages is closed under the operation of concatenation with overlap* $\flat$.

The concatenation-with-overlap $\flat$ does not satisfy the associative law. For example, let $L_1 = \{ \mathtt{001} \}, L_2 = \{ \mathtt{12} \}, L_3 = \{ \mathtt{0123} \}$. Then

$$\begin{aligned} (L_1 \flat L_2) \flat L_3 &= \{ \mathtt{0012}, \mathtt{00112} \} \flat \{ \mathtt{0123} \} = \{ \mathtt{00123}, \mathtt{00120123}, \mathtt{001120123} \}, \\ L_1 \flat (L_2 \flat L_3) &= \{ \mathtt{001} \} \flat \{ \mathtt{120123} \} = \{ \mathtt{00120123}, \mathtt{001120123} \}. \end{aligned} \tag{3.88}$$

The concatenation-with-nonempty-overlap $\natural$ also fails to satisfy the associative law. For the same example of L_1, L_2, L_3,

$$(L_1 \natural L_2) \natural L_3 = \{ \mathtt{0012} \} \natural \{ \mathtt{0123} \} = \{ \mathtt{00123} \}, \tag{3.89}$$

$$L_1 \natural (L_2 \natural L_3) = \{\, \mathtt{001} \,\} \natural \emptyset = \emptyset. \tag{3.90}$$

In the following discussion, without further explanation, a sequence of concatenation appearing without parenthesis means the concatenation is done from the left to the right, as follows:

$$L_1 \flat L_2 \flat L_3 \flat \cdots \flat L_k \stackrel{\text{def}}{=} (((L_1 \flat L_2) \flat L_3) \flat \cdots) \flat L_k, \tag{3.91}$$

$$L_1 \natural L_2 \natural L_3 \natural \cdots \natural L_k \stackrel{\text{def}}{=} (((L_1 \natural L_2) \natural L_3) \natural \cdots) \natural L_k. \tag{3.92}$$

Both concatenations with overlap $\flat$ and with non-empty overlap $\natural$ satisfy the associative law in the case where the arguments are the same language.

Proposition 3.4.15. *Let L be a language over Σ. Then $L \flat (L \flat L) = (L \flat L) \flat L$ and $L \natural (L \natural L) = (L \natural L) \natural L$.*

Proof. First of all, by definition, $L \subseteq L \flat L$ and $L \subseteq L \natural L$.

Let $s \in L \flat (L \flat L)$. Then $s = wuv$, where $wu \in L, uv \in L \flat L$, and $uv = xyz$, where $xy \in L, yz \in L$. So $s = wuv = wxyz$. Consider the length of u. If $|\, u \,| > |\, xy \,|$, then

$$s \in wu \flat yz \subseteq L \flat L \subseteq (L \flat L) \flat L. \tag{3.93}$$

Otherwise $|\, u \,| \leq |\, xy \,|$. Then

$$s \in (wu \flat xy) \flat yz \subseteq (L \flat L) \flat L. \tag{3.94}$$

Hence $L \flat (L \flat L) \subseteq (L \flat L) \flat L$. Considering the language L^R that contains the reverse of all words in L, it follows that $L^R \flat (L^R \flat L^R) \subseteq (L^R \flat L^R) \flat L^R$. Applying reversal operator on both sides, we get $(L \flat L) \flat L \subseteq L \flat (L \flat L)$. So $L \flat (L \flat L) = (L \flat L) \flat L$.

The proof of $L \natural (L \natural L) = (L \natural L) \natural L$ is similar. The only difference is that u and y are non-empty. □

As an immediate consequence, the following corollary holds.

Corollary 3.4.16. *Let L be a language over Σ. Then the order of calculation does not matter for two or more applications of $\flat$ (respectively, $\natural$) of L.*

Let $S^{\flat}$ be the *concatenation-with-overlap closure* of S, which is the set of words that can be written as concatenation-with-overlap of finitely many words in S. In other words,

$$S^{\flat} = \{\, \epsilon \,\} \cup S \cup (S \flat S) \cup (S \flat S \flat S) \cup \cdots. \tag{3.95}$$

The concatenation-with-overlap closure is well-defined, since the i-times $\flat$ of S with itself is a subset of $(i+1)$-times $\flat$ of S with itself, and thus the infinite

set-union converges. Similarly, for the $\natural$ operation, let $S^{\natural}$ be the *concatenation-with-nonempty-overlap closure* of S, which is the set of words that can be written as concatenation-with-nonempty-overlap of finitely many words in S. In other words,

$$S^{\natural} = \{\,\epsilon\,\} \;\cup\; S \;\cup\; (S \natural S) \;\cup\; (S \natural S \natural S) \;\cup\; \cdots . \tag{3.96}$$

By similar reasoning, the concatenation-with-nonempty-overlap closure is also well-defined. Furthermore, the family of regular languages is closed under both of the closure operations. In 2009, Shallit [154] proposed a way to prove the following proposition by considering a normal form where at most two words overlap at any given position.

Proposition 3.4.17. *[154] If L is a regular language, then both $L^{\flat}$ and $L^{\natural}$ are regular languages as well. Furthermore, if L is accepted by an NFA of n states, then there is an NFA of $O(n^2)$ states accepting each of $L^{\flat}$ and $L^{\natural}$, respectively.*

The construction in Theorem 3.2.5 can be altered to accept $L = S^{\flat}$ by replacing the transition function by

$$\delta([x,T],a) = \begin{cases} [xa, U], & \text{if } |\,x\,| < n-1; \\ [x[2\ldots|\,x\,|]a, U \setminus \{\,n\,\}], & \text{if } |\,x\,| = n-1, \text{ where} \end{cases} \tag{3.97}$$

$$U = \begin{cases} \{\,0, n+1 : n \in T\,\}, & \text{if } x[n-i\ldots|\,x\,|]a \in S \text{ for some } j \in T,\ j \le i; \\ \{\,n+1 : n \in T\,\}, & \text{otherwise.} \end{cases}$$

Replacing "$j \le i$" by "$j < i$" in the definition of δ, the resulting DFA accepts $L = S^{\natural}$. The numbers of states in both DFAs are

$$\le \frac{2}{2\,|\Sigma| - 1}(2^{\nu}\,|\Sigma|^{\nu} - 1) = O(2^{\nu}\,|\Sigma|^{\nu}), \tag{3.98}$$

where $\nu = \mathrm{llw}\,(S)$.

We will now discuss the co-finiteness in the concatenation closure, with overlap and with non-empty overlap, and define the corresponding generalizations of the Frobenius problem. Nevertheless, finite words over an alphabet with neither the $\flat$ operation nor the $\natural$ operation define a monoid, since the operations do not satisfy the associative laws.

Problem 3.4.18. *Given k non-empty words $x_1, x_2, \ldots, x_k \in \Sigma^*$ over a finite alphabet Σ such that there are only finitely many words that cannot be written as concatenations with overlap (respectively, with non-empty overlap) of words in $\{\,x_1, x_2, \ldots, x_k\,\}$, then what is the longest such word(s)?*

Let S be the set of words $x_1, x_2, \ldots, x_k$. Our first observation about co-finiteness is the following proposition, which is the analog of Proposition 2.1.4.

Proposition 3.4.19. *Let S be a set of words over the alphabet Σ. Then $S^\flat$ (respectively, $S^\natural$) is co-finite if and only if there is an integer $y \geq 2$ such that $\Sigma^y \subseteq S^\flat$ (respectively, $S^\natural$). Furthermore, the length of the longest words not in $S^\flat$ (respectively, $S^\natural$) is $\leq y - 1$.*

Proof. $\Leftarrow$: If there is such a y that $\Sigma^y \subseteq S^\flat$ (respectively, $S^\natural$), then by concatenating i words in Σ^y with non-empty overlap, S can generate all words of length $y + i - 1$, for $i \geq 1$. Hence $S^\flat$ (respectively, $S^\natural$) is co-finite, and the length of the longest words not in the generated languages is $\leq y - 1$.

$\Rightarrow$: Suppose $L = S^\flat$ (respectively, $L = S^\natural$) is co-finite. Let y' be the length of the longest words not in L, and $y = \max\{2, y' + 1\}$. Then $\Sigma^y \subseteq L$, and $y' \leq y - 1$ holds. □

Let S be a language such that $S^\flat$ or $S^\natural$ is co-finite. Then below we prove that there is a finite subset $T \subseteq S$ such that $T^\flat = S^\flat$ or $T^\natural = S^\natural$. So we can always assume a set of words to generate a language is finite, since otherwise we can just choose the equivalent finite set. The following proposition is analogous to Proposition 2.1.3.

Proposition 3.4.20. *Let S be a set of words over Σ. If $S^\flat$ (respectively, $S^\natural$) is co-finite, then there exists a finite subset $T \subseteq S$ such that $T^\flat = S^\flat$ (respectively, $T^\natural = S^\natural$).*

Proof. Let $L = S^\flat$ (respectively, $L = S^\natural$). If $L = \Sigma^*$, then $\Sigma \subseteq S$. Let $T = \Sigma$. Then $T^* = \Sigma^* = S^*$. Otherwise, let l be the length of the longest words not in L. Consider the language

$$U = \left(\Sigma \cup \Sigma^2 \cup \cdots \cup \Sigma^{l+1}\right) \cap L. \tag{3.99}$$

Let T be the set of all words that appear in a factorization of a word in U in the basis S with respect to $\flat$ (respectively, $\natural$). Then T is a finite subset of S, and thus $T^\flat \subseteq S^\flat$ (respectively, $T^\natural \subseteq S^\natural$). On the other hand, any word of length $\geq l + 1$ can be written as a concatenation with non-empty overlap of words of length $l + 1$. Hence $S^\flat \subseteq U^\flat \subseteq T^\flat$ (respectively, $S^\natural \subseteq U^\natural \subseteq T^\natural$). Therefore, $T^\flat = S^\flat$ (respectively, $T^\natural = S^\natural$). □

Since $S^* \subseteq S^\flat$, if S^* is co-finite, then $S^\flat$ is also co-finite. But for $S^\natural$, when S^* is co-finite, $S^\natural$ is not necessarily co-finite. A trivial counter-example is $S = \Sigma$, in which case $S^* = \Sigma^*$, but $S^\natural = S = \Sigma$ is not co-finite. Let p, q be two distinct positive integers with $\gcd(p, q) = 1$, and we assume $|\Sigma| \geq 2$. Then

$$S = \{\mathtt{0}\} \cup \Sigma^p \cup \Sigma^q \setminus \{\mathtt{0}^p, \mathtt{0}^q\} \tag{3.100}$$

generates a co-finite language S^* and $S^* \neq \Sigma^*$, but $S^\natural$ is not co-finite, since none of the words in $\mathtt{000}^*$ is in $S^\natural$. On the other hand, when either $S^\flat$ or $S^\natural$ is co-finite,

S^* is not necessarily co-finite. For example, let $S = \Sigma^2$. Then both $S^\flat$ and $S^\natural$ are co-finite, but S^* is not co-finite.

Since $S^\natural \subseteq S^\flat$, if $S^\natural$ is co-finite, then $S^\flat$ is also co-finite. The converse is not true in general, and the counter-example $S = \Sigma$ also applies. If all the lengths of words in S are ≥ 2, however, the co-finiteness in $S^\natural$ and in $S^\flat$ are equivalent.

Proposition 3.4.21. *Let S be a set of words of lengths ≥ 2 over Σ. Then $S^\natural$ is co-finite if and only if $S^\flat$ is co-finite.*

Proof. Since $S^\natural \subseteq S^\flat$, if $S^\natural$ is co-finite, then $S^\flat$ is also co-finite.

Now we suppose $S^\flat$ is co-finite. By Proposition 3.4.19, there is an integer $y \geq 2$ such that $\Sigma^y \subseteq S^\flat$. Let w be a word of length $2y - 1$. Since each of the words

$$w[1..y], w[2..y+1], \ldots, w[y..2y-1] \tag{3.101}$$

is of length y, they can be factorized in the basis S with respect to $\flat$. Let u_i, v_i be the first and last factors in the factorization of each of the words, for $1 \leq i \leq y$. Then all the u_i and the v_i are of lengths ≥ 2, and we can write w as

$$w \in u'_1 \natural u'_2 \natural \cdots \natural u'_z \natural w[y..2y-1], \quad w \in w[1..y] \natural v_1 \natural v_2 \natural \cdots \natural v_y, \tag{3.102}$$

where $u'_1 = u_1, u'_2 = u_{|u'_1|}, u'_3 = u_{|u'_1 u'_2|}, \ldots$. Hence

$$w \in u'_1 \natural u'_2 \natural \cdots \natural v_1 \natural v_2 \natural \cdots \natural v_y, \tag{3.103}$$

and thus w is in $S^\natural$. So $\Sigma^{2y-1} \subseteq S^\natural$. Then by Proposition 3.4.19, $S^\natural$ is co-finite. □

Since the co-finiteness of $S^\flat$ and $S^\natural$ is equivalent when words in S are of lengths ≥ 2, in the following discussion, we will only show the co-finiteness of $S^\flat$; the co-finiteness of $S^\natural$ then follows. Furthermore, by the proof of Proposition 3.4.21, the length of the longest omitted words differs at most by a multiple of 2.

When either $S^\flat$ or $S^\natural$ is co-finite, then both S^ω and ${}^\omega S$ are co-finite in Σ^ω and ${}^\omega\Sigma$ respectively, and thus by Proposition 3.4.10 ${}^\omega S^\omega$ is co-finite in ${}^\omega\Sigma^\omega$. The converse is not true in general. For example, let

$$S = \{ \mathtt{00}, \mathtt{010}, \mathtt{011}, \mathtt{10}, \mathtt{11} \}. \tag{3.104}$$

Then S^ω is co-finite in Σ^ω, but neither $S^\flat$ nor $S^\natural$ is co-finite in Σ^*, since $(\mathtt{01})^* \subseteq \overline{S^\flat}$.

Proposition 3.4.22. *Let S be a finite set of words of lengths ≥ 2 over Σ. Then $S^\flat$ is co-finite in Σ^* if and only if both S^ω and ${}^\omega S$ are co-finite in Σ^ω and in ${}^\omega\Sigma$, respectively.*

Proof. $\Rightarrow$: If $S^\flat$ is co-finite, by Proposition 3.4.19, there is an integer k such that $\Sigma^k \subseteq S^\flat$. Then any infinite word w has a prefix of length k, which is in $S^\flat$, and

thus has a non-empty prefix in S. So by Proposition 3.4.3 S^ω is co-finite. The case ${}^\omega S$ is similar.

$\Leftarrow$: Let n be the length of the longest words in S, and $w = a_1 a_2 \cdots a_{2n-2}$ be a word of length $2n-2$. Consider the following words

$$u_1 = a_1 a_2,\ u_2 = a_2 a_3,\ \ldots,\ u_{n-1} = a_{n-1} a_n,\ v_1 = a_{n-1} a_n,\ \ldots,\ v_{n-1} = a_{2n-3} a_{2n-2}. \tag{3.105}$$

Since each of the languages $w[i..2n-2] \cdot \Sigma^\omega$ and ${}^\omega\Sigma \cdot w[1..i+n-1]$ is infinite for $1 \le i \le n-1$, there is a word in each of them that can be factorized in the basis S. Consider the first factor, say u'_i, and the last factor, say v'_i, respectively, in the factorizations of such words. Since words in S are of lengths ≥ 2, u_i must be a prefix of u'_i, and v_i must be a suffix of v'_i. Now let $u''_1 = u'_1$, $u''_2 = u'_{|u''_1|}$, $u''_3 = u'_{|u''_1 u''_2|}$, $\ldots$, u''_p, $v''_1 = v'_{|u''_1 u''_2 \ldots u''_p|-n+1}$, $v''_2 = v'_{|u''_1 u''_2 \ldots u''_p v''_1|-n+1}$, $\ldots$, v''_q. Then one can verify that

$$w \in u''_1 \natural u''_2 \natural \cdots u''_p \natural v''_1 \natural v''_2 \natural \cdots \natural v''_q. \tag{3.106}$$

So $\Sigma^{2n-2} \subseteq S^\natural$. Since $2n-2 \ge 2$, by Proposition 3.4.19, $S^\natural$ is co-finite. □

From the proof of Proposition 3.4.22, it follows that if S is a finite set of words of lengths ≥ 2 over Σ, and $S^\natural$ is co-finite, then we have

$$\mathrm{llw}\left(\overline{S^\natural}\right) \le 2\nu - 3, \tag{3.107}$$

where ν is the length of the longest words in S. The proof is in fact also valid for $\natural$, and thus the length of the longest words not in $S^\natural$ is also $\le 2\nu - 3$. When $\nu \ge 3$, then by a similar construction, the upper bound can be improved to $2\nu - 4$. Both bounds are tight. The upper bound $2\nu - 3$ is linear in ν, while in the FPFM an upper bound on the length of the longest omitted words can be exponential in ν.

In the 1FPFM, where the basis S consists of words of the same length n, by Proposition 2.4.1, if S^* is co-finite, then n must be 1 and $S = \Sigma$. In the setting of $\natural$ and $\natural$, however, there are bases consist of words of the same length that generate non-trivially co-finite languages. Here non-trivially co-finite means co-finite but $\neq \Sigma^*$.

Proposition 3.4.23. *Let $S \subseteq \Sigma^n$ for some $n \ge 2$. Then $S^\natural$ is co-finite if and only if $S = \Sigma^n$.*

Proof. If $S = \Sigma^n$, then one can verify by definition that $S^\natural$ is co-finite.

Suppose $S^\natural$ is co-finite. Let $w \in \Sigma^n$. Then the set $w\Sigma^* \cap S^\natural$ is not empty. Let wu be a word in $S^\natural$. Then

$$wu \in v_1 \natural v_2 \natural \cdots \natural v_j, \tag{3.108}$$

where all the v's are in S and thus each is of length n. Comparing lengths shows $v_1 = w$ and thus $w \in S$. Therefore, $S = \Sigma^n$. □

When $S \subseteq \Sigma^n$, where $n \geq 2$, and $S^\flat$ is co-finite, then the longest words not in $S^\flat$ are of length exactly $n-1$. None of the words of length $n-1$ can be written as a concatenation with overlap of words of length n. Any word w of length $n+i$, for $i \geq 0$, can be written as a concatenation of $i+1$ words of length n with non-empty overlap, and thus w is in $S^\flat$. It is also true for $S^\natural$, that when S consists of words of the same length n, where $n \geq 2$, and $S^\natural$ is co-finite, then the longest omitted words are of length exactly $n-1$.

Now we will discuss the analog of the 2FPFM in the setting of $\flat$ and $\natural$. Let m, n be two lengths of words. In contrast to the original 2FPFM, here $\gcd(m,n)$ may be an integer other than 1. First, we will prove an analog of Corollary 2.4.13.

Theorem 3.4.24. *Let S be a set of words of lengths m and n, where $1 < m < n$, over the alphabet Σ. Then $S^\flat$ is co-finite if and only if S is of the form*

$$(\Sigma^m \setminus T) \ \cup \ \Sigma^{n-m}T \ \cup \ T\Sigma^{n-m} \ \cup \ U, \tag{3.109}$$

where $T \subseteq \Sigma^m, U \subseteq \Sigma^n$.

Proof. $\Leftarrow$: Let w be a word of length $2n-m-1$. Consider every factor $u_i = w[i..i+m-1]$ for $1 \leq i \leq 2n-2m$. Since $m \geq 2$, we have

$$w \in u_1 \flat u_2 \flat \cdots \flat u_{2n-2m}. \tag{3.110}$$

Now define u_i' as follows:

$$u_i' = \begin{cases} u_i, & \text{if } u_i \in S; \\ w[i+m-n..i+m-1], & \text{if } u_i \notin S, i+m > n; \\ w[i..i+n-1], & \text{if } u_i \notin S, i+m \leq n. \end{cases} \tag{3.111}$$

Then all the u_i' are well-defined and are in $(\Sigma^m \setminus T) \ \cup \ \Sigma^{n-m}T \ \cup \ T\Sigma^{n-m}$. Furthermore, the u_i' cover all letters of w, and each u_i' covers at least those letters in $w[i..i+m-1]$. Now let v_1 be u_1', and v_i be $u'_{|v_1v_2\cdots v_{i-1}|}$ for $i \geq 2$. Then $w \in v_1 \flat v_2 \flat \cdots \flat v_j$ for some j. Hence

$$\Sigma^{2n-m-1} \subseteq \left((\Sigma^m \setminus T) \ \cup \ \Sigma^{n-m}T \ \cup \ T\Sigma^{n-m}\right)^\flat \subseteq S^\flat. \tag{3.112}$$

Since $2n-m-1 > n-1 \geq 2$, we have $S^\flat$ is co-finite by Proposition 3.4.19.

$\Rightarrow$: Let $T = \Sigma^m \setminus S$, $w \in T$, and $u \in \Sigma^{n-m}$. Since $S^\flat$ is co-finite, the set $wu\Sigma^* \ \cap \ S^\flat$ is not empty. Let wuv be a word in $S^\flat$. Consider a factorization of wuv induced by $\flat$. Since w is not in S, the first factor is of length n, and thus it is wu. Hence wu is in S. By the arbitrary choice of w and u, $T\Sigma^{n-m} \subseteq S$. Similarly $\Sigma^{n-m}T \subseteq S$. Therefore, S is of the form $S = (\Sigma^m \setminus T) \cup \Sigma^{n-m}T \cup T\Sigma^{n-m} \cup S$. $\square$

From the preceding proof, it follows that when the words in S are of two lengths m, n with $1 < m < n$ and $S^\flat$ is co-finite, then $\operatorname{llw}\left(\overline{S^\flat}\right) \leq 2n-m-2$. One can

verify that the construction in the proof for $\flat$ in fact also valid for $\natural$, and thus the length of the longest words not in $S^{\natural}$ is also $\leq 2n-m-2$. The bound is tight. For example, assume $|\Sigma| \geq 2$ and let

$$S = \{ \mathtt{0}^m \} \ \cup \ \Sigma^n \setminus (\Sigma^{n-m}\mathtt{0}^m \ \cup \ \mathtt{0}^m\Sigma^{n-m}). \tag{3.113}$$

Then $S^{\flat}$ is co-finite, but the word $\mathtt{0}^{n-m-1}\mathtt{1}^m\mathtt{0}^{n-m-1}$ is in neither $S^{\flat}$ nor $S^{\natural}$.

Additive alphabets

Now we will discuss another type of concatenation with overlap, which takes place over an additive abelian monoid as alphabet, such as the infinite alphabet $\mathbb{N}$, and the finite alphabet $\mathbb{Z}_k$.

Let Σ be an alphabet such that the addition "+" is defined in Σ and $(\Sigma, +)$ is an abelian monoid. Now we define *concatenation with additive overlap* of two words over Σ as follows:

$$u \sharp v = \left\{ u'w'v' : u = u'a_1 \cdots a_n, v = b_1 \cdots b_n v', w' = (a_1 + b_1) \cdots (a_n + b_n), n \geq 0 \right\}, \tag{3.114}$$

and we define concatenation with additive overlap of two languages L, R as

$$L \sharp R = \left\{ w : w \in u \sharp v, u \in L, v \in R \right\}. \tag{3.115}$$

For example, $\mathtt{123} \sharp \mathtt{321} = \{ \mathtt{123321}, \mathtt{12621}, \mathtt{1551}, \mathtt{444} \}$. If $\Sigma = \{ \mathtt{0} \}$ with normal addition, then concatenation with additive overlap is the same as concatenation with overlap over the unary alphabet. Generally, the operation $\sharp$ does not satisfy the associative law. For example, over $\mathbb{N}$, let $L_1 = L_3 = \{ \mathtt{11} \}, L_2 = \{ \mathtt{1} \}$. Then

$$(L_1 \sharp L_2) \sharp L_3 = \{ \mathtt{111}, \mathtt{12} \} \ \sharp \ \{ \mathtt{11} \} = \left\{ \mathtt{1}^5, \mathtt{1121}, \mathtt{122}, \mathtt{1211}, \mathtt{131}, \mathtt{23} \right\}, \tag{3.116}$$

$$L_1 \sharp (L_2 \sharp L_3) = \{ \mathtt{11} \} \ \sharp \ \{ \mathtt{111}, \mathtt{21} \} = \left\{ \mathtt{1}^5, \mathtt{1211}, \mathtt{221}, \mathtt{1121}, \mathtt{131}, \mathtt{32} \right\}. \tag{3.117}$$

Even in the case $L \sharp L$, the associative law does not hold. For example, let $L = \{ \mathtt{1}, \mathtt{111} \}$ over $\mathbb{N}$. Consider the word $w = \mathtt{1231}$. Since the sum of all letters appearing in w is 7, w can only be factorized into seven $\mathtt{1}$'s or into two $\mathtt{111}$'s and one $\mathtt{1}$. In the later case, $w \in (\mathtt{111} \sharp \mathtt{1}) \sharp \mathtt{111} \subseteq (L \sharp L) \sharp L$, but $w \notin \mathtt{111} \sharp (\mathtt{1} \sharp \mathtt{111}) \subseteq L \sharp (L \sharp L)$. Nevertheless, the *concatenation-with-additive-overlap closure* of S can still be defined as the set of words that can be written as concatenation-with-additive-overlap of finitely many words in S using all possible orders of association of the operations. In other words,

$$S^{\sharp} = \{ \epsilon \} \ \cup \ S \ \cup \ S^{\sharp_2} \ \cup \ S^{\sharp_3} \ \cup \ \cdots, \tag{3.118}$$

where $S^{\sharp_i} = \bigcup_{p,q>0, p+q=i} S^{\sharp_p} \sharp S^{\sharp_q}$ for $i \geq 2$.

Over the infinite alphabet $\mathbb{N} = \{ \mathtt{0}, \mathtt{1}, \mathtt{2}, \dots \}$, a set S of words can generate a co-finite language non-trivially with respect to $\sharp$. (Here, non-trivial means $S^{\sharp} \neq \mathbb{N}^*$.) For example, let $S = \{ \mathtt{3}, \mathtt{5} \} \ \cup \ \{ \mathtt{00}, \mathtt{01}, \mathtt{10} \}$. Then

$$S^{\sharp} = (\mathbb{N}^* \setminus \mathbb{N}) \ \cup \ \langle \mathtt{3}, \mathtt{5} \rangle = \mathbb{N}^* \setminus \{ \mathtt{0}, \mathtt{1}, \mathtt{2}, \mathtt{4}, \mathtt{7} \}. \tag{3.119}$$

To see this, note that $\{ \mathtt{00}, \mathtt{01}, \mathtt{10} \}$ generates $\{ \mathtt{00}, \mathtt{01}, \mathtt{10}, \mathtt{000}, \mathtt{001}, \mathtt{010}, \mathtt{100} \}$, which again generates all words in $\mathbb{N}^2 \cup \mathbb{N}^3$, which finally generates all words in $\mathbb{N}^* \setminus \mathbb{N}$. In addition, $\gcd(\mathtt{3}, \mathtt{5}) = 1$, so $\langle \mathtt{3}, \mathtt{5} \rangle$ is co-finite in $\mathbb{N}$.

