Angles of the Cookie Monster Problem

Leigh Marie Braswell

under the direction of
Dr. Tanya Khovanova and Mr. Benjamin Iriarte

Massachusetts Institute of Technology
Research Science Institute
January 15, 2014

Abstract

The Cookie Monster Problem supposes that the Cookie Monster wants to empty a set S of jars filled with various numbers of cookies. On each of his moves, he may choose any subset of jars and take the same number of cookies from each of those jars. The *Cookie Monster number* of S, $\mathrm{CM}(S)$, is the minimum number of moves the Cookie Monster must use to empty all of the jars. We explicitly find the Cookie Monster number for jars containing cookies in n-nacci sequences. We also construct sequences of k jars such that their Cookie Monster numbers are asymptotically rk, where r is any real number: $0 \leq r \leq 1$. We find that for a set S of m jars containing $S = \{s_1, s_2, \ldots, s_m\}$ cookies to have $\mathrm{CM}(S) < m$, the set S must satisfy some equation of the form

$$\sum_{i=1}^{m} a_i s_i = 0$$

where $a_i \in \mathbb{Z}$. By modeling the problem with matrices, we recursively compute the equations which describe $\mathrm{CM}(S) = n$ where S is any cookie sequence of length $n + 1$. We bound the number of these equations and describe some of their coefficients using hyperplanes. We find that using established techniques to determine whether a set S of m jars has $\mathrm{CM}(S) < m$ is NP-hard. We lastly analyze a generating function and an algebra that models the Cookie Monster Problem.

1. Introduction

In 2002, the Cookie Monster appeared in *The Inquisitive Problem Solver* [5]. The hungry monster wants to empty a set of jars filled with various numbers of cookies. On each of his moves, he may choose any subset of jars and take the same number of cookies from each of those jars. The *Cookie Monster number* is the minimum number of moves the Cookie Monster must use to empty all of the jars. This number depends on the initial distribution of cookies in the jars. We consider the set of m jars containing $S = \{s_1, s_2, \ldots, s_m\}$ cookies. We call $s_1, s_2, \ldots, s_m$ a *cookie sequence*. Suppose the *Cookie Monster number* of S, which we denote $\text{CM}(S)$, is n. On move j for $j = 1, 2, \ldots, n$ the Cookie Monster removes x_j cookies from every jar that belongs to some subset of the jars. We call each x_j a *move amount*. Each jar can be represented as a sum of move amounts. Michael Cavers [4] first presented the formalization of Cookie Monster Problem. Both he and O. Bernardi and T. Khovanova [3] found that $\text{CM}(S) \geq \lceil \log_2(m+1) \rceil$ for a set of m non-empty, distinct jars.

In Section 2 of this paper, we explicitly find the Cookie Monster number of sets of jars containing cookies in the n-nacci sequences. We also construct sequences of k jars such that their Cookie Monster numbers are asymptotically rk, where r is any real nonnegative number less than 1. We look at the Cookie Monster Problem from a variety of angles to study the conditions a set S of m jars must satisfy to have $\text{CM}(S) = n$. The cases $n \in \{1, 2, 3\}$ were listed by Megan Belzner [2]. For example, when $\text{CM}(S) = 2$, S may contain any set of two distinct jars or any set of three distinct jars where one jar is the sum of the other two. In Section 3, we prove that the set of cookie sequences of length m that can be eaten in at most $m-1$ moves is contained in a finite union of hyperplanes defined by equations $\sum_{i=1}^{m} a_i s_i = 0$, where $a_i \in \mathbb{Z}$. We therefore model the problem as a finite set of $n \times m$ matroids whose columns represent the jars in S. In Section 4, we generate the equations for a set S of $n+2$ jars to have $\text{CM}(S) = n+1$ from the equations for a set S of $n+1$ jars to have $\text{CM}(S) = n$.

In Section 5, we study the hyperplane arrangements that describe the conditions for $\text{CM}(S) < m$ for all sets S of m jars. We find a lower bound on the number of distinct hyperplane equations and a condition some of their coefficients always satisfy. We prove that computing and checking these equations to determine whether a set S of m jars has $\text{CM}(S) < m$ is NP-hard. In Section 6, we define the Cookie Monster's generating function $F_m(k, n)$, the number of cookie sequences of length m with k total cookies which can be eaten in at most n moves. We fix m and n and study the behavior of $F_m(k,n)$ as a function of k. We show that $F_m(n,k)$ is approximately $C(m,n)\,k^n$, where $C(m,n)$ is some constant. For $F_m(m,k)$ we find that $C(m,m) \sim \frac{1}{m!}$. We find $C(m,m-1)$ approaches zero as k gets large by visualizing the hyperplane arrangements described in Section 5.

In Section 7, we use the polynomial algebra $A = k[x_1,\ldots,x_m]$ over any base ring or field to model the Cookie Monster Problem. We define the multivariable Hilbert series of the Cookie Monster algebra. The coarsening of this Hilbert Series is directly related to our work in earlier sections, as the coefficients count the number of cookie sequences containing a particular total number of cookies with a particular Cookie Monster Number. Additionally, we impose another condition on the Cookie Monster Problem. For any positive integer r, we define the r-dieting Cookie Monster as a monster that is not allowed to take more than r cookies from a single jar each move.

2. Naccis and Beyond

We now present our Cookie Monster with interesting sequences of cookies in his jars. First, we challenge our monster to empty a set of jars containing cookies in the Fibonacci sequence. The Fibonacci sequence is defined as $F_0 = 0$, $F_1 = 1$, and $F_i = F_{i-2} + F_{i-1}$ for $i \geq 2$. A jar with 0 cookies and 2 jars containing 1 cookie are irrelevant, so our smallest jar will contain F_2 cookies. Belzner [2] found that the set S of k jars containing $\{F_2, F_3, \ldots, F_{k+1}\}$ cookies has $\mathrm{CM}(S) = \lfloor \frac{k}{2} \rfloor + 1$.

There exist lesser-known and perhaps more challenging sequences of numbers similar to Fibonacci called n-nacci [1]. We define the n-nacci sequence as $N_i = 0$ for $0 \leq i < n-1$, $N_i = 1$ for $n-1 \leq i \leq n$, and $N_i = N_{i-n} + N_{i-n+1} + \cdots + N_{i-1}$ for $i \geq n$. The main property of the n-nacci sequence, like the Fibonacci sequence, is that the next term is the sum of previous terms. We can use this fact to make a strategy for emptying jars with n-nacci numbers.

