HKUST Theoretical Computer Science Center Research Report HKUST-TCSC-1999-12

New Upper and Lower Bounds on the Channel Capacity of Read/Write Isolated Memory*

Mordecai J. Golin† Xuerong Yong‡ Yuanping Zhang§
Li Sheng¶

January 28, 2000

Abstract

In this paper we refine upper and lower bounds for the channel capacity of a serial, binary rewritable medium in which no consecutive locations may store '1's and no consecutive locations may be altered during a single rewriting pass. This problem was originally examined by Cohn [4] who proved that C, the channel capacity of the memory, in bits per symbol per rewrite, satisfies

$$0.50913\ldots \le C \le 0.56029\ldots.$$

In this paper we show how to model the problem as a constrained two-dimensional binary matrix problem and then modify recent techniques for dealing with such matrices to derive improved bounds of

$$0.53500\ldots \le C \le 0.55209\ldots.$$

Key words: capacity, channel graph, eigenvalue, two-dimensional codes, runlength-limited codes, constrained arrays.

*The work of the first three authors was partially supported by Hong Kong CERG grants HKUST652/95E, 6082/97E and 6137/98E and DIMACS.

†Department of Computer Science, Hong Kong UST, Clear Water Bay, Kowloon, Hong Kong. *Email*: golin@cs.ust.hk.

‡Department of Computer Science, Hong Kong UST, Clear Water Bay, Kowloon, Hong Kong. *Email*: xryong@cs.ust.hk. Also at the Department of Mathematics, Xinjiang University, Urumqi, 830046, PRC.

§Department of Computer Science, Hong Kong UST, Clear Water Bay, Kowloon, Hong Kong. *Email*: ypzhang@cs.ust.hk. Also at the Department of Mathematics, Hunan Normal University, Changsha 410081, PRC.

¶Department of Mathematics and Computer Science, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104, USA. *Email*: lsheng@mcs.drexel.edu.

1 Introduction

A serial binary $(0,1)$ memory is *read isolated* if no two consecutive positions in the memory may both store 1's. This restriction occurs, for example, in some codes used in magnetic recording and optical recording. Freiman and Wyner [6] considered this problem and showed that the capacity of such a memory is $\log_2 \varphi = 0.694...$ bits per symbol, in which $\varphi = \frac{1+\sqrt{5}}{2}$ is the larger eigenvalue of the *Fibonacci* recurrence: $F_{n+2} = F_{n+1} + F_n$.

A serial binary $(0,1)$ memory that undergoes rewriting is *write isolated* if it satisfies the restriction that no two consecutive positions in the memory can be changed during one rewriting phase. Such restrictions have arisen, for example in the contexts of asymmetric error-correcting ternary codes [8] and of rewritable optical discs [3]. In this case, again, the capacity is $\log_2 \varphi$.

A *read/write isolated memory* (RWIM) is a binary, linearly ordered, rewritable storage medium that obeys both the read and write restrictions. This type of memory was considered by Cohn in [4], who examined its channel capacity. The set of all permissible binary memory configurations can be considered as a channel alphabet. The rewriting restrictions determine which characters may follow which characters in the channel. The *channel capacity* of this process can then be defined as follows [3] [11]: let k be the size of the memory in binary symbols, r the lifetime of the memory in rewrite cycles and $N(k,r)$ the number of distinct sequences of r characters. For fixed k, the *channel capacity,* measured in bits per rewrite, is defined to be [9]

$$C_k = \lim_{r\to\infty} \frac{1}{r} \log_2 N(k,r). \tag{1}$$

The channel capacity of the read/write isolated memory, in bits per symbol per rewrite, is then defined to be

$$C = \lim_{k\to\infty} \frac{1}{k} C_k. \tag{2}$$

In [4] Cohn established several expressions for the capacities C_k and derived the following upper and lower bounds on C:

$$0.50913\ldots \leq C \leq 0.56029\ldots.$$

In this paper we continue the investigation of the channel capacity and manage to refine the bounds to

$$0.53500\ldots \leq C \leq 0.55209\ldots.$$

Our approach is to no longer view the capacity as a function of the read/write process but instead as the capacity of a certain type of *constrained matrix*. This permits us to modify tools that have recently been used for counting constrained matrices and then use them to derive the new bounds.