Over the infinite alphabet $\hat{\mathbb{N}} = \{ \mathtt{1}, \mathtt{2}, \dots \}$, however, a finite set S of words can only generate a co-finite language trivially as $S^* = \hat{\mathbb{N}}^*$ with respect to $\sharp$. To see this, let S be a finite set of words. If the word $\mathtt{1} \in S$, then $S^\sharp = \hat{\mathbb{N}}^*$, and $S^\sharp$ is trivially co-finite. Now we assume $\mathtt{1} \notin S$. Consider the language

$$L = \mathtt{1}\hat{\mathbb{N}}\mathtt{1}. \tag{3.120}$$

Let $w = \mathtt{1}a\mathtt{1}$ be a word in L such that w is in $S^\sharp$. Then either w is in S, or $w \in \mathtt{1}b \sharp c\mathtt{1}$, where $\mathtt{1}b, c\mathtt{1} \in S$ and $a = b + c$. Since S is finite, there are only finitely many such w in $S^\sharp$. Hence S^* cannot be co-finite.

Over the modulo-k residue classes $\mathbb{N}_k$, a set S of words can generate a co-finite language non-trivially with respect to $\sharp$. For example, let $S = \mathtt{1}\mathbb{N}_k$. Then $S^\sharp = \mathbb{N}_k^* \setminus \mathbb{N}_k$ is co-finite. To see this, let $w = a_1 a_2 \cdots a_k$ be a word of length $k \geq 2$. Then $w \in (\mathtt{10})^{\sharp a_1} \sharp (\mathtt{10})^{\sharp a_2} \cdots (\mathtt{10})^{\sharp a_{k-1}} \sharp (\mathtt{01})^{\sharp a_k} \subseteq S^\sharp$.

3.4.3 Other variations

Slender languages

In the previous discussion, all our generalizations of the Frobenius problem dealt with co-finiteness, which means only finitely many words are omitted. For any alphabet Σ with $|\Sigma| \geq 2$, the number of words of lengths $\leq j$ is $O(|\Sigma|^{j+1})$. In view of this, a finite language is a set of words where the number of words of lengths $\leq j$ is $O(1)$. Now, we allow infinitely many words omitted but the number of omitted words of lengths $\leq j$ must be $O(j)$.

A language L is *slender* if there is a constant c such that for all integers j, the number of words of length j in L is less than c. For example, $(\mathtt{01})^* = \{ \epsilon, \mathtt{01}, \mathtt{0101}, \dots \}$ is a slender language. Finite languages are special cases of slender languages. In analogy with co-finiteness, a language is called *co-slender* if its complement is slender. Thus, a co-finite language is co-slender.

Now we consider sets T such that by adding one word T generates a co-finite language, but T itself does not generate a co-finite language. In other words, let S be a set of words such that S^* is co-finite, and $T = S \setminus \{ w \}$, where T^* is not co-finite. Then either T^* is co-slender, or T^* is not. Both cases are possible. For example, let $S = \{ \mathtt{1}, \mathtt{00}, \mathtt{01}, \mathtt{10}, \mathtt{000}, \mathtt{010} \}$ over $\Sigma = \{ \mathtt{0}, \mathtt{1} \}$. Then $S^* = \Sigma^* \setminus \{ \mathtt{0} \}$ is co-finite. Let $T_1 = S \setminus \{ \mathtt{00} \}$, and $T_2 = S \setminus \{ \mathtt{010} \}$. Then $T_1^* \subseteq \Sigma^* \setminus \mathtt{001}\Sigma^*$ is not co-slender, but $T_2^* = \Sigma^* \setminus \mathtt{0}(\mathtt{10})^*$ is co-slender.

Obviously there are bases that generate co-slender languages but not co-finite languages (the T_2 discussed above). But the following proposition shows a relation between co-slender languages and co-finite languages of infinite words.

Proposition 3.4.25. *Suppose $|\Sigma| \geq 2$. Let S be a finite set of finite words over Σ such that S^* is a co-slender language. Then S generates co-finite languages in Σ^ω, in ${}^\omega\Sigma$, in ${}^\omega\Sigma^\omega$, and in finite words with overlap-concatenation.*

Proof. Since co-finiteness of generated languages in both right-infinite words and left-infinite words implies co-finiteness of generated languages in finite words with overlap-concatenation and co-finiteness in bi-infinite words, it is sufficient to show that S^ω is co-finite in Σ^ω. Since S is a set of finite words, $(S^R)^*$ is also a co-slender language. By symmetry, the co-finiteness of ${}^\omega S$ in ${}^\omega\Sigma$ follows.

Let $n = \mathrm{llw}\,(S)$. If for each word w of length n, there is a non-empty prefix of w in S, then by Proposition 3.4.3, $S^\omega = \Sigma^\omega$ is co-finite. Otherwise, suppose there is a word w of length n such that no non-empty prefix of w is in S. Then $w\Sigma^* \cap S^* = \emptyset$, which contradicts the fact that S^* is co-slender. □

Conversely, co-slenderness cannot be implied by any co-finiteness in Proposition 3.4.25. For example, let the basis $S = \Sigma^2$. Then S^ω is co-finite in Σ^ω, ${}^\omega S$ is co-finite in ${}^\omega\Sigma$, ${}^\omega S^\omega$ is co-finite in ${}^\omega\Sigma^\omega$, and both $S^\flat, S^\natural$ are co-finite in Σ^*; but S^* is not co-slender.

The condition in Proposition 3.4.25 that S is a finite set cannot be omitted. There are infinite sets S of finite words such that S^* is co-slender in Σ^* and S^ω is not co-finite in Σ^ω. For example, let $S = \Sigma^*\mathtt{1}\Sigma^*$ over the binary alphabet. Then $S^* = S = \Sigma^* \setminus \mathtt{0}^+$ is co-slender, and $S^\omega = \Sigma^\omega \setminus \Sigma^*\mathtt{0}^\omega$ is not co-finite in Σ^ω.

There are infinite sets S such that S^* is co-slender but no finite subset of S generates a co-slender language. For example, let $S = \mathtt{0}^*\mathtt{1}\mathtt{0}^*$. Then $S^* = \Sigma^* \setminus \mathtt{0}^+$ is co-slender, but for any finite subset $T \subsetneq S$, we have $T^* \cap w\mathtt{1}\Sigma^* = \emptyset$, where $w \in \mathtt{0}^*\mathtt{1} \setminus T$.

Reverse of the basis

The following problem can be viewed as a special case of the FPFM where the basis T satisfies $T^R = T$. Let S be a set of words over Σ. Then what is the longest words not in $\left(S \cup S^R\right)^*$? Suppose each letter in Σ represents a unique color, and there are sufficient supplies of every tile of size $1 \times n$ that is colored according to each word of length n in S. Then the longest $1 \times m$ floor that cannot be mosaics of the given tapes are of length $m = \mathrm{llw}\left(\overline{(S \cup S^R)^*}\right)$. Here we assume the tapes can be reversed but cannot overlap.

Let $\mathcal{L}^R = \mathrm{llw}\left(\overline{(S \cup S^R)^*}\right)$. By considering the basis $T = S \cup S^R$ in the FPFM, we have

$$\mathcal{L}^R < 4^{\mu-\kappa+1} = O(4^\mu), \quad \mathcal{L}^R < \frac{2}{2\,|\Sigma| - 1}(2^\nu\,|\Sigma|^\nu - 1) = O(2^\nu\,|\Sigma|^\nu). \tag{3.121}$$

where μ is the number of symbols in S, κ is the number of words in S, and $\nu = \mathrm{llw}(S)$. There are also examples in each of which the length of the longest words not in the generated language is

$$\breve{\mathcal{L}}^{\mathcal{R}} = \Theta(|\Sigma|^{\nu/2}), \tag{3.122}$$

which will be shown in Chapter 4.

3.5 Generalized local postage-stamp problem

As we saw in Chapter 1, the Frobenius problem is related to the local postage-stamp problem. While the Frobenius problem is to ask for the *largest* integer that cannot be written as a non-negative integer linear combination of given integers, the local postage-stamp problem is to ask for the *smallest* positive integer that cannot be written as a non-negative integer linear combination of given integers with the sum of coefficients being bounded above by a constant.

Now we will discuss a generalized form of the local postage-stamp problem, the *Local Postage Stamp Problem in a Free Monoid* (LPSPFM), which in some cases is related to the FPFM, particularly to the word graph for the 2FPFM.

Problem 3.5.1 (*local postage-stamp problem in a free monoid*)**.** *Given a set S of words of lengths $1 = c_1 < c_2 < \cdots < c_k$, and an integer $h \geq 1$, what is the shortest word(s) that cannot be written as the concatenation of h or fewer words from S?*

Here the property $\Sigma \subseteq S$ is required to avoid trivial cases. Otherwise, the shortest words are exactly $\Sigma \setminus S$, and they are of length 1.

We define the closure of a finite number of concatenation of S, $S^{\leq n}$, and the length of the shortest words not in $S^{\leq n}$, $N_h(S)$, for $n \geq 0$ and $h \geq 1$ as follows:

$$S^{\leq n} = \left(S \cup \{\epsilon\}\right)^n = \bigcup_{0 \leq i \leq n} S^i, \tag{3.123}$$

$$N_h(S) = \min_{w \notin S^{\leq h}} |w|. \tag{3.124}$$

First, I will show some easy results on special cases of the LPSPFM, which follow rather straightforwardly from results on the LPSP for integers. For integers, $N_h(1) = h+1$ and, by Formula (1.83), $N_h(1,n) = n(h+3-n)-1$ if $h \geq n-1$ [60].

Proposition 3.5.2. *For $h \geq 1$, $N_h(\Sigma) = h+1$.*

Proposition 3.5.3. *For $n \geq 2$, $h \geq 1$,*

$$N_h(\Sigma \cup \Sigma^n) = \begin{cases} h+1, & \text{if } h \leq n-2; \\ n(h+3-n)-1, & \text{if } h \geq n-1. \end{cases} \tag{3.125}$$

Now, we consider a subproblem of the LPSPFM, in which the set of words are of a fixed number of distinct lengths. Let 1LPSPFM denote the local postage-stamp problem in a free monoid with basis consisting of words of the same length, and let 2LPSPFM denote the *local postage-stamp problem in a free monoid with basis consisting of words of two lengths.* As we saw in Proposition 3.5.2, the problem 1LPSPFM is trivial.

Problem 3.5.4 (*2LPSPFM*)**.** *Given a set S of words of two distinct lengths $1 = m < n$, and an integer $h \geq 1$, what is the shortest word(s) that cannot be written as the concatenation of h or fewer words from S?*

Proposition 3.5.5. *Let $\Sigma \subseteq S \subseteq \Sigma \cup \Sigma^n$. If $h < n - 1$, then $N_h(S) = h + 1$.*

Proof. Observe that words of length of $h + 1$ cannot be factorized into h words of length 1, and the only other words available are of length $n > h + 1$. So we have $\Sigma^{h+1} \setminus S^{\leq h} \neq \emptyset$. On the other hand, all words of lengths $\leq h$ are in $S^{\leq h}$. Hence $N_h(S) = h + 1$. □

In what follows, we use the concept of word graph, which was introduced in the discussion on the 2FPFM. Recall, for a set S of words of two lengths m, n with $0 < m < n$ over the alphabet Σ, the word graph $G_S^{(m,n)}$ is a digraph

$$(\Sigma^{n-m}, \Sigma^n \setminus S, \psi), \tag{3.126}$$

where $\psi(w) = (u, v)$ when u is the prefix of w and v is the suffix of w of length $n - m$. Let $w = uw'$. The arc w is labeled by $\varphi(w) = w'$, and for any walk $v_0, a_1, v_1, a_2, v_2, \ldots, v_{k-1}, a_k, v_k$, the labeling of that walk is $v_0\varphi(a_1)\varphi(a_2)\cdots\varphi(a_k)$. In particular, for the basis in the 2LPSPFM, m is always 1.

Theorem 3.5.6. *Let $\Sigma \subseteq S \subseteq \Sigma \cup \Sigma^n$, $n \geq 2$. If there is a cycle in the word graph $G_S^{(1,n)}$, then $N_h(S) = h + 1$.*

Proof. If $h < n - 1$, then $N_h(S) = h + 1$ holds. Assume $h \geq n - 1$. Since there is a cycle in $G_S^{(1,n)}$, there are arbitrarily long closed walks in $G_S^{(1,n)}$. Choose one closed walk of length $(h + 1) - (n - 1)$ and let w be the label of that walk. Then w is of length $h + 1$. Let $w = u_1u_2\cdots u_k$ be a factorization of w into words from S. Then by the definition of the word graph $G_S^{(1,n)}$, none of the factors u_i is of length n. Hence the factorization of w is unique, and at least $h + 1$ words are required. On the other hand, every word of length $\leq h$ can be factorized into a concatenation of at most h words of length 1. So $N_h(S) = h + 1$. □

Let S be a set of words such that $\Sigma \subseteq S \subseteq \Sigma \cup \Sigma^n$ and $n \geq 2$. Now we assume there is no cycle in $G_S^{(1,n)}$, and the label of a longest path in $G_S^{(1,n)}$ is a word of length l. In other words, the longest path in $G_S^{(1,n)}$ is of length $l - (n - 1)$. Then $N_h(S)$ can be calculated recursively by the following two lemmas.

Lemma 3.5.7. *Let $\Sigma \subseteq S \subseteq \Sigma \cup \Sigma^n$, $n \geq 2$. If there is no cycle in the word graph $G_S^{(1,n)}$, and the length of the word labeling a longest path in $G_S^{(1,n)}$ is l, then $N_h(S) = h + 1$ for $h \leq l - 1$.*

Proof. All words of lengths $\leq h$ can be factorized into a concatenation of at most h words of length 1 in S. Let w be a word of length l labeling a longest path in $G_S^{(1,n)}$, and take a prefix u of w of length $h + 1$. Then this word u can be uniquely factorized in the basis S, where each factor is of length 1, and thus at least $h + 1$ words are required to factorize u. Hence $N_h(S) = h + 1$. □

Lemma 3.5.8. *Let $\Sigma \subseteq S \subseteq \Sigma \cup \Sigma^n$, $n \geq 2$. If there is no cycle in the word graph $G_S^{(1,n)}$, and the length of the word labeling a longest path in $G_S^{(1,n)}$ is l, then $N_h(S) = N_{h-(l-n+2)}(S) + l + 1$ for $h \geq l$.*

Proof. Let w be a word of length $\leq N_{h-(l-n+2)}(S) + l$. If $|\, w \,| \leq l$, then w is in $S^{\leq h}$, as $h \geq l$. Assume $|\, w \,| \geq l + 1$. Let $w = uv$, where u is of length $l + 1$ and v is of length $\leq N_{h-(l-n+2)}(S) - 1$. Since there is no cycle in $G_S^{(1,n)}$ and l is the length of the label of a longest path in $G_S^{(1,n)}$, any word of length $l + 1$ can be factorized as the concatenation of a word of length n and $l - n + 1$ words of length 1 in S, not necessarily in that order. So u can be written as the concatenation of $l - n + 2$ words in S. The suffix v is of length $\leq N_{h-(l-n+2)}(S) - 1$, so v can be written as the concatenation of at most $h - (l - n + 2)$ words in S. Therefore, w can be factorized into the concatenation of at most h words, and thus $\Sigma^{\leq N_{h-(l-n+2)}(S)+l} \subseteq S^{\leq h}$.

Now we prove that there is a word of length $N_{h-(l-n+2)}(S)+l+1$ not in $S^{\leq h}$. Let u be a word of length l labeling a longest path in $G_S^{(1,n)}$, and let a be a letter. Then the factorization of ua in S is unique, and is composed of $l-n+1$ words of length 1 and one word of length n, in exactly that order. On the other hand, by definition, there is a word v of length $N_{h-(l-n+2)}(S)$ such that any factorization of v needs at least $h - (l - n + 2) + 1 = h - l + n - 1$ words in S. In addition, any suffix of v of length $|\, v \,| - i$ needs at least $h-l+n-1-i$ words in S. Otherwise, by concatenating with i words of length 1, the factorization of the suffix of v gives a factorization of v with a smaller number of factors. Consider the word $w = uav$. Let $w = w_1 w_2 \cdots w_k$ be a factorization of w in S. If $ua = w_1 w_2 \cdots w_j$ and $v = w_{j+1} w_{j+2} \cdots w_k$, then u contains $l - n + 2$ factors, and v contains at least $h - l + n - 1$ factors. So we have

$$k \geq (l - n + 2) + (h - l + n - 1) = h + 1 > h \tag{3.127}$$

Otherwise, there is a word $w_j = w'_j w''_j$ of length n, such that $ua = w_1 w_2 \cdots w_{j-1} w'_j$ and $v = w''_j w_{j+1} w_{j+2} \cdots w_k$. Then w contains $l + 1 - \left|\, w'_j \,\right|$ factors, and v contains at least $h - l + n - 1 - \left|\, w''_j \,\right|$ factors. So we have

$$k \geq (l + 1 - \left|\, w'_j \,\right|) + (h - l + n - 1 - \left|\, w''_j \,\right|) + 1 = h + 1 > h \tag{3.128}$$

Hence w is of length $N_{h-(l-n+2)}(S) + l + 1$ and is not in $S^{\leq h}$.

Therefore, $N_h(S) = N_{h-(l-n+2)}(S) + l + 1$. □

From the previous two lemmas, Theorem 3.5.9 follows immediately.

Theorem 3.5.9. *Let $\Sigma \subseteq S \subseteq \Sigma \cup \Sigma^n$, $n \geq 2$. If there is no cycle in the word graph $G_S^{(1,n)}$, and the length of a word labeling a longest path in $G_S^{(1,n)}$ is l, then*

$$N_h(S) = \max\left\{ 0, (n-1)\left\lfloor \frac{h-n+2}{l-n+2} \right\rfloor \right\} + h + 1. \tag{3.129}$$

Proof. By Lemmas 3.5.7 and 3.5.8, the recursion for $N_h(S)$ is

$$N_h(S) = \begin{cases} h+1, & \text{if } h \leq l-1; \\ N_{h-(l-n+2)}(S) + l + 1, & \text{if } h \geq l. \end{cases} \tag{3.130}$$

Now we prove Formula (3.129) by induction on h. For $1 \leq h \leq l-1$, we have $(h-n+2)/(l-n+2) < 1$. Then

$$\max\left\{ 0, (n-1)\left\lfloor \frac{h-n+2}{l-n+2} \right\rfloor \right\} + h + 1 = 0 + h + 1 = h + 1 = N_h(S), \tag{3.131}$$

and thus Formula (3.129) is true. Now we assume for all $h < H$, where $H \geq l$, Formula (3.129) is true, and we prove Formula (3.129) is true for $h = H$:

$$\begin{aligned} &N_H(S) = N_{H-(l-n+2)}(S) + l + 1 \\ =&\max\left\{ 0, (n-1)\left\lfloor \frac{(H-(l-n+2))-n+2}{l-n+2} \right\rfloor \right\} + (H-(l-n+2)) + 1 + l + 1 \\ =&(n-1)\left\lfloor \frac{H-n+2}{l-n+2} \right\rfloor - (n-1) + H + n \qquad (3.132) \\ =&\max\left\{ 0, (n-1)\left\lfloor \frac{H-n+2}{l-n+2} \right\rfloor \right\} + H + 1. \qquad (3.133) \end{aligned}$$

Therefore, Formula (3.129) is correct. □

For the set $S = \Sigma \cup \Sigma^n$, the corresponding l is $n-1$. Then by Formula (3.129), $N_h(S) = (n-1)(h-n+2) + (h+1) = n(h+3-n) - 1$, which is the formula in the LPSP for integers. For the set $S = \Sigma$, the digraph $G_S^{(1,n)}$ has cycles. In other words, the length l of the longest path in $G_S^{(1,n)}$ is ∞, which means that there is a cycle. The right-hand-side of Formula (3.129) converges to $h+1$ as l goes to ∞. In the case $S = \Sigma$, $N_h(S) = h + 1$ also holds.

Let S be a set of words of lengths m and n with $m = 1, n \geq 2$ over the alphabet Σ. Then the length of the longest path in the word graph $G_S^{(1,n)}$ can only be in $\{ 0, 1, 2, \ldots, |\Sigma|^{n-1} - 1, \infty \}$, where ∞ means that there is a cycle in $G_S^{(1,n)}$. Then all possible l, the lengths of words labeling longest paths in $G_S^{(1,n)}$, are given by

$$n-1,\ n,\ \ldots,\ |\Sigma|^{n-1} + n - 2,\ \infty. \tag{3.134}$$

Table 3.1: Spectrum of length $N_h(S)$ of shortest words not in $S^{\le h}$ in 2LPSPFM

h	$\lvert\Sigma\rvert=2, n=2$	$\lvert\Sigma\rvert=2, n=3$	$\lvert\Sigma\rvert=2, n=4$
$h=1$	2_{15}[a], 3_1	2	2
$h=2$	$3_{13}, 4_2, 5_1$	$3_{255}, 5_1$	3
$h=3$	$4_{13}, 5_2, 7_1$	$4_{243}, 6_{12}, 8_1$	$4_{65535}, 7_1$
$h=4$	$5_{13}, 7_2, 9_1$	$5_{225}, 7_{30}, 11_1$	$5_{65191}, 8_{344}, 11_1$
$h=5$	$6_{13}, 8_2, 11_1$	$6_{217}, 8_{26}, 10_{12}, 14_1$	$6_{63491}, 9_{2044}, 15_1$
$h=6$	$7_{13}, 10_2, 13_1$	$7_{217}, 9_{26}, 11_{12}, 17_1$	$7_{60841}, 10_{4350}, 13_{344}, 19_1$
$h=7$	$8_{13}, 11_2, 15_1$	$8_{217}, 10_8, 12_{18}, 14_{12}, 20_1$	$8_{59105}, 11_{6086}, 14_{344}, 23_1$
$h=8$	$9_{13}, 13_2, 17_1$	$9_{217}, 11_8, 13_{18}, 15_{12}, 23_1$	$9_{58089}, 12_{5402}, 15_{1700}, 18_{344}, 27_1$
$h=9$	$10_{13}, 14_2, 19_1$	$10_{217}, 14_{26}, 18_{12}, 26_1$	$10_{57641}, 13_{5850}, 16_{1700}, 19_{344}, 31_1$
$h=10$	11[b], $16, 21$	$11_{217}, 15_8, 17_{18}, 19_{12}, 29_1$	$11, 14, 17, 23, 35$

h	$\lvert\Sigma\rvert=3, n=3$	$\lvert\Sigma\rvert=3, n=2$	$\lvert\Sigma\rvert=4, n=2$
$h=1$	2	$2_{511}, 3_1$	$2_{65535}, 3_1$
$h=2$	$3, 5$	$3_{499}, 4_{12}, 5_1$	$3_{65449}, 4_{86}, 5_1$
$h=3$	$4, 6, 8$	$4_{487}, 5_{24}, 7_1$	$4_{65185}, 5_{350}, 7_1$
$h=4$	$5, 7, 11$	$5_{487}, 6_{12}, 7_{12}, 9_1$	$5_{64993}, 6_{456}, 7_{86}, 9_1$
$h=5$	$6, 8, 10, 14$	$6_{487}, 7_{12}, 8_{12}, 11_1$	$6_{64993}, 7_{456}, 8_{86}, 11_1$
$h=6$	$7, 9, 11, 17$	$7_{487}, 9_{12}, 10_{12}, 13_1$	$7_{64993}, 8_{192}, 9_{264}, 10_{86}, 13_1$
$h=7$	$8, 10, 12, 14, 20$	$8_{487}, 10_{12}, 11_{12}, 15_1$	$8_{64993}, 9_{192}, 10_{264}, 11_{86}, 15_1$
$h=8$	$9, 11, 13, 15, 23$	$9_{487}, 11_{12}, 13_{12}, 17_1$	$9_{64993}, 11_{456}, 13_{86}, 17_1$
$h=9$	$10, 12, 14, 18, 26$	$10_{487}, 13_{12}, 14_{12}, 19_1$	$10, 12, 13, 14, 19$
$h=10$	$11, 13, 15, 17, 19, 29$	$11_{487}, 14_{12}, 16_{12}, 21_1$	$11, 13, 14, 16, 21$

[a]The subscript k in m_k means there are k distinct bases S in which $N_h(S)=m$.
[b]Unsubscripted results are computed from the formula, not from calculations.

Then by Theorems 3.5.6 and 3.5.9, the complete spectrum of the length of shortest words as solutions to 2LPSPFM can be obtained. Some of the results of calculations are given in Table 3.1, which validates the spectrum for small n and alphabet size.

