On Minimum Saturated Matrices

Andrzej Dudek∗ Oleg Pikhurko∗† Andrew Thomason‡

September 10, 2009

Abstract

Motivated by the work of Anstee, Griggs, and Sali on forbidden submatrices and the extremal sat-function for graphs, we introduce sat-type problems for matrices. Let $F$ be a family of $k$-row matrices. A matrix $M$ is called $F$-admissible if $M$ contains no submatrix $F \in F$ (as a row and column permutation of $F$). Moreover, $M$ is $F$-saturated if $M$ is $F$-admissible but the addition of any column not present in $M$ violates this property. In this paper we consider the function $\text{sat}(n,F)$ which is the minimum number of columns of an $F$-saturated matrix with $n$ rows. We establish the estimate $\text{sat}(n,F) = O(n^{k-1})$ for any family $F$ of $k$-row matrices and also compute the sat-function for a few small forbidden matrices.

1 Introduction

First, we introduce some notation. Let the shortcut ‘an $m \times n$-matrix' $M$ mean a matrix with $m$ rows (which we view as horizontal arrays) and $n$ ‘vertical' columns. Usually, we restrict entries to only two values, 0 and 1; in all other cases, we indicate the range explicitly. For an $n \times m$-matrix $M$, its order $v(M) = n$ is the number of rows and its size $e(M) = m$ is the number of columns. We distinguish the expressions like ‘an $n$-row matrix’ and ‘an $n$-row' standing respectively for a matrix with $n$ rows and for a row containing $n$ elements.

For an $n \times m$-matrix $M$ and sets $A \subseteq [n]$ and $B \subseteq [m]$, $M(A,B)$ is the $|A| \times |B|$-submatrix of $M$ formed by the rows indexed by $A$ and the columns indexed by $B$. We use the following obvious shorthands: $M(A,\_)=M(A,[m])$, $M(A,i)=M(A,\{i\})$, etc. For example, the rows and the columns of $M$ are denoted by $M(1,\_),\ldots,M(n,\_)$ and $M(\_,1),\ldots,M(\_,m)$ respectively while individual entries – by $M(i,j)$, $i \in [n]$, $j \in [m]$.

∗Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213, USA
†Partially supported by the National Science Foundation, Grant DMS-0758057.
‡Department of Pure Mathematics and Mathematical Statistics, University of Cambridge, Cambridge, CB3 0WB, UK
A matrix $F$ is a submatrix of a matrix $A$ (denoted as $F \subseteq A$) if deleting some set of rows and columns of $A$ we can obtain a submatrix which is a row/column permutation of $F$. The transpose of $M$ is denoted by $M^T$ (which we use mostly to denote vertical columns for typographical reasons); $(a)^i$ is the (horizontal) sequence containing element $a$ $i$ times. The $n \times (m_1 + m_2)$-matrix $[M_1, M_2]$ is obtained by concatenating an $n \times m_1$-matrix $M_1$ and an $n \times m_2$-matrix $M_2$. We write $N \cong M$ to say that $N$ is a column/row permutation of $M$. Thus, $N \subseteq M$ if $N \cong M(A, B)$ for some index sets $A$ and $B$. The complement $1 - M$ of a matrix $M$ is obtained by interchanging ones and zeros in $M$. The characteristic function $\chi_Y$ of $Y \subseteq [n]$ is the $n$-column with $i$th entry being 1 if $i \in Y$ and 0 otherwise.

Many interesting and important properties of matrices (that is, classes of matrices) can be defined by listing forbidden submatrices. (Some authors use the term ‘forbidden configurations’ then.) More precisely, given a family $F$ of matrices (referred to as forbidden), we say that a matrix $M$ is $F$-admissible (or $F$-free) if $M$ contains no $F \in F$ as a submatrix. A simple matrix $M$ (that is, a matrix without repeated columns) is called $F$-saturated or ($F$-critical) if $M$ is $F$-free but the addition of any column not present in $M$ violates this property; this is denoted by $M \in \text{SAT}(n, F)$, $n = v(M)$. Note that, although the definition requires that $M$ is simple, we allow multiple columns in matrices belonging to $F$.

One well-known extremal problem is to consider $\text{forb}(n, F)$, the maximum size of a simple $F$-free matrix with $n$ rows or, equivalently, the maximal size of $M \in \text{SAT}(n, F)$. Many different results on the topic have been obtained, some of them will be mentioned in the due course, but we do not even try to give a comprehensive survey here. We would only mention a remarkable fact that one of the first forb-type results, namely formula (1) here, proved independently by Vapnik and Chervonenkis [21], Perles and Shelah [19], and Sauer [18], was motivated by such different topics as probability, logic and a problem of Erdős on infinite set systems.

