June 2017

# Patterns in Words Related to DNA Rearrangements

Lukas Nabergall
*University of South Florida*, lnabergall@mail.usf.edu

# Patterns in Words Related to DNA Rearrangements

by

Lukas Nabergall

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Arts
Department of Mathematics and Statistics
College of Arts and Sciences
University of South Florida

Co-Major Professor: Nataša Jonoska, Ph.D.
Co-Major Professor: Masahiko Saito, Ph.D.
Dmytro Savchuk, Ph.D.

Date of Approval:
June 28, 2017

Keywords: Reduction operations, Double occurrence words, Pattern indices, Nesting index, Ciliates

# Acknowledgments

I would like to thank my advisors, Dr. Nataša Jonoska and Dr. Masahiko Saito, for their extensive support and guidance in the development and completion of this thesis. I would like to express my gratitude to Dr. Dmytro Savchuk for serving on the defense committee and carefully reviewing this thesis. I also could not have completed this thesis without the support of the other members of the USF Math-Bio Lab, including Jasper Braun, Daniel Cruz, Jonathan Burns, Derek Covert, and Mustafa Hajij, as well as all of the faculty of the Department of Mathematics and Statistics at the University of South Florida. I would also like to thank all of my family and friends who made this journey possible, enjoyable, and enriching. The work presented here has been supported in part by the National Institutes of Health under Grant No. R01GM109459.

# Table of Contents

List of Tables ii

List of Figures iii

Abstract iv

1 Introduction 1

1.1 Main Results and Thesis Organization . . . . . . . . . . . . . . . . . . . . 3

1.2 Notation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

2 Generalized Patterns 8

2.1 Definitions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

2.2 Distances . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

2.3 Indices . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

3 Patterns in Double Occurrence Words 27

3.1 Definitions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

3.2 Complete Patterns . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

3.3 Pattern Recurrence Index . . . . . . . . . . . . . . . . . . . . . . . . . . 36

3.4 Nesting Index . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43

4 Highly Scrambled Genome Rearrangements in *O. trifallax* 51

5 Conclusion 54

References 56

# List of Tables

4.1 Compared with an identically distributed random sample of 22 words, the 22 highly scrambled cases exhibit significantly lower averages on indices that include the tangled cord pattern. . . . . . . . . . . . . . . . . . . . . . . . . . . . 53

# List of Figures

2.1 Visual representations of the ten cases. Red intervals represent $u_1$ and $u'_1$, while green intervals represent $u_2$ and $u'_2$. . . . . . . . . . . . . . . . . . . . . . . . 15

3.1 A schematic depicting the process of removing insertions that precede deletions in a path from $u$ to $v$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

3.2 A reduction tree for the reduction $(T_7, T_2T_4, T_2T_2T_1, \epsilon)$ of $T_7$. . . . . . . . . . . 39

3.3 The chord diagram associated with the words 121323, 123213, and 123132. . . 46

3.4 The chord diagrams of the words 12321434 and 121323, respectively. . . . . . . 46

3.5 The chord diagram of $v_5 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 9, 7, 5, 3, 1, 2, 4, 6, 8, 10$. . . . 49

4.1 DNA rearrangement in *Oxytricha trifallax*. The MDS sequence $\overline{\mathrm{M}}_1\mathrm{M}_2\mathrm{M}_3\mathrm{M}_5\mathrm{M}_4$ with pointer sequence 11223434 (left) can be represented as a double occurrence word and visualized using a chord diagram (right). . . . . . . . . . . . . . . . . 51

# Abstract

Patterns, sequences of variables, have traditionally only been studied when morphic images of them appear as factors in words. In this thesis, we initiate a study of patterns in words that appear as subwords of words. We say that a pattern appears in a word if each pattern variable can be morphically mapped to a factor in the word. To gain insight into the complexity of, and similarities between, words, we define pattern indices and distances between two words relative a given set of patterns. The distance is defined as the minimum number of pattern insertions and/or removals that transform one word into another. The pattern index is defined as the minimum number of pattern removals that transform a given word into the empty word. We initially consider pattern distances between arbitrary words. We conjecture that the word distance is computable relative the pattern $\alpha\alpha$ and prove a lemma in this direction. Motivated by patterns detected in certain scrambled ciliate genomes, we focus on double occurrence words (words where every symbol appears twice) and consider recursive patterns, a generalization of the notion of a pattern which includes new types of words. We show that in double occurrence words the distance relative so-called complete sets of recursive patterns is computable. In particular, the pattern distance relative patterns $\alpha\alpha$ (repeat words) and $\alpha\alpha^R$ (return words) is computable for double occurrence words. We conclude by applying pattern indices and word distances towards the analysis of highly scrambled genes in *O. trifallax* and discover a common pattern.

# 1 Introduction

A word is a sequence, finite or countable, of elements from a finite or countable set $\Sigma$ known as an alphabet. For example, the word 102120102 is a sequence of symbols from the alphabet $\{0, 1, 2\}$, while the word "electric" is a sequence of letters from the English alphabet $\{a, b, \ldots, y, z\}$. The study of words dates back to at least the work of Axel Thue [4, 5] in the early 20th century on square-free words, those that do not contain any factor twice in a row, and has numerous applications to a variety of fields, including automata theory, symbolic dynamical systems, coding theory, natural language processing, bioinformatics, and many more [3].

Patterns, sequences of variables from a set $X$, are primary objects of study in combinatorics on words. A pattern $p$ is said to appear in a word $u$ if there exists a morphism $f\colon X^* \to \Sigma^*$ such that $f(p)$ is a factor of $u$ (see Section 1.2 for definitions of this notation). Patterns in words have a long history. For example, the square $uu \in \Sigma^*$ corresponding to a pattern $\alpha\alpha \in X^*$ is an archetypal pattern. In the study of such patterns, a classical result due to Thue is that an infinite word without a square factor is only possible over alphabet with at least three symbols. This result answers the question of when the pattern $\alpha\alpha$ is avoidable, that is, when there are infinitely many words in $\Sigma^*$ that do not contain an appearance of $\alpha\alpha$. In this language, Thue's theorem says that $\alpha\alpha$ is avoidable if and only if $|\Sigma| \geqslant 3$.

The concept of avoidability of patterns was first introduced by Bean, Ehrenfeucht, and McNulty [6] and studied by many authors, including Zimin [7, 8], Baker, McNulty, and Taylor [9], Schmidt [10, 11], Cassaigne [12, 13], and others. Thue's theorem can be stated more precisely by introducing the concept of k-avoidability: a pattern $p$ is $k$-avoidable if $p$ is avoidable on any alphabet of size $k$. This leads to the definition of the avoidability index $\mu(p)$ of a pattern $p$, the smallest integer $k$ such that $p$ is $k$-avoidable. If $p$ is unavoidable, then $\mu(p) := \infty$. Now Thue's theorem becomes the simple assertion that $\mu(\alpha\alpha) = 3$. Computing the avoidability index of a given pattern is a difficult problem of primary interest in the study of patterns in words. Although the problem of determining whether a given pattern is avoidable has been solved [6, 7], it remains an open problem to determine whether a given pattern is $k$-avoidable and hence compute its avoidability index. In this direction, Schmidt began to answer this question for binary patterns, patterns on two variables. This work was completed by Cassaigne, who was able to completely classify binary patterns according to their avoidability index (which in this case can be either 2, 3, or $\infty$). In 2006, building on Cassaigne's work, Ochem [14] completed the classification of ternary patterns. For arbitrary patterns, only bounds on the avoidability index have been obtained; see e.g. [9, 2].

Other problems related to patterns and avoidability that have been studied include bounding the length of patterns of a given avoidability index and studying the growth, topological structure, and other properties of a set of words avoiding a given pattern. For results in this direction, see Cassaigne and Roth [12, 13], Baker, McNulty, and Taylor [9], and Currie [15], among others. Although avoidability is central to the study of patterns, there are many other problems involving patterns that have been explored in the combinatorics on words literature. Knuth, Morris, and Pratt [20], Abrahamson [16], Baker [19], Apostolico and Galil [18], Amir, Aumann, Cole, Lewenstein, and Porat [17], Amir and Nor [21], and others have studied the pattern matching problem, which seeks to find efficient algorithms for finding all the occurrences of a given pattern in a word. An inverse problem, finding patterns common to a set of words, has also been explored (see e.g. Angluin [22] and Ng and Shinohara [23]), as has the NP-complete problem of determining whether a word is an instance of a given pattern (see e.g. Reidenbach and Schmid [24], Fernau and Schmid [25], and Fernau, Manea, Mercas, and Schmid [26]).

The problem of approximate string matching, which has applications to computational biology, signal processing, text retrieval, and many other fields, is also concerned with locating a pattern in a given word (or string) [27]. In this context, a pattern is simply a word from $\Sigma$ and the goal is to find all factors of a given word that match a pattern with up to $k$ errors. Given a distance function $d\colon \Sigma^* \times \Sigma^* \to \mathbb{R}$, we say that two words $u$ and $v$ match up to $k$ errors if $d(u, v) \leqslant k$. The distance function $d$ is typically taken to be a type of edit distance, where $d(u, v)$ is defined as the minimal cost of a sequence of edit operations, each with an associated cost, that transform $u$ into $v$ (and if no such sequence exists, then

$d(u, v) := \infty$). There are four primary edit operations considered in the literature: insertion of a letter, deletion of a letter, substitution of one letter for another, and, less commonly, transposition of two adjacent letters; the most widely studied edit distances are defined using some subset of these four edit operations.

Perhaps the oldest edit distance, the Levenshtein distance, was defined by Vladimir Levenshtein in 1965 [28] as the edit distance allowing insertions, deletions, and substitutions with cost 1. Other commonly considered edit distances include the Hamming distance [29] (substitution), longest common subsequence distance [30] (insertion and deletion), and the Damerau-Levenshtein distance [31] (insertion, deletion, substitution, and transposition). See the work of Ukkonen [32, 33, 34], Wagner and Fischer [35], Baeza-Yates and Navarro [36, 37], Myers [38], and others for fast algorithms for approximate string matching and other results on this problem.

In this thesis, we generalize the traditional notion of a pattern and consider the problem of describing the complexity of the appearance of generalized patterns in words. Along the lines of the literature on edit distances and approximate string matching, we also study word distances defined via edit operations involving inserting and removing generalized patterns, not just letters or subwords.

## 1.1 Main Results and Thesis Organization

In Section 1.2, we begin by describing the standard notation from the combinatorics on words literature used throughout the thesis. The remainder of the thesis is separated into three chapters.

In the first chapter, we study a generalization of the traditional notion of a pattern which allows for subword appearances in a word. We begin by defining generalized patterns in Section 2.1. We then introduce our primary tool in the study of appearances of generalized patterns in words: reductions of a word, defined by the iterative removal of pattern instances via so-called reduction operations[1]. In Section 2.2, we use reductions to define paths between two words, essentially sequences of edit operations, that is, pattern instance removals and insertions, transforming one word into another. Paths naturally induce a distance $d_P$ between

[1] Traditionally known as “edit operations”.

words by defining $d_P(u, v)$ to be the minimum length of all paths between words $u$ and $v$ relative a fixed set of patterns $P$. These word distances serve as a measure of the similarity of two words relative a given set of patterns. In order for this to be practically useful, we need to be able to compute the word distance. Although this problem appears largely infeasible in the general case, for arbitrary words and arbitrary sets of patterns, we do make progress on computing the word distance for a particularly simple and practically useful pattern, the repeat word $\alpha\alpha$. In this direction, we prove the following theorem, which essentially affirms that if we can remove and/or insert two pattern instances two transform $u$ into $v$, those pattern instances do not need to be "too large"; in particular, they are bounded by a constant multiple of $|u| + |v|$.

**Theorem 2.2.8.** *Let $P = \{\alpha\alpha\}$, $u$ and $v$ be words, and suppose there exists a minimal path $\rho$ from $u$ to $v$ of the form $((u, w), (w, v))$ for some word $w$. Then there exists such a $w$ satisfying*

$$|w| \leqslant 4(|u| + |v|).$$

We then outline a possible proof of our main conjecture:

**Conjecture 2.2.9.** *Let $P = \{\alpha\alpha\}$. For all words $u$ and $v$, $d_P(u, v)$ is computable.*

We are also interested in studying how well a set of patterns describes (or "generates") a given word. In Section 2.3, we introduce a measure of the complexity of such a description by defining pattern indices $I_P(u)$, the minimum length of all reductions from $u$ to the empty word $\epsilon$.

As Conjecture 2.2.9 indicates, in general it is very difficult to prove useful, nontrivial statements about word distances and pattern indices for arbitrary words or arbitrary sets of patterns. Without a means to make more progress in this direction, in Chapter 3 we focus our efforts on analyzing biologically-motivated double occurrence words, words with exactly two occurrences of each letter. We also further generalize our notion of a pattern to include more interesting pattern languages by defining recursive patterns, essentially sequences of patterns that can be recursively generated by iteratively adding two occurrences of a variable. There exist particularly well-behaved sets of recursive patterns, so-called complete sets of recursive patterns, for which we can make significant progress on a number of important problems. The repeat word $\alpha\alpha$ and return word $\alpha\alpha^R$ are the two most notable examples of complete recursive patterns. Our primary result is the following theorem:

**Theorem 3.2.10.** *Let $\Pi$ be a complete set of recursive patterns. For all double occurrence words $u$ and $v$ in the same connected component, there exists a minimal path $\rho$ between $u$ and $v$ of the form $(r_1)$ or $(r_1, r_2^R)$, where $r_1$ and $r_2$ are reductions. In particular, $d_P(u, v)$ is computable.*

Not only does this imply that the word distance is computable for double occurrence words and complete sets of recursive patterns, but it is essentially as easy to compute as the pattern index by searching through all possible reductions of $u$ and $v$. In Sections 3.3 and 3.4, we analyze two pattern indices associated with the repeat word and return word, the pattern recurrence index $PI$ and the nesting index $NI$. We prove a variety of results about these indices, including a computation of the pattern recurrence index and nesting index of the tangled cord, a complex recursive pattern or, equivalently, type of word with biological applications.

We conclude the thesis in Chapter 4 with an analysis of twenty-two highly scrambled DNA rearrangements that occur in *Oxytricha trifallax* during sexual reproduction in the production of a protein-cording macronucleus from the nonfunctional micronucleus. Note that this work of the author was first described in [43]. Every DNA rearrangement in *O. trifallax* can be represented by a double occurrence word. Although it was previously discovered that the vast majority of these genome rearrangements are nested concatenations of repeat words and return words [41, 42], there are twenty-two rearrangements which retain at least four letters after iterative removal of all repeat words and return words. Since the repeat word and return word do not well-described these highly scrambled rearrangements, we search for other recursive patterns hidden within these rearrangements. Using the word distances and pattern indices studied in Chapters 2 and 3, we identify the tangled cord as a commonly-occurring recursive pattern. Biologically, this indicates that during the rearrangement process DNA strands may often be folding into a "tangled cord" configuration in order to produce new strands.

## 1.2 Notation

An alphabet is a finite or a countable set $\Sigma$ whose elements are called *symbols* or *letters*. A *word* $u$ over $\Sigma$ is a finite sequence of symbols in $\Sigma$ and $\Sigma^*$ denotes the set of all words. If we write $u = a_1 \cdots a_n$ for some $a_i \in \Sigma$, then $n$ is the length of $u$, denoted $|u| = n$. The empty word is denoted by $\epsilon$ and has length 0. A subset $L \subseteq \Sigma^*$ is called a *language*. The set of all words of length at least $n$ is denoted by $\Sigma^{\geqslant n}$; in particular, we write $\Sigma^+$ for $\Sigma^{\geqslant 1}$, the set of all words of positive length. Furthermore, for a word $u$ over $\Sigma$ we let $\Sigma[u]$ denote the alphabet composed of all symbols appearing in $u$. The reverse of $u$ is denoted $u^R = a_n \cdots a_2 a_1$ and we write

$$u^k = \underbrace{uu \cdots u}_{k \text{ copies}}$$

to denote the $n$-fold concatenation of $u$. The number of appearances of a symbol $a$ in a word $u$ is denoted by $|u|_a$.

A word $v$ is a *factor* of $u$ if there exist $u_1, u_2 \in \Sigma^*$ such that $u = u_1 v u_2$. In this case, we write $v \sqsubseteq u$ and $u(v^{-1}) = u_1 u_2$. Note that $u(v^{-1})$ does not necessarily uniquely determine $u_1 u_2$. If $u_1 = \epsilon$, then we say that $v$ is a *prefix* of $u$, while if $u_2 = \epsilon$, we say that $v$ is a *suffix* of $u$. We say that $v = v_1 \cdots v_k$ is a *subword* of $u$, written $v \preceq u$, if $u = u_0 v_1 u_1 \cdots v_k u_k$ for some $u_i \in \Sigma^*$. As one might expect, we write $v \prec u$ when $v \neq u$ and use $u \succeq v$ to denote $u$ as a word that contains $v$ as a subword. We point out the distinction between a subword and a factor. In the literature, the term "subword" may often be used to denote a factor, rather than a subsequence (as we use it here). Our notation follows several books from the reference literature on combinatorics of words [1, 2].

Throughout this thesis, $X$ will denote a set of *variables* such that for every variable $\alpha \in X$ there is a variable $\alpha^R \in X$ distinct from $\alpha$ satisfying $(\alpha^R)^R = \alpha$. The elements in $X$ are denoted by greek symbols $\alpha$, $\beta$, etc. For a word $p \in X^*$, we set $X[p] = \{\alpha \in X \mid |p|_\alpha \geqslant 1\}$, the set of variables that appear in $p$.