Here are some examples over the binary alphabet $\{\,\mathtt{0},\mathtt{1}\,\}$. Let $S_1=\Sigma$, $S_2=S_1\ \cup\ \{\,\mathtt{00},\mathtt{01},\mathtt{11}\,\}$, $S_3=\Sigma\ \cup\ \Sigma^2$. Then there are $|\Sigma|^{(2-1)}+1=3$ different cases given by

$$N_2(S_1)=3,\quad N_2(S_2)=4,\quad N_2(S_3)=5. \tag{3.135}$$

Let $T_1=\Sigma$, $T_2=T_1\cup\{\,\mathtt{000},\mathtt{001},\mathtt{101},\mathtt{111}\,\}$, $T_3=T_2\cup\{\,\mathtt{011}\,\}$, $T_4=T_3\cup\{\,\mathtt{100}\,\}$, $T_5=\Sigma\ \cup\ \Sigma^3$. Then there are $|\Sigma|^{(3-1)}+1=5$ different cases given by

$$N_7(T_1)=8,\quad N_7(T_2)=10,\quad N_7(T_3)=12,\quad N_7(T_4)=14,\quad N_7(T_5)=20. \tag{3.136}$$

Let $U_1=\Sigma$, for $\Sigma=\{\,\mathtt{0},\mathtt{1},\mathtt{2}\,\}$, $U_2=U_1\cup\{\,\mathtt{00},\mathtt{01},\mathtt{02},\mathtt{11},\mathtt{12},\mathtt{22}\,\}$, $U_3=U_2\cup\{\,\mathtt{10}\,\}$, $U_4=\Sigma\ \cup\ \Sigma^2$. Then there are $|\Sigma|^{(2-1)}+1=4$ different cases given by

$$N_4(U_1)=5,\quad N_4(U_2)=6,\quad N_4(U_3)=7,\quad N_4(U_4)=9. \tag{3.137}$$

Over $\Sigma=\{\,\mathtt{0},\mathtt{1},\mathtt{2},\mathtt{3}\,\}$, let $V_1=\Sigma$, $V_2=V_1\cup\{\,\mathtt{00},\mathtt{01},\mathtt{02},\mathtt{03},\mathtt{11},\mathtt{12},\mathtt{13},\mathtt{22},\mathtt{23},\mathtt{33}\,\}$, $V_3=V_2\ \cup\ \{\,\mathtt{10}\,\}$, $V_4=V_3\ \cup\ \{\,\mathtt{32}\,\}$, $V_5=\Sigma\ \cup\ \Sigma^2$. Then there are $|\Sigma|^{(2-1)}+1=5$

different cases given by

$$N_6(V_1) = 7, \quad N_6(V_2) = 8, \quad N_6(V_3) = 9, \quad N_6(V_4) = 10, \quad N_6(V_5) = 13. \quad (3.138)$$

For some h, the $N_h(S)$ corresponding to different cases of l, which is the length of the longest path in the word graph $G_S^{(1,n)}$, may be the same. For example, let $|\Sigma| = 2, n = 3, h = 9$. Since

$$\left\lfloor \frac{h-n+2}{4-n+2} \right\rfloor = \left\lfloor \frac{8}{3} \right\rfloor = 2 = \left\lfloor \frac{8}{4} \right\rfloor = \left\lfloor \frac{h-n+2}{5-n+2} \right\rfloor, \quad (3.139)$$

we have $N_h(S_1) = N_h(S_2) = 14$, where the longest paths in $G_{S_1}^{(1,n)}$ and in $G_{S_2}^{(1,n)}$ are of lengths 2 and 3, respectively. Nevertheless, for sufficiently large h, the $N_h(S)$ for different cases of l are distinct. To see this, when $h > |\Sigma|^{2(n-1)} + n - 2$, we have

$$\left| \frac{h-n+2}{l_1-n+2} - \frac{h-n+2}{l_2-n+2} \right| = \frac{(h-n+2)\,|\,l_2 - l_1\,|}{(l_1-n+2)(l_2-n+2)} > 1, \quad (3.140)$$

where $n-1 \le l_1, l_2 \le |\Sigma|^{n-1} + n - 2$.

Chapter 4

Examples of the FPFM

In this chapter, I will provide several families of examples that achieve some of the upper bounds that appeared in the discussion on the FPFM and its variations in Chapters 2 and 3. Let S be a set of words over the alphabet Σ such that S^* is co-finite. By Corollary 2.3.2, we have

$$\mathcal{L} = \mathrm{llw}\left(\overline{S^*}\right) < \frac{2}{2\left|\Sigma\right| - 1}(2^\nu \left|\Sigma\right|^\nu - 1) = \left|\Sigma\right|^{O(\nu)}. \tag{4.1}$$

In §4.1, I will provide examples where the length $\mathcal{L}$ is exponential in ν. By Corollary 3.3.2, the number of words not in S^* is

$$\mathcal{M} = \left|\,\overline{S^*}\,\right| \leq \frac{\left|\Sigma\right|^q - 1}{\left|\Sigma\right| - 1} = \left|\Sigma\right|^{\left|\Sigma\right|^{O(\nu)}}, \tag{4.2}$$

where $q = \frac{2}{2|\Sigma|-1}(2^\nu \left|\Sigma\right|^\nu - 1)$. In §4.2, I will provide examples where the number $\mathcal{M}$ is doubly-exponential in ν. Finally, in §4.3, I will provide some statistics from experiments.

4.1 Exponential length of the longest words $\notin S^*$

All examples in this section in fact belong to the 2FPFM, the special subproblem of the FPFM. Let S be a set of words of lengths m and n with $0 < m < n$ such that S^* is co-finite. As we saw in Theorem 2.4.12, we have

$$\mathcal{L} = \mathrm{llw}\left(\overline{S^*}\right) \leq g(m, l) = ml - m - l, \tag{4.3}$$

where $l = m|\Sigma|^{n-m} + n - m$. The examples given below achieve $\mathcal{L} = g(m, l)$ in (4.3).

4.1.1 Examples of the 2FPFM with $0 < m < n < 2m$

We now look at some examples achieving an exponential upper bound in $\nu = n$, the length of the longest words in the basis. Without loss of generality, we assume

the alphabet Σ is $\{\,\mathtt{0},\mathtt{1},\mathtt{2},\ldots\}$. We define ${}_c(n)_k$ to be the unique word of length c that represents the non-negative integer n in base k, possibly with leading zeros, and define $[w]_k$ to be the non-negative integer represented by the word w in base k. For example, ${}_5(11)_2 = \mathtt{01011}$, and $[\mathtt{01011}]_2 = 11$. When c and k are clear from the context, we write ${}_c(n)_k$ and $[w]_k$ as (n) and $[w]$ for short, respectively. For two integers m and n with $0 < m < n < 2m$, define

$$T(m,n) = \left\{ {}_c(i)_k \mathtt{0}^{2m-n} {}_c(i+1)_k : 0 \le i \le |\Sigma|^{n-m} - 2,\ k = |\Sigma|\,,\ c = n-m \right\}. \tag{4.4}$$

For example, over the binary alphabet $\Sigma = \{\,\mathtt{0},\mathtt{1}\,\}$, we have

$$T(3,5) = \{\,\mathtt{00001},\mathtt{01010},\mathtt{10011}\,\}\,. \tag{4.5}$$

Theorem 4.1.1. *[83, 84] Let m, n be two integers with $0 < m < n < 2m$ and $\gcd(m,n) = 1$, and let $S = \Sigma^m \cup \Sigma^n \backslash T(m,n)$. Then S^* is co-finite, and the longest words not in S^* are of length $g(m,l) = ml - m - l$, where $l = m\,|\Sigma|^{n-m} + n - m$.*

Proof. Let $l = m\,|\Sigma|^{n-m} + n - m$. Since $\Sigma^m \subseteq S$ and $l \equiv n \pmod m$, in order to show S^* is co-finite and the longest words not in S^* are of length $g(m,l)$, by Theorem 2.5.3, it is sufficient to prove that $S^* \setminus \Sigma^{l-m} \neq \emptyset$ and $S^* \setminus \Sigma^l = \emptyset$.

Let x be a word of length l. Then we can write x uniquely as

$$x = y_0 z_0 y_1 z_1 \cdots y_{|\Sigma|^{n-m}-1} z_{|\Sigma|^{n-m}-1} y_{|\Sigma|^{n-m}}, \tag{4.6}$$

where all the y_i are of length $n-m$, and all the z_i are of length $2m-n$. If all the $y_i z_i y_{i+1}$ are in $T(m,n)$, then $[y_{i+1}]_{|\Sigma|} = [y_i]_{|\Sigma|} + 1$, for $0 \le i < |\Sigma|^{n-m}$. So

$$[y_{|\Sigma|^{n-m}}]_{|\Sigma|} = [y_0]_{|\Sigma|} + |\Sigma|^{n-m}\,. \tag{4.7}$$

But $y_{|\Sigma|^{n-m}}$ is of length $n-m$, and thus it cannot be the base-$|\Sigma|$ expansion of a number $\ge |\Sigma|^{n-m}$, a contradiction. Hence some $y_i z_i y_{i+1}$ is in S, and so

$$x = \left(\prod_{0 \le p < i} y_p z_p \right) y_i z_i y_{i+1} \left(\prod_{i+1 \le q \le |\Sigma|^{n-m}} z_q y_q \right). \tag{4.8}$$

Since $\Sigma^m \subseteq S$, all the $y_p z_p$ and $z_q y_q$ are in S. Therefore, x can be factorized into elements of S, and thus $S^* \setminus \Sigma^l = \emptyset$.

Let $c = n - m$, and $k = |\Sigma|$. Now we claim that there is a word τ of length $l - m$ that is not in S^*, which is

$$\tau = {}_c(0)_k \mathtt{0}^{2m-n} {}_c(1)_k \mathtt{0}^{2m-n} {}_c(2)_k \mathtt{0}^{2m-n} \cdots \mathtt{0}^{2m-n} {}_c(|\Sigma|^{n-m} - 1)_k. \tag{4.9}$$

Suppose there is a factorization of τ in S given by $\tau = w_1 w_2 \cdots w_t$, where all the w_i are in S. Since m does not divide $|\,\tau\,|$, at least one of the factors is of length n. Let w_j be the first factor of length n. Comparing lengths shows $w_i = {}_c(i-1)_k \mathtt{0}^{2m-n}$ for $1 \le i < j$ and $w_j = {}_c(j-1)_k \mathtt{0}^{2m-n} {}_c(j)_k$. Then w_j is in both S and $T(m,n)$, a contradiction. Therefore, $\tau \notin S^*$. □

The word $\tau = {}_c(0)_k\mathtt{0}^{2m-n}{}_c(1)_k\mathtt{0}^{2m-n}\cdots\mathtt{0}^{2m-n}{}_c(|\Sigma|^{n-m}-1)_k$ is the only word in $S^* \setminus \Sigma^{l-m}$ for $S = \Sigma^m \cup \Sigma^n \setminus T(m,n)$. By the same arguments in the proof of Theorem 4.1.1, every word in $S^* \setminus \Sigma^{l-m}$ can be uniquely written as $x = y_0z_0y_1z_1\cdots y_{|\Sigma|^{n-m}-1}$, where all the y_i are of length $n-m$, and all the z_i are of length $2m-n$, and $y_iz_iy_{i+1} \notin S$. So $y_iz_iy_{i+1} \in T(m,n)$ for $0 \le i \le |\Sigma|^{n-m}-1$, and τ is the only such word. By Corollary 2.5.5, the set of all longest words not in S^* is $(\tau\Sigma^m)^{m-2}\tau$.

Example 4.1.2. Let $m = 3, n = 5, \Sigma = \{\,\mathtt{0},\mathtt{1}\,\}$. In this case, $l = 3\cdot 2^2+2 = 14$, $S = \Sigma^3 \cup \Sigma^5 \setminus \{\,\mathtt{00001},\mathtt{01010},\mathtt{10011}\,\}$. Then one of the longest words not in S^* is

$$\mathtt{00001010011\ 000\ 00001010011} \tag{4.10}$$

of length $25 = g(3,14)$. The members of the set $\Sigma^* \setminus S^*$ are given in Table 4.1.

Theorem 4.1.3. *Let $v_1, v_2, \ldots, v_{|\Sigma|^{n-m}}$ be any permutation of all words of length $n-m$ over the alphabet Σ, and let $S = \Sigma^m \cup \Sigma^n \setminus T$, where*

$$T = \left\{\, v_i\mathtt{0}^{2m-n}v_{i+1} : 1 \le i \le |\Sigma|^{n-m}-1 \,\right\}. \tag{4.11}$$

Then S^* is co-finite, and the longest words not in S^* are of length $g(m,l) = ml - m - l$, where $l = m\,|\Sigma|^{n-m} + n - m$.

Using v_i instead of ${}_{n-m}(i)_{|\Sigma|}$ for $0 \le i \le |\Sigma|^{n-m}-1$, the proof of Theorem 4.1.1 is also valid for Theorem 4.1.3.

By Theorem 2.5.3, we know that in order to construct a basis S of words of lengths m and n, where $\gcd(m,n) = 1$, such that S^* is co-finite and the length of the longest words not in S^* attains the upper bound in Theorem 2.4.12, it is essential to construct an S such that S^* is co-finite and there is a word τ of length $m\,|\Sigma|^{n-m} + n - m$ that is not in S^*. When $0 < m < n < 2m$, the S and τ can be found easily, as shown in Theorem 4.1.1. The examples for small m and n are summarized in Table 4.2. Since the proof of Theorem 2.4.12 does not rely on the part $\mathtt{0}^{2m-n}$, one can verify that replacing the $\mathtt{0}^{2m-n}$ part of each word in the set $T(m,n)$ by any word of length $2m-n$ (they do not even have to be the same), then $S = \Sigma^m \cup \Sigma^n \setminus T(m,n)$ is also an example to attain the upper bound in Theorem 2.4.12.

Now we consider a variation on the FPFM. In Chapter 3, we discussed a variation on the FPFM where we consider the language $\left(T \cup T^R\right)^*$ instead of T^*.

Proposition 4.1.4. *Let S be a set of words of lengths m, n with $0 < m < n < 2m$ such that $S^R \subseteq S$ and S^* is co-finite. Then the length of the longest words not in S^* is strictly less than $g(m,l) = ml - m - l$, where $l = m\,|\Sigma|^{n-m} + n - m$. Furthermore, there are examples in which the length of the longest words not in S^* is $\Theta(|\Sigma|^{n-m})$.*

Proof. Consider the word graph $G_S^{(m,n)}$ as defined in Chapter 2, where the set of vertices is Σ^{n-m} and the set of arcs is $\Sigma^n \setminus S$. Then we claim any path in $G_S^{(m,n)}$ encounters at most one vertex that is a palindrome.

Table 4.1: All the words in $\{0,1\}^* \setminus (\{0,1\}^3 \cup \{0,1\}^5 \setminus \{00001, 01010, 10011\})^*$

1	[1]0	89	[7]0111111	177	[10]0000110010	265	[10]1000100001	353	[13]0000101000101
2	[1]1	90	[7]1000000	178	[10]0000110011	266	[10]1000101010	354	[13]0000101000110
3	[2]00	91	[7]1000001	179	[10]0000110100	267	[10]1000110011	355	[13]0000101000111
4	[2]01	92	[7]1000010	180	[10]0000110101	268	[10]1001000001	356	[13]0000101001000
5	[2]10	93	[7]1000011	181	[10]0000110110	269	[10]1001001010	357	[13]0000101001001
6	[2]11	94	[7]1000100	182	[10]0000110111	270	[10]1001010011	358	[13]0000101001010
7	[4]0000	95	[7]1000101	183	[10]0000111000	271	[10]1001100000	359	[13]0000101001011
8	[4]0001	96	[7]1000110	184	[10]0000111001	272	[10]1001100001	360	[13]0000101001100
9	[4]0010	97	[7]1000111	185	[10]0000111010	273	[10]1001100010	361	[13]0000101001101
10	[4]0011	98	[7]1001000	186	[10]0000111011	274	[10]1001100011	362	[13]0000101001110
11	[4]0100	99	[7]1001001	187	[10]0000111100	275	[10]1001100100	363	[13]0000101001111
12	[4]0101	100	[7]1001010	188	[10]0000111101	276	[10]1001100101	364	[13]0000101010000
13	[4]0110	101	[7]1001011	189	[10]0000111110	277	[10]1001100110	365	[13]0000101010001
14	[4]0111	102	[7]1001100	190	[10]0000111111	278	[10]1001100111	366	[13]0000101010010
15	[4]1000	103	[7]1001101	191	[10]0001000001	279	[10]1001101000	367	[13]0000101010011
16	[4]1001	104	[7]1001110	192	[10]0001001010	280	[10]1001101001	368	[13]0000101010100
17	[4]1010	105	[7]1001111	193	[10]0001010011	281	[10]1001101010	369	[13]0000101010101
18	[4]1011	106	[7]1010000	194	[10]0001100001	282	[10]1001101011	370	[13]0000101010110
19	[4]1100	107	[7]1010001	195	[10]0001101010	283	[10]1001101100	371	[13]0000101010111
20	[4]1101	108	[7]1010010	196	[10]0001110011	284	[10]1001101101	372	[13]0000101011000
21	[4]1110	109	[7]1010011	197	[10]0010000001	285	[10]1001101110	373	[13]0000101011001
22	[4]1111	110	[7]1010100	198	[10]0010001010	286	[10]1001101111	374	[13]0000101011010
23	[5]00001	111	[7]1010101	199	[10]0010010011	287	[10]1001110000	375	[13]0000101011011
24	[5]01010	112	[7]1010110	200	[10]0010100001	288	[10]1001110001	376	[13]0000101011100
25	[5]10011	113	[7]1010111	201	[10]0010101010	289	[10]1001110010	377	[13]0000101011101
26	[7]0000000	114	[7]1011000	202	[10]0010110011	290	[10]1001110011	378	[13]0000101011110
27	[7]0000001	115	[7]1011001	203	[10]0011000001	291	[10]1001110100	379	[13]0000101011111
28	[7]0000010	116	[7]1011010	204	[10]0011001010	292	[10]1001110101	380	[13]0000101100001
29	[7]0000011	117	[7]1011011	205	[10]0011010011	293	[10]1001110110	381	[13]0000101101010
30	[7]0000100	118	[7]1011100	206	[10]0011100001	294	[10]1001110111	382	[13]0000101110011
31	[7]0000101	119	[7]1011101	207	[10]0011101010	295	[10]1001111000	383	[13]0000110000001
32	[7]0000110	120	[7]1011110	208	[10]0011110011	296	[10]1001111001	384	[13]0000110001010
33	[7]0000111	121	[7]1011111	209	[10]0100000001	297	[10]1001111010	385	[13]0000110010011
34	[7]0001000	122	[7]1100000	210	[10]0100001010	298	[10]1001111011	386	[13]0000110100001
35	[7]0001001	123	[7]1100001	211	[10]0100010011	299	[10]1001111100	387	[13]0000110101010
36	[7]0001010	124	[7]1100010	212	[10]0100100001	300	[10]1001111101	388	[13]0000110110011
37	[7]0001011	125	[7]1100011	213	[10]0100101010	301	[10]1001111110	389	[13]0000111000001
38	[7]0001100	126	[7]1100100	214	[10]0100110011	302	[10]1001111111	390	[13]0000111001010
39	[7]0001101	127	[7]1100101	215	[10]0101000000	303	[10]1010000001	391	[13]0000111010011
40	[7]0001110	128	[7]1100110	216	[10]0101000001	304	[10]1010001010	392	[13]0000111100001
41	[7]0001111	129	[7]1100111	217	[10]0101000010	305	[10]1010010011	393	[13]0000111101010
42	[7]0010000	130	[7]1101000	218	[10]0101000011	306	[10]1010100001	394	[13]0000111110011
43	[7]0010001	131	[7]1101001	219	[10]0101000100	307	[10]1010101010	395	[13]0001000001010
44	[7]0010010	132	[7]1101010	220	[10]0101000101	308	[10]1010110011	396	[13]0001001010011
45	[7]0010011	133	[7]1101011	221	[10]0101000110	309	[10]1011000001	397	[13]0001100001010
46	[7]0010100	134	[7]1101100	222	[10]0101000111	310	[10]1011001010	398	[13]0001101010011
47	[7]0010101	135	[7]1101101	223	[10]0101001000	311	[10]1011010011	399	[13]0010000001010
48	[7]0010110	136	[7]1101110	224	[10]0101001001	312	[10]1011100001	400	[13]0010001010011
49	[7]0010111	137	[7]1101111	225	[10]0101001010	313	[10]1011101010	401	[13]0010100001010
50	[7]0011000	138	[7]1110000	226	[10]0101001011	314	[10]1011110011	402	[13]0010101010011
51	[7]0011001	139	[7]1110001	227	[10]0101001100	315	[10]1100000001	403	[13]0011000001010
52	[7]0011010	140	[7]1110010	228	[10]0101001101	316	[10]1100001010	404	[13]0011001010011
53	[7]0011011	141	[7]1110011	229	[10]0101001110	317	[10]1100010011	405	[13]0011100001010
54	[7]0011100	142	[7]1110100	230	[10]0101001111	318	[10]1100100001	406	[13]0011101010011
55	[7]0011101	143	[7]1110101	231	[10]0101010000	319	[10]1100101010	407	[13]0100000001010
56	[7]0011110	144	[7]1110110	232	[10]0101010001	320	[10]1100110011	408	[13]0100001010011
57	[7]0011111	145	[7]1110111	233	[10]0101010010	321	[10]1101000001	409	[13]0100100001010
58	[7]0100000	146	[7]1111000	234	[10]0101010011	322	[10]1101001010	410	[13]0100101010011
59	[7]0100001	147	[7]1111001	235	[10]0101010100	323	[10]1101010011	411	[13]0101000000001
60	[7]0100010	148	[7]1111010	236	[10]0101010101	324	[10]1101100001	412	[13]0101000001010
61	[7]0100011	149	[7]1111011	237	[10]0101010110	325	[10]1101101010	413	[13]0101000010011
62	[7]0100100	150	[7]1111100	238	[10]0101010111	326	[10]1101110011	414	[13]0101000100001
63	[7]0100101	151	[7]1111101	239	[10]0101011000	327	[10]1110000001	415	[13]0101000101010
64	[7]0100110	152	[7]1111110	240	[10]0101011001	328	[10]1110001010	416	[13]0101000110011
65	[7]0100111	153	[7]1111111	241	[10]0101011010	329	[10]1110010011	417	[13]0101001000001
66	[7]0101000	154	[8]00001010	242	[10]0101011011	330	[10]1110100001	418	[13]0101001001010
67	[7]0101001	155	[8]01010011	243	[10]0101011100	331	[10]1110101010	419	[13]0101001010011
68	[7]0101010	156	[10]0000000001	244	[10]0101011101	332	[10]1110110011	420	[13]0101001100000
69	[7]0101011	157	[10]0000001010	245	[10]0101011110	333	[10]1111000001	421	[13]0101001100001
70	[7]0101100	158	[10]0000010011	246	[10]0101011111	334	[10]1111001010	422	[13]0101001100010
71	[7]0101101	159	[10]0000100000	247	[10]0101100001	335	[10]1111010011	423	[13]0101001100011
72	[7]0101110	160	[10]0000100001	248	[10]0101101010	336	[10]1111100001	424	[13]0101001100100
73	[7]0101111	161	[10]0000100010	249	[10]0101110011	337	[10]1111101010	425	[13]0101001100101
74	[7]0110000	162	[10]0000100011	250	[10]0110000001	338	[10]1111110011	426	[13]0101001100110
75	[7]0110001	163	[10]0000100100	251	[10]0110001010	339	[11]00001010011	427	[13]0101001100111
76	[7]0110010	164	[10]0000100101	252	[10]0110010011	340	[13]0000000001010	428	[13]0101001101000
77	[7]0110011	165	[10]0000100110	253	[10]0110100001	341	[13]0000001010011	429	[13]0101001101001
78	[7]0110100	166	[10]0000100111	254	[10]0110101010	342	[13]0000100000001	430	[13]0101001101010
79	[7]0110101	167	[10]0000101000	255	[10]0110110011	343	[13]0000100001010	431	[13]0101001101011
80	[7]0110110	168	[10]0000101001	256	[10]0111000001	344	[13]0000100010011	432	[13]0101001101100
81	[7]0110111	169	[10]0000101010	257	[10]0111001010	345	[13]0000100100001	433	[13]0101001101101
82	[7]0111000	170	[10]0000101011	258	[10]0111010011	346	[13]0000100101010	434	[13]0101001101110
83	[7]0111001	171	[10]0000101100	259	[10]0111100001	347	[13]0000100110011	435	[13]0101001101111
84	[7]0111010	172	[10]0000101101	260	[10]0111101010	348	[13]0000101000000	436	[13]0101001110000
85	[7]0111011	173	[10]0000101110	261	[10]0111110011	349	[13]0000101000001	437	[13]0101001110001
86	[7]0111100	174	[10]0000101111	262	[10]1000000001	350	[13]0000101000010	438	[13]0101001110010
87	[7]0111101	175	[10]0000110000	263	[10]1000001010	351	[13]0000101000011	439	[13]0101001110011
88	[7]0111110	176	[10]0000110001	264	[10]1000010011	352	[13]0000101000100	440	[13]0101001110100

