THE MORPHISMS WITH UNSTACKABLE IMAGE WORDS

C. ROBINSON TOMPKINS

Abstract. In an attempt to classify all of the overlap-free morphisms constructively using the Latin-square morphism, we came across an interesting counterexample, the Leech square-free morphism. We generalize the combinatorial properties of the Leech square-free morphism to gain insights on a larger class of both overlap-free morphisms and square-free morphisms.

1. Introduction

The study of overlap-free words and their generators was originated by Axel Thue in 1912 [9]. Thue stumbled across overlap-free words in the attempt to find infinite words that are cube-free. We quickly note that \(XXX\) is a cube where \(X\) is some string of symbols, and a word \(W\) avoids cubes if there is no subword \(XXX\) in \(W\). We know the infinite binary word that avoids cubes to be the Thue-Morse infinite word,

\[
011010011001011010011001101001\ldots
\]

[9]. This infinite word can be generated by function composition of the Thue-Morse morphism \(\mu\) on the letter 0. Note that the Thue-Morse morphism is defined as

\[
\mu(t) = \begin{cases} 
01 & \text{if } t = 0 \\
10 & \text{if } t = 1.
\end{cases}
\]

Further we define a morphism as a mapping \(h : \Sigma^* \rightarrow \Delta^*\) with \(\Sigma, \Delta\) being alphabets such that for any two words \(V, W \in \Sigma^*\), we have \(h(VW) = h(V)h(W)\). A morphism \(h\) is called cube free provided \(h(W)\) is cube-free if and only if \(W \in \Sigma^*\) is also cube-free.

Our primary concern however is dealing with overlaps instead of cubes. An overlap is the pattern \(cXcXc\) where \(c\) represents a single letter and \(X\) is a word with possibly zero letters. The standard example of a word that is an overlap in its entirety is “alfalfa”, and an overlap-free word is a word in which no overlap occurs.

A morphism \(h\) is said to be overlap free so long as we have \(X \in \Sigma^*\) overlap-free if and only if \(h(X)\) is overlap-free. Surprisingly it is known that \(\mu\) and its natural complement are the only non-trivial overlap-free morphisms on the two letter alphabet \(\{-0, 1\}\) [2].

In the early 80’s Crochemore, Ehrenfeucht, and Rozenberg made substantial progress towards classifying the square-free morphisms [3],[4]. Further in 2004, Richomme and Wlazinski published a result classifying all overlap-free morphisms [8]. However, their result much like the results of Crochemore, Ehrenfeucht, and Rozenberg rely on test-sets of words for the morphism in question. Furthermore, the tests for Richomme and Wlazinski grow factorially with the size of the input alphabet.

Since the late 1990’s and early 2000’s, several results have surfaced pursuing a constructive understanding of the class of overlap-free morphisms. In 2001, Frid suggested using the structure of the cyclic group of order \(n\) to define each image word accordingly for a morphism on an alphabet of \(n\) letters [5]. In 2007 we extended Frid’s result to the use of the Latin-square structure to define our morphism structure [10].

In a vain attempt to use the Latin-square morphism construction to classify all of the overlap-free morphisms, we stumbled across the Leech square-free morphism in [1]. The following is the Leech square-free morphism
which originally appeared in [6]. This lead us to the definition of the morphism with unstackable image words. Note that the definition depends upon a combinatorial property and is not entirely constructive. We have yet to overcome this problem.

2. Preliminaries

We will use the standard definitions from the Lothaire book on combinatorics on words for our definitions with a few additions [7].

We begin by defining an alphabet $\Sigma$ to be a finite set of symbols from which we will make words by concatenation (note, we will use capital Greek letters for alphabets). Further, we define a word $W$ to be a list of symbols from any alphabet $\Sigma$ written horizontally (we will use capital letters to denote words and lower case letters to denote letters). We will denote the word with no letters, that is the empty word, by $\varepsilon$.

