PATTERN AVOIDANCE AND EXTREMAL WORDS

NIKHIL RAVISHANKAR

Abstract. This expository paper will focus on exploring combinatorics on words, the study of finite and infinite sequences of symbols. Specifically, we observe when certain kinds of patterns, words following some defined structure, must necessarily occur in words, or if they can be avoided. We discuss how infinite words can be generated with morphisms, look at the infinite Thue-Morse word and its pattern-avoiding properties, define an infinite family of unavoidable patterns, show generalized results about all unavoidable patterns with Zimin's reducting algorithm, and explore a new branch of research based on a variation of avoidance.

1. Introduction

Combinatorics on words is a relatively new branch of mathematics that studies both the properties of finite and infinite sequences of symbols (or *words*) where each symbol is taken from a set of symbols Σ^*. Axel Thue is credited with making the first developments in the field by studying square-free words in the 20th century. Many more mathematicians such as Zimin, the group Lothaire, Dejean, and the group Grytczuk, Kordulewski, and Niewiadomski have made their own contributions like proving the unavoidability of the Zimin patterns, providing two comprehensive books on the subject, posing a conjecture about repetition of words which was later proven, and beginning the study of extremal words, respectively. First, we provide basic definitions and notation commonly used in combinatorics on words. Next, we define morphisms, powerful functions that can generate infinite words, sometimes with special properties related to pattern avoidance. We then formally define patterns and what it means to encounter or avoid them. We observe the Thue-Morse word and the two patterns it avoids, as well as use it to generate a new word avoiding a different pattern. We then explore the infinite family of Zimin patterns, and show that they are all unavoidable, and characterize all unavoidable patterns. We begin exploring extremal words, a subset of pattern-avoiding words that lose their avoidance at the addition of a single letter, and end with some open problems related to this branch.

2. Preliminaries

We begin by introducing rudimentary notation, definitions, and examples for the reader to understand common terminology used when doing combinatorics on words. For more reading, see [Lot02].

Definition 2.1. An *alphabet* is a set of symbols that are called *letters*. Alphabets are typically denoted by Σ. They are typically finite in the context of words, but can be infinite in the context of patterns.

Example. The English alphabet $\{A, B, C, \ldots Z\}$ would be considered an alphabet under this definition.

Date: July 2025.

Example. Furthermore, the set $\{0, 1\}$ can also be considered an alphabet.

Definition 2.2. A *word* is a sequence (possibly empty) of letters, all of which are elements of a particular alphabet Σ. For a finite word $w, |w|$ denotes its length. In particular, there exists a unique *empty word* of length 0, denoted by ε.

Words may be finite, infinite in one direction, or infinite in both directions. However, this paper primarily focuses on finite words and one-sided infinite words. The set of words that can be formed from a given alphabet Σ is denoted as Σ^*, and the set of *nonempty* words that can be formed from that alphabet as Σ^+. The words can be classified by the size of their alphabets (for example, binary and ternary words are made from alphabets of sizes 2 and 3, respectively.

Definition 2.3. A *factor* of a word w is a contiguous subsequence of the letters of w. A *prefix* is a factor found at the beginning of w and a *suffix* is a factor found at the end of w.

Example. The word *bababba* has the factor b as a prefix, *bba* as a suffix, *ba* as a prefix and a suffix, and does not have the word *aa* as a factor (since the instances of a are not adjacent).

Definition 2.4. The *concatenation* of two words u and v, is simply denoted as uv. More precisely, uv is a single word that consists of all the letters of u in order, and then all the letters of v in order. As long as u is finite, this concatenation is well-defined. In particular, concatenation is similar to multiplication in that the words being concatenated are factors. Furthermore, an integer *power* of a word w^n is simply w concatenated to itself n times. In particular, any word raised to the power of 0 results in the empty word ε.

Example. If $u = aba$ and $v = aab$ then $uv = abaaab, vu = aababa$, and $u^2 = abaaba$.

Remark 2.5. Note that unlike multiplication, concatenation is not commutative, so uv is not necessarily equal to vu. However, if $uv = vu$, it can be proven by induction on the length of uv that there exists some word z such that u and v are both powers of z.

3. Morphisms

Morphisms are functions that can be applied to words to generate new words. They are extremely useful tools for generating infinite words and manipulating them to prove results about those words.

Definition 3.1. A *morphism* is a function $h : \Sigma^* \to \Gamma^*$ that is always distributive over concatenation. In other words, for any words $u, v \in \Sigma^*$, $h(uv) = h(u)h(v)$. We call $h(u)$ the *image* of u with respect to h, and u the *preimage* of $h(u)$.