441	[13]0101001110101	529	[16]0000100000001010	617	[16]0101001101001010	705	[19]0000101001111001010
442	[13]0101001110110	530	[16]0000100001010011	618	[16]0101001101010011	706	[19]0000101001111010011
443	[13]0101001110111	531	[16]0000100100001010	619	[16]0101001101100001	707	[19]0000101001111100001
444	[13]0101001111000	532	[16]0000100101010011	620	[16]0101001101101010	708	[19]0000101001111101010
445	[13]0101001111001	533	[16]0000101000000001	621	[16]0101001101110011	709	[19]0000101001111110011
446	[13]0101001111010	534	[16]0000101000001010	622	[16]0101001110000001	710	[19]0000101010000001010
447	[13]0101001111011	535	[16]0000101000010011	623	[16]0101001110001010	711	[19]0000101010001010011
448	[13]0101001111100	536	[16]0000101000100001	624	[16]0101001110010011	712	[19]0000101010100001010
449	[13]0101001111101	537	[16]0000101000101010	625	[16]0101001110100001	713	[19]0000101010101010011
450	[13]0101001111110	538	[16]0000101000110011	626	[16]0101001110101010	714	[19]0000101011000001010
451	[13]0101001111111	539	[16]0000101001000001	627	[16]0101001110110011	715	[19]0000101011001010011
452	[13]0101010000001	540	[16]0000101001001010	628	[16]0101001111000001	716	[19]0000101011100001010
453	[13]0101010001010	541	[16]0000101001010011	629	[16]0101001111001010	717	[19]0000101011101010011
454	[13]0101010010011	542	[16]0000101001100000	630	[16]0101001111010011	718	[19]0000101100001010011
455	[13]0101010100001	543	[16]0000101001100001	631	[16]0101001111100001	719	[19]0000110000001010011
456	[13]0101010101010	544	[16]0000101001100010	632	[16]0101001111101010	720	[19]0000110100001010011
457	[13]0101010110011	545	[16]0000101001100011	633	[16]0101001111110011	721	[19]0000111000001010011
458	[13]0101011000001	546	[16]0000101001100100	634	[16]0101010000001010	722	[19]0000111100001010011
459	[13]0101011001010	547	[16]0000101001100101	635	[16]0101010001010011	723	[19]0101000000001010011
460	[13]0101011010011	548	[16]0000101001100110	636	[16]0101010100001010	724	[19]0101000100001010011
461	[13]0101011100001	549	[16]0000101001100111	637	[16]0101010101010011	725	[19]0101001000001010011
462	[13]0101011101010	550	[16]0000101001101000	638	[16]0101011000001010	726	[19]0101001100000001010
463	[13]0101011110011	551	[16]0000101001101001	639	[16]0101011001010011	727	[19]0101001100001010011
464	[13]0101100001010	552	[16]0000101001101010	640	[16]0101011100001010	728	[19]0101001100100001010
465	[13]0101101010011	553	[16]0000101001101011	641	[16]0101011101010011	729	[19]0101001100101010011
466	[13]0110000001010	554	[16]0000101001101100	642	[16]0101100001010011	730	[19]0101001101000001010
467	[13]0110001010011	555	[16]0000101001101101	643	[16]0110000001010011	731	[19]0101001101001010011
468	[13]0110100001010	556	[16]0000101001101110	644	[16]0110100001010011	732	[19]0101001101100001010
469	[13]0110101010011	557	[16]0000101001101111	645	[16]0111000001010011	733	[19]0101001101101010011
470	[13]0111000001010	558	[16]0000101001110000	646	[16]0111100001010011	734	[19]0101001110000001010
471	[13]0111001010011	559	[16]0000101001110001	647	[16]1000000001010011	735	[19]0101001110001010011
472	[13]0111100001010	560	[16]0000101001110010	648	[16]1000100001010011	736	[19]0101001110100001010
473	[13]0111101010011	561	[16]0000101001110011	649	[16]1001000001010011	737	[19]0101001110101010011
474	[13]1000000001010	562	[16]0000101001110100	650	[16]1001100000001010	738	[19]0101001111000001010
475	[13]1000001010011	563	[16]0000101001110101	651	[16]1001100001010011	739	[19]0101001111001010011
476	[13]1000100001010	564	[16]0000101001110110	652	[16]1001100100001010	740	[19]0101001111100001010
477	[13]1000101010011	565	[16]0000101001110111	653	[16]1001100101010011	741	[19]0101001111101010011
478	[13]1001000001010	566	[16]0000101001111000	654	[16]1001101000001010	742	[19]0101010000001010011
479	[13]1001001010011	567	[16]0000101001111001	655	[16]1001101001010011	743	[19]0101010100001010011
480	[13]1001100000001	568	[16]0000101001111010	656	[16]1001101100001010	744	[19]0101011000001010011
481	[13]1001100001010	569	[16]0000101001111011	657	[16]1001101101010011	745	[19]0101011100001010011
482	[13]1001100010011	570	[16]0000101001111100	658	[16]1001110000001010	746	[19]1001100000001010011
483	[13]1001100100001	571	[16]0000101001111101	659	[16]1001110001010011	747	[19]1001100100001010011
484	[13]1001100101010	572	[16]0000101001111110	660	[16]1001110100001010	748	[19]1001101000001010011
485	[13]1001100110011	573	[16]0000101001111111	661	[16]1001110101010011	749	[19]1001101100001010011
486	[13]1001101000001	574	[16]0000101010000001	662	[16]1001111000001010	750	[19]1001110000001010011
487	[13]1001101001010	575	[16]0000101010001010	663	[16]1001111001010011	751	[19]1001110100001010011
488	[13]1001101010011	576	[16]0000101010010011	664	[16]1001111100001010	752	[19]1001111000001010011
489	[13]1001101100001	577	[16]0000101010100001	665	[16]1001111101010011	753	[19]1001111100001010011
490	[13]1001101101010	578	[16]0000101010101010	666	[16]1010000001010011	754	[22]0000101000000001010011
491	[13]1001101110011	579	[16]0000101010110011	667	[16]1010100001010011	755	[22]0000101000100001010011
492	[13]1001110000001	580	[16]0000101011000001	668	[16]1011000001010011	756	[22]0000101001000001010011
493	[13]1001110001010	581	[16]0000101011001010	669	[16]1011100001010011	757	[22]0000101001100000001010
494	[13]1001110010011	582	[16]0000101011010011	670	[16]1100000001010011	758	[22]0000101001100001010011
495	[13]1001110100001	583	[16]0000101011100001	671	[16]1100100001010011	759	[22]0000101001100100001010
496	[13]1001110101010	584	[16]0000101011101010	672	[16]1101000001010011	760	[22]0000101001100101010011
497	[13]1001110110011	585	[16]0000101011110011	673	[16]1101100001010011	761	[22]0000101001101000001010
498	[13]1001111000001	586	[16]0000101100001010	674	[16]1110000001010011	762	[22]0000101001101001010011
499	[13]1001111001010	587	[16]0000101101010011	675	[16]1110100001010011	763	[22]0000101001101100001010
500	[13]1001111010011	588	[16]0000110000001010	676	[16]1111000001010011	764	[22]0000101001101101010011
501	[13]1001111100001	589	[16]0000110001010011	677	[16]1111100001010011	765	[22]0000101001110000001010
502	[13]1001111101010	590	[16]0000110100001010	678	[19]0000100000001010011	766	[22]0000101001110001010011
503	[13]1001111110011	591	[16]0000110101010011	679	[19]0000100100001010011	767	[22]0000101001110100001010
504	[13]1010000001010	592	[16]0000111000001010	680	[19]0000101000000001010	768	[22]0000101001110101010011
505	[13]1010001010011	593	[16]0000111001010011	681	[19]0000101000001010011	769	[22]0000101001111000001010
506	[13]1010100001010	594	[16]0000111100001010	682	[19]0000101000100001010	770	[22]0000101001111001010011
507	[13]1010101010011	595	[16]0000111101010011	683	[19]0000101000101010011	771	[22]0000101001111100001010
508	[13]1011000001010	596	[16]0001000001010011	684	[19]0000101001000001010	772	[22]0000101001111101010011
509	[13]1011001010011	597	[16]0001100001010011	685	[19]0000101001001010011	773	[22]0000101010000001010011
510	[13]1011100001010	598	[16]0010000001010011	686	[19]0000101001100000001	774	[22]0000101010100001010011
511	[13]1011101010011	599	[16]0010100001010011	687	[19]0000101001100001010	775	[22]0000101011000001010011
512	[13]1100000001010	600	[16]0011000001010011	688	[19]0000101001100010011	776	[22]0000101011100001010011
513	[13]1100001010011	601	[16]0011100001010011	689	[19]0000101001100100001	777	[22]0101001100000001010011
514	[13]1100100001010	602	[16]0100000001010011	690	[19]0000101001100101010	778	[22]0101001100100001010011
515	[13]1100101010011	603	[16]0100100001010011	691	[19]0000101001100110011	779	[22]0101001101000001010011
516	[13]1101000001010	604	[16]0101000000001010	692	[19]0000101001101000001	780	[22]0101001101100001010011
517	[13]1101001010011	605	[16]0101000001010011	693	[19]0000101001101001010	781	[22]0101001110000001010011
518	[13]1101100001010	606	[16]0101000100001010	694	[19]0000101001101010011	782	[22]0101001110100001010011
519	[13]1101101010011	607	[16]0101000101010011	695	[19]0000101001101100001	783	[22]0101001111000001010011
520	[13]1110000001010	608	[16]0101001000001010	696	[19]0000101001101101010	784	[22]0101001111100001010011
521	[13]1110001010011	609	[16]0101001001010011	697	[19]0000101001101110011	785	[25]0000101001100000001010011
522	[13]1110100001010	610	[16]0101001100000001	698	[19]0000101001110000001	786	[25]0000101001100100001010011
523	[13]1110101010011	611	[16]0101001100001010	699	[19]0000101001110001010	787	[25]0000101001101000001010011
524	[13]1111000001010	612	[16]0101001100010011	700	[19]0000101001110010011	788	[25]0000101001101100001010011
525	[13]1111001010011	613	[16]0101001100100001	701	[19]0000101001110100001	789	[25]0000101001110000001010011
526	[13]1111100001010	614	[16]0101001100101010	702	[19]0000101001110101010	790	[25]0000101001110100001010011
527	[13]1111101010011	615	[16]0101001100110011	703	[19]0000101001110110011	791	[25]0000101001111000001010011
528	[16]0000000001010011	616	[16]0101001101000001	704	[19]0000101001111000001	792	[25]0000101001111100001010011

Table 4.2: Examples of the exponential length construction for $0 < m < n < 2m$

$\lvert\Sigma\rvert$	m	n	τ	$\lvert\tau\rvert$	w	$\lvert w\rvert$
2	2	3	001	3	τ	3
2	3	4	0001	4	$\tau 0^3 \tau$	11
2	3	5	00001010011	11	$\tau 0^3 \tau$	25
2	4	5	00001	5	$\tau 0^4 \tau 0^4 \tau$	23
2	4	7	0000001001000110100010101100111	31	$\tau 0^4 \tau 0^4 \tau$	101
2	5	6	000001	6	$\tau 0^5 \tau 0^5 \tau 0^5 \tau$	39
2	5	7	00000010001000011	17	$\tau 0^5 \tau 0^5 \tau 0^5 \tau$	83
2	5	8	000000010001000011001000010100110001111	38	$\tau 0^5 \tau 0^5 \tau 0^5 \tau$	167
2	5	9	000000001000100001100100001010011000111-0100001001010100101101100011010111001111	79	$\tau 0^5 \tau 0^5 \tau 0^5 \tau$	331
2	6	7	0000001	7	$\tau 0^6 \tau 0^6 \tau 0^6 \tau 0^6 \tau$	59
2	7	8	00000001	8	$\tau 0^7 \tau 0^7 \tau 0^7 \tau 0^7 \tau 0^7 \tau$	83
2	7	9	000000001000000100000011	23	$\tau 0^7 \tau 0^7 \tau 0^7 \tau 0^7 \tau 0^7 \tau$	173
2	8	9	000000001	9	$\tau 0^8 \tau 0^8 \tau 0^8 \tau 0^8 \tau 0^8 \tau 0^8 \tau$	111
3	2	3	00102	5	τ	5
3	3	4	0001002	7	$\tau 0^3 \tau$	17
3	3	5	00001002010011012020021022	26	$\tau 0^3 \tau$	55
3	4	5	000010002	9	$\tau 0^4 \tau 0^4 \tau$	35
4	2	3	0010203	7	τ	7
4	3	4	0001002003	10	$\tau 0^3 \tau$	23
4	3	5	00001002003010011012013-02002102202303003103203 3	47	$\tau 0^3 \tau$	97
4	4	5	0000100020003	13	$\tau 0^4 \tau 0^4 \tau$	47

Assume there are two palindromes a_1, a_2 in the same path $a_1 w_0 b_1 w_1 \cdots b_k w_k a_2$. Since $S^R \subseteq S$, the words $w_0^R, w_1^R, \ldots, w_k^R$ are also in S. Hence $a_2^R w_k^R b_k^R \cdots w_1^R b_1^R a_1^R$ is also a path. But a_1, a_2 are palindromes, so $a_1^R = a_1$ and $a_2^R = a_2$. Then the concatenation of the two paths gives a cycle in $G_S^{(m,n)}$, which contradicts the fact that S^* is co-finite. Therefore, the longest path in $G_S^{(m,n)}$ cannot encounter every vertex, so by Theorem 2.5.3, the longest words not in S^* are of length less than $g(m,l) = ml - m - l$, where $l = m\lvert\Sigma\rvert^{n-m} + n - m$

We can construct a set of words S such that the longest path in $G_S^{(m,n)}$ encounters every non-palindrome vertex and 0^{n-m}. We start from $w_0 = 0^{n-m}$ and T being empty. First we pick any non-palindrome vertex w_1 of length $n - m$, and add $w_0 0^{2m-n} w_1$ and $w_1^R 0^{2m-n} w_0^R$ to T. Then we pick any non-palindrome w_2 of length $n - m$ such that neither w_2 nor w_2^R was considered in previous steps, and add $w_1 0^{2m-n} w_2$ and $w_2^R 0^{2m-n} w_1^R$ to T. Continue this procedure until no word can be added. Then we claim $\Sigma^m \cup \Sigma^n \setminus T$ is the required set. To see this, first, no non-palindrome vertex is left, or the procedure will not stop. Since at each step the vertex w is picked only if it was not considered, it remains to check w^R was not considered in any previous step. Otherwise, for some previous chosen vertex u, we have $u = w^R$. Then $w = u^R$ was already considered, a contradiction. So the longest path in $G_S^{(m,n)}$ encounters each non-palindrome vertex exactly once, and the longest words not in S^* are of length $g(m,l') = \Theta(\lvert\Sigma\rvert^{n-m})$, where $l' = m(\lvert\Sigma\rvert^{n-m} - \lvert\Sigma\rvert^{\lceil (n-m)/2 \rceil}) + n - m$. □

Shallit [154] considered examples with other input forms, such as NFAs, regular expressions, and PDAs, accepting the example in Theorem 4.1.1 and variations. The following constructions are mainly based on his ideas.

There is an NFA with $O(n^2)$ states accepting the example S in Theorem 4.1.1

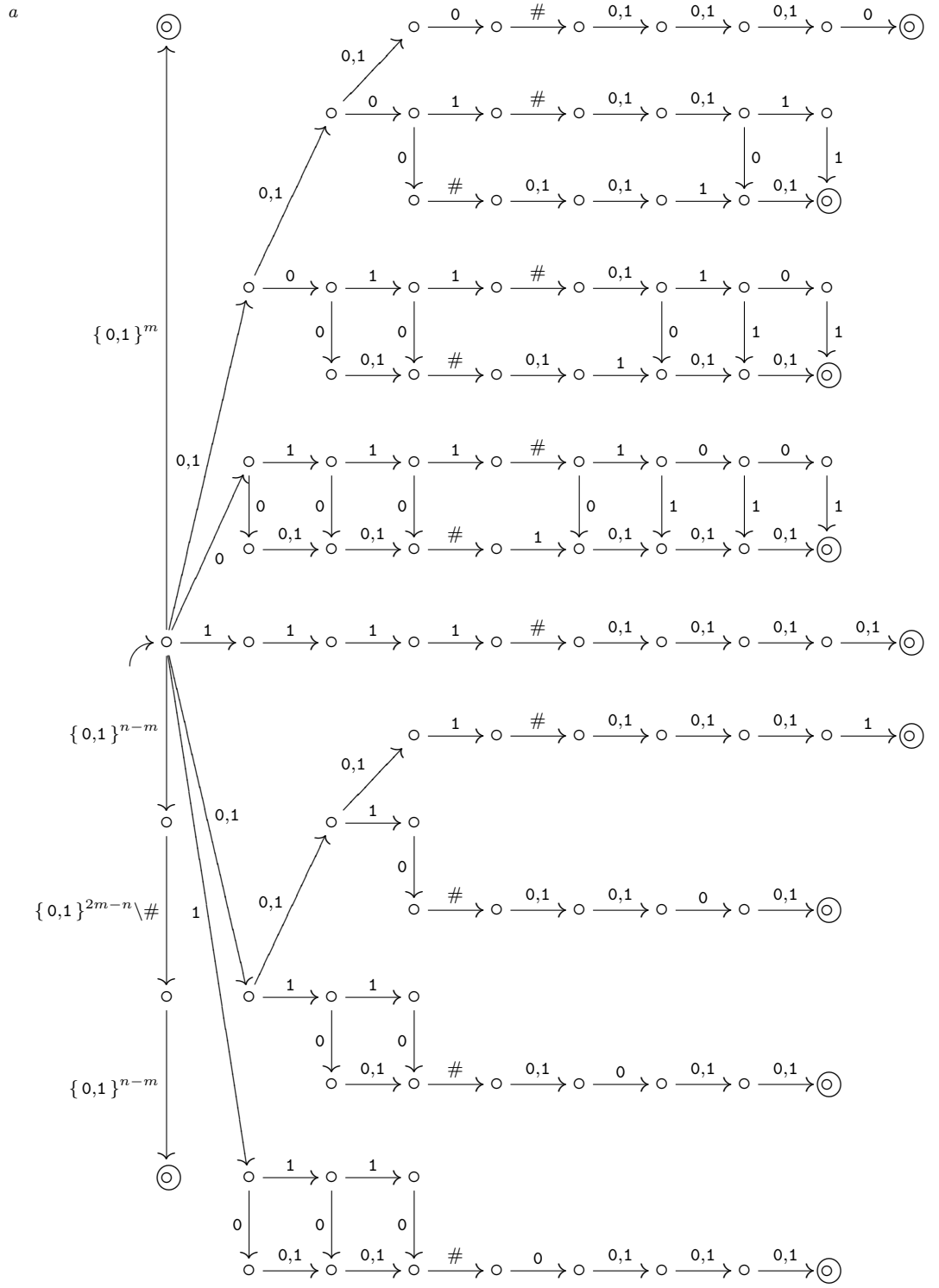
over the alphabet Σ. For two integers m, n with $0 < m < n < 2m$ and $\gcd(m, n) = 1$, the set S contains all words of lengths m and n except words of the form $(i)\mathtt{0}^{2m-n}(i+1)$ for $0 \leq i \leq |\Sigma|^{n-m} - 1$. For brevity, write $\#$ to represent the middle zeros $\mathtt{0}^{2m-n}$. The NFA M is composed of five parts, an NFA accepting all words of length m and words of length n not of the form $u\#u'$ and G_a, A_a, C_a, T for letters $a \in \Sigma$. Then M is constructed by combining the initial states of the five. Each of the latter four parts handles a particular type of the word $u\#v$ such that $[v]_{|\Sigma|} \neq [u]_{|\Sigma|} + 1$. Let $\mathtt{z}$ be the lexicographically last letter in Σ, and let Σ_a denote $\Sigma \setminus \{ a \}$. The NFA G_a accepts words of the form $\Sigma^{n-m-i-1} a L_i \# \Sigma^{n-m-i-1} \Sigma_a \Sigma^i$, where L_i is the set of all words of length i that contain at least one letter in $\Sigma_{\mathtt{z}}$, for $1 \leq i \leq n-m-1$. The NFA A_a, $a \neq \mathtt{z}$, accepts words of the form $\Sigma^{n-m-1} a \# \Sigma^{n-m-1} \Sigma_b$, where $[b]_{|\Sigma|} = [a]_{|\Sigma|} + 1$. The NFA C_a, $a \neq \mathtt{z}$, accepts words of the form $\Sigma^{n-m-i-1} a \mathtt{z}^i \# \Sigma^{n-m-i-1} (\Sigma^{i+1} \setminus b\mathtt{0}^i)$, where $[b]_{|\Sigma|} = [a]_{|\Sigma|} + 1$, for $1 \leq i \leq n-m-1$. The NFA T accepts the language $\mathtt{z}^{n-m} \# \Sigma^{n-m}$. One can verify that all words accepted by the four NFAs are of length n.

First of all, none of the words accepted by any of the NFAs is of the form $(i)\mathtt{0}^{2m-n}(i+1)$. Let $u\#v$ be a word of n, where u, v are of length $n-m$. If G_a accepts $u\#v$, then $u = waw' \in \Sigma^{n-m} a L_i$, for $w' \in L_i$. Since w' contains a letter in $\Sigma_{\mathtt{z}}$, u plus 1 will not change the letter a even when a carry occurs. But u and v are different at the position of a, so $[v] \neq [u] + 1$. If A_a accepts $u\#v$, then $u = wa$ for $w \in \Sigma^{n-m-1}$. Since $a \neq \mathtt{z}$, the addition of u and 1 will not carry, and equals wb, where b is the letter that follows a. The last letter in v is not b, so $[v] \neq [u] + 1$. If $u = wa\mathtt{z}^i$ for $w \in \Sigma n - m - i$, then u plus 1 will carry right to the position of a, and result in $wb\mathtt{0}^i$, where b is the letter that follows a. Now $b\mathtt{0}^i$ is not a suffix of the word v and thus $[v] \neq [u] + 1$. Finally, the word $\mathtt{z}^{n-m}$ represents the largest integer with length $n-m$, and thus none of the words accepted by T is of the form $(i)\mathtt{0}^{2m-n}(i+1)$.

Now we prove that if $u\#v$ is not of the form $(i)\mathtt{0}^{2m-n}(i+1)$, then $u\#v$ is accepted by one of the four NFAs. If $u = z^{n-m}$, then T accepts $u\#v$. Otherwise, assume $u = xy$, such that $[xy] + 1 = [xz]$ and the first letters of y, z are different. Then y contains at least one letter that is not $\mathtt{z}$, and $v \neq xz$. The mismatching can happen either in x or in z. If there is a letter in x which does not match v, then G_a accepts $u\#v$. If the mismatching is in z, then either A_a or C_a accepts $u\#v$, depending on whether the calculation $[xy] + 1$ in base $|\Sigma|$ carries.

There is an NFA of $m+1$ states accepting words of length m, and an NFA of $n+1$ states accepting words of length n not in the form of $w\#w'$, where w's are of length $n-m$. Each of the G_a with i has $(n-m-i) + 2i + (2m-n) + (n-m) \leq 2n$ states. Each of the A_a has $(2n-2m) + (2m-n) + 1 \leq n+1$ states. Each of the C_a with i has $(n-m) + 1 + (2m-n) + (n-m-i) + 2i \leq 2n$ states. T has $(2n-2m) + (2m-n) + 1 \leq n+1$ states. So the total number of states is $< 4\,|\Sigma|\,n^2 + |\Sigma|\,(n+1) = O\,(|\Sigma|\,n^2)$. For $|\Sigma| = 2, n-m = 4$, the NFA is illustrated in Table 4.3, where the number for each state is omitted, and repeated structures are combined.

Table 4.3: NFA accepting $\Sigma^m \cup \Sigma^n \setminus T(m,n)$



[a] Here ◎ denotes final states.

Now we consider regular expressions. By the same construction in the NFAs, we can effectively convert them into a regular expression of length $O(n^2 \log n)$, which is also composed of five parts. Let Σ represent the regular expression $\sum_{a\in\Sigma} a$, $\# = \mathtt{0}^{2m-n}$, and let L^k abbreviate the regular expression of k-times concatenation of L (when we count the length of the regular expression, $\operatorname{alph}(L^k) = k \cdot \operatorname{alph}(L)$). We define $\operatorname{omit}(w)$ to be the regular expression for the language $\Sigma^n \setminus \{\, w \,\}$, where w is a word of length n over the alphabet Σ. It can be recursively constructed using divide-and-conquer method as follows:

$$\operatorname{omit}(a_1 a_2 \cdots a_n) = \Sigma^{\lfloor n/2 \rfloor} \operatorname{omit}(a_{\lfloor n/2 \rfloor + 1} \cdots a_n) + \operatorname{omit}(a_1 \cdots a_{\lfloor n/2 \rfloor}) a_{\lfloor n/2 \rfloor + 1} \cdots a_n, \tag{4.12}$$

and thus has $O(n \log n)$ alphabetic symbols [41]. Let $R_1 = R_1' + R_2''$, where $R_1' = \Sigma^m$ is of length $m\,|\Sigma|$ and contains all words of length m, and $R_2'' = \Sigma^{n-m} \operatorname{omit}(\#)\Sigma^{n-m}$ is of length $O(n + m \log m)$. Let R_2 be a regular expression for G_a, that is $\sum_{1\le i\le n-m-1} \sum_{a\in\Sigma} \big(\Sigma^{n-m-i-1} a L_i \# \Sigma^{n-m-i-1} \Sigma_a \Sigma^i\big)$, where L_i is the set of all words of length i that contain at least one of the non-$\mathtt{z}$ letters and can be constructed recursively as follows:

$$L_i = L_{\lfloor i/2 \rfloor} \Sigma^{\lceil i/2 \rceil} + \Sigma^{\lfloor i/2 \rfloor} L_{\lceil i/2 \rceil}. \tag{4.13}$$

Thus L_i has length $O(i \log i)$. Hence the total length of R_2 is $O(|\Sigma|\, n^2 \log n)$. Let R_3 be a regular expression for A_a, that is $\sum_{a\in\Sigma, a\ne\mathtt{z}} \Sigma^{n-m-1} a \# \Sigma^{n-m-1} b$, where b is the letter that follows a. Then the length of R_3 is $\le 2\,|\Sigma|^2\, n$. Let R_4 be a regular expression for C_a, that is $\sum_{1\le i\le n-m-1} \sum_{a\in\Sigma, a\ne\mathtt{z}} \big(\Sigma^{n-m-i-1} a \mathtt{z}^i \# \Sigma^{n-m-i-1} \operatorname{omit}(b\mathtt{0}^i)\big)$, where b is the letter that follows a. Since $\operatorname{omit}(b\mathtt{0}^i)\big)$ has length $O(i \log i)$, then the total length of R_4 is $O(|\Sigma|\, n^2 \log n)$. Finally, R_5 is just $\mathtt{z}^{n-m} \# \Sigma^{n-m}$, which is of length $n - m + m\,|\Sigma|$. Therefore, there is a regular expression $R_1 + R_2 + R_3 + R_4 + R_5$ of length $O(\nu^2 \log \nu)$ specifying the example in Theorem 4.1.1, where $\nu = n$ is the length of the longest words in the basis S.