The forb-problem is reminiscent of the Turán function $\text{ex}(n, F)$: given a family $F$ of forbidden graphs, find the maximum size of an $F$-free graph on $n$ vertices not containing any member of $F$ as a subgraph (see e.g. surveys [15, 20]). Erdős, Hajnal, and Moon [11] considered the ‘dual’ function $\text{sat}(n, F)$, the minimum size of a maximal $F$-free graph on $n$ vertices. This is an active area of extremal graph theory, see [6, Section 3] for a short overview.

Here we consider the ‘dual’ of the forb-problem for matrices. Namely, we are interested in the value of $\text{sat}(n, F)$, the minimum size of an $F$-saturated matrix with $n$ rows:

$$\text{sat}(n, F) = \min \{e(M) : M \in \text{SAT}(n, F)\}.$$ 

We decided to use the same notation as for its graph counterpart. This should not cause any confusion as this paper will deal with matrices alone.

Obviously, $\text{sat}(n, F) \leq \text{forb}(n, F)$. If $F = \{F\}$ consists of a single forbidden matrix $F$ then we write $\text{SAT}(n, F) = \text{SAT}(n, \{F\})$, etc.

A simple $n$-row matrix $M$ is monotonically $F$-saturated, denoted $M \in \text{m-SAT}(n, F)$,
if the addition of any new \( n \)-column \( C \) to \( M \) creates a new forbidden submatrix, that is, there is \( A \subseteq [n] \) such that \( M(A) \) is \( \mathcal{F} \)-free while \( [M, C](A) \) is not. Clearly, \( m\text{-SAT}(n, \mathcal{F}) \supseteq \text{SAT}(n, \mathcal{F}) \), so \( m\text{-sat}(n, \mathcal{F}) \leq \text{sat}(n, \mathcal{F}) \) where as always \( m\text{-sat}(n, \mathcal{F}) = \min\{\epsilon(M) : M \in m\text{-SAT}(n, \mathcal{F})\} \). This is the natural analog of the \( m\text{-sat} \)-function for graphs, which was studied already in [11]. It is useful in proving lower bounds on \( \text{sat}(n, \mathcal{F}) \) via induction on \( n \).

There is an obvious generalization of these problems when we consider the class of \([0, l]\)-matrices (matrices whose entries can assume \((l + 1)\)-values from \( \{0, \ldots, l\} \)) with the above definitions going practically unchanged. In this case we will use symbols like \( \text{SA T}(n, \mathcal{F}; l) \), etc., whereas the default \( \text{SA T}(n, \mathcal{F}) = \text{SAT}(n, \mathcal{F}; 1) \) is the usual notion.

By \( T^l_k \) we denote the simple \( k \times \binom{k}{l} \)-matrix consisting of all \( k \)-columns with exactly \( l \) ones and by \( K^l_k \)–the \( k \times 2^k \)-matrix of all possible columns of size \( k \). Naturally, \( T^\leq_l k \) denotes the \( k \times f(k, l) \)-matrix consisting of all distinct columns with at most \( l \) ones, etc, where we use the shortcut \( f(n, k) = \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{k} \).

Vapnik and Chervonenkis [21], Perles and Shelah [19], and Sauer [18] showed independently that
\[
\text{forb}(n, K^l_k) = f(n, k - 1). \tag{1}
\]

As formula (1) plays an important role in our consideration of the \( \text{sat} \)-function, we give a proof, for the sake of completeness, of a more general result which is of independent interest. Namely, in the class of \([0, l]\)-matrices, we compute the maximum size of a \( K^l_k \)-free \( n \)-row matrix, where \( K^l_k \) is the \( k \)-row matrix of size \((l + 1)^k \) consisting of all distinct \( k \)-columns made of symbols from \([0, l] = \{0, \ldots, l\} \). Our proof is based on the ideas of Frankl introduced in [13].

**Theorem 1.1** For any \( l \geq 1, k \geq 1 \) and \( n \geq k - 1 \),
\[
\text{forb}(n, K^l_k; l) = \sum_{i=0}^{k-1} m^{-i} \binom{n}{i}. \tag{2}
\]

**Proof.** To prove the lower bound, consider the matrix made of all columns containing \( l \) at most \( k - 1 \) times which is obviously \( K^l_k \)-admissible.

It remains to prove the upper bound. Consider any \( K^l_k \)-admissible matrix \( M \) of order \( n \) and size \( m \). Fix an integer \( i, 1 \leq i \leq n \). Divide the set of columns of \( M \) into disjoint sets \( C_1, \ldots, C_p \), \( 1 \leq p \leq m \), such that all columns in \( C_j \), \( 1 \leq j \leq p \), agree except in the \( i \)th row. Clearly, \( |C_j| \leq l + 1 \). Let \( C'_j \) be obtained from \( C_j \) by replacing the set of entries corresponding to the \( i \)th row in columns of \( C_j \) by \( \{0, \ldots, |C_j| - 1\} \).

Let \( M'_i \) be a matrix consisting of all columns from \( C'_j \) over all \( 1 \leq j \leq p \).