Here is the Cookie Monster's strategy for dealing with n-nacci sequences, which we call *cookie-monster-knows-addition*: He takes $n-1$ moves to empty the $(k-i)$-th largest jar and reduce the k-th largest jar for each i such that $0 < i < n$. In doing this, n jars are emptied in $n-1$ moves. This process can be repeated, until at most n elements remain, which he empties one by one. Thus, when $S = \{N_n, \ldots, N_{n+k-1}\}$, we will prove that the

Cookie Monster number

$$\mathrm{CM}(S) = \left\lfloor \frac{(n-1)k}{n} \right\rfloor + 1.$$

We first prove a lemma and the necessary inequalities relating the n-nacci numbers.

Lemma 1. *Let S be a set of k jars with $s_1 < s_2 < \ldots < s_k$ cookies. If $s_i > \Sigma_{k=1}^{i-1} s_k$ for any $i > 1$, then* $\mathrm{CM}(S) = k$.

Proof. As the largest jar has more cookies than all the other jars together, any strategy has to include a step in which the Cookie Monster takes cookies from the largest jar. The Cookie Monster will not jeopardize the strategy if he takes all the cookies from the largest jar on the first move. Applying the induction process, we see that we need at least n moves. □

Lemma 2. *The n-nacci sequence satisfies the inequality:*

$$N_{k+1} > \sum_{i=1}^{k-1} N_i.$$

Proof. The proof is the same as that of well-known inequality [6] for Fibonacci numbers. □

Theorem 3. *The n-nacci sequence satisfies the inequality for any $0 \leq j \leq n-2$:*

$$N_{k+j} - \sum_{i=k+1}^{k+j-1} N_i > \sum_{i=1}^{k-1} N_i.$$

Proof. By the definition, $N_{k+j} - \sum\limits_{i=k+1}^{k+j-1} N_i = \sum\limits_{i=k+j-n}^{k} N_i$. By the inequality in Lemma 2,

$$\sum_{i=k+j-n}^{k} N_i = N_k + \sum_{i=k+j-n}^{k-1} N_i > \sum_{i=1}^{k-2} N_i + N_{k-1} = \sum_{i=1}^{k-1} N_i.$$

□

Theorem 4. *When k jars contain a set of n-nacci numbers $S = \{N_n, \ldots, N_{n+k-1}\}$, the Cookie Monster number is:*

$$\mathrm{CM}(S) = \left\lfloor \frac{(n-1)k}{n} \right\rfloor + 1.$$

Proof. Consider the largest n jars. The largest $n-1$ jars each have more cookies than the remaining $k-n$ jars do in total. That means the Cookie Monster must perform a move that includes the largest $n-1$ jars and does not touch the smallest $k-n$ jars. Suppose he touches the $(n-1)$-th largest jar on his first move. After that, even if he took cookies from the largest $n-2$ jars on his previous move, the $(n-2)$-th largest jar will still have more cookies than all of the smallest $k-n$ jars combined (due to inequalities in Theorem 3). That means there must be a move that touches the $(n-2)$-th largest jar and does not touch the smallest $k-n$ jars. Continuing this, there should be a move that touches the $(n-3)$-th largest jar and does not touch the smallest $k-n$ jars, and so on.

Summing up for every jar among the $n-1$ largest jars, there are moves that touch each jar and possibly the jars larger than it. Hence, there must be at least $n-1$ moves that do not touch the smallest $k-n$ jars. We know that we can empty the largest n jars in $n-1$ moves if the Cookie Monster uses his *cookie-monster-knows-addition* strategy. Because this strategy empties $n-1$ jars in n moves, the Cookie Monster may optimally continue in this way. Thus, because at least $n-1$ moves are needed to touch and discard the last n jars, discarding all n jars in $n-1$ moves is optimal.

We can continue doing this until we have no more than n jars left. Because the smallest n jars in set S are powers of two, we must empty these jars one by one. If k has nonzero remainder x modulo n, the Cookie Monster needs x additional moves for the last jars. Hence, the total number of moves is $(n-1)\lfloor k/n \rfloor + x$. If k has remainder 0 modulo n, he needs n additional moves to empty the final n jars for a total of $(n-1)\lfloor k/n \rfloor + 1$ moves. Therefore, for any k, we save one move for every group of n jars besides the last n jars. Hence, we save $\left\lfloor \frac{k-1}{n} \right\rfloor$ moves, and the Cookie Monster number of S is:

$$\mathrm{CM}(S) = k - \left\lfloor \frac{k-1}{n} \right\rfloor = \left\lfloor \frac{(n-1)k}{n} \right\rfloor + 1.$$

□

We found sequences representing k jars such that their Cookie Monster numbers are asymptotically rk, where r is a rational number of the form $(n-1)/n$. Is it possible to invent other sequences whose Cookie Monster numbers are asymptotically rk, where r is any rational number not exceeding 1? Before discussing sequences and their asymptotic behavior, we go back to the bounds on the Cookie Monster number of a set and check if any value between the bounds is achieved. A set $S = \{s_1, s_2, \ldots, s_k\}$ of increasing numbers s_i is called *basic* if it contains all the powers of 2 not exceeding $\max(S) = s_k$. We can calculate the Cookie Monster number of a basic set:

Lemma 5. *Given a basic set $S = \{s_1, s_2, \ldots, s_k\}$, its Cookie Monster number is the smallest power of two not in S:* $\mathrm{CM}(S) = \lfloor \log_2 s_k \rfloor + 1$.

Proof. Let m be the smallest power of 2 not in S: $m = \lfloor \log_2 s_k \rfloor + 1$. Then S contains a subset of powers of 2, namely $S' = \{2^0, 2^1, \ldots, 2^{m-1}\}$. This subset has a Cookie Monster number m. A superset of S' cannot have a smaller Cookie Monster number, so $\mathrm{CM}(S) \geq m$. On the other hand, all numbers in S can be represented as the sum of a subset of S'. Thus, all jars in S can be emptied together with the jars in S', and $\mathrm{CM}(S) = m$. □

Theorem 6. *For any k and m such that $m \leq k < 2^m$, there exist a set S of jars of length k such that* $\mathrm{CM}(S) = m$.