For example, $T(5, 9) = \{\, \mathtt{000000001}, \mathtt{000100002}, \mathtt{000200010}, \ldots, \mathtt{222102222} \,\}$, for $\Sigma = \{\, \mathtt{0}, \mathtt{1}, \mathtt{2} \,\}$. The regular expression for $S = \Sigma^5 \cup \Sigma^9 \setminus T(5, 9)$ is

$$\begin{gathered}
\Sigma^2\mathtt{0}(\mathtt{0}+\mathtt{1})\#\Sigma^2(\mathtt{1}+\mathtt{2})\Sigma+\Sigma\mathtt{0}\big((\mathtt{0}+\mathtt{1})\Sigma+\Sigma(\mathtt{0}+\mathtt{1})\big)\#\Sigma(\mathtt{1}+\mathtt{2})\Sigma^2+\mathtt{0}\big(\big((\mathtt{0}+\mathtt{1})\Sigma+\Sigma(\mathtt{0}+\mathtt{1})\big)\Sigma+\Sigma^2(\mathtt{0}+\mathtt{1})\big)\#(\mathtt{1}+\mathtt{2})\Sigma^3+\\
\Sigma^2\mathtt{1}(\mathtt{0}+\mathtt{1})\#\Sigma^2(\mathtt{0}+\mathtt{2})\Sigma+\Sigma\mathtt{1}\big((\mathtt{0}+\mathtt{1})\Sigma+\Sigma(\mathtt{0}+\mathtt{1})\big)\#\Sigma(\mathtt{0}+\mathtt{2})\Sigma^2+\mathtt{1}\big(\big((\mathtt{0}+\mathtt{1})\Sigma+\Sigma(\mathtt{0}+\mathtt{1})\big)\Sigma+\Sigma^2(\mathtt{0}+\mathtt{1})\big)\#(\mathtt{0}+\mathtt{2})\Sigma^3+\\
\Sigma^2\mathtt{2}(\mathtt{0}+\mathtt{1})\#\Sigma^2(\mathtt{0}+\mathtt{1})\Sigma+\Sigma\mathtt{2}\big((\mathtt{0}+\mathtt{1})\Sigma+\Sigma(\mathtt{0}+\mathtt{1})\big)\#\Sigma(\mathtt{0}+\mathtt{1})\Sigma^2+\mathtt{2}\big(\big((\mathtt{0}+\mathtt{1})\Sigma+\Sigma(\mathtt{0}+\mathtt{1})\big)\Sigma+\Sigma^2(\mathtt{0}+\mathtt{1})\big)\#(\mathtt{0}+\mathtt{1})\Sigma^3+\\
\Sigma^3\mathtt{0}\#\Sigma^3(\mathtt{0}+\mathtt{2})+\Sigma^3\mathtt{1}\#\Sigma^3(\mathtt{0}+\mathtt{1})+\Sigma^3\mathtt{2}\#\Sigma^3(\mathtt{1}+\mathtt{2})+\mathtt{2222}\#\Sigma^4+\\
\mathtt{0222}\#\big(\big((\mathtt{0}+\mathtt{2})\Sigma+\mathtt{1}(\mathtt{1}+\mathtt{2})\big)\Sigma+\mathtt{10}\big((\mathtt{1}+\mathtt{2})\Sigma+\mathtt{0}(\mathtt{1}+\mathtt{2})\big)\big)+\Sigma\mathtt{022}\#\Sigma\big(\big((\mathtt{0}+\mathtt{2})\Sigma+\mathtt{1}(\mathtt{1}+\mathtt{2})\big)\Sigma+\mathtt{10}(\mathtt{1}+\mathtt{2})\big)+\\
\Sigma^2\mathtt{02}\#\Sigma^2\big((\mathtt{0}+\mathtt{2})\Sigma+\mathtt{1}(\mathtt{1}+\mathtt{2})\big)+\mathtt{1222}\#\big(\big((\mathtt{0}+\mathtt{1})\Sigma+\mathtt{2}(\mathtt{1}+\mathtt{2})\big)\Sigma+\mathtt{20}\big((\mathtt{1}+\mathtt{2})\Sigma+\mathtt{0}(\mathtt{1}+\mathtt{2})\big)\big)+\\
\Sigma\mathtt{122}\#\Sigma\big(\big((\mathtt{0}+\mathtt{1})\Sigma+\mathtt{2}(\mathtt{1}+\mathtt{2})\big)\Sigma+\mathtt{20}(\mathtt{1}+\mathtt{2})\big)+\Sigma^2\mathtt{02}\#\Sigma^2\big((\mathtt{0}+\mathtt{1})\Sigma+\mathtt{2}(\mathtt{1}+\mathtt{2})\big)+\Sigma^5+\Sigma^4(\mathtt{1}+\mathtt{2})\Sigma^4,
\end{gathered}$$

where $\Sigma = (\mathtt{0}+\mathtt{1}+\mathtt{2})$ and $\# = \mathtt{0}$. The set S contains 19846 words.

Now we consider DPDAs. A DPDA of $O(n)$ states can be constructed to accept a variation on the example in Theorem 4.1.1. The idea is to construct a DPDA M accepting all words of length n except those of the form $(i)\mathtt{0}^{2m-n}(i+1)^R$ for $0 \le i \le |\Sigma|^{n-m} - 2$. The machine M simply reads the first $n - m$ symbols in

the input, and stores the result in the stack. Then M checks that the following $2m-n$ symbols are zeros. Finally, M does an addition of the word in the stack with 1 in base $|\Sigma|$ and matches the last $n-m$ symbols of the input. If these match, then M rejects. If any mismatching appears at any step, M continues to read input symbols and verifies that the total number of input symbols is n. If it is, M accepts; otherwise, M rejects. One can modify M such that when M reads m symbols and there is no other symbol left on the tape, then M also accepts. So a DPDA accepting

$$L = \Sigma^m \ \cup \ \Sigma^n \setminus \left\{ (i)\mathtt{0}^{2m-n}(i+1)^R : 0 \leq i \leq |\Sigma|^{n-m} - 2 \right\}. \tag{4.14}$$

can be constructed. The DPDA needs to store the information on the number of symbols it had read, so M needs $O(n)$ states.

For $|\Sigma|$ even and $n-m$ odd, Shallit proved in our paper [102] with Lubiw and Shallit that starting from $x_0 = \mathtt{0}^{n-m}$ and applying the recursion $x_{k+1} = ([x_k]+1)^R$, all words of length $n-m$ will be generated and $\mathtt{z}^{n-m}$ is the last one generated, where $\mathtt{z}$ is the lexicographically last letter in Σ. This means each word of length $n-m$ except $\mathtt{z}^{n-m}$ appears as a prefix of a word not in L exactly once and each word of length $n-m$ except $\mathtt{0}^{n-m}$ appears as a suffix of a word not in L exactly once. Then by Theorem 4.1.3, L^* is co-finite and the length of the longest words not in L^* is $g(m,l) = ml - m - l$, where $l = m\,|\Sigma|^{n-m} + n - m$.

4.1.2 Examples of the 2FPFM with $0 < 2m < n$

In this subsection, I will show how to construct examples achieving exponential longest omitted words in the general case without the constraint $n < 2m$. By the equivalence between finding the longest words not in S^* and the words τ not in S^* in Theorem 2.5.3, in order to construct an example for the 2FPFM with exponential length of longest omitted words $g(m,l) = ml - m - l$, where $l = m\,|\Sigma|^{n-m} + n - m$, it is essential to find a word τ of length $l - m$. When $0 < m < n < 2m$, as we saw in the last subsection, such a τ can be effectively found. In order to do this for $0 < 2m < n$, we again use the word graph.

By Theorem 2.6.6, the longest words not in S^* are related to the longest paths in the word graph. The labeling of a longest path in $G_S^{(m,n)}$ gives the word τ that can be used to describe the longest omitted words. Then it remains to find a set of words S with the longest path of length $|\Sigma|^{n-m} - 1$, which is a Hamilton path in the generalized de Bruijn graph $\Gamma(m,n)$. For each Hamilton path in $\Gamma(m,n)$, let T be the set of arcs in the path, and $S = \Sigma^m \ \cup \ \Sigma^n \setminus T$. Then S^* is co-finite and the word graph $G_S^{(m,n)}$ is exactly the spanning subgraph induced by the Hamilton path, and thus S can achieve the upper bound in the 2FPFM.

We know that a de Bruijn word, or a Hamilton cycle in the usual de Bruijn graph, can be constructed. I will show how to find a Hamilton cycle in the generalized de Bruijn graph by finding two de Bruijn words of particular lengths over particular alphabets.

When $m = 1$, $\Gamma(1, n)$ is the usual de Bruijn graph, and a Hamilton cycle can be found accordingly. Furthermore, if m divides n, then a Hamilton cycle in $\Gamma(m, n)$ can be obtained as follows. Let Σ be the alphabet. Now we consider the de Bruijn graph $G'(1, n/m)$ on the alphabet Δ, where $|\,\Delta\,| = |\Sigma|^m$. The vertices in $G'(1, n/m)$ are $\Delta^{n/m-1}$ and the arcs are $\Delta^{n/m}$. Let $v_0 a_1 v_1 a_2 \dots a_{|\,\Delta\,|} v_0$ be a Hamilton cycle in $G'(1, n/m)$. For each of the vertices and arcs in the cycle, treat them as base-$|\,\Delta\,|$ expansion of integers and re-encode them in base $|\Sigma|$, possibly with leading zeros as follows:

$$_{(n-m)}([v_0]_\Delta)_\Sigma \cdot {}_{(n)}([a_1]_\Delta)_\Sigma \cdot {}_{(n-m)}([v_1]_\Delta)_\Sigma \cdot \dots \cdot {}_{(n)}([a_{|\,\Delta\,|}]_\Delta)_\Sigma \cdot {}_{(n-m)}([v_0]_\Delta)_\Sigma. \quad (4.15)$$

Then one can verify that the result gives a Hamilton cycle in $\Gamma(m, n)$. For example, to find a Hamilton cycle in $\Gamma(2, 4)$ on the alphabet $\{\,\mathtt{0}, \mathtt{1}\,\}$, first we find a Hamilton cycle in $\Gamma(1, 2)$ on the alphabet $\{\,\mathtt{0}, \mathtt{1}, \mathtt{2}, \mathtt{3}\,\}$ given by $\mathtt{0}, \mathtt{01}, \mathtt{1}, \mathtt{12}, \mathtt{2}, \mathtt{23}, \mathtt{3}, \mathtt{30}, \mathtt{0}$. Then a Hamilton cycle in $\Gamma(2, 4)$ is given by $\mathtt{00}, \mathtt{0001}, \mathtt{01}, \mathtt{0110}, \mathtt{10}, \mathtt{1011}, \mathtt{11}, \mathtt{1100}, \mathtt{00}$.

When $0 < m < n < 2m$, a Hamilton cycle in $\Gamma(m, n)$ can be found by first finding a Hamilton cycle in $\Gamma(n - m, 2n - 2m)$ and then adding extra letters in the middle. More precisely, let $v_0, v_0v_1, v_1, v_1v_2, v_2, \dots, v_{|\Sigma|^{n-m}-1}, v_{|\Sigma|^{n-m}-1}v_0, v_0$ be a Hamilton cycle in $\Gamma(n - m, 2n - 2m)$. Then one can verify that

$$v_0, v_0\mathtt{0}^{2m-n}v_1, v_1, v_1\mathtt{0}^{2m-n}v_2, v_2, \dots, v_{|\Sigma|^{n-m}-1}, v_{|\Sigma|^{n-m}-1}\mathtt{0}^{2m-n}v_0, v_0 \quad (4.16)$$

is a Hamilton cycle in $\Gamma(m, n)$. Here the padded factor $\mathtt{0}^{2m-n}$ can be any word of length $2m - n$, not even necessarily the same. Then there exists a Hamilton cycle in $\Gamma(m, n)$. The construction here is precisely the same as I used to construct the examples for $0 < m < n < 2m$ in the previous subsection.

Now we are ready to see the construction for a Hamilton cycle in a general generalized de Bruijn graph $\Gamma(m, n)$ with $0 < m < n$. Let $n = km + r$, where $0 \le r < m$. If either $\gcd(m, n) = m$ or $k = 1$ holds, we already saw how to construct a Hamilton cycle in $\Gamma(m, n)$. So we assume $m > 1$, $\gcd(m, n) \neq m$, and $k > 1$. Let $m_1 = r, n_1 = (k + 1)r, N_1 = |\Sigma|^{n_1-m_1}$. Then m_1 divides n_1, and we can construct a Hamilton cycle of length N_1. Suppose the labeling of the arcs in one Hamilton cycle are $u_1, u_2, \cdots, u_{N_1}$, starting from the vertex $\mathtt{0}^{n_1-m_1}$. Define $u_j = u_{N_1+j}$ for $j \le 0$, and $u'_i = u_j$, where $j = 1 + \big(i \bmod (N_1 - 1)\big)$. Then by comparing lengths we have that the i^{th} vertex in the cycle is $u_{i-k} \cdots u_{i-1}$, and thus $u_{N_1} = u_{N_1-1} = \cdots = u_{N_1-k+1} = \mathtt{0}^{m_1}$. Let $m_2 = m - r, n_2 = k(m - r), N_2 = |\Sigma|^{n_2-m_2}$. Similarly, suppose the labeling of arcs in a Hamilton cycle are $v_1, v_2, \cdots, v_{N_2}$, starting from the vertex $\mathtt{0}^{n_2-m_2}$. Define $v_j = v_{N_2+j}$ for $j \le 0$, and $v'_i = v_j$, where $j = 1+(i-1 \bmod N_2)$. Then the i^{th} vertex in the cycle is $v_{i-k+1} \cdots v_{i-1}$, and $v_{N_2} = v_{N_2-1} = \cdots = v_{N_2-k+2} = \mathtt{0}^{m_2}$. Consider the following two cycles in $\Gamma(m, n)$. Starting from vertex $\mathtt{0}^{n-m}$, one cycle is labeled by

$$v_1\mathtt{0}^{m_1}, v_2\mathtt{0}^{m_1}, \dots, v_{N_2-1}\mathtt{0}^{m_1}, v_{N_2}\mathtt{0}^{m_1}. \quad (4.17)$$

Then one can check this cycle with labels in (4.17) visits every vertex in

$$(\mathtt{0}^{m_1}\Sigma^{m_2})^{k-1}\mathtt{0}^{m_1} \quad (4.18)$$

exactly once. Starting from the vertex $\mathtt{0}^{(k-1)m}u_1$, the other cycle is labeled by

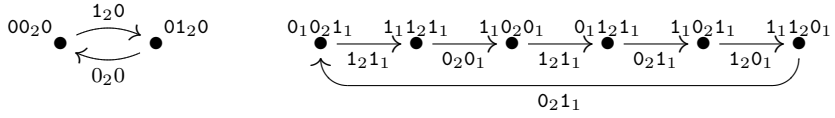
$$v'_1u'_1, v'_2u'_2, \cdots, v'_{(N_1-1)N_2-1}u'_{(N_1-1)N_2-1}, v'_{(N_1-1)N_2}u'_{(N_1-1)N_2}. \tag{4.19}$$

Since $\gcd(N_2, N_1-1) = 1$, this cycle with labels in (4.19) visits every vertex in $(\Sigma^{m_1}\Sigma^{m_2})^{k-1}\Sigma^{m_1} \setminus (\mathtt{0}^{m_1}\Sigma^{m_2})^{k-1}\mathtt{0}^{m_1}$ exactly once. Linking the above two cycles together by modifying two arcs, the labeling of the arcs in the new cycle are given below:

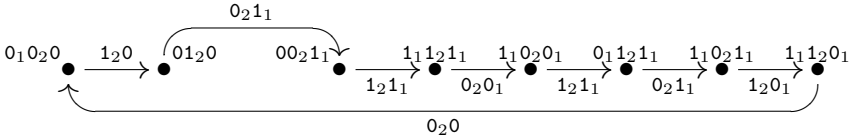
$$v_1\mathtt{0}^{m_1}, v_2\mathtt{0}^{m_1}, \cdots, v_{N_2-1}\mathtt{0}^{m_1}, v_{N_2}u'_{(N_1-1)N_2},$$
$$v'_1u'_1, v'_2u'_2, \cdots, v'_{(N_1-1)N_2-1}u'_{(N_1-1)N_2-1}, v'_{(N_1-1)N_2}\mathtt{0}^{m_1}.$$

Since the set of vertices in the two cycles are disjoint, and the length of the new cycle is $N_2 + (N_1-1)N_2 = N_1N_2 = |\Sigma|^{n_1-m_1+n_2-m_2} = |\Sigma|^{km-m+r} = |\Sigma|^{n-m}$, the new cycle in $\Gamma(m,n)$ is a Hamilton cycle.

Example 4.1.5. For $m = 2, n = 5$, we construct a τ of length $2|\Sigma|^3 + 1 = 17$ and a set S of words of lengths 2 and 5 such that the longest words not in S^* are of length $g(2,19) = 17$. Since $n = km + r$, where $k = 2, r = 1$, we first find Hamilton cycles in $G\big(r,(k+1)r\big) = \Gamma(1,3)$ and in $G\big(m-r,k(m-r)\big) = \Gamma(1,2)$. One Hamilton cycle in $\Gamma(1,3)$ is $\mathtt{00},\mathtt{001},\mathtt{01},\mathtt{011},\mathtt{11},\mathtt{110},\mathtt{10},\mathtt{100},\mathtt{00}$, which starts from $\mathtt{00}$ and is labeled by $\mathtt{1},\mathtt{1},\mathtt{0},\mathtt{0}$. Omit the first label $\mathtt{1}$. One Hamilton cycle in $\Gamma(1,2)$ is given by $\mathtt{0},\mathtt{01},\mathtt{1},\mathtt{10},\mathtt{0}$, which starts from $\mathtt{0}$ and is labeled by $\mathtt{1},\mathtt{0}$. Then we construct two cycles in $\Gamma(m,n)$. One starts from $\mathtt{000}$ and is labeled by $\mathtt{1}_2\mathtt{0}, \mathtt{0}_2\mathtt{0}$. The subscript here denotes whether the letter is from the Hamilton cycle in $\Gamma(1,3)$ or from $\Gamma(1,2)$. The other is from $\mathtt{001}$ and is labeled by $\mathtt{1}_2\mathtt{1}_1, \mathtt{0}_2\mathtt{0}_1, \mathtt{1}_2\mathtt{1}_1, \mathtt{0}_2\mathtt{1}_1, \mathtt{1}_2\mathtt{0}_1, \mathtt{0}_2\mathtt{1}_1$.



Now we connect the two cycles. Finally we obtain a Hamilton cycle in $\Gamma(2,5)$, which starts from $\mathtt{000}$ and is labeled by $\mathtt{1}_2\mathtt{0}, \mathtt{0}_2\mathtt{1}_1, \mathtt{1}_2\mathtt{1}_1, \mathtt{0}_2\mathtt{0}_1, \mathtt{1}_2\mathtt{1}_1, \mathtt{0}_2\mathtt{1}_1, \mathtt{1}_2\mathtt{0}_1, \mathtt{0}_2\mathtt{0}$.



So there is a $\tau = \mathtt{00010011100110110}$, and a set of words

$$S = \Sigma^2 \ \cup \ \Sigma^5 \setminus \{\, \mathtt{00010}, \mathtt{01001}, \mathtt{00111}, \mathtt{11100}, \mathtt{10011}, \mathtt{01101}, \mathtt{10110} \,\}, \tag{4.20}$$

such that the longest word not in S^* is τ of length $g(2,19) = 17$.

When $m = 1$, the word τ labeling a Hamilton path in $\Gamma(1,n)$ is essentially a de Bruijn word. For example, $m = 1, n = 4$, then one de Bruijn word is $\tau = \mathtt{0001011100}$, which contains every word of length 3 exactly once:

$$\underline{\mathtt{000}}\,\mathtt{1011100}, \quad \mathtt{0}\,\underline{\mathtt{001}}\,\mathtt{011100}, \quad \mathtt{00}\,\underline{\mathtt{010}}\,\mathtt{11100}, \quad \mathtt{000}\,\underline{\mathtt{101}}\,\mathtt{1100}$$

0001 011 100, 00010 111 00, 000101 110 0, 0001011 100.

For more general cases of τ with m, n, τ contains every word of length $m-n$ exactly once at a particular position in τ. For example, $m = 2, n = 5$, then one τ is $\tau =$ 00010011100110110:

000 10011100110110, 00 010 011100110110, 0001 001 1100110110, 000100 111 00110110,
00010011 100 110110, 0001001110 011 0110, 000100111001 101 10, 00010011100110 110.

The definition of word graph and generalized de Bruijn graph does not require the condition $\gcd(m, n) = 1$. But in order to construct a basis to generate a co-finite language, the condition $\gcd(m, n) = 1$ is required. For the sole purpose of discussion on co-finiteness, $m = 1$ only leads to trivial cases since a language is co-finite when $m = 1$ if and only if the basis contains all letters. But as we saw, when $m = 1$, the discussion on the word graph is not trivial. For small m, n, the lexicographically least τ, the labeling of a Hamilton cycle in $\Gamma(m, n)$ is summarized in Table 4.4. When $\gcd(m, n) = 1$, there is a basis S for each of the τ such that the set of the longest words not in S^* is $(\tau\Sigma^m)^{m-2}\tau$.

Table 4.4: Examples of generalized de Bruijn words τ

$\lvert\Sigma\rvert$	m	n	τ[a]	$\lvert\tau\rvert$
2	1	2	01	2
2	1	3	00110	5
2	1	4	0001011100	10
2	1	5	0000100110101111000	19
2	1	6	000001000110010100111010110111110000	36
2	1	7	000000100001100010100011100100101100110100111101010111011011111100000	69
2	1	8 : 9	(too long, omitted)	134 : 263
2	2	4	00011011	8
2	2	5	00001001011011110	17
2	2	6	000001001000110101100111010111100	34
2	2	7	000000100001000110001010010011001110011010110101010111011110111111000	67
2	2	8 : 9	(too long, omitted)	132 : 261
2	3	6	000001010011100101110111	24
2	3	7	000000100001001100101010010101110011011110111111110	49
2	3	8 : 9	(too long, omitted)	98 : 195
2	4	6	00000100100011	14
2	4	8	000000010010001101000101011001111000100110101011110010111101111	64
2	4	9	(too long, omitted)	129
2	6	8	00000001000010000011	20
2	6	9	000000001000010000011000100000101000110000111	45
3	2	4	000102101112202122	18
3	2	5	00001000200101101020110120210202111121120221221212222220	55
4	2	4	00010203101112132021222330313233	32
4	2	5	(too long, omitted)	129

[a] Those already in Table 4.2 are omitted.

4.2 Doubly-exponential number of words $\notin S^*$

I will give the following examples, in which the number of words not in the generated co-finite language is doubly exponential in ν, the length of the longest words in the basis. Here we also assume the alphabet Σ is $\{0, 1, 2, \dots\}$. The notation ${}_c(n)_k$ and

$[w]_k$ is the same as defined in the last section. Here ${}_c(n)_k$ represents the expansion of the non-negative integer n in base k of length c with possible leading zeros, and $[w]_k$ represents the non-negative integer represented by the word w in base k. For two integers m and n with $0 < m < n < 2m$, define

$$U(m,n) = \left\{ {}_c(i)_k w_c(j)_k : 0 \le i < j \le |\Sigma|^{n-m} - 1, w \in \Sigma^{2m-n}, k = |\Sigma|, c = n - m \right\}. \tag{4.21}$$

For example, over the binary alphabet $\{ \mathtt{0}, \mathtt{1} \}$, we have

$$U(3,5) = \{ \mathtt{00001}, \mathtt{00010}, \mathtt{00011}, \mathtt{01010}, \mathtt{01011}, \mathtt{10011}, \mathtt{00101}, \mathtt{00110}, \mathtt{00111}, \mathtt{01110}, \mathtt{01111}, \mathtt{10111} \}. \tag{4.22}$$

Theorem 4.2.1. *[83, 84] Let m, n be two integers with $0 < m < n < 2m$ and $\gcd(m,n) = 1$, and let $S = \Sigma^m \cup \Sigma^n \setminus U(m,n)$. Then S^* is co-finite, and the number of words not in S^* is $= |\Sigma|^{\Omega(|\Sigma|^{n-m})}$.*

Proof. Consider the word graph $G_S^{(m,n)}$. By the definition of $U(m,n)$, we know that the arcs in $G_S^{(m,n)}$ are words of the form ${}_c(i)_k w_c(j)_k$, for $0 \le i < j \le |\Sigma|^{n-m} - 1, w \in \Sigma^{2m-n}, k = |\Sigma|, c = n - m$, which joins vertex ${}_c(i)_k$ to vertex ${}_c(j)_k$. So following any walk in $G_S^{(m,n)}$, the vertices visited must be strictly increasing in lexicographical order. Then $G_S^{(m,n)}$ does not contain a cycle. By the construction of S, we also have $\Sigma^m \subseteq S$. By Theorem 2.6.6, S^* is co-finite.

Furthermore, for any two vertices u and v such that v is lexicographically strictly greater than u, then there are $|\Sigma|^{2m-n}$ arcs join u to v as uwv for any $w \in \Sigma^{2m-n}$. So the paths in $G_S^{(m,n)}$ are exactly of the form

$$u_1 x_1 u_2 x_2 \cdots u_k x_k u_{k+1}, \tag{4.23}$$

where $u_1, u_2, \ldots, u_{k+1}$ are distinct vertices in lexicographical order from a smaller vertex to a bigger vertex, and $x_i = u_i y_i u_{i+1}$ for $y \in \Sigma^{2m-n}$. Then by Corollary 2.6.9, we have

$$\begin{aligned} V = \Big\{ & (c_1) w_2 (c_2) w_3 \cdots w_l (c_l) : \\ & 0 \le c_1 < c_2 < \cdots < c_l \le |\Sigma|^{n-m} - 1, 1 \le l \le |\Sigma|^{n-m}, w_i \in \Sigma^{2m-n} \Big\} \end{aligned} \tag{4.24}$$

and the set of words not in S^* is

$$\Sigma^* \setminus S^* = \left(\bigcup_{j \notin \langle m,n \rangle} \Sigma^j \right) \bigcup \left(\bigcup_{i=0}^{m-2} (V \Sigma^m)^i V \right), \tag{4.25}$$

where the cardinality of V is

$$|V| = \sum_{i=1}^{|\Sigma|^{n-m}} \binom{|\Sigma|^{n-m}}{i} |\Sigma|^{(i-1)(2m-n)} = \frac{(1 + |\Sigma|^{2m-n})^{|\Sigma|^{n-m}} - 1}{|\Sigma|^{2m-n}} = |\Sigma|^{\Omega(|\Sigma|^{n-m})}. \tag{4.26}$$

Hence the number of words not in S^* is $\ge V = |\Sigma|^{\Omega(|\Sigma|^{n-m})}$. □

Corollary 4.2.2. *Let m, n be two integers with $0 < m < n < 2m, \gcd(m, n) = 1$, and let $S = \Sigma^m \cup \Sigma^n \setminus U(m, n)$. Then S^* is co-finite, and the number of the longest words not in S^* is $= |\Sigma|^{\Omega(|\Sigma|^{n-m})}$.*

Proof. Similarly to the proof of Theorem 4.2.1, one can verify that the length of the longest omitted words is $g(m, l)$ for $l = m\,|\Sigma|^{n-m} + n - m$. In addition, the set of words of length $l - m$ not in S^* is given by

$$V' = \big\{\, (0)w_2(1)w_3 \cdots w_{|\Sigma|^{n-m}}(|\Sigma|^{n-m} - 1) : w_2, w_3, \ldots, w_{|\Sigma|^{n-m}} \in \Sigma^{2m-n} \,\big\}. \quad (4.27)$$

Then by Corollary 2.6.7, the set of the longest words not in S^* is

$$(V'\Sigma^m)^{m-2}V', \quad (4.28)$$

which is of cardinality

$$|\,V'\,|^{m-1}\,|\Sigma|^{m(m-2)} = |\Sigma|^{(2m-n)(m-1)|\Sigma|^{n-m}+(mn-m^2-n)} = |\Sigma|^{\Omega(|\Sigma|^{n-m})}. \quad (4.29)$$

□

By the preceding two proofs, one can also verify that both the total number of symbols in words not in S^* and the total number of symbols in the longest words not in S^* are $= |\Sigma|^{\Omega(|\Sigma|^{n-m})}$.