2.1. Words. The length (or number of letters) for a word $W$ will be written $|W|$. Note that we will use the same symbol to represent the size of a set or absolute value. The difference will be clear based on context. Notice that $|\varepsilon| = 0$. Further we will represent $|W|_a$ to represent the number of times the letter $a$ occurs in $W$. Also we will use $|W|_{aba}$ to represent the number of times the word $aba$ occurs in $W$. For example if $C = aboaababa$, then we have $|C|_{aba} = 3$ along with $|C| = 8$.

A word $U$ is a factor of a word $V$ if there exist two (possibly empty) words $S$ and $T$ such that $V = TUS$. We will also say that $U$ is a subword of $V$ (or $V$ contains $U$). If $T = \varepsilon$, then we call $U$ the prefix of $V$. Similarly, if $S = \varepsilon$, then we call $U$ the suffix of $V$.

For some alphabet $\Sigma$, $\Sigma^*$ is the Kleene closure of our alphabet. That is, $\Sigma^*$ is all of the possible words over the alphabet $\Sigma$. Notice that $\Sigma^*$ is the free monoid over the set $\Sigma$.

2.2. Morphisms. A morphism $h$ is a mapping from $\Sigma^*$ into $\Delta^*$, where $\Sigma$ and $\Delta$ are alphabets, such that $h(WV) = h(W)h(V)$ for all words $W, V \in \Sigma^*$, and $h(\varepsilon) = \varepsilon$. Note that $W$ and $V$ could potentially be single letters. We note that if $X \subseteq \Sigma$ (X represents a set of words) for some alphabet $\Sigma$, $h(X)$ represents the set of words $\{h(W) : W \in X\}$. Further, we call $h$ non-erasing if for all $a \in \Sigma$, where $\Sigma$ is an alphabet, $h(a) \neq \varepsilon$.

Recall from earlier that the Thue-Morse morphism, $\mu$ defined as

$$
\mu(t) = \begin{cases} 
01, & \text{for } t = 0 \\
10, & \text{for } t = 1, 
\end{cases}
$$

is a morphism defined on the alphabet with two letters. For convenience, we will call the alphabet with $n$ letters $\Sigma_n$. Infinite words are possible with such a morphism. We have displayed the $n^{th}$ Thue-Morse word as being $\mu^n(0)$. We will use $\omega$ to represent the first infinite ordinal. So the Thue-Morse infinite word becomes

$$
T = \lim_{n \to \infty} \mu^n(0) = \mu^\omega(0),
$$

as previously seen. Note that we will use bold capital letters to represent infinite words, with $T$ here representing the Thue-Morse infinite word.

When discussing $\Sigma_2 = \{0, 1\}$, the two letter alphabet we will use $\bar{0}$ to denote the complement of 0 (or 1 if we need that complement). That is $\bar{0} = 1$ and $\bar{1} = 0$. This will become necessary in Chapter 2.

For some morphism $h : \Sigma^* \to \Delta^*$, we will call $h$ uniform if $|h(a)| = n$ for some integer $n$ for all $a \in \Sigma$ (more exactly, in this case we will call $h$ $n$-uniform). We will call a morphism $h : \Sigma^* \to \Delta^*$ square-free when $h(W)$ is square-free if and only if $X \in \Sigma^*$ is square-free. Similarly, we will call $h : \Sigma^* \to \Delta^*$ an overlap-free morphism when $h(W)$ is overlap-free if an only if $W \in \Sigma^*$ is overlap-free.
3. The Morphism With Unstackable Image Words

In a vain attempt to classify all of the overlap-free morphisms using the latin square morphism [10], we stumbled across the Leech square-free morphism in [1]. The following is the Leech square-free morphism

\[ h(t) = \begin{cases} 
0121021201210 & \text{for } t = 0 \\
1202102012021 & \text{for } t = 1 \\
2012010210120 & \text{for } t = 2,
\end{cases} \]

which originally appeared in [6]. Noticing that this morphism was overlap-free put a hole in our attempt to classify all of the overlap-free morphisms using Latin square morphisms. But on the other hand, we now could potentially find another class of overlap-free morphisms that could be explained in a better manner than with test-sets as in [8].

