5-2012

Preferential Arrangement Containment in Strict Superpatterns

Martha Louise Liendo
East Tennessee State University

Preferential Arrangement Containment in Strict Superpatterns

A thesis

presented to

the faculty of the Department of Mathematics

East Tennessee State University

In partial fulfillment

of the requirements for the degree

Master of Science in Mathematical Sciences

---

by

Martha Liendo

May 2012

Anant Godbloe, Ph.D., Chair

Robert A. Beeler, Ph.D.

Robert B. Gardner, Ph.D.

Keywords: pattern containment, preferential arrangements, superpatterns

ABSTRACT

Preferential Arrangement Containment in Strict Superpatterns

by

Martha Liendo

Most results on pattern containment deal more directly with pattern avoidance, or the enumeration and characterization of strings which avoid a given set of patterns. Little research has been conducted regarding the word size required for a word to contain all patterns of a given set of patterns. The set of patterns for which containment is sought in this thesis is the set of preferential arrangements of a given length. The term preferential arrangement denotes strings of characters in which repeated characters are allowed, but not necessary. Cardinalities for sets of all preferential arrangements of given lengths and alphabet sizes are found, as well as cardinalities for sets where reversals fall into the same equivalence class and for sets in higher dimensions. The minimum word length and the word length necessary for a strict superpattern to contain all preferential arrangements for alphabet sizes two and three are also detailed.

DEDICATION

Dedicated to my husband, Patrick Liendo, and my father, Rod McMurray, the two pillars of strength in my life, the men who make me the woman I am.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Anant Godbole, and my committee members, Dr. Robert A. Beeler and Dr. Robert B. Gardner, for all their support, guidance, and encouragement during this chapter of my life. I could not have succeeded otherwise.

CONTENTS

LIST OF TABLES

1 INTRODUCTION

A word, or string, is said to contain a pattern if any *order-isomorphic* subsequence to that pattern can be found within that word. Order-isomorphic subsequences are subsequences which contain equivalent ranking structures. For example, the word 5371473 contains the subsequences 571, 574, and 473, each of which is order-isomorphic to the string 231. We call the string 231 the pattern that is contained in the word since it is comprised of the lowest possible ordinal numbers. In many cases, the term *pattern* is reserved for strings of characters in which each character is unique, as in [7, 10, 13]. This traditional definition of pattern is adhered to in this thesis, while the term *preferential arrangement* denotes those strings of characters in which repeated characters are allowed, but not necessary. The example word above also contains the subsequences 373 and 343 which are both order-isomorphic to the string 121. Both the string 121 and the string 231 are unique preferential arrangements contained in the word since they are comprised of the lowest possible ordinal numbers that describe unique sets of order-isomorphic strings contained in the word. This order-isomorphism on the preferential arrangements is equivalent to a *dense ranking system*, where items that compare equal with respect to an ascending ranking order receive the same ranking number, and the next item(s) receive the immediately following ranking number.

The concept of a systematic study of *pattern containment*, i.e., words which contain all patterns of a given set of patterns, was first proposed by Herb Wilf in his 1992 address to the SIAM meeting on Discrete Mathematics [4]. However, most results on pattern containment deal more directly with *pattern avoidance*, or the enumeration and characterization of strings which avoid a given pattern or set of patterns. Of the few results available on pattern containment, most deal with specified sets of patterns contained in fixed length permutations, i.e., strings without repeated letters, such as in [2, 10, 13, 7]. Research in this area mainly includes enumerating occurrences of a given set of patterns, which may only include one pattern, contained in a permutation of fixed length. Burstein et al. [4, 5] have expanded this research further by not only allowing repeated letters in the word that is to contain the set of patterns, but also allowing repeated letters within the contained patterns themselves.

Very little research has been conducted regarding the word size required for a word to contain all patterns or preferential arrangements of a given length. The only exception seems to be the idea of a superpattern. A *superpattern* is a word which contains all patterns of a given set of patterns. Eriksson et al.[6], Albert et al.[2], and Miller [10] have all found bounds on the word length of superpatterns dealing with permutations and pattern containment. Burstein et al. in [4] studied bounds on shortest superpatterns without the permutation requirement under preferential arrangement containment. In the second section of this thesis, cardinalities for sets of all preferential arrangements of given lengths and alphabet sizes are found. In addition, cardinalities for sets where reversals fall into the same equivalence class of preferential arrangements and for sets in higher dimensions are given. The expected word length necessary for a superpattern to contain all preferential arrangements for alphabet sizes two and three is detailed in the third section.

2 CARDINALITIES OF PREFERENTIAL ARRANGEMENTS

A *word* is a string of characters in which repeated occurrences of a character are allowed, but not necessary.

Let $[k] = \{1, 2, \ldots, k\}$ be a totally ordered alphabet of k letters, meaning that for all $i, j \in [k]$ either $i < j$ or $j < i$. Let $[k]^n$ denote the set of all words of length n over this alphabet. For example, $[3]^2 = \{11, 22, 33, 12, 13, 21, 23, 32, 31\}$.

The words, $\pi = \pi(1), \pi(2), \ldots, \pi(n) \in [k]^n$ and $\pi' = \pi'(1), \pi'(2), \ldots, \pi'(n) \in [k]^n$ are *order-isomorphic* if for all $i, j \in [n]$, $\pi(i) \leq \pi(j)$ if and only if $\pi'(i) \leq \pi'(j)$. In the above example, the words 12, 13, and 23 are order-isomorphic. This order-isomorphism partitions $[k]^n$ into a set of equivalence classes, where the equivalence class representative for each equivalence class is denoted here as π, such that $[\pi] = \{\pi' \in [k]^n : \pi' \textit{ is order-isomorphic to } \pi, \pi(i) \leq \pi'(i) \textit{ for all } i \in [n]\}$. In other words, π is the word with the lowest possible ordinal numbers in the set of words which are all order-isomorphic to π and contained in $[k]^n$. In the example of order-isomorphic words given above, π is the word 12. The word π is called a *preferential arrangement.*

Preferential arrangements can also exist in higher dimensions, where the preferential arrangement in dimension d is an $n_1 \times n_2 \times n_3 \times \cdots \times n_d$ array. Of particular interest is the set of all preferential arrangements in dimension d of length n with $k \leq a$ ranks allowed, denoted as $\Pi_d(n, a)$. Dimension one is shown first.

Proposition 2.1 *For all values of $n \geq 1$ and $a \geq 1$, $\Pi_1(n, a) = \sum_{k=1}^{a} k! S(n, k)$, where $S(n, k) = \sum_{i=0}^{k} (-1)^i \binom{k}{i} (k-i)^n$ are Stirling numbers of the second kind.*

Proof. Let $[n] = \{1, 2, \ldots, n\}$ be the length of a word and $[k] = \{1, 2, \ldots, k\}$ be the canonical totally ordered alphabet of k letters. Consider the partition, P, of $[k]^n$ into equivalence classes under order-isomorphism, with $P = \{[\pi] : \pi \in [k]^n\}$. Define ϕ as the order-isomorphism that maps each $i \in [n]$ to $\pi(i) \in [k]$. Since each $\pi(i)$ is unique for each $i \in [n]$, there exists at most n unique values for $\pi(i)$, for all $\pi(i) \leq n$. Therefore, only values for $k \leq n$ are used in ϕ, making $\phi : [n] \to [k]$ a surjective function. This single mapping corresponds to a unique preferential arrangement of length n over a k-letter alphabet. The total number of all such surjective functions $\phi : [n] \to [k]$ is $k!S(n,k)$, where $S(n,k)$ denotes Stirling numbers of the second kind [3]. This corresponds to the count of all preferential arrangements of length n over a k-letter alphabet. Therefore, the total number of preferential arrangements of length n with $k \leq a$ ranks allowed is $\sum_{k=1}^{a} k!S(n,k)$. ■

Table 1: Table of values for $\Pi_1(n, a)$

$a \setminus n$	1	2	3	4	5	6	7	8	$\cdots$
1	1	1	1	1	1	1	1	1	$\cdots$
2	1	3	7	15	31	63	127	255	$\cdots$
3	1	3	13	51	181	603	1933	6051	$\cdots$
4	1	3	13	75	421	2163	10333	47875	$\cdots$
5	1	3	13	75	541	3963	27133	172875	$\cdots$
6	1	3	13	75	541	4683	42253	364395	$\cdots$
7	1	3	13	75	541	4683	47293	505515	$\cdots$
8	1	3	13	75	541	4683	47293	545835	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

This table of values exists as entry number $A000670$ on the On-line Encyclopedia for Integer Sequences [11].

The following result ties the number of preferential arrangements in higher dimensions to the number of preferential arrangements in one dimension.

Proposition 2.2 *For all values of* $n \geq 1$, $a \geq 1$, *and* $d \geq 1$, $\Pi_d(n, a) = \Pi_1(n^d, a)$.