A function $f\colon X \to \Sigma^*$ naturally extends to a morphism $f\colon X^* \to \Sigma^*$. We say that $f$ is *reverse-preserving* on $X$ if $f(\alpha)^R = f(\alpha^R)$ for all $\alpha \in X$. In the rest of the text we assume that all functions are reverse-preserving on $X$. We say that words $u \in \Sigma_1^*$ and $v \in \Sigma_2^*$ are *equivalent*, and write $u \equiv v$, if there exists a bijection $f\colon \Sigma_1 \to \Sigma_2$ such that $f(u) = v$ for the induced morphism $f\colon \Sigma_1^* \to \Sigma_2^*$.

# 2 Generalized Patterns

## 2.1 Definitions

Traditionally, patterns have only been considered as factors in word. In this chapter, we extend the notion of a pattern appearing in a word to include subwords.

**Definition 2.1.1.** *Let $X$ be a set of variables. A* pattern *$p$ is an element of $X^*$. For a word $w \in \Sigma^*$, $p = \alpha_1 \cdots \alpha_n$* appears *in $w$ if there is a reverse-preserving map $f\colon X[p] \to \Sigma^+$ and, for $0 \leqslant i \leqslant n$, $z_i \in \Sigma^*$ such that*

$$w = z_0 f(\alpha_1) z_1 f(\alpha_2) \cdots z_{n-1} f(\alpha_n) z_n.$$

*The words $z_1, \ldots, z_{n-1}$ are called* gaps *and the word $f(\alpha_1) \cdots f(\alpha_n)$ is an* instance *of $p$ in $w$.*

For clarity, a subword of a pattern is called a *subpattern*. If $u = f(\alpha_1) \cdots f(\alpha_n)$ is an instance of a pattern $p = \alpha_1 \cdots \alpha_n$, and $u' = f'(\alpha_1) \cdots f'(\alpha_n)$ is an instance of $p$ in $u$ satisfying $f'(\alpha_i) \sqsubseteq f(\alpha_i)$, then we call $u'$ a *sub-instance* of $p$ in $u$. If $u' \neq u$, then it is a *proper sub-instance*. If all of $z_1, \ldots, z_{n-1}$ are the empty word, then we say that $p$ appears *strictly* in $w$ and that there are no gaps in the appearance of $p$. Note that if $u \equiv v$, then $p$ appears in $u$ if and only if $p$ appears in $v$. If $p$ appears in $w$ then $f(p) = f(\alpha_1) \cdots f(\alpha_n)$ is a subword of $w$, and if it appears strictly in $w$ then it is a factor of $w$. When a pattern $p$ appears in such a way that $|f(p)| = |p|$, that is, $f$ maps variables to symbols, then we say that $p$ appears *literally* in $w$.

**Example 2.1.2.** *The pattern $p = \alpha\alpha$ appears in the word $w = abcabd$, where $\alpha \mapsto ab$. When $p$ appears strictly it is called a* square. *In the above example the appearance is not strict because $c$ is a gap. Another instance of $p$ in $w$ is $bb$; this is a literal appearance of $p$. When $p$ appears both strictly and literally as a single instance it is called a* loop.[1]

[1] See Chapter 4 for the motivation behind using this term.

**Definition 2.1.3.** *An instance of the pattern $\alpha\alpha$ is called a* repeat word *and an instance of the pattern $\alpha\alpha^R$ is called a* return word*. We will often refer to the patterns $\alpha\alpha$ and $\alpha\alpha^R$ themselves as the repeat word and return word, respectively.*

We note that an appearance of a pattern can be very different than the strict appearance. For every finite alphabet $\Sigma$ there is $n$ such that a pattern $\alpha^k$ appears in all words in $\Sigma^{\geqslant n}$,[2] however, it is well known that this is not the case for strict appearance. If $\Sigma$ contains at least three symbols, then, for every $n \in \mathbb{N}$, $\Sigma^n$ contains words where $\alpha^2$ does not appear strictly [2].

**Example 2.1.4.** *The pattern $\alpha\alpha^R$ appears in the word abcbad, where $\alpha \mapsto ab$. This pattern also appears literally as bb.*

**Lemma 2.1.5.** *For every pattern $p$ there is a pattern $q_p$ such that for every word $w \in \Sigma^*$, if $q_p$ appears strictly in $w$ then $p$ appears in $w$. Furthermore, if $p$ appears with positive gaps then $q_p$ appears strictly in $w$.*

*Proof.* For $p = \alpha_1 \cdots \alpha_k$, we set $q_p = \alpha_1\beta_1\alpha_2 \cdots \beta_{k-1}\alpha_k$, where $\beta_1, \ldots, \beta_k$ are all distinct. ∎

In the following, we define the usual set operations on patterns.

**Definition 2.1.6.** *Let $p_1$ and $p_2$ be patterns. Then we write*

$$p_1 \cap p_2 = \{p \mid p \sqsubseteq p_1 \text{ and } p \sqsubseteq p_2\} \quad \text{and} \quad p_1 - p_2 = \{p_1(p^{-1}) \mid p \in p_1 \cap p_2\}.$$

We present some additional preliminary definitions and lemmas which use the notation from Definition 2.1.1.

**Definition 2.1.7.** *Given a set of patterns $P$ and a word $w$, we say that $w'$ is obtained from $w$ by* reduction operation $\vdash_p$ *if*

$$w = z_0 f(\alpha_1) z_1 f(\alpha_2) \cdots z_{n-1} f(\alpha_n) z_n \quad \text{and} \quad w' = z_0 z_1 \cdots z_n$$

*for some instance $f(\alpha_1) \cdots f(\alpha_n)$ of $p = \alpha_1 \cdots \alpha_n$ in $P$. In this case, we write $w \vdash_p w'$ or $w' = w - u$.*

[2] In particular, by the pigeonhole principle, $n = (k-1)|\Sigma| + 1$.

**Definition 2.1.8.** *Given a set of patterns $P$, define a* reduction *of a word $u$ to be a sequence $r = (w_0, w_1, \ldots, w_n)$ such that*

1. *$w_0 = u$,*
2. *and for all $1 \leqslant i \leqslant n$, there exists $p \in P$ such that $w_{i-1} \vdash_p w_i$.*

*If such a reduction exists, we say that $u$ can be* reduced *to $w_n$ in $n$ steps and the reduction has* size *or* length *$n$, written $|r|$. Furthermore, we say that the reduction is with $P$.*

**Example 2.1.9.** *The sequence $(abcdabcece, abceab, ce)$ is a reduction of length $2$ of the word $abcdabcece$ with $P = \{\alpha\alpha\}$.*

If $u$ can be reduced to the empty word $\epsilon$, then we say that $P$ *reduces* $u$. If $u$ can be reduced to $v$ with a set of patterns $P$, then we say that $v$ *expands* to $u$ with $P$. In that case, $r^R = (w_n, \ldots, w_0)$ is the *reverse* of the reduction $r$ and, naturally, we set $|r^R| = |r|$.

**Definition 2.1.10.** *A set of patterns $P$ is* confluent for a word *$u$ if for any reduction $r = (w_0, w_1, \ldots, w_n)$ of $u$, $P$ reduces $w_n$. Then $P$ is* confluent *if it is confluent for all words $u$ that are reduced by $P$.*

**Example 2.1.11.** *Patterns $p = \alpha\alpha$ and $p' = \alpha\alpha^R$ are confluent since $p$ (or $p'$) reduces a word to $\epsilon$ if and only if every symbol in the word appears an even number of times. So every reduction of a word with $P = \{p, p'\}$ keeps the parity of the number of occurrences of any symbol the same. However $q = \alpha\beta\alpha$ is not confluent since $ababba \vdash_q \epsilon$ by setting $\alpha \mapsto a$ and $\beta \mapsto babb$. On the other hand, $ababba \vdash_q aa$ by setting $\alpha \mapsto b$ and $\beta \mapsto ab$, and $aa$ cannot be further reduced by $q$.*

**Lemma 2.1.12.** *For a pattern $p$ with $|p| \geqslant 2$, if there exists $\beta \in X_p$ such that $|p|_\beta = 1$, then $p$ is not confluent.*

*Proof.* Suppose $p = p_1\beta p_2$ where $p_1$ and $p_2$ are patterns that do not contain $\beta$ and consider the word $w = a^{|p_1|+1}ba^{|p_2|+1}$. Then by setting $p_1 \mapsto a^{|p_1|}$, $p_2 \mapsto a^{|p_2|}$, and $\beta \mapsto aba$, we obtain $w \vdash_p \epsilon$. On the other hand, setting $p_1 \mapsto a^{|p_1|}$, $\beta \mapsto ab$, and $p_2 \mapsto a^{|p_2|}$, we obtain $w \vdash_p a$, and $a$ cannot be reduced by $p$. ∎

## 2.2 Distances

Reductions yield the notion of a *path* between two words.

**Definition 2.2.1.** *Given a set of patterns $P$, define a* path *between words $u$ and $v$ to be a sequence $\rho = (r_1, \dots, r_k)$ such that*

1. *the first word in $r_1$ is $u$,*
2. *the last word in $r_k$ is $v$,*
3. *for all $1 \leqslant i \leqslant k$, $r_i$ is a reduction or the reverse of a reduction and the last word in $r_i$ is the first word in $r_{i+1}$.*

*We call $|\rho| := |r_1| + \cdots + |r_k|$ the length of $\rho$, and say that $\rho$ is composed of $k$ reductions.*

Note that if a set of patterns $P$ reduces words $u$ and $v$, then there exists a path between $u$ and $v$ and, in general, such a path is not unique. In light of the fact that we are now looking at both reductions and the reverses of reductions, we write $w \dashv_p w'$ if $w = w' - u$ for an instance $u$ of some pattern $p \in P$. In this case, we say that $w$ is obtained from $w'$ by the *removal* of $u$ and that $w'$ is obtained from $w$ by the *insertion* of $u$. Paths naturally induce a distance between words:

**Definition 2.2.2.** *For a set of patterns $P$, define the* word distance *$d_P$ between $u$ and $v$ by*

$$d_P(u, v) = \begin{cases} \min |\rho|, & \rho \text{ is a path between } u \text{ and } v, \\ \infty, & \text{there is no path from } u \text{ to } v. \end{cases}$$

*Where $P$ is understood, we simply use $d$ to denote the word distance. A path $\rho$ between $u$ and $v$ is called minimal if $|\rho| = d_P(u, v)$.*

**Example 2.2.3.** *Consider the pattern $p = \alpha\alpha$. Then $\{p\}$ reduces every word in which every symbol appears an even number of times. However, if a word $w$ contains a symbol an odd number of times then $\{p\}$ does not reduce $w$. In this case, there is no path from a word reduced by $\{p\}$ to $w$, so the distance between these words is $\infty$.*

We observe that $d_P$ does indeed satisfy the axioms of a distance function:

**Lemma 2.2.4.** *For all sets of patterns $P$, $d_P$ is a distance.*

*Proof.* Clearly $d_P$ is symmetric, non-negative, and $d_P(u,v) = 0$ if and only if $u = v$. Let $w_1, w_2, w_3$ be words. If there does not exist a path between $w_1$ and $w_2$ or between $w_2$ and $w_3$ then the triangle inequality holds trivially. Suppose there do exist minimal paths $(r_1, \ldots, r_k)$ and $(r'_1, \ldots, r'_{k'})$ between $w_1$ and $w_2$ and $w_2$ and $w_3$, respectively. Then

$$(r_1, \ldots, r_k, r'_1, \ldots, r'_{k'})$$

is a path between $w_1$ and $w_3$ of size

$$\sum_{i=1}^{k} |r_i| + \sum_{i=1}^{k'} |r'_i| = d_P(w_1, w_2) + d_P(w_2, w_3),$$

implying that

$$d_P(w_1, w_3) \leqslant d_P(w_1, w_2) + d_P(w_2, w_3)$$

and hence the triangle inequality holds, as desired.

■

We say that two words $u$ and $v$ belong to the same *connected component* in $\Sigma^*$ if there is a path from $u$ to $v$. Words within the same connected component are within finite distance of each other. Each set of patterns $P$ partitions $\Sigma^*$ into connected components $C_P[w]$ for $w \in \Sigma^*$, where

$$C_P[w] = \{u \mid d_P(w,u) < \infty\}.$$

If $P$ reduces $u$, then $u$ and $\epsilon$ are in the same connected component, that is,

$$\{u \mid P \text{ reduces } u\} \subseteq C_P[\epsilon].$$

**Example 2.2.5.** *For $P = \{\alpha\alpha, \alpha\alpha^R\}$ and $\Sigma = \{a, b\}$, the connected components are $C_P[\epsilon], C_P[a], C_P[b]$, and $C_P[ab] = C_P[ba]$.*

For arbitrary sets of patterns, bounding the word distance is largely infeasible. Yet by restricting to $P \subseteq \{\alpha\alpha, \alpha\alpha^R\}$, we can obtain a nice bound:

**Lemma 2.2.6.** *Let $u$ and $v$ be words and $P \subseteq \{\alpha\alpha, \alpha\alpha^R\}$. If $v \in C_P[u]$, then*

$$d_P(u,v) \leqslant |u| + \frac{|v|}{2} + 1.$$

*Proof.* Since $v \in C_P[u]$, for all $a \in \Sigma[u] \cap \Sigma[v]$, $|u|_a \equiv |v|_a \mod 2$, and for all $b \in \Sigma[u] \triangle \Sigma[v]$, the symmetric difference of $\Sigma[u]$ and $\Sigma[v]$, $|u|_b$ and $|v|_b$ must be even. We proceed to construct a path $\rho$ from $u$ to $v$. For each $a \in \Sigma[u]$, remove $\lfloor |u|_a/2 \rfloor$ pairs of $a$ from $u$. This results in a reduction $r$ from $u$ to $u'$, where each letter in $u'$ appears exactly once. Similarly, we construct a reduction $r'$ from $v$ to $v'$, where $v'$ is a permutation of $u'$ since all letters in $\Sigma[u] \triangle \Sigma[v]$ are removed from $u$ and $v$ during the reduction process. Hence $u'v'$ can be reduced to $\epsilon$ with $P$. Observe that $|r| \leq |u|/2$ and $|r'| \leq |v|/2$. Then a path $\rho'$ from $u'$ to $v'$ is obtained with the following: insert $v'v'$ as a prefix to $u'$ and obtain $v'v'u'$. Then perform a reduction on $v'u'$ to $\epsilon$ by removing each pair of symbols one at a time. The length of $\rho'$ is $|u'| + 1$. Since $|u'| = |u| - 2|r| = |v| - 2|r'|$, we conclude that $\rho = (r, \rho', r')$ is a path from $u$ to $v$ of length at most

$$\begin{aligned} |r| + |r'| + |u'| + 1 &= |r| + |r'| + |u| - 2|r| + 1 \\ &= |u| - |r| + |r'| + 1 \\ &\leqslant |u| + \frac{|v|}{2} + 1. \end{aligned}$$

■

Ideally, we would like to go significantly further and actually be able to compute the distance between arbitrary words. Although this seems completely infeasible for arbitrary sets of patterns (being a substantially harder problem than merely obtaining an upper bound), we make some progress in the direction of computing the word distance with $P = \{\alpha\alpha\}$. First, we require a key lemma.

**Lemma 2.2.7.** *Let $u$, $v$, and $w$ be words satisfying*

$$wu = vw. \tag{2.2.1}$$

*Then there exists a prefix $w'$ of $w$ such that*

$$w'u = vw' \quad \textit{and} \quad |w'| \leqslant |u|, |v|.$$

*Proof.* Suppose $|w| > |u| = |v|$, that is, there exists a word $u_1$ such that, by (2.2.1), $w = u_1u$. Then substituting this expression into (2.2.1), we infer that

$$u_1uu = vu_1u,$$

or $u_1u = vu_1$. If $|u_1| > |u| = |v|$, we set $w = u_1$ and repeat the above procedure, obtaining a smaller word $u_2$. Otherwise, if $|u_1| \leqslant |u| = |v|$, then $u_1$ is the desired prefix $w'$. Since $w$, $u$, and $v$ are finite words and this procedure reduces the size of $w$ each step, eventually this process will end in a finite number of steps. Hence the result holds.