Remark 3.2. Thus, since words can always be expressed as concatenations of individual letters, it follows that morphisms can be thought of as replacing each *letter* of an input word with another *word* to get the newly generated word. In particular, morphisms can be uniquely defined just by their evaluations on every letter of Σ.

Claim 3.3. *For all morphisms $h : \Sigma^* \to \Gamma^*$, we have $h(\varepsilon) = \varepsilon$.*

Proof. Choosing $u = v = \varepsilon$, we have $h(\varepsilon\varepsilon) = h(\varepsilon)h(\varepsilon)$. But since $\varepsilon\varepsilon = \varepsilon$ by the definition of the empty word,we have $h(\varepsilon\varepsilon) = h(\varepsilon) = h(\varepsilon)h(\varepsilon)$. Looking at the two rightmost terms of this equation, the only way to concatenate $h(\varepsilon)$ to a word and still have them equal is for $h(\varepsilon)$ to equal ε. ■

Although it is possible for morphisms to map nonempty words to the empty word, these morphisms are typically not as useful. Thus, we tend to observe morphisms in which nonempty words cannot map to the empty word.

Definition 3.4. A *nonerasing* morphism is a morphism $h : \Sigma^* \to \Gamma^*$ such that $h(u) = \varepsilon \implies u = \varepsilon$.

Example. In the alphabet $\Sigma = \{0, 1\}$, the morphism $h : \Sigma^* \to \Sigma^*$ defined by $h(0) = 1, h(1) = \varepsilon$ is not nonerasing.

Example. However, the one defined by $h(1) = 11,\ h(0) = 0$ is.

Definition 3.5. Furthermore, morphisms that map words to other words in the same alphabet (denoted $h : \Sigma^* \to \Sigma^*$) are known as *endomorphisms.* These endomorphisms are particularly useful for generating infinite words, as they can be applied repetitively without their outputs leaving the domain.

Definition 3.6. The process of using morphisms to generate infinite words is detailed below. For a starting letter a and a morphism h, we compute the following limit, where $h^n(a)$ is the composition of h with itself n times, evaluated at a:

$$\lim_{n\to\infty} h^n(a)$$

Definition 3.7. If, when given the morphism h, there exists at least one a such that this limit exists and is an infinite word, we call the morphism *prolongable*, and the resulting word *morphic*. If the limit exists (it can be infinite or finite), we call the resulting word w a *fixed point* of h because it satisfies $h(w) = w$. We have mentioned that infinite words can be generated from endomorphisms (although it is possible h is not an endomorphism if the codomain contains all possible letters appearing in repeated evaluations of h on a.) However, repeated morphisms do not always generate a morphic word. We now characterize what qualities are present in these morphic words.

Proposition 3.8. *An infinite morphic word w over a morphism $h : \Sigma^* \to \Sigma^*$ satisfies the following conditions:*

1. *If a is the starting letter of w, then $h(a) = au$ for some $u \in \Sigma^+$.*
2. *For a given starting leter a and morphism h, w is unique and equal to $auh(u)h^2(u)h^3(u)\ldots$*

Proof. In order to prove (1) is true, assume for the sake of contradiction that $h(a) \neq au$ for some $u \in \Sigma^+$. There are two possibilities for the form of $h(a)$:

If $h(a)$ does not start with a, yet w does start with a as is stated, we have $w = ax$ for some infinite word x, and $h(w) = h(ax) = h(a)h(x)$. We assumed $h(a)$ does not start with a, so $h(w)$ cannot start with a, so it is impossible for $h(w)$ to equal w, contradicting our earlier finding that $h(w) = w$.

The only other way for $h(a)$ to not equal a that the earlier case does not cover is if $h(a) = a$. However, $lim_{n\to\infty} h^n(a) = a$, which is not an infinite word. Thus, the only remaining possibility is that $h(a) = au$ for some nonempty word u. This possibility works, because the length increases by a nonzero amount (at least $|u|$) each time h is applied.

Because morphisms behave like functions, evaluating h on anything can only have at most one value, thus, if the limiting word w exists, after each successive composition of h, the generated word can only have at most one value, so the final infinite word can only have at most one value. Thus, if w exists, it is unique.