*Claim.* If \( M'_i \supseteq K^l_k \) then \( M \supseteq K^l_k \).
Suppose \( M_i' \supseteq K^k_i \). Let \( R \) be the subset of \( k \) rows of \( M_i' \) such that \( M_i'(R,.) \supseteq K^k_i \).

If \( i \) is not in \( R \) then \( M \) also contains \( K^k_i \) (since \( M \) and \( M_i' \) are identical apart from row \( i \)). Thus, we may assume that \( i \) is in \( R \). Let \( C \) be any column of \( K^k_i \). Let \( C' \) be the same column with \( l \) in the \( i \)th entry. Then some column \( D_l \) of \( M_i' \) gives \( C' \) when restricted to \( R \). And \( D_l \) has an \( l \) in the \( i \)th entry, so if \( D_l \) is in \( M_i' \), \( D_l \) is also in \( M \), and every column obtained from \( D_l \) by putting anything in the \( l \)th entry is also in \( M \). But then \( C \) is a column of \( M \) when restricted to \( R \). Thus \( M \supseteq K^k_i \), as required.

In order to prove the theorem we recursively repeat the claim for every row \( i \), \( 1 \leq i \leq n \), obtaining a matrix \( M' \) of size \( m \). Observe that if any column of \( M' \) has \( k \) entries equal to \( l \) then \( M' \supseteq K^k_i \) and so by the claim \( M \supseteq K^k_i \), a contradiction. Hence, every column of \( M' \) has at most \( k-1 \) entries equal to \( l \) which gives the theorem.

\[ \square \]

## 2 General Results

Here we present some results dealing with \( \text{sat}(n,\mathcal{F}) \) for a general family \( \mathcal{F} \).

The following simple observation may be useful in tackling these problems. Let \( M' \) be obtained from \( M \in \text{SAT}(n,\mathcal{F}) \) by duplicating the \( n \)th row of \( M \), that is, we let \( M'([n],.) = M \) and \( M'(n+1,.) = M(n,.) \). Suppose that \( M' \) is \( \mathcal{F} \)-admissible. Complete \( M' \), in an arbitrary way, to an \( \mathcal{F} \)-saturated matrix. Let \( C \) be any added \((n+1)\)-column. As both \( M'([n],.) \) and \( M'([n-1]\cup\{n+1\},.) \) are equal to \( M \in \text{SAT}(n,\mathcal{F}) \), we conclude that both \( C([n]) \) and \( C([n-1]\cup\{n+1\}) \) must be columns of \( M \). As \( C \) is not an \( M' \)-column, \( C = (C',b,1-b) \) for some \((n-1)\)-column \( C' \) and \( b \in \{0,1\} \) such that both \((C',0)\) and \((C',1)\) are columns of \( M \). This implies that \( \text{sat}(n+1,\mathcal{F}) \leq e(M)+2d \), where \( d \) is the number of pairs of equal columns in \( M \) after we delete the \( n \)th row.

The above argument works for the \( m \)-sat-function. Namely, duplicate a row of some \( M \in \text{m-SAT}(n,\mathcal{F}) \) and add one by one missing columns so that no new forbidden submatrix appears; clearly the resulting matrix is monotonically \( \mathcal{F} \)-saturated and, like above, we add at most \( 2d \) extra rows. In particular, the following theorem follows.

**Theorem 2.1** Either \( m \)-sat\((n,\mathcal{F}) \) is constant for large \( n \) or \( m \)-sat\((n,\mathcal{F}) \) \( \geq n+1 \) for every \( n \). The analogous statement is true for the sat-function if no matrix in \( \mathcal{F} \) has two equal rows.

**Proof.** If we have some \( M \in \text{m-SAT}(n,\mathcal{F}) \) with at most \( n \) columns then a well-known theorem of Bondy [7] (see, e.g., Theorem 2.1 in [5]) implies that there is \( i \in \{n\} \) such that the removal of the \( i \)th row does not create two equal columns. Hence, the duplication of the \( i \)th row gives a monotonically \( \mathcal{F} \)-saturated matrix, which implies \( m \text{-sat}(n+1,\mathcal{F}) \leq m \text{-sat}(n,\mathcal{F}) \) and the first part follows. The same argument establishes the second part as the duplication of a row cannot create a forbidden submatrix by the condition on \( \mathcal{F} \).

\[ \square \]

Suppose that \( \mathcal{F} \) consists of \( k \)-row matrices. Is there any good general upper bound on \( \text{forb}(n,\mathcal{F}) \) or \( \text{sat}(n,\mathcal{F}) \)? There were different papers dealing with general upper
bounds on \(\text{forb}(n, \mathcal{F})\), e.g., by Anstee and Füredi [1], by Frankl, Füredi and Pach [13] and by Anstee [11], until the conjecture of Anstee and Füredi [4] that \(\text{forb}(m, \mathcal{F}) = O(n^k)\) for any fixed \(\mathcal{F}\) was elegantly proved by Füredi (see [2] for a proof).