Proof. The given constraint allows us to build a basic set S of length k such that $2^{m-1} \leq s_k < 2^m$. This basic set satisfies the condition. □

Suppose $s_1, s_2, \ldots$ is an infinite increasing sequence. Let us denote the set of first k elements of this sequence as S_k. We are interested in the ratio of $\mathrm{CM}(S_k)/k$ and its asymptotic behaviour. If $s_i = 2^{i-1}$, then $\mathrm{CM}(S_k)/k = 1$. If $s_i = i$, then $\mathrm{CM}(S_k)/k = (\lfloor \log_2 k \rfloor + 1)/k$, which tends to zero when i tends to infinity. We know that for Fibonacci numbers the ratio is $1/2$, for Tribonacci it is $2/3$, and for n-naccis it is $(n-1)/n$. Are other ratios possible?

Yes, we claim that any ratio $r : 0 \leq r \leq 1$ is possible. We will prove this by constructing sequences with any given r. The idea is to take a sequence that contains all the powers of 2 and to add some numbers to the sequence as needed. Let us first construct the sequence explicitly. We build the sequence by induction. We start with $s_1 = 1$. Then $\mathrm{CM}(S_1)/1 = 1 \geq r$. We process natural numbers one by one and decide whether to add a number to the sequence by the following rules:

- If it is a power of 2 we always add it.
- If it is not a power of 2 we add it if the ratio does not go below r.

Now we would like to study the sequence and prove some lemmas regarding it. Let us denote the elements of this sequence by s_i, its partial sums by $S_k = \{s_1, s_2, \ldots, s_k\}$, and the ratio $CM(S_k)/k$, by r_k. We need to prove that $\lim_{k\to\infty} r_k = r$.

Suppose $CM(S_k) = m$ so that the current ratio r_k is m/k. If s_{k+1} is a power of two, then $r_{k+1} = (m+1)/(k+1)$ and the difference $r_{k+1} - r_k = (k-m)/k(k+1) < 1/(k+1)$. In this case the ratio does not decrease, but the increases are guaranteed to be smaller and smaller as k grows. If s_{k+1} is not a power of two, then $r_{k+1} = m/(k+1)$ and the difference $r_{k+1} - r_k = -m/k(k+1)$. In this case the ratio always decreases.

Lemma 7. *If $r = 1$ then the sequence contains only powers of 2. If $r = 0$, then the sequence contains all the natural numbers.*

Proof. We start with the ratio 1 for the first term of the sequence. Every non-power of 2 decreases the ratio. So if $r = 1$, we cannot include non-powers of 2. Otherwise, the ratio is always positive, so we include every non-power of 2 as long as the ratio does not go below r. □

The sequences in the previous lemma produce the ratios 0 and 1, so from now on we can assume that $0 < r < 1$. Let us see what happens if we include all numbers between two consecutive powers of 2 in the sequence. Because all powers of 2 are present in the sequence, let us denote the index of 2^m in the sequence by k_m. Hence, $CM(S_k) = m$ if $k_{m-1} \leq k < k_m$. Also, $r_{k_m} = (m+1)/k_m$.

Lemma 8. *If all the non-powers of 2 are included in the sequence between k_m and k_{m+1}, then the ratio difference is bounded: $r_{k_{m+1}}/r_{k_m} \leq (m+2)/2(m+1)$.*

Proof. Suppose by the algorithm we need to add all the numbers between k_m and k_{m+1} to the sequence. Therefore, $k_{m+1} = k_m + 2^m$. The ratios are then $r_{k_m} = (m+1)/k_m$ and $r_{k_{m+1}} = (m+2)/(k_m+2^m)$. So the ratio of ratios is $r_{k_{m+1}}/r_{k_m} = (m+2)/(m+1) \cdot k_m/(k_m+2^m)$. Using the fact that $k_m \leq 2^m$, we get $r_{k_{m+1}}/r_{k_m} \leq (m+2)/2(m+1)$. So as m grows, the ratio is almost halved. Starting from $m = 3$, we can guarantee that this ratio is never more than 2/3. □

Corollary 9. *If all of the non-powers of 2 are included for $m > 2$, then the ratios $r_{k_{m+1}}/r_{k_m} <$ 2/3.*

Theorem 10. *For any real number $r : 0 \leq r \leq 1$, there exists a sequence s_i with partial sums $S_k = \{s_1, s_2, \ldots, s_k\}$ that have Cookie Monster numbers such that CM$(S_k)/k$ tends to r when k tends to infinity.*

Proof. As we mentioned before, we can assume that $0 < r < 1$. While building the sequence, if we need to skip the next number, we have approached r within $m/k(k+1)$. That is,

$$r \leq r_k \leq r + \frac{m}{k(k+1)} \leq r + \frac{1}{k+1}.$$

If our sequence contains all but a finite amount of natural numbers, the partial ratio will tend to zero. Because the ratio should never go below r, we get a contradiction. Hence, we must drop infinitely many numbers. Each time we drop a number, the partial ratio gets within $1/(k+1)$ of r. Therefore, with each number dropped we get closer and closer to r. Now we must prove that after we get close to r we never wonder off too far from it.

Take ϵ such that $\epsilon < r/6$, and consider k such that $1/(1+k) < \epsilon$. We can find a number t such that $t > k$ and $r_t < \epsilon$. Thus, we have approached r within the distance of ϵ, and we continue building the sequence. If the next number is a non-power of 2, then the ratio approaches r. When we reach the next power of 2, then the ratio increases by no more than ϵ. Therefore, the ratio stays within 2ϵ, so it will not exceed $4r/3$.

We claim that after this power of 2 we cannot add all non-powers of 2 until the next power of 2. Indeed, if that were the case, then the ratio would drop to a number below $4r/3 \cdot 2/3 < r$. Therefore, we will have to drop a non-power of 2 from the sequence after the first encountered power of 2. We will then approach the ratio again and get at least ϵ-close to it. Thus, for numbers greater than t, the ratio will never be more than 2ϵ away from r. □

3. Modeling the Problem with Matrices

After we digest the statement of the general Cookie Monster Problem, we naturally wonder which sequences of cookies can be eaten in a given number of moves. What special property must a set of m jars have to be emptied in less than m moves? When we have a set S of m jars with $\mathrm{CM}(S) = n$, we can model the Cookie Monster Problem as an $n \times m$ binary matrix $\mathbf{A}$. Each column of the matrix represents one of the m jars, and each row represents one of the n move amounts in a particular optimal choice of n moves that empty the jars. If a jar is reduced by a move amount, the entry in the intersection of the column representing that jar with the row representing that move amount is 1. Otherwise, the entry is 0. This matrix model is illustrated in Example 21.