Using the test-set result given by Richomme and Wlazinski, we found the following overlap-free morphisms on four letters

\[ f(t) = \begin{cases} 
0123120103213210 & \text{for } t = 0 \\
123023012303210 & \text{for } t = 1 \\
2301301231031032 & \text{for } t = 2 \\
30120123032102103 & \text{for } t = 3,
\end{cases} \]

and

\[ g(t) = \begin{cases} 
0123012212103210 & \text{for } t = 0 \\
123013003033010321 & \text{for } t = 1 \\
230120123310221032 & \text{for } t = 2 \\
301230110100132103 & \text{for } t = 3.
\end{cases} \]

The morphism \( g \) raised a considerable number of questions as to why it was overlap-free. It seemed to avoid a considerable number of the techniques used in the proof for the Latin square morphisms. So the natural question was: what does the morphism \( g \) have in common with the Leech square-free morphism that causes its overlap-freeness.

4. Definitions and Theorems

The overlap-free morphisms displayed above are tied together with the following definition.

**Definition 4.1.** Let \( h : \Sigma^* \to \Delta^* \) be an \( n \)-uniform morphism. We say that \( h \) is a morphism with unstackable image words if it satisfies the following properties:

(i) \( h(W) \) is overlap-free for all overlap-free words \( W \in \Sigma^* \) with \( |W| = 3 \).

(ii) For \( a, b \in \Sigma \), and for all \( V \in \Sigma^* \) such that \( |V| \leq \lfloor n/2 \rfloor \),

\[ h(a) = SV \quad \text{and} \quad h(b) = VU \]

if and only if \( S \) is not a suffix of any image word of \( h \) and \( U \) is not a prefix of any image word of \( h \).

We now prove a lemma that captures the combinatorial properties in the first portion of Definition 4.1.

**Lemma 4.2.** Let \( \Sigma \) be an alphabet with more than one letter. Let \( h : \Sigma^* \to \Delta^* \) be a morphism such that \( h(W) \) is overlap-free for all overlap-free \( W \in \Sigma^* \) with \( |W| = 3 \). We then have the following properties:

(i) \( h(a) \) is overlap-free for all \( a \in \Sigma \).

(ii) \( h(a)h(b) \) is overlap-free for all \( a, b \in \Sigma \).

(iii) \( h(a) \) and \( h(b) \) do not begin or end with the same letter, whenever \( a, b \in \Sigma \) and \( a \neq b \).

**Proof.** (i) Let us first state that the result does not apply when \( |\Sigma| = 1 \) because there are no overlap-free word of length three for this alphabet. For \( |\Sigma| > 1 \) this result is clear because if we assume for a contradiction that \( h(a) \) contained an overlap for any \( a \in \Sigma \), then \( h(bab) \), with \( b \neq a \), would contain an overlap which contradicts our assumption.
(ii) Similar to (i), if we assume that \( h(a)h(b) = h(ab) \) contained an overlap, then \( h(aba) \) would contain an overlap. Again this contradicts our assumption. We also must show that \( h(aa) \) does not contain an overlap. Assume for a contradiction that it does, and we quickly obtain our contradiction by observing that then \( h(ab) \) must contain an overlap.

(iii) Assume that for some \( a, b \in \Sigma \), \( h(a) \) and \( h(b) \) begin with the same letter. Then, \( h(aab) \) would contain an overlap, which contradicts our assumption. The argument for \( h(b) \) and \( h(b) \) ending with different letters is similar.

**Theorem 4.3.** Any morphism with unstackable image words is overlap-free.

**Proof.** We begin by assuming that \( h \) is a morphism with unstackable image words with \( |h(a)| = n \) for all \( a \in \Sigma \). We must show that for all \( W \in \Sigma^* \), \( W \) is overlap-free if and only if \( h(W) \) is overlap-free. We will begin with the easy direction first.