On the other hand, we can show that \(\text{sat}(n, \mathcal{F}) = O(n^{k-1})\) for any family \(\mathcal{F}\) of \(k\)-row matrices (including infinite families). This is the matrix analog of the main result in [17]. Note we cannot decrease the exponent of \(k - 1\) with the estimate remaining true for any \(\mathcal{F}\); for example, \(\text{sat}(n, T_{k}^k) = f(n, k - 1)\).

**Theorem 2.2** For any family \(\mathcal{F}\) of \(k\)-row matrices, \(\text{sat}(n, \mathcal{F}) = O(n^{k-1})\).

**Proof.** We may assume that \(K_k\) is \(\mathcal{F}\)-admissible (i.e. every matrix of \(\mathcal{F}\) contains a pair of equal columns) for otherwise we are home by (1) as then \(\text{sat}(n, \mathcal{F}) \leq \text{forb}(n, K_k) = O(n^{k-1})\).

Let \(l \in [0, k]\) be the smallest number such that there exists \(m\) for which \([mT_{k}^{\leq l}, T_{k}^{> l}]\) is not \(\mathcal{F}\)-admissible. (Clearly, such \(l\) exists as \(T_{k}^{\leq k} = K_k\) is the complete matrix.) Let \(d\) be the maximal integer such that \([mT_{k}^{< l}, dT_{k}^{l}, T_{k}^{> l}]\) is \(\mathcal{F}\)-admissible for any \(m\). Note that \(d \geq 1\) as \([mT_{k}^{< l}, T_{k}^{l}, T_{k}^{> l}] = [mT_{k}^{< l}, T_{k}^{> l}]\) cannot contain a forbidden submatrix by the choice of \(l\). Choose minimal \(m\) such that \([mT_{k}^{< l}, (d+1)T_{k}^{l}, T_{k}^{> l}]\) is not \(\mathcal{F}\)-admissible.

Suppose first that \(l < k\). Given \(n\), let \(N\) be the \(n\)-row matrix corresponding to the following set system:

\[
H = \bigcup_{j \in [d-1]} \{Y \subseteq [n]^{(l+1)} : \sum_{y \in Y} y \equiv j \pmod{n}\}.
\]

Here \(X^{(i)} = \{Y \subseteq X : |Y| = i\}\) denotes the set of all subsets of \(X\) of size \(i\).

Note that any \(A \subseteq [n]^{(l)}\) is covered by at most \(d-1\) edges of \(H\) as there are at most \(d-1\) possibilities to choose \(i \in [n] \setminus A\) so that \(A \cup \{i\} \in H\): namely, \(i \equiv j - \sum_{a \in A} a \pmod{n}\) for \(j \in [d-1]\).

On the other hand, the set \(H\) of all \(l\)-subsets of \([n]\) covered by fewer than \(d-1\) edges of \(H\) has size at most \(2(d-1){n \choose l}\). Indeed, if \(A \in H\) then, for some \(j \in [d-1]\) and \(x \in A\), \(2x = j - \sum_{a \in A \setminus \{x\}} a \pmod{n}\) so, once \(A \setminus \{x\}\) and \(j\) have been chosen, there are at most \(2\) choices for \(x\).

Call \(X \subseteq [n]^{(k)}\) **bad** if, for some \(A \subseteq X^{(l)}\),

\[
|\{Y \in H : Y \supseteq A, Y \cap (X \setminus A) = \emptyset\}| \leq d - 2.
\]  

(3)

To obtain a bad \(k\)-set \(X\), we either complete some \(A \in H\) to any \(k\)-set or take any \(l\)-set \(A\) and let \(X \supseteq A\) intersect some \(H\)-edge covering \(A\). Therefore, the number of bad sets is at most

\[
2(d-1){n \choose l-1}{n \choose k-l} + {n \choose l}(d-1){n \choose k-l-1} = O(n^{k-1}).
\]

Let \(M' = [N, T_{k}^{l}]\). Clearly,

\[
M'X \subseteq \left[{n \choose l}T_{k}^{\leq l}, dT_{k}^{l}, T_{k}^{l+1}\right].
\]  

for any \(X \subseteq [n]^{(k)}\).
Hence, \( M' \) cannot contain a forbidden submatrix. Now complete it to arbitrary \( M = [M', M''] \in \text{SAT}(n, \mathcal{F}) \).

Suppose that \( e(M'') \neq \Omega(n^{k-1}) \). Then, by \( \Box \), \( K_k \cong M''(X, Y) \) for some \( X, Y \).