Lemma 11. *All sets of cookie sequences of length m that can be eaten in less than m moves are defined by equations in the form*

$$\sum_{i=1}^{m} a_i s_i = 0, \tag{1}$$

where $a_i \in \mathbb{Z}$.

Proof. Let S be any cookie sequence $\{s_1, s_2, \ldots, s_m\}$ such that CM(S) $= n = m-1 < m$, and let $\mathbf{A}$ be its corresponding matrix. We know that $\mathbf{A}$ is a $n \times m$ binary matrix representing one of the optimal strategies to empty S. We make S the $n+1$-th row of $\mathbf{A}$. Because the first n rows of $\mathbf{A}$ span a vector space, S can be written as a linear combination of the previous rows. Therefore, the determinant of $\mathbf{A}$ is zero. Equivalently,

$$\sum_{i=1}^{m} M_{m,i} s_i = 0, \tag{2}$$

where $M_{i,j}$ denotes the minor of the entry in the i-th row and j-th column. All maximal minors of the $(n+1) \times m$ binary matrix $\mathbf{A}$ are the coefficients of a linear equation relating the elements of the cookie sequence. □

Computing the equations of the form (1) for any set S of m jars with CM(S) $= m-1 < m$ is equivalent to finding all matrices $\mathbf{A}$ of optimal move amount distributions and computing the kernel of these matrices. A coefficient vector $\mathbf{a} = (a_1, a_2, \ldots, a_m)$ which represents the coefficients of equation (1) is any element of the union of these null spaces. To find these coefficient vectors, we perform Gaussian elimination on a matrix $\mathbf{A}$ so that we obtain a matrix of the form $(\ \mathbf{I} \mid \mathbf{C}\ )$ where $\mathbf{I}$ is the $n \times n$ identity matrix. The $m-n$ columns of $\mathbf{A}$ which make up $\mathbf{C}$ may then be written in terms of the first n columns. If $m - n > 1$, an equation of the form (1) is found for each of the $m - n$ columns. Finding this equation for the set of jars modeled in Example 21 is shown in Example 22 in Appendix A. The equation relating the jars in cookie sequences of length m that can be eaten in less than m moves is dependent upon the matrix of move amounts $\mathbf{A}$. Two optimal strategies, or different choices of a set of move amounts with the smallest size, will yield different equations as shown in Example 23 in Appendix A.

$$
\begin{array}{ccc}
W & & Y \\
\begin{pmatrix}
1 & 0 & \dots & 0 & a_{1m} \\
0 & 1 & \dots & 0 & a_{2m} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \dots & 1 & a_{nm}
\end{pmatrix}
& \Longrightarrow &
\begin{pmatrix}
1 & 0 & \dots & 0 & b_1 & a_{1m} \\
0 & 1 & \dots & 0 & b_2 & a_{2m} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & \dots & 1 & b_n & a_{nm} \\
c_1 & c_2 & \dots & c_n & 1 & d
\end{pmatrix}
\end{array}
$$

FIGURE 1. Building a New Cookie Sequence

4. RECURSIVELY COMPUTING CM(S) EQUATIONS

To generalize all sets of jars with a particular Cookie Monster number, we define S_n to be any set of jars with $\mathrm{CM}(S_n) = n$. We can compute the equations which describe $\mathrm{CM}(S_{n+1}) = n + 1$ where S_{n+1} is any cookie sequence of length $n + 2$ from those which describe $\mathrm{CM}(S_n) = n$ where S_n is any cookie sequence of length $n + 1$. All combinations of the equations which describe $\mathrm{CM}(S_{n+1}) = n + 1$ for S_{n+1} of length $n + 2$ give the equations that cookie sequences S_{n+1} of any length with $\mathrm{CM}(S_{n+1}) = n+1$ satisfy. In cookie sequences of length greater than $n+2$, each jar s_i with $i > n+2$ satisfies one of the generated equations. As a result, cookie sequences S_{n+1} with $\mathrm{CM}(S_{n+1}) = n + 1$ of length $(n + 2) + k$ for some positive integer k will satisfy $k + 1$ of the equations we compute for S_{n+1} of length $n + 2$.

Conjecture 12. *If* $\mathrm{CM}(S_n) = n$ *where* S_n *is any cookie sequence of length* m*, the rank of its* $n \times m$ *binary matrix of move amounts* $\mathbf{A}$ *is exactly* n.

Suppose we have some generic distribution of n independent move amounts in $m = n + 1$ jars. After a possible permutation of the columns corresponding to the jars, Conjecture 12 guarantees that Gaussian elimination can be performed on the corresponding matrix $\mathbf{A}$ of full row rank to obtain the $n \times n$ identity matrix and a column vector of length n, where $(a_{1m}, a_{2m}, \dots, a_{nm})$ represents all values that such a matrix can yield. This column vector gives the coefficients of the equation the $n + 1$ jars satisfy and is shown in matrix W in Figure 1. To build the equations for $m + 1 = n + 2$ jars that give $\mathrm{CM}(S_{n+1}) = n + 1$, we add to $\mathbf{A}$ any row and column which represent the addition of a jar and move amount. We can do this because we compute a $n \times n$ identity matrix in the process of finding the $(n+1) \times (n+1)$ identity matrix for the set of $n + 2$ jars. We call the binary sequence which corresponds to the column representing the new jar $\{b_i\}$, and the binary sequence which corresponds to the row representing the new move amount $\{c_i\}$. We let the set K contain the indices i such

$$
\overset{W}{\begin{pmatrix} 1 & 0 & \dots & 0 & b_1 & a_{1m} \\ 0 & 1 & \dots & 0 & b_2 & a_{2m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & b_n & a_{nm} \\ 0 & 0 & \dots & 0 & 1-f & d - \sum\limits_{i \in L} x_i \end{pmatrix}} \Longrightarrow \overset{Y}{\begin{pmatrix} 1 & 0 & \dots & 0 & b_1 & a_{1m} \\ 0 & 1 & \dots & 0 & b_2 & a_{2m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & b_n & a_{nm} \\ 0 & 0 & \dots & 0 & 1 & z \end{pmatrix}}
$$

FIGURE 2. Row Reducing our New Matrix

that $b_i = 1$, and we let the set L contain the indices i such that $c_i = 1$. We denote the number of elements such that $b_i = c_i = 1$ by f and let d be 0 or 1. This is illustrated in matrix Y in Figure 1.