■

**Theorem 2.2.8.** *Let $P = \{\alpha\alpha\}$, $u$ and $v$ be words, and suppose there exists a minimal path $\rho$ from $u$ to $v$ of the form $((u, w), (w, v))$ for some word $w$. Then there exists such a $w$ satisfying*

$$|w| \leqslant 4(|u| + |v|).$$

*Proof.* Let $\rho = ((u, w), (w, v))$ be of the desired form. Then there exist words $u_1$ and $u_2$, not necessarily distinct, such that $u_1u_1'$ and $u_2u_2'$ are instances of $\alpha\alpha$ in $w$ such that

$$w \vdash_{\alpha\alpha} u \quad \text{and} \quad w \vdash_{\alpha\alpha} v,$$

respectively. Although $u_1 = u_1'$ (and $u_2 = u_2'$), we use different notations for the two words to distinguish the two appearances of $\alpha$ in $w$. Writing $w = w_1u_2w_2u_2'w_3$ and $v = w_1w_2w_3$, there are 10 cases up to symmetry:

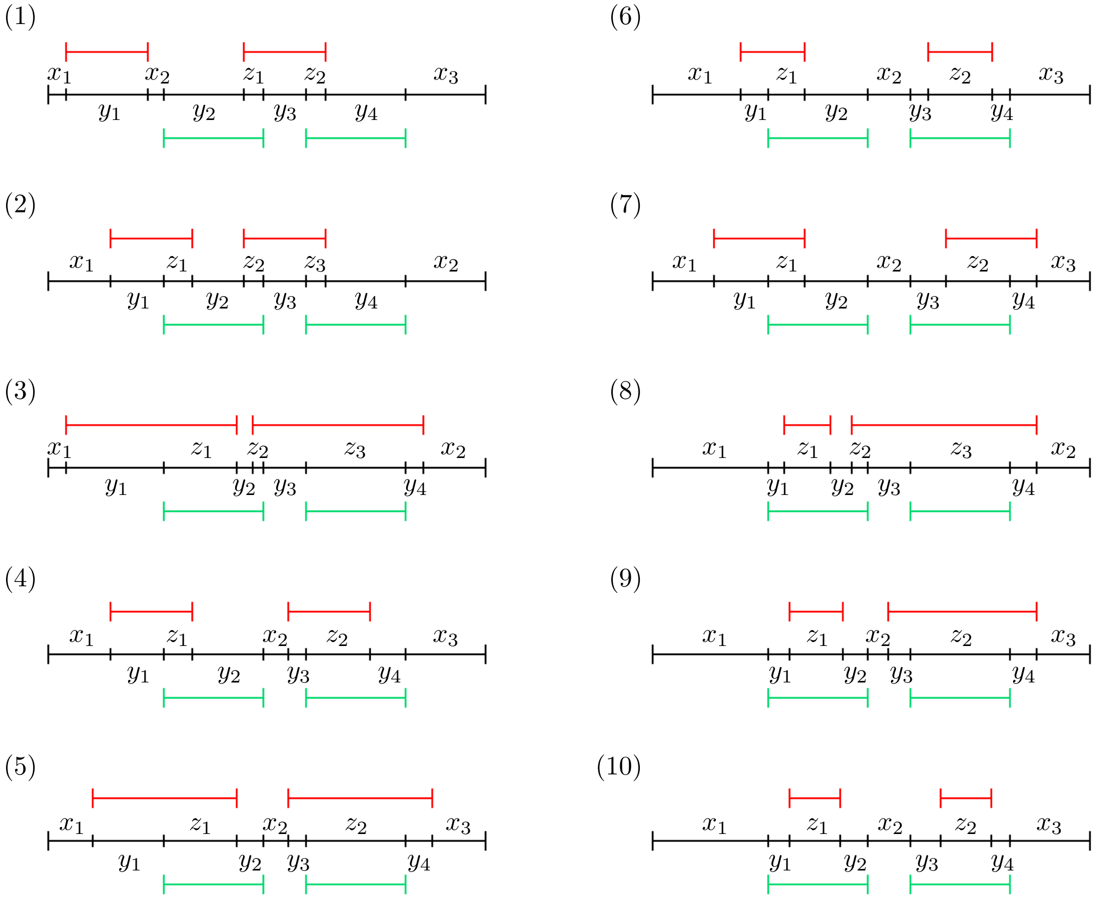
(1)
$x_1$ $x_2$ $z_1$ $z_2$ $x_3$
$y_1$ $y_2$ $y_3$ $y_4$
(2)
$x_1$ $z_1$ $z_2$ $z_3$ $x_2$
$y_1$ $y_2$ $y_3$ $y_4$
(3)
$x_1$ $z_1$ $z_2$ $z_3$ $x_2$
$y_1$ $y_2$ $y_3$ $y_4$
(4)
$x_1$ $z_1$ $x_2$ $z_2$ $x_3$
$y_1$ $y_2$ $y_3$ $y_4$
(5)
$x_1$ $z_1$ $x_2$ $z_2$ $x_3$
$y_1$ $y_2$ $y_3$ $y_4$
(6)
$x_1$ $z_1$ $x_2$ $z_2$ $x_3$
$y_1$ $y_2$ $y_3$ $y_4$
(7)
$x_1$ $z_1$ $x_2$ $z_2$ $x_3$
$y_1$ $y_2$ $y_3$ $y_4$
(8)
$x_1$ $z_1$ $z_2$ $z_3$ $x_2$
$y_1$ $y_2$ $y_3$ $y_4$
(9)
$x_1$ $z_1$ $x_2$ $z_2$ $x_3$
$y_1$ $y_2$ $y_3$ $y_4$
(10)
$x_1$ $z_1$ $x_2$ $z_2$ $x_3$
$y_1$ $y_2$ $y_3$ $y_4$

Figure 2.1: Visual representations of the ten cases. Red intervals represent $u_1$ and $u_1'$, while green intervals represent $u_2$ and $u_2'$.

(1) $u_1 \sqsubseteq w_1w_2w_3$;

(2) $u_1 \sqsubseteq w_1u_2$ and $u_1' \sqsubseteq u_2w_2u_2'$;

(3) $u_1 \sqsubseteq w_1u_2$ and $u_1' \sqsubseteq u_2w_2u_2'w_3$;

(4) $u_1 \sqsubseteq w_1u_2$ and $u_1' \sqsubseteq w_2u_2'$;

(5) $u_1 \sqsubseteq w_1u_2$ and $u_1' \sqsubseteq w_2u_2'w_3$;

(6) $u_1 \sqsubseteq w_1u_2$ and $u_1' \sqsubseteq u_2'$;

(7) $u_1 \sqsubseteq w_1u_2$ and $u_1' \sqsubseteq u_2'w_3$;

(8) $u_1 \sqsubseteq u_2$ and $u_1' \sqsubseteq u_2w_2u_2'w_3$;

(9) $u_1 \sqsubseteq u_2$ and $u_1' \sqsubseteq w_2u_2'w_3$;

(10) $u_1 \sqsubseteq u_2$ and $u_1' \sqsubseteq u_2'$.

We proceed to verify the bound for all 10 cases. To facilitate our analysis, we let $x_i$ denote factors of $w_j$'s, $y_i$'s denote factors of $u_k$'s or $u_k'$'s and $w_j$'s, and $z_i$ denote factors of $u_1$ or $u_1'$ and $u_2$ or $u_2'$ (see Figure 2.1). The idea is to show that the common factors, the $z_i$'s, that are inserted in $u$ to produce $w$ and removed from $w$ to produce $v$ can be chosen sufficiently small such that $|w|$ is bounded by the length of $u$ and $v$.

(1) Since $u_1 \sqsubseteq v$, $|u_1| \leqslant |v|$, implying that $|u_1 u_1'| \leqslant 2|v|$. Hence $w \leqslant 2|v| + |u|$, as desired.

Henceforth we may assume that $u_1 \cap u_2 u_2' \neq \epsilon$ and $u_1' \cap u_2 u_2' \neq \epsilon$, as otherwise the bound follows by (1).

(2) Write

$$w = x_1 y_1 z_1 y_2 z_2 y_3 z_3 y_4 x_2,$$

where $x_1 = w_1 - u_1$, $y_1 = u_1 - u_2$, $z_1 = u_1 \cap u_2$, $y_2 = (u_2 - u_1) - u_1'$, $z_2 = u_1' \cap u_2$, $y_3 = w_2$, $z_3 = u_1' \cap u_2'$, $y_4 = u_2' - u_1'$, and $x_2 = w_3$. Then either $|z_3| \leqslant |z_1|$ or $|z_1| \leqslant |z_3|$. Suppose the former holds. Then there exists factors $y_3' \sqsubseteq y_3$ and $y_4' \sqsubseteq y_4$ such that

$$z_1 = y_3' z_3 = z_3 y_4'.$$

By Lemma 2.2.7, we may assume that $|z_3| \leqslant |y_3'|, |y_4'|$. Then since $|y_3'| \leqslant |y_3|$, we infer that $|z_1| = |z_3| + |y_3'| \leqslant 2|y_3|$. Since

$$|z_2| \leqslant |u_2| - |z_1| \leqslant |u_2| - |z_3| = |y_4|,$$

we conclude that

$$\begin{aligned} |w| &= |x_1| + |x_2| + |y_1| + |z_1| + |y_2| + |z_2| + |y_3| + |z_3| + |y_4| \\ &\leqslant |u| + |y_1| + 2|y_3| + |y_2| + |y_4| + |y_3| + |y_3| + |y_4| \\ &\leqslant |u| + |v| + |v| + |v| + |v| + |u| + |u| \\ &\leqslant 3|u| + 4|v|. \end{aligned}$$

(3) Write

$$w = x_1 y_1 z_1 y_2 z_2 y_3 z_3 y_4 x_2,$$

where $x_1 = w_1 - u_1$, $y_1 = u_1 - u_2$, $z_1 = u_1 \cap u_2$, $y_2 = (u_2 - u_1) - u_1'$, $z_2 = u_1' \cap u_2$, $y_3 = w_2$, $z_3 = u_2'$, $y_4 = w_3 - x_2$, and $x_2 = w_3 - u_1'$. Then we have

$$y_1 z_1 = z_2 y_3 z_3 y_4 \quad \text{and} \quad z_1 y_2 z_2 = z_3, \tag{2.2.2}$$

implying that

$$y_1z_1 = z_2y_3z_1y_2z_2y_4. \tag{2.2.3}$$

Then $z_2y_3$ is a prefix of $y_1$, so we can write $y_1 = z_2y_3y_3'$ for some word $y_3'$. Applying (2.2.3), we infer that

$$z_2y_3y_3'z_1 = z_2y_3z_1y_2z_2y_4 \implies y_3'z_1 = z_1y_2z_2y_4.$$

Thus by Lemma 2.2.7, there exists $z_1'$ such that $|z_1'| \leqslant |y_3'|, |y_2z_2y_4|$ and $y_3'z_1' = z_1'y_2z_2y_4$. Writing $z_3' = z_1'y_2z_2$, we see that left multiplying both sides of the equation by $z_2y_3$ and reapplying the expression for $y_1$ yields (2.2.2) with $z_1$ and $z_3$ replaced by $z_1'$ and $z_3'$, respectively. Hence we may assume that $|z_1| \leqslant |y_2z_2y_4|$ and $|z_3| = |z_1y_2z_2| \leqslant |y_3'y_2z_2|$, where (2.2.3) gives the equality $2|z_2| = |y_1| - |y_3| - |y_2| - |y_4|$. Therefore

$$\begin{aligned}
|w| &= |u| + |y_1| + |y_3| + |y_4| + |z_1| + |z_2| + |z_3| \\
&\leqslant |u| + |y_1| + |y_3| + |y_4| + |y_2z_2y_4| + (|y_1| - |y_3| - |y_2| - |y_4|)/2 + |y_3'y_2z_2| \\
&\leqslant |u| + |y_1| + |y_3| + |y_4| + |y_2| + 3(|y_1| - |y_3| - |y_2| - |y_4|)/2 + |y_4| + |y_1| + |y_2| \\
&\leqslant |u| + 7|y_1|/2 - |y_3|/2 + |y_4|/2 + |y_2|/2 \\
&\leqslant 3|u|/2 + 7|v|/2.
\end{aligned}$$

(4) Write

$$w = x_1y_1z_1y_2x_2y_3z_2y_4x_3,$$

where $x_1 = w_1 - u_1$, $y_1 = u_1 - u_2$, $z_1 = u_1 \cap u_2$, $y_2 = u_2 - u_1$, $x_2 = w_2 - u_1'$, $y_3 = u_1' - u_2'$, $z_2 = u_1' \cap u_2'$, $y_4 = u_2' - u_1'$, and $x_3 = w_3$. Then we have

$$y_1z_1 = y_3z_2 \quad \text{and} \quad z_1y_2 = z_2y_4, \tag{2.2.4}$$

implying that either $y_3$ is a prefix of $y_1$ or $y_1$ is a prefix of $y_3$. Suppose first that the former holds. Then $y_4$ is a suffix of $y_2$, implying that we may write $y_1 = y_3y_3'$ and

$y_2 = y_4' y_4$ for some words $y_3'$ and $y_4'$. Then substitution into (2.2.4) gives

$$y_3 y_3' z_1 = y_3 z_2 \implies y_3' z_1 = z_2 \tag{2.2.5}$$

and

$$z_1 y_4' y_4 = z_2 y_4 \implies z_1 y_4' = z_2. \tag{2.2.6}$$

Combining these equalities, we infer that

$$y_3' z_1 = z_1 y_4'.$$

By Lemma 2.2.7, there exists a prefix $z_1'$ of $z_1$ such that $|z_1'| \leqslant |y_3'|, |y_4'|$ and $y_3' z_1' = z_1' y_4'$. Writing $z_1 = z_1' z_1''$, by (2.2.5), $z_2 = y_3' z_1' z_1''$. Then setting $z_2' = y_3' z_1' = z_1' y_4'$, we have

$$y_3 y_3' z_1' = y_3 z_2' \implies y_1 z_1' = y_3 z_2'$$

and

$$z_1' y_4' y_4 = z_2' y_4 \implies z_1' y_2 = z_2' y_4,$$

which are simply (2.2.4) with $z_1'$ replacing $z_1$ and $z_2'$ replacing $z_2$. Thus we may assume that

$$|z_1| \leqslant |y_3'| \leqslant |y_1| \quad \text{and} \quad |z_1| \leqslant |y_4'| \leqslant |y_2|,$$

and similarly $|z_2| \leqslant 2|y_1|, 2|y_2|$. Hence

$$\begin{aligned} |w| &= |u| + |y_1| + |y_3| + |z_1| + |z_2| \\ &\leqslant |u| + 4|y_1| + |y_3| \\ &\leqslant |u| + 4|v|, \end{aligned}$$

and

$$\begin{aligned} |w| &= |v| + |y_2| + |y_4| + |z_1| + |z_2| \\ &\leqslant |v| + 4|y_2| + |y_4| \end{aligned}$$

$$\leqslant |v| + 4|u|,$$

implying that

$$|w| \leqslant |u| + |v| + 3\min\{|u|, |v|\},$$

as desired. The analysis for the second case, where $y_1$ is a prefix of $y_3$ and thus $z_1$ and $z_2$ swap positions in (2.2.5) and (2.2.6), proceeds similarly, also giving the same bound.

(5) Write

$$w = x_1y_1z_1y_2x_2y_3z_2y_4x_3,$$

where $x_1 = w_1 - u_1$, $y_1 = u_1 - u_2$, $z_1 = u_1 \cap u_2$, $y_2 = u_2 - u_1$, $x_2 = w_2 - u_1'$, $y_3 = u_1' - u_2'$, $z_2 = u_2'$, $y_4 = w_3 - x_3$, and $x_3 = w_3 - u_1'$. Then we have

$$y_1z_1 = y_3z_2y_4 \quad \text{and} \quad z_1y_2 = z_2, \tag{2.2.7}$$

implying that

$$y_1z_1 = y_3z_1y_2y_4. \tag{2.2.8}$$

Thus $y_3$ is a prefix of $y_1$, so we can write $y_1 = y_3y_3'$ for some word $y_3'$. Then applying (2.2.8), we infer that

$$y_3y_3'z_1 = y_3z_1y_2y_4 \implies y_3'z_1 = z_1y_2y_4.$$

Then by Lemma 2.2.7 there exists a prefix $z_1'$ of $z_1$ such that $y_3'z_1' = z_1'y_2y_4$ and $|z_1'| \leqslant |y_3'|, |y_2y_4|$. Writing $z_2' = z_1'y_2$, we have $y_3'z_1' = z_2'y_4$. Then concatenating $y_3$ onto both sides, we infer that

$$y_1z_1' = y_3z_2'y_4,$$

yielding (2.2.7) with $z_1$ and $z_2$ replaced by $z_1'$ and $z_2'$. Hence we may assume that $|z_1| \leqslant |y_1|, |y_2y_4|$ and $|z_2| = |z_1y_2| \leqslant |y_1y_2|, |y_2y_4y_2|$, implying that

$$\begin{aligned}|w| &= |v| + |y_2| + |z_1| + |z_2|\\ &\leqslant |v| + |y_2| + |y_1| + |y_1y_2|\end{aligned}$$

$$\leqslant 3|v| + 2|u|.$$

(6) Write

$$w = x_1 y_1 z_1 y_2 x_2 y_3 z_2 y_4 x_3,$$

where $x_1 = w_1 - u_1$, $y_1 = u_1 - u_2$, $z_1 = u_1 \cap u_2$, $y_2 = u_2 - u_1$, $x_2 = w_2$, $y_3 = s_1$, $z_2 = u_1' \cap u_2'$, $y_4 = s_2$, $x_3 = w_3$, and $u_2' = s_1 u_1' s_2$. Then we have

$$y_1 z_1 = z_2 \quad \text{and} \quad z_1 y_2 = y_3 z_2 y_4, \tag{2.2.9}$$

implying that

$$z_1 y_2 = y_3 y_1 z_1 y_4. \tag{2.2.10}$$

Hence $y_4$ is a suffix of $y_2$, so we can write $y_2 = y_4' y_4$ for some word $y_4'$. Substituting this back into (2.2.10), we infer that

$$z_1 y_4' y_4 = y_3 y_1 z_1 y_4 \implies z_1 y_4' = y_3 y_1 z_1.$$

Then applying Lemma 2.2.7 yields a prefix $z_1'$ of $z_1$ such that $z_1' y_4' = y_3 y_1 z_1'$ and $|z_1'| \leqslant |y_4'|, |y_3 y_1|$. Writing $z_2' = y_1 z_1'$, we have $z_1' y_4' = y_3 z_2'$. Then concatenating $y_4$ to both sides of the equation gives

$$z_1' y_2 = y_3 z_2' y_4,$$

yielding (2.2.9) with $z_1'$ and $z_2'$ replacing $z_1$ and $z_2$, respectively. Thus we may assume that

$$|z_1| \leqslant |y_4'| \leqslant |y_2| \quad \text{and} \quad |z_2| \leqslant |y_1 y_2|,$$

implying that

$$\begin{aligned} |w| &= |u| + |y_1| + |z_1| + |z_2| \\ &\leqslant |u| + 2|y_2| + 2|y_1| \\ &\leqslant 3|u| + 2|v|. \end{aligned}$$

(7) Write

$$w = x_1 y_1 z_1 y_2 x_2 y_3 z_2 y_4 x_3,$$

where $x_1 = w_1 - u_1$, $y_1 = u_1 - u_2$, $z_1 = u_1 \cap u_2$, $y_2 = u_2 - u_1$, $x_2 = w_2$, $y_3 = u_2' - u_1'$, $z_2 = u_1' \cap u_2'$, $y_4 = w_3 - x_3$, and $x_3 = w_3 - u_1'$. Then we have

$$y_1 z_1 = z_2 y_4 \quad \text{and} \quad z_1 y_2 = y_3 z_2, \tag{2.2.11}$$

implying that either $y_1$ is a prefix of $z_2$ or $z_2$ is a prefix of $y_1$; suppose the former holds. Then $y_4$ is a suffix of $z_1$, implying that we can write $z_2 = y_1 y_1'$ and $z_1 = y_4' y_4$ for some words $y_1'$ and $y_4'$, giving

$$y_1 z_1 = y_1 y_1' y_4 \implies z_1 = y_1' y_4.$$

Hence $y_1' = y_4'$. Substituting the resulting expressions for $z_1$ and $z_2$ into the second equation of (2.2.11), we see that

$$y_1' y_4 y_2 = y_3 y_1 y_1'.$$

Thus by Lemma 2.2.7, there exists a prefix $y_1''$ of $y_1'$ such that $|y_1''| \leqslant |y_4 y_2|, |y_3 y_1|$ and $y_1'' y_4 y_2 = y_3 y_1 y_1''$. Setting $z_1' = y_1'' y_4$ and $z_2' = y_1 y_1''$, we recover (2.2.11) with $z_1$ and $z_2$ replaced by $z_1'$ and $z_2'$, respectively. Hence we may assume that $|z_1| = |y_1'' y_4| \leqslant |y_4 y_2 y_4|, |y_3 y_1 y_4|$ and $|z_2| = |y_1 y_1''| \leqslant |y_1 y_4 y_2|, |y_1 y_3 y_1|$. It follows then that

$$\begin{aligned} |w| &= |v| + |y_2| + |z_1| + |z_2| + |y_3| \\ &\leqslant |v| + |y_2| + |y_3 y_1 y_4| + |y_1 y_4 y_2| + |y_3| \\ &\leqslant 3|v| + 2|u|, \end{aligned}$$

as desired. Suppose then that the second case holds, that is, $z_2$ is a prefix of $y_1$. Then $|z_2| \leqslant |y_1|$ and thus $|z_1| \leqslant |z_2 y_4| \leqslant |y_1| + |y_4|$. Hence

$$\begin{aligned} |w| &= |v| + |y_2| + |z_1| + |z_2| + |y_3| \\ &\leqslant |v| + |u| + |y_1| + |y_1| + |y_4| \end{aligned}$$

$$\leqslant 3|v| + |u|.$$

(8,9) Neither of these cases can occur since $|u_1| \leqslant |u_2|$, yet

$$|u_1| = |u_1'| > |u_2'| = |u_2|,$$

a contradiction.