To prove the second part of (2), we can easily verify the form of w satisfies $h(w) = w$:

$$\begin{aligned} w &= auh(u)h^2(u)h^3(u)h^4(u)\dots \\ h(w) &= h(auh(u)h^2(u)h^3(u)h^4(u)\dots) \\ h(w) &= h(a)h(u)h(h(u))h(h^2(u))\dots \\ h(w) &= auh(u)h^2(u)h^3(u)h^4(u)\dots \\ h(w) &= w \end{aligned}$$

The evaluation of a becoming au is highlighted in red. Essentially, when the morphism is applied, a evaluates to au, u to $h(u)$, and in general, $h^n(u)$ to $h^{n+1}(u)$. Every part of w is generated, and the parts are generated in the same order they appear. We have found a word that is a fixed point of h, so this is the unique value of w. ■

We also prove another useful characteristic about the finite words generated by successive applications of morphisms, namely the following:

Proposition 3.9. *If applying the morphism $h : \Sigma^* \to \Sigma^*$ repeatedly to the starting letter $a \in \Sigma$ produces the infinite word w, and we define w_n to be the finite word $h^n(a)$, then w_n is a prefix of w_{n+1} for all $n \in \mathbb{Z}^+$.*

Proof. By the definitions of the terms w_i, we have $h(w_n) = w_{n+1}$ for all integers $n \geq 1$. We first define $h(a)$ as au for some $u \in \Sigma^+$. Observing the first few values of w_i, we find these:

$$\begin{aligned} w_1 &= a \\ w_2 &= au \\ w_3 &= auh(u) \\ w_4 &= auh(u)h^2(u) \\ w_5 &= auh(u)h^2(u)h^3(u) \\ &\vdots \end{aligned}$$

We can conjecture that for all $n \geq 3$, w_n is equal to the concatenation $auh(u)h^2(u)\dots h^{n-2}(u)$. In fact, we can easily prove this is the case with induction:

Our base case starts at $n = 3$, and this is true because $auh(u) = auh^{3-2}(u)$. Our inductive hypothesis is that $w_n = auh(u)h^2(u)\dots h^{n-2}(u)$, and we would like to show our inductive step that if we assume w_n is of the aforementioned form, then w_{n+1} is as well.

Applying the morphism h to our expression for w_n, we find the following:

$$\begin{aligned} h(w_n) &= h\left(auh(u)h^2(u)\dots h^{n-2}(u)\right) \\ w_{n+1} &= h(a)\ h(u)\ h(h(u)\ h(h^2(u))\dots h(h^{n-2}(u) \\ w_{n+1} &= auh(u)h^2(u)\dots h^{n-1}(u) \end{aligned}$$

We used the fact that $h(w_n) = w_{n+1}$, and a similar finding to the previous proof occurs where all the current terms are generated by u up to the previous term $h^{n-3}(u)$, essentially replicating the entirety of w_n and then appending $h^{n-1}(u)$. This completes the inductive proof.

Now that we know the forms of all the w_i, we can verify our main prefix claim, but we must manually check our first two cases: a is a prefix of au, and au is a prefix of $auh(u)$. For $n \geq 3$, we have the following:

$$w_{n+1} = auh(u)h^2(u)h^3(u)\dots h^{n-2}(u)h^{n-1}(u) = w_n h^{n-1}(u)$$

■

Remark 3.10. This prefix constraint can be strengthened (since being a prefix of a word is a transitive property): for any $i, j \in \mathbb{N}$, if $i \leq j$, then w_i is a prefix of w_j. Furthermore, all w_i are prefixes of the infinite morphic word w, because to allow the limiting word w to exist at all, the first n letters of w for any finite n must stay the same after applying the morphism h.

We end this section by using morphisms to generate a well-known example of a morphic word, namely the Fibonacci word. (The Thue-Morse word is another prominent example, which will be covered in more detail later.)

We are working with the binary alphabet $\Sigma = \{0, 1\}$. Consider the following morphism $\varphi : \Sigma^* \to \Sigma^*$, defined as follows:

$$\begin{aligned} \varphi(0) &= 01 \\ \varphi(1) &= 0 \end{aligned}$$

Our starting letter will be 0 since applying the morphism to 0 results in a word that both starts with 0 and is longer than just 0 itself. We define f_n as the word resulting from evaluating $\varphi^n(0)$. We can repeatedly apply φ and observe the first few letters of the Fibonacci word:

$$\begin{aligned} f_0 &= 0 \\ f_1 &= 01 \\ f_2 &= 010 \\ f_3 &= 01001 \\ f_4 &= 01001010 \\ f_5 &= 0100101001001 \\ &\vdots \end{aligned}$$

The limiting word f is known as the *Fibonacci word*, primarily because an equivalent definition to generate new f_i recursively is $f_{n+1} = f_n f_{n-1}$ for all $n \geq 1$.