In the next section we describe constrained matrices and transfer matrices, use constrained matrix notation to rederive some known results on the channel capacity of RWIMs and then derive a first, slightly improved, lower bound. Section 3 describes a second transfer matrix for RWIMs and uses it to derive a new upper bound. Section 4 shows how to use the maximum principle for eigenvalues to derive a better lower bound. Section 5 concludes by reviewing our results and conjecturing that $C = 0.53500\ldots$.

Note: In what follows we use the following conventions for writing vectors and matrices. If v, v' are two $1 \times k$ matrices then $\begin{pmatrix} v \\ v' \end{pmatrix}$ is the $2 \times k$ matrix whose first row is v and whose second row is v'. Also $1(0)$ will be used as shorthand for the appropriate sized vectors all of whose elements are $1s$ $(0s)$. Thus $\begin{pmatrix} v \\ 1 \end{pmatrix}$ is the $2 \times k$ matrix whose first row is v and whose second row is all 1s. Finally, if B is a $r \times k$ matrix we will use the notation $B(j, *)$ to denote the $1 \times k$ matrix that is the jth row of B. All vectors in this paper are row vectors unless otherwise explicitly indicated.

2 Constrained Matrices

We start by pointing out that there is another way of viewing the rewriting process. Suppose k, the size of the memory and r, the number of rewrites, are known. Then we can define, B, a $r \times k$ binary matrix: $\forall 1 \le i \le k$, $1 \le j \le r$,

$$B(j, i) = \text{content of location } i \text{ after the } j\text{th rewrite.}$$

Thus $B(j, *)$ is the content of the memory after the jth rewrite. Translating the RWIM rules into matrix notation shows that B satisfies the following two constraints:

1. *read restriction:* B does not contain any two horizontally consecutive ones, i.e., it does not contain any 1×2 submatrix $(1\,1)$;

$$B_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & \mathbf{0} & \mathbf{1} \\ 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{0} \end{pmatrix}.$$

Figure 1: B_1 is a good matrix. B_2 is not a good matrix for two reasons: the two consecutive bold 1's in the second row contradict the read restriction and the bold 2×2 submatrix in the lower right corner contradicts the write restriction.

2. *write restriction:* B does not contain any 2×2 submatrix of the form

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \textbf{or} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Note too that if B is *any* $r \times k$ binary matrix that obeys the two conditions above then B can be viewed as modelling a memory with $B(j, *)$, the jth row of B, being the content of the memory at time j. The memory thus modelled satisfies the read/write isolated conditions. We have therefore just seen that $N(k, r)$, previously defined as the number of distinct sequences of r characters, is also

the number of $r \times k$ binary matrices that satisfy conditions 1 and 2.

In what follows we will call matrices that satisfy conditions 1 and 2 *good* matrices. In Figure 1, B_1 is a *good* matrix and B_2 is not a *good* matrix.

The problem of finding, $f(k, r)$, the number of $r \times k$ binary matrices that do not contain certain forbidden submatrices has recently been the focus of much study. Examples are

- Matrices that do not contain any two *horizontally* or *vertically* adjacent 1s, i.e., do not contain submatrices $(1\,1)$ or $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ [5] [1]. This is sometimes known as the *independent sets of grid graphs* problem since the matrices correspond to independent sets of grid graphs.

- *Two-Dimensional* (a, b) *Run-Length Limited* matrices. In these, two parameters $0 \le a < b \le \infty$ are given. Between every two consecutive horizontal or vertical 1s in the matrix the number of 0s must be between a and b [7]. (The independent sets of grid graphs problem is equivalent to the $(1, \infty)$ run-length limited matrix problem).

- Checkerboard code matrices in which every 1 must be surrounded by a particular pattern of 0s [10].