Theorem 13. *Suppose we have all of the coefficients of the equations describing when* $\mathrm{CM}(S_n) = n$ *for any set* S_n *with size* $n+1$*. For* $\mathrm{CM}(S_{n+1}) = n+1$ *where* S_{n+1} *is any set with size* $n+2$*, the coefficients of all equations are given by permutations of the set of integers* $\{k_1, k_2, \dots, k_{n+2}\}$ *described by*

$$
k_i = \begin{cases} a_i - \frac{d - \sum_{i \in L} x_i}{1-f} & \text{for } i \in K \\ a_i & \text{for } i \notin K \\ \frac{d - \sum_{i \in L} x_i}{1-f} & \text{for } i = n+2. \end{cases}
$$

Proof. To guarantee that row reduction of the matrix containing the $n \times n$ identity, $\{b_i\}$, $\{c_i\}$, $\{a_{im}\}$, and d is possible, we suppose each of the original $n+1$ jars are distinct for $n > 1$ and nonempty. If we do not assume this, the equations we calculate would have coefficients identical to equations describing a lower Cookie Monster number. Without loss of generality, we also suppose that the added jar contains the new move amount and $f \neq 1$. We can permute the coefficients to obtain all equations.

We first clear the $n+1$-th row of our matrix by subtracting from it each of the other rows with a nonzero entry in a column indexed by set L and replacing the $n+1$-th row with the difference. We obtain matrix W in Figure 2. We then divide the $n+1$-th row by $1-f$ and replace $\frac{d - \sum_{i \in L} x_i}{1-f}$ by z to obtain matrix Y in Figure 2. We clear the $n+1$-th column by subtracting the $n+1$-th row from each b_i with $i \in K$ and replacing the rows containing b_i with the difference. We obtain the $(n+1) \times (n+1)$ identity matrix and a column representing an equation describing $\text{CM}(S_{n+1}) = n+1$. The coefficients are $a_i - z$ for $i \in K$, a_i for $i \notin K$, and z for the $n+2$-th jar. $\square$

We consider an application of Theorem 13 for the $n = 1$ case in Example 24 in Appendix A. We find a lower bound on the total number of equations for $\text{CM}(S_{n+1}) = n + 1$ and $|S_{n+1}| = n + 2$ by fixing $\{a_i\}$ and $\{c_i\}$. Under these constraints, any change in $\{b_i\}$ yields a distinct equation. Thus, there are at least $2^n = 2^{m-1}$ distinct equations for $\text{CM}(S_{n+1}) = n+1$ and $|S_{n+1}| = n + 2$. Therefore, the number of equations one has to check for a sequence of m jars to determine whether $\text{CM}(S) < m$ grows at least exponentially with m.

5. NP-hardness

In Section 4, we generated the equations for $\text{CM}(S_{n+1}) = n + 1$ where S_{n+1} is any cookie sequence of length $n + 2$ from those which describe $\text{CM}(S_n) = n$ where S_n is any cookie sequence of length $n+1$. We can bound the number of these equations below more precisely by considering the hyperplanes they represent.

Theorem 14. *For $m > 2$, the number of distinct hyperplane equations for all sets S of jars of size m with $\text{CM}(S) < m$ is at least $m!$.*

Proof. Consider all $(m - 1) \times m$ binary matrices. These represent all possible move amount distributions for all cookie sequences S of length m with $\text{CM}(S) < m$. The possible rows are all points with nonzero binary coordinates in m dimensions. The extensions of the lines connecting these $2^m - 1$ points with the origin intersect the $m - 1$ dimensional hypersimplex given by fixing the sum of the m coordinates to a positive integer. For three jars with $\text{CM}(S) \leq 2$, the intersections occur at the vertices, midpoints, and centroid of the triangle where the sum of the cookies is fixed and all coordinates are positive. The case with hyperplanes for four jars with $\text{CM}(S) \leq 3$ is shown in Figure 3. The subsets of $m - 1$ of these points that do not give lower dimensional affine spaces determine the equations of all $m - 1$ dimensional hyperplanes where $\text{CM}(S) < m$. We can recursively bound the number of distinct hyperplanes for all sets S of size m with $\text{CM}(S) < m$ given that we know the number of distinct hyperplanes for all sets S of size $m - 1$ with $\text{CM}(S) < m - 1$. Suppose there exist h hyperplane equations for all sets S of jars of size $m - 1$ with $\text{CM}(S) < m - 1$. We count mh distinct hyperplanes for $\text{CM}(S) < m$ that pass through a vertex and $m - 2$

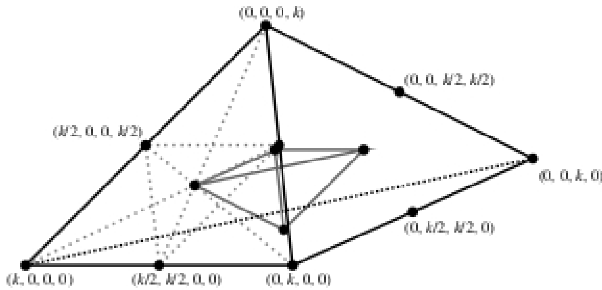

FIGURE 3. The span of three move amounts in four jars is represented by the set of nonzero binary vectors of length four. Shown here are the intersections of scalar multiples of these vectors (labeled) with a hypersimplex where the sum of cookies is a fixed positive integer.