4. Patterns and Avoidabililty

In this section, we introduce patterns, which are not merely words, but structures that simultaneously encompass all words of a certain form. Then, we explore what patterns must necessarily occur in sufficiently long words on certain alphabets and what patterns can be avoided for a well-constructed infinite word.

Definition 4.1. We denote Δ to be the *pattern alphabet*, the letters of which are called *variables*. Furthermore, words in Δ^* are called *patterns*.

Example. A common pattern is the square, which can be written in variables as $\alpha\alpha$.

Definition 4.2. We say a word $w \in \Sigma^*$ *follows* a pattern $p \in \Delta^*$ if there exists some non-erasing morphism $h : \Delta^* \to \Sigma^*$ such that $h(p) = w$. In other words, if each variable in p can be substituted with a word in Σ^* to generate w, w follows the pattern.

Definition 4.3. For longer finite and infinite words, we say that a word w *encounters* a pattern if there exists some factor of w that follows the pattern p. If no such patterns follow p, then w *avoids* p. Patterns are called *avoidable* or *unavoidable* based on if an infinite word can be constructed that avoids p, or if there are none.

Example. The word *abab* follows the patterns $xx, xy, xyzy$, but not xxx. The morphisms showing the following of the patterns are $h(x) = ab$ for xx, $h(x) = h(y) = ab$ for xy, and $h(x) = h(z) = a, h(y) = b$ for $xyzy$. However, *abab* can't be written as some nonempty word repeated three times. It is also notable that distinct variables can map to the same words, so the morphisms need not be injective.

Definition 4.4. Patterns can be labeled *k-avoidable* if an alphabet of size k is sufficiently large to be able to construct an infinite word avoiding the pattern. In this case, k is called the *avoidability index* of the pattern if k is the minimal alphabet size required to avoid the pattern. Patterns may be labeled *k-unavoidable* if such a construction is impossible with a size k alphabet. If a pattern is unavoidable no matter how large the alphabet is, its avoidability index is ∞.

This is the central concept of pattern avoidance. Which patterns can be avoided and on which alphabets?

This paper focuses primarily on the following patterns:

- Squares, which are 2-unavoidable and 3-avoidable.
- Cubes, which are 2-avoidable.
- Overlaps (of the form $\alpha\beta\alpha\beta\alpha$), which are 2-avoidable.
- The infinite family of Zimin patterns, which are unavoidable.

We can start our observations with squares, noting that a binary alphabet is insufficient to avoid squares infinitely:

Claim 4.5. *All binary words of length 4 or more encounter a square.*

Proof. Without loss of generality, $\Sigma = \{a, b\}$, and the first letter is a. We cannot put another a without creating the square aa, so the next letter is b. Continuing this logic, we must alternate a's and b's. But after 4 letters, we form $abab = (ab)^2$ and encounter a square. ■

However, we can show a binary alphabet can avoid cubes indefinitely. A prominent example of an infinite binary cube-free word is the Thue-Morse word. It is defined by the following morphism $\mu : \Sigma^* \to \Sigma^*$ where $\Sigma = \{a, b\}$.

$$\mu(a) = ab$$
$$\mu(b) = ba$$

Let $t_0 = a$, and t_n be the n-fold composition of μ evaluated at a, $\mu^n(a)$, for $n \geq 1$. The first few iterations are shown below:

$$\begin{aligned} t_0 &= a \\ t_1 &= ab \\ t_2 &= abba \\ t_3 &= abbabaab \\ t_4 &= abbabaabbaababba \\ t_5 &= abbabaabbaababbabaababbaabbabaab \\ &\vdots \end{aligned}$$

The infinite word t, the Thue-Morse word, is the fixed point of this morphism. We can prove t is cube-free, overlap-free, and use it to generate a square-free word over a ternary alphabet. We will need to prove some smaller lemmas about t first, however.

