Mining the Online Encyclopedia of Integer Sequences

Hieu D. Nguyen and Douglas Taggart
Department of Mathematics
Department of Computer Science
Rowan University
Glassboro, NJ 08028
nguyen@rowan.edu
taggar17@students.rowan.edu

3/6/2013

Abstract

This paper describes an experimental mathematics project aimed at mining the Online Encyclopedia of Integer Sequences for new mathematical identities. We discuss methods used to store, compare and match integer sequences and describe an implementation using MySQL and Mathematica. A summary of our results is presented, along with a sample of ten new conjectures that were found, some which we believe to be new and interesting and some which illustrate how false matches can arise.

1 Introduction

Integer sequences such as the Fibonacci sequence $F_n = \{0, 1, 1, 2, 3, 5, 8, ...\}$ are fundamental objects that arise in practically all areas of mathematics and science. A study of their properties, as expressed through mathematical identities, is important towards making fruitful connections between these different areas. In the past, mathematical identities were discovered empirically and derived from patterns that were inferred through hand computations, e.g. Catalan discovered in 1680 that the squares of the Fibonacci numbers satisfy the identity

$$F_n^2 = F_{n-1}F_{n+1} - (-1)^n \tag{1}$$

With the power of modern computers and the Internet, it is now relatively easily to generate transformations of sequences and identify them using the Online Encyclopedia of Integer Sequences (OEIS) ([7]), a searchable database containing information on over 200,000 integer sequences. Originally created by Neil J. A. Sloane in the 1970's as a handbook, the OEIS is now widely used by mathematicians and scientists around the world to help them identify integer sequences; not only does it give a list of initial terms of the sequence, but it also describes any known properties and connections to other sequences. For example, feeding the list of entries '0,1,1,2,3,5,8' into the OEIS search engine yields the Fibonacci sequence as

a match, labeled by record number A000045, and 88 other sequences having the same terms. One learns from this record in OEIS that the Fibonacci sequence is defined by its well-known recurrence $F_n = F_{n-1} + F_{n-2}$ and satisfies many other identities besides (1), another example being the classic identity

$$\sum_{k=0}^{n} F_k = F_{n+2} - 1 \tag{2}$$

However, instead of typing sequences by hand one at a time into OEIS to find matches, we propose a data mining approach where we automate this process using high-performance computers. The idea of data mining the OEIS has been proposed as early as 1994 by Liu ([5]) and more recently by Colton [3]. Of course, there have been efforts at developing algorithms to recognize integers sequences, e.g. Bergeron and Plouffe's *gfun* Maple package (see [2]) for computing the generating function of a given series given the first few terms, or to recognize certain families of integer sequences, e.g. see [1]. The package *gfun* is now a part of the OEIS's *superseeker* email server (see [8]), which incorporates a host of software packages to identity a given sequence, including applying over 120 transformations to determine if they matches sequences in the OEIS. Users can submit sequences to *superseeker* by sending an email message to superseeker@oeis.org. However, superseeker has its disadvantages besides requiring users to submit requests one at a time: it is not able to match two different transformations (of possibly different sequences) unless one of them is already stored in the OEIS. Moreover, it does not use a similarity measure to report the quality of its matches, i.e. how well two sequences matches. In comparison, what we propose in this paper goes beyond the standard approach of investigating a particular type of sequence or a small subset of them and instead seeks to mine the OEIS as a whole using basic data mining techniques to obtain interesting matches between integer sequences.

This paper describes Project Eureka, an experimental mathematics research project aimed at mining the OEIS for new mathematical identities. Our approach is to store integer sequences and their transformations in a database and apply an appropriate similarity measure to match sequences numerically. For example, such a database would contain not only the Fibonacci sequence F_n as specified by list A000045S1T1=$\{0, 1, 1, 2, 3, ..., 39088169\}$, but also many of its transformations, including its partial sums $\sum_{k=0}^{n} F_k$ (denoted by T2), specified by list A000045S1T2=$\{0, 1, 2, 4, 7, ..., 102334154\}$. We then perform a computer-automated search of this database where we compare and match entries using a similarity measure defined later in this paper. As a result, we find that the two lists A000045S1T1 and A000045S1T2 indeed match (more precisely, their first-order differences match); this translates to identity (2). Our goal is to discover new identities using this approach.