Example 4.2.3. Let $m = 3$, $n = 5$, $\Sigma = \{\, \mathtt{0}, \mathtt{1} \,\}$. In this case, $S = \Sigma^3 \cup \Sigma^5 \setminus U(3, 5) = \Sigma^3 \cup \{\, \mathtt{00000}, \mathtt{00100}, \mathtt{01000}, \mathtt{01001}, \mathtt{01100}, \mathtt{01101}, \mathtt{10000}, \mathtt{10001}, \mathtt{10010}, \mathtt{10100}, \mathtt{10101}, \mathtt{10110}, \mathtt{11000}, \mathtt{11001}, \mathtt{11010}, \mathtt{11011}, \mathtt{11100}, \mathtt{11101}, \mathtt{11110}, \mathtt{11111} \,\}$. Then S^* is co-finite, and the set $\Sigma^* \setminus S^*$ is of cardinality 11562. In this case, the cardinality of V is $40 = \big((1 + 2)^4 - 1\big)/2$, the cardinality of V' is $8 = 2^3$, and the number of longest omitted words is $512 = 2^{2\cdot 4+1}$. The total number of symbols is 200638 for all omitted words and 12800 for the longest omitted words.

Theorem 4.2.4. *Let $v_1, v_2, \ldots, v_{|\Sigma|^{n-m}}$ be any permutation of all distinct words of length $n - m$ over the alphabet Σ, and let $S = \Sigma^m \cup \Sigma^n \setminus U$, where*

$$U = \big\{\, v_i w v_j : 1 \le i < j \le |\Sigma|^{n-m}, w \in \Sigma^{2m-n} \,\big\}. \quad (4.30)$$

Then S^* is co-finite, and the number of the longest words not in S^* is $= |\Sigma|^{\Omega(|\Sigma|^{n-m})}$.

Using v_i instead of ${}_{n-m}(i)_{|\Sigma|}$ for $0 \le i \le |\Sigma|^{n-m} - 1$, the proof in Theorem 4.2.1 is also valid for Theorem 4.2.4.

Now, we consider a variation on the FPFM, where the basis is of the form $S \cup S^R$. Let $w_0 = \mathtt{0}^{n-m}, w_1, w_2, \ldots, w_k$ be the vertices defined in the proof of Proposition 4.1.4, and $U' = \big\{\, w_i^R w w_j, w_i w w_j : 0 \le i < j \le k, w \in \Sigma^{2m-n} \,\big\}$, where $k = (|\Sigma|^{n-m} - |\Sigma|^{\lceil (n-m)/2 \rceil})/2$. Then the basis $S = \Sigma^m \cup \Sigma^n \setminus (U' \cup U'^R)$ generates a co-finite language, and one can verify that the number of the longest words not in S^* is also $= |\Sigma|^{\Omega(|\Sigma|^{n-m})}$.

The examples given in Theorem 4.2.1 with $0 < m < n < 2m$ have the maximum number of omitted words among all bases S of two lengths m, n such that S^* is co-finite. To see this, note that any basis S of lengths m, n such that S^* is co-finite can be reduced to a basis of the form in Proposition 4.2.4. Since reducing the basis S will not decrease the number of words not in S^*, the number of omitted words in the original S^* is less than that of the reduced one. It remains to show that the reduction can be done. First, we construct the word graph $G_S^{(m,n)}$, which is a spanning subgraph of $\Gamma(m, n)$. The language S^* is co-finite, so $G_S^{(m,n)}$ has no cycle. There is an order of the vertices in $G_S^{(m,n)}$ such that every arc in $G_S^{(m,n)}$ joins a smaller vertex to a bigger vertex. Since $0 < m < n < 2m$, in $\Gamma(m, n)$ there are arcs between each pair of vertices in both directions, and thus there is a Hamilton path that passes through all vertices from the smallest vertex to the biggest vertex. Then, use the vertices in the order of the Hamilton path to construct a basis as in Theorem 4.2.4, which defines the reduced basis of S.

Furthermore, when $0 < m < n < 2m$, any example S with the maximum number of omitted words among all bases of lengths m, n that generate co-finite languages must also be an example with the longest omitted words. Suppose it is not. Then consider the word graph $G_S^{(m,n)}$. By the same reasoning, there is a order of the vertices in $G_S^{(m,n)}$ and a Hamilton path in $\Gamma(m, n)$ that passes through all vertices in the same order. Use the vertices in the same order to construct a basis as in Theorem 4.2.4, which defines a reduced set S' that has the longest omitted words. Furthermore, since the longest omitted words are not in S'^* but in S^*, the number of omitted words in $\overline{S'^*}$ is strictly greater than that of $\overline{S^*}$, which contradicts the maximality of the number of words not in S^*.

In general, an example with the maximal number of omitted words may not be an example with the longest omitted words. For example, let $m = 2$ and $n = 5$. Let $S_1 = \Sigma^2 \cup \Sigma^5 \setminus \{ \mathtt{00001}, \mathtt{00100}, \mathtt{10010}, \mathtt{01011}, \mathtt{01101}, \mathtt{10111}, \mathtt{11110} \}$, and $S_2 = \{ \mathtt{00}, \mathtt{01}, \mathtt{10}, \mathtt{11}, \mathtt{00000}, \mathtt{00010}, \mathtt{01010}, \mathtt{10000}, \mathtt{10001}, \mathtt{10010}, \mathtt{10011}, \mathtt{10101}, \mathtt{11000}, \mathtt{11001}, \mathtt{11010}, \mathtt{11011}, \mathtt{11101}, \mathtt{11111} \}$. Both bases generate co-finite languages. The longest word not in S_1^* is $\mathtt{00001001011011110}$ of length 17, which is the maximum among all examples with $m = 2, n = 5$, and the longest words not in S_2^* are of length 13, namely $\mathtt{01000(0+1)10111(0+1)0}$. But there are only 38 words not in S_1^*, while there are 112 words not in S_2^*.

4.3 Experiment statistics

In this section, I will summarize some of my experimental results. Most of the experiments were done with GRAIL [131]. GRAIL automatically determines the alphabet by choosing all letters and numbers that appear in the input. So, although $\{ \mathtt{00}, \mathtt{000} \}$ does not generate a co-finite language over the binary alphabet, GRAIL will treat it as over the unary alphabet and result in a co-finite language over the unary alphabet. In addition, when a basis S generates Σ^*, we have $\mathrm{llw}\left(\overline{S^*}\right) = -1$,

but such an S is counted into the case of length 0 for programming reasons. Considering the number of words, these cases are not numerous and will not significantly affect the statistics.

For words of three lengths over the binary alphabet, I randomly chose each word with fixed probability according to their length. The numbers of bases in my experiments that have the same length of the longest omitted words are given in Table 4.5.

Table 4.5: Experiment summary on the number of different cases — one

length-probability	∞	$0/\cdot 0^a$	$1/\cdot 1$	$2/\cdot 2$	$3/\cdot 3$	$4/\cdot 4$	$5/\cdot 5$	$6/\cdot 6$	$7/\cdot 7$	$8/\cdot 8$	$9/\cdot 9$
80%2,80%3,80%4[b]	829213	0	71918	0	80178	0	42076	4376	4154	3063	2153
(1038275 in total)		93	1027	0	24						
80%3,80%4,80%5	939533	0	0	2	0	9	11477	0	21190	23379	0
(1080802 in total)		21036	24278	1136	13331	9673	2462	4505	3798	928	1946
		780	279	456	367	51	69	66	34	7	5
		1	3	0	0	0	0	1			
all2,90%5	41121	0	0	0	3676	0	32440	0	20260	0	5786
(104690 in total)		0	1172	0	211	0	24				
all3,90%5	293447	0	0	0	0	0	0	0	14880	0	0
(440370 in total)		0	0	0	79090	0	0	0	0	0	44041
		0	0	0	0	0	8912				
all2,80%3,80%4	6600	0	1682	0	1978			(10260 in total)			
all2,90%3,90%4	8930	0	10182	0	4898			(24010 in total)			
all3,90%4,90%5	254543	0	0	12794	0	13519	649183	0	676868	211910	0
(2010370 in total)		141202	25184	0	20486	2209	0	2114	154	0	174
		15	0	14	0	0	1				
90%2,90%3,all4	33772	0	23557	0	10965	0	14836		(83130 in total)		
70%3,70%4,all5	263442	0	0	61	0	410	0	0	17412	0	0
(308790 in total)		0	27465								
90%3,90%4,all5	19438	0	0	6979	0	7048	0	0	22870	0	0
(85450 in total)		0	29115								
all combination $1, 2, 3$	12014	4104	259	0	6			(16383 in total)			
all3,half4,most5	36341	0	0	0	0	0	0	0	23616	336	0
(144263 in total)		50792	180	0	29535	0	0	2916	0	0	547
all3,half4,half5	87528	0	0	0	0	0	0	0	0	0	0
(87539 in total)		0	0	0	0	0	0	0	1	0	0
		4	0	5	0	0	1				

[a] Symbol ∞ means the language is not co-finite. Symbol $\cdot 1$ means lengths $11, 21, 31, \ldots$.

[b] We chose each word of length 2 with 80% probability, of length 3 with 80% probability, and of length 4 with 80% probability. The number 829213 below is the total number of samples.

Table 4.5 shows that in general, a basis that generates a co-finite language does not necessarily contain all words of the smallest length or all words of the greatest length. For words of lengths $3, 4, 5$, there are bases that contain all words of length 3 and bases that contain all words of length 5, which generate co-finite languages and the longest words not in the generated languages are rather long. In addition, there indeed exist examples of words of lengths p, q, r achieving longer length of the longest omitted words than those in the 2FPFM where the two lengths are $\{ p, q \}$, $\{ p, r \}$, or $\{ q, r \}$. For example, let $\{ p, q, r \} = \{ 3, 4, 5 \}$. Among all possible bases in the 2FPFM where the two lengths $\in \{ 3, 4, 5 \}$, the longest omitted words not in a generated language are of length 25. The set of words of lengths $3, 4, 5$, say $S = \{$ `000`, `001`, `010`, `100`, `111`, `0001`, `0010`, `0011`, `0100`, `0101`, `0110`, `0111`, `1000`, `1001`, `1010`,

1011, 1100, 1101, 1110, 00000, 00010, 00011, 00101, 00110, 00111, 01001, 01010, 01100, 01110, 10000, 10010, 10011, 10100, 10101, 10110, 10111, 11000, 11001, 11010, 11011, 11100, 11101, 11110, 11111 }, generates a co-finite language and one of the longest words not in S^* is 000011010101010110001111100001101111, which is of length 36.

Experiments suggest that if a set of words contains words of three consecutive lengths, then it must contain most of the words ($\geq$ 50%) in order to generate a co-finite language. The bases were randomly chosen over the binary alphabet, containing words of lengths $3, 4, 5$ with different possibilities. The numbers of bases that have the same length of the longest omitted words are summarized in Table 4.6. The few examples that generate a co-finite language in the experiments with low probability of 5% and 10% are those that contain only one letter, which is not co-finite over the binary alphabet.

Table 4.6: Experiment summary on the number of different cases — two

length-probability	∞	0/·0	1/·1	2/·2	3/·3	4/·4	5/·5	6/·6	7/·7	8/·8	9/·9
5%3,5%4,5%5	4706	0	0	0	0	0	3	0	3		
10%3,10%4,10%5	4982	0	0	0	0	0	1				
15–45%3,15–45%4,15–45%5	5000	0									
50%3,50%4,50%5	4999	0	0	0	0	0	0	0	0	0	0
(5000 in total)		0	0	0	0	0	0	0	0	0	0
		0	0	1							
55%3,55%4,55%5	4999	0	0	0	0	0	0	0	0	0	0
(5000 in total)		0	0	0	1						
60%3,60%4,60%5	4994	0	0	0	0	0	0	0	0	0	0
(5000 in total)		0	2	0	1	0	0	1	0	0	2
65%3,65%4,65%5	4972	0	0	0	0	0	0	0	1	3	0
(5000 in total)		2	3	0	7	7	1	1	1	0	1
		0	0	1							
70%3,70%4,70%5	4925	0	0	0	0	0	1	0	6	9	0
(5000 in total)		14	17	3	13	4	2	2	2	1	1
75%3,75%4,75%5	4751	0	0	0	0	0	12	0	21	50	0
(5000 in total)		34	34	3	38	20	2	15	4	1	6
		3	0	3	2	0	0	1			
80%3,80%4,80%5	4392	0	0	0	0	0	55	0	95	108	0
(5000 in total)		86	105	2	56	49	11	11	10	5	7
		3	0	2	2	0	0	0	0	0	1
85%3,85%4,85%5	3570	0	0	0	0	1	217	0	319	221	0
(5000 in total)		174	242	10	89	71	26	22	23	4	6
		0	1	1	3						
90%3,90%4,90%5	2271	0	0	19	0	10	707	0	716	221	0
(5000 in total)		164	641	13	72	73	43	8	14	9	16
		0	1	0	1	1					
95%3,95%4,95%5	625	0	0	197	0	90	1214	0	690	65	0
(3625 in total)		35	635	1	20	25	16	0	4	2	4
		0	1	0	1						

In 2008, Bassino, Giambruno, and Nicaud [5] proved that for the uniform distribution over all bases S of fixed number of words, the average state complexity $\mathrm{sc}(S^*) = \Theta(\mu)$, where $\mu = \sum_{w \in S} |\, w \,|$. So the case of a basis S such that $\mathrm{llw}\left(\overline{S^*}\right)$ is exponential in μ, if such a basis exist, is very rare.

Chapter 5

Computational Complexity of the FPFM

In this chapter, I will discuss algorithms for and the computational complexity of a decision problem related to the FPFM and its variations. In §5.1, I will provide two exponential-time algorithms for the general decision problem. In §5.2, I will give a polynomial-time algorithm for the particular case of the 2FPFM. In §5.3 and §5.4, I will show two polynomial-time algorithms for decision problems related to the variations on 2FPFM of infinite words and with overlap, respectively. In the last section, I will discuss the computational complexity of some related decision problems of the 2FPFM.

5.1 Algorithm for the FPFM

Recall that the FPFM is to find the longest words that are not in a generated co-finite language. Since there is no simple test to specify whether the given words generate a co-finite language, a natural variation on the FPFM is to check whether the given words do indeed generate a co-finite language. So, more formally, we want to find algorithms to solve the following decision problem.

Problem 5.1.1 (DECISION PROBLEM FOR THE FPFM)**.**

INPUT: *k words $x_1, x_2, \ldots, x_k$, not necessarily distinct.*

OUTPUT: *YES, if $L = \{ x_1, x_2, \ldots, x_k \}^*$ is co-finite; or NO otherwise.*

The first algorithm to solve the problem was given by Shallit [154] as follows: first construct an NFA M_1 to accept $S = \{ x_1, x_2, \ldots, x_k \}$, where M_1 has transitions from the initial state to accept each of the words x_i. Then, by adding ϵ-transitions from each final state to the initial state, an NFA-ϵ M_2 can be constructed to accept $\{ x_1, x_2, \ldots, x_k \}^*$. Using the subset construction, convert M_2 to a DFA M_3, and

then exchange final states and non-final states to get M_4. Finally, detect whether there is a reachable cycle including a state that leads to a final state.

The machine M_2 can be constructed without making an explicit M_1 in linear time in the size of the input, which is the total number of symbols in the x's, plus k. The constructed NFA M_2 is of the same size as the input plus a constant. The conversion of M_3 can be carried out in $O(|\Sigma|\, n2^n)$ time, where n is the number of states in M_2. The new DFA M_3 has $\leq 2^n$ states. The machine M_4 does not have to be constructed explicitly. To detect a reachable cycle, we first use depth-first search from all final states in the reversed-transition of the machine to eliminate states that do not lead to a final state. Then we do depth-first search from the initial states and record the states visited to detect if there is a cycle. There is a cycle if and only if there is a state visited twice. The running time for constructing M_1, M_2 is polynomial in the size of the input, and the running time for converting M_3 and finding a cycle can be exponential in the size of the input. By this algorithm, if there is no cycle, then all longest paths in the machine M_4 starting from the initial state and going to a final state are also obtained, which essentially solves the corresponding FPFM as well.

In the Step 3, one can also use Brzozowski's algorithm for minimization of finite automata, which runs in $O(n2^{2n})$ time [23]. Although it is slower than the usual NFA-DFA convention algorithm, it can reduce the size of M_3, and thus reduce the running time of the last step. Usually, the number of states in the minimal DFA equivalent to M_3 is much smaller than the number of states in M_3. Experiments using GRAIL [131] with Brzozowski's algorithm showed that the algorithm can solve the DECISION PROBLEM FOR THE FPFM (and accordingly the FPFM) very quickly when the longest words in the basis are of length ≤ 7.

We can also solve the decision problem by first building the DFA M described in Theorem 3.2.5 on page 81, which has $\leq \frac{2}{2|\Sigma|-1}(2^n\,|\Sigma|^n - 1)$ states, and then detecting if there is any cycle in the DFA M by breadth-first-traversal to record the states visited, which costs exponential time and uses exponential space in the worst case. But it is not easy to construct the DFA M in Theorem 3.2.5.

5.2 Algorithm for the 2FPFM

Theorem 5.2.1. *The 2FPFM can be solved in polynomial time.*

I will present a polynomial-time algorithm to solve the decision problem of the FPFM in the special case of the 2FPFM.

By Theorem 2.6.6, a given set of words S generates a co-finite language if and only if there is no cycle in the word graph $G_S^{(m,n)}$. So we can decide if S^* is co-finite by detecting whether there is any cycle in $G_S^{(m,n)}$, which can be constructed from S in polynomial time. The first several steps check special cases and ensure the cardinality of the input words is exponential in n. By Theorem 2.6.6, the set

Figure 5.1: A polynomial-time algorithm to solve the 2FPFM

Input: words $x_1, x_2, \ldots, x_k$.

Output: YES, if x's are of only two lengths and $\{ x_1, x_2, \ldots, x_k \}^*$ is co-finite; NO, otherwise.

enumerate all words in the input and find the lengths m, n with $m < n$;
if *no such* m, n *exist* **then**
  return NO ;
else if $\gcd(m, n) \neq 1$ **then**
  return NO ;
else if $m = 1$ *or* $n = 1$ **then**
  if *input does not contains all words of length* 1 *;*
  then return NO ;
else if *the number of words in input is* $\leq |\Sigma|^m + |\Sigma|^n / n$ **then**
  return NO ;
else if *the input does not contains all words of length* m **then**
  return NO ;
else
  construct the word graph $G_S^{(m,n)}$;
end
if *there is any cycle in* $G_S^{(m,n)}$ **then**
  return NO ;
else
  return YES ;
end

$S = \{ x_1, x_2, \ldots, x_k \}$ of two lengths m, n, where $1 < m < n$ and $\gcd(m, n) = 1$, generates a co-finite language if and only if S contains all words of length m and there is no cycle in $G_S^{(m,n)}$. Now it remains to see that the word graph can be constructed in polynomial time. The arcs in $G_S^{(m,n)}$ are words of length n and vertices in $G_S^{(m,n)}$ are words of length $n-m$, the total number of which is less than $2|\Sigma|^n$. But by Theorem 2.6.23, it follows that a basis that generates a co-finite language consists at least $|\Sigma|^m + |\Sigma|^n/n$ words. So we first check that the number of words in the input is at least $|\Sigma|^m + |\Sigma|^n/n$. If it is not, then it cannot generate a co-finite language, and we just return NO. This ensures that the computation, if any, of the word graph $G_S^{(m,n)}$ is polynomial in the size of the input.

Lines 1–11 in the algorithm can be done in linear time. The construction of the word graph in Line 14 can be done in time less than n times the size of the input. Cycles can be detected by breadth-first-traversal from each unvisited vertex. Since each vertex is visited at most once, the cycle-detection can be done in time linear in the size of the word graph $G_S^{(m,n)}$, the size of which is again bounded by n times the size of the input.

When $G_S^{(m,n)}$ has no cycle, then one can find the longest path in the directed acyclic graph $G_S^{(m,n)}$ in polynomial time. Suppose j is the length of the longest path. Calculating $l = n + jm$, then $g(m, l) = ml - m - l$ is the length of the longest words not in S^*. By finding the labeling of the longest path, the longest words not in S^* can be constructed to solve the corresponding 2FPFM.

5.3 Algorithm for the case of infinite words

In this section, we will discuss instead the co-finiteness of $\{ x_1, x_2, \ldots, x_k \}^\omega$ and ${}^\omega\{ x_1, x_2, \ldots, x_k \}^\omega$. By applying the reverse operation on each word, the result on the co-finiteness of ${}^\omega\{ x_1, x_2, \ldots, x_k \}$ can be obtained accordingly from the result on the co-finiteness of $\{ x_1, x_2, \ldots, x_k \}^\omega$.

Problem 5.3.1 (DECISION PROBLEM FOR FPFM OF INFINITE WORDS)**.**

INPUT: *k finite words $x_1, x_2, \ldots, x_k$, not necessarily distinct.*

OUTPUT: *YES, if $L = \{ x_1, x_2, \ldots, x_k \}^\omega$ is co-finite in Σ^ω; or NO otherwise.*

There is a polynomial algorithm for the DECISION PROBLEM FOR FPFM OF INFINITE WORDS as in the following theorem. The algorithm is based on Shallit [154]'s idea.

Theorem 5.3.2. *The problem DECISION PROBLEM FOR FPFM OF INFINITE WORDS can be solved in polynomial time.*

Proof. We assume $x_i \neq \epsilon$. Let $S = \{ x_1, x_2, \ldots, x_k \}$. By Proposition 3.4.4, S^ω is co-finite in Σ^ω if and only if $S^\omega = \Sigma^\omega$. Then by Proposition 3.4.3, $S^\omega = \Sigma^\omega$ if and only if for any word $w \in \Sigma^\omega$ there is a finite nonempty prefix of w that is in S.

The algorithm is as follows. Let S' be the minimal subset of S such that for any $w \in S \setminus S'$ there is a word $u \in S'$ and u is a prefix of w. In the first step, we can construct a tree T for S, where each node is labeled by a word in Σ^*. The root of T is labeled by ϵ. The node u has a child v if $v = ua$ for some $a \in \Sigma$. The leaves of T correspond precisely to words from S'. This tree T can be constructed in linear time in the size of the input S. To see this, we simply add each word from S into an empty tree. When we add a word w, we first find the longest prefix $u = w[1..k]$ that appears in the tree. If that prefix u is a leaf, then we simply ignore w since $w \notin S'$. If that prefix $u = w$, then we cut off all children $\{ v_i \}$ of that node since w is a prefix of every v_i. Otherwise, from the node u, we construct a list of nodes labeled by $w[1..k+1], w[1..k+2], \ldots, w$. In the second step, we check whether every internal node of T has exactly $|\Sigma|$ children. If so, then S^ω is co-finite in Σ^ω; otherwise, S^ω is not co-finite in Σ^ω. This can be done in linear time in the size of the tree.

To see the correctness of the algorithm, first we suppose every internal node of T has exactly $|\Sigma|$ children. Let w be any word in Σ^ω and let u be the longest prefix of w that appears in the tree T. Then u must be a leaf, or otherwise u has less than $|\Sigma|$ children. So, there is a nonempty prefix u of w that is in S. Then S^ω is co-finite. Now we suppose some internal node of T has less than $|\Sigma|$ children, say u. Then none of the words w in $u\Sigma^\omega$ can be factorized into words in S. Then S^ω is not co-finite. □

For bi-infinite words, we do not yet know any polynomial algorithm to decide the co-finiteness. But it is decidable as in the following theorem.

Theorem 5.3.3. *Given k words $x_1, x_2, \ldots, x_k$, to decide whether ${}^\omega\{ x_1, \ldots, x_k \}^\omega$ is co-finite in ${}^\omega\Sigma^\omega$ can be done in exponential time.*

Proof. We assume $x_i \neq \epsilon$. Let $S = \{ x_1, x_2, \ldots, x_k \}$. By Proposition 3.4.12, ${}^\omega S^\omega$ is co-finite in ${}^\omega\Sigma^\omega$ if and only if ${}^\omega S^\omega = {}^\omega\Sigma^\omega$. Then by Proposition 3.4.11, $S^\omega = \Sigma^\omega$ if and only if for any word w in Σ^* there are words $u, v \in \Sigma^*$ such that $uwv \in S^*$.

There is an algorithm proposed by Shallit [154] as follows. First we construct an NFA M accepting the language $\{ w : uwv \in S^*, |u|, |v| < \mathrm{llw}(S) \}$. The NFA M is similar to the NFA accepting S^* except M guesses u and v at the beginning and at the end respectively (this can be done by ϵ transitions). Then we convert M to a DFA and see if $L(M) = \Sigma^*$. Then ${}^\omega\{ x_1, \ldots, x_k \}^\omega$ is co-finite in ${}^\omega\Sigma^\omega$ if and only if $L(M) = \Sigma^*$. □

5.4 Algorithm for the case of concatenation with overlap

Now we will discuss the co-finiteness of $\{ x_1, x_2, \ldots, x_k \}^\natural$ and $\{ x_1, x_2, \ldots, x_k \}^\flat$. By Proposition 3.4.21, if all x_i are of lengths ≥ 2, then $\{ x_1, x_2, \ldots, x_k \}^\natural$ is co-finite if

and only if $\{ x_1, x_2, \ldots, x_k \}^\flat$ is co-finite. So we only discuss $\{ x_1, x_2, \ldots, x_k \}^\flat$ here.

Problem 5.4.1 (DECISION PROBLEM FOR FPFM WITH OVERLAP)**.**

INPUT: *k words $x_1, x_2, \ldots, x_k$, not necessarily distinct.*

OUTPUT: *YES, if $L = \{ x_1, x_2, \ldots, x_k \}^\flat$ is co-finite; or NO otherwise.*

In the case where all the input words are of lengths ≥ 2, there is a polynomial algorithm for the DECISION PROBLEM FOR FPFM WITH OVERLAP as in the following theorem.