points on the boundary of the hypersimplex, which lie on the facet opposite to the vertex and which correspond to the h hyperplanes of the previous case. This recursive bound can be made explicit by noting that there are six equations all sets S of jars of size three with $\mathrm{CM}(S) < 3$ satisfy, namely the permutations in $\{s_1, s_2, s_3\}$ of $s_1 = s_2$ and $s_1 = s_2 + s_3$. Thus, the number of distinct hyperplane equations for all sets S of jars of size m with $\mathrm{CM}(S) < m$ is at least $m!$. To find a better bound, we should count the hyperplanes that pass through $m - 1$ midpoints of edges not in the same facet of the hypersimplex and the hyperplanes which do not contain vertices or midpoints of the edges of the hypersimplex. However, these two types of hyperplanes may contain multiple subsets of $m - 1$ points, so they are more difficult to count. □

There are $2^m - 1$ nonzero binary vectors of length m that represent possible distributions of move amounts in the jars of cookie sequences with length m. These vectors correspond to vertices of the m dimensional hypercube that are visible from the $m - 1$ dimensional hypersimplex given by fixing the sum of the cookies to k. Theorem 14 illustrates how hyperplane cuts of this hypersimplex allow us to determine which cookie sequences S of length m containing a total of k cookies have $\mathrm{CM}(S) < m$. The points not lying on these cuts represent sets of m jars with $\mathrm{CM}(S) = m$.

Consider the $2^m - 1$ intersection points of the rays spanned by nonzero binary points with the hypersimplex and all affine subspaces generated by all choices of $m - 1$ of these points. The cuts are the intersections of these affine spaces with the hypersimplex. Integer lattice points lying in these cuts contain the sets of cookie sequences of length m that can be emptied in less than m moves. Moreover, for any integer ℓ with $1 < \ell \leq m$, the set of all integer lattice points in the hypersimplex contained in the affine subspaces generated by all choices of $\ell - 1$ of these intersection points contains the set of cookie sequences of length m that can be solved in less than ℓ moves.

We project these cuts onto the union of the facets of the m dimensional hypercube that are visible from the $m-1$ dimensional hypersimplex, or what we call the *visible boundary* of the m-cube. These projections are the intersections of the visible boundary of the m-cube with the linear subspaces generated by each choice of $m-1$ of its nonzero vertices and the origin. In other words, they are the intersections of the linear subspaces spanned by the $m-1$ nonzero vertices of the m-cube and the visible boundary. We prove a condition satisfied by certain hyperplanes that pass through the m-cube and correspond to cookie sequences S of length m with $\mathrm{CM}(S) < m$. We suppose $\mathbf{a} = (a_1, a_2, \ldots, a_m)$ is an integer lattice point to a hyperplane that passes through the origin, cuts the interior of the m-cube, and does not intersect an edge of the m-cube at its relative interior.

Theorem 15. *The vector* $\mathbf{a}$ *can be any vector with integer coordinates of magnitude at most* 1, *such that* $\mathbf{a}$ *contains at least one coordinate equal to 1 and at least one coordinate equal to* -1.

Proof. Suppose $\mathbf{a}$ is normal to a hyperplane that passes through 0. By assumption, this hyperplane does not pass through a midpoint of the hypercube, so there do not exist two adjacent vertices v_1, v_2 on the m dimensional hypercube such that $\mathbf{a} \cdot \mathbf{v_1} < \mathbf{0}$ and $\mathbf{a} \cdot \mathbf{v_2} > \mathbf{0}$. Without loss of generality, let $a_1 > 0$ and let $\mathbf{v_1} = (\mathbf{0}, \ldots)$ and $\mathbf{v_2} = (\mathbf{1}, \ldots)$, where the remaining entries of v_1 and v_2 are identical coordinate sequences of magnitude at most 1. We see $\mathbf{a} \cdot \mathbf{v_1}$ and $\mathbf{a} \cdot \mathbf{v_2}$ differ only by a_1, or $\mathbf{a} \cdot \mathbf{v_2} = \mathbf{a} \cdot \mathbf{v_1} + a_1 > 0$, where $\mathbf{a} \cdot \mathbf{v_1} < \mathbf{0}$. Because this cannot happen, this means that whenever there is a subset I of $\{2, \ldots, m\}$ such that $a_I = \sum_{i \in I} a_i < 0$, then $a_1 \leq |a_I|$. In particular, since at least one of the coordinates of $\mathbf{a}$ must be negative and at least one coordinate must be positive, then the absolute value of the nonzero coordinates must be the same. Thus, their magnitudes must all be 1. All such nonzero binary vectors $\mathbf{a}$ pass through the origin and the interior of the m-cube and are therefore obtainable. □

The Cookie Monster number of a set of m jars may be found using one of two methods. We may check all of the possible equations a set of m jars can satisfy if $\mathrm{CM}(S) < m$.

Alternatively, we may use a brute-force algorithm mentioned by Cavers [4]. In this algorithm, all possible sets with elements less than or equal to those in the cookie sequence are tested. All subsets of each of these sets are summed. These partial sums are checked to see if they contain all of the elements of the original cookie sequence. The algorithm stops when the smallest set with partial sums that make up each jar in the cookie sequence is found, and the magnitude of this set is the Cookie Monster number. Both of these methods involve checking many partial sums and require at least exponential time.

Theorem 16. *Determining whether a sequence S of m jars has* $\mathrm{CM}(S) < m$ *by using brute force or by checking equations is NP-hard.*

Proof. Using the brute force technique or determining the equations for CM(S) $< m$ and checking those equations is reminiscent of the NP-hard *Subset Sum Problem*. The *Subset Sum Problem* asks whether a non-empty subset of a given set of integers sums to zero. The Cookie Monster Problem asks if a non-empty subset of a given set of integers satisfies one of some computed equations. We consider any one of these equations in the form

$$\sum_{i \in A} s_i = \sum_{i \in B} s_i \tag{3}$$

found in Theorem 15 where A and B are disjoint subsets of S. We negate at least any $|B|$ elements of the cookie sequence and check whether the cookie sequence now has two subsets that satisfy (3). Therefore, a subproblem of the Cookie Monster Problem is equivalent to the *Subset Sum Problem*, and it is one of many checks necessary to determine whether $\mathrm{CM}(S) < m$ for m jars. Thus, using established techniques to determine whether a sequence of m jars has $\mathrm{CM}(S) < m$ is at least as hard as the *Subset Sum Problem* and is NP-hard. □

However, we cannot say that the Cookie Monster Problem is NP-hard in general. We do not know if considering the remaining possible equations with coefficients of magnitude greater than 1 simplifies the process of determining whether a sequence S of m jars has $\mathrm{CM}(S) < m$. For example, suppose we have a set of m integers, and we want to determine if a subset of them has a sum between a positive integer a and a negative integer b inclusive. This problem is easier than the *Subset Sum Problem*, as we can develop a faster, more selective checking strategy by considering the relative magnitude of the elements in the original set. However, in the context of the Cookie Monster Problem, we do not know if checking all equations, including those with coefficients of magnitude greater than 1, actually reduces the complexity of the problem. A technique different from brute force or equation checking could be more efficient. However, considering the randomness of the equations found in Theorem 13 for $\mathrm{CM}(S) < m$ as m increases, we conjecture that using any method to determine whether a sequence S of m jars has $\mathrm{CM}(S) < m$ is NP-hard.