In addition to discussing our methods for storing, comparing, and matching sequences and outlining an implementation using MySQL and Mathematica, we present a summary of our results and describe ten experimental conjectures derived from our work, some of which we believe to be new and interesting and some which illustrate how false matches can arise. A searchable database containing all of our experimental matches can be accessed on our Eureka website ([4]).

2 Matching Integer Sequences

2.1 Similarity Measure

The problem of matching finite integer sequences using an effective similarity measure becomes a mathematically interesting one if we take into account how entries are stored in OEIS, the nature of initial terms in these entries, and the types of identities that we are searching for:

1. Sequences stored in OEIS vary in length from 4 to 100 terms, e.g. the list for record A028444 (Busy Beaver sequence) contains only 5 terms, $\{0, 1, 4, 6, 13\}$, whereas the list for record A000045 (Fibonacci sequence) contains 39 terms, $\{0, 1, 1, 2, 3, ..., 39088169\}$.
2. Many sequences have the same initial terms 0 and 1.
3. Sequences may be shifts, translations or scalar multiples (or all three) of one another as illustrated by identity (2).

For example, let us consider two sequences a_n and b_n whose initial terms are given by the lists

$$\begin{aligned} a_n &= \{1, 1, 2, 3, 5, 8, 13, 21, \mathbf{34}, 55\} \\ b_n &= \{1, 1, 2, 3, 5, 8, 13, 21, \mathbf{47}, 55\} \end{aligned} \tag{3}$$

Would these two sequences be considered a match even though their lists disagree only at the 9th position (highlighted in bold) and yet match everywhere else? How about the two lists

$$\begin{aligned} a_n &= \{1, 1, 2, 3, 5, 8, 13, 21, 34, \mathbf{55}\} \\ c_n &= \{\mathbf{55}, 89, 144, 233, 377, 610\} \end{aligned} \tag{4}$$

where there is a match of only one term between the two lists, namely 55 (highlighted in bold), with the tail of c_n (last term) matching the head of d_n (first term)? Which is the better match, (3) or (4)? By this we mean the likelihood that two sequences are the same (modulo a shift in their indices). Or how about a third case where the two lists are

$$\begin{aligned} a_n &= \{1, 1, \mathbf{2}, \mathbf{3}, \mathbf{5}, \mathbf{8}, \mathbf{13}, \mathbf{21}, \mathbf{34}, \mathbf{55}\} \\ d_n &= \{\mathbf{2}, \mathbf{3}, \mathbf{5}, \mathbf{8}, \mathbf{13}, \mathbf{21}, \mathbf{34}, \mathbf{55}, 89, 144, 233\} \end{aligned} \tag{5}$$

Which is the best match, (3), (4) or (5)? Assuming that the sequences a_n, b_n, c_n, and d_n are increasing and that lists given for them are correct and free of errors, we argue that (5) is the best match since there is a high probability that $a_{n+2} = d_n$ (identically as sequences) whereas the probability of $a_{n+9} = c_n$ is much less and the probability of $a_n = b_n$ is zero.

Motivated by the above examples, we therefore employ a simple similarity measure based on a 'head-bites-tail' contiguous overlap of sequences. We begin with preliminary definitions to help us mathematically describe this notion.

Definition 1. Let $\{a_n\}_{n=0}^{N-1}$ and $\{b_m\}_{m=0}^{M-1}$ be two finite sequences (i.e. lists) of length N and M, respectively.

(a) A *run* of a_n of length L is a finite subsequence $\{a_{n_0}, a_{n_0+1}, ..., a_{n_0+L-1}\}$ consisting of L consecutive elements of a_n starting at a_{n_0}.

(b) An *overlap* between a_n and b_m is a run of length L that appears in both sequences, i.e. there exists non-negative integers n_0, m_0 and a positive integer L such that $a_{n_0+k} = b_{m_0+k}$ for $k = 0, 1, ..., L-1$.

(c) A *head-bites-tail (HBT) overlap* of length L between a_n and b_n is an overlap that begins at the head of one sequence and ends at the tail of either sequence, i.e. either $a(N-L+k+1) = b(k)$ for $k = 0, 1, ..., L-1$ or $a(k) = b(M-L+k+1)$ for $k = 0, 1, ..., L-1$.

(d) The *maximum HBT overlap* between a_n and b_n, denoted by $L_{\max}$, is the length of their longest HBT overlap. We set $L_{\max} = 0$ if no HBT overlap exists.