Theorem 5.4.2. *When all words are of lengths ≥ 2, the problem DECISION PROBLEM FOR FPFM WITH OVERLAP can be solved in polynomial time.*

Proof. Let $S = \{ x_1, x_2, \ldots, x_k \}$. By Proposition 3.4.22, $S^\flat$ is co-finite if and only if both S^ω and ${}^\omega S$ are co-finite. To decide the latter two problems can be done in polynomial time. □

5.5 Computational Complexity

One natural question is what is the complexity of the decision problem for the FPFM if the input is compressed in some way such that the algorithm does not require exponential time in $\nu = \mathrm{llw}(S)$ to read the input S. As shown in Chapter 3, we can consider some variations on the FPFM.

Problem 5.5.1 (*the FPFM, Variations on input*)**.** *Given a DFA M (or NFA M, or regular expression E) such that there are only finitely many words that cannot be written as concatenations of words in the language accepted by M (or language specified by E), then what is the longest such word(s)?*

A *star-free regular expression* is a regular expression in which the only operators are $\cdot$ and $+$. A variation on the decision problem for the FPFM is the following problem, which is NP-hard as I showed in a draft [157].

Problem 5.5.2 (CO-FINITENESS OF STAR OF STAR-FREE REGULAR EXPRESSION)**.** *Given a star-free regular expression E, decide if the language represented by E^* is not co-finite.*

Theorem 5.5.3. *[157] The problem CO-FINITENESS OF STAR OF STAR-FREE REGULAR EXPRESSION is NP-hard.*

Proof. We reduce from 3SAT. Let $U = \{ u_1, u_2, \ldots, u_n \}$ be a set of variables and $C = \{ c_1, c_2, \ldots, c_m \}$ be a set of clauses over U making up an arbitrary instance of 3SAT. Discard any variable in U that does not appear in any clause in C, which can be done in polynomial time. So now we assume there is no useless variable in U and

C is not empty. Then $n \leq 3m$. For each clause c_i, we construct a star-free regular expression $e_i = t_{(i,1)}t_{(i,2)} \cdots t_{(i,n)}$, where $t_{(i,j)} = F$ if u_j appears in c_i, and $t_{(i,j)} = T$ if $\bar{u}_j$ appears in c_i, and $t_{(i,j)} = (T + F)$ otherwise. Let $E' = e_1 + e_2 + \cdots + e_m$, $E'' = (T + F)^n = (T + F)(T + F) \cdots (T + F)$, and $E = E' + E''(T + F)$. One can verify the construction of E from the instance of 3SAT can be performed in polynomial time. It remains to see the set C of clauses is satisfiable if and only if the language represented by E^* is not co-finite in $\{ T, F \}^*$.

Let S be the finite language represented by the star-free regular expression E. Then, by the construction of E, words in S are of two possible lengths n and $n+1$, and S contains all words of length $n + 1$. If C is satisfiable, then one can verify that $E' \neq E''$, which means that S does not contain all words of length n. Then by the First Lemma of the 2FPFM on page 44, S^* is not co-finite, since a basis S consisting of lengths $n, n + 1$ such that S^* is co-finite must contain all words of length n. If S^* is not co-finite, since $\gcd(n, n + 1) = 1$ and S contains all words of length $n + 1$, then S cannot contain all words of length n, which implies $E' \neq E''$. So C is satisfiable. This finishes the NP-hardness proof. □

By the polynomial-time reduction in the proof, restricting CO-FINITENESS OF STAR OF STAR-FREE REGULAR EXPRESSION to the binary alphabet and/or demanding that the language represented by the regular expression consists of words of only two distinct lengths also results in a NP-hard problem. Furthermore, as a direct consequence, the following problems are also NP-hard.

Problem 5.5.4. *Given a regular expression E, decide if $L(E)^*$ is not co-finite.*

Problem 5.5.5. *Given an NFA M, decide if $L(M)^*$ is not co-finite.*

In 2007, Shallit [154] showed the following more general problem is PSPACE-complete.

Problem 5.5.6 (CO-FINITENESS OF REGULAR LANGUAGE)**.** *Given an NFA M (or regular expression E), decide if $L(M)$ (or $L(E)$) is not co-finite.*

Theorem 5.5.7. *[154] The problem CO-FINITENESS OF REGULAR LANGUAGE is PSPACE-complete.*

Proof. Let E be an arbitrary regular expression, and t be the length of E. Then an NFA M can be constructed in polynomial time to accept E with at most $t + 1$ states [70]. It therefore suffices to prove the case where the language is specified by a regular expression is PSPACE-hard; and the case where the language is accepted by an NFA, is in NPSPACE which by Savitch's theorem [145] is equal to PSPACE.

First we show to decide whether $L(E)$ is co-finite for a regular expression E is PSPACE-hard. The idea is to construct a regular expression E for a PSPACE-bounded TM T and the input x such that T accepts x if and only if $L(E)$ is not co-finite. Here we assume TMs accept input by going into a halting state. First

we construct a new PSPACE-bounded TM T' by adding a new state q' into T. When T goes into the halting state, T' goes into q' instead. Then from q', the machine T' can arbitrarily move its head and stay on q' or transits into the halting state. Then T accepts x if and only if T' accepts x by infinitely many possible computations. Now we construct $E = \Sigma^* \setminus \{ w \}$, where $\{ w \}$ represents the set of all halting computations of the machine T' for the input x. This construction of E can be done efficiently (see [2, §10.4] or [156, Lemma 6.7.1]). Then T accepts x if and only if $L(E)$ is not co-finite.

Now we prove to decide whether $L(M)$ is co-finite for an NFA M with n states is in NPSPACE. Let $N = 2^n$. Now we nondeterministically guess a word w of length i for $N \leq i < 2N$, and check whether M rejects w. Then $L(M)$ is not co-finite if and only if M rejects such a w. To check whether M rejects the word w can be done by storing a square $0-1$ matrix of dimension at most $n+1$, where the entry (p, q) is 1 if and only if q is reachable from p on a given word. At the beginning, this matrix is initialized as an identity matrix, corresponding to the word ϵ. At each step, the matrix is updated to process one guessed letter. At the end, one can check that M rejects the word w by observing that no final state is reachable from the initial state. This algorithm uses only polynomial space.

To prove the correctness of this algorithm, notice that a DFA M' accepting $L(M)$ has at most $N = 2^n$ states. By the pumping lemma, $L(M)^*$ is not co-finite if and only if there is a word of length i not in $L(M)^*$, where $N \leq i < 2N$. In other words, M' rejects some word of length i, where $N \leq i < 2N$. □

Since CO-FINITENESS OF REGULAR LANGUAGE is PSPACE-complete, it follows immediately that both Problem 5.5.4 and Problem 5.5.5 are in PSPACE.

Corollary 5.5.8. *[157] To decide the co-finiteness of the star of a language represented by a regular expression E (or accepted by an NFA M) is in PSPACE.*

Proof. A regular expression (or an NFA) for the star of a language represented by a regular expression (or an NFA) can be constructed in polynomial time. Since CO-FINITENESS OF REGULAR LANGUAGE is PSPACE-complete, the problem in the assertion is also in PSPACE. □

In 2008, Shallit [155] showed that over the unary alphabet, Problems 5.5.4 and 5.5.5 can be solved in polynomial time.

Now we consider another variation on the FPFM — in terms of concatenation with overlap.

Theorem 5.5.9. *Given a star-free regular expression E, to decide if the language represented by $E^\flat$ (or $E^\natural$) is not co-finite is NP-hard and is in PSPACE.*

The proof is similar to that for the FPFM. Let $S = L(E)$. First of all, by Proposition 3.4.21, $E^\flat$ is co-finite if and only if $E^\natural$ is co-finite, when all words in S

are of lengths ≥ 2. Furthermore, when S consists only of words of the same length $n \geq 2$ over Σ, then $S^\flat$ is co-finite if and only if $S = \Sigma^n$. Similarly to the proof of Theorem 5.5.3, one can reduce from 3SAT to show the problem is NP-hard. To show it is in PSPACE, by Proposition 3.4.17, an NFA-ϵ can be built effectively to accept the language S^* with a quadratic number of states in the length of E. Since CO-FINITENESS OF REGULAR LANGUAGE is PSPACE-complete, the problem in the assertion is in PSPACE. The proof of PSPACE is also valid when E is a regular expression.

Theorem 5.5.10. *The two problems*

(a) Given a regular expression E, decide if $L(E)^\flat$ (or $L(E)^\natural$) is not co-finite.

(b) Given an NFA M, decide if $L(M)^\flat$ (or $L(M)^\natural$) is not co-finite.

are both NP-hard and in PSPACE.

By the equivalence of the co-finiteness of $L(E)^\flat$ and $L(E)^\omega, {}^\omega L(E)$, the following theorem holds.

Theorem 5.5.11. *Given a star-free regular expression E, decide if S^ω (or ${}^\omega S$) is not co-finite is NP-hard and in PSPACE, where S is the language $L(E)$. It is also true if the input is a regular expression, or an NFA.*

If the language is specified by a CSG, then to decide whether it is co-finite becomes undecidable.

Theorem 5.5.12. *The following problems are undecidable.*

1. *For two CFGs G_1 and G_2, decide whether $\overline{L(G_1) \cap L(G_2)}$ is co-finite.*
2. *For a CFG G, decide whether $L(G)$ is co-finite.*

Proof. (1) Let M be an arbitrary TM. Here we assume TMs accept input by going into a halting state. Now we construct a new TM M' by adding a new state q' into M. When M goes into the halting state, M' goes into q' instead. From state q', M' can either stay on q' and move its head arbitrarily or goes into the halting state. Then $L(M)$ is not empty if and only if M' accepts some word by infinitely many different computations. Two CFGs G_1 and G_2 such that the set of valid computations of M' is $L(G_1) \cap L(G_2)$ can be effectively constructed from M' (see [71, Lemma 8.6] or [156, Theorem 6.6.1]). $L(G_1) \cap L(G_2)$ is either empty or infinite. Then $L(M)$ is empty if and only if $L(G_1) \cap L(G_2)$ contains finitely many words (when it is empty).

(2) Similarly, we can construct a TM M' for each M such that $L(M)$ is not empty if and only if M' accepts some word by infinitely many different computations. The set of invalid computations of M' is a CFL and there is an algorithm to produce the grammar G for that CFL (see [71, Lemma 8.7] or [156, Theorem 6.6.3]). $\overline{L(G)}$ is either empty or infinite. Then $L(M)$ is empty if and only if $L(G)$ is co-finite (when $\overline{L(G)}$ is empty). □

Nevertheless, the decision problem "Given a finite language S, is S^* co-finite?" has no obvious relationship, complexity-wise, to the problem "Given a finite language S such that S^* is co-finite, find a longest word not in S^*." In addition, if we are only interested in the length of the longest words not in S^*, then finding such a length could possibly be faster than that of finding the longest words not in S^*.

There are algorithms for some related problems in the literature. For example, in 2005, Clément, Duval, Guaiana, Perrin and Rindone [32] developed an algorithm to find all factorizations of a word w into elements in a given finite set S in $O(\nu \mid w \mid + \mu)$ time, where $\nu = \mid S \mid$ and $\mu = \sum_{w \in S} \mid w \mid$, and developed an algorithm to decide whether a word $w \in S^*$ in $O(r \mid w \mid + \mu)$ time, where $r \leq \nu$ is determined by the basis S. In 1977, Maier and Storer [105] showed the following problem is NP-complete: given a finite set S of words, find the shortest word w that contains every word in S as a factor at least once. In 1990, Neraud [116] showed the following problem is co-NP-complete: given a finite set S of words, decide whether $\min_{X \subseteq Y^*} \mid Y \mid = \mid X \mid$.

Chapter 6

Conclusion

6.1 Summary of results on the FPFM and variations

The Frobenius problem in a free monoid (FPFM) is to find the longest words not in a language generated by a given set of words in the case that the generated language is co-finite. There are extensive works on the integer Frobenius problem in the literature; a book [130] on the Frobenius problem lists over 400 references. The FPFM, the generalization of the Frobenius problem on words, however, is a new topic. Research on descriptional complexity of the Kleene star operator for DFAs [174], NFAs, and regular expressions [70] provides another point of view on the FPFM. The difference is that the latter focuses on descriptional complexity in general cases, and the FPFM focuses on co-finiteness.

In the thesis, I studied and completely solved the 2FPFM, which is a special case of the FPFM, where words in the basis are of two distinct lengths m and n. I gave examples, where the longest words not in a generated co-finite language is exponentially long, in the 2FPFM. More precisely, the length of the longest words is $\leq g(m,l) = ml - m - l$, where $l = m\,|\Sigma|^{n-m} + n - m$. This upper bound is tight. There are two equivalent problems of the 2FPFM. One is in combinatorics on words and arises by considering words of length $\equiv n \pmod m$. The other is about the word graph. As a side product of the discussion on word graphs, I studied a generalized form of the de Bruijn graphs (words).

In the 2FPFM, let S be a set of words of lengths m and n over the alphabet Σ, where $1 < m < n$ and $\gcd(m,n) = 1$. The set of the longest words not in S^* is

$$(T\Sigma^m)^{m-2}T, \tag{6.1}$$

where $T = \Sigma^{l-m} \setminus S^*$. The set of words not in S^* is

$$\left(\bigcup_{j \not\in \langle m,n \rangle} \Sigma^j\right) \bigcup \left(\bigcup_{i=0}^{m-2} (T'\Sigma^m)^i T'\right), \tag{6.2}$$

where $T' = \Sigma^{n-m} \cup \Sigma^n \cup \Sigma^{n+m} \cup \cdots \cup \Sigma^{l-m} \setminus S^*$. The set of lengths of words in S^* is $\langle m, n \rangle$ and the set of lengths of words not in S^* is

$$\mathbb{N} \setminus \langle m, l \rangle. \tag{6.3}$$

The number l here can be obtained by either of the two conditions:

1. l is the integer such that $l \equiv n \pmod{m}$, $\Sigma^{l-m} \setminus S^* \neq \emptyset$, and $\Sigma^l \setminus S^* = \emptyset$.
2. $l = n + jm$, where j is the length of the longest path in the word graph $G_S^{(m,n)}$.

In the thesis, I discussed some variations on the FPFM, such as those about infinite (bi-infinite) words, and those about concatenation with overlap. Roughly speaking, the co-finiteness property behaves differently in each case and there are non-trivially co-finite languages in most cases. (Here co-finiteness is non-trivial if the complement of that language is not the empty set.) Let S be a finite set of words of lengths ≥ 2. If S^* is co-finite, then both S^ω and ${}^\omega S$ are co-finite. If both S^ω and ${}^\omega S$ are co-finite, then ${}^\omega S^\omega$ is co-finite. In addition, $S^\natural$ is co-finite, if and only if $S^\flat$ is co-finite, if and only if both S^ω and ${}^\omega S$ are co-finite. If S is a finite set and S^* is co-slender, then both S^ω and ${}^\omega S$ are co-finite. In the thesis, I also discussed some related problems, such as the size of S with S^* co-finite, the number of omitted words in the 2FPFM, the generalized local postage-stamp problem, and related algorithms and computational complexity.

Table 6.1: Summary for the unary alphabet/integers

	$\lvert S \rvert$	$\sum \lvert x_i \rvert$	$\max \lvert x_i \rvert$	$\mathrm{sc}(S)$	$\mathrm{nsc}(S)$	$\mathrm{alph}(S)$	CSG LBA
$\mathcal{L} = \mathrm{llw}\left(\overline{S^*}\right)$	unbnd. unbnd.	quadratic quadratic	quadratic quadratic	quadratic quadratic	quadratic quadratic	quadratic quadratic	unbnd. unbnd.
$\mathrm{sc}(S^*)$	unbnd. unbnd.	quadratic quadratic	quadratic quadratic	quadratic quadratic	quadratic quadratic	quadratic quadratic	*
$\mathrm{sc}(\overline{S^*})$	unbnd. unbnd.	quadratic quadratic	quadratic quadratic	quadratic quadratic	quadratic quadratic	quadratic quadratic	*
$\mathrm{nsc}(S^*)$	unbnd. unbnd.	linear linear	linear linear	linear linear	linear linear	linear linear	*
$\mathrm{nsc}(\overline{S^*})$	unbnd. unbnd.	quadratic quadratic	quadratic quadratic	quadratic quadratic	quadratic quadratic	quadratic quadratic	*
$\mathrm{alph}(S^*)$	unbnd. unbnd.	linear linear	linear linear	linear linear	quadratic linear	linear linear	*
$\mathrm{alph}(\overline{S^*})$	unbnd. unbnd.	quadratic quadratic	quadratic quadratic	quadratic quadratic	quadratic quadratic	quadratic quadratic	*
$\mathcal{M} = \left\lvert \overline{S^*} \right\rvert$	unbnd. unbnd.	quadratic quadratic	quadratic quadratic	quadratic quadratic	quadratic quadratic	quadratic quadratic	unbnd. unbnd.
$\mathcal{D}(w)$ is $\begin{smallmatrix}\text{expntl.}\\ \text{expntl.}\end{smallmatrix}$ in $\lvert w \rvert$ for unary words, and $\begin{smallmatrix}\text{polynomial}\\ \text{polynomial}\end{smallmatrix}$ for integers.							
$\#S^a$	*	$\sqrt{\ }$	linear linear	linear linear	linear linear	linear linear	unbnd. unbnd.

[a] Only in the case where S is a finite set.

The bounds on the output of the FPFM are briefly summarized in Tables 6.1 and 6.2. The top part of each entry is the upper bound on the output measure

Table 6.2: Summary for larger alphabets

	$\lvert S \rvert$	$\sum \lvert x_i \rvert$	$\max \lvert x_i \rvert$	$\mathrm{sc}(S)$	$\mathrm{nsc}(S)$	$\mathrm{alph}(S)$	CFG PDA	CSG LBA
$\mathcal{L} = \mathrm{llw}\left(\overline{S^*}\right)$	unbnd. unbnd.	expntl. linear	expntl. expntl.	expntl. linear	expntl. expntl.	expntl. expntl.	Open [a] expntl.	unbnd. unbnd.
$\mathcal{I} = \left\lvert \Sigma^{\mathcal{L}} \setminus S^* \right\rvert$	Open	db-exp. linear	db-exp. db-exp.	db-exp. expntl.	db-exp. expntl.	db-exp. expntl.	Open expntl.	unbnd. unbnd.
$\mathrm{sc}(S^*)$	unbnd. unbnd.	expntl. linear	expntl. expntl.	expntl. linear	expntl. expntl.	expntl. expntl.	$*$	$*$
$\mathrm{sc}(\overline{S^*})$	unbnd. unbnd.	expntl. linear	expntl. expntl.	expntl. linear	expntl. expntl.	expntl. expntl.	$*$	$*$
$\mathrm{nsc}(S^*)$	unbnd. unbnd.	linear linear	expntl. linear	linear linear	linear linear	linear linear	$*$	$*$
$\mathrm{nsc}(\overline{S^*})$	unbnd. unbnd.	linear linear	expntl. linear	expntl. linear	expntl. linear	expntl. linear	$*$	$*$
$\mathrm{alph}(S^*)$	unbnd. unbnd.	linear linear	expntl. linear	expntl. linear	expntl. linear	linear linear	$*$	$*$
$\mathrm{alph}(\overline{S^*})$	unbnd. unbnd.	linear linear	expntl. expntl.	db-exp. linear	db-exp. expntl.	db-exp. expntl.	$*$	$*$
$\mathcal{M}=\left\lvert \overline{S^*} \right\rvert$, and $\mathcal{W}=\sum_{w \in \overline{S^*}} \lvert w \rvert$	unbnd. unbnd.	db-exp. linear	db-exp. db-exp.	db-exp. expntl.	db-exp. expntl.	db-exp. expntl.	Open [b] expntl.	unbnd. unbnd.
$\mathcal{D}(w)$ is $\begin{smallmatrix}\text{expntl.}\\ \text{expntl.}\end{smallmatrix}$ in $\lvert w \rvert$.								
$\#S^c$	$*$	linear linear	expntl. expntl.	expntl. expntl.	expntl. expntl.	expntl. expntl.	Open expntl.	unbnd. unbnd.

[a] It is at least expntl., which is also true for DPDA.
[b] At least expntl. for DPDA.
[c] Only in the case where S is a finite set.

that labels the row in the input measure that labels the column, and the bottom part of each entry is the lower bound on achievable examples S with S^* co-finite in the same output measure and input measure. (The reader can refer to Tables 2.2 and 2.3 for notation of the measures.) The entries unbnd., linear, quadratic, expntl., db-exp., $*$, and Open mean the bound is unbounded, linear, quadratic, exponential, doubly-exponential, not sensible, and still open, respectively.

Some of my work was published in the STACS 2008 proceedings [84], and some is still to be published.

6.2 Open problems

There are still some problems about the FPFM that remain open. First of all, as discussed in the Chapter 2, there are several possible measures used to describe the size of the input words. Although the bound on the length of the answer to the FPFM is exponential in ν (the length of the longest input word) and this bound is tight as I showed in Chapter 2, there is still the possibility that the exponential bound in μ, the total number of symbols in the input words, is not tight.

Open Problem 6.2.1. *Is the exponential bound $\mathcal{L} = O(2^{\mu})$ tight, where $\mathcal{L} = \mathrm{llw}\left(\overline{S^*}\right)$ and μ is the total number of symbols in S, for a finite set S of words?*

A regular expression (respectively, an NFA) with size polynomial in ν (the length of the longest input word) can be made to represent (respectively, to accept) each S in the family of languages I gave in Chapter 4, where the longest omitted words are of length exponential in ν. But for the case of DFAs, the corresponding problem is still open. In addition, for CFGs and PDAs, we don't even know a proper upper bound on the length of the longest omitted words, except that it must be at least exponential since DPDAs can be used to accept the exponential example in Chapter 4.

Open Problem 6.2.2. *Is the exponential bound* $\mathcal{L} = O\big(2^{\mathrm{sc}(S)}\big)$ *tight, where* $\mathcal{L} = \mathrm{llw}\left(\overline{S^*}\right)$ *and* $\mathrm{sc}(S)$ *is the state complexity of a finite language* S*?*

Open Problem 6.2.3. *What is a bound on* $\mathcal{L}$*, the length of the longest words not in* S^**, in terms of the size of a PDA* M *(or a CFG* G*), where* $S = L(M)$ *(or* $S = L(G)$*)?*

The FPFM with 2 lengths, the 2FPFM, is solved in Chapter 2 of the thesis. Some other interesting topics to pursue include the FPFM with a fixed number of lengths, such as the 3FPFM. It is not likely that the kFPFM for $k \geq 3$ can be solved easily in general, since for the analog of integers, the case of a larger number of integers is dramatically harder than the case for two integers. There is a simple expression for the Frobenius number of two integers, which is at least 100 years old, but for three integers the expression was not discovered until recently and the expression involves some implicit constants. Nevertheless, for some particular situations, there is the possibility of the existence of explicit expressions or fast algorithms for the FPFM with input given by certain patterns.

Open Problem 6.2.4. *Is there any special integer sequence, such that when the lengths of words in the basis* S *constitute such a sequence, the length of the longest words not in* S^* *can be computed efficiently?*

In Chapter 3, some analogous problems of the FPFM are discussed in the setting of right-infinite words, left-infinite words, and bi-infinite words. Let S be an infinite set of finite words. Then I showed that some S can generate a non-trivially co-finite language in the one-sided infinite words. Here co-finiteness is non-trivial if the complement of that language is not the empty language. But it is still open whether there is any non-trivially co-finite language in the bi-infinite words, which is generated by some S.

Open Problem 6.2.5. *Is there any infinite set* S *of finite words such that* ${}^{\omega}S^{\omega}$ *is co-finite in* ${}^{\omega}\Sigma^{\omega}$ *and* $\overline{{}^{\omega}S^{\omega}}$ *is not the empty set?*

Open Problem 6.2.6. *Is there any infinite set* S *of finite words such that* S^{ω} *is co-finite in* Σ^{ω} *and* $\overline{S^{\omega}}$ *is of size* ≥ 2*?*[1]

[1]See the construction (3.73) on page 100 for a set S such that $\left|\,\overline{S^{\omega}}\,\right| = 1$.

Open Problem 6.2.7. *Is there any set S of finite words such that ${}^{\omega}S^{\omega}$ is co-finite in ${}^{\omega}\Sigma^{\omega}$ but neither of the languages S^{ω} and ${}^{\omega}S$ is co-finite in the corresponding set of one-sided infinite words?*

In Chapter 4, I showed how to construct a generalized de Bruijn word τ by first finding two de Bruijn words of specific orders over specific alphabets. The word I constructed, however, is not the lexicographically least generalized de Bruijn word of the same length. Since the lexicographically least de Bruijn word can be calculated efficiently, it is likely that the lexicographically least generalized de Bruijn word can also be calculated efficiently, although we do not have any such algorithm yet.

Open Problem 6.2.8. *How can we efficiently compute the lexicographically least generalized de Bruijn word τ?*

Although there is a tight exponential bound on the length of the answer for the FPFM in the length of the longest words in the input, there is the possibility that the FPFM could be solved in time faster than exponential in terms of the total number of symbols in the input. So far, we only know that any algorithm to solve the FPFM must run in at least linear time, and there is an exponential-time algorithm to solve the FPFM. Furthermore, if we only ask for the length of the longest omitted words instead of the longest words, then it is also possible that a faster algorithm may exist. Any bound on the answer for the FPFM described in terms of other measures is still to be studied. For a co-finite language, there may or may not be a relationship between different measures. Some measures may be more related and describe computational complexity better.

Open Problem 6.2.9. *Is there any efficient algorithm to solve the FPFM?*

The original Frobenius problem is NP-hard. It is very likely that no polynomial algorithm for the FPFM exists, but we have no proof of this. We do not even know if there is an efficient algorithm to decide, given a finite set S of finite words, whether S^* is co-finite. The decision problem with compressed input (by NFAs or by regular expressions) is NP-hard and in PSPACE.

Open Problem 6.2.10. *What is the computational complexity of the problem to decide whether a given finite language generates a co-finite language?*

References

[1] T. van Aardenne-Ehrenfest and N. G. de Bruijn. Circuits and trees in oriented linear graphs. *Simon Stevin*, 28:203–217, 1951.

[2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[3] M. Ajtai, J. Komlós, and E. Szemerédi. An $O(n \log n)$ sorting network. In *Proc. Fifteenth Ann. ACM Symp. Theor. Comput.*, pages 1–9, 1983.