6. Cookie Monster's Generating Function

We now consider the function $F_m(k, n)$, the number of cookie sequences of length m with k total cookies which can be eaten in at most n moves. We fix m and n and study the behavior of $F_m(k, n)$ as a function of k. We see that $F_m(n, k)$ is approximately $C(m, n)k^n$, where $C(m, n)$ is some constant.

Theorem 17. *For $F_m(m, k)$, the coefficient $C(m, m) \sim \frac{1}{m!}$.*

Proof. All cookie sequences of size m may be eaten in at most m moves. Therefore, $F_m(m, k)$ counts all partitions of k cookies into m jars with no jars empty. This is

$$\binom{k-1}{m-1} \sim \frac{k^m}{m!} = C(m, m)k^m.$$

□

We wonder if we may simplify the function $F_m(k, m-1) = C(m, m-1)k^m$ similarly. Each cookie sequence of length m with k total cookies which can be eaten in at most $m-1$ moves satisfies some hyperplane equation of the form (1) discussed in Theorem 14 and Theorem 15. We can bound the number of these hyperplane equations and describe their coefficients, but computing $C(m, m-1)$ is difficult. We see that as the fixed sum k of the cookies grows, the intersections of the hyperplanes describing when $\mathrm{CM}(S) < m$ for any set S of m jars with the hypersimplex become more insignificant. In other words, most cookie sequences with many cookies do not lie on one of the hyperplanes describing when $\mathrm{CM}(S) < m$ for any set S of m jars with cookie sum k. Thus, $C(m, m-1)$, the coefficient of k^m in the generating function $F_m(k, m-1)$, approaches zero as k approaches infinity.

7. Cookie Monster Algebra

We now discuss an algebraic model of the Cookie Monster Problem. We consider the polynomial algebra $A = k[x_1, \ldots, x_m]$ over any base ring or field, and we define a filtration on A, the Cookie Monster Filtration, by the condition that the degrees of the generators are 1. For each $T \subset \{1, 2, \ldots, m\}$ we define $x_T = \prod_{i \in T} x_i$. The generators of the algebra are of the form $x(n, T) = x_T^n$ where n is a positive integer. We let $gr(A)$ be the associated graded algebra, and we call it the Cookie Monster algebra. This algebra is relevant because for a cookie sequence $s_1, \ldots, s_m$ and its corresponding monomial $x_S = x_1^{s_1} \ldots x_m^{s_m}$, the degree of its image in $gr(A)$ is CM(S). Therefore, any minimal way of eating the cookies corresponds to a representation of x_S in $gr(A)$ as a product of generators. The multivariable Hilbert series of the Cookie Monster algebra is given by

$$H_m(z_1, \ldots, z_m, t) = \sum_S z_1^{s_1} \ldots z_m^{s_m} t^{\mathrm{CM}(S)}.$$

In the earlier sections, we observed how difficult it is to solve the Cookie Monster Problem in its entirety. To make the problem more tractable using algebraic techniques, we impose another condition on the Cookie Monster Problem. For any positive integer r, we define the r-dieting Cookie Monster as a monster that is not allowed to take more than r cookies from a single jar each move. We define dieting Cookie Monster algebras with generators $x(n, T)$ only for $n \leq r$.

Theorem 18. *For an r-dieting Cookie Monster emptying a set of jars S with largest element s_m,*

$$\left\lceil \frac{s_m}{r} \right\rceil \leq \mathrm{CM}(S) \leq \left\lfloor \frac{s_m}{r} \right\rfloor + \mathrm{CM}(R)$$

where $R = \{1, 2, \ldots, r-1\}$.

Proof. In $\left\lfloor \frac{s_m}{r} \right\rfloor$ moves, the Cookie Monster can reduce each jar to a number of cookies less than r. He does so by taking r cookies from every jar as many times as possible, which is bounded by the largest element in the cookie sequence. If all of the other jars contain multiples of r cookies, CM(S) $= \left\lceil \frac{s_m}{r} \right\rceil$. Otherwise, after each jar is reduced to a number of cookies less than r, the dieting Cookie Monster empties the jars as the regular Cookie Monster would. In the worst case, this means that he must empty jars containing the set of $\{1, 2, \ldots, r-1\}$ cookies. □

However, the strategy of reducing each jar to its modulus in r by taking as many multiples of r cookies as possible and then emptying the jars optimally is not always the fastest approach.

Theorem 19. *The multivariable Hilbert series of the dieting Cookie Monster algebra $H_{D,m}(z_1, ..., z_m, t)$ is a rational function.*

Proof. We consider the syzygies between the generators of the dieting Cookie Monster algebra corresponding to the r-dieting Cookie Monster. There are finitely many of these generators, so the rationality of dieting Cookie Monster algebras follows from Hilbert's syzygy theorem. We can coarsen the Hilbert series in Theorem 19, where $|S|$ is the number of cookies in S. □

$$h_{D,m}(z,t) = H_{D,m}(z, \ldots, z, t) = \sum_S z^{|S|} t^{\mathrm{CM}_D(S)}.$$