Lemma 4.6. *All occurrences of aa or bb in t occur at even positions. We assume the first letter is indexed at 1, and the occurence's position is indexed by the position of the leftmost letter.*

Proof. Because t is a fixed point of μ, we know $\mu(t) = t$. Furthermore, we know each letter in t maps to a two letter word with exactly one a and one b after applying μ. In other words, the first two letters (positions 1-2) are an a and a b in some order, the next two (positions 3-4) are an a and a b in some order, and so on. Thus, all two letter words at odd positions cannot possibly be aa or bb. Thus, all occurrences of aa or bb can only happen at even positions. ■

Corollary 4.7. *It follows that aaa and bbb do not occur in t, because regardless of the parity of the position in which they occur, they would place an aa or bb at an odd position, which is impossible.*

Lemma 4.8. *For some subword u of t such that u starts on an odd position, the word formed by taking just the odd-position letters of u also appears in t.*

Proof. Since u starts on an odd position, it is a concatenation of several two-letter words that are either ab or ba, and possibly an unpaired a or b (also at an odd position) if $|u|$ is odd. Because the 2-letter words of u occur at odd positions, each is the image of some earlier

letter in t. The preimages of ab and ba are a and b, notably the letter at the odd position, since ab and ba are themselves at odd positions. ■

Remark 4.9. If u starts at an even position, we can determine the letter before the leftmost letter of u, and similarly if u ends at an odd position, we can determine the letter after the rightmost letter, by using the fact that any two consecutive letters (where the first of the two letters has an odd position) must consist of one a and one b. In either case, if we can determine a letter, it would be the opposite of what letter it is adjacent to.

We have the tools to prove a larger result.

Proposition 4.10. *The Thue-Morse word is cube-free.*

Proof. For the sake of contradiction, assume that t contains a nonempty word of the form uuu, and assume uuu is the smallest such instance of a cube in t. By 4.7, uuu cannot equal aaa or bbb (corresponding to the only possible 1-letter values for u), so $|u| \geq 2$ and $|uuu| \geq 6$.

It is also impossible for uuu to alternate between a and b, because that implies the existence of $ababab$ or $bababa$ somewhere in uuu. No matter if these subwords start at even or odd positions, determining any extra letters as detailed in 4.9 would preserve this alternating property. The contradiction arises when taking a preimage with respect to μ, as this would result in either an aaa or a bbb in t, which are impossible by 4.7.

Thus, uuu must contain some occurrence of aa or bb. Because $4 = |uu| \geq |aa|, |bb| = 2$, the occurrence must be contained in the factor uu. Furthermore, it must occur in two places: the first uu of uuu and the second uu of uuu. Since the second instance of uu (and thus the second occurrence of aa or bb) is precisely $|u|$ letters later than the first, by 4.6, both occurrences must be at even positions, so $|u|$ is even.

If uuu starts at an odd position, we can take the odd position letters of uuu. Since $|u|$ is even, taking all the odd letters of uuu is equivalent to taking the odd letters of u three times. Thus, it must also be a cube in t, and one with half the length of uuu, which would contradict the minimality of uuu.

If uuu starts at an even position, taking all the odd position letters of uuu (positions based on the start of t) is equivalent to taking all the even position letters of u (where positions are based on the start of u) three times. This also creates a cube in t with a smaller length, a contradiction. ■

We can use a similar proof to show the Thue-Morse word avoids a different pattern: overlaps, which can be represented in variables as $\alpha\beta\alpha\beta\alpha$. They are called overlaps because there are two instances of a single word $\alpha\beta\alpha$ such that the two overlap, sharing an α.

Proposition 4.11. *The Thue-Morse word is overlap-free.*

Proof. We once again assume, for the sake of contradiction, that t has a factor of the form $uvuvu$ of minimal length. We then aim to show that this implies a smaller overlap, contradicting the minimality.

Once again, we can show that it is impossible for $uvuvu$ to alternate between a and b. Since $uvuvuv \geq 5$, if it were to alternate, it implies the existence of either $ababa$ or $babab$ in

t. Since 5 is odd, either the leftmost letter is at an even position or the rightmost letter is at an odd position. It must be the case that the procedure described in 4.9 will determine an additional letter, implying the existence of the subword *ababab* or *bababa* in t, moreover starting at an odd position and ending at an even position (thus consisting of three 2-letter words generated by an earlier word in t). However, taking the preimage implies the existence of one of *aaa* or *bbb* in t, which is impossible by 4.7.

Once again, if $uvuvu$ cannot alternate between a and b, it must have at least one occurrence of aa or bb, and since this length of 2 is less than the minimum length of $|uvu| = 3$, there are occurrences of aa or bb in one instance of uvu, and another instance of uvu that is $|vu|$ letters later. Based on the restrictions on where aa and bb can occur as detailed in 4.6, $|vu|$ is even.