Proof. The proof proceeds by induction on d.

If $d = 1$, then $\Pi_1(n, a) = \Pi_1(n^1, a)$ by definition, so the theorem holds true for $d = 1$.

Let $d = 2$. In two dimensions, each value of n represents an $n \times n$ matrix, causing there to exist n^2 labeled elements. List each labeled element in the following manner: $n_{11}, n_{12}, \ldots, n_{1n}, n_{2n}, n_{2(n-1)}, \ldots, n_{21}, n_{31}, n_{32}, \ldots, n_{3n}, \ldots, n_{nn}$. This listing corresponds to a unique preferential arrangement of the n^2 labeled elements in dimension one that represents a unique preferential arrangement of the same labeled elements in dimension two. Therefore, $\Pi_2(n, a) = \Pi_1(n^2, a)$ and the theorem holds true for $d = 2$.

Assume the theorem holds true for some value of $d > 2$. Thus, $\Pi_d(n, a) = \Pi_1(n^d, a)$.

In $d+1$ dimensions, each value of n represents an $n_1 \times n_2 \times n_3 \times \cdots \times n_{d+1}$ array, causing there to exist $n^{d+1} = n \times n^d$ elements. Following the listing technique used in dimension two, arrange the lists of each of the preferential arrangements represented by the n^d elements such that the last element of one n^d list is adjacent to the first element of another n^d list in the $d+1$ dimension. One such arrangement of all the n^d lists in this manner corresponds to a unique preferential arrangement of the n^{d+1} labeled elements in dimension n^d that represents a unique preferential arrangement of the same labeled elements in dimension n^{d+1}. This arrangement consists of n lists of n^d preferential arrangements in dimension n^d. Therefore, $\Pi_{d+1}(n,a) = \Pi_1(n \times n^d, a) = \Pi_1(n^{d+1}, a)$ by the induction hypothesis. ■

The reversal of a word $\pi \in [k]^n$, commonly denoted by $r(\pi)$, is obtained by writing the labeled elements of π in the reverse order, that is, the i-th labeled element of $r(\pi)$ is equal to the $(n-i+1)$-th labeled element of π [7]. For example, $r(1323) = 3231$ and $r(12321) = 12321$. One may wish to consider how setting the reversals of preferential arrangements equal may affect the total number of preferential arrangements. Define $\Pi'_1(n,a)$ as the number of preferential arrangements in dimension one of n labeled elements with $k \leq a$ ranks allowed and reversals equal. Clearly, some preferential arrangements are self-reversals, or palindromic. The amount of this type of preferential arrangement must be considered when determining the total count of $\Pi'_1(n,a)$.

Lemma 2.3 *The number of palindromic preferential arrangements in dimension one on n labeled elements with $k \leq a$ ranks allowed is $\Pi_1(\lceil \frac{n}{2} \rceil, a)$.*

Proof. For any palindromic preferential arrangement, $\pi \in [k]^n$, each i-th labeled element of π is equal to the $(n-i+1)$-th labeled element of π. Thus, the first $\lceil \frac{n}{2} \rceil$ labeled elements determine the last $\lfloor \frac{n}{2} \rfloor$ labeled elements. All preferential arrangements of the first $\lceil \frac{n}{2} \rceil$ labeled elements with $k \leq a$ ranks allowed will therefore give all unique rank-respecting palindromes of n labeled elements with $k \leq a$ ranks allowed. So the number of palindromic preferential arrangements of n labeled elements with $k \leq a$ ranks allowed is $\Pi_1(\lceil \frac{n}{2} \rceil, a)$. ■

Using this Lemma, the algebraic formula for $\Pi'_1(n,a)$ is easily found.

Proposition 2.4 *For all values of* $n \geq 1$ *and* $a \geq 1$, $\Pi'_1(n,a) = \frac{1}{2}[\Pi_1(n,a) + \Pi_1(\lceil \frac{n}{2} \rceil, a)]$.

Proof. The number of preferential arrangements in dimension one of n labeled elements with $k \leq a$ ranks allowed and reversals equal is the total number of non-palindromic preferential arrangements with reversals equal plus the total number of palindromic preferential arrangements. Since each unique non-palindromic preferential arrangement will equal its unique reversal, the total number of non-palindromic preferential arrangements with reversals equal will be half the total number of non-palindromic preferential arrangements. This is the total number of preferential arrangements minus the total number of palindromic preferential arrangements. Thus,

$$\Pi'_1(n,a) = \frac{1}{2}[\Pi_1(n,a) - \Pi_1(\lceil \frac{n}{2} \rceil, a)] + \Pi_1(\lceil \frac{n}{2} \rceil, a) = \frac{1}{2}[\Pi_1(n,a) + \Pi_1(\lceil \frac{n}{2} \rceil, a)].$$

■

Table 2: Table of values for $\Pi'_1(n,a)$

$a \setminus n$	1	2	3	4	5	6	7	8	$\cdots$
1	1	1	1	1	1	1	1	1	$\cdots$
2	1	2	5	9	19	35	71	135	$\cdots$
3	1	2	8	27	97	308	992	3051	$\cdots$
4	1	2	8	39	217	1088	5204	23475	$\cdots$
5	1	2	8	39	277	1988	13604	86475	$\cdots$
6	1	2	8	39	277	2348	21164	182235	$\cdots$
7	1	2	8	39	277	2348	23684	252795	$\cdots$
8	1	2	8	39	277	2348	23684	272955	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

3 CONTAINMENT LENGTHS OF STRICT SUPERPATTERNS

The general definition of a *superpattern* is a pattern or word which contains all patterns of a given set of patterns. Superpatterns have been a topic of research under permutation patterns for some time, with examples of such research found in [2, 6, 10]. In all these cases, the set of patterns for which containment is sought are permutation patterns and, in most cases, the superpattern is a permutation pattern as well. Burstein et al.[4] studied superpatterns without the permutation requirement under preferential arrangement containment. The main goal of previous research on superpatterns, however, is to bound the length of the shortest superpattern. This thesis seeks not only minimum values of superpattern length but also the expected superpattern length needed for the containment of all preferential arrangements.

Further characterization of superpatterns is necessary for clarity in this thesis. A *minimal superpattern* is a superpattern in which no two adjacent letters are the same. Thus, a *minimum superpattern* is a minimal superpattern of the shortest length possible in which every letter is necessary for the containment of all preferential arrangements. A *strict superpattern* is a superpattern in which the last letter of the superpattern is needed to complete one of the preferential arrangements contained in the superpattern. Clearly, all minimum superpatterns are strict superpatterns.

Following the notation of Burstein et al.[4], let a superpattern for $[k]^m$ be a word that contains every preferential arrangement of length m on at most k letters. Let $n(k, m)$ be the length of a strict superpattern for $[k]^m$. Since preferential arrangements follow a dense ranking system any word containing them can be reduced to a dense ranking system, making $n(k, m) = n(m, m)$ for $m \leq k$. For this reason, we are only interested in the values of $n(k, m)$ for $m \geq k$, and most specifically for $m = k$.

The minimum length and the expected length of strict superpatterns for the cases of $n(2, 2)$ and $n(3, 3)$ are specifically studied in detail in the following sections.

3.1 Strict Superpatterns for $[2]^2$

In the binary case, a strict superpattern for $[2]^2$ is a word that contains all the preferential arrangements of $[2]^2$, namely 11, 12, and 21, and the last letter of the superpattern is needed to complete one of the preferential arrangements contained in the superpattern. The minimum containment length for superpatterns of this case are trivial and more interest lies in the expected containment length for $n(2, 2)$

3.2 Minimum Length for $n(2, 2)$

The minimum length for the trivial case of $n(2, 2)$ is given as $n(2, 2) = 3$ by Burstein et al. (see [4]). The minimum superpattern 121 is given as an example, since 121 contains all the preferential arrangements of $[2]^2$. The pattern 121 is, in fact, the only minimum superpattern for $[2]^2$ up to isomorphism.