For example, $L_{\max} = 0$ for the two lists in (3) whereas $L_{\max} = 8$ for the two lists in (5). We provide pseudcode for computing maximum HBT overlap in Appendix B.

The notion of maximum HBT overlap now allows us to define a similarity measure based on 'distance' between two sequences, which we discuss next.

Definition 2. Let $\{a_n\}_{n=0}^{N-1}$ and $\{b_m\}_{m=0}^{M-1}$ be two finite sequence of length N and M, respectively, with maximum HBT overlap $L_{\max}$.

(a) The *HBT distance* between a_n and b_m is defined to be

$$d := d(a_n, b_m) = N + M - 2L_{\max}$$

(b) The *relative HBT distance* between a_n and b_m is defined to be

$$d_r := d_r(a_n, b_m) = \frac{N + M - 2L}{N + M} = 1 - \frac{2L_{\max}}{N + M}$$

Some comments are in order regarding definition (2):

1. Our definition of HBT distance essentially counts the number of terms in a_n and b_n that do *not* overlap. It follows that if a_n and b_n are exactly the same sequence, then $d = d_r = 0$. If no HBT overlap exists, as in (3), then $d_r = 1$ since $L_{\max} = 0$. As for (4), we have $L_{\max} = 1$, $d = 14$, and thus $d_r = 7/8$. Of course, (5) yields the smallest distance since $L_{\max} = 8$ which implies $d = 5$ and $d_r = 5/21$.
2. Unfortunately, our definition of HBT distance does not define a true distance function in that it fails the triangle inequality. As a counterexample, define $a_n = \{1, 2, 3\}$, $b_n = \{0, 2, 3\}$, and $c_n = \{2\}$. Then $d(a_n, b_n) = 6$, $d(a_n, c_n) = 2$, and $d(b_n, c_n) = 2$. The triangle inequality $d(a_n, b_n) \leq d(a_n, c_n) + d(c_n, b_n)$ thus fails in this case.

Definition 3. Two finite sequences a_n and b_m are said to:

(a) *match* if $0 \leq d_r < 1$ and write $a_n \approx b_n$ to indicate this; if $d_r = 0$, then it is said to be a *perfect match*.

(b) *not match* if $d_r = 1$.

Thus, we conclude from the definition above that the two sequences a_n and b_n in (3) do not match since $d_r = 1$ whereas (4) and (5) yield matches ($d_r = 7/8$ and $d_r = 5/21$, respectively).

2.2 Linear Matches

To expand our search for interesting identities, we allow for matches between sequences that have a linear relationship, i.e. translations and/or scalar multiples of each other. For example, the two lists

$$\begin{aligned} a_n &= \{1, 1, 2, 3, 5, 8, 13, 21, 34, 55\} \\ b_n &= \{13, 22, 37, 61, 100, 163, 265, 430, 697, 1129\} \end{aligned} \tag{6}$$

satisfy the identity

$$b_n = 3a_{n+4} - 2 \tag{7}$$

More generally, two sequences $\{a_n\}$ and $\{b_n\}$ are said to be *linear* if there exists constants s, t, and C such that

$$sa_n + tb_n = C \tag{8}$$

To catch this relationship between a_n and b_n, it suffices to compute their first differences and normalize these differences by their greatest common divisor (GCD). The following theorem justifies our approach:

Theorem 4. *Let a_n and b_n be two non-trivial finite sequences whose first differences are given by $\Delta a_n = a_{n+1} - a_n$ and $\Delta b_n = b_{n+1} - b_n$. Moreover, let $A = GCD\{\Delta a_n\}$ and $B = GCD\{\Delta b_n\}$. Then*

$$\frac{\Delta a_n}{A} = \frac{\Delta b_n}{B} \tag{9}$$

if and only if a_n and b_n are linear and satisfy (8) with $s = B$, $t = -A$, and $C = Ba_0 - Ab_0$.

Proof. Assume that (9) holds, which is equivalent to

$$Ba_{n+1} - Ab_{n+1} = Ba_n - Ab_n \tag{10}$$

To prove that a_n and b_n are linear and satisfy (8) with $s = B$, $t = -A$, and $C = Ba_0 - Ab_0$, we use mathematical induction. The base case ($n = 1$) is true because of (10):

$$sa_1 + tb_1 = Ba_1 - Ab_1 = Ba_0 - Ab_0 = C$$

As for the inductive step, assume $sa_n + tb_n = C$. Then again, because of (10), we have

$$sa_{n+1} + tb_{n+1} = Ba_{n+1} - Ab_{n+1} = Ba_n - Ab_n = sa_n + tb_n = C$$

as desired.