One fact that has been repeatedly noted is, that for the constraints so far examined in the literature, the *capacity* of a two-dimensional constrained matrix,

$$C_f = \lim_{k,r\to\infty} \frac{\log_2 f(k,r)}{kr},$$

exists and is independent of how k, r run to infinity. We therefore state as fact the following lemma and point the readers interested in a proof to [7] which proves a very similar result for two-dimensional run-length limited matrices:

Lemma 1 *The channel capacity of the read/write isolated memory C defined in (2) exists and satisfies*

$$C = C_N = \lim_{k,r\to\infty} \frac{\log_2 N(k,r)}{kr}.$$

In particular,

$$C = \lim_{k\to\infty} \frac{1}{k} \lim_{r\to\infty} \frac{\log_2 N(k,r)}{r} = \lim_{r\to\infty} \frac{1}{r} \lim_{r\to\infty} \frac{\log_2 N(k,r)}{k}.$$

Before continuing we point out one way in which the RWIM problem does differ from previous constrained matrix ones. In the cases previously analyzed, such as the ones cited above, the constraints were all invariant under transposition. That is, if B satisfied a particular set of constraints then its transpose B^t, also satisfied that constraint. This meant that $\forall k, r$, $f(k,r) = f(r,k)$, and this equality was often implicitly used in the analysis. In our case this is obviously not true, e.g., $B = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$ is a good matrix but $B^t = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$ is not a good matrix. It is also not difficult to see that for most k, r, $f(k,r) \neq f(r,k)$. The majority of the new work in this paper is in modifying previously known analytic techniques to deal with this asymmetry.

We now describe the *transfer matrix* approach that has been used to attack many counting problems of the constrained-matrix type. Consider the rows of a constrained matrix; they are k-tuples or row vectors. Let R_k be the number of possible row vectors. In good matrices these vectors are just the k-tuples with no two consecutive '1's and one can calculate [4] that $R_k = F_{k+2}$ where F_k is the kth Fibonacci number. Order these vectors as $v_1, v_2, \ldots, v_{R_k}$. Now define the *transfer matrix* A_k to be the $R_k \times R_k$ binary matrix whose (i,j)th entry, $1 \leq i, j \leq R_k$, is 1 if and only if the $2 \times k$ matrix $\begin{pmatrix} v_i \\ v_j \end{pmatrix}$ is a valid constrained matrix, e.g., in our case it satisfies conditions 1 and 2. Otherwise the (i,j)th entry is 0.

With this definition, $N(k,r)$, the number of distinct $r \times k$ constrained matrices, satisfies $N(k,r) = \mathit{1} A_k^{r-1} \mathit{1}^t$ where $\mathit{1}$ is the R_k-vector containing all 1s. To see this we let $N_{v_i,v_j}(k,r)$ be the number of $r \times k$ constrained matrices whose *top* row is v_i and whose *bottom* row is v_j. Set $A^{(r-1)} = A_k^{r-1}$. We claim that $\forall i,j,k,r$ with $r > 1$, $N_{v_i,v_j}(k,r) = A^{(r-1)}(i,j)$. Note that $N_{v_i,v_j}(k,2) = A_k(i,j) = A^{(1)}(i,j)$. Thus, by induction,

$$N_{v_i,v_j}(k,r) = \sum_t N_{v_i,v_t}(k,r-1) A_k(t,j) = \sum_t A^{(r-2)}(i,t) A_k(t,j) = A^{(r-1)}(i,j).$$

Summing over all i,j gives

$$N(k,r) = \sum_{i,j} N_{v_i,v_j}(k,r) = \sum_{i,j} A^{(r-1)}(i,j) = \mathit{1} A^{(r-1)} \mathit{1}^t = \mathit{1} A_k^{r-1} \mathit{1}^t. \quad (3)$$

Notice that the definition of the transfer matrix is not fully determined until we specify the ordering of the R_k k-tuples. For example, in the case of our good matrices we can order the rows lexicographically, e.g., when $k = 4$ we order the rows as

$$(0000, 0001, 0010, 0100, 0101, 1000, 1001, 1010).$$

Using this ordering we have[1] [4]

[1] These transfer matrices were derived in [4] as the adjacency matrices of the *channel-adjacency graph* of the read-write isolated channel. The terminology here is different but the derivation there is equivalent to that in the constrained matrix problem.

$$A_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix},$$

$$A_4 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Figure 2: The transfer matrices A_1, A_2, A_3, A_4.

Theorem 1 *The transfer matrices of the read/write isolated memory problem satisfy*

$$A_0 = (1), \quad A_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad A_k = \begin{pmatrix} A_{k-1} & \hat{A}_{k-2} \\ \hat{A}^t_{k-2} & A_{k-2} \end{pmatrix},$$

where A_k is an $F_{k+2} \times F_{k+2}$ matrix, $F_{k+2} = F_{k+1} + F_k$, $F_0 = 0$, $F_1 = 1$, and $\hat{A}_{k-2} = \begin{pmatrix} A_{k-2} \\ 0 \end{pmatrix}$ is an $F_{k+1} \times F_k$ matrix.