Theorem 20. *For the r-diet with a cookie sequence made only of r multiples $h_{D,m}(z,t)$ is of the form*

$$\frac{1}{\prod_{n=1}^{m}(1 - tz^{rn})^{\binom{m}{n}}}.$$

Proof. While on the r-diet, the Cookie Monster may take r cookies from any subset of the jars with at least r cookies. $\prod_{n=1}^{m}(1-tz^{rn})^{\binom{m}{n}}$ represents choosing n of the m cookie jars and taking r cookies out of those jars on a number of moves corresponding to the magnitude of t's exponent. When all jars are multiples of r, all optimal moves will be of this form. Thus, $h_{D,m}(z,t)$ may be written as

$$\frac{1}{\prod_{n=1}^{m}(1 - tz^{rn})^{\binom{m}{n}}}.$$

□

8. Acknowledgements

I would like to thank my PRIMES mentor, Dr. Tanya Khovanova of the Massachusetts Institute of Technology, for guidance throughout the project. I would also like to thank my RSI mentor, Mr. Benjamin Iriarte of the Massachusetts Institute of Technology, for his assistance with the project. I would like to thank Dr. Pavel Etingof of MIT and the RSI staff, especially my tutor Dr. John Rickert and Jacob McNamara, for invaluable advice. I am grateful for the resources and research opportunity provided by the Center for Excellence in Education, RSI, PRIMES, and the MIT Mathematics Department. I would also like to thank Dr. Tom Leighton, CEO of Akamai Technologies, and Akamai Technologies for sponsoring me this summer at RSI.

References

[1] Generalizations of the Fibonacci Numbers. Available at http://en.wikipedia.org/w/index.php?title=Generalizations_of_Fibonacci_numbers&oldid=560561954 (2013/09/14).

[2] M. Belzner. The Cookie Monster Problem. Available at http://arxiv.org/abs/1304.7508 (2013/07/02).

[3] O. Bernardi and T. Khovanova. The Cookie Monster Problem. Available at http://blog.tanyakhovanova.com/?p=325 (2013/07/02).

[4] M. Cavers. Cookie Monster Problem Notes. University of Calgary Discrete Math Seminar, 2010.

[5] P. Vaderlind, R. Guy, and L. Larson. *The Inquisitive Problem Solver*. The Mathematical Association of America, 2002.

[6] Wikipedia. Fibonacci Numbers. Available at http://en.wikipedia.org/w/index.php?title=Fibonacci_number&oldid=572271013 (2013/09/14).

Appendix A.

Example 21. Suppose we have a set of jars $S = \{s_1, s_2, s_3, s_4\}$ and a set of move amounts $\{x_1, x_2, x_3\}$. Suppose that $s_1 = x_1$, $s_2 = x_2 + x_3$, $s_3 = x_1 + x_2$, and $s_4 = x_1 + x_3$. The matrix corresponding to this distribution of the move amounts in the jars is

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Example 22. When the matrix corresponding to the distribution of move amounts in the jars in Example 21 is put in the form $(\ \mathbf{I} \mid \mathbf{C}\ )$ by Gaussian elimination we obtain

$$\begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}.$$

By writing the fourth column in terms of the first three, we obtain $s_4 = 2s_1 + s_2 - s_3$ or, of the form (1), $2s_1 + s_2 - s_3 - s_4 = 0$.

Example 23. Suppose we have a set of jars $S = \{1, 2, 1\}$. Two choices of $\mathbf{A}$ with the least number of rows possible are

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

where the set of move amounts $\{x_1, x_2\} = \{1, 1\}$. The first $\mathbf{A}$ gives the equation $s_1 = s_3$, while the second $\mathbf{A}$ gives the equation $s_2 = s_1 + s_3$. Both equations are of the form (1) but represent different optimal move sequences of the Cookie Monster.

Example 24. The equation for $\mathrm{CM}(S) = 1$ and $|S| = 2$ is $s_1 = s_2$, which corresponds to the 1×2 matrix $\left(\begin{smallmatrix} 1 & 1 \end{smallmatrix}\right)$. To find the coefficients describing all equations for $\mathrm{CM}(S) = 2$ and $|S| = 3$, we add all possible binary sequences corresponding to the addition of a nonempty jar and move amount to this matrix.

Case 1. $\{b_i\} = 0, \{c_i\} = 0$.

The matrix that represents this situation is $\left(\begin{smallmatrix} 1 & 0 & 1 \\ 0 & 1 & d \end{smallmatrix}\right)$, *which is already reduced. Setting* $d = 0$

and $d = 1$, we obtain the equations

$$s_1 = s_3 \quad \textit{and} \quad s_3 = s_1 + s_2. \tag{4}$$

Case 2. $\{b_i\} = 1, \{c_i\} = 0$.

The matrix that represents this situation is $\left(\begin{smallmatrix} 1 & 1 & 1 \\ 0 & 1 & d \end{smallmatrix}\right)$. *We reduce the matrix by replacing the first row with the positive difference between the two rows:* $\left(\begin{smallmatrix} 1 & 0 & 1-d \\ 0 & 1 & d \end{smallmatrix}\right)$. *Setting $d = 0$ and $d = 1$, we obtain the equations*

$$s_1 = s_3 \quad \textit{and} \quad s_3 = s_2. \tag{5}$$

Case 3. $\{b_i\} = 0, \{c_i\} = 1$.

The matrix that represents this situation is $\left(\begin{smallmatrix} 1 & 0 & 1 \\ 1 & 1 & d \end{smallmatrix}\right)$. *We reduce the matrix by replacing the second row with the positive difference between the two rows:* $\left(\begin{smallmatrix} 1 & 0 & 1 \\ 0 & 1 & d-1 \end{smallmatrix}\right)$. *Setting $d = 0$ and $d = 1$, we obtain the equations*

$$s_3 = s_1 - s_2 \quad \textit{and} \quad s_1 = s_3. \tag{6}$$

Case 4. $\{b_i\} = 1, \{c_i\} = 1$.

The matrix that represents this situation is $\left(\begin{smallmatrix} 1 & 1 & 1 \\ 1 & 1 & d \end{smallmatrix}\right)$. *Because we cannot reduce this case, we already have the equation for any d,*

$$s_1 = s_2. \tag{7}$$

Combining and permuting (4), (5), (6), and (7) in $\{s_1, s_2, s_3\}$, we obtain all possible equations describing $\mathrm{CM}(S) = 2$ and $|S| = 3$ from those describing $\mathrm{CM}(S) = 1$ and $|S| = 2$. These are $s_1 = s_2$, $s_1 = s_3$, $s_2 = s_3$, $s_3 = s_1 + s_2$, $s_2 = s_1 + s_3$, and $s_1 = s_2 + s_3$. These match the equations found by Belzner [2].