Conversely, assume a_n and b_n are linear, i.e. there exists values s, t, and C such that (8) holds. Since (8) also holds when n is replaced by $n + 1$, we subtract these two cases to obtain

$$s\Delta a_n = -t\Delta b_n \tag{11}$$

which implies

$$s \cdot GCD\{\Delta a_n\} = -t \cdot GCD\{\Delta b_n\} \tag{12}$$

Now divide (11) by (12) to obtain (9). This completes the proof. □

The above lemma justifies our next definition of a linear match between two finite sequences.

Definition 5. Two finite sequences a_n and b_m are said to be a *linear match* if $\frac{\Delta a_n}{A}$ and $\frac{\Delta b_n}{B}$ match. We denote this by $a_n \sim b_n$.

For example, the two lists a_n and b_n given in (6) form a linear match since

$$\begin{aligned} \Delta a_n &= \{0, 1, 1, 2, 3, 5, 8, 13, 21\} \\ \Delta b_n &= \{9, 15, 24, 39, 63, 102, 165, 267, 432\} \end{aligned} \tag{13}$$

It follows that $A = 1$, $B = 3$, and therefore $\frac{\Delta a_n}{A}$ and $\frac{\Delta a_n}{B}$ match. This linear match corresponds to the identity

$$b_n = 3a_{n+4} - 2 \tag{14}$$

3 Mining the OEIS

In this section we describe an implementation using MySQL and Mathematica to automate our search for linear matches.

3.1 MySQL

We considered only the first 170,000 integer sequences stored in the OEIS (even though it now contains over 200,000 sequences) since there are gaps that remain to be filled beyond this range. Seventeen different transformations were applied to each sequence (including the identity transformation so as to include the original sequence itself); see Table 5 in the Appendix for a complete list of transformations. This resulted in a collection of almost 3 million sequences, which we store in a MySQL database as a single table, called *Sequence Transformations*. Each term in the list for each sequence is stored as a separate string and allowed to be up to 100 digits long; larger terms were truncated from the sequence. Initially, we stored each term in a separate row in our table. However, it was found that search times were significantly reduced if the terms were stored in rows containing three consecutive terms (called EntryOne, EntryTwo, and EntryThree) as illustrated for the list A000045=$\{0, 1, 1, 2, 3, ..., 39088169\}$ in Table 1. This window format tripled the amount of memory that was needed to store all of our lists to over 7 gigabytes, but it was well worth the expense since memory is relatively cheap compared to the additional computing power that would have been needed to achieve the same performance, something we discuss further later in this paper.

To catch linear matches, a second table called *Sequence Transformations Differences GCD* was created to store first differences of each sequence in Sequence Transformations divided by its GCD. This second table is where we apply our matching algorithm using Mathematica, which we discuss next.

ID	Label	Position	EntryOne	EntryTwo	EntryThree
1	A000045S1T1	0	0	1	1
2	A000045S1T1	1	1	1	2
3	A000045S1T1	2	1	2	3
4	A000045S1T1	3	2	3	5
...	...	...	...	...	...
38	A000045S1T1	37	24157817	39088169	Null
39	A000045S1T1	38	39088169	Null	Null

Table 1: Sequence Transformations MySQL Table Format - Entry A000045S1T1

3.2 Mathematica

The following three steps describe an implementation of our matching algorithm using Mathematica software version 8.0 ([6]) . We used Mathematica's Databaselink package to communicate with MySQL. One of the nice features of Mathematica is its ability to store and calculate arbitrarily long integers.

Step 1. Each list in table Sequence Transformations Differences GCD (STDG) is compared against all other entries in this table for potential matches using the MySQL 'select' command. Given such a list, called the *reference*, we extract from it a window of three terms (or entries) greater than or equal to 10,000. For example, suppose the reference list is given by record A000045S1T1 (Fibonacci sequence). We then extract the three terms $\{10946, 17711, 28657\}$ and use them to fetch other lists, called *candidates*, in STDG containing the same window of three terms. We find that this MySQL query outputs 146 candidates, each representing a potential linear match with the reference since they all contain the same window of three terms $\{10946, 17711, 28657\}$. Here is a partial list of the first 15 candidates and the last 10 candidates in lexicographic order:

A000045S1T1	A001595S1T1	A006327S1T1	A157727S1T1	A167616S1T1
A000045S1T2	A001611S1T1	A006355S1T2	A157728S1T1	A167816S1T2
A000071S1T1	A001891S1T4	A006355S1T4 ...	A157729S1T1	A168193S1T1
A000126S1T4	A001911S1T1	A007435S1T17	A161468S1T2	A168193S1T4
A001588S1T1	A001911S1T4	A007436S1T16	A166876S1T1	A169622S1T1

Our rationale for using large terms to find candidates as opposed to small terms, e.g. $\{0, 1, 1\}$, is because the number of candidates in this case would be extremely large and many of them would *not* lead to true identities, thus making our algorithm quite inefficient.

Step 2. Before doing a full list comparison between the reference with each of the candidates found in Step 1, we trim all of them to remove initial terms that are trivial, i.e. equal to -1,0, or 1, in order to catch matches where the reference and candidate represent the same sequence yet initialized differently. For example, the reference A000045S1T1 would be trimmed to $\{2, 3, 5, ..., 39088169\}$ where the initial terms 0,1,1 have been deleted.

Step 3. Maximum HBT overlap and relative distance are then computed between the reference and each candidate (pseudocode for computing maximum HBT overlap is given

Window Size (Number of Terms)	Run Time (Days)
1	38.96
2	3.5
3	2.67

Table 2: Search Run Times Based on Window Size

Computer (Model Year)	Configuration (Processor, RAM)	Run Time (Days)
Apple iMac (mid-2011)	2.7 GHz Intel Core i5 quad-core, 4 GB RAM	2.67
Apple Mac Pro (mid-2010)	3.2 GHz Intel Xeon quad-core, 32 GB RAM	0.62

Table 3: Search Run Times Based on Computer Model

in Appendix B). Observe that a candidate would be considered a match with the reference if $d_r < 1$ according to Definition 3. However, to avoid catching weak, trivial, and even false matches which do not correspond to a true identity, we consider only strong matches where $L_{\max} \geq 4$ and $d_r \leq 1/2$. This is the criteria we use to indicate a match and store all such matches in a MySQL table called Linear Matches. For example, we find that A000045S1T1 (reference) and A000045S1T2 (candidate) indeed form a linear match with $L_{\max} = 34$ and $d_r = 0.02857$. On the other hand, A000045S1T1 does not match with A137574S1T1 $= \{2, 3, 5, 8, 13, 21, 89, 233, 1597, 17711, ..., 53316291173\}$, defined as Fibonacci numbers and their distinct prime divisors having the same number of decimal digits. In this case we find $L_{\max} = 0$ and thus $d_r = 1$.

3.3 Search Run Times

Table 2 gives a summary of the search run times for various window sizes based on a search of the entire STDG table for linear matches, which at the time, contained eleven transformations of the first 170,000 sequences in OEIS. Observe that using a window size of 3 terms dramatically reduced our run time by almost a factor of 15 in comparison to a window size of 1 term. Moreover, two trials were performed, both using a window size of 3 terms, but each on a different Apple computer: iMac and Mac Pro. A comparison of their run times is given in Table 3. The superior performance of the Mac Pro was due to it having 8 times more RAM than the iMac.

3.4 Summary of Results

Using the implementation described above, we found approximately 650,000 linear matches thus far between sequences in OEIS and their transformations. These matches we stored in a MySQL table called *Linear Matches*. An example of such a match (A000045S1T1 $\sim$ A000045S1T2) illustrating Linear Matches is given in Table 4. Actually, each match appears twice in Linear Matches (with the order of the two labels reversed) because of our search algorithm; each sequence serves as a reference in one match and then as a candidate in the other.

ID	Label1	Label2	Overlap	Distance	Scaling	Translation	Shift
2087	A000045S1T1	A000045S1T2	34	0.02857	1	1	-2

Table 4: Record of match A000045S1T1 ∼ A000032S1T1 in Linear Matches table

However, based on an examination of a small sample of these matches, it appears that almost all of them correspond to identities that are already known or trivial due to variations of the same sequence being stored in the OEIS. For example, the match A000045S1T3 ≈ A000045S1T8 corresponds to the following well-known identity for the Fibonacci numbers:

$$\sum_{k=0}^{n} F_n^2 = F_n F_{n+1} \tag{15}$$

Another example is the linear match A000045S1T1 ∼ A000071S1T1, which corresponds to the trivial identity

$$a_n = b_n + 1 \tag{16}$$

where $a_n = F_n$ is the Fibonacci sequence (A000045) and $b_n = F_n - 1$ (A000071). Here, both sequences are essentially the same, differing only by 1. One of our goals in the future is to improve our matching algorithm so that it ignores such trivial matches. Fortunately, we were still able find some matches that appear to be new and interesting. Thus, we conclude our paper by presenting ten such matches in the next section.