4.1. The \( \iff \) direction. Assume that \( W = AcXcXCB \), so that we can argue by contrapositive that \( h(W) \) must also contain an overlap. Notice that

\[
h(W) = h(A)h(c)h(X)h(c)h(X)h(c)h(B).
\]

Set \( h(c) = dY \) where \( d \in \Sigma \) and \( Y \in \Sigma^* \), then \( h(W) = h(A)dYh(X)dYh(x)dYh(B) \). So then \( h(W) \) contains the overlap \( dYh(x)dYh(X)d\), and we are done with the first portion of our argument.

4.2. The \( \Rightarrow \) direction. Conversely we will argue by contrapositive. We will assume that \( h(W) \) contains an overlap and show that \( W \) must also contain an overlap. So assume that for some \( W \in \Sigma^* \) we have

\[
h(W) = Ac_{j_0}Xc_{j_1}Xc_{j_2}B,
\]

where \( c = c_{j_0} = c_{j_1} = c_{j_2} \). We use the 0, 1 and 2 to denote which \( c \) we will refer to. Further, the index \( j_i \) will refer to which letter in the word \( h(W) \) we are referring to, noting that we are indexing beginning with 0.

We will proceed with two separate arguments. The first argument will be that it is not possible to write \( h(W) \) with \( |cX| \neq 0 \mod n \). The second argument will be that \( W \) must contain an overlap if \( |cX| \equiv 0 \mod n \).

4.2.1. The \( |cX| \neq 0 \mod n \) case. Notice that we must have the overlap in \( h(W) \) contained in \( h(Z) \) where \( |Z| > 3 \) is some subword of \( W \). Otherwise we would be breaking hypothesis (i) in the definition of Pooh morphisms.

We begin by setting

\[
r_i \equiv j_i \mod n,
\]

where \( i \in \{0, 1, 2\} \) and \( r_i \in \{0, 1, \ldots, n-1\} \). We will argue first based on the number of tiles that the overlap occurs in, and then by cases. When the overlap occurs over four tiles (noting that occurring over three tiles contradicts the hypothesis), we will observe four cases. The cases are

\[
\begin{align*}
r_0 &\leq r_2 < r_1, \\
r_2 &< r_0 < r_1, \\
r_1 &< r_0 \leq r_2, \\
r_1 &< r_2 < r_0.
\end{align*}
\]

We note that the cases \( r_0 < r_1 < r_2 \) and \( r_2 < r_1 < r_0 \) force the overlap to occur in a number other than four tiles. When the overlap occurs in more than four tiles we will more simply consider the two cases \( r_0 < r_1 \) and \( r_1 < r_0 \). Finally, we note the following relationship between \( r_0, r_1, \) and \( r_2 \).

\[
(1) \quad r_2 \equiv 2r_1 - r_0 \mod n.
\]

Consider the notion of the tiling of a line segment. We will use this notion of tiling in application to working with \( h(W) \). The tiles we speak of are the image words of \( h \). Note that all the image words must be of the same length \( n \), this is crucial to our argument. For ease we will use \( T_{xi} \) with \( i \in \{0, 1, 2\} \) to denote
the tile containing \(c_{j_i}\). Note that \(s_i\) is the number of the tile if we numbered them starting with the first tile as \(T_0\).