Now, remove the columns corresponding to \( X \) from \( M'' \) and repeat the procedure as long as possible to obtain \( > \Omega(n^{k-1}) \) column-disjoint copies of \( K_k \) in \( M'' \). If some \( X \in [n]^{(k)} \) appears more than \( d \) times then (because \( T_n^l(X, \alpha) \supseteq mT_k^{<l} \) for \( n \ge m + k \)) \( M(X, \alpha) \supseteq [mT_k^{<l}, (d + 1)K_k] \) is not \( \mathcal{F} \)-admissible. Otherwise, \( K_k \cong M''(X, \alpha) \) for some good (i.e., not bad) \( X \in [n]^{(k)} \); but then \( N(X, \alpha) \supseteq (d - 1)T_k^l \) and \( M(X, \alpha) \cong \) contains a forbidden matrix. This contradiction proves the required bound for \( l \leq k \).

Suppose that \( l = l(\mathcal{F}) \) equals \( k \); the above argument does not work in this case because \( M' \cong T_n^k \) is too large. Let \( N_s \) denote the \((d + s) \times d\)-matrix made of \( \chi_{[s + d] \setminus \{i\}} \), \( i \in [s + 1, s + d] \), where \( d = d(\mathcal{F}) \) is as above; clearly, each \( N_s \) is \( \mathcal{F} \)-free. Also let \( C^* \) be obtained from a column \( C \) by adding \( d \) zeros to the end; to obtain \( M^* \), we apply this operation to every column of a matrix \( M \); this increases the number of rows by \( d \).

Let \( \mathcal{F}^* \) consist of all matrices \( M \) such that \( [M^*, N_v(M)] \) is not \( \mathcal{F} \)-free; clearly, this property is not affected by a row/column permutation of \( M \).

Note that \( T = [mT_k^{<k}, T_k^l] \) is not \( \mathcal{F}^* \)-admissible as

\[
[T^*, N_k([k]),] = [T, dT_k^l] \supseteq [mT_k^{<k}, (d + 1)T_k^l]
\]

so \( l(\mathcal{F}^*) < k \) and, by the above argument, we can find \( L \in \text{SAT}(n - d, \mathcal{F}^*) \) with \( O(n^{k-1}) \) columns. Now, \( M' = [L^*, N_{n-d}] \) is \( \mathcal{F} \)-free so complete it to an arbitrary \( M \in \text{SAT}(n, \mathcal{F}) \). Let \( C \) be any added column; as \([M', C][([n-d]),] \) is \( \mathcal{F}^* \)-free (otherwise \( M \) contains a forbidden submatrix), \( C([n-d]) \) either is a column of \( L \) or equals \( ((1)^{n-d})^T \). Hence, \( e(M) \leq 2^d(e(M_1) + 1) \) and the theorem follows. \( \Box \)

3 Forbidding Complete Matrices

Let us investigate the value of \( \text{sat}(n, K_k) \). (Recall that \( K_k \) is the \( k \times 2^k \)-matrix consisting of all distinct \( k \)-columns.) We are able to settle the cases \( k = 2 \) and \( k = 3 \).

We will use the following trivial lemma a couple of times.

**Lemma 3.1** Each row of any \( M \in m\text{-SAT}(n, K_k) \), \( n \ge k \), contains at least \( l \) ones and at least \( l \) zeros, \( l = 2^{k-1} - 1 \).

**Proof.** Suppose that the first row \( M(1, \cdot) \) has \( m_0 \) zeros followed by \( m_1 \) ones with \( m_0 \geq m_1 \) and \( l > m_1 \).

For \( i \in [m_0] \), let \( C_i \) equal the \( i \)th column of \( M \) with the first entry 0 replaced by 1. Then \( M' = [M, C_i] \) cannot contain \( K_k \) because the first row of \( M' \) contains too few 1’s while \( C_i([2, n]) \) is a column of \( M([2, n]) \). Therefore, \( C_i \) must be a column of \( M \) and \( m_0 = m_1 \).

But then \( M \) has at most \( 2^k - 2 \) columns, which is a contradiction. \( \Box \)
The following result sounds as a rather natural (and not difficult) question if reformulated in the terms of set systems but we have not been able to find it in the literature.

**Theorem 3.2** For \( n \geq 1 \), \( m \text{-sat}(n, K_2) = \text{sat}(n, K_2) = \text{forb}(n, K_2) = n + 1 \).

*Proof.* Suppose that the statement is not true, that is, there exists a monotonically \( K_2 \)-saturated matrix with its size not exceeding its order. By Theorem 2.1, \( m \text{-sat}(n, K_2) \) is eventually constant so we can find an \( n \times m \)-matrix \( M \in m \text{-SAT}(n, K_2) \) having two equal rows for some \( n \in \mathbb{N} \).

As we are free to complement and permute rows, we may assume that, for some \( i \geq 2 \), \( M(1, \ldots, i) = \cdots = M(i, \ldots, n) \) while \( M(j, \ldots, n) \neq M(1, \ldots, i) \) and \( M(j, \ldots, n) \neq 1 - M(1, \ldots, i) \) for any \( j \in [i + 1, n] \). Note that \( i < n \) as we do not allow multiple columns in \( M \) (and \( m \geq v(K_2) - 1 = 3 \)). By permuting the rows in \( M([i + 1, n]) \), we may assume that for some \( l \in [i, n] \), we have \( M(\{1, \ldots, l\}, \ldots, n) \supseteq K_2 \) if and only if \( j \in [i + 1, l] \).