3.3 Expected Length for $n(2, 2)$

For expected length, consider that for any word of length n to contain the patterns 11, 12, and 21 there must exist at least two runs on the first $n-1$ letters of the word. The nth letter of this word must be the letter that correctly completes a minimum superpattern for $[2]^2$ that is contained within the word. The number of ways to partition $n-1$ letters into two parts is $\binom{n-2}{1} = (n-2)$. Since there are a total of two minimum superpatterns for $[2]^2$, namely 121 and 212, there are $2(n-2)$ words of length $n-1$ out of the total 2^{n-1} words that exist on $n-1$ that satisfy this condition. Therefore, the probability that a word on n letters contains all preferential arrangements of $[2]^2$ is

$$p_2(n) = \frac{2(n-2)}{2^{n-1}} \times \frac{1}{2} = \frac{n-2}{2^{n-1}}.$$

By using different forms of the derivative of the geometric series, the expected waiting time W till the sequence becomes a superpattern can be computed as follows,

$$\begin{aligned}
E(W) &= \sum_{n\geq 3} np_2(n) \\
&= \sum_{n\geq 3} \frac{n(n-2)}{2^{n-1}} \\
&= \frac{1}{4}\sum_{n\geq 3} \frac{(n-1)(n-2)}{2^{n-3}} + \frac{1}{4}\sum_{n\geq 3} \frac{n-2}{2^{n-3}} \\
&= \frac{1}{4}\sum_{m\geq 2} \frac{(m)(m-1)}{2^{m-2}} + \frac{1}{4}\sum_{m\geq 1} \frac{m}{2^{n-1}} \\
&= \frac{1}{4} \times \frac{2}{(1-\frac{1}{2})^3} + \frac{1}{4} \times \frac{1}{(1-\frac{1}{2})^2} \\
&= 4+1 \\
&= 5.
\end{aligned}$$

Similarly, the variance is found to be

$$\begin{aligned}
V(W) &= E(W^2) - E(W)^2 \\
&= E(W(W-1)) + E(W) - E(W)^2 \\
&= \sum_{n\geq 3} \frac{n(n-1)(n-2)}{2^{n-3}} + 5 - 25 \\
&= \frac{1}{4}\sum_{n\geq 3} \frac{n(n-1)(n-2)}{2^{n-1}} - 20 \\
&= \frac{1}{4} \times \frac{6}{(1-\frac{1}{2})^4} - 20 \\
&= 24 - 20 \\
&= 4.
\end{aligned}$$

The ordinary generating function is

$$\begin{aligned}
G_2(x) &= \sum_{n\geq 3} \frac{x^n(n-2)}{2^{n-1}} \\
&= \sum_{n\geq 3} \frac{x^{(n-3+3)}(n-2)}{2^{n-3+2}} \\
&= \frac{x^3}{4}\sum_{n\geq 3} \frac{x^{(n-3)}(n-2)}{2^{n-3}} \\
&= \frac{x^3}{4}\sum_{m\geq 1} m\left(\frac{x}{2}\right)^{m-1} \\
&= \frac{x^3}{4} \times \frac{1}{(1-\frac{x}{2})^2} \\
&= \frac{x^3}{2^2} \times \frac{1}{(1-\frac{x}{2})^2} \\
&= \frac{x^3}{(2-x)^2}.
\end{aligned}$$

3.4 Strict Superpatterns for $[3]^3$

A superpattern for $[3]^3$ is a word that contains all 13 preferential arrangements of $[3]^3$, namely 111, 112, 121, 211, 122, 212, 221, 123, 132, 213, 231, 312, and 321. Following the notation of Bóna (see [3]), let $\pi = \pi_1, \pi_2, \ldots, \pi_k$ be a partition of the set $[n]$ where $n = n(3,3)$ and π_i denotes a block of π. Thus, $a = (a_1, a_2, \ldots, a_k)$ is a partition of the integer n where $a_i = |\pi_i|$ and $a_1 \geq a_2 \geq \cdots \geq a_k \geq 1$ and $a_1 + a_2 + \cdots + a_k = n$. For example, if $n = 7$, then one such partition of the integer 7 is $(5, 1, 1)$. It should be noted that for any minimal superpattern no $a_i > \lceil \frac{n}{2} \rceil$ since this would cause adjacent letters to be the same letter. This fact, combined with the following lemma, proves very useful in determining the word length of strict superpatterns for $[3]^3$.

Lemma 3.1 *Any superpattern for $[3]^3$ contains a jk and a kj pattern both before and after at least one i, where $i, j, k \in [3]$ with $i \neq j \neq k$.*

Proof. Let σ be a superpattern for $[3]^3$ and let $i, j, k \in [3]$ with $i \neq j \neq k$.

Case 1: Assume σ does not contain a jk pattern before an i. Thus, σ does not contain the pattern jki and σ is not a superpattern for $[3]^3$. This is a contradiction and therefore σ contains a jk pattern before at least one i.

Case 2: Assume σ does not contain a jk pattern after an i. Thus, σ does not contain the pattern ijk and σ is not a superpattern for $[3]^3$. This is a contradiction and therefore σ contains a jk pattern after at least one i.

Case 3: Assume σ does not contain a kj pattern before an i. Thus, σ does not contain the pattern kji and σ is not a superpattern for $[3]^3$. This is a contradiction and therefore σ contains a kj pattern before at least one i.

Case 4: Assume σ does not contain a kj pattern after an i. Thus, σ does not contain the pattern ikj and σ is not a superpattern for $[3]^3$. This is a contradiction and therefore σ contains a kj pattern after at least one i.

Therefore, σ contains a jk and a kj pattern both before and after at least one i. ■

3.5 Minimum Length for $n(3,3)$

The minimum length for $n(3,3)$ is found through theorems which count the total number of strict minimal superpatterns of length $n = n(3,3)$. Clearly there are no strict minimal superpatterns for $n = 3$ or $n = 4$ since in all partitions of the integers 3 and 4 into three parts there does not exist an $a_i \geq 3$ and therefore no 111 pattern exists.

Lemma 3.2 *There are no strict minimal superpatterns of length $n = 5$.*

Proof. The integer 5 can be partitioned into three parts in two ways, namely $(3,1,1)$ and $(2,2,1)$.

Case 1: Consider a strict minimal superpattern $\sigma = \sigma(1), \sigma(2), \ldots, \sigma(5) \in (3,1,1)$ with $a_i = 3$. Thus, $\sigma(1) = \sigma(3) = \sigma(5) = i$ since no two adjacent letters are the same letter. There cannot exist both a jk and a kj pattern before at least one i, which contradicts σ as a strict minimal superpattern. Therefore, there is no strict minimal superpattern $\sigma \in (3,1,1)$.

Case 2: Consider a strict minimal superpattern $\sigma = \sigma(1), \sigma(2), \ldots, \sigma(5) \in (2,2,1)$. There does not exist an $a_i \geq 3$ and, therefore, no 111 pattern exists. This contradicts

σ as a strict minimal superpattern. Therefore, there is no strict superpattern $\sigma \in (2, 2, 1)$.

Thus, there are no strict minimal superpatterns of length $n = 5$. ■

Lemma 3.3 *There are no strict minimal superpatterns of length $n = 6$.*

Proof. The integer 6 can be partitioned into three parts in three ways, namely $(4, 1, 1)$, $(3, 2, 1)$, and $(2, 2, 2)$.

Case 1: Consider a strict minimal superpattern $\sigma = \sigma(1), \sigma(2), \ldots, \sigma(6) \in (4, 1, 1)$. There exists an $a_i > \lceil \frac{n}{2} \rceil = 3$, causing two adjacent letters to be the same letter. This contradicts σ as a strict minimal superpattern. Therefore, there is no strict minimal superpattern $\sigma \in (4, 1, 1)$.

Case 2: Consider a strict minimal superpattern $\sigma = \sigma(1), \sigma(2), \ldots, \sigma(6) \in (3, 2, 1)$ with $a_i = 1$. Let $\sigma(a) = i$ for some $a \in [6]$. Thus, $a \geq 4$ since there exists both a jk and a kj pattern before i and $a \leq 3$ since there exists both a jk and a kj pattern after i. No such a exists and, therefore, there is no strict minimal superpattern $\sigma \in (3, 2, 1)$.

Case 3: Consider a strict minimal superpattern $\sigma = \sigma(1), \sigma(2), \ldots, \sigma(6) \in (2, 2, 2)$. There does not exist an $a_i \geq 3$ and, therefore, no 111 pattern exists, which contradicts σ as a strict minimal superpattern. Therefore, there is no strict minimal superpattern $\sigma \in (2, 2, 2)$.

Thus, there are no strict minimal superpatterns of length $n = 6$. ■

Lemma 3.4 *There exist seven strict minimal superpatterns of length $n = 7$ up to isomorphism.*

Proof. The integer 7 can be partitioned into three parts in four ways, namely $(5, 1, 1)$, $(4, 2, 1)$, $(3, 3, 1)$, and $(3, 2, 2)$.

Case 1: Consider a strict minimal superpattern $\sigma = \sigma(1), \sigma(2), \ldots, \sigma(7) \in (5, 1, 1)$. Thus, there exists an $a_i > \lceil \frac{n}{2} \rceil = 4$, causing two adjacent letters to be the same letter, which contradicts σ as a strict minimal superpattern. Therefore, there is no strict minimal superpattern $\sigma \in (5, 1, 1)$.