[4] P. Bachmann. *Niedere Zahlentheorie — Additive Zahlentheorie*. Chelsea, 1910.

[5] F. Bassino, L. Giambruno, and C. Nicaud. The average state complexity of the star of a finite set of words is linear. In *DLT 2008, LNCS 5257*, pages 134–145, 2008.

[6] P. T. Bateman. Remark on a recent note on linear forms. *Amer. Math. Monthly*, 65(7):517–518, 1958.

[7] D. Beauquier. Ensembles reconnaissables de mots bi-infinis limite et déterminisme. In *Automata on Infinite Words, LNCS 192*, pages 28–46, 1985.

[8] M. Beck, R. Diaz, and S. Robins. The Frobenius problem, rational polytopes, and Fourier-Dedekind sums. *J. Number Theory*, 96(1):1–21, 2002.

[9] D. Beihoffer, J. Hendry, A. Nijenhuis, and S. Wagon. Fast algorithms for Frobenius numbers. *Electronic J. Combinatorics*, 12:R27, 2005. `http://www.emis.de/journals/EJC/Volume_12/PDF/v12i1r27.pdf`

[10] E. R. Berlekamp. *Algebraic Coding Theory*. McGraw-Hill, 1968.

[11] E. R. Berlekamp, J. C. Conway, and R. K. Guy. *Winning Ways for Your Mathematical Plays*. Academic Press, 1982.

[12] M.-G. D. Birkhoff. Quelques théorèmes sur le mouvement des systèmes dynamiques. *Bull. Soc. Math. France*, 40:305–323, 1912.

[13] J. Błażewicz, P. Formanowicz, M. Kasprzak, W. T. Markiewicz, and J. Węglarz. DNA sequencing with positive and negative errors. *J. Comput. Biol.*, 6(1):113–123, 1999.

[14] J. Błażewicz, P. Formanowicz, M. Kasprzak, P. Schuurman, and G. J. Woeginger. DNA sequencing, Eulerian graphs, and the exact perfect matching problem. In *WG 2002, LNCS 2573*, pages 13–24, 2002.

[15] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. MacMillan, 1976.

[16] E. Boros. On a linear Diophantine problem for geometrical type sequences. *Discrete Math.*, 66(1/2):27–33, 1987.

[17] A. Brauer. On a problem of partitions. *Amer. J. Math.*, 64(1):299–312, 1942.

[18] A. Brauer and B. M. Seelbinder. On a problem of partitions. II. *Amer. J. Math.*, 76(2):343–346, 1954.

[19] A. Brauer and J. E. Shockley. On a problem of Frobenius. *J. Reine Angew. Math.*, 211:215–220, 1962.

[20] T. C. Brown, W.-S. Chou, and P. J.-S. Shiue. On the partition function of a finite set. *Australas. J. Combin.*, 27:193–204, 2003.

[21] T. C. Brown and P. J.-S. Shiue. A remark related to the Frobenius problem. *Fibonacci Quart.*, 31(1):32–36, 1993.

[22] N. G. de Bruijn. A combinatorial problem. *Indag. Math.*, 8(4):461–467, 1946.

[23] J. Brzozowski. Canonical regular expressions and minimal state graphs for definite events. In *Mathematical Theory of Automata, MRI Sympos. Series 12*, pages 529–561, 1962.

[24] J. R. Büchi. On a decision method in restricted second order arithmetic. In *Proc. 1960 Internat. Congr. on Logic, Method., and Philos. of Sci.*, pages 1–11, 1962.

[25] J. S. Byrnes. On a partition problem of Frobenius. *J. Combin. Theory Ser. A*, 17(2):162–166, 1974.

[26] J. S. Byrnes. On a partition problem of Frobenius, II. *Acta Arith.*, 28(1):81–87, 1975.

[27] C. Câmpeanu and W.-H. Ho. The maximum state complexity for finite languages. *J. Autom. Lang. Comb.*, 9(2/3):189–202, 2004.

[28] C. Câmpeanu, K. Culik II, K. Salomaa, and S. Yu. State complexity of basic operations on finite languages. In *WIA'99, LNCS 2214*, pages 60–70, 2001.

[29] C. Câmpeanu, K. Salomaa, and S. Yu. State complexity of regular languages: finite versus infinite. In *Finite Versus Infinite: Contributions to an Eternal Dilemma*, pages 53–73, 2000.

[30] N. Chomsky. *Syntactic Structures*. Mouton, the Hague, 1957.

[31] P. Chrząstowski-Wachtel and M. Raczunas. Liveness of weighted circuits and the Diophantine problem of Frobenius. In *FCT '93, LNCS 710*, pages 171–180, 1993.

[32] J. Clément, J.-P. Duval, G. Guaiana, D. Perrin, and G. Rindone. Parsing with a finite dictionary. *Theoret. Comput. Sci.*, 340:432–442, 2005.

[33] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2nd edition, 2001.

[34] F. Curtis. On formulas for the Frobenius number of a numerical semigroup. *Math. Scand.*, 67(2):190–192, 1990.

[35] J. L. Davison. On the linear Diophantine problem of Frobenius. *J. Number Theory*, 48(3):353–363, 1994.

[36] R. Dawson and I. J. Good. Exact Markov probabilities from oriented linear graphs. *Ann. Math. Statist.*, 28:946–956, 1957.

[37] G. Denham. Short generating functions for some semigroup algebras. *Electron. J. Combin.*, 10:R36, 2003.
`http://www.emis.ams.org/journals/EJC/Volume_10/PDF/v10i1r36.pdf`

[38] J. Dixmier. Proof of a conjecture by Erdös and Graham concerning the problem of Frobenius. *J. Number Theory*, 34(2):198–209, 1990.

[39] A. L. Dulmage and N. S. Mendelsohn. Gaps in the exponent set of primitive matrices. *Illinois J. Math.*, 8:642–656, 1964.

[40] D. Einstein, D. Lichtblau, A. Strzebonski, and S. Wagon. Frobenius numbers by lattice point enumeration. *Integers*, 7:A15, 2007.
`http://www.emis.ams.org/journals/INTEGERS/papers/h15/h15.pdf`

[41] K. Ellul, B. Krawetz, J. Shallit, and M.-W. Wang. Regular expressions: New results and open problems. *J. Autom. Lang. Comb.*, 10(4):407–437, 2005.

[42] P. Erdös and R. L. Graham. On a linear diophantine problem of Frobenius. *Acta Arith.*, 21:339–408, 1972.

[43] P. Erdös and R. L. Graham. *Old and New Problems and Results in Combinatorial Number Theory*, volume 28 of *Monographies de L'Enseignement Mathématique*. Université de Genève, 1980.

[44] L. Euler. Observ. anal. de combinationibus. *Comm. Acad. Pretrop.*, 13, ad annum 1741–1743:64–93, 1751.

[45] L. G. Fel. Frobenius problem for semigroups $S(d_1, d_2, d_3)$. *Funct. Anal. Other Math.*, 1(2):119–157, 2006.

[46] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume I. John Wiley & Sons, Inc., 1950.

[47] C. Flye Sainte-Marie. Solution to question nr. 48. *L'Intermédiaire Math.*, 1:107–110, 1894.

[48] H. Fredricksen. A survey of full length nonlinear shift register cycle algorithms. *SIAM Review*, 24(2):195–221, 1982.

[49] G. Frobenius. Über Matrizen aus nicht negativen Elementen. *Math.-Nat. Kl., S.-B. Königl. Preuss. Akad. Wiss. Berlin*, pages 456–477, 1912.

[50] R. G. Gallager. *Information Theory and Reliable Communication*. John Wiley & Sons, Inc., 1968.

[51] M. R. Garey and D. S. Johnson. *Computers and Intractability — A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[52] P. Gawrychowski, 2008. Private communication.

[53] S. Ginsburg. *The Mathematical Theory of Context-Free Languages*. McGraw-Hill, 1966.

[54] V. M. Glushkov. The abstract theory of automata. *Russian Math. Surveys*, 16(5):1–53, 1961.

[55] E. L. Goldberg. On a linear diophantine equation. *Acta Arith.*, 31:239–246, 1976.

[56] S. W. Golomb. *Shift Register Sequences*. Holden-Day, 1967.

[57] I. J. Good. Normal recurring decimals. *J. London Math. Soc.*, 21(3):167–169, 1946.

[58] H. Greenberg. An algorithm for a linear Diophantine equation and a problem of Frobenius. *Numer. Math.*, 34(4):349–352, 1980.

[59] H. Greenberg. Solution to a linear Diophantine equation for nonnegative integers. *J. Algorithms*, 9(3):343–353, 1988.

[60] R. K. Guy. *Unsolved Problems in Number Theory*. Springer-Verlag, 1981.

[61] G. H. Hardy and S. Ramanujan. Asymptotic formulae in combinatory analysis. *Proc. London Math. Soc.*, 17:75–115, 1918.

[62] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers*. Oxford University Press, 1938.

[63] B. R. Heap and M. S. Lynn. A graph-theoretic algorithm for the solution of a linear Diophantine problem of Frobenius. *Numer. Math.*, 6:346–354, 1964.

[64] B. R. Heap and M. S. Lynn. On a linear Diophantine problem of Frobenius: an improved algorithm. *Numer. Math.*, 7(3):226–231, 1965.

[65] O. Heden. The Frobenius number and partitions of a finite vector space. *Arch. Math. (Basel)*, 42(2):185–192, 1984.

[66] C.-W. Ho, J. L. Parish, and J.-S. Shiue. On the sizes of elements in the complement of a submonoid of integers. In *Proc. of the 4th Internat. Conf. on Fibonacci Numbers and Their applications*, pages 139–144, 1991.

[67] G. Hofmeister. Remark on linear forms. *Arch. Math. (Basel)*, 65(6):511–515, 1995.

[68] G. R. Hofmeister. Zu einem Problem von Frobenius. *Norske Vid. Selsk. Skr. (Trondheim)*, 5:1–37, 1966.

[69] T. Høholdt, J. H. van Lint, and R. Pellikaan. Algebraic geometry codes. In W. C. Huffman and R. A. Brualdi, editors, *Handbook of Coding Theory*, volume 1, pages 871–961. Springer-Verlag, 1998.

[70] M. Holzer and M. Kutrib. Nondeterministic descriptional complexity of regular languages. *Internat. J. Found. Comput. Sci.*, 14(6):1087–1102, 2003.

[71] J. E. Hopcroft and J. D. Ullman. *Introduction To Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

[72] H.-S. Huang. An algorithm for the solution of a linear Diophantine problem of Frobenius. *Chinese J. Math.*, 9(1):67–74, 1981.

[73] M. Hujter. On a sharp upper and lower bound for the Frobenius problem. Technical Report MO/32, Computer and Automation Inst., Hungarian Academy of Sciences, 1982.

[74] M. Hujter. On the lowest value of the Frobenius number. Technical Report MN/31, Computer and Automation Inst., Hungarian Academy of Sciences, 1987.

[75] M. Hujter and B. Vizvári. The exact solution to the Frobenius problem with three variables. *J. Ramanujan Math. Soc.*, 2(2):117–143, 1987.

[76] J. Incerpi and R. Sedgewick. Improved upper bounds on Shellsort. *J. Comput. System Sci.*, 31(2):210–224, 1985.

[77] M. Ito, L. Kari, Z. Kincaid, and S. Seki. Duplication in DNA sequences. In *DLT 2008, LNCS 5257*, pages 419–430, 2008.

[78] N. Jacobson. *Lectures in Abstract Algebra*, volume I. Springer-Verlag, 1951.

[79] T. Jiang, M. Li, and P. Vitányi. A lower bound on the average-case complexity of Shellsort. *J. Assoc. Comput. Mach.*, 47(5):905–911, 2000.

[80] S. M. Johnson. A linear Diophantine problem. *Canad. J. Math.*, 12:390–398, 1960.

[81] R. Kannan. Solution of the Frobenius problem. Technical Report CMU-CS-89-204, Carnegie-Mellon University, Dept. of Computer Science, 1989.

[82] R. Kannan. Lattice translates of a polytope and the Frobenius problem. *Combinatorica*, 12(2):161–177, 1992.

[83] J.-Y. Kao, J. Shallit, and Z. Xu. The Frobenius problem in a free monoid. *CoRR*, abs/0708.3224, 2007. `http://arxiv.org/abs/0708.3224`

[84] J.-Y. Kao, J. Shallit, and Z. Xu. The Frobenius problem in a free monoid. In *STACS 2008, Proc. 25th Internat. Symp. Theoretical Aspects of Comp. Sci.*, pages 421–432, 2008.

[85] H. G. Killingbergtrø. Using figures in Frobenius's problem. *(Norwegian) Normat*, 2:75–82, 2000.

[86] L. F. Klosinski, G. L. Alexanderson, and L. C. Larson. The fifty-second William Lowell Putnam mathematical competition. *Amer. Math. Monthly*, 99(8):715–724, 1992.

[87] M. J. Knight. A generalization of a result of Sylvester's. *J. Number Theory*, 12:364–366, 1980.

[88] A. Kunz. The value-semigroup of a one-dimensional Gorenstein ring. *Proc. Amer. Math. Soc.*, 25(4):748–751, 1970.

[89] S.-Y. Kuroda. Classes of languages and linear bounded automata. *Inform. Control*, 7(2):207–223, 1964.

[90] G. Lallement. *Semigroups and Combinatorial Applications*. Pure & Applied Mathematics. John Wiley & Sons, Inc., 1979.

[91] P. S. Landweber. Three theorems on phrase structure grammars of type 1. *Inform. Control*, 6(2):131–136, 1963.

[92] D.-T. Lee, C.-L. Liu, and C.-K. Wong. $(g_0, \ldots, g_k)$-trees and unary OL systems. *Theoret. Comput. Sci.*, 22(1/2):209–217, 1983.

[93] T. Leighton. Tight bounds on the complexity of parallel sorting. In *Proc. Sixteenth Ann. ACM Symp. Theor. Comput.*, pages 71–80, 1984.

[94] E. Leiss. Constructing a finite automaton for a given regular expression. *SIGACT News*, 12(3):81–87, 1980.

[95] H. W. Lenstra and C. Pomerance. Primality testing with Gaussian periods, 2005. `http://math.dartmouth.edu/~carlp/aks0221109.pdf`
(Abstract appeared in Proc. 22nd Conf. Kanpur on Found. of Software Tech. Theor. Comp. Sci., LNCS 2556, page 1, 2002).

[96] V. F. Lev. The continuous postage stamp problem. *J. London Math. Soc.*, 73(3):625–638, 2006.

[97] M. Lewin. On a linear diophantine problem. *Bull. London Math. Soc.*, 5(1):75–78, 1973.

[98] M. Lewin. An algorithm for a solution of a problem of Frobenius. *J. Reine Angew. Math.*, 276:68–82, 1975.

[99] M. Lewin. On a problem of Frobenius for an almost consecutive set of integers. *J. Reine Angew. Math.*, 273:134–137, 1975.

[100] S. Lin and T. Rado. Computer studies of Turing machine problems. *J. Assoc. Comput. Mach.*, 12:196–212, 1965.

[101] M. Lothaire, editor. *Combinatorics on Words*. Cambridge University Press, 2nd edition, 1997.

[102] A. Lubiw, J. Shallit, and Z. Xu. A discrete iteration, 2009. Manuscript.

[103] G. S. Lueker. Two NP-complete problems in nonnegative integer programming. Technical Report TR-178, Computer Science Laboratory, Princeton University, 1975.

[104] P. A. MacMahon. Applications of a theory of permutations in circular procession to the theory of numbers. *Proc. London Math. Soc.*, 23(1):305–313, 1892.

[105] D. Maier and J. A. Storer. A note on the complexity of the superstring problem. Technical Report 233, Computer Science Laboratory, Princeton University, 1977.

[106] A. Marathe, A. E. Condon, and R. M. Corn. On combinatorial DNA word design. *J. Comput. Biol.*, 8(3):201–219, 2001.

[107] A. N. Maslov. Estimates of the number of states of finite automata. *Soviet Math. Dokl.*, 11(5):1373–1375, 1970.

[108] C. J. H. McDiarmid and J. Ramírez-Alfonsín. Sharing jugs of wine. *Discrete Math.*, 125:279–287, 1994.

[109] R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Inform. Control*, 9(5):521–530, 1966.

[110] R. McNaughton and H. Yamada. Regular expressions and state graphs for automata. *IRE Trans. Electron. Comput.*, EC-9(1):39–47, 1960.

[111] N. S. Mendelsohn. A linear diophantine equation with applications to nonnegative matrices. *Ann. New York Acad. Sci.*, 175:287–294, 1970.

[112] M. Morales. Syzygies of monomial curves and a linear Diophantine problem of Frobenius. Internal Report, Max Planck Institut für Mathematik, Bonn, 1987.

[113] M. Morse and G. Hedlund. Symbolic dynamics. *Amer. J. Math.*, 60(4):815–866, 1938.

[114] D. E. Muller. Infinite sequences and finite machines. In *Proc. 4th Ann. IEEE Symp. Switching Circuit Theory and Logical Design*, pages 3–16, 1963.

[115] D. A. Narayan and A. J. Schwenk. Tiling large rectangles. *Math. Mag.*, 75(5):372–380, 2002.

[116] J. Neraud. Elementariness of a finite set of words is co-NP-complete. *Theor. Inform. Appl.*, 24(5):459–470, 1990.

[117] A. Nijenhuis and H. S. Wilf. Representations of integers by linear forms in nonnegative integers. *J. Number Theory*, 4:98–106, 1972.

[118] M. Nijenhuis. A minimal-path algorithm for the "money changing problem". *Amer. Math. Monthly*, 86(10):832–838, 1979.

[119] D. C. Ong and V. Ponomarenko. The Frobenius number of geometric sequences. *Integers*, 8:A33, 2008.
`http://emis.dsd.sztaki.hu/journals/INTEGERS/papers/i33/i33.pdf`

[120] T. Ottman, H.-W. Six, and D. Wood. On the correspondence between AVL trees and brother trees. *Computing*, 23:43–54, 1979.

[121] R. W. Owens. An algorithm to solve the Frobenius problem. *Math. Mag.*, 76(4):264–275, 2003.

[122] D. Perrin and J.-É. Pin. *Infinite Words: Automata, Semigroups, Logic and Games*, volume 141 of *Pure and Applied Mathematics*. Academic Press, 2004.

[123] T. Popoviciu. Asupra unei probleme de patitie a numerelor. *Acad. Rep. Pop. Romane, Filiala Cluj, Studii si cercetari stiintifice*, 4:7–58, 1953.

[124] V. Pratt. *Shellsort and Sorting Networks*. PhD thesis, Stanford University, 1971.

[125] Z.-M. Qiu and C.-Y. Niu. On a problem of Frobenius. *J. Shandong Univ. Nat. Sci.*, 21(1):1–6, 1986.

[126] M. Raczunas and P. Chrząstowski-Wachtel. A Diophantine problem of Frobenius in terms of the least common multiple. *Discrete Math.*, 150(1/3):347–357, 1996.

[127] T. Rado. On non-computable functions. *Bell System Tech. J.*, 41(3):877–884, 1962.

[128] A. Ralston. De Bruijn sequences — a model example of the interaction of discrete mathematics and computer science. *Math. Mag.*, 55(3):131–143, 1982.

[129] J. L. Ramírez-Alfonsín. Complexity of the Frobenius problem. *Combinatorica*, 16(1):143–147, 1996.

[130] J. L. Ramírez-Alfonsín. *The Diophantine Frobenius Problem*. Oxford University Press, 2005.

[131] D. Raymond and D. Wood. *Grail*: A C++ library for automata and expressions. *J. Symbolic Comput.*, 11:1–10, 1995.

[132] G. Reinert, S. Schbath, and M. Waterman. Probabilistic and statistical properties of words: an overview. *J. Comput. Biol.*, 7(1/2):1–46, 2000.

[133] E. Remy and E. Thiel. Medial axis for chamfer distances: computing look-up tables and neighbourhoods in 2D or 3D. *Pattern Recognition Lett.*, 23(6):649–661, 2002.

[134] A. de Rivière. Question nr. 48. *L'Intermédiaire Math.*, 1:19–20, 1894.

[135] J. B. Roberts. Note on linear forms. *Proc. Amer. Math. Soc.*, 7(3):465–469, 1956.

[136] J. B. Roberts. On a Diophantine problem. *Canad. J. Math.*, 9:219–222, 1957.

[137] Ö. J. Rödseth. On a linear Diophantine problem of Frobenius. *J. Reine Angew. Math.*, 301:171–178, 1978.

[138] Ö. J. Rödseth. On a linear Diophantine problem of Frobenius. II. *J. Reine Angew. Math.*, 307/308:431–440, 1979.

[139] Ö. J. Rödseth. A note on Brown and Shiue's paper on a remark related to the Frobenius problem. *Fibonacci Quart.*, 32(5):407–408, 1994.

[140] H. Rohrbach. Anwendung eines Satzes der additiven Zahlentheorie auf eine gruppentheoretische Frage. *Math. Z.*, 42(1):538–542, 1937.

[141] H. Rohrbach. Ein Beitrag zur additiven Zahlentheorie. *Math. Z.*, 42(1):1–30, 1937.

[142] H. Rohrbach. Einige neuere Untersuchungen über die Dichte in der additiven Zahlentheorie. *Jahres. Deutscher Math.-Verein.*, 48:199–236, 1938.

[143] J. C. Rosales, P. A. García-Sánchez, and J. I. García-García. Every positive integer is the Frobenius number of a numerical semigroup with three generators. *Math. Scand.*, 94(1):5–12, 2004.

[144] F. Ruskey. Information on necklaces, Lyndon words, de Bruijn sequences. `http://www.theory.csc.uvic.ca/~cos/inf/neck/NecklackInfo.html`

[145] W. J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *J. Comput. System Sci.*, 4(2):177–192, 1970.

[146] H. E. Scarf and D. F. Shallcross. The Frobenius problem and maximal lattice free bodies. *Math. Oper. Res.*, 18(3):511–515, 1993.

[147] I. J. Schur. Zur additiven Zahlentheorie. *Phys. Math. Kl., S.-B. Königl. Preuss. Akad. Wiss. Berlin*, pages 488–495, 1926.

[148] R. Sedgewick. Analysis of Shellsort and related algorithms. In *Algorithms – ESA'96, LNCS 1136*, pages 1–11, 1996.

[149] E. S. Selmer. On the linear diophantine problem of Frobenius. *J. Reine Angew. Math.*, 293/294(1):1–17, 1977.

[150] E. S. Selmer and Ö. Beyer. On the linear Diophantine problem of Frobenius in three variables. *J. Reine Angew. Math.*, 301:161–170, 1978.

[151] S. Sertöz. On the number of solutions of the Diophantine equation of Frobenius. *Discrete Appl. Math.*, 8(2):153–162, 1998.

[152] S. Sertöz and A. Özlük. On a Diophantine problem of Frobenius. *İstanbul Tek. Üniv. Bül*, 39(1):41–51, 1986.

[153] J. Shallit. The computational complexity of the local postage stamp problem. *SIGACT News*, 33(1):90–94, 2002.

[154] J. Shallit, 2007–2009. Private communication.

[155] J. Shallit. The Frobenius problem and its generalizations. In *DLT 2008, LNCS 5257*, pages 72–83, 2008.

[156] J. Shallit. *A Second Course in Formal Languages and Automata Theory.* Cambridge University Press, 2009.

[157] J. Shallit and Z. Xu. An NP-hardness result on the monoid Frobenius problem. *CoRR*, abs/0805.4049, 2008. `http://arxiv.org/abs/0805.4049`

[158] J.-Y. Shao. Some computational formulas of the Frobenius numbers. *J. Math. (Wuhan)*, 4:375–388, 1988.

[159] D. L. Shell. A high-speed sorting procedure. *Comm. ACM*, 2(7):30–32, 1959.

[160] H.-J. Shyr. *Free Monoid and Languages.* Soochow University, 1979.

[161] G. Sicherman. Theory and practice of Sylver coinage. *Integers*, 2:G2, 2002. `http://www.emis.ams.org/journals/INTEGERS/papers/cg2/cg2.pdf`

[162] Z. Skupień. A generalization of Sylvester's and Frobenius's problems on numerical semigroups. *Acta Arith.*, 65(4):353–366, 1993.

[163] M. Z. Spivey. Quadratic residues and the Frobenius coin problem. *Math. Mag.*, 80(1):64–67, 2007.

[164] J. J. Sylvester. On subvariants, i.e. semi-invariants to binary quantics of an unlimited order. *Amer. J. Math.*, 5(1):79–136, 1882.

[165] J. J. Sylvester. Problem 7382. *Math. Quest. with their Sol., Educ. Times*, 41:ix, 21, 1884.

[166] E. Teruel, P. Chrząstowski-Wachtel, J. M. Colom, and M. Silva. On weighted T-systems. In *Application and Theory of Petri Nets 1992, 13th Internat. Conf., LNCS 616*, pages 348–367, 1992.

[167] A. Tripathi. The number of solutions to $ax + by = n$. *Fibonacci Quart.*, 38(4):290–293, 2000.

[168] I. Vardi. *Computational Recreations in Mathematica®*. Addison-Wesley, 1991.

[169] Y. Vitek. Bounds for a linear Diophantine problem of Frobenius. *J. London Math. Soc.*, 10(2):79–85, 1975.

[170] B. Vizvári. *Beiträge zum Frobenius-Problem.* PhD thesis, Technische Hohschule Carl Schorlemmer, 1987.

[171] B. Vizvári. Generation of uniformly distributed random vectors of good quality. Technical Report RRR 17–93, Rutcor Research Report, 1994.

[172] H. S. Wilf. A circle-of-lights algorithm for the "money-changing problem". *Amer. Math. Monthly*, 85(7):562–565, 1978.

[173] A.-G. Xu and Z.-H. Wu. The Petri net method for solving first-degree indeterminate equations (III): research on the Frobenius problem. *Shandong Kuangye Xueyuan Xuebao*, 12(1):63–69, 1993.

[174] S. Yu, Q.-Y. Zhuang, and K. Salomaa. The state complexities of some basic operations on regular languages. *Theoret. Comput. Sci.*, 125:315–328, 1994.

[175] F.-J. Zhang and G.-N. Lin. On the de Bruijn-Good graphs. *(Chinese) Acta Math. Sinica*, 30(2):195–205, 1987.

[176] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Trans. Inform. Theory*, 23(3):337–343, 1977.

Index