(10) Write

$$w = x_1 y_1 z_1 y_2 x_2 y_3 z_2 y_4 x_3,$$

where $x_1 = w_1$, $y_1 = s_1$, $z_1 = u_1 \cap u_2$, $y_2 = s_2$, $u_2 = s_1 z_1 s_2$, $x_2 = w_2$, $y_3 = s_1'$, $z_2 = u_1' \cap u_2'$, $y_4 = s_2'$, $u_2' = s_1' z_2 s_2'$, and $x_3 = w_3$. Then we have

$$z_1 = z_2 \quad \text{and} \quad y_1 z_1 y_2 = y_3 z_2 y_4, \tag{2.2.12}$$

implying that

$$y_1 z_1 y_2 = y_3 z_1 y_4$$

and either $y_1$ is a prefix of $y_3$ or $y_3$ is a prefix of $y_1$. Suppose the former holds. Then $y_4$ is a suffix of $y_2$ and thus we can write $y_3 = y_1 y_1'$ and $y_2 = y_4' y_4$ for words $y_1'$ and $y_4'$. Then substituting these back into the above equation implies that

$$y_1 z_1 y_4' y_4 = y_1 y_1' z_1 y_4 \implies z_1 y_4' = y_1' z_1.$$

By Lemma 2.2.7, there exists a prefix $z_1'$ of $z_1$ such that $|z_1'| \leqslant |y_1'|, |y_4'|$ and $z_1' y_4' = y_1' z_1'$. Then by reversing the above implications, we recover (2.2.12) with $z_1$ and $z_2$ replaced by $z_1'$ and $z_2'$, where $z_1' = z_2'$. Hence we may assume that $|z_1| = |z_2| \leqslant |y_1'|, |y_4'|$, implying that

$$\begin{aligned} |w| &= |u| + |z_1| + |z_2| \\ &\leqslant |u| + |y_1'| + |y_1'| \\ &\leqslant |u| + 2|y_3| \end{aligned}$$

$$\leqslant 3|u|,$$

as desired. The second case, where $y_3$ is a prefix of $y_1$, yields the same bound via a similar argument.

■

For $P = \{\alpha\alpha\}$ and words $u$ and $v$ with arbitrary minimal paths between them, we hypothesize that this result can be generalized to yield a bound on the maximum word found in some minimal path as a function of $|u|$, $|v|$, and $d_P(u, v)$. This leads to the following conjecture.

**Conjecture 2.2.9.** *Let $P = \{\alpha\alpha\}$. For all words $u$ and $v$, $d_P(u, v)$ is computable.*

We sketch an outline of a possible proof of the conjecture. Suppose there exists a minimal path $\rho$ between words $u$ and $v$ of the form $(r_1^R, r_2)$ for reductions $r_1$ and $r_2$. Then a given repeat word $u_iu_i$ removed (inserted) at the $i$th step in $\rho$ appears in $w$, the word at the end of $r_1^R$ and start of $r_2$, partitioned by some repeat words removed (inserted) before (after) the $i$th step. It is not difficult to see then that we may further partition $u_iu_i$ into a number of smaller repeat words $b_{i_j}b_{i_j}$, each of which appears in $w$, and that each partitioning repeat word forces the addition of at most 2 repeat words to this partition of $u_iu_i$ in $w$. Letting $b_1b_1, \ldots, b_nb_n$ and $c_1c_1, \ldots, c_mc_m$ be the resulting inserted and removed repeat words which appear in $w$, it follows that $n, m = O(d_P(u, v)^2)$. At this point we can apply Theorem 2.2.8 (or, more specifically, a variant of this lemma based on its proof) to bound the size of the intersection $z_{ij}$ of each inserted repeat word $b_ib_i$ and each removed repeat word $c_jc_j$ if $b_ib_i$ intersects at most one $c_jc_j$, and vice versa. But this can likely be achieved by further partitioning the $b_ib_i$'s and $c_jc_j$'s into smaller repeat words.[3] Assuming as much, combining the bound on each intersection $z_{ij}$ results in a bound on $w$ of the form $O(d_P(u, v)^N(|u|+|v|))$ for some $N > 1$.

To extend this argument to arbitrary minimal paths $\rho$ between words $u$ and $v$, we convert $\rho$ into a minimal path of the form $(r_1^2, r_2)$ and thereby conclude the bound for

[3] Note that this is the difficult part in converting this outline into a rigorous proof.

arbitrary $u$ and $v$ in the same connected component. We prove that such a conversion is possible by induction; most of the work is done in the base case where $\rho$ is of the form $(r_1, r_2^R)$. Yet it is not difficult to see that we can reverse the order of $\rho$, first inserting the repeat words to $u$ that were added in $r_2^R$ and then removing the repeat words that were removed in $r_1$—this yields the base case. For the inductive step, suppose $\rho = (r_1, r_2, \ldots, r_k)$. Then the induction hypothesis says that we can convert the minimal subpath $(r_1, r_2, \ldots, r_{k-1})$ into a minimal path $(r, r')$ such that $r^R$ and $r'$ are reductions. If $r_{k-1}$ is a reduction, then $(r, r'r_{k-1})$ is the desired path. Otherwise, we can also "flip" the minimal path $(r', r_{k-1})$ into a minimal path $(r'', r''')$ such that $(r'')^R$ and $r'''$ are reductions. In this case, $(r'r'', r''')$ is the desired path. Note that in Chapter 3, we use a similar path flipping argument to prove a stronger form of Conjecture 2.2.9 in the case where $u$ and $v$ are so-called double occurrence words.

With a general bound in terms of $d_p(u, v)$, $|u|$, and $|v|$ on the size of words appearing in some minimal path between $u$ and $v$ the conjecture follows by applying Lemma 2.2.6 and subsequently observing that there are only a finite computable number of candidate minimal paths. Consequently, a brute force search yields a minimal path, giving $d_P(u, v)$.

## 2.3 Indices

We are interested in investigating whether a word is "generated" by a given set of patterns. In this case, pattern indices define a measure of the complexity of that generation:

**Definition 2.3.1.** *Given a set of patterns $P$ that reduces $w$, define the* pattern index *of a word $w$ by*

$$I_P(w) := \min\{n \mid (w_0, w_1, \ldots, w_n = \epsilon) \text{ is a reduction of } w\}.$$

*Where $P$ is clearly understood, we simply use $I$ to denote the pattern index.*

In the sequel, whenever we write $I_P(w)$ for a word $w$ we assume that $P$ reduces $w$. If $P = \{p\}$, then we may write $I_p$ instead of $I_P$. A pattern $p$ is *trivial* if for all $w \in \Sigma^*$ with $|w| \geqslant |p|$, $I_p(w) = 1$. Note that if $|p|_\alpha \leqslant 1$ for all $\alpha \in X$, then $p$ is trivial. Hence a nontrivial pattern must contain at least one symbol appearing twice.

**Example 2.3.2.** *It is not difficult to check that $(abacddabca, acddca, \epsilon)$ is a minimal reduction of the word $w = abacddabca$ to the empty word, implying that $I_P(w) = 2$.*

**Definition 2.3.3.** *For a set of patterns $P$, the $P$-*language *$L_P$ is the set of all words $w$ such that $I_p(w) = 1$.*

Note that words in the $P$-language are at distance at most two from each other and are at distance 1 from $\epsilon$. In the following lemma, we record some basic properties of pattern indices.

**Lemma 2.3.4.** *Let $w_1$ and $w_2$ be words on alphabets $\Sigma_1$ and $\Sigma_2$, respectively, and $P_1$ and $P_2$ be sets of patterns that reduce $w_1$ and $w_2$, respectively. Then the following hold:*

*(1) $I_{P_1}(w_1) \geqslant 0$, and $I_{P_1}(w_1) = 0$ if and only if $w_1 = \epsilon$,*

*(2) $I_{P_1}(w_1) \leqslant |w_1|$,*

*(3) if $w_1 = w_2$, then $I_{P_1}(w_1) = I_{P_1}(w_2)$,*

*(4) if $P_1 \subseteq P_2$, then $P_2$ reduces $w_1$ and $I_{P_1}(w_1) \geqslant I_{P_2}(w_1)$,*

*(5) if $w_2 \vdash_p w_1$ for some $p \in P_1 \cup P_2$, then $I_{P_1 \cup P_2}(w_2) \leqslant I_{P_1 \cup P_2}(w_1) + 1$,*

*(6) $I_{P_1 \cup P_2}(w_1 w_2) \leqslant I_{P_1}(w_1) + I_{P_2}(w_2)$.*

*Proof.* Set $P := P_1 \cup P_2$. Note that (1), (2), and (3) follow immediately by definition. Result (5) follows by noting that combining any minimal reduction of $w_1$ with $P$ with an initial application of reduction operation $\vdash_p$ on $w_2$ yields a reduction of $w_2$ with $P$ of size $I_P(w_1) + 1$, implying that

$$I_P(w_2) \leqslant I_P(w_1) + 1,$$

as desired. Result (4) follows by observing that with the given conditions, any reduction of $w_1$ with $P_1$ is also a reduction of $w_1$ with $P_2$. For (6), note that combining any two minimal reductions of $w_1$ and $w_2$ with $P_1$ and $P_2$, respectively, generates a reduction of $w_1 w_2$ of size $I_{P_1}(w_1) + I_{P_2}(w_2)$ with $P$. Hence the inequality follows.

■

We record the following straightforward results.

**Lemma 2.3.5.** *Given a set of patterns $P$, and words $u$ and $v$ such that $u$ can be reduced to $v$, for all reductions $r$ from $u$ to $v$,*

$$|r| \geqslant I_P(u) - I_P(v).$$

*Proof.* If there exists a reduction $r$ from $u$ to $v$ such that $|r| < I_P(u) - I_P(v)$, then there exists a reduction $r'$ from $v$ to $\epsilon$ such that $|rr'| \leqslant I_P(v) + |r| < I_P(u)$. But $rr'$ is a reduction from $u$ to $\epsilon$, so this contradicts the definition of $I_P(u)$. Hence the inequality holds. ■

**Proposition 2.3.6.** *For all words $u$ and $v$ and sets of patterns $P$ that reduce $u$ and $v$,*

$$d_P(u, v) \leqslant I_P(u) + I_P(v).$$

*Proof.* The inequality follows immediately by observing that there exists a path of the form

$$((w_0 = u, w_1, \ldots, w_{I_P(u)} = \epsilon), (w_{I_P(u)} = \epsilon, \ldots, w_{I_P(u)+I_P(v)} = v))$$

for all $u$ and $v$. ■

# 3 Patterns in Double Occurrence Words

## 3.1 Definitions

For brevity, in the sequel we will often assume that $\Sigma = \mathbb{N}$ and, if $u \equiv v$, abuse notation by regarding the two words as identical, writing $u = v$. As such, many of the following definitions and theorems assume a *labeling* of words, that is, a choice of alphabet, but generalize to words over any alphabet.

**Definition 3.1.1.** *A word $w \in \Sigma^*$ is a* double occurrence word *if for all $a \in \Sigma$, $|w|_a = 2$. We call $|w|/2$ the* size *of $w$.*

**Example 3.1.2.** *The word* 1213424355 *is a double occurrence word, while the word $w =$* 113234324 *is not because there are three occurrences of the letter* 3 *in $w$.*

Unless otherwise stated, we now assume all words are double occurrence words.

**Definition 3.1.3.** *A double occurrence word $w$ is* irreducible *if $w$ cannot be written as a product $uv$ of two non-empty double occurrence words $u$ and $v$. If $w$ has no double occurrence factors,*[1] *then $w$ is* strongly irreducible*.*

**Remark 3.1.4.** *Patterns may or may not appear in double occurrence words as double occurrence subwords. When working with double occurrence words, we say that a pattern $p$ is a double occurrence pattern if all instances of $p$ in any given word are themselves double occurrence words or, equivalently, if $|p|_\alpha + |p|_{\alpha^R} = 2$ for all $\alpha \in X[p]$. Similarly, we extend the notions of irreducible and strongly irreducible to patterns.*

As the terms suggest, it is clear that a strongly irreducible word is also irreducible. Pattern indices are particularly well-behaved on double occurrence words:

[1] that is, factors that are double occurrence words.

**Lemma 3.1.5.** *Let $w_1$ and $w_2$ be double occurrence words on alphabets $\Sigma_1$ and $\Sigma_2$, respectively, and $P_1$ and $P_2$ be sets of double occurrence patterns that reduce $w_1$ and $w_2$, respectively. Then the following hold:*

*(a) If $\Sigma_1 \cap \Sigma_2 = \emptyset$ and all $p \in P_1 \cup P_2$ are irreducible, then $I_{P_1 \cup P_2}(w_1 w_2) = I_{P_1}(w_1) + I_{P_2}(w_2)$,*

*(b) if $w_1 \sqsubseteq w_2$ and all $p \in P_1 \cup P_2$ are strongly irreducible, then $I_{P_1 \cup P_2}(w_1) \leqslant I_{P_1 \cup P_2}(w_2)$.*

*Proof.* Set $P = P_1 \cup P_2$. Suppose there exist double occurrence words $w_1$ and $w_2$ satisfying the conditions for (b) for which the inequality does not hold. Then, writing $w_2 = uw_1v$, without loss of generality there exists $p \in P$ such that an instance $u'$ of $p$ intersects $w_1$ and either $u$ or $v$ (or both). Otherwise, we may iteratively remove pattern instances from $w_1$ until either (1) there exists a pattern with an instance of the desired properties (with $w_1$ replaced by the reduced word $w_1'$), or (2) we reach the empty word. In the former case, we set $w_1 = w_1'$ and $w_2 = uw_1'v$, while in the latter case all reductions of $w_2$ contain as a subsequence some reduction of $w_1$, so the inequality follows. Thus either the subword of $u'$ intersecting $w_1$ is a double occurrence word, in which case $u'$ is not strongly irreducible, or it is not, in which case $u'$ is not a double occurrence word. Since both cases yield contradictions, we conclude that

$$I_P(w_1) \leqslant I_P(w_2),$$

as desired. If $\Sigma_1 \cap \Sigma_2 = \emptyset$ and all $p \in P_1 \cup P_2$ are irreducible, then

$$I_P(w_1 w_2) = I_{P_1}(w_1) + I_{P_2}(w_2)$$

follows by observing, as in the case of (b), that if not then there must exist an irreducible double occurrence pattern instance $u$ which intersects both $w_1$ and $w_2$. Then neither $u \cap w_1$ and $u \cap w_2$ are double occurrence words, contradicting the assumption that $w_1$ and $w_2$ are double occurrence words with disjoint alphabets. Hence we conclude (a). ■

Recall that instances of the patterns $\alpha\alpha$ and $\alpha\alpha^R$ are called repeat words and return words, respectively. In the sequel, we also refer to the patterns themselves by these names. We now further generalize our notion of a pattern to include more interesting languages.