Case 2: Consider a strict minimal superpattern $\sigma = \sigma(1), \sigma(2), \ldots, \sigma(7) \in (4, 2, 1)$ with $a_i = 1$, $a_j = 4$, and $a_k = 2$. Let $\sigma(a) = i$ for some $a \in [7]$. Thus, $a \geq 4$ since there exists both a jk and a kj pattern before i and $a \leq 4$ since there exists both a jk and a kj pattern after i. Therefore, $a = 4$. Without loss of generality let $\sigma(1) = \sigma(3) = j$ and $\sigma(2) = k$. Since no two adjacent letters are the same letter, $\sigma(5) = \sigma(7) = j$ and $\sigma(6) = k$. There exists one such strict minimal superpattern $\sigma \in (4, 2, 1)$ up to isomorphism.

Case 3: Consider a strict minimal superpattern $\sigma = \sigma(1), \sigma(2), \ldots, \sigma(7) \in (3, 3, 1)$ with $a_i = 1$, $a_j = 3$, and $a_k = 3$. Let $\sigma(a) = i$ for some $a \in [7]$. Thus, $a \geq 4$ since there exists both a jk and a kj pattern before i and $a \leq 4$ since there exists both a jk and a kj pattern after i. Therefore, $a = 4$. Without loss of generality let $\sigma(1) = \sigma(3) = j$ and $\sigma(2) = k$. Since no two adjacent letters are the same letter, $\sigma(5) = \sigma(7) = k$ and $\sigma(6) = j$. There exists one such strict minimal superpattern $\sigma \in (3, 3, 1)$ up to isomorphism.

Case 4: Consider a strict minimal superpattern $\sigma = \sigma(1), \sigma(2), \ldots, \sigma(7) \in (3, 2, 2)$

with $a_i = 3$, $a_j = 2$, and $a_k = 2$. Let $\sigma(a) = \sigma(b) = \sigma(c) = i$ for some $a, b, c \in [7]$ with $1 \leq a < b < c \leq 7$. Since no two adjacent letters are the same letter, $3 \leq b \leq 5$.

Case 4.1: If $b = 3$, then $a = 1$ and $c = 5$, 6, or 7 since no two adjacent letters are the same letter.

Case 4.1.1: If $c = 5$, then there does not exist both a jk and a kj pattern before at least one i, which contradicts σ as a strict minimal superpattern. Therefore, $c \neq 5$.

Case 4.1.2: If $c = 6$, then without loss of generality let $\sigma(2) = \sigma(5) = j$ and $\sigma(4) = k$ since there exists both a jk and a kj pattern before at least one i. Thus, $\sigma(7) = k$ since $a_j = a_k = 2$ and there does not exist an ij pattern after at least one k, which contradicts σ as a strict minimal superpattern. Therefore, $c \neq 6$.

Case 4.1.3: If $c = 7$, then without loss of generality let $\sigma(4) = \sigma(6) = j$ and $\sigma(5) = k$ since no two adjacent letters are the same letter. Thus, $\sigma(2) = k$ since $a_j = a_k = 2$ and there does not exist an ik pattern after at least one j, which contradicts σ as a strict minimal superpattern. Therefore, $c \neq 7$.

Therefore, $b \neq 3$.

Case 4.2: If $b = 4$, then $a = 1$ or 2 and $c = 6$ or 7 since no two adjacent letters are the same letter.

Case 4.2.1: If $a = 1$ and $c = 6$, then without loss of generality let $\sigma(2) = j$ and $\sigma(3) = k$ since no two adjacent letters are the same letter. Thus, $\sigma(5) = j$ since both a jk and a kj pattern exist before at least one i and $\sigma(7) = k$ since $a_j = a_k = 2$. There exist one such strict minimal superpattern $\sigma \in (3, 2, 2)$ up to isomorphism.

Case 4.2.2: If $a = 1$ and $c = 7$, then without loss of generality let $\sigma(2) = j$ and $\sigma(3) = k$ since no two adjacent letters are the same letter. If $\sigma(5) = j$ then $\sigma(6) = k$

since $a_j = a_k = 2$ and there exists one such strict minimal superpattern $\sigma \in (3, 2, 2)$ up to isomorphism. If $\sigma(5) = k$ then $\sigma(6) = j$ since $a_j = a_k = 2$ and there exists one such strict minimal superpattern $\sigma \in (3, 2, 2)$ up to isomorphism.

Case 4.2.3: If $a = 2$ and $c = 6$, then without loss of generality let $\sigma(1) = \sigma(5) = j$ and $\sigma(3) = k$ since both a jk and a kj pattern exist before at least one i. Thus, $\sigma(7) = k$ since $a_j = a_k = 2$ and there exists one such strict minimal superpattern $\sigma \in (3, 2, 2)$ up to isomorphism.

Case 4.2.4: If $a = 2$ and $c = 7$, then without loss of generality let $\sigma(3) = \sigma(6) = j$ and $\sigma(5) = k$ since both a jk and a kj pattern exist after at least one i. Thus, $\sigma(1) = k$ since $a_j = a_k = 2$ and there exists one such strict minimal superpattern $\sigma \in (3, 2, 2)$ up to isomorphism.

Case 4.3: If $b = 5$, then $a = 1$, 2, or 3 and $c = 7$ since no two adjacent letters are the same letter.

Case 4.3.1: If $a = 1$, then without loss of generality let $\sigma(2) = \sigma(4) = j$ and $\sigma(3) = k$ since no two adjacent letters are the same letter. Thus, $\sigma(6) = k$ since $a_j = a_k = 2$ and there does not exist an ij pattern after at least one k, which contradicts σ as a strict minimal superpattern. Therefore, $a \neq 1$.

Case 4.3.2: If $a = 2$, then without loss of generality let $\sigma(3) = \sigma(6) = j$ and $\sigma(4) = k$ since there exists both a jk and a kj pattern after at least one i. Thus, $\sigma(1) = k$ since $a_j = a_k = 2$ and there does not exist an ik pattern after at least one j, which contradicts σ as a strict minimal superpattern. Therefore, $a \neq 2$.

Case 4.3.3: If $a = 3$, then there does not exist both a jk and a kj pattern after at least one i, which contradicts σ as a strict minimal superpattern. Therefore, $a \neq 3$.

Therefore, $b \neq 5$.

Thus, there exist seven strict minimal superpatterns of length $n = 7$ up to isomorphism. ■

Corollary 3.5 *The length of a minimum superpattern for* $[3]^3$ *is* $n(3,3) = 7$.

Proof. There are no strict minimal superpatterns for $n < 3$ since $n < 3$ cannot be partitioned into 3 parts. There are no strict minimal superpatterns for $n = 3$ or $n = 4$ since no partition of 3 or 4 into 3 parts contains an $a_i \geq 3$, and therefore no 111 pattern exists for these cases. By Lemmas 3.2, 3.3, and 3.4 the result follows. ■

Burstein et al. (see [4]) gives a construction proof for $n(\ell, \ell) \leq \ell^2 - 2\ell + 4$ and conjectures that $n(\ell, \ell) = \ell^2 - 2\ell + 4$. The corollary above clearly supports that conjecture for the case of $n(3,3)$.

The seven unique strict minimal superpatterns up to isomorphism of length $n = 7$ are 1213121, 1213212, 1231213, 1231231, 1231321, 1232123, and 1232132. Since the alphabet size is 3, there are 3! ways to permute the letters isomorphically in each strict minimal superpattern of length $n = 7$, giving a total of $3!(7) = 42$ strict minimal superpatterns of length $n = 7$.

3.6 Expected Length for $n(3,3)$

For expected containment length, consider the total amount of possible minimal superpatterns up to isomorphism of any length n. Since all minimal superpatterns are comprised of an alternating pattern, then, up to isomorphism, the first two letters can be fixed as i and j for $i, j \in [3]$ with $i \neq j$. There exist 2^{n-2} total words on the remaining $n-2$ positions that have alternating patterns since each letter can be chosen from an alphabet size of 2. However, not all of these 2^{n-2} words will result in a superpattern of $[3]^3$ on length n. The following lemma aids in determining the amount of words which fail to create a superpattern of $[3]^3$ of length n.

Lemma 3.6 *Any strict minimal superpattern, σ, for $[3]^3$ of length $n \geq 7$ contains a minimum superpattern for $[3]^3$ with the last letter of the minimum superpattern occurring on the last letter of σ.*

Proof. Consider a strict minimal superpattern $\sigma = \sigma(1), \sigma(2), \ldots, \sigma(n)$ up to isomorphism for $[3]^3$ of length $n \geq 7$. Let $i, j, k \in [3]$ with $i < j < k$. Without loss of generality, let $\sigma(n) = i$ and $\sigma(n-1) = k$ since there are no two adjacent letters the same. Thus, there exists some $\sigma(b_1) = i$ as the first occurrence of i in σ, and, without loss of generality, there exists $\sigma(c_1), \sigma(c_2) = kj$ with $\sigma(c_2) = i$ as the last occurrence of j in σ where $b_1 < c_1 < c_2 < n-1$ since there exists both a jk and a kj pattern after at least one i and, up to isomorphism, it can be assumed that a kjk pattern satisfies this condition. If $b_1 > 3$ then there exists a jk and a kj pattern before it, causing σ to contain either a $jkjikjk$ or a $kjkikjk$ pattern, both of which are strict superpatterns of length $n = 7$ and therefore $\sigma(n) = i$ is unnecessary for the the containment of all preferential arrangements. This contradicts the given fact that σ is a strict minimal superpattern. Therefore, $b_1 \leq 3$.