The first four transfer matrices A_1, A_2, A_3, A_4 are shown in Figure 2.

Note that in all of the constrained matrix problems previously analyzed as well as in our own there has been another type of symmetry: $\begin{pmatrix} v_j \\ v_i \end{pmatrix}$ is a good matrix if and only if $\begin{pmatrix} v_i \\ v_j \end{pmatrix}$ is a good one. Thus all of the A_k are real symmetric matrices and their largest modulus eigenvalues are all real and positive. This leads immediately to the following observation

k	F_{k+2}	λ_k	$\frac{\log_2 \lambda_k}{k}$	$\frac{\log_2 \lambda_k}{k}\left(2-e^{1/k}\right)$	$\log_2 \frac{\lambda_k}{\lambda_{k-1}}$
1	2	2.000000	1.000000		
2	3	2.414215	0.635777	0.223335	0.271554
3	5	4.086133	0.676912	0.409117	0.759182
4	8	5.345956	0.604612	0.432887	0.387712
5	13	8.434573	0.615263	0.479042	0.657867
6	21	11.510559	0.587481	0.480935	0.448571
7	34	17.517792	0.590107	0.499487	0.605863
8	55	24.487541	0.576747	0.499954	0.483227
9	89	36.525244	0.576758	0.508978	0.576846
10	144	51.788146	0.569455	0.509565	0.503728
11	233	76.349987	0.568596	0.514483	0.560006
12	377	109.182967	0.564217	0.515184	0.516048
13	610	159.856988	0.563126	0.518099	0.550034
14	987	229.787737	0.560297	0.518812	0.523520

Table 1: F_k is the kth Fibonacci number. λ_k is the largest eigenvalue of the $F_{k+2} \times F_{k+2}$ transfer matrix A_k. The λ_k values are from Table 1 in [4].

Lemma 2 *Let A_k be the transfer matrices as defined above and λ_k the largest eigenvalue of A_k. Then*

$$\lim_{r\to\infty} \frac{\log_2 N(k,r)}{r} = \lim_{r\to\infty} \frac{\log_2 1A_k^{r-1}1^t}{r} = \log_2 \lambda_k$$

and

$$C = \lim_{k\to\infty} \frac{1}{k} \lim_{r\to\infty} \frac{\log_2 N(k,r)}{r} = \lim_{k\to\infty} \frac{\log_2 \lambda_k}{k}.$$

Much of the work on calculating capacities has thus concentrated on getting good bounds on the λ_k for the appropriate A_k, and finding out how quickly $\frac{\log_2 \lambda_k}{k}$ approaches C.

One important observation (made by Cohn [4] for this problem and by Weeks and Blahut [10] for Checkerboard Codes) is:

Lemma 3 *Let C, and λ_k be as defined above. Then*

$$\forall k, \quad C \leq \frac{\log_2 \lambda_k}{k}.$$

This immediately leads to a good method for upper bounding C : construct bigger and bigger transfer matrices A_k, calculate their largest eigenvalues, λ_k and then $\frac{\log_2 \lambda_k}{k}$. This is one of the techniques described in [4]. The results in that paper are reproduced[2] in Table 1 and yield the upper bound

$$C \leq \frac{\log_2 \lambda_{14}}{14} = 0.560297\ldots. \tag{4}$$

Surprisingly one can use similar techniques to *lower bound C*.

Lemma 4 *Let C, and λ_k be as defined above. Then*

$$\forall k,\ \frac{\log_2 \lambda_k}{k} \leq Ce^{1/k}. \tag{5}$$

Combining this with Lemma 3 yields

$$\forall k,\ 0 \leq \frac{\log_2 \lambda_k}{k} - C \leq C\left(e^{1/k} - 1\right). \tag{6}$$

Yet another application of Lemma 3 implies that

$$\forall k,\ \frac{\log_2 \lambda_k}{k}\left(2 - e^{1/k}\right) \leq C. \tag{7}$$