**Definition 3.1.6.** *Let $X$ be a set of variables. A* recursive pattern *$\pi = \{p_1, p_2, \ldots\}$ is a subset of $X^*$ such that there exists $f, g\colon \mathbb{N} \to \mathbb{N}$ and a symbol $\alpha_i \in X$, with $\alpha_i \notin \Sigma[p_i]$, satisfying*

$$p_i = s_i r_i t_i \in X^+ \quad \textit{and} \quad p_{i+1} = s_i \alpha_i r_i \alpha_i t_i$$

*for all $i \geqslant 1$, where $f(i) = |r_i|$ and $g(i) = |t_i|$. We say that a recursive pattern $\pi$* appears *in a word $w$ if there exists $p \in \pi$ appearing in $w$. If $p_1 = \alpha\alpha$, $f \equiv 1$, and $g \equiv 0$, we call $\pi$ the* tangled cord*, written $\pi_T$, while if $p_1 = \alpha\alpha$ and $f = g \equiv 0$, we call $\pi$ the* loop pattern*, written $\pi_L$.*

It is straightforward to show that neither the tangled cord nor the loop pattern can be defined as single patterns. In the sequel, we only consider the tangled cord and loop pattern as appearing strictly and literally. Note that, in double occurrence words, the repeat word and return word can be equivalently viewed as the patterns $\alpha\alpha$ and $\alpha\alpha^R$ or as literally-appearing recursive patterns $\pi_R$ and $\pi'_R$ with $p_1 = \alpha\alpha$, $f(i) = i$, and $g(i) = 0$ and $p_1 = \alpha\alpha$, $f(i) = 0$, and $g(i) = i$, respectively, which appear strictly save for a single gap allowed in the "center" of the pattern. When using the latter definition, we call each factor $\alpha_1\alpha_2\cdots\alpha_n$ (which is equivalent to $\alpha$ from the pattern definition) a *half* of the repeat or return word.

We now define reductions under this new notion of a pattern.

**Definition 3.1.7.** *Given a set of recursive patterns $\Pi$, define a* reduction *of a word $u$ to be a sequence $r = (w_0, w_1, \ldots, w_n)$ such that*

1. *$w_0 = u$,*

2. *and for all $1 \leqslant i \leqslant n$, there exists $\pi \in \Pi$ and $p \in \pi$ such that $w_{i-1} \vdash w_i$.*

With this notion of a reduction with a set of recursive patterns, the definitions of indices and distances with $\Pi$ follow straightforwardly; in particular, we now write $I_\Pi$ and $d_\Pi$ instead of $I_P$ and $d_P$, respectively. All other notation similarly extends to the new conception of a pattern.

## 3.2 Complete Patterns

We introduce several important properties of the repeat word and return word patterns which partially explain their "well-behaved" nature.

**Definition 3.2.1.** *Let $p = \alpha_1 \cdots \alpha_n$ be a pattern and let $w$ be a word containing an instance $f(\alpha_1) \cdots f(\alpha_n)$ of $p$, where $f \colon X_p \to \Sigma^+$. Suppose that for all maps $f' \colon X_p \to \Sigma^+$ with $\Sigma[f'(\alpha_i)] \subseteq \Sigma[f(\alpha_i)]$ for all $1 \leqslant i \leqslant n$, $f'(\alpha_1) \cdots f'(\alpha_n)$ is a sub-instance of $p$ in $w$ and removing $f'(\alpha_1) \cdots f'(\alpha_n)$ from $f(\alpha_1) \cdots f(\alpha_n)$ yields another instance of $p$. If this holds for all $w$, then $p$ is called* instance-closed.

**Definition 3.2.2.** *A recursive pattern $\pi = \{p_1, p_2, \ldots\}$ is called* instance-closed *if for all words $w$ and $1 \leqslant j \leqslant i$, $p_j \subseteq p_i$, that is, an instance $u_i$ of $p_i$ in $w$ contains a sub-instance $u_j$ of $p_j$, and $p_i - p_j \in \pi$, that is, removing the instance $u_j$ from $u_i$ yields another instance of some $p \in \pi$.*

**Definition 3.2.3.** *Two patterns $p$ and $p'$ are* compatible in a word $w$ *if for any two instances of $p$ and $p'$, respectively, such that*

$$w = z_0 f(\alpha_1) z_1 \cdots f(\alpha_n) z_n = z_0' g(\beta_1) z_1' \cdots g(\beta_k) z_k',$$

*either*

1. *$g(\beta_1) \cdots g(\beta_k)$ is a subword of $z_0 z_1 \cdots z_n$ and $f(\alpha_1) \cdots f(\alpha_n)$ is a subword of $z_0' z_1' \cdots z_k'$, or*

2. *$n = k$ and, for all $i$, $f(\alpha_i)$ and $g(\beta_i)$ have a common factor $x_i$ such that $x_1 \cdots x_n$ is an instance of both $p$ and $p'$ in $w$.*

*We say that $p$ and $p'$ are* compatible *if they are compatible for every word $w$. Recursive patterns $\pi$ and $\pi'$ are* compatible *if for all $p \in \pi$ and $p' \in \pi'$, $p$ and $p'$ are compatible.*

**Definition 3.2.4.** *We call a set of recursive patterns $\Pi$* complete *if*

1. *for all $\pi \in \Pi$, $\pi$ is instance-closed, and*

2. *for all $\pi, \pi' \in \Pi$, $\pi$ and $\pi'$ are compatible.*

*A recursive pattern $\pi$ is said to be* complete *if $\{\pi\}$ is complete.*

It is straightforward to show that the repeat word $\pi_R$, return word $\pi'_R$, and loop pattern $\pi_L$ are complete. For example, the repeat word $w = 1234512345$ contains the repeat word 3434 which, when removed from $w$, yields the repeat word 125125. In fact, any subset of $\{\pi_R, \pi'_R, \pi_L\}$ is complete. This list turns out to be exhaustive (in double occurrence words):

**Proposition 3.2.5.** *In double occurrence words, any literally-appearing complete recursive pattern $\pi$ with $p_1 = \alpha\alpha$ is either the repeat word, return word, or loop pattern.*

*Proof.* We proceed via induction on $n$ to show that either $p_n \in \pi_R$, $p_n \in \pi'_R$, or $p_n \in \pi_L$ for all $n \in \mathbb{N}$. By assumption, $p_1$ is either the repeat word, return word, or loop pattern of size 1, as desired. Now suppose $p_i$ is equivalent to the repeat word, return word, or loop pattern for all $1 \leqslant i \leqslant n$. Then either

$$p_i = \alpha_1\alpha_2 \cdots \alpha_i\alpha_1\alpha_2 \cdots \alpha_i, \text{ or} \tag{3.2.1}$$

$$p_i = \alpha_1\alpha_2 \cdots \alpha_i\alpha_i\alpha_{i-1} \cdots \alpha_1, \text{ or} \tag{3.2.2}$$

$$p_i = \alpha_1\alpha_1\alpha_2\alpha_2 \cdots \alpha_i\alpha_i \tag{3.2.3}$$

for all $1 \leqslant i \leqslant n$, where we note that in the third case $p_i$ appears strictly, while in the first two cases $p_i$ appears strictly except it may have a gap between each occurrence of $\alpha_i$. Suppose (3.2.1) holds for all $1 \leqslant i \leqslant n$. Then $p_{n+1}$ is constructed by inserting two occurrences of a new variable $\alpha_{n+1}$. These occurrences cannot "asymmetrically interrupt" both halves of $p_n$, in the sense that if $p_{n+1}$ is of the form

$$p_{n+1} = s_1\alpha_j\alpha_{n+1}\alpha_{j+1}s_2,$$

then

$$s_1 = s_3\alpha_j\alpha_{n+1}\alpha_{j+1}s_4 \quad \text{or} \quad s_2 = s_3\alpha_j\alpha_{n+1}\alpha_{j+1}s_4$$

for some subpatterns $s_3$ and $s_4$; otherwise, $p_n \nsubseteq p_{n+1}$. Then, after relabeling, there are 9 remaining cases:

$$p_{n+1} = \alpha_{n+1}\alpha_{n+1}\alpha_1\alpha_2 \cdots \alpha_n Z\alpha_1\alpha_2 \cdots \alpha_n, \tag{3.2.4}$$

$$p_{n+1} = \alpha_{n+1}\alpha_1\alpha_2\cdots\alpha_n\alpha_{n+1}Z\alpha_1\alpha_2\cdots\alpha_n, \tag{3.2.5}$$

$$p_{n+1} = \alpha_{n+1}\alpha_1\alpha_2\cdots\alpha_n Z\alpha_1\alpha_2\cdots\alpha_n\alpha_{n+1}, \tag{3.2.6}$$

$$p_{n+1} = \alpha_1\alpha_2\cdots\alpha_n\alpha_{n+1}\alpha_{n+1}Z\alpha_1\alpha_2\cdots\alpha_n, \tag{3.2.7}$$

$$p_{n+1} = \alpha_1\alpha_2\cdots\alpha_n\alpha_{n+1}Z\alpha_{n+1}\alpha_1\alpha_2\cdots\alpha_n, \tag{3.2.8}$$

$$p_{n+1} = \alpha_1\alpha_2\cdots\alpha_n\alpha_{n+1}Z\alpha_1\alpha_2\cdots\alpha_n\alpha_{n+1}, \tag{3.2.9}$$

$$p_{n+1} = \alpha_1\alpha_2\cdots\alpha_n Z\alpha_{n+1}\alpha_{n+1}\alpha_1\alpha_2\cdots\alpha_n, \tag{3.2.10}$$

$$p_{n+1} = \alpha_1\alpha_2\cdots\alpha_n Z\alpha_{n+1}\alpha_1\alpha_2\cdots\alpha_n\alpha_{n+1}, \tag{3.2.11}$$

$$p_{n+1} = \alpha_1\alpha_2\cdots\alpha_n Z\alpha_1\alpha_2\cdots\alpha_n\alpha_{n+1}\alpha_{n+1}, \tag{3.2.12}$$

where we use $Z$ to make it clear where the gap carried over from $p_n$ is located. Removing the repeat word subpattern $\alpha_1\cdots\alpha_n\alpha_1\cdots\alpha_n$ from (3.2.4), (3.2.5), (3.2.7), (3.2.10), (3.2.11), and (3.2.12) yields a strictly-appearing subpattern of the form $\alpha\alpha$, which is not equivalent to the repeat word of size 1. Hence, by completeness, we may exclude these cases. Suppose (3.2.6) holds. Then, by applying a similar argument, we infer that

$$\begin{aligned}
p_{n+2} &= \alpha_{n+2}\alpha_{n+1}\alpha_1\alpha_2\cdots\alpha_n Z\alpha_1\alpha_2\cdots\alpha_n\alpha_{n+1}\alpha_{n+2},\\
p_{n+2} &= \alpha_{n+1}\alpha_1\alpha_2\cdots\alpha_n\alpha_{n+2}Z\alpha_{n+2}\alpha_1\alpha_2\cdots\alpha_n\alpha_{n+1},\\
p_{n+2} &= \alpha_{n+1}\alpha_1\alpha_2\cdots\alpha_n\alpha_{n+2}Z\alpha_1\alpha_2\cdots\alpha_n\alpha_{n+1}\alpha_{n+2}, \text{ or}\\
p_{n+2} &= \alpha_{n+2}\alpha_{n+1}\alpha_1\alpha_2\cdots\alpha_n Z\alpha_{n+2}\alpha_1\alpha_2\cdots\alpha_n\alpha_{n+1},
\end{aligned}$$

as otherwise $p_{n+1} \nsubseteq p_{n+2}$. Yet in first two cases, removing $\alpha_1\cdots\alpha_n\alpha_1\cdots\alpha_n$ results in a word of the form $\alpha\beta Z\beta\alpha$, which is not equivalent to $p_2$, and in the latter two cases, removing both occurrences of $\alpha_{n+1}$ (an instance of $p_1$) gives the repeat word of size $n+1$, which is not equivalent to $p_{n+1}$. Thus, we exclude (3.2.6) and, similarly, (3.2.8). This leaves only (3.2.9), the repeat word of size $n+1$, as desired. Now suppose (3.2.2) holds for all $1 \leqslant i \leqslant n$. As before, completeness implies that $p_{n+1}$ can be constructed by inserting two occurrences of a variable into $p_n$ that do not asymmetrically interrupt both halves of $p_n$. This leaves 9 remaining cases:

$$p_{n+1} = a_{n+1}\alpha_{n+1}\alpha_1\alpha_2\cdots\alpha_n Z\alpha_n\alpha_{n-1}\cdots\alpha_1, \tag{3.2.13}$$

$$p_{n+1} = \alpha_{n+1}\alpha_1\alpha_2 \cdots \alpha_n\alpha_{n+1}Z\alpha_n\alpha_{n-1} \cdots \alpha_1, \tag{3.2.14}$$

$$p_{n+1} = \alpha_{n+1}\alpha_1\alpha_2 \cdots \alpha_n Z\alpha_{n+1}\alpha_n\alpha_{n-1} \cdots \alpha_1, \tag{3.2.15}$$

$$p_{n+1} = \alpha_1\alpha_2 \cdots \alpha_n\alpha_{n+1}\alpha_{n+1}Z\alpha_n\alpha_{n-1} \cdots \alpha_1, \tag{3.2.16}$$

$$p_{n+1} = \alpha_1\alpha_2 \cdots \alpha_n\alpha_{n+1}Z\alpha_{n+1}\alpha_n\alpha_{n-1} \cdots \alpha_1, \tag{3.2.17}$$

$$p_{n+1} = \alpha_1\alpha_2 \cdots \alpha_n\alpha_{n+1}Z\alpha_n\alpha_{n-1} \cdots \alpha_1\alpha_{n+1}, \tag{3.2.18}$$

$$p_{n+1} = \alpha_1\alpha_2 \cdots \alpha_n Z\alpha_{n+1}\alpha_{n+1}\alpha_n\alpha_{n-1} \cdots \alpha_1, \tag{3.2.19}$$

$$p_{n+1} = \alpha_1\alpha_2 \cdots \alpha_n Z\alpha_{n+1}\alpha_n\alpha_{n-1} \cdots \alpha_1\alpha_{n+1}, \tag{3.2.20}$$

$$p_{n+1} = \alpha_1\alpha_2 \cdots \alpha_n Z\alpha_n\alpha_{n-1} \cdots \alpha_1\alpha_{n+1}\alpha_{n+1}. \tag{3.2.21}$$

As before, removing the return word subpattern $\alpha_1 \cdots \alpha_n\alpha_n \cdots \alpha_1$ from (3.2.13), (3.2.14), (3.2.16), (3.2.19), (3.2.20), and (3.2.21) gives a strictly-appearing subpattern of the form $\alpha\alpha$, which is not equivalent to $p_1$. Therefore, by completeness, we exclude these cases. Suppose then that (3.2.15) holds. Applying a similar argument, we infer that

$$\begin{aligned}
p_{n+2} &= a_{n+1}\alpha_1\alpha_2 \cdots \alpha_n\alpha_{n+2}Z\alpha_{n+1}\alpha_n\alpha_{n-1} \cdots \alpha_1\alpha_{n+2},\\
p_{n+2} &= \alpha_{n+2}\alpha_{n+1}\alpha_1\alpha_2 \cdots \alpha_n Z\alpha_{n+2}\alpha_{n+1}\alpha_n\alpha_{n-1} \cdots \alpha_1,\\
p_{n+2} &= \alpha_{n+1}\alpha_1\alpha_2 \cdots \alpha_n\alpha_{n+2}Z\alpha_{n+2}\alpha_{n+1}\alpha_n\alpha_{n-1} \cdots \alpha_1, \text{ or}\\
p_{n+2} &= \alpha_{n+2}\alpha_{n+1}\alpha_1\alpha_2 \cdots \alpha_n Z\alpha_{n+1}\alpha_n\alpha_{n-1} \cdots \alpha_1\alpha_{n+2},
\end{aligned}$$

as otherwise $p_{n+1} \nleq p_{n+2}$. In the first two cases, removing $\alpha_1 \cdots \alpha_n\alpha_n \cdots \alpha_1$ results in a word of the form $\alpha\beta Z\alpha\beta$, which is not equivalent to $p_2$. On the other hand, in the latter two cases, removing both occurrences of $\alpha_{n+1}$ yields a word which is not equivalent to $p_{n+1}$. Thus, by completeness, we exclude (3.2.15) and, similarly, (3.2.18). This leaves only (3.2.17), the return word of size $n+1$, as desired. Finally, suppose (3.2.3) holds for all $1 \leqslant i \leqslant n$. Then clearly

$$p_{n+1} = \alpha_1\alpha_1\alpha_2\alpha_2 \cdots \alpha_n\alpha_n\alpha_{n+1}\alpha_{n+1},$$

as otherwise $p_n \nleq p_{n+1}$. Since this is the loop pattern of size $n+1$, we conclude that $p$ is the repeat word, return word, or loop pattern, as desired.

■

**Example 3.2.6.** *The tangled cord $\pi_T$ is not complete—no smaller tangled cord appears in*

12132434*, a tangled cord of size* 4*.*

Working with double occurrence words and complete sets of recursive patterns makes it feasible to obtain a wealth of results that remain out of reach for arbitrary words and arbitrary sets of recursive patterns. We proceed to prove that the distance is computable for this class of words with these types of recursive patterns. In the following, we assume that all reductions and paths are relative a complete set of recursive patterns $\Pi$.

**Lemma 3.2.7.** *Let $u$ and $v$ be words and $\rho$ be a path from $u$ to $v$ that consists of two steps, a single insertion and a single deletion. Then there exists a path $\rho'$ from $u$ to $v$ that consists of one deletion, or one deletion and one insertion.*

*Proof.* Without loss of generality, assume that $|u| \geqslant |v|$. Let $x$ and $y$ be the instances of recursive patterns in $\Pi$ inserted and removed in $\rho$, respectively; that is, $u = w - x$ and $v = w - y$ for some $w$. If $x$ and $y$ are disjoint, then they are contained in $v$ and $u$, respectively. Hence $w' = u - y = v - x$ defines a path $\rho'$ consisting of a single deletion followed by a single insertion. If $x$ and $y$ are not disjoint we set $z \in x \cap y$ to be of maximal length. Then by completeness, each of $x - z$ and $y - z$ is either $\epsilon$ or an instance of a recursive pattern in $\Pi$. If one of them is the empty word then, since $|u| \geqslant |v|$, we have $x - z = \epsilon$. Then $v = u - (y - z)$, implying that $\rho' = (u, v)$ is a path of a single deletion. In the other case, if neither $x - z$ nor $y - z$ is the empty word, then since $x - z$ and $y - z$ are disjoint by the choice of $z$, we conclude that $w' = u - (y - z) = v - (x - z)$ defines a path consisting of a single deletion and a single insertion.