Case 1: If $b_1 = 3$, then $\sigma(1), \sigma(2) = jk$ or kj. If $\sigma(1), \sigma(2) = jk$, then σ contains the minimum superpattern $jkikjki$ with the last letter of the minimum superpattern occurring on the last letter of σ. If $\sigma(1), \sigma(2) = kj$, then σ contains the minimum superpattern $kjikjki$ with the last letter of the minimum superpattern occurring on the last letter of σ.

Case 2: If $b_1 = 2$, then $\sigma(1) = j$ or k. If $\sigma(1) = j$, then there exists the pattern ki before $\sigma(c_2) = j$ since there exists a ki pattern before at least one j and thus it must also exist before the last j. Thus, σ contains the minimum superpattern $jikijki$ with the last letter of the minimum superpattern occurring on the last letter of σ. If $\sigma(1) = k$, then there exists a ji pattern before $\sigma(n-1) = k$ since there exists a ji pattern before at least one k and $\sigma(n-1)$ is the last occurrence of k. Since no two adjacent letters are the same letter, $\sigma(3) = j$ or k. If $\sigma(3) = j$, then σ contains either a $kijikjk$, a $kijkijk$, or a $kijkjik$ pattern on the first $n-1$ letters, all of which are strict superpatterns of length $n = 7$ and therefore $\sigma(n) = i$ is unnecessary for the the containment of all preferential arrangements. This contradicts the given fact that σ is a strict minimal superpattern. Therefore, $\sigma(3) \neq j$. If $\sigma(3) = k$, then σ contains the minimum superpattern $kikjiki$ with the last letter of the minimum superpattern occurring on the last letter of σ.

Case 3: If $b_1 = 1$, then $\sigma(2) = j$ or k since no two adjacent letters are the same letter. If $\sigma(2) = j$, then there exists a ki pattern before $\sigma(c_2) = j$ since there exists a ki pattern before at least one j and $\sigma(c_2)$ is the last occurrence of j. Therefore, σ contains the minimum superpattern $ijkijki$ with the last letter of the minimum superpattern occurring on the last letter of σ. If $\sigma(2) = k$, then $\sigma(3) = i$ or j. If $\sigma(3) = i$, note that there exists a ji pattern before $\sigma(n-1) = k$ since there exists a ji pattern before at least one k and $\sigma(n-1)$ is the last occurrence of k and thus σ contains the minimum superpattern $ikijiki$ with the last letter of the minimum superpattern occurring on the last letter of σ. If $\sigma(3) = j$, note that there exists a

ki pattern where $\sigma(2) = k$ is the k of the pattern before $\sigma(c_2) = j$ since there exists a ki pattern before at least one j, $\sigma(2) = k$ is the first occurrence of k, and $\sigma(c_2)$ is the last occurrence of j. Thus, σ contains the minimum superpattern $ikjijki$ with the last letter of the minimum superpattern occurring on the last letter of σ.

Since any $i, j, k \in [3]$ can be permuted by isomorphisms, all strict minimal superpatterns for $[3]^3$ of length $n \geq 8$ contain a minimum superpattern for $[3]^3$ with the last letter of the minimum superpattern occurring on the last letter of σ. ■

Therefore, the words that fail to create a superpattern of $[3]^3$ do not contain a complete embedding of one of the strict minimal superpatterns of length seven, since by Lemma 3.6 all strict minimal superpatterns contain a strict minimal superpatterns of length seven. All the words contain some portion of a strict minimal superpattern of length seven up to isomorphism since the first two letters are fixed as i and j and each strict minimal superpattern of length seven can be written in the same manner. Define an *i-fold progression* as the amount of the 2^{n-2} words which contain the first through the ith letters of a unique strict minimal superpattern of length seven but not the $i+1$st letter. Thus, 2-fold progression is guaranteed by the fixed i and j occurring on the first and second positions of each word. The third position must be an i or a k since no two adjacent letters are the same letter. Let the strict minimal superpatterns of length seven with the first three positions containing the pattern iji be called type A and the strict minimal superpatterns of length seven with the first three positions containing the pattern ijk be called type B.

First, consider the strict minimal superpatterns of type A, namely $ijikiji$ and $ijikjij$, where $i, j, k \in [3]$ with $i \neq j \neq k$. A word that satisfies 3-fold progression contains the pattern iji on the first three positions, but no k afterwards. There is one such word, namely $ijijij\ldots$, which satisfies a 3-fold progression. For a 4-fold progression to occur, the word must contain the pattern iji on the first three positions followed by a k which has no i or j after it, otherwise a 5-fold progression will occur. There is only one such word, namely $ijijij\ldots k$, where the only occurrence of k is at the end of the word which satisfies a 4-fold progression. Since any other occurrence of k on the $(n-4)$ remaining positions results in a 5-fold progression, there are $n-4$ ways for the word to contain a 5-fold progression for each possible letter that can follow k. It is guaranteed that the word will contain an i or a j after any k not occurring on the last position, therefore a 5-fold progression is contained in the word if the pattern ki is not followed by a j or the pattern kj is not followed by an i. There are $2(n-4)$ such words, namely any word which follows the pattern $ijijij\ldots kikiki\ldots$ or the pattern $ijijij\ldots kjkjkj\ldots$, which satisfy a 5-fold progression. In order for a word to contain a 6-fold progression, it must contain the 5-fold progression pattern $ijijij\ldots kikiki\ldots$ followed by a j or the 5-fold progression pattern $ijijij\ldots kjkjkj\ldots$ followed by an i. This corresponds to all the ways in which non-consecutive choices can be made on length $n-3$. There are $2\binom{n-4}{2}$ such words, namely any word which follows the pattern $ijijij\ldots kikiki\ldots jkjkjk\ldots$ or the pattern $ijijij\ldots kjkjkj\ldots ikikik\ldots$, which satisfy a 6-fold progression. Therefore, the total count for the amount of words which do not contain a complete embedding of one of the type A strict minimal superpatterns of length seven is

$$\begin{aligned} B_A(n) &= 1+1+2(n-4)+2\binom{n-4}{2} \\ &= n^2-7n+14. \end{aligned}$$

Next, consider the strict minimal superpatterns of type B, namely $ijkijki$, $ijkikji$, $ijkijik$, $ijkjijk$ and $ijkjikj$, where $i, j, k \in [3]$ with $i \neq j \neq k$. There are no words that satisfy a 3-fold progression since all words containing the pattern ijk on the first three positions contain either an i or a j immediately afterwards and there exists either the pattern $ijki$ or the pattern $ijkj$ on at least one of the strict minimal superpatterns of type B. This causes at least a 4-fold progression to occur. For only a 4-fold progression to occur, the word must contain either the pattern $ijki$ on the first four positions with no j or k afterwards, which is impossible, or the pattern $ijkj$ on the first four positions with no i afterwards. Otherwise a 5-fold progression will occur. There is only one such word, namely $ijkjkjk\ldots$, which satisfies a 4-fold progression. For a 5-fold progression to occur using the pattern $ijki$ as a basis pattern on the first four positions, the word must contain either the pattern $ijkij$ on the first five position with no i or k afterwards, which is impossible, or the pattern $ijkik$ on the first five positions with no j afterwards. Otherwise a 6-fold progression will occur. There is only one such word, namely $ijkikiki\ldots$, which satisfies a 5-fold progression using the pattern $ijki$ as a basis pattern on the first four positions. For a 5-fold progression to occur using the pattern $ijkj$ as a basis pattern on the first four positions, the word must contain the pattern $ijkij$ on the first five positions followed by an i which has no j or k after it. Otherwise a 6-fold progression will occur. There is only one such word, namely $ijkjkjk\ldots i$, where the only occurrence of i after position four is at the end of the word, which satisfies a 5-fold progression using the pattern $ijkj$ as a basis pattern on the first four positions. Since any other occurrence of i on the $(n-5)$ remaining positions results in a 6-fold progression, there are $n-5$ ways for the word to contain a 6-fold progression for each possible letter that can follow i using the pattern $ijkj$ as a basis pattern on the first four positions. It is guaranteed that the word will contain a j or a k after any i not occurring on the last position, therefore a 6-fold progression is contained in the word if the pattern ij is not followed by a k or the pattern ik is not followed by a j. There are $2(n-5)$ such words, namely any word which follows the pattern $ijkjkjk\ldots ijijij\ldots$ or the pattern $ijkjkjk\ldots ijijij\ldots$, which satisfy a 6-fold progression using the pattern $ijkj$ as a basis pattern on the first four positions. A word can also contain a 6-fold progression using the pattern $ijkij$ as a basis pattern on the first five positions if the word contains either the pattern $ijkiji$ on the first six positions with no k afterwards or the pattern $ijkijk$ on the first six positions with no i afterwards. There exists only one such word for each of these cases, namely $ijkijijij\ldots$ and $ijkijkjkjk\ldots$, which satisfy a 6-fold progression using the pattern $ijkij$ as a basis pattern on the first five positions. Lastly, a word can also contain a 6-fold progression if it contains the pattern $ijkik$ as a basis pattern on the first five positions followed by a j on one of the $n-5$ remaining positions that is not followed by an i. There are $n-5$ such words, namely any word that follows the pattern $ijkikiki\ldots jkjkjk\ldots$, which satisfy a 6-fold position using the pattern $ijkik$ as a basis pattern on the first five positions. Therefore, the total count for the amount of words which do not contain a complete embedding of one of the type B strict minimal superpatterns of length seven is