Just as we did in the last proof, we can form a word by taking all the letters at odd positions within $uvuvu$. Since $|uv|$ is even, the set of letters taken from the first uv ,which we will call $u'v'$ (u' being the letters taken from u, and v' is defined identically) is identical to the letters taken from the second uv, and the letters taken from the final u will be u' as well. Thus, we have found a smaller overlap $u'v'u'v'u'$ in t and contradicted the minimality of $uvuvu$. Hence, t is overlap-free. ■

We have shown cubes and overlaps are 2-avoidable, but we can also use t to generate an infinite square-free word over a ternary (3-letter) alphabet. The process is as follows:

Definition 4.12. Let the infinite word u be generated in the following manner: We write out the entirety of the Thue-Morse word t. We then note how many b's appear between each consecutive instance of a. This number must be 0, 1, or 2 because if it were 3 or more, there would be a cube in t, which is impossible. Using the alphabet $\Gamma = \{0, 1, 2\}$, we append the corresponding number of b's, continuing this process indefinitely.

$$t = abbabaabbaababbabaababbaabbabaab\ldots$$
$$u = 2102012101202102\ldots$$

Definition 4.13. We define a new morphism $\pi : \{0, 1, 2\}^* \to \{a, b\}^*$ as follows:

$$\pi(0) = a$$
$$\pi(1) = ab$$
$$\pi(2) = abb$$

This morphism generates an a with n copies of b afterward when evaluated on n (where n is 0, 1, or 2). Importantly, t starts with a, and by the nature of how we constructed u by counting how many b's appear between instances of a, we have that $\pi(u) = t$.

Proposition 4.14. *The ternary infinite word u is square-free.*

Proof. For the sake of contradiction, assume that there exists a square vv in u. Consider vv and whichever letter immediately follows vv. It is either a 0, 1, or 2, but we will call it x. Since we have assumed vvx is in u, and we know that $\pi(u) = t$, it follows that $\pi(vvx)$ is a factor of t.

Morphisms are distributive over concatenation, therefore $\pi(vvx) = \pi(v)\pi(v)\pi(x)$ appears in t. No matter what v and x are, their images under π must start with a, because all of $0, 1$, and 2 map to words starting with a. If we let $\pi(v) = aw_1$ and $\pi(x) = aw_2$ for possibly empty words w_1, w_2, then we know that $aw_1aw_1aw_2$ appears in t. However, this word contains the factor aw_1aw_1a. Depending on if w_1 is empty or nonempty, this factor is either a cube or an overlap, respectively. However, t is cube-free and overlap-free, so this is a contradiction. Hence u is square-free. ■

5. The Unavoidable Zimin Patterns

In the previous sections, we have touched on patterns like squares, cubes, and overlaps, which can all be avoided by words on a sufficiently large alphabet. Now, we will explore an infinite family of patterns called the *Zimin patterns*, also known as the *sesquipowers* in the pattern alphabet. Unlike the patterns we have previously observed, all of them are unavoidable, meaning no infinite word can avoid them, no matter how large the alphabet is.

Definition 5.1. Assume the pattern alphabet Δ has infinitely many variables. We recursively define the nth Zimin pattern Z_n as follows:

- $Z_0 = \varepsilon$
- $Z_{n+1} = Z_n x Z_n$, where x is a new variable in Δ that has not been used in Z_n.

The first few Zimin patterns are shown below for illustrative purposes, and we use Greek letters as variables:

- $Z_1 = \alpha$
- $Z_2 = \alpha\beta\alpha$
- $Z_3 = \alpha\beta\alpha\gamma\alpha\beta\alpha$

Theorem 5.2 ([Zim84])**.** *All Zimin patterns are unavoidable.*

Proof. This can be shown by induction on n. Our inductive hypothesis is that given a size k alphabet, all words of some finite length l encounter Z_n.

The base case is when $n = 1$. $Z_1 = \alpha$ is unavoidable for all words of length 1, as we can substitute that letter to represent α and find an occurence of the pattern.

Now we prove the inductive step. Given an infinite word w on a k-letter alphabet, we consider w in blocks of l letters each, with one letter of space between each block. There are k^l distinct l-letter words over the k-letter alphabet, and importantly this quantity is finite. After k^l+1 blocks with 1-letter gaps between (or $(l+1)(k^l+1)-1$ letters), by the Pigeonhole Principle, there must be one block of length l that was repeated. If the block is repeated, the instance of Z_n that the particular block encountered was also repeated. And because we left 1 letter of space between consecutive blocks, we are guaranteed that the word between the two instances of Z_n is nonempty. Therefore, we can set the new variable x equal to this word between the copies of Z_n, and we have found an occurrence of $Z_n x Z_n = Z_{n+1}$ in w. This completes the inductive proof. ■

Zimin himself characterized all unavoidable patterns in the following theorem:

Theorem 5.3 (Zimin's Theorem)**.** *A pattern p is unavoidable if and only if it is encountered in a Zimin pattern.*

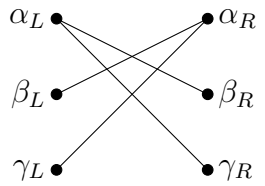
It is easy to show that p is unavoidable if it is encountered in a Zimin pattern. Once a Zimin pattern is encountered, the factors within that pattern are also necessarily encountered. It is much harder to prove that p being unavoidable implies it is encountered by some Zimin pattern.