4 Ten Experimental Conjectures

In this section we present a sample of ten experimental conjectures, corresponding to linear matches, that were found through our search. Some of these conjectures we believe are new and sufficiently interesting that they deserve further study, suitable as student research projects. Other conjectures serve to illustrate how false matches can arise and be salvaged. All linear matches described below, except those found through false matches, can be found on the Eureka website [4] using its search engine.

CONJECTURE 1: (A002212S1T15 ∼ A032908S1T1, $L_{\max} = 10$, $d_r = 0.43$)

$$\det[(a_{i+j})_{i,j=0}^{n}] = b_{n+1} - 1 \tag{17}$$

where

a_n = A002212 - Number of restricted hexagonal polyominoes with n cells.

b_n = A032908 - One of 4 3rd-order recurring sequences for which the first derived sequence and the Galois transformed sequence coincide.

CONJECTURE 2: (A004441S1T12 ∼ A065619S1T7, $L_{\max} = 21$, $d_r = 0.45$)

$$\sum_{k=0}^{n} \binom{n}{k} E_{n-k} a_k = \sum_{k=0}^{n} \binom{n}{k} b_k \tag{18}$$

where E_n are the zig-zag numbers with generating function $\sec x + \tan x$ and
$a_n =$ A004441 - Numbers that are not the sum of 4 distinct nonzero squares.
$b_n =$ A065619 - E.g.f. $x(\tan(x) + \sec(x))$.

CONJECTURE 3: (A008410S1T17 $\sim$ A022523S1T2, $L_{\max} = 16$, $d_r = 0.16$)

$$\sum_{d|n} \mu(n/d) a_d = 480 \sum_{k=0}^{n-1} b_k \tag{19}$$

where
$a_n =$ A008410 - a(0) = 1, $a(n) = 480\sigma_7(n)$.
$b_n =$ A022523 - Nexus numbers $(n+1)^7 - n^7$.

CONJECTURE 4: (A026375S1T5 $\sim$ A144180S1T10, $L_{\max} = 17$, $d_r = 0.11$)

$$\sum_{k=0}^{n} a_k a_{n-k} = \frac{5}{4} \sum_{k=0}^{n} s(n,k) b_k - \frac{1}{4} \tag{20}$$

where
$a_n =$A026375 - $a(n) = \sum_{k=0}^{n} \binom{n}{k}\binom{2k}{k}$.
$b_n =$ A144180 - Number of ways of placing n labeled balls into n unlabeled (but 5-colored) boxes.

CONJECTURE 5: (A037164S1T17 $\sim$ A022527S1T2, $L_{\max} = 11$, $d_r = 0.19$)

$$\sum_{d|n} \mu(n/d) a_d = \sum_{k=0}^{n-1} b_k \tag{21}$$

where
$a_n =$ A037164 - Numerators of coefficients of Eisenstein series $E_12(q)$ (or $E_6(q)$ or $E_24(q)$).
$b_n =$ A022527 - Nexus numbers $(n+1)^{11} - n^{11}$.

CONJECTURE 6: (A151821S1T3 $\sim$ A018903S1T9, $L_{\max} = 16$, $d_r = 0.34694$)

$$\sum_{k=0}^{n-1} a_k^2 = \frac{b_n b_{n+2} - b_{n+1}^2 - 13}{3} \tag{22}$$

where
$a_n =$A151821 - Powers of 2, omitting 2 itself.
$b_n =$A018903 - Define the sequence $S(a_0, a_1)$ by a_{n+2} is the least integer such that $a_{n+2}/a_{n+1} > a_{n+1}/a_n$ for $n >= 0$. This is $S(1,5)$.

We note that this conjecture originated from a false match between A018903 and the following sequence:

a_n = A046055 - Orders of finite Abelian groups having the incrementally largest numbers of nonisomorphic forms (A046054).