$$\begin{aligned} B_B(n) &= 1+1+1+2(n-5)+1+1+(n-5) \\ &= 3n-10. \end{aligned}$$

Therefore, the total amount of words that do not contain a complete embedding of one of the strict minimal superpatterns of length seven and thus fail to create a superpattern of $[3]^3$ is

$$\begin{aligned} B_{total}(n) &= B_A(n) + B_B(n) \\ &= n^2 - 7n + 14 + 3n - 10 \\ &= (n-2)^2. \end{aligned}$$

Thus, the total amount of minimal superpatterns up to isomorphism of any length n is

$$G_{total}(n) = 2^{n-2} - (n-2)^2.$$

The sequence generated by $G_{total}(n)$ exists as entry number $A024012$ in the On-line Encyclopedia of Integer Sequences [11], but with no context. Submission of this context is underway.

Lemma 3.7 *For all $n \geq 7$, the total amount of strict minimal superpattern of length n, $S_m(n) = (n-4)^2 - 2$.*

Proof. The last letter is unnecessary in a non-strict superpattern for the completion of any preferential arrangement of $[3]^3$, making the word on the first $n-1$ letters a valid minimal superpattern of length $n-1$. There are two choices for the nth letter since no two adjacent letters in the word are the same letter. Therefore, the total amount of non-strict superpatterns of length n up to isomorphism is the total amount of minimal superpatterns of length $n-1$ up to isomorphism times two. The amount of strict minimal superpattern of length n up to isomorphism, $S_m(n)$, is the total amount of minimal superpatterns of length n up to isomorphism minus any non-strict superpatterns of length n up to isomorphism. Therefore,

$$\begin{aligned} S_m(n) &= G_{total}(n) - 2[G_{total}(n-1)] \\ &= [2^{n-2} - (n-2)^2] - 2[2^{n-3} - (n-3)^2] \\ &= 2^{n-2} - n^2 + 4n - 4 - 2^{n-2} + 2n^2 - 12n + 18 \\ &= n^2 - 8n + 14 \\ &= (n-4)^2 - 2. \end{aligned}$$

■

The sequence generated by $S_m(n)$ exists as entry number $A008865$ in the On-line Encyclopedia of Integer Sequences [11], but with very little context. Submission of this context is underway.

Lemma 3.7 only accounts for the amount of strict minimal superpatterns of length n up to isomorphism. In order to obtain the amount of all strict superpatterns of $[3]^3$ of length n up to isomorphism, the amount of strict superpatterns of length n up to isomorphism in which there exist occurrences of two adjacent letters as the same letter must be added to the amount of strict minimal superpatterns of length n up to isomorphism.

Lemma 3.8 *For all $n \geq 8$, the amount of strict superpatterns of length n up to isomorphism in which there exist occurrences of two adjacent letters as the same letter, $S_a(n) = \sum_{m=7}^{n-1}[(n-4)^2 - 2]\binom{n-2}{m-2}$.*

Proof. Any strict superpattern of length n up to isomorphism in which there exists occurrences of two adjacent letters as the same letter will contain an embedded occurrence of a strict minimal superpattern of length m, where $7 \leq m < n$. Therefore, all such superpatterns up to isomorphism are found by inserting $n-m$ letters which cause two adjacent letters to be the same into strict minimal superpatterns of length m. These insertions can take place anywhere in the word except before the last letter since an occurrence of two adjacent letters as the same letter at the end of the word contradicts the fact that the last letter is necessary for the completion of at least one preferential arrangement of $[3]^3$ contained in that superpattern. Therefore, there are $n-m$ insertions into $n-1$ possible positions and there are $\binom{n-1}{n-m}$ ways to do this. Use of combination identities gives $\binom{n-1}{n-m} = \binom{n-1}{(n-1)-(m-1)} = \binom{n-1}{m-1}$. The total amount ways to insert all $m-1$ letters into the $n-1$ positions is then $\binom{n-2}{m-2}$ and since this insertion method can be done for all strict minimal superpatterns of length m up to isomorphism,

$$\begin{aligned} S_a(n) &= \sum_{m=7}^{n-1} S_m(n)\binom{n-2}{m-2} \\ &= \sum_{m=7}^{n-1}[(n-4)^2-2]\binom{n-2}{m-2}. \end{aligned}$$

■

Theorem 3.9 *For all $n \geq 7$ the total amount of strict superpatterns of length n, $S(n) = 6\sum_{m=7}^{n}[(n-4)^2-2]\binom{n-2}{m-2}$.*

Proof. The total amount of strict superpatterns of length $n \geq 7$, $S(n)$, is six times the total amount of strict minimal superpatterns of length n up to isomorphism plus the six times the amount of strict superpatterns of length n up to isomorphism in which there exists occurrences of two adjacent letters as the same letter. Therefore,

$$\begin{aligned} S(n) &= 6[S_m(n) + S_a(n)] \\ &= 6\Big[(n-4)^2 - 2 + \sum_{m=7}^{n-1}[(n-4)^2-2]\binom{n-2}{m-2}\Big] \\ &= 6\sum_{m=7}^{n}[(n-4)^2-2]\binom{n-2}{m-2}. \end{aligned}$$

■

There are a total of 3^n words of length n with alphabet size 3. Thus, the probability that a word on n letters contains all preferential arrangements of $[3]^3$ is

$$p_3(n) = \frac{6}{3^n}\sum_{m=7}^{n}[(n-4)^2-2]\binom{n-2}{m-2}.$$

It may be verified that $p_3(n)$ is a valid probability function by assuring that $\sum_{n\geq 7} p_3(n) = 1$. By using different forms of the derivatives of the geometric series, the verification is given as follows,

$$\begin{aligned} \sum_{n\geq 7} p_3(n) &= \sum_{n=7}^{\infty}\frac{6}{3^n}\sum_{m=7}^{n}[(n-4)^2-2]\binom{n-2}{m-2} \\ &= 6\sum_{m=7}^{\infty}(m^2-8m+14)\sum_{n=m}^{\infty}\frac{\binom{n-2}{m-2}}{3^n} \\ &= 6\sum_{m=7}^{\infty}(m^2-8m+14)\sum_{l=m-2}^{\infty}\frac{\binom{l}{m-2}}{3^{l+2+m-m}} \\ &= 6\sum_{m=7}^{\infty}\frac{(m^2-8m+14)}{3^m}\sum_{l=m-2}^{\infty}\frac{\binom{l}{m-2}}{3^{l-(m-2)}} \\ &= 6\sum_{m=7}^{\infty}\frac{(m^2-8m+14)}{3^m}\times\frac{3^{m-1}}{2^{m-1}} \\ &= 2\sum_{m=7}^{\infty}\frac{(m^2-8m+14)}{2^{m-1}} \end{aligned}$$

$$
\begin{aligned}
&= 2\sum_{m=7}^{\infty}\frac{m^2}{2^{m-1}} - 2\sum_{m=7}^{\infty}\frac{8m}{2^{m-1}} + 2\sum_{m=7}^{\infty}\frac{14}{2^{m-1}} \\
&= 2\sum_{m=7}^{\infty}\frac{m^2}{2^{m-1}} - 16\sum_{m=7}^{\infty}\frac{m}{2^{m-1}} + 28\sum_{m=7}^{\infty}\frac{1}{2^{m-1}} \\
&= \frac{33}{8} - 4 + \frac{7}{8} \\
&= 1.
\end{aligned}
$$