Let \( j \in [i + 1, n] \). By Lemma 3.1 the \( j \)th row \( M(j, \ldots, n) \) has both entries 0 and 1. By the definition of \( i \), \( M(j, \ldots, n) \) is not equal to \( M(1, \ldots, i) \) nor to \( 1 - M(1, \ldots, i) \). It easily follows that there are \( f_j, g_j \in [m] \) with \( M(1, f_j) = M(1, g_j) \), \( M(j, f_j) \neq M(j, g_j) \); furthermore, we can find \( h_j \in [m] \) with \( M(1, h_j) = 1 - M(1, f_j) \). Let \( b_j = M(j, h_j) \); we may assume that \( M(j, g_j) = b_j \). Furthermore, for \( j \in [i + 1, l] \), we choose \( f_j, g_j, h_j \) so that \( b_j = M(j, 1) \), which is possible as \( M(\{1, \ldots, l\}, \ldots, n) \supseteq K_2 \).

Now, as \( M \in \text{SAT}(n, K_2) \), the addition of the column

\[
C = (1, (0)^{i-1}, b_{i+1}, \ldots, b_n)^T
\]

(which is not in \( M \) because \( C(1) \neq C(2) \)) must create a new \( K_2 \)-submatrix, say in the \( x \)th and \( y \)th rows, some \( 1 \leq x < y \leq m \). Clearly, \( \{x, y\} \notin [i] \) because each column of \( M([i], [m]) \) is either \( (0)^{i}\) or \( (1)^{i}\). Also, it is impossible that \( x \in [i] \) and \( y \in [i + 1, n] \) because then, for some \( a_1, a_2 \in [m] \), \( M(y, a_1) = M(y, a_2) = 1 - C(y) = 1 - b_y \), \( M(1, a_1) = 1 - M(1, a_2) \) and we can see that \( K_2 \) is isomorphic to \( M(\{x, y\}, \{a_1, a_2, g_y, h_y\}) \), which contradicts \( K_2 \not\subseteq M(\{x, y\}, \ldots, n) \). So we have to assume that \( i < x < y \leq n \). As \( M([i + 1, l], 1) = C([i + 1, l]), y > l \).

As \( K_2 \not\subseteq M(\{x, y\}, \ldots, n) \), no column of \( M(\{x, y\}, [m]) \) can equal \( C(\{x, y\}) = (b_x, b_y)^T \); in particular, \( M(y, g_x) = M(y, h_x) = 1 - b_y \) (as \( M(x, y) = M(x, h_x) = b_x \)). But then

\[
K_2 \cong M(\{1, y\}, \{g_x, h_x, g_y, h_y\}),
\]

which is a contradiction proving our theorem. \( \square \)

Theorem 3.2 yields that \( \text{sat}(n, K_2) = \text{forb}(n, K_2) = n + 1 \) which, in our opinion, is rather surprising. A greater surprise is yet to come as we are going to show now that \( \text{sat}(n, K_3) \) is constant for \( n \geq 4 \).

**Theorem 3.3** For \( K_3 \) the following holds:

\[
\text{sat}(n, K_3) = \begin{cases} 
7, & \text{if } n = 3, \\
10, & \text{if } n \geq 4.
\end{cases}
\]
Proof. The claim is trivial for \( n = 3 \), so assume \( n \geq 4 \). A computer search \[10\] revealed that

\[
\text{sat}(4, K_3) = \text{sat}(5, K_3) = \text{sat}(6, K_3) = \text{sat}(7, K_3) = 10,
\]

which suggested that \( \text{sat}(n, K_3) \) is constant. An example of a \( K_3 \)-saturated \( 6 \times 10 \)-matrix is the following.

\[
M = \\
\begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
\end{bmatrix}.
\]

It is possible (but very boring) to check by hand that \( M \) is indeed \( K_3 \)-saturated as is, in fact, any \( n \times 10 \)-matrix \( M' \) obtained from \( M \) by duplicating any row, cf. Theorem 2.1. (The symmetries of \( M \) shorten the verification.) A \( K_3 \)-saturated \( 5 \times 10 \)-matrix can be obtained from \( M \) by deleting one row (any). For \( n = 4 \), we have to provide a special example:

\[
M = \\
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
\end{bmatrix}.
\]

So \( \text{sat}(n, K_3) \leq 10 \) for each \( n \geq 4 \) and, to prove the theorem, we have to show that no \( K_3 \)-saturated matrix \( M \) with at most 9 columns and at least 4 rows can exist. Let us assume the contrary.

Claim 1. Any row of \( M \in \text{SAT}(n, K_3) \) necessarily contains at least four 0’s and at least four 1’s, for \( n \geq 4 \).