The overlap is contained in 4 tiles. Let us consider the case where there is some subword of \(W\), say \(Z\), with \(|Z| = 4\) and the overlap in \(h(W)\) is contained in \(h(Z)\). As in the argument for a Latin square morphism to be overlap-free we will consider the word \(h(Z)\) to be a line. We will draw in small vertical lines to signify the edges of the tiles, and we will draw taller labeled vertical lines to signify the \(c\)'s in the overlap.

\[ \begin{array}{c|c|c|c|c} c_{j_0} & V & U & c_{j_1} & c_{j_2} \\ \hline \hline c_{j_0} & c_{j_1} & V & U & c_{j_2} \end{array} \]

**Figure 1.** The short overlap with \(r_2 < r_0 < r_1\)

In Figure 1, we have taken \(c_{j_0}Xc_{j_1}Xc_{j_2}\) and written it twice aligning \(c_{j_0}Xc_{j_1}\) in the upper line with \(c_{j_1}Xc_{j_2}\) in the lower line for the purpose of equating the terms through the overlap. Figure 1 displays the case when \(r_2 < r_0 < r_1\). We remark here that the case when \(r_2 = r_0 < r_1\) proceeds in the same manner.

Let \(V\) to be the final \(r_1 - r_0\) letters in the tile \(T_{s_0}\) as we have drawn in Figure 1. Similarly we choose \(U\) to be the first \(r_1 - r_2\) letters in \(T_{s_1}\). Now equation (1) gives that in the \(r_2 < r_0 < r_1\) situation we have that \(n - (r_1 - r_0) = r_1 - r_2\). Clearly then we must have \(|U| = r_1 - r_2 \leq \lfloor n/2 \rfloor\) or \(|V| = r_1 - r_0 \leq \lfloor n/2 \rfloor\). In the case when \(|V| \leq \lfloor n/2 \rfloor\) we cannot equate \(U\) with any prefix of a tile which leads to a contradiction. In the other case when \(|U| \leq \lfloor n/2 \rfloor\) we cannot equate \(V\) with any suffix of a tile which leads to a contradiction. So this case is not possible.

We now consider the case where \(r_0 < r_2 < r_1\) as shown in Figure 2.

\[ \begin{array}{c|c|c|c|c} c_{j_0} & V & U & c_{j_1} & c_{j_2} \\ \hline \hline c_{j_0} & c_{j_1} & V & U & c_{j_2} \end{array} \]

**Figure 2.** The short overlap with \(r_0 < r_2 < r_1\)

In this case we choose \(V\) to be the final \(r_1 - r_0\) letters in \(T_{s_0}\), and we also pick \(U\) to be the first \(r_1 - r_2\) letters in \(T_{s_1}\) as drawn in Figure 2. Again we notice that \(n - (r_1 - r_0) = r_1 - r_2\) so either \(|V| \leq \lfloor n/2 \rfloor\) or \(|U| \leq \lfloor n/2 \rfloor\), either of which is impossible. So we cannot have this case occurring either.

We now consider the cases with \(r_1 < r_2 \leq r_0\) and \(r_1 < r_0 < r_2\). Figure 3 gives the situation when \(r_1 < r_2 < r_0\) (note that the case when \(r_1 < r_2 = r_0\) is similar, and the same applies to the arguments above). Notice that in both of these cases we have that \(n - (r_2 - r_1) = r_0 - r_1\).

For the case depicted in Figure 3, \(r_1 < r_2 < r_0\), we assume that \(V\) is the final \(r_0 - r_1\) letters in \(T_{s_1}\), and we also assume that \(U\) is the first \(r_2 - r_1\) letters in \(T_{s_2}\). Now either \(|U| \leq \lfloor n/2 \rfloor\) or \(|V| \leq \lfloor n/2 \rfloor\). In either case we have a contradiction.

Now we consider the case where \(r_1 < r_0 < r_2\), which is displayed in Figure 4.

In the case displayed here in Figure 4 we again assume that \(V\) occurs in the final \(r_0 - r_1\) letters of \(T_{s_1}\), and we also assume that \(U\) occurs in the first \(r_2 - r_1\) letters of \(T_{s_2}\). We then have that either \(|V| \leq \lfloor n/2 \rfloor\) or \(|U| \leq \lfloor n/2 \rfloor\).
or that $|U| \leq \lfloor n/2 \rfloor$. Either case is a contradiction. So we cannot have our overlap occurring in four tiles. Thus, we consider the case when the overlap occurs in more than four tiles.