■

**Lemma 3.2.8.** *Let $u$ and $v$ be words such that there exists a path between $u$ and $v$ of the form $(r_1^R, r_2)$, where $r_1$ and $r_2$ are reductions. Then there exists a path of the form $(r_1', {r_2'}^R)$ such that $|r_1| + |r_2| \leqslant |r_1'| + |r_2'|$.*

*Proof.* Let $\rho = (r_1^R, r_2)$ be a path from $u$ to $v$. Then we iteratively replace each consecutive insertion-deletion of instances of patterns in $P$ with a deletion-insertion (or just deletion) as described in Lemma 3.2.7. In this way, the path $\rho$ from $u$ to $v$ can be replaced with a sequence of deletions of instances of patterns followed with a sequence of insertions of instances of patterns in $P$. Such a path is of form $(r_1', {r_2'}^R)$ for some reductions $r_1'$ and $r_2'$ and its length is at least as long as $\rho$.

■

With this, we obtain our main general result.

**Theorem 3.2.9.** *For all words $u$ and $v$, there exists a minimal path $\rho$ between $u$ and $v$ of the form $(r_1)$ or $(r_1, r_2^R)$, where $r_1$ and $r_2$ are reductions.*

*Proof.* Let $\rho = (r_1, \ldots, r_k)$ be a path between $u$ and $v$. We may assume that for each $i$ in $\rho$ it is either $r_i r_{i+1}^R$ or $r_i^R r_{i+1}$. The theorem follows by induction on $k$. If $k \leqslant 2$, then the result follows by Lemma 3.2.8. Suppose the result holds for $1 \leqslant l \leqslant k-1$. By hypothesis there exists a minimal path $\rho' = (r', r^R)$ from $u$ to $w_{k-2}$, the last word in $r_{k-2}$, such that $r'$ and $r$ are reductions. Similarly, by Lemma 3.2.8 we may take that the last two reductions forming a path from $w_{k-2}$ to $v$ (note $w_{k-2}$ is also the first word in $r_{k-1}$) in $\rho$ are of form $(r_{k-1}, r_k^R)$ without increasing the length of $\rho$.

Then we have a new path $\rho' = (r', r^R, r_{k-1}, r_k^R)$. Another application of Lemma 3.2.8 flips the subpath $(r^R, r_{k-1})$ to $(r'', {r'''}^R)$ without increasing the length of the path and yields a path $\rho'' = (r'r'', {r'''}^R r_k^R)$ where the reduction $r'r''$ is a sequence of deletions and ${r'''}^R r_k^R$ is a sequence of insertions. (see Figure 3.1).

■

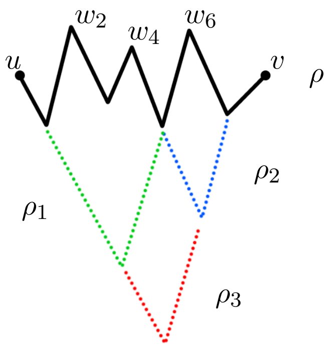

Figure 3.1: A schematic depicting the process of removing insertions that precede deletions in a path from $u$ to $v$.

In addition to giving a straightforward procedure for computing the distance between two double occurrence words, an upper and lower bound on the distance $d_P$ follows from this result.

**Corollary 3.2.10.** *For words $u, v \in L_P$,*

$$|I_P(u) - I_P(v)| \leqslant d_P(u,v) \leqslant I_P(u) + I_P(v).$$

*Proof.* It follows immediately from the triangle inequality of the distance $d_P$, that is, $I_P(u) = d_p(u,\epsilon) \leq d_P(u,v) + d_P(v,\epsilon) = d_P(u,v) + I_P(v)$. The other inequality follows similarly.

■

**Example 3.2.11.** *The distance between words* 121323 *and* 123321 *relative the set $P = \{\alpha\alpha, \alpha\alpha^R\}$ achieves the bound in Corollary 3.2.10 because there is no path with a single insertion and a single deletion that reduces* 121323 *to* 123321*. Therefore*

$$d_P(121323, 123321) = 3 = 2 + 1 = I_P(121323) + I_P(123321).$$

## 3.3 Pattern Recurrence Index

We now turn to the study of particular pattern indices. A natural and (as we will see in Chapter 4) highly applicable choice is the index generated by two of the simplest and most well-behaved patterns we have encountered thus far, the repeat word and return word.

**Definition 3.3.1.** *Define the* pattern recurrence index *to be the pattern index $PI := I_\Pi$, where $\Pi$ contains the repeat word and return word.*

**Example 3.3.2.** *A straightforward calculation shows that $PI(12134234) = 2$.*

For the rest of this section, we assume that all reductions and reduction operations are defined with $\Pi$, that is, with the repeat word and return word. Furthermore, unless otherwise specified, the last word in any reduction is the empty word $\epsilon$. Note that Lemma 3.1.5 implies that $PI(uv) = PI(u) + PI(v)$ for all (double occurrence) words $u$ and $v$ on disjoint alphabets since the repeat word and return word are irreducible. We are interested in considering the tangled cord as both a recursive pattern and as words of a given type:

**Definition 3.3.3.** *For all $n \in \mathbb{N}$, the* tangled cord $T_n$ *is defined by setting $T_n = f(t_n)$, where $\tau = \{t_1, t_2, \ldots\}$ is the tangled cord recursive pattern and $f\colon X_{t_n} \to \{1, 2, \ldots, n\}$. Equivalently, we set $T_0 = \epsilon$, $T_1 = 11$, and*

$$T_n = 12132 \cdots (n-1)(n-2)n(n-1)n$$

*for all $n \geqslant 2$.*

Unless otherwise specified, 'tangled cord' will now refer to words of the above type. It has been speculated that the tangled cord may maximize certain pattern indices defined by the removal of the repeat word and return word, including the pattern recurrence index and the nesting index (considered in the following section). The central focus of this section and the following section is exploring this claim. We begin our analysis by considering the pattern recurrence index of the tangled cord (as a word). The following lemmas will aid in our calculation.

**Lemma 3.3.4.** *Let $w$ be a word, $r = (u_0, u_1, u_2, \ldots, u_{n-1}, u_n)$ be a reduction of $w$ of size $n$, and $r_1, r_2, \ldots, r_n$ be the sequence of removed repeat words and return words corresponding to $r$. For all $1 \leqslant j < i \leqslant n$, if $r_i$ is contained in $u_j$, then*

$$r' = (u_0, u_1, u_2, \ldots, u_j, u_j - r_i, u_{j+1} - r_i, \ldots, u_{i-1} - r_i, u_{i+1}, \ldots, u_{n-1}, u_n)$$

*is a reduction of $w$ of size $n$.*

*Proof.* The result follows by noting that $r_{j+1}, r_{j+2}, \ldots, r_{i-1}$ are contained in $u_j - r_i, u_{j+1} - r_i, \ldots, u_{i-2} - r_i$, respectively. ■

**Lemma 3.3.5.** *Suppose $w = u_1 u_2 \cdots u_n$ for double occurrence words $u_1, u_2, \ldots, u_n \in \Sigma^+$ and let $\sigma$ be a permutation of $\{1, \ldots, n\}$. Then any reduction of $w$ is also a reduction of $w' = u_{\sigma(1)} u_{\sigma(2)} \cdots u_{\sigma(n)}$.*

*Proof.* For a reduction $r = (w_0, w_1, \ldots, w_m)$ of $w$, $1 \leqslant i \leqslant m$, and $1 \leqslant j \leqslant n$, suppose $w_i = w_{i-1} - v_i$ for some $v_i$ contained in $u_j$. Then we can construct a corresponding reduction $r' = (w'_0, w'_1, \ldots, w'_m)$ of $w'$ with $w'_i = w'_{i-1} - v_i$, where $v_i$ is contained in $u_{\sigma(j)}$. ■

**Lemma 3.3.6.** *For all $n \geqslant 3$, applying a reduction operation to $T_n$ yields $T_i T_{n-i-1}$ for some $0 \leqslant i \leqslant n-1$.*

*Proof.* Note that any reduction of $T_n$ necessarily begins with the removal of a literal repeat word since there are no non-literal repeat words in $T_n$. Hence, the result is a straightforward consequence of the definition of $T_n$—in particular, it follows by removing both occurrences of $i+1$ and relabeling appropriately.

■

**Lemma 3.3.7.** *For $m \geqslant 1$, let $T_{i_1}, T_{i_2}, \ldots, T_{i_m}$ be tangled cords of sizes $i_1, \ldots, i_m$, respectively, such that $i_k \geqslant 3$ for some $1 \leqslant k \leqslant m$. Then some minimal reductions of $T^m = T_{i_1} T_{i_2} \cdots T_{i_m}$ begin with the removal of both occurrences of a letter (i.e. a literal appearance of the repeat word).*

*Proof.* We proceed via induction on $m$. Note that all tangled cords of size 3 or greater contain no non-literal repeat words or return words. It follows that the result holds for $m = 1$. Now assume that the result holds for $m = n \in \mathbb{N}$. Let $T^{n+1} = T_{i_1} T_{i_2} \cdots T_{i_n} T_{i_{n+1}}$ and observe that we may assume without loss of generality that $i_1, \ldots, i_k \geqslant 3$ and $i_{k+1}, \ldots, i_n \leqslant 2$ for some $1 \leqslant k \leqslant n$ by Lemma 3.3.5. Then by Lemma 3.3.4, there exists a minimal reduction $r$ of $T^{n+1}$ that begins with a reduction $r'$ of $T_{i_1}$. Since $r$ is minimal, $r'$ is minimal, so by the induction hypothesis we may assume that the first step in $r'$ is the removal of a literal repeat word. But then that is also the first step in $r$, as desired. Therefore, we conclude the result by induction.

■

We can now calculate the pattern recurrence index of the tangled cord.

**Theorem 3.3.8.** *For all $n \geqslant 1$,*

$$\left\lceil \frac{2n}{3} \right\rceil - 1 \leqslant PI(T_n) \leqslant 2\left\lfloor \frac{n}{3} \right\rfloor + \left\lceil \frac{n \mod 3}{3} \right\rceil.$$

*In particular, this gives $PI(T_{3i-1}) = 2i - 1$ for all $i \geqslant 1$.*

*Proof.* Note that Lemma 3.3.7 implies that there exists a minimal reduction $r$ of $T_n$ in which literal repeat words are removed until we are left with a conjunction of tangled cords of size at most 2. Since Lemma 3.3.6 shows that any removal of a literal repeat word from $T_n$ produces $T_i T_{n-i-1}$ (after removing both occurrences of letter $i+1$), we can view the initial sequence of literal repeat word removals as edges in a "reduction tree" with vertices of tangled cord subwords. For definition purposes, suppose we have reduced $T_n$ to $T_{i_1} \cdots T_{i_k}$ for some $k \geqslant 1$ and assume there exists $1 \leqslant j \leqslant k$ such that $i_j \geqslant 3$. Then a removal of the literal repeat word $(i_1 + \cdots + i_j - 2)(i_1 + \cdots + i_j - 2)$ results in two new vertices, $T_{i_j - 3}$ and $T_2$, connected to $T_{i_j}$ in the reduction tree. Figure 3.2 shows an example reduction tree.

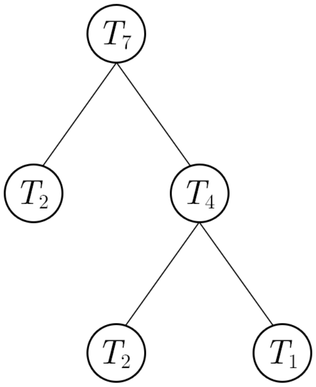

Figure 3.2: A reduction tree for the reduction $(T_7, T_2T_4, T_2T_2T_1, \epsilon)$ of $T_7$.

Each removal of a literal repeat word in a reduction of $T_n$ corresponds to two unique edges in the reduction tree, unless the literal repeat word includes the first or last letter from a tangled cord subword; in that case, only one edge is added to the reduction tree. Hence we infer that

$$R = T + L,$$

where $R = |r|$, $T$ is the number of tangled cord subwords of size 1 or 2 remaining after the initial sequence of literal repeat word removals [2], and $L$ is the number of literal repeat word removals in the initial sequence of reduction operations [3]. Letting $m$ be the size of word remaining after the initial sequence of literal repeat word removals, observe also that the following relations hold:

$$L = T + E - 1, \qquad T \geqslant \left\lceil \frac{m}{2} \right\rceil, \qquad m = n - L, \qquad E \geqslant 0,$$

[2] That is, the number of leaves in the reduction tree associated with $r$.

[3] That is, one half of the number of edges in the reduction tree of $r$.

where $E$ is the number of literal repeat word removals that involve deleting the first or last letter in a tangled cord subword in the reduction tree. Combining these, we see that

$$R = 2T + E - 1 \quad \text{and} \quad m = n - T - E + 1, \tag{3.3.22}$$

from which we infer that

$$T \geqslant \left\lceil \frac{n - T - E + 1}{2} \right\rceil \geqslant \left\lfloor \frac{n - E}{2} \right\rfloor - \left\lceil \frac{T}{2} \right\rceil + 1,$$

or

$$\left\lceil \frac{3T}{2} \right\rceil \geqslant \left\lfloor \frac{n - E}{2} \right\rfloor + 1.$$

Multiplying across by 2/3 yields the inequality

$$T \geqslant \frac{n - E}{3}.$$

Applying this to (3.3.22), we have

$$\begin{aligned} R &\geqslant \left\lceil 2\left(\frac{n-E}{3}\right) + E - 1 \right\rceil \\ &\geqslant \left\lceil \frac{2n + E}{3} \right\rceil - 1 \\ &\geqslant \left\lceil \frac{2n}{3} \right\rceil - 1 \end{aligned}$$

since $R$ is an integer. Thus it suffices to show that for all $T_n$ there exists a reduction

$$r = (T_n = T^1, T^2, \ldots, T^{L-1}, T^L, u_{L+1}, \ldots, u_{R-1}, \epsilon)$$

of size $2\lfloor n/3 \rfloor + \lceil (n \mod 3)/3 \rceil$. In that direction, suppose, for all $2 \leqslant i \leqslant L$, $T^i$ is obtained from $T^{i-1}$ by removing the repeat word 33[4] Then we show by induction on $n$ that $R = 2\lfloor n/3 \rfloor + \lceil (n \mod 3)/3 \rceil$. Manual calculation confirms that the base cases $n = 1, 2, 3$ satisfy the equality. Suppose, for $k \geqslant 2$, the result holds for $n < k$. Then by construction of

[4] Assuming relabeling into ascending order after each literal repeat word removal.

the reduction tree of $r$, we have

$$\begin{aligned} R(k) &= R(k-3) + R(2) + 1 \\ &= 2\left\lfloor \frac{k-3}{3} \right\rfloor + \left\lceil \frac{(k-3) \mod 3}{3} \right\rceil + 1 + 1 \\ &= 2\left\lfloor \frac{k}{3} - 1 \right\rfloor + \left\lceil \frac{k \mod 3}{3} \right\rceil + 1 + 1 \\ &= 2\left\lfloor \frac{k}{3} \right\rfloor + \left\lceil \frac{k \mod 3}{3} \right\rceil . \end{aligned}$$

Thus the desired equality holds by induction and we conclude the result.

■

Using this result, we have the following.

**Corollary 3.3.9.** *For all $n \geqslant 1$, there exists a loopless double occurrence word $w$ with $PI(w) = n$.*

*Proof.* Let $n \geqslant 1$ be given. Then by Theorem 3.3.8 there exists $i \geqslant 1$ such that either $PI(T_{2i}) = n$ or $PI(T_{2i}) = n - 1$, from which it follows that

$$PI(T_{2i}) = n \quad \text{or} \quad PI(T_{2i}T_2) = n.$$

Thus, $w = T_{2i}$ or $w = T_{2i}T_2$ is the desired loopless word.

■

**Remark 3.3.10.** *More straightforwardly, we can also simply demonstrate that there exists a double occurrence word $w$ with $PI(w) = n$ for all $n \geqslant 1$ by considering $w = 1122 \cdots nn$, an instance of the loop pattern of size $n$.*

Using Theorem 3.3.8, we now show that in general the tangled cord does not maximize the pattern recurrence index, even if we allow at most one loop. This may be surprising considering that in some ways tangled cords are the antithesis of repeat words and return words.[5]

[5] See Chapter 4 for details on this point.

**Theorem 3.3.11.** *For all $n \geqslant 8$, there exists a word $w$ of size $n$ with at most one loop such that $PI(w) > PI(T_n)$. Furthermore, there exists $k \in \mathbb{R}^+$ such that*

$$PI(w) \geqslant PI(T_n) + kn$$

*for sufficiently large $n$.*

*Proof.* For the double occurrence word $v = 1231435425$, observe that $PI(v) = 4$. Then by additivity, we infer that

$$PI(v^i T_j) = PI(v^i) + PI(T_j) = \begin{cases} 4i+1, & j = 1, \\ 4i+1, & j = 2, \\ 4i+2, & j = 3, \\ 4i+3, & j = 4, \end{cases}$$

while $|v^i T_j| = 5i + j$ for $i \geqslant 1$ and $1 \leqslant j \leqslant 4$. Note also that $v^i T_j$ has at most one loop. On the other hand,

$$PI(T_{5i+j}) \leqslant 2\left\lfloor \frac{5i+j}{3} \right\rfloor + 1 \leqslant 2\left\lfloor \frac{5i}{3} \right\rfloor + 5 \leqslant \frac{10i}{3} + 5,$$

giving

$$PI(v^i T_j) - PI(T_{5i+j}) \geqslant \begin{cases} 4i+1-\frac{10i}{3}-5, & j = 1, \\ 4i+1-\frac{10i}{3}-5, & j = 2, \\ 4i+2-\frac{10i}{3}-5, & j = 3, \\ 4i+3-\frac{10i}{3}-5, & j = 4, \end{cases} = \begin{cases} \frac{2i-12}{3}, & j = 1, \\ \frac{2i-12}{3}, & j = 2, \\ \frac{2i-9}{3}, & j = 3, \\ \frac{2i-6}{3}, & j = 4, \end{cases} \tag{3.3.23}$$

for $i \geqslant 1$ and $1 \leqslant j \leqslant 4$. Hence $PI(v^i T_j) - PI(T_{5i+j})$ is greater than 0 for $i \geqslant 8$ and $0 \leqslant j \leqslant 4$. For $i, j$ such that $8 \leqslant |v_i T_j| \leqslant 40$, it can be checked by hand that the result holds. This yields the first part of the result. The second part with $k \approx 2/15$ follows from (3.3.23).