CONJECTURE 7: (A098411S1T15 ~ A002416S1T1)

$$\det[(a_{i+j})_{i,j=0}^{n}] = \frac{1}{2}b_{n+1} = 2^{(n+1)^2-1} \tag{23}$$

where
a_n = A098411 - Expansion of $1/(\sqrt{1-4x}\cdot\sqrt{1-12x})$.
b_n = A002416 - 2^{n^2}.

This conjecture was found as a result of a false match (A098411S1T15 ~ A139685S1T8, $L_{\max} = 8$, $d_r = 0.16$)

$$\det[(a_{i+j})_{i,j=0}^{n}] = \frac{1}{2}c_n c_{n+1} \tag{24}$$

where
c_n = A139685 - Number of n x n symmetric binary matrices with no row sum greater than 9.

We note that the OEIS entry for A098411 mentions the following conjecture (due to R. J. Mathar):

$$na_n + 8(1-2n)a_{n-1} + 48(n-1)a_{n-2} = 0 \tag{25}$$

CONJECTURE 8: (A122162S1T17 ~ A008384S1T2, $L_{\max} = 26$, $d_r = 0.05$)

$$\sum_{d|n} \mu(n/d)a_d = \sum_{k=0}^{n-1} b_k \tag{26}$$

where
a_n = A122162 - Coefficient of q-series for constant term of Tate curve.
b_n = A008384 - Crystal ball sequence for A_4 lattice.

CONJECTURE 9: (A170762S1T7 ~ A152262S1T9)

$$\sum_{k=0}^{n} \binom{n}{k} a_k = \frac{43}{252}(b_n b_{n+2} - b_{n+1}^2) - \frac{1}{42} \tag{27}$$

where
a_n =A170762 - G.f.: (1+x)/(1-42*x).
b_n = A152262 - $a(n) = 14 * a(n-1) - 43 * a(n-2)$, $n > 1$; $a(0) = 1$, $a(1) = 7$.

This conjecture was again found from a false match between A152262S1T9 and A169344S1T7 ($L_{\max} = 13$, $d_r = 0.07$), where

a_n =A169344 - Number of reduced words of length n in Coxeter group on 43 generators S_i with relations $(S_i)^2 = (S_iS_j)^{30} = I$.

The reason for the false match is that the terms of A169344 and A170762 agree up to the first 29 terms (and then disagree starting at the 30th term). Since the OEIS stores only the first 15 terms of the A169344, the two sequences appear the same.

CONJECTURE 10:

$$a_n = \sum_{k=0}^{\infty} a(k)b(n-25k) \tag{28}$$

where

a_n = A000009 - Expansion of $\prod_{m=1}^{\infty}(1+x^m)$; number of partitions of n into distinct parts; number of partitions of n into odd parts.

b_n =A034320 - McKay-Thompson series of class 50a for the Monster group with a(0) = 1.

This conjecture arose from a false match between the two transformations A000009S1T3 and A058703S1T3 ($L_{\max} = 22$, $d_r = 0.41$), namely

$$\sum_{k=0}^{n} a_n^2 = \sum_{k=0}^{n} c_n^2 \quad \text{(false)}$$

where a_n and c_n denote sequences corresponding to the entries

A000009 = {1, 1, 1, 2, 2, 3, 4, 5, 6, 8, 10, ..., 89, 104, 122, 142, 165, 192, ..., 5718}
A058703 = {1, 0, 1, 2, 2, 3, 4, 5, 6, 8, 10, ..., 89, 104, 122}

respectively, and

c_n = A058703 - McKay-Thompson series of class 50a for Monster.

This initially led us to believe that the two sequences a_n and c_n were perhaps identical since all terms of A058703 except for the second term overlap with those of A000009. However, this is not true as the two sequences differ after the term 122. To find additional terms of A058703, we looked at those given for A034320, namely

A034320 = {1, 1, 1, 2, 2, 3, 4, 5, 6, 8, 10, ..., 89, 104, 122, 141, 164, 191, ..., 6082}

which is essentially the same sequence as A058703 but whose second term now matches with that of A000009, i.e. $b_n = c_n$ except for $n = 0$. An analysis of the differences $a_n - b_n$ between the terms of A000009 and A034320 then led us to conjecture the convolution formula (28).

5 Acknowledgements

The authors thank Drs. Anthony Breitzman and Gabriela Hristescu at Rowan University for their helpful suggestions and feedback about our work. We also thank Fritz Mineus, an

undergraduate student at Rowan, for his help in developing the Eureka website [4]. This work was supported in part by a Non-Salary Financial Support Grant from Rowan University.