We also note that if we are in the case when the overlap occurs in five tiles, the same arguments hold.

**The overlap is contained in more than four tiles.** We will look at the cases with $r_1 < r_0 < r_2$, and we will only look at the beginning of the overlap. So we consider Figure 4 for the case when $r_0 < r_1$. We will consider the case with $r_0 < r_1$ and $r_1 - r_0 \geq \lfloor n/2 \rfloor$, as we will cover the logic behind the argument for $r_1 - r_0 \leq \lfloor n/2 \rfloor$ in Figure 5.

Let $V$ be the final $r_1 - r_0$ letters in $T_{s_0}$, then equating yields $V$ as the beginning $r_0 - r_1$ letters of $T_{s_1+1}$. Since $|V| \geq \lfloor n/2 \rfloor$ we can equate the suffix of $T_{s_1+1}$, call it $U$ (which is labeled with a dotted line in Figure 5), with $T_{s_0+1}$. So we have $T_{s_1+1} = VU$. Similarly we can set $S \in \Delta^*$ such that $T_{s_0+1} = US$. Notice now that $|U| = n - (r_1 - r_0) \leq \lfloor n/2 \rfloor$. Thus $S$ cannot begin any image word of $h$ so the overlap is impossible.

A note for the case when $r_1 - r_0 \leq \lfloor n/2 \rfloor$. In this case $|V| \leq \lfloor n/2 \rfloor$ and we would not be able to equate $U$.

Figure 6 gives the case when $r_1 < r_0$ with $r_0 - r_1 \leq \lfloor n/2 \rfloor$. In a similar manner to the case where $r_0 < r_1$ we pick $V$ to be the suffix of $r_0 - r_1$ letters in $T_{s_1}$. Now we can possibly find an image word $T_{s_0+1} = VU$ for some $U \in \Delta^*$. But because $|V| = r_0 - r_1 \leq \lfloor n/2 \rfloor$, $U$ cannot be the prefix of any image word, so this formulation of the overlap is impossible.
4.2.2. The $|cX| \equiv 0 \pmod{n}$ case. From Lemma 4.2 we know that the beginning letters and ending letter for each image word in $h$ must be distinct. Further we know that the suffix of $T_{s_1}$ must be identical to the suffix of $T_{s_0}$ as $r_0 = r_1$. This implies that $T_{s_0} = T_{s_1}$. Similarly $T_{s_1} = T_{s_2}$.

Pick $d$ to be the letter such that $h(d) = T_{s_0} = T_{s_1} = T_{s_2}$. Further because $h(W) = Ac_{j_0}Xc_{j_1}Xc_{j_2}XB$, we can find subwords $C, D, Y$ of $W$ so that $h(W) = h(CdYdYdD) = AcXcXB$.

Now we must have that $W = CdYdYdD$ which contains an overlap. Thus we are done.

5. The Square-Free Adaptation

Similarly to the definition of the overlap-free morphisms with unstackable image words we can define square-free morphisms with unstackable image words in the following manner.

**Definition 5.1.** Let $h : \Sigma^* \to \Delta^*$ be an $n$-uniform morphism. We call $h$ a square-free morphism with unstackable image words if it satisfies the following properties:

(i) $h(W)$ is square-free for all square-free words $W \in \Sigma^*$ with $|W| = 3$

(ii) $h(a)$ and $h(b)$ do not begin or end with the same letter for all $a, b \in \Sigma$ with $a \neq b$.

(iii) For $a, b \in \Sigma$, and for all $V \in \Sigma^*$ such that $|V| \leq \lfloor n/2 \rfloor$,

$$h(a) = SV \quad \text{and} \quad h(b) = VU$$

if and only if $S$ is not a suffix of any image word of $h$ and $U$ is not a prefix of any image word of $h$.

Because we cannot consider words like $aab$ to put into $h$, we must add property (ii) in Definition 5.1 so that we can use a similar preimage argument in the final portion of the argument. Thus we have the following theorem.