Proof. Set $d_k = \frac{\log_2 \lambda_k}{k}$. Recall that $N(2k+1, r)$ is the number of good $r \times (2k+1)$ matrices. But, given any two good $r \times k$ matrices B_1 and B_2, the matrix $(B_1\, 0^t\, B_2)$ is also good, where 0 represents a $1 \times r$ matrix of all zeros. Thus

$$N(k,r) \cdot N(k,r) \leq N(2k+1, r),$$

so

$$\frac{2\log_2 N(k,r)}{r} \leq \frac{\log_2 N(2k+1,r)}{r}.$$

Taking $\lim\limits_{r\to\infty}$ and applying Lemma 2 give

$$2\log_2 \lambda_k \leq \log_2 \lambda_{2k+1}$$

so

$$d_k = \frac{\log_2 \lambda_k}{k} \leq \frac{\log_2 \lambda_{2k+1}}{2k+1} \cdot \frac{2k+1}{2k} = d_{2k+1}\left(1 + \frac{1}{2k}\right).$$

[2]The authors thank Martin Cohn for his gracious permission to reproduce here the data from Table 1 in [4].

Telescoping this inequality gives, $\forall i \geq 0$,

$$\begin{aligned}
d_k &\leq d_{2k+1}\left(1+\frac{1}{2k}\right) \\
&\leq d_{2(2k+1)+1}\left(1+\frac{1}{2(2k+1)}\right)\left(1+\frac{1}{2k}\right) \\
&\leq d_{2((2k+1)+1)+1}\left(1+\frac{1}{2(2(2k+1)+1)}\right)\left(1+\frac{1}{2(2k+1)}\right)\left(1+\frac{1}{2k}\right) \\
&\vdots \\
&\leq d_{2^i k+\sum_{j=0}^{i-1} 2^j}\prod_{j\leq i}\left(1+\frac{1}{2^j k+\sum_{l=1}^{j-1} 2^l}\right)
\end{aligned}$$

From Lemma 1

$$\lim_{i\to\infty} d_{2^i k+\sum_{j=0}^{i-1} 2^j} = C.$$

Also, $\forall j$, $1+\frac{1}{2^j k+\sum_{l=1}^{j-1} 2^l} \leq 1+\frac{1}{2^j k}$ so we can take limits to find that

$$d_k \leq C\prod_{j=1}^{\infty}\left(1+\frac{1}{2^j k}\right) \leq C\prod_{j=1}^{\infty} e^{\frac{1}{2^j k}} = Ce^{1/k}.$$

$\square$

Plugging the value of λ_{14} from Table 1 into (7) we derive our first improved lower bound

$$C \geq \frac{\log_2 \lambda_{14}}{14}\left(2-e^{1/14}\right) = 0.51881181\ldots \tag{8}$$

3 A Second Transfer Matrix

In the previous section we introduced the transfer matrices A_k and examined some of their properties. A_k can be thought of as encoding information of how a *vertical* strip matrix of fixed width k can grow in height. We can just as easily introduce a different type of transfer matrix, though, one that encodes how a *horizontal* strip matrix of fixed height r can grow in width.

More specifically, note that *every* column vector of height r can appear in a good matrix. This is because if v is any r-tuple or vector then $(0^t\, v^t)$ is a good $r \times 2$ matrix. Now let $v_1, v_2, \ldots v_{2^r}$ be the 2^r r-tuples or vectors sorted lexicographically from smallest to largest. For example, when $r = 3$, we have

$$(000, 001, 010, 011, 100, 101, 110, 111).$$

$$\bar{A}_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \; B_1 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}, \bar{A}_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \; B_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\bar{A}_3 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \; B_3 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Figure 3: The horizontal transfer matrices $\bar{A}_1$, $\bar{A}_2$, $\bar{A}_3$ and associated matrices B_1, B_2 and B_3.

Definition 1 *Let $r > 0$ and $v_1, \ldots, v_{2^r}$ be the 2^r binary r-tuples or vectors sorted lexicographically. Define the* horizontal transfer matrix *$\bar{A}_r$ to be the $2^r \times 2^r$ matrix*

$$\bar{A}_r(i,j) = \begin{cases} 1 & \textit{if } \begin{pmatrix} v_i^t \, v_j^t \end{pmatrix} \textit{ is a good } r \times 2 \textit{ matrix} \\ 0 & \textit{otherwise} \end{cases}$$

The horizontal transfer matrices $\bar{A}_1$, $\bar{A}_2$, $\bar{A}_3$ and associated matrices B_1, B_2 and B_3 (defined in the following Lemma) are shown in Figure 3.