By using different forms of the derivatives of the geometric series, the the expected wait time can be computed as follows,

$$
\begin{aligned}
E(W) &= \sum_{n=7}^{\infty}\frac{6n}{3^n}\sum_{m=7}^{n}[(n-4)^2-2]\binom{n-2}{m-2} \\
&= 6\sum_{m=7}^{\infty}(m^2-8m+14)\sum_{n=m}^{\infty}\frac{n\binom{n-2}{m-2}}{3^n} \\
&= 6\sum_{m=7}^{\infty}(m^2-8m+14)\sum_{n=m}^{\infty}\frac{n-1+1\binom{n-2}{m-2}}{3^n} \\
&= 6\sum_{m=7}^{\infty}(m^2-8m+14)\sum_{n=m}^{\infty}\frac{\binom{n-2}{m-2}}{3^n} + 6\sum_{m=7}^{\infty}(m^2-8m+14)\sum_{n=m}^{\infty}\frac{n-1\binom{n-2}{m-2}}{3^n} \\
&= \sum_{n=7}^{\infty}p_3(n) + 6\sum_{m=7}^{\infty}\frac{(m^2-8m+14)(m-1)}{3^m}\sum_{l=m-1}^{\infty}\frac{\binom{l}{m-1}}{3^{l-(m-1)}} \\
&= 1 + 6\sum_{m=7}^{\infty}\frac{(m^3-9m^2+22m-14)}{3^m}\times\frac{3^m}{2^m} \\
&= 1 + 3\sum_{m=7}^{\infty}\frac{(m^3-9m^2+22m-14)}{2^{m-1}} \\
&= 1 + 3\sum_{m=7}^{\infty}\frac{m^3}{2^{m-1}} - 3\sum_{m=7}^{\infty}\frac{9m^2}{2^{m-1}} + 3\sum_{m=7}^{\infty}\frac{22m}{2^{m-1}} - 3\sum_{m=7}^{\infty}\frac{14}{2^{m-1}} \\
&= 1 + 3\sum_{m=7}^{\infty}\frac{m^3}{2^{m-1}} - \frac{27}{2}\Big[2\sum_{m=7}^{\infty}\frac{m^2}{2^{m-1}}\Big] + \frac{33}{8}\Big[16\sum_{m=7}^{\infty}\frac{m}{2^{m-1}}\Big] - \frac{3}{2}\Big[28\sum_{m=7}^{\infty}\frac{1}{2^{m-1}}\Big]
\end{aligned}
$$

$$
\begin{aligned}
&= 1+3\sum_{m=7}^{\infty}\frac{m(m-1)(m-2)+3m^2-2m}{2^{m-1}} - \frac{27}{2}\Big(\frac{33}{8}\Big)+\frac{33}{8}\Big(4\Big)-\frac{3}{2}\Big(\frac{7}{8}\Big)\\
&= \frac{3}{4}\sum_{m=7}^{\infty}\frac{m(m-1)(m-2)}{2^{m-3}}+\frac{9}{2}\Big[2\sum_{m=7}^{\infty}\frac{m^2}{2^{m-1}}\Big]-\frac{3}{8}\Big[16\sum_{m=7}^{\infty}\frac{m}{2^{m-1}}\Big]-\frac{79}{2}\\
&= \frac{3}{4}\Big(\frac{183}{4}\Big)+\frac{9}{2}\Big(\frac{33}{8}\Big)-\frac{3}{8}\Big(4\Big)-\frac{79}{2}\\
&= \frac{95}{8}\\
&= 11.875.
\end{aligned}
$$

This validates a numerical analysis of the problem done in Matlab [9]which gives the expected wait time for the creation of a strict superpattern as 11.820 using 1000 trials. See Appendix A for the Matlab code.

The ordinary generating function is

$$
\begin{aligned}
G_3(x) &= \sum_{n=7}^{\infty}\frac{6t^n}{3^n}\sum_{m=7}^{n}[(n-4)^2-2]\binom{n-2}{m-2}\\
&= 6\sum_{m=7}^{\infty}(m^2-8m+14)\sum_{n=m}^{\infty}\frac{\binom{n-2}{m-2}}{(\frac{3}{t})^n}\\
&= 6\sum_{m=7}^{\infty}(m^2-8m+14)\sum_{l=m-2}^{\infty}\frac{\binom{l}{m-2}}{(\frac{3}{t})^{l+2+m-m}}\\
&= 6\sum_{m=7}^{\infty}\frac{(m^2-8m+14)}{(\frac{3}{t})^m}\sum_{l=m-2}^{\infty}\frac{\binom{l}{m-2}}{(\frac{3}{t})^{l-(m-2)}}\\
&= 6\sum_{m=7}^{\infty}\frac{(m^2-8m+14)}{(\frac{3}{t})^m}\times\frac{3^{m-1}}{(3-t)^{m-1}}\\
&= 2t\sum_{m=7}^{\infty}\frac{(m^2-8m+14)}{(\frac{3-t}{t})^{m-1}}\\
&= 2t\sum_{m=7}^{\infty}\frac{m^2}{(\frac{3-t}{t})^{m-1}}-2t\sum_{m=7}^{\infty}\frac{8m}{(\frac{3-t}{t})^{m-1}}+2t\sum_{m=7}^{\infty}\frac{14}{(\frac{3-t}{t})^{m-1}}
\end{aligned}
$$

$$
\begin{aligned}
&= 2t\sum_{m=7}^{\infty}\frac{m^2}{(\frac{3-t}{t})^{m-1}} - 16t\sum_{m=7}^{\infty}\frac{m}{(\frac{3-t}{t})^{m-1}} + 28t\sum_{m=7}^{\infty}\frac{1}{(\frac{3-t}{t})^{m-1}} \\
&= 2t\sum_{m=7}^{\infty}\frac{m(m-1)}{(\frac{3-t}{t})^{m-1}} + 2t\sum_{m=7}^{\infty}\frac{m}{(\frac{3-t}{t})^{m-1}} - 16t\sum_{m=7}^{\infty}\frac{m}{(\frac{3-t}{t})^{m-1}} + 28t\sum_{m=7}^{\infty}\frac{1}{(\frac{3-t}{t})^{m-1}} \\
&= \frac{2t^2}{3-t}\sum_{m=7}^{\infty}\frac{m(m-1)}{(\frac{3-t}{t})^{m-2}} - 14t\sum_{m=7}^{\infty}\frac{m}{(\frac{3-t}{t})^{m-1}} + 28(3-t)\sum_{m=7}^{\infty}\frac{1}{(\frac{3-t}{t})^{m}} \\
&= \frac{2t^2}{3-t}\Big[\frac{142t^7-126t^6+378t^5}{(3-t)^4(3-2t)^3}\Big] - 14t\Big[\frac{21t^6-13t^7}{(3-t)^5(3-2t)^2}\Big] + 28(3-t)\Big[\frac{t^7}{(3-t)^6(3-2t)}\Big] \\
&= \frac{2t^7(16t^2-63t+63)}{(3-t)^5(3-2t)^2}.
\end{aligned}
$$

4 OPEN PROBLEMS

This work for the cases of $n(2,2)$ and $n(3,3)$ creates open problems of interest. A few such question which require further investigation are:

(i) What are the lower bounds and containment length for cases of $n(4,4)$?

(ii) What are the lower bounds and containment lengths for cases of $n(m,m)$ with $m \geq 5$ and can some form of a generalization be arrived at for higher cases?

(iii) What are the lower bounds and containment lengths for cases in higher dimensions?

Some investigation into case (i) has revealed that many difficulties will arise in the achievement of a counting method and the fold method of counting used in the $n(3,3)$ case for containment will not work for the case of $n(4,4)$. One major complication to note is that a minimum superpattern for $[4]^4$ of length 12 can be constructed using the construction method found in [4], but there exist strict superpatterns for $[4]^4$ of lengths larger than 12 which do not contain one of the minimum superpatterns. One such example can be constructed following the construction of the type A strict superpatterns for $[3]^3$, i.e., 121312141213121.

BIBLIOGRAPHY

[1] S. Abraham, G. Brockman, A. Godbole, S. Sapp, Omnibus sequences, coupon collection, and missing word counts, to appear in *Methodology and Computing in Applied Probability* DOI 10.1007/s11009-011-9247-6, (2011).

[2] M. H. Albert, M. D. Atkinson, C. C. Handley, D. A. Holton, W. Stromquist, On packing densities of permutations, *Electron. J. Combin.* 9 (2002), #R5.

[3] M. Bóna, *A Walk Through Combinatorics: An Introduction to Enumeration and Graph Theory*, Singapore: World Scientific Publishing Co. Pte. Ltd. (2002).

[4] A. Burstein, P. Hästö, T. Mansour, Packing patterns into words, *Electronic Journal of Combinatorics* 9(2) (2003), #R20.