■

Note that in the above proof, for most $n$, $w$ does not have any loops. We now present a strengthening of Corollary 3.3.9 based on the observation that for a word $w$, adding a loop to $w$ such that the loop does not become "part" of a larger return word in $w$ necessarily increases the pattern recurrence index of $w$ by 1.

**Proposition 3.3.12.** *For all $n \in \mathbb{N}$ and $1 \leqslant i \leqslant n$, there exists a word $w$ of size $n$ such that $PI(w) = i$.*

*Proof.* Let $n$ be given and let $j = i - 1$. For

$$w = 1122 \cdots jj(j+1)(j+2) \cdots (n)(j+1) \cdots (n),$$

we have

$$PI(w) = PI(1122 \cdots jj) + PI((j+1)(j+2) \cdots (n)(j+1) \cdots (n)) = j + 1 = i$$

by additivity. This gives the result for all cases except $i = 1$; in that case, we let $w$ be a repeat word of size $n$.

■

## 3.4 Nesting Index

We analyze the nesting index of the tangled cord, defined below [40].

**Definition 3.4.1.** *For a set of recursive patterns $\Pi$, let $\mathcal{R}_\Pi$ denote the set of all instances of all recursive patterns $\pi \in \Pi$ and let $w$ be a double occurrence word. Then a word $u$ is said to be a* maximal *pattern instance in $w$ if $u \preceq w$, $u \in \mathcal{R}_\Pi$, and $u \preceq v$ and $v \preceq w$ implies that $v \notin \mathcal{R}_\Pi$ or $u = v$.*

**Definition 3.4.2.** *For a set of recursive patterns $\Pi$ and a word $w$, we say $w'$ is obtained from $w$ by a* maximal reduction operation *if*

$$w' = w - \{u \mid u \textit{ is a maximal pattern instance in } w\}.$$

*We say $w'$ is obtained from $w$ by a* letter removal *if for some $a \in \Sigma$, $w' = w - a$.*

**Definition 3.4.3.** *Given a set of recursive patterns* $\Pi$*, a* reduction *of a word* $w$ *is a sequence of words* $(w_0, w_1, \ldots, w_n)$ *in which*

1. $w_0 = w$,
2. *for all* $0 \leqslant k < n$, $w_{k+1}$ *is obtained from* $w_k$ *by applying a maximal reduction operation or a letter removal.*

**Definition 3.4.4.** *Letting* $\Pi$ *contain the strictly-appearing repeat word and strictly-appearing return word,*

$$NI(w) := \min\{n \mid (u_0, u_1, \ldots, u_n = \epsilon) \text{ is a reduction of } w\}$$

*is the nesting index of the double occurrence word* $w$*.*

**Example 3.4.5.** *Since* $(12132345676754, 121245676754, 4554, \epsilon)$ *is a minimal reduction of the word* $12132345676754$*,* $NI(12132345676754) = 3$*.*

It is not difficult to see that Lemmas 3.3.4, 3.3.5, 3.3.6, and 3.3.7 all still hold for this modified notion of a reduction, where reduction operation now refers to either a maximal reduction operation or a letter removal. As in the case of the pattern recurrence index, we can similarly use these results to calculate the nesting index of the tangled cord.

**Proposition 3.4.6.** *For all* $n \geqslant 1$*,*

$$\lfloor n/3 \rfloor \leqslant NI(T_n) \leqslant \lfloor n/3 \rfloor + 1.$$

*Proof.* Modulo small modifications, the proof mirrors the argument used to determine the pattern recurrence index of the tangled cord. Lemma 3.3.7 implies that there exists minimal reduction(s) of $T_n$ that begin with a sequence of letter removals until we are left with a conjunction of tangled cords of size at most 2. Thus, we can similarly associate each reduction $r$ (prior to applying maximal reduction operations) with a reduction tree. Note that

$$R := |r| = T + E, \tag{3.4.24}$$

where $E$ is the number of removals of the first or last letter in a tangled cord subword in the reduction tree corresponding to $r$. It follows that (3.3) becomes

$$T \geqslant \left\lceil \frac{m}{2} \right\rceil, \qquad m = n - R + 1, \qquad E \geqslant 0,$$

where $m$ is similarly defined as the size of the word after all letter removals. Combining these relations and (3.4.24), we have

$$R \geqslant T \geqslant \left\lceil \frac{m}{2} \right\rceil \geqslant \left\lceil \frac{n-R+1}{2} \right\rceil \geqslant \left\lfloor \frac{n}{2} \right\rfloor - \left\lceil \frac{R}{2} \right\rceil,$$

implying that

$$\left\lceil \frac{3R}{2} \right\rceil \geqslant \left\lfloor \frac{n}{2} \right\rfloor.$$

Multiplying across by 2/3 yields the inequality $R \geqslant \lfloor n/3 \rfloor$. Thus it suffices to show that for all $T_n$ there exists a reduction $r = (T_n, u_1, \ldots, u_{R-1}, \epsilon)$ of size $\lfloor n/3 \rfloor + 1$. In that direction, suppose, for all $1 \leqslant i \leqslant R-1$, $u_i$ is attained from $u_{i-1}$ by removing 3 (assuming relabeling to ascending order after each application of reduction operation 2). Then after $\lfloor n/3 \rfloor$ letter removals, what remains is a concatenation of tangled cords of size 1 or 2. Since these are repeat words on disjoint alphabets, they are all removed in one maximal reduction operation. Hence $|r| = \lfloor n/3 \rfloor + 1$, as desired.

■

In 2013, Ryan Arredondo conjectured that for all $n \in \mathbb{N}$, there exists a word of size $n + \lfloor \sqrt{n-1} \rfloor$ with nesting index $n$ [40]. By contrast, the tangled cord with nesting index $n$ is approximately of size $3n$. Although we do not make any headway on this stronger conjecture, Proposition 3.4.10 presents a counterexample to the conjecture that the tangled cord maximizes the nesting index.

Arredondo defined a double occurrence word $w$ as being *1-reducible* if there exists a reduction $(u_0, u_1, \ldots, u_n)$ of $w$ such that for all $i$, $u_{i+1}$ is obtained from $u_i$ by applying a maximal reduction operation. We can visualize double occurrence words using *chord diagrams*, pictorial representations of a word $w$ obtained by arranging the letters of $w$ around the circumference of a circle and joining the two occurrences of each letter of $w$ by a chord of the circle (see Figure 3.4). We use a line placed perpendicular to the circle to indicate the start of the word. A chord diagram $C'$ is called a *sub-chord diagram* of a chord diagram $C$ if the chords of $C'$ make up a subset of the chords of $C$. Arrendondo discovered the following forbidden sub-chord diagram characterization of 1-reducible words:

**Theorem 3.4.7.** *[40] A word $w$ is 1-reducible if and only if the chord diagram of $w$ does not contain the chord diagram in Figure 3.3 as a sub-chord diagram.*

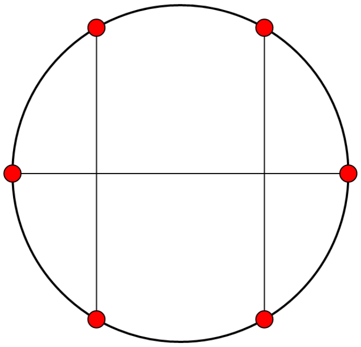

Figure 3.3: The chord diagram associated with the words 121323, 123213, and 123132.

**Remark 3.4.8.** *Unlike the nesting index, there is no forbidden sub-chord diagram characterization of a 1-reducible word $w$. This follows from considering the chord diagrams in Figure 3.4—even when taking into account the starts/ends of the words, the chord diagram of* 12321434 *contains all of the sub-chord diagrams of the chord diagram of* 121323*, yet the former is 1-reducible while the latter is not.*

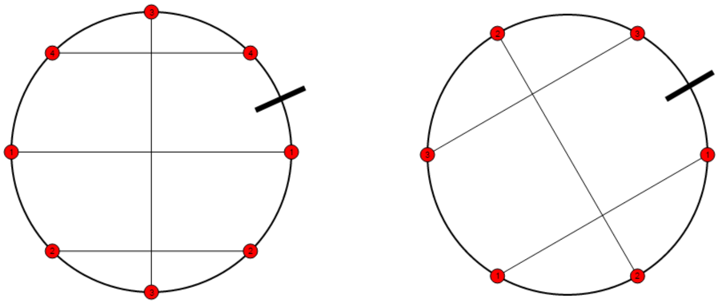

Figure 3.4: The chord diagrams of the words 12321434 and 121323, respectively.

We now present a result demonstrating the independence of the nesting index and the pattern recurrence index.

**Proposition 3.4.9.** *For all* $(m,n) \in \{1,2,3,\ldots\} \times \{2,3,4,\ldots\}$, *there exists a word* $w$ *with* $NI(w) = m$ *and* $PI(w) = n$.

*Proof.* We separate the proof into three cases.

**Case 1**. Suppose first that $2 \leqslant n \leqslant m/2$ and let

$$\begin{aligned} w = (1, mn) \left( \left\lfloor \frac{m-1}{n-1} \right\rfloor, 1 \right) \left( 2 \left\lfloor \frac{m-1}{n-1} \right\rfloor, \left\lfloor \frac{m-1}{n-1} \right\rfloor + 1 \right) \\ \cdots \left( (n-2) \left\lfloor \frac{m-1}{n-1} \right\rfloor, (n-3) \left\lfloor \frac{m-1}{n-1} \right\rfloor + 1 \right) \\ \left( m-1, (n-2) \left\lfloor \frac{m-1}{n-1} \right\rfloor + 1 \right) (mn, m), \end{aligned}$$

where we use $(i,j)$ to denote the word $(i)(i-1)\cdots(j+1)(j)$ or the word $(i)(i+1)\cdots(j-1)(j)$ for $i \geqslant j$ or $i \leqslant j$, respectively. We proceed to show that $w$ attains the desired values of the nesting and pattern recurrence indices. Note that $w$ is indeed a (double occurrence) word and that it has no contiguous repeat or return words, although it is a composition of $n$ non-contiguous return words of size at least 2 since

$$\left\lfloor \frac{m-1}{n-1} \right\rfloor \geqslant 2.$$

Thus to calculate the nesting index, it suffices to calculate the number of letter removals required to reduce $w$ to a contiguous repeat or return word. For a contiguous return word, it is clear that we must remove all the letters between the two instances of some letter of the desired contiguous return; consequently, by inspection, we see that the minimum number of letter removals is $m-1$, obtained by deleting the letters $1, 2, \ldots, m-1$. This leaves a single contiguous return word, implying that $NI(w) \leqslant m$. On the other hand, to reduce $w$ to a contiguous repeat word, note that we can include at most one letter from each return word of $w$ in the desired contiguous repeat word. Hence we must remove all the other letters of $w$. Since there are $n$ return words in $w$, this gives a reduction of size at least $mn - n + 1 > m$. Hence we conclude that $NI(w) = m$, as desired. For the pattern recurrence index of $w$, it suffices to show that

$$PI((1, i_n)(i_1, 1)(i_2, i_1 + 1) \cdots (i_n, i_{n-1} + 1)) = n$$

for $i_1, \dots, i_n \geqslant 2$ such that $i_k - i_{k-1} \geqslant 2$ for all $2 \leqslant k \leqslant n$ since $w$ is a word of this form. In that direction, we induct on $n$. The base case with $n = 2$ is clear:

$$PI((1)(2)\cdots(i_2)(i_1)(i_1-1)\cdots(1)(i_2)(i_2-1)(\cdots)(i_1+1)) = 2$$

since there is no reduction of size 1, yet deleting the return word $12\cdots i_1 i_1 \cdots 21$ followed by the return word $(i_1+1)(i_1+2)\cdots(i_2)(i_2)\cdots(i_1+2)(i_1+1)$ yields a reduction of size 2. Now suppose the result holds for $n = k$ and consider the word

$$v_{k+1} = (1, i_{k+1})(i_1, 1)(i_2, i_1+1)\cdots(i_{k+1}, i_k+1).$$

As previously, observe that we can choose at most one letter from each return word to construct a repeat word, so a reduction using such a strategy has size at least $k+1$. A reduction $r$ which proceeds by removing a return word from $v_k$ either yields a word $w'$ in the form of $v_{k+1}$ or $v_k$. In the former case, we let $v_{k+1} = w$ and start over. In the latter case, we have $|r| \geqslant PI(v_k) + 1 \geqslant k+1$, as desired. By induction, we conclude that $PI(v_n) = n$. This yields the result for $2 \leqslant n \leqslant m/2$.

**Case 2.** Now suppose that $m/2 < n \leqslant m$ and let

$$w_{m,n} = (1, m+n-1)(2,1)(4,3)\cdots(2n-2, 2n-3)(m+n-1, 2n-1).$$

It is clear that a straightforward replication of the arguments in Case 1 also shows that $NI(w_{m,n}) = m$ and $NI(w_{m,n}) = n$, since $w_{m,n}$ as defined here is also a conjunction of non-contiguous return words of size at least 2.

**Case 3.** Now suppose that $1 \leqslant m \leqslant n$ and let $w = l_{n-m} w_{m,m}$, where $l_0 = \epsilon$, $l_i = 1122\cdots ii$, and $w_{m,m}$ is as defined in Case 2. By additivity of the pattern recurrence index, it suffices to show that $NI(w_{m,m}) = PI(w_{m,m}) = m$ since

$$PI(w) = PI(l_{n-m}) + PI(w_{m,m}) = n - m + PI(w_{m,m}).$$

Yet Case 2 shows that $NI(w_{m,m}) = PI(w_{m,m}) = m$, so we conclude the result for this case as well.

■

The fact that the tangled cord does not necessarily maximize the nesting index, even amongst strongly irreducible words, is somewhat easier to recognize than for the pattern recurrence index. This should not be surprising given that the proof of Theorem 3.4.6 essentially shows that "half" of the tangled cord is disjoint repeat words, which can all be removed in a single reduction operation.

**Proposition 3.4.10.** *For all $m \geqslant 2$, there exists a strongly irreducible word $w$ of size $2m$ such that $NI(w) > NI(T_{2m})$.*

*Proof.* For $m \geqslant 2$, let

$$v_m = (1, 2m)(2m-1)(2m-3)\cdots(3)(1)(2)(4)\cdots(2m) \tag{3.4.25}$$

(see Figure 3.5). We proceed via induction on $m$ to show that $NI(v_m) = m$, which by

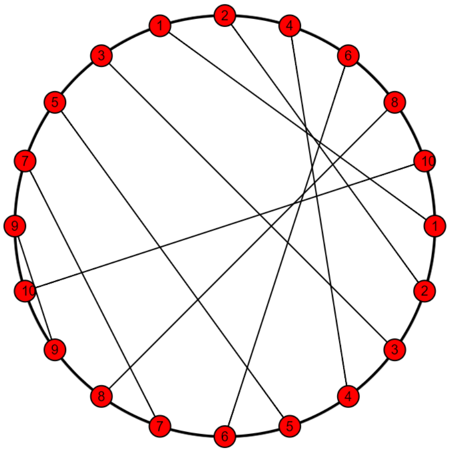

Figure 3.5: The chord diagram of $v_5 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 9, 7, 5, 3, 1, 2, 4, 6, 8, 10$.

Proposition 3.4.6 gives the desired result. The base case with $m = 2$ is trivial, $v_2$ is simply the repeat word 1212. Assume then that $NI(v_m) = m$ for all $1 \leqslant m < n$, where $n \geqslant 2$. Suppose we are given a reduction $r = (v, u_1, u_2, \ldots, u_{p-1}, \epsilon)$ of $v_n$. Without loss of generality, we may assume that there exists $1 \leqslant i \leqslant p-1$ such that $u_i$ contains a contiguous repeat or return word $i_1 i_2 \cdots i_{2k-1} i_{2k}$ of size $1 \leqslant k \leqslant n+1$. Furthermore, we may take $u_i$ to be the first word in $r$ to contain such a contiguous repeat or return word. Suppose it is a repeat word, that is, $i_j = i_{k+j}$ for $1 \leqslant j \leqslant k$. Then clearly $i_2, \ldots, i_k$ are even and $u_i$ contains only one contiguous repeat word, so counting implies that we need have removed $2n - k$ letters beforehand, that is, $u_i = i_1 i_2 \cdots i_{2k-1} i_{2k}$. Since $n + 1 \geqslant k$, we see that $|r| \geqslant 2n - k + 1 \geqslant n$. Then suppose instead that $i_1 i_2 \cdots i_{2k-1} i_{2k}$ is a return word, or $i_j = i_{2k-j+1}$. Then we similarly see that unless $k = 1$ and $i_1 = 2n$, $i_1, \ldots, i_{k-1}$ are odd and $u_i$ contains only one contiguous return word. If $k = 1$ and $i_1 = 2n$, then we need to have removed $2n - 1$ letters beforehand and thus $|r| = 2n$. Otherwise, there are two cases, namely where $i_k$ is odd and where $i_k$ is even. If $i_k$ is even, then counting implies that we need have removed $2n - k$ letters beforehand, and thus $|r| \geqslant n$, as before. If instead $i_k$ is odd, then note that we need have removed all letters greater than $i_1$, save $i_2, \ldots, i_k$; hence

$$i \geqslant \left\lceil \frac{2n - i_1}{2} \right\rceil$$

and we may assume without loss of generality that

$$u_i = (1, i_1)(i_2)(i_3) \cdots (i_k)(i_k)(i_{k-1}) \cdots (i_1)(i_1 - 2)(i_1 - 4) \cdots (1)(2)(4) \cdots (i_1 - 1)$$

since if $r$ involves removing any letters among $1, 2, \ldots, i_1 - 1$ prior to $u_i$, then it is easy to see that there exists a reduction $r'$ of equivalent size with those removals occurring after the $i$th step. Either the reduction from $u_i$ to $u_{i+1}$ involves an application of a maximal reduction operation or a letter removal. Note that there can be at most one contiguous repeat or return word in $u_j$ for all $j$, so it follows that there exist minimal reductions $r$ which proceed via a maximal reduction operation at the $(i + 1)$th step. This gives

$$u_{i+1} = (1, i_1 - 1)(i_1 - 2)(i_1 - 4) \cdots (1)(2)(4) \cdots (i_1 - 1)$$

which by the induction hypothesis has nesting index $(i_1 - 1)/2$. Thus

$$|r| \geqslant \left\lceil \frac{2n - i_1}{2} \right\rceil + \frac{i_1 - 1}{2} + 1 \geqslant \frac{2n - i_1}{2} + \frac{i_1 + 1}{2} \geqslant n,$$

so we conclude that $NI(v_n) \geqslant n$. Finally, note that removing all odd letters except 1 leaves a single contiguous repeat word, implying that there exists a reduction of size $n$ and thus $NI(v_n) = n$, as desired. Consequently, by induction, we infer that $NI(v_m) = m$.