A straightforward combinatorial argument yields

Lemma 5 *Let $\bar{A}_r$ be as above. Let B_r be the $2^r \times 2^r$ binary matrix defined by*

$$B_r(i,j) = \begin{cases} 1 & \textit{if } \begin{pmatrix} 0 & 1 \\ v_i^t & v_j^t \end{pmatrix} \textit{ is a good } (r+1) \times 2 \textit{ matrix} \\ 0 & \textit{otherwise} \end{cases}$$

Then $\bar{A}_r$ and B_r satisfy the initial conditions:

$$\bar{A}_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \qquad B_1 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix},$$

and recurrences, $\forall r > 1$,

$$\bar{A}_r = \begin{pmatrix} \bar{A}_{r-1} & B_{r-1} \\ B_{r-1}^t & 0 \end{pmatrix}, \qquad B_{r-1} = \begin{pmatrix} \bar{A}_{r-2} & B_{r-2} \\ 0 & 0 \end{pmatrix}.$$

Proof. It is easy to check the initial conditions.

Let $r > 1$ and $v_1, \ldots, v_{2^{r-1}}$ be the 2^{r-1} binary $(r-1)$-tuples or vectors sorted lexicographically. For $0 \leq i, j, < 2^{r-1}$, $\alpha, \beta \in \{0, 1\}$ we have that $\bar{A}_r(\alpha 2^{r-1} + i, \beta 2^{r-1} + j) = 1$ if and only if matrix $\begin{pmatrix} \alpha & \beta \\ v_i^t & v_j^t \end{pmatrix}$ is good. Enumerating the cases

1. matrix $\begin{pmatrix} 0 & 0 \\ v_i^t & v_j^t \end{pmatrix}$ is good if and only if matrix $\begin{pmatrix} v_i^t \, v_j^t \end{pmatrix}$ is good. Thus, $\bar{A}_r(i,j) = \bar{A}_{r-1}(i,j)$;
2. By definition , $\bar{A}_r(i, j + 2^{r-1}) = B_{r-1}(i,j)$, $\bar{A}_r(i + 2^{r-1}, j) = B_{r-1}(j,i)$;
3. matrix $\begin{pmatrix} 1 & 1 \\ v_i^t & v_j^t \end{pmatrix}$ is not good. Thus $\bar{A}_r(i + 2^{r-1}, j + 2^{r-1}) = 0$.

The recurrence relation for $\bar{A}_r$ follows.

Similarly $B_r(\alpha 2^{r-1} + i, \beta 2^{r-1} + j) = 1$ if and only if matrix $\begin{pmatrix} 0 & 1 \\ \alpha & \beta \\ v_i^t & v_j^t \end{pmatrix}$ is good. Enumerating the cases

1. matrix $\begin{pmatrix} 0 & 1 \\ 0 & 0 \\ v_i^t & v_j^t \end{pmatrix}$ is good if and only if matrix $\begin{pmatrix} v_i^t \, v_j^t \end{pmatrix}$ is good. Thus, $B_r(i,j) = \bar{A}_{r-1}(i,j)$;
2. matrix $\begin{pmatrix} 0 & 1 \\ 0 & 1 \\ v_i^t & v_j^t \end{pmatrix}$ is good if and only if matrix $\begin{pmatrix} 0 & 1 \\ v_i^t & v_j^t \end{pmatrix}$ is good. Thus, $B_r(i, j + 2^{r-1}) = B_{r-1}(i,j)$;
3. matrices of the form $\begin{pmatrix} 0 & 1 \\ 1 & 0 \\ * & * \end{pmatrix}$ and $\begin{pmatrix} 0 & 1 \\ 1 & 1 \\ * & * \end{pmatrix}$ are not good. Thus $B_r(i + 2^{r-1}, j) = B_r(i + 2^{r-1}, j + 2^{r-1}) = 0$.

r	2^r	μ_r	$\frac{\log_2 \mu_r}{r}$	$\log_2 \frac{\mu_r}{\mu_{r-1}}$
1	1	1.618	0.6942116	
2	2	2.302775637	0.6016869270	0.5091622465
3	4	3.346462191	0.5808789050	0.5392628601
4	8	4.845619214	0.5691702593	0.5340443229
5	16	7.021562462	0.5623584200	0.5351110622
6	32	10.17359346	0.5577929075	0.5349653450
7	64	14.74105370	0.5545382497	0.5350103028
8	128	21.35908135	0.5520972116	0.5350099454

Table 2: μ_r is the largest eigenvalue of the $2^r \times 2^r$ transfer matrix $\bar{A}_r$.