Suppose, on the contrary to the claim, that the first row \( M(1, \cdot) \) contains only 3 zeros, say in the first three columns. (By Lemma 3.1 we must have at least 3 zeros.) If we replace the \( i \)-th of these zeros by 1, \( i \in [3] \), then the obtained column \( C_i \), if added to \( M \), does not create any \( K_3 \)-submatrix. Indeed, the first row of \([M, C_i]\) contains at most three 0’s while \( C_i([2, n]) \) is a column of \( M([2, n]) \). As \( M \) is \( K_3 \)-saturated, \( C_1, C_2 \) and \( C_3 \) are columns of \( M \). These columns differ only in the first entry from \( M(1, \cdot), M(2, \cdot) \) and \( M(3, \cdot) \) respectively. Therefore, for each \( A \in [2, n]^{(3)} \), the matrix \( M(A, \cdot) \) can contain at most \( e(M) - 3 \leq 6 \) distinct columns. But then any column \( C \) which is not the column of \( M \) and has the leading entry 1 (\( C \) exists as \( n \geq 4 \)), contradicts the \( K_3 \)-saturation of \( M \) as the first row of \([M, C]\) contains at most 3 zeros. This contradiction proves Claim 1.

Therefore, \( e(M) \) is either 8 or 9. As we are free to complement the rows, we may assume that each row of \( M \) contains exactly four 1’s. Call \( A \in [n]^{(3)} \) (and also \( M(A, \cdot) \)) nearly complete if \( M(A, \cdot) \) has 7 distinct columns.
Claim 2. Any nearly complete $M(A,)$ contains $(0, 0, 0)^T$ as a column.

Indeed, otherwise $M(A,)$ $\supseteq T_3^2$ which already contains four 1’s in each row; this implies that the (one or two) remaining columns must contain zeros only. This implies that $M(A,)$ $\supseteq K_3$, which is a contradiction.

Claim 3. Every nearly complete $M(A,)$ contains $T_3^1$ as a submatrix.

Indeed, if $(0, 0, 1)^T$ is the missing column of $M(A,)$, then some 7 columns span a copy of $K_3 \setminus (0, 0, 1)^T$. By counting 1’s in the rows we deduce that the remaining column(s) must have exactly one non-zero entry and one of them equals $(0, 0, 1)^T$, which is a contradiction.

Now fix any nearly complete $A$ which exists by the $K_3$-saturation of $M$. Assume that $A = [3]$ and that the first 7 columns of $M([3],)$ are distinct. We know that the 3-column missing from $M([3],[7])$ has at least two 1’s.

If $(1, 1, 1)^T$ is missing, then $M([3],[7])$ contains exactly three ones in each row, so the remaining column(s) of $M$ must contain an extra 1 in each row. As $(1, 1, 1)^T$ is the missing column, we conclude that $e(M) = 9$ and the 8th and 9th columns of $M([3],)$ are, up to a row permutation, $(0, 0, 0)^T$ and $(1, 1, 0)^T$. This implies that $M([3],)$ contains the column $(0, 0, 0)^T$ only once. On the other hand, by Claims 2 and 3, $M$ must contain the columns $((0)^n)^T$ and $((0)^{n-1}, 1)^T$ (as they cannot create $K_3$) whose first three entries are zeros, which is a contradiction.

Similarly, if $(1, 1, 0)^T$ is missing, one can deduce that, up to a row permutation, $M([3],)$ contains 7 distinct columns plus the columns $(1, 0, 0)^T$ and $(0, 1, 0)^T$ and, again, the multiplicity of $(0, 0, 0)^T$ is only one, which is a contradiction as above, completing the proof of the theorem.

We do not have any non-trivial results concerning $K_k$, $k \geq 4$ except that a computer search [10] showed that $\text{sat}(5, K_4) = 22$ and $\text{sat}(6, K_4) \leq 24$. (We do not know if a $K_4$-saturated $6 \times 24$-matrix discovered by a partial search is minimal.)

Problem 3.4 For which $k \geq 4$, $\text{sat}(n, K_k) = O(1)$?

4 Forbidding Small Matrices

Here we will try to compute $\text{sat}(n, F)$ for forbidden matrices with at most 3 rows.

4.1 Forbidding 1-Row Matrices

For any given 1-row matrix $F$, we can determine $\text{sat}(n, F)$ for all but finitely many values of $n$:
Theorem 4.1 Let $F = ((0)^m, (1)^l) = [mT_0^n, lT_1^n]$ with $l \geq m$. Then, for $n \geq l - 1$,

$$\text{sat}(n, F) = \begin{cases} 
1, & \text{if } l = 1 \text{ and } n \geq 1, \\
2, & \text{if } m = 0 \text{ and } l = 2, \\
l + 1, & \text{if } m = 0 \text{ and } l \geq 3, \\
l + m - 1, & \text{if } m \geq 1 \text{ and } l \geq 2.
\end{cases}$$

Proof. Assume that $l \geq 2$, as the case $l = 1$ is trivial.