■

# 4 Highly Scrambled Genome Rearrangements in *O. trifallax*

Genome rearrangement processes are observed in many species, on both evolutionary and developmental scale. *Oxytricha trifallax*, a species of ciliate, undergoes massive genome rearrangements during the development of a somatic macronucleus (MAC) from a germline micronucleus (MIC) and is used as a model organism to study DNA rearrangements [41]. During the macronuclear development, thousands of genetic segments are rearranged to form gene-sized chromosomes. Pairs of short homologous ($1-20$ bps) DNA sequences called *pointers* are present at the ends of consecutive segments in the MIC and are considered to play a significant role in the recombination process. By representing pointer loci by symbols, we represent the scrambled genes by *double occurrence words* (DOW), words with each letter appearing exactly twice.

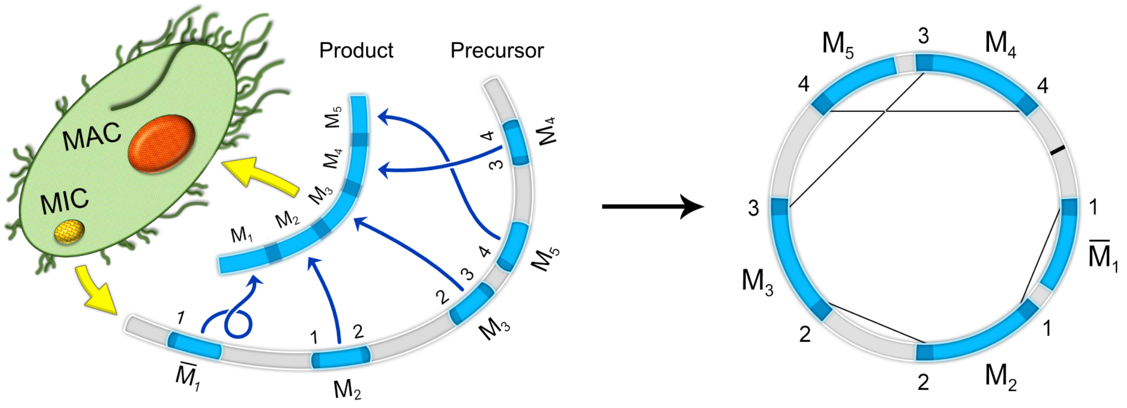

Figure 4.1: DNA rearrangement in *Oxytricha trifallax*. The MDS sequence $\overline{\mathrm{M}}_1\mathrm{M}_2\mathrm{M}_3\mathrm{M}_5\mathrm{M}_4$ with pointer sequence 11223434 (left) can be represented as a double occurrence word and visualized using a chord diagram (right).

These situations are schematically depicted in Figure 4.1. In the figure, the segments are located in a longer band representing a MIC contig in the order $\overline{M_1}M_2M_3M_5M_4$ with interruption by other segment. The ends of the segments destined to assemble in a nano chromosome are labeled by integers representing the pointers, corresponding to a double occurrence word 11223434.

In [42], it was observed that repeat and return patterns of double occurrence words are present very often, and it was shown that words with pattern indices at most 5 explain scrambling of over 95% of the genes. Out of 2021 scrambled sequences studied in *O. trifallax*, 1948 reduced to the empty word with reductions by the set of patterns $P = \{\alpha\alpha, \alpha\alpha^R\}$ and only with maximal reduction operations, implying that they are compositions of nested repeat and return words. In these reductions we considered only pattern instances that are not literal (the instances had at least two distinct symbols). Twenty-two scrambled sequences were identified which retained at least four letters at the end of the reduction operations indicating that the repeat and return patterns do not describe well these highly scrambled rearrangements.

An analysis of the resulting reduced 22 double occurrence words was performed in an attempt to find new common patterns. Upon inspection of the 22 words, the existence of an embedded pattern called the *tangled cord* was identified as a common pattern. A majority of these embedded tangled cords are cyclically equivalent to tangled cords which we also call tangled cords. Two of the words are themselves tangled cords, while 7 of them are realized as a combination of tangled cords after a single letter removal. Additionally, 7 words are a combination of tangled cords after inserting 1 letter, 2 are a combination of tangled cords after swapping two adjacent letters, and 3 are a combination of tangled cords after removing or inserting 2 letters. The largest reduced word in the set (with 17 symbols) is the only one that does not appear to be close to a combination of tangled cords.

To more systematically determine whether the tangled cord commonly appears in the 22 highly scrambled rearrangements, we computed by brute force search three pattern indices $P = \{\alpha\alpha, \alpha\alpha^R\}$, $P' = \{\alpha\alpha, \alpha\alpha^R, T_n\}$, and $P'' = \{T_n, aa\}$ (for literally-appearing $\alpha\alpha \mapsto aa$), respectively, for each of the 22 words. We then compared these computations to the average of these indices on three random samples of 22 words with the same distribution of word sizes as 22 highly scrambled cases; that is, if there are $n$ words of size $k$ among the 22 highly scrambled rearrangements, we uniformly sample $n$ words of size $k$ at random from the set of all double occurrence words of size $k$. Table 4.1 gives a summary of the computations.

|  | Average |  |  |
|---|---|---|---|
|  | $I_P$ | $I_{P'}$ | $I_{P''}$ |
| Highly scrambled cases | 3.91 | 3.59 | 3.91 |
| Random sample | 3.50 | 3.29 | 4.36 |

Table 4.1: Compared with an identically distributed random sample of 22 words, the 22 highly scrambled cases exhibit significantly lower averages on indices that include the tangled cord pattern.

Given that all maximal repeat words and return words have been removed from the 22 reduced words, the the repeat-return pattern index is on average significantly greater on the 22 words than on a random sample. After adding the tangled cord into the pattern set, the difference between the average pattern index on the two sets of words reduces from 0.41 to 0.3, indicating that the tangled cord is encountered more often in a reduction of the 22 highly scrambled cases than in a reduction of a random sample. This is confirmed by the average index $I_{P''}$, which is significantly greater for the random samples than for the 22 reduced words. Overall, the pattern index computations indicate that the tangled cord may be another commonly appearing pattern in scrambled genomes.

# 5 Conclusion

We developed a generalization of the notion of a pattern that allows a pattern to appear as a subword, rather than only as a factor as has traditionally been studied in the literature. We then introduced the notions of reductions and paths between words using reduction operations involving the removal of a (generalized) pattern. These reductions and paths were used to define word distance and pattern indices, measures of the similarity of two words and the complexity of a word with respect to a given set of patterns, respectively. Despite the fact that we made progress in computing the word distance with the repeat word $\alpha\alpha$, and conjecture that this is indeed possible for any two words, this problem is likely to be infeasible for arbitrary patterns; that is, it seems that there may not be a general algorithm for computing the word distance between two words relative an arbitrary set of patterns. The proof of Lemma 2.2.6, which seems likely to be a necessary first step towards proving Conjecture 2.2.9, points in this direction since it heavily relies upon the exceptional properties of the repeat word.

The situation became more tractable when we restricted to biologically-motivated double occurrence words, even after further generalizing our notion of a pattern with the introduction of recursive patterns, which essentially take into account similarities between patterns (under our definition, allowing them to be considered together as a single (recursive) pattern). In this case, for certain relevant sets of recursive patterns that satisfy the completeness property, we proved that computing the word distance is indeed feasible since there exists a minimal path from $u$ to $v$ of the form $(r)$ or $(r_1, r_2^R)$ for reduction $r$, $r_1$, and $r_2$. This result allowed us to apply the word distance with the repeat word and return word in analyzing 22 highly scrambled DNA rearrangements in *Oxytricha trifallax.* Continuing work started by Ryan Arredondo in 2013 [40], we also studied several pattern indices, the pattern recurrence index and the nesting index, and used them to identify a new common pattern, the tangled cord, in the 22 highly scrambled rearrangements.

Many open questions remain. For example, Arredondo's conjecture that

$$\min\{|w| \mid NI(w) = n\} = 2(n + \lfloor\sqrt{n-1}\rfloor)$$

remains unresolved, although we did confirm that the tangled cord does not maximize the nesting index or pattern recurrence index, even among strongly irreducible words. A proof of the computability of the word distance relative the pattern $\alpha\alpha$ could be straightforwardly modified to demonstrate the computability of the distance with the return word, or perhaps any subset of $\{\pi_R, \pi'_R\}$. It may be of interest to determine which sets of patterns admit computable word distances and which do not. It would also be of interest to determine if there are other classes of words, besides double occurrence words, that admit computable distances, or have fast algorithms for computing distances. It is also unclear how restricting patterns (or recursive patterns) to appearing strictly and/or literally affects the computability of word distances. We also did not consider the problem of determining, for arbitrary sets of patterns, when two words belong to the same connected component and, in particular, when a given word belongs to the same connected component as the empty word $\epsilon$. It could be the case that these two problems are also intractable in general, but have interesting answers for certain types of patterns or classes of words.

Paths between words naturally define a global graph of words, where vertices are words and edges connect two words that differ by a single pattern instance. Studies of the structures of these graphs may reveal other relationships between classes of words. For example, it may be of interest to see if these graphs have a finite number of connected components; we conjecture that this is true for patterns that are confluent.

# References

[1] C. Choffrut, J. Karhumăki, Combinatorics of Words, *Handbook of Formal Languages*, (Eds. G. Rozenberg, A. Salomaa) Springer–Verlag (1997) 329–438.

[2] M. Lothaire, Algebraic Combinatorics on Words, *Encyclopedia of Mathematics and its Applications* (Cambridge University Press) (2002).

[3] M. Lothaire, Applied Combinatorics on Words, *Encyclopedia of Mathematics and its Applications* (Cambridge University Press) (2005).

[4] A. Thue, Über unendliche Zeichenreihen, *Norske Vid. Skrifter I Mat.-Nat. Kl.*, Christiania **7** (1906) 1–22.

[5] A. Thue, Über die gegenseitige Lage gleicher Teile gewisser Zeichenreihen, *Norske Vid. Skrifter I Mat.-Nat. Kl.*, Christiania **1** (1912) 1–67.

[6] D. Bean, A. Ehrenfeucht, G. McNulty, Avoidable patterns in strings of symbols, *Pacific J. Math.* **85** (1979) 261–294.

[7] A. I. Zimin, Blocking sets of terms, *Mathematics of the USSR-Sbornik*, **47** (1984) 353.

[8] A. I. Zimin, Blocking sets of terms, *Mat. Sb. (N.S.)*, **119** (1982) 363–375.

[9] K. A. Baker, G. F. Mcnulty, and W. Taylor, Growth problems for avoidable words, *Theoretical Computer Science*, **69** (1989) 319–345.

[10] U. Schmidt, Motifs inévitables dans les mots, Doctoral thesis, Université Paris 6, LITP research report (1986) 63–86.

[11] U. Schmidt, Avoidable patterns on two letters, *Theoretical Computer Science*, **63** (1989) 1–17.

[12] J. Cassaigne, Unavoidable binary patterns, *Acta Informatica*, **30** (1993) 385–395.

[13] J. Cassaigne, Motifs évitables et régularités dans les mots, Doctoral thesis, Université Paris 6, LITP research report TH 94-04 (1994).

[14] P. Ochem, A generator of morphisms for infinite words, *RAIRO-Theoretical Informatics and Applications*, **40** (2006) 427–441.

[15] J. D. Currie, Open problems in pattern avoidance, *American Mathematical Monthly*, **100** (1993) 790–793.

[16] K. Abrahamson, Generalized string matching, *SIAM Journal of Computing*, **16** (1987) 1039–1051.

[17] A. Amir, A. Aumann, R. Cole, M. Lewenstein, E. Porat, Function matching: Algorithms, applications, and a lower bound, *Proc. 30th ICALP*, (2003) 929–942.

[18] A. Apostolico, Z. Galil (Eds.), Pattern Matching Algorithms, *Oxford University Press*, Oxford (1997).

[19] B.S. Baker, A theory of parameterized pattern matching: algorithms and applications, *Proc. 25th Annual ACM Symposium on the Theory of Computation*, (1993) 71-80.

[20] D.E. Knuth, J.H. Morris, V.R. Pratt, Fast pattern matching in strings, *SIAM Journal of Computing*, **6** (1977) 323-350.

[21] A. Amir, I. Nor, Generalized function matching *Journal of Discrete Algorithms*, 5 (2007) 514–523.

[22] Angluin D, Finding patterns common to a set of strings, *Journal of Computer and System Sciences*, **21**:1 (1980) 46–62.

[23] Y. K. Ng, T. Shinohara, Developments from enquiries into the learnability of the pattern languages from positive data, *Theoretical Computer Science*, **397** (2008) 150–165.

[24] Reidenbach D, Schmid M L, Patterns with bounded treewidth, *Information and Computation*, **239** (2014) 87–99.

[25] Fernau H, Schmid M L, Pattern matching with variables: A multivariate complexity analysis, *Information and Computation*, **242** (2015) 287–305.

[26] Fernau H, Manea F, Mercaş R, Schmid M L, Pattern Matching with Variables: Fast Algorithms and New Hardness Results, Proceedings of the 32nd Symposium on Theoretical Aspects of Computer Science (STACS), (2015) 302–315.

[27] G. Navarro, A guided tour to approximate string matching, *ACM Computing Surveys*, **33**:1 (2001) 31–88.

[28] V. Levenshtein, Binary codes capable of correcting spurious insertions and deletions of ones, *Probl. Inf. Transmission*, **1** (1965) 8–17.

[29] D. Sankoff and J. Kruskal, Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison, (Addison-Wesley) (1983).

[30] S. Needleman and C. Wunsch, A general method applicable to the search for similarities in the amino acid sequences of two proteins, *Journal of Molecular Biology*, **48** (1970) 444–453.

[31] F. J. Damerau, A technique for computer detection and correction of spelling errors, *Communications of the ACM*, **7**:3 (1964) 171–176.

[32] E. Ukkonen, On approximate string matching, *Foundations of Computation Theory*, (1983) 487–495.

[33] E. Ukkonen, Finding approximate patterns in strings, *Journal of Algorithms*, **5** (1985) 132–137.

[34] E. Ukkonen, Algorithms for approximate string matching, *Information and Control*, **64** (1985) 100–118.

[35] R. Wagner and M. Fischer, The string-to-string correction problem, *Journal of the ACM*, **21** (1974) 168–178.

[36] R. Baeza-Yates and G. Navarro, A faster algorithm for approximate string matching, *Combinatorial Pattern Matching, LNCS 1075*, (1996) 1–23.

[37] R. Baeza-Yates and G. Navarro, Fast Approximate String Matching in a Dictionary, *Proceedings of SPIRE'98, IEEE CS Press*, (1998) 14–22.

[38] G. Myers, A fast bit-vector algorithm for approximate string mathcing based on dynamic programming, *Journal of the ACM*, **46**:3 (1999) 395–415.

[39] Burns J, Kukushkin D, Lindblad K, Chen X, Jonoska N, Landweber LF, <mds_ies_db>: a database of ciliate genome rearrangements, *Nucleic Acids Research*, **44** (2016) (Database issue) doi: gkv1190.

[40] Arredondo R, Reductions On Double Occurrence Words. Proceedings of the Forty-fourth Southeastern International Conference on Combinatorics, Graph Theory and Computing, *Congressus Numerantium*, **218** (2013) 43–56.

[41] X. Chen, J. R. Bracht, A. D. Goldman, E. Dolzhenko, D. M. Clay, E. C. Swart, D. H. Perlman, T. G. Doak, A. Stuart, C. T. Amemiya, R. P. Sebra, and L. F. Landweber, The Architecture of a Scrambled Genome Reveals Massive Levels of Genomic Rearrangement during Development, *Cell*, **158**:5 (2014) 1187–1198.

[42] J. Burns, D. Kukushkin, X. Chen, L. F. Landweber, M. Saito, and N. Jonoska, Re-occurring Patterns Among Scrambled Genes in the Encrypted Genome of the Ciliate *Oxytricha trifallax*, *Journal of Theoretical Biology*, **410** (2016) 171–180.

[43] Braun J, Nabergall L, Landweber L F, Saito M, Jonoska N, Complex nested and scrambled rearrangements in the genome of *Oxytricha trifallax*, in preparation.