The recurrence relation for B_r follows. □

By working through the details one finds that almost all of the properties of the transfer matrix A_k that were derived in the previous section correspond to analogous properties for $\bar{A}_r$.

Lemma 6 *Let μ_r be the largest eigenvalue of $\bar{A}_r$. Then*

1. $N(k,r) = 1\bar{A}_r^{k-1}1^t$.
2. $\lim_{k\to\infty} \frac{\log_2 N(k,r)}{k} = \lim_{k\to\infty} \frac{\log_2 1\bar{A}_r^{k-1}1^t}{k} = \log_2 \mu_r$.
3. $C = \lim_{r\to\infty} \frac{\log_2 \mu_r}{r}$.
4. $\forall r \geq 0, \quad C \leq \frac{\log_2 \mu_r}{r}$.

Note that there seems to be no lemma for μ_r corresponding to Lemma 4 that permits lower bounding C.

Substituting the calculated value for μ_8 into part 4 of the lemma yields

$$C \leq \frac{\log_2 \mu_8}{8} = 0.55209\ldots,$$

which is our improved upper bound.

4 The Maximum Principle and Better Lower Bounds

In this section we show how to use the *maximum principle* for eigenvalues to prove better lower bounds. This method was originally used by Engel [5] and Calkin and Wilf [1] for the independent sets of grid graphs problem.

The maximum principle [2] states that if A is a real and symmetric $n \times n$ matrix with largest eigenvalue λ and x is any nonzero real $1 \times n$ matrix then, for every positive integer p,

$$\frac{xA^p x^t}{xx^t} \le \lambda^p.$$

Let $k, q \ge 0$ and set $A = A_k$, $x = 1A_k^q$. Then the maximum principle, the fact that A_k is symmetric, (3) and Lemma 6 together yield

$$\lambda_k^p \ge \frac{1A_k^{2q+p}1^t}{1A_k^{2q}1^t} = \frac{N(k, 2q+p+1)}{N(k, 2q+1)} = \frac{1\bar{A}_{2q+p+1}^{k-1}1^t}{1\bar{A}_{2q+1}^{k-1}1^t}. \tag{9}$$

From Lemma 2 we have that $\lim_{k\to\infty} (\lambda_k)^{1/k} = 2^C$. From Lemma 6 we have that

$$\lim_{k\to\infty} \left(1\bar{A}_{2q+p+1}^{k-1}1^t\right)^{1/k} = \mu_{2q+p+1}, \quad \lim_{k\to\infty} \left(1\bar{A}_{2q+1}^{k-1}1^t\right)^{1/k} = \mu_{2q+1}. \tag{10}$$

Thus, taking both sides of (9) to the power $1/k$ and $\lim_{k\to\infty}$ yields

$$\forall p \ge 1, q \ge 0, \quad 2^{pC} \ge \frac{\mu_{2q+p+1}}{\mu_{2q+1}}. \tag{11}$$

A symmetric argument gives

$$\forall p \ge 1, q \ge 0, \quad 2^{pC} \ge \frac{\lambda_{2q+p+1}}{\lambda_{2q+1}}. \tag{12}$$

Taking $\log_2$ of both sides of (11) and (12) yields new lower bounds for C.

Lemma 7 *For all* $p \ge 1, q \ge 0$

$$C \ge \frac{1}{p}\log_2 \frac{\lambda_{2q+p+1}}{\lambda_{2q+1}}, \qquad C \ge \frac{1}{p}\log_2 \frac{\mu_{2q+p+1}}{\mu_{2q+1}}.$$

Referring back to Tables 1 and 2 we find that the best lower bound achievable using this lemma and our calculated data is achieved by setting $q = 3$, $p = 1$ to derive

$$C \ge \log_2 \frac{\mu_8}{\mu_7} = .53500\ldots.$$

5 Conclusion and Open Problems

In this paper we described how to model the binary read/write isolated memory as a particular type of constrained binary matrix. This permitted us to transform the problem of finding the channel capacity of the memory into one of finding the capacity of the matrix. That in turn let us use relatively recent analytic tools that had been developed for attacking such matrix problems.