**Theorem 5.2.** Any square-free morphism with unstackable image words is square-free.

**Proof.** Assume that $h$ is a square-free morphism with unstackable image words such that $|h(a)| = n$ for all $a \in \Sigma$. We must show that for some $W \in \Sigma^+$, $W$ is square-free if and only if $h(W)$ is square-free. We will begin with the easy direction.

5.1. **The $\Leftarrow$ direction.** We will proceed by contrapositive. So assume that $W = AX XB$, where $X \in \Sigma^+$ and $A, B \in \Sigma^*$. Write

$$h(W) = h(AX XB) = h(A)h(X)h(X)h(B),$$

which contains the square $h(X)h(X)$. So we are done with this direction.
5.2. The ⇒ direction. Again we proceed by arguing the contrapositive. So we assume that

$$h(W) = Ac_{j_0}Xd_{i_0}c_{j_1}Xd_{i_1}B,$$

where $c = c_{j_0} = c_{j_1} \in \Sigma$, $d = d_{i_0} = d_{i_1} \in \Sigma$ and $A, X, B \in \Sigma^*$. Note that we are using $c_{j_0}$ and $c_{j_1}$ so that we can mark the beginning of the square, and similarly for the $d$'s and the end of the square.

There are two cases to consider here $|cXd| \not\equiv 0 \pmod{n}$ and $|cXd| \equiv 0 \pmod{n}$. We show that it is impossible for $|cXd| \not\equiv 0 \pmod{n}$ in an analogous manner as in Theorem 4.3, as seen in section 5.1.2.1.

So assume that $|cXd| \equiv 0 \pmod{n}$. From the definition of the pooh square-free morphism we know that each image word for $h$ must begin and end with distinct letters. Further we know the suffix of the tile containing $c_{j_0}$ must be identical to the suffix of the tile containing $c_{j_1}$. Thus they are the same image word, call it $h(z)$ for some $z \in \Sigma$. So because

$$h(W) = AcXcXB,$$

we can find subwords $C, D, Y$ of $W$ such that

$$h(W) = h(CzYzYD) = AcXcXB.$$

Thus we have that $W = AzXzXB$ which contains a square, and we are done. □

6. Acknowledgments

I especially would like to thank Dr. George F. McNulty, my thesis advisor, for all of his insights on the ideas presented here.

References

1. Jean-Paul Allouche and Jeffrey Shallit. *Automatic sequences*. Cambridge University Press, Cambridge, 2003. Theory, applications, and generalizations.
2. Jean Berstel and Patrice Séébold. A characterization of overlap-free morphisms. *Discrete Appl. Math.*, 46(3):275–281, 1993.
3. Max Crochemore. Sharp characterizations of squarefree morphisms. *Theor. Comput. Sci.*, 18(2):221–226, 1982.
4. Andrzej Ehrenfeucht and Grzegorz Rozenberg. On finite sets testing square free property of all homomorphisms between two given alphabets. Technical Report 230, Department of Computer Science, University of Colorado at Boulder, 1982.
5. Anna Frid. Overlap-free symmetric d0l words. *Discrete Math. Theor. Comput. Sci.*, 4(2):357–362, 2001.
6. John Leech. A problem on strings of beads. *Math. Gazette*, 41:277–278, 1957.
7. M. Lothaire. *Algebraic combinatorics on words*, volume 90 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2002.
8. Gwénaël Richomme and Francis Wlazinski. Overlap-free morphism and finite test-sets. *Discrete Appl. Math.*, 143(1-3):92–109, 2004.
9. Axel Thue. Über die gegenseitige lage gleicher teile gewisser zeichenreihen. *Norske vid. Selsk. Skr. Mat. Nat. Kl.*, (1):1–67, 1912.
10. C. Robinson Tompkins. Latin square thue-morse sequences are overlap-free. *Discrete Math. Theor. Comput. Sci.*, 9(1):239–246, 2007.