Example. Although the pattern xyz is not explicitly contained in $\alpha\beta\alpha$ (since Z_2 is not the concatenation of three distinct variables), it is still true that xyz is still encountered if we let x and z map to the same word.

Zimin used a technique called the Zimin algorithm to reduce patterns to smaller patterns, and then proved that if reduction resulted in an unavoidable pattern, the original pattern was itself unavoidable. We present an abridged version of the reduction process. All words following the pattern $\alpha\beta\alpha$ must also follow xyz. This was first presented in [Zim84] but we present it more closely to how it was presented in [Lot02].

For a pattern $p \in \Delta^*$, define P as the set of all distinct variables occurring in p. Then we construct the *adjacency graph* $G(p)$, which is a bipartite undirected graph, where the set of vertices is simply P repeated twice, P_L and P_R. Edges are drawn between vertices x_L and y_R if and only if xy occurs in the pattern p. In other words, an edge is drawn for each two-variable pattern that p contains.

Example. The adjacency graph of the pattern $\alpha\beta\alpha\gamma\alpha\beta$ would have four edges, and is depicted below:



We then define a *free set* F of P as a subset of P such that for all pairs of variables $x, y \in F$, there is no path between x_L and y_R that travels along the edges of $G(p)$. In other words, the vertices x_L and Y_R are on separate connected components of the adjacency graph, regardless of the choice of x and y. In the above example, $\{\alpha\}, \{\beta\}, \{\gamma\}, \{\beta, \gamma\}$ are all the possible free sets.

The reduction of a pattern p given a free set F is simply the resulting pattern q from deleting all letters in F from p. We could then perform the same process on q. If there exists at least one sequence of these reductions (noting the changes in the adjacency graph as we go along) takes a pattern p to the empty word, we call p reducible, and irreducible otherwise.

Not all sequences of reductions will behave the same. Using our previous example, reducing with the free set $\{\alpha\}$ results in $q = \beta\gamma\beta$, which could be reduced further using the free sets $\{\beta\}$ then $\{\gamma\}$ in that order. However, using $\{\beta, \gamma\}$ first would result in the irreducible $\alpha\alpha\alpha$. Thus, it is necessary to check all possible reduction sequences before deeming a pattern irreducible.

This idea can be used to prove Zimin's theorem by showing an equivalent result:

Theorem 5.4. *A pattern p is unavoidable if and only if it is reducible.*

The proof of this is complicated, but one direction of the proof relies on induction on the size of the alphabet.

6. Extremal Words

This section is devoted to exploring a more recent field of thought in pattern avoidance, extremal words, which were first studied by Grytczuk, Kordulewski, and Niewiadomski in 2020 when the trio looked at extremal square-free words. Unlike the traditional notions of pattern avoidance, we primarily work with finite words.

Definition 6.1. An *extension* of a word w is the word resulting from an addition of a single letter to w at any location. More formally, a word is an extension of w if there exist possibly empty words w_1, w_2 and letter x such that the extension can be written as w_1xw_2 and $w_1w_2 = w$.

Example. Extensions of the English word "par" over the English alphabet include the words "spar," "pear," "pair," and "part." Each example highlights an extension at a different location.

Definition 6.2. A word w is *extremal pattern-free* if w itself avoids the pattern, but all extensions of w encounter the pattern. In other words, w cannot be extended while preserving its pattern-free property.

We will mainly look at extremal square-free words.

Claim 6.3. *The only extremal square-free binary words are aba and bab.*

Proof. By 4.5, it suffices to check just words of length 1 through 3. Furthermore, all possibilities must alternate in order to be square-free. The words of length 1 and 2 can be extended while staying square-free (e.g. *a* to *ab*, or *ba* to *bab*), but the words of length 3,*aba* and *bab*, cannot be extended without staying square-free, and hence they are extremal. ■

Example. If we look at words over a ternary alphabet, it turns out the shortest extremal square-free word here is the 25-letter word abcabacbcabcbabcabacbcabc. The verification, discovery of this solution, and the proof that there are no shorter extremal square-free ternary words were all done by computer search.