Our main result was that, C the channel capacity of the read/write isolated memory, satisfies

$$0.53500\ldots \leq C \leq 0.55209\ldots.$$

The main open question is how to improve this. We conjecture that

$$C = 0.5350\ldots,$$

matching the lower bound. Our reasoning follows: recall from Lemma 7 that for all k, r,

$$C \geq \log_2 \frac{\lambda_{2k+2}}{\lambda_{2k+1}}, \qquad C \geq \log_2 \frac{\mu_{2r+2}}{\mu_{2r+1}}.$$

We have no such corresponding formulas for $\log_2 \frac{\lambda_{2k+1}}{\lambda_{2k}}$ and $\log_2 \frac{\mu_{2r+1}}{\mu_{2r}}$ but our admittedly scant data does seem to support the conjecture that $\forall k, r$,

$$C \leq \log_2 \frac{\lambda_{2k+1}}{\lambda_{2k}}, \quad C \leq \log_2 \frac{\mu_{2r+1}}{\mu_{2r}}. \tag{13}$$

This conjecture parallels a similar one in the independent sets of grid graphs problem [5] where data seems to suggest the same behavior. If (13) is true then from Table 2

$$0.5350\ldots = \log_2 \frac{\mu_8}{\mu_7} \leq C \leq \log_2 \frac{\mu_7}{\mu_6} = 0.5350\ldots.$$

We conclude by pointing out that our derivation of the lower bound in Section 4 using the maximum principle essentially followed a similar derivation in [1] and [5] for the capacity of the independent set of grid graph problem. The difference between their result and ours was that their derivation implicitly used the fact that in the grid-graph problem if B is a good matrix then its transpose B^t is also a good matrix. This in turn implied that their transfer matrix for growing vertical strips was exactly the same as their transfer matrix for growing horizontal strips, i.e., $\forall r$, $A_r \equiv \bar{A}_r$ and $\lambda_r = \mu_r$, which simplified their version of equation (9). An interesting consequence of our derivation is that it is not necessary to have this symmetry condition of $B = B^t$. All that is necessary to perform the analysis and derive variants of Lemma 7 is that all of the *transfer matrices* A_r and $\bar{A}_r$ be symmetric.

Acknowledgments. This research was performed while the first three authors were visiting DIMACS, at Rutgers, The State University of New Jersey.

References

[1] N. Calkin and H. Wilf "The number of independent sets in a grid graph," *SIAM J. Discrete Math,* **11** (1998), 54-60.

[2] F. Chatelin, *Eigenvalues of Matrices,* John Wiley & Sons, New York (1993).

[3] G. Cohen and G. Zemor, "Write-isolated memories," *Discrete Math,* **114** (1993) 105-113.

[4] M. Cohn, "On the channel capacity of read/write isolated memory," *Discrete Applied Math,* **56** (1995) 1-8.

[5] Konrad Engel "On the Fibonacci Number of an M X N Lattice," *Fibonacci Quarterly,* **28** (1990) 72-78.

[6] C. V. Freiman and A. D. Wyner, "Optimum block codes for noiseless input-restricted channels," *Inform. and Control,* **7** (1964) 398-415.

[7] A. Kato and K. Zeger "On the Capacity of Two-Dimensional Run-Length Constrained Channels," *IEEE Transactions on Information Theory,* **45**(5), (July 1999) 1527-1540.

[8] J. P. Robinson, "An asymmetric error-correcting ternary code," *IEEE Trans. Inform. Theory* **24** (1978) 258-261.

[9] C. E. Shannon, *The Mathematical Theory of Communication,* University of Illinois Press, Urbana, IL, (1949).

[10] W. Weeks and R. Blahut "The Capacity and Coding Gain of Certain Checkerboard Codes," *IEEE Transactions on Information Theory,* **44**(3), (May 1998) 1193-1203.

[11] J. K. Wolf, A. D. Wyner, J. Ziv and J. Körner, "Coding for write-once memory," *AT&T Bell Laboratories Tech. Journal,* **63**(6)(1984) 1089-1112.