[5] A. Burstein, P. Hästö, Packing sets of patterns, *European Journal of Combinatorics* 31 (2010), 241-253.

[6] H. Eriksson, K. Eriksson, S. Linusson, J. Wästlund, Dense packing of patterns in a permutation, *Proceedings of the 15th Conference on Formal Power Series and Algebraic Combinatorics* Melbourne, Australia, 812 July 2002, paper #26.

[7] V. Jalínek, T. Mansour, Wilf-equivalence on k-ary words, compositions, and parking functions, *Electronic Journal of Combinatorics* 16 (2009), #R58.

[8] A. Marcus, G. Tardos, Excluded permutation matrices and the Stanley-Wilf conjecture, *Journal of Combinatorial Theory Series A* 107 (2004), 153-160.

[9] Matlab & Simulink. Version 7.10.0.499(R2010a) Student Version, 2010.

[10] A. Miller, Asymptotic bounds for permutations containing many different patterns, *Journal of Combinatorial Theory Series A* 116 (2009), 92-108.

[11] N. Sloane, *The Encyclopedia of Integer Sequences.* March 18, 2012. http://oeis.org.

[12] H. F. Wilf, *generatingfunctionology*, Philadelphia: Academic Press, Inc. (1994).

[13] H. F. Wilf, The patterns of permutations, *Discrete Mathematics* 257 (2002), 575-583.

APPENDIX

Matlab Code

In this Appendix, the code used in Matlab [9] for the numerical analysis of $p_3(n)$ is given verbatim. The reader should note that all comments given in Matlab are preceded by the % symbol.

AverageLength

```
function AverageLength(k,a,t)

%Finds the average minimum length needed for a randomly generated string of
%numbers with a distinct elements to contain all preferential patterns of k
%distinct elements using t trials.

n=zeros(1,t);

for i=1:t
    n(i)=FindLength(k,a);
end

M=mean(n);
V=var(n);
```

```
fprintf('Results for %d trials:\n',t)

fprintf('Average Length: %6.4f \n',M)

fprintf('Variance: %6.4f \n',V)
```

The function AverageLength depends on other written functions, each of which is given with internal comments.

FindLength

```
function Length = FindLength(k,a)


%Finds the minimum length, n, needed for a randomly generated string of

%numbers with a distinct elements to contain all preferential patterns of

%k distinct elements.


x=zeros(1,a);


for i=1:a

    x(i)=1/i;

end


y=sum(x);
```

```
c=ceil(a*k*y); %Gives expected average.

F=cell(c,1);

b=length(PreferentialArrangements(k,k));

f=cell(1,b*c);


for i=1:c

    F{i}=FindPatterns(k,a,i); %Finds patterns in random text on length i.

    if all(F{i})

        F{i}=F{i};

    else F{i}=[];

    end

    f{i}=size(F{i},1); %Finds sum of each cell array.

end


m=cell2mat(f); %Converts to matrix.

n=find(m==b,1,'first'); %Finds first b, which corresponds to minimum length

                        %of random text needed for all permutations to be

                        %found.


Length=n; %Displays minimum length.


end
```

PreferentialArrangements

```
function pa = PreferentialArrangements(n,a)


%Finds all preferential arrangements of n labeled elements when only k<=a

%ranks are allowed.


m=min(n,a);

p=permsrep(1:m,n);

u=cell(length(p),1);


for i=1:length(p)

    [r,r,r]=unique(p(i,:));

    u{i}=r;

end


u=cell2mat(u);

u=unique(u,'rows');


pa=u;

end
```

FindPatterns

```
function positions = FindPatterns(k,a,n)


%Finds the positions of all preferential patterns of k distinct elements in
%randomly generated string of numbers with a distinct elements of length n.


P=permsrep(1:k,k);
P=num2str(P);
P=cellstr(P);
P=regexprep(P,' ','');
P=cell2mat(P);
R=randi(a,[1,n]);
R=num2str(R);
R=regexprep(R,' ','');
y=cell((k*length(P)),1);
x=zeros((k*length(P)),1);


for h=1:(length(P))
    for j=1:k
        i=j+(k*(h-1));
        y{i}=strfind(R,P(h,j)); %Lists position values of patterns in cells.
        x(i)=length(y{i}); %Gives length of each cell.
```

```
    end
end

s=max(x); %Finds the maximum length of all the cell lengths.

for i=1:(k*length(P))
     y{i}(end+1+(s-length(y{i}))) = 0; %Pads cells with zeros to make all
                                       %cells equal length.
end

Y=cell2mat(y); %Forms matrix Y from all the cells.
C=cell(length(P),1);
f=cell(length(P),1);

for i=1:length(P)
    C{i}=Y((((i-1)*k)+1):(i*k),:); %Partitions Y into blocks.
    f{i}=FindPositions2(C{i}); %Gives position values of each pattern.
    if ischar(f{i})
        f{i}=zeros(1,k);
    end
end
```

```
f=cell2mat(f);

u=cell(length(P),1);


for i=1:length(P)

    [r,r,r]=unique(P(i,:));

    u{i}=r;

end


u=cell2mat(u);

H=horzcat(u,f);

h=size(H,1);


for i=1:h-1

    for j=i+1:h

        if and(isequal(H(i,1:k),H(j,1:k)),

        and(H(i,2*k)>H(j,2*k),H(j,2*k)>0))

        H(i,:)=H(j,:);

        elseif and(isequal(H(i,1:k),H(j,1:k)),

        and(H(i,2*k)>H(j,2*k),H(j,2*k)==0))

        H(j,:)=H(i,:);

        elseif and(isequal(H(i,1:k),H(j,1:k)),

        and(H(i,2*k)<H(j,2*k),H(i,2*k)>0))
```

```
        H(j,:)=H(i,:);

        elseif and(isequal(H(i,1:k),H(j,1:k)),

        and(H(i,2*k)<H(j,2*k),H(i,2*k)==0))

        H(i,:)=H(j,:);

        elseif and(isequal(H(i,1:k),H(j,1:k)),H(i,2*k)==H(j,2*k))

        H(i,:)=H(j,:);

        end

    end

end


U=unique(H,'rows');


positions=U; %Displays ordered position values of found preferential patterns

             %in matrix form. Each row consists of 2k elements, the first k

             %elements are the preferential pattern and the last k elements

             %are the position values for that pattern.


end
```

permsrep

```
function pr = permsrep(v,n)

%Finds all permutations of length n with replacement of the elements of
%the vector v. Majority of code due to Peter Acklam and found online at
%http://www.mathworks.com/matlabcentral/newsreader/view_thread/52610.

 m = length(v);
 X = cell(1, n);
 [X{:}] = ndgrid(v);
 X = X(end : -1 : 1);
 y = cat(n+1, X{:});
 y = reshape(y, [m^n, n]);

 pr=y;
end
```

FindPositions2

```
function Position = FindPositions2(Y)

%Finds the position values in order for each preferential arrangement.
```

```
s=size(Y,2);
k=size(Y,1);

for h=2:k
    for j=1:s-1
        Y(1,2:end)=0; %Replaces all elements in first row except the first
                      %with zero.
        if  and(Y(h,j)<=max(max(Y(1:h-1,1:end))),Y(h,j)>0)
            Y(h,1:j)=0; %Finds all values of a row that are less than the
                        %largest value of the preceding rows and replaces
                        %them with zeros.
        elseif Y(h,j)>max(max(Y(1:h-1,1:end)))
            Y(h,j+1:end)=0; %Finds the first value of a row that is greater
                            %than the largest value of the preceding rows
                            %and replaces every value to the right of it
                            %with zeros.
        end
    end
end

Y=transpose(Y); %Changes rows to columns so elements are listed in correct
```

```
                 %order.
T=transpose(Y(Y>0)); %Finds all nonzero values of Y and lists them in
                     %vector form. This vector is the position values of
                     %the letters of the message in order.

if length(T)<k %Creates an error message for the event that the letters
               %of the message could not be found in order in the text.
    T='Error: Entire message not found in this text.';
end

t=sort(T);

if T==t       %Takes care of anomaly in code. (If the last value of a
              %row before the padded single zero is less than the
              %largest value of the preceding rows, it is still displayed
              %even though the code should replace it with a zero.)
else T='Error: Entire message not found in this text.';
end

Position=T; %Displays vector of position values of the message.

end
```

VITA

MARTHA LIENDO

Education:	A.A.S. Education, Teacher Preparation Virginia Highlands Community College Abingdon, Virginia 2007 A.A.S. General Studies Virginia Highlands Community College Abingdon, Virginia 2007 B.S. Mathematics King College Bristol, Tennessee 2010 M.S. Mathematical Sciences East Tennessee State University Johnson City, Tennessee 2012
Professional Experience:	National Science Foundation Graduate GK-12 Fellow, East Tennessee State University Johnson City, Tennessee 2010-2012