6 Appendix

6.1 A. List of Transformations

Symbol (Txx)	Transformation Name	Formula
T1	Identity	a_n
T2	Partial Sums	$\sum_{k=0}^{n} a_k$
T3	Partial Sums of Squares	$\sum_{k=0}^{n} a_k^2$
T4	Inverse Binomial Transform	$\sum_{k=0}^{n} (-1)^k \binom{n}{k} a_k$
T5	Self-Convolution	$\sum_{k=0}^{n} a_k a_{n-k}$
T6	Linear Weighted Partial Sums	$\sum_{k=0}^{n} k a_k$
T7	Binomial	$\sum_{k=0}^{n} \binom{n}{k} a_k$
T8	Product of Two Consecutive Elements	$a_n a_{n+1}$
T9	Cassini	$a_n a_{n+2} - a_{n+1}^2$
T10	First Stirling	$\sum_{k=0}^{n} s(n,k) a_k$
T11	Second Stirling	$\sum_{k=0}^{n} S(n,k) a_k$
T12	Boustrophedon	$\sum_{k=0}^{n} \binom{n}{k} E_{n-k} a_k$
T13	First Differences	$a_{n+1} - a_n$
T14	Catalan	$\sum_{k=0}^{n} \frac{k}{n} \binom{2n-k-1}{n-k} a_k$
T15	Hankel	$\det(a_{i+j})_{i,j=0}^{n}$
T16	Sum of Divisors	$\sum_{d\|n} a_d$
T17	Moebius	$\sum_{d\|n} \mu(n/d) a_d$

Table 5: List of Transformations of a_n

6.2 B. Pseudocode for Computing Maximum HBT Overlap $L_{\max}$

HTB[$\{a(n)\}$,$\{b(n)\}$]:
1. Input sequences $\{a(n)\}$,$\{b(n)\}$;
2. M = Length[$\{a(n)\}$];
   N = Length[$\{b(n)\}$];
3. $\{p(k)\}$ = positions of $a(M)$ in $\{b(n)\}$ (decreasing order);
   $\{q(k)\}$ = positions of $b(N)$ in $\{a(n)\}$ (decreasing order);
4. P = Length($\{p(k)\}$);
   Q = Length($\{q(k)\}$);
5. $i = 0$; $k = 1$; $m = 0$;
6. While $i = 0$ and $k \leq P$;

7. $m = min(M, p(k))$;
8. If $\{a(M-m), a(M-m+1), ..., a(M)\} = \{b(p(k)-m), b(p(k)-m+1), ..., b(p(k))\}$, then $i = 1$;
9. k++;
10. $i = 0$; $k = 1$; $n = 0$;
11. While $i = 0$ and $k \leq Q$;
12. $n = min(N, q(k))$;
13. If $\{b(N-n), b(N-n+1, ..., b(N)\} = \{a(q(k)-n), a(q(k)-n+1), ..., a(q(k))\}$, then $i = 1$;
14. k++;
15. $L_{\max} = max(m, n)$;

References

[1] J.-P. Allouche and J. Shallit, *The ring of k-regular sequences*, II, Theor. Comput. Sci. 307 (2003), 3-29.

[2] F. Bergeron and S. Plouffe, *Computing the Generating Function of a Series Given Its First Few Terms*, Experiment. Math. 1 (1992), No. 4, 307-312.

[3] S. Colton, *Mathematics - A New Domain for Dataming?*, Proceedings of the IJCAI-01 Workshop on Knowledge Discovery from Distributed, Dynamic, Heterogenous, Autonomous Sources, Seattle, US, 2001.

[4] Eureka, http://elvis.rowan.edu/datamining/eureka.

[5] P. Liu, *Efficient Recognition of Integer Sequences*, Master's Thesis (1994), Univ. Waterloo, Canada.

[6] Mathematica (Version 8.0), Wolfram Research, http://wolfram.com.

[7] The Online Encyclopedia of Integer Sequences (OEIS), http://oeis.org.

[8] Superseeker, The Online Encyclopedia of Integer Sequences (OEIS), http://oeis.org/demos.html.

2010 *Mathematics Subject Classification*: Primary 11Y55, 68R99.
Keywords: experimental math, matching integer sequences, Online Encyclopedia of Integer Sequences.