Furthermore, Grytczuk, Kordulewski, and Niewiadomski also showed that there are infinitely many extremal square-free ternary words, and that there exist extremal square-free ternary words for an arbitrary length that is sufficiently large. The proof that infinitely many of these words exist is somewhat complicated, but uses a recursive construction considering walks on a directed graph of vertices, a morphism that generates words based on the steps of the walk, and computer verifications. A proof in more detail can be found in [GKN20].

The proof that for a sufficiently large length (in particular, for all lengths at least 87) an extremal square-free ternary word exists, mainly uses the graph-walking idea, as well as building blocks called *nearly extremal* square-free words, words that have 1 or 2 extensions that still do not contain squares. Furthermore, these extensions should be additions of letters at the leftmost or the rightmost position.

The proof consists of finding two of these words that could be extended at both sides, such that the two have relatively prime lengths. We also need two ending words that only

have extensions on a single side. In particular, we can modify the words we have found with permutations of the alphabet $\{a, b, c\}$ as well as reversals, to find other suitable words.

The idea of the construction is to keep chaining these words together, and use the Chicken McNugget theorem to show that for a large enough length, the words of relatively prime lengths can be combined in such a way to reach the exact length we desire (after adding ending words). Then, we add our ending words to make sure that no extensions remain. The bound was then lowered to 87 by the use of a multitude of similar constructions with differing word lengths, as well as brute-force computer searching.

7. Open Problems

This section is devoted to lightly exploring some open problems and conjectures, particularly those related to extremal words, and not just extremal square-free ones.

Conjecture 7.1 ([GKN20])**.** *No extremal square-free words exist for alphabets of size 4 or larger.*

In [GKP21] there is computer evidence that a single letter on a 4-letter alphabet could be extended to a length of 50,000 while staying square-free, suggesting that it is always possible to find a square-free extension and that no extremal square-free words exist in this alphabet. Furthermore, their extension algorithm only considered extensions at the last or second to last position, suggesting that even more extensions are possible if the entire word were to be considered.

At the time this conjecture was made, there was no knowledge of an upper bound, in other words, an alphabet size large enough that there was always a square-free extension available. However, in [HZ22], two MIT students proved that there exist no extremal square-free words for alphabets of size 17 or larger, providing an upper bound to the problem. Their proof relied mainly on extensive casework and proof by contradiction.

However, it is still open whether there exist extremal-square free words on alphabets of size 4 through 16.

Of course, the notion of extremal words does not only apply to the property of being square-free. The following are a few more open questions revolving around other patterns:

- Do any extremal cube-free words exist on binary alphabets?
- It was proven in [MRS20] that there are infinitely many binary words that are extremal overlap-free. It is unknown, however, if there are any extremal overlap-free words over larger alphabets.
- In a similar vein, it is also unknown if there exist any extremal cube-free words over the binary alphabet.

Acknowledgments

The author would like to thank Simon Rubinstein-Salzedo and Hongxu Chen for helpful conversations regarding this paper.

References

[BK04] Jean Berstel and Juhani Karhumäki. Combinatorics on words – a tutorial. In *Current trends in theoretical computer science. The challenge of the new century. Vol. 2: Formal models and semantics.*, pages 415–475. River Edge, NJ: World Scientific, 2004.

[GKN20] Jarosław Grytczuk, Hubert Kordulewski, and Artur Niewiadomski. Extremal square-free words. *Electron. J. Comb.*, 27(1):research paper p1.48, 9, 2020.

[GKP21] Jarosław Grytczuk, Hubert Kordulewski, and Bartłomiej Pawlik. Squarefree extensions of words. *J. Integer Seq.*, 24(8):article 21.8.7, 17, 2021.

[HZ22] Letong Hong and Shengtong Zhang. No extremal square-free words over large alphabets. *Combinatorial Theory*, 2(1), March 2022.

[Lot02] M. Lothaire. *Algebraic combinatorics on words*, volume 90 of *Encycl. Math. Appl.* Cambridge: Cambridge University Press, 2002.

[MRS20] Lucas Mol, Narad Rampersad, and Jeffrey Shallit. Extremal overlap-free and extremal β-free binary words. *Electron. J. Comb.*, 27(4):research paper p4.42, 15, 2020.

[Zim84] A. I. Zimin. Blocking sets of terms. *Math. USSR, Sb.*, 47:353–364, 1984.