Suppose that $m = 0$. The case $l = 2$ is trivial, so assume $l \geq 3$. An example of $M \in \text{SAT}(n, F)$ can be built for $n \geq l - 1$ by taking $T_0^n$ plus $T_1^n$ plus $\chi_{[n]\{i\}}, i \in [l-2]$, and $\chi_{[l-2]}$. Clearly, each row of $M \supseteq T_0^n$ has exactly $l - 1$ ones so $M$ is $F$-saturated and the upper bound follows.

On the other hand, suppose on the contrary that some $F$-saturated matrix $M$ has $n \geq l - 1$ rows and $c \leq l$ columns. As $c < 2^n$ and $M$ contains the all-0 column, $c = l$ and some row $M(i, \cdot)$ contains at $l - 1$ ones. As we are not allowed multiple columns in $M$, some other row, say $M(j, \cdot)$, has at most $l - 2$ ones. Then $\chi_{(j)}$ is not a column of $M$ because its $i$th entry is zero but its addition does not create $l$ ones in a row. This contradiction establishes the case $m = 0$.

For $m > 1$, let $M$ consist of $T_0^n$ plus $\chi_{\{i\}}, i \in [m-2]$, plus $\chi_{[n]\{i\}}, i \in [l-1]$ and $\chi_{[m-1,l-1]}$. Clearly, each row of $M$ contains $l$ ones and $m - 1$ zeros so any new column (which must contain at least one 0) creates an $F$-submatrix and the upper bound follows. The lower bound is trivial. \hfill \Box

Remark 4.2 The case when $n \leq l - 2$ in Theorem 4.1 seems messy so we do not investigate it here.

4.2 Forbidding 2-Row Matrices

Now let us consider some particular 2-row matrices.

Let $F = lT_2^n$, that is, $F$ consists of the column $(1, 1)^T$ taken $l$ times. Trivially, for $l = 1$ or 2, $\text{sat}(n, lT_2^n) = n + l$ with $T_2^{\leq 1}$ and $[T_2^{\leq 1}, T_2^n]$ being the only extremal matrices. For $l \geq 3$, we can only show the following lower bound which is almost sharp for $l = 3$ when we can build a $3T_2^n$-saturated $n \times (2n + 2)$-matrix by taking $T_2^{\leq 1}$, $\chi_{[n-1]}$, $\chi_{[n]}$, plus $\chi_{\{i\}}$ for $i \in [n-1]$.

Lemma 4.3 For $l \geq 3$ and $n \geq 3$, $\text{sat}(n, lT_2^n) \geq 2n + 1$.

Proof. Let $M = [T_2^{\leq 1}, M']$ be $lK_2^n$-saturated. Note that $M'$ must have the property that every column $\chi_A$ with $A \in [n]^{(2)}$ either belongs to $M'$ or creates an $F$-submatrix; in the latter case, exactly $l - 1$ columns of $M'$ have ones in both positions of $A$. Therefore, adding to $M'$ some columns of $T_2^{\leq 1}$ (possibly multiple), we can obtain a new matrix $M''$ such that, for every $A \in [n]^{(2)}$, $M''(A, \cdot)$ contains the column $(1, 1)^T$ exactly $l - 1$ times. If we let set $X_i$ be encoded by the $i$th row of $M''$ as its characteristic
vector, we have that \(|X_i \cap X_j| = l - 1\) for every \(1 \leq i < j \leq n\). The result of Bose [8] (see [16] Theorem 14.6), which can be viewed as an extension of the famous Fisher’s inequality [12], asserts that the rows of \(M''\) are linearly independent over the reals or \(M''\) has two equal rows, say \(X_i = X_j\). The last case is impossible because then \(|X_i| = l - 1\) and each other \(X_h\) contains \(X_i\) as a subset; this in turn implies that the column \((1^n)^T\) appears at least \(l - 1 \geq 2\) times in \(M''\) and (since \(n \geq 3\)) the same number of times in \(M'\), a contradiction. Thus the rank of \(M''\) over the reals is \(n\). Since every column added to \(M'\) when we were constructing \(M''\) was already present in \(M'\), the matrices \(M'\) and \(M''\) has the same rank over the reals. Thus \(M'\) has at least \(n\) columns and the lemma follows. ~

Let us show that Lemma 4.3 is sharp for some \(n\). Suppose there exists a symmetric \((n, k, 2)\)-design (meaning we have \(n\) \(k\)-sets \(X_1, \ldots, X_n \in [n]^{(k)}\) such that every pair \(\{i, j\} \in [n]^{(2)}\) is covered by exactly two \(X_i\)’s). Let \(M\) be the \(n \times n\)-matrix whose rows are the characteristic vectors of the sets \(X_i\). Then \([T_n^{\leq 3}, M]\) is a \(3T_2^2\)-saturated \(n \times (2n + 1)\)-matrix. For \(n = 4\), we can take all 3-subsets of \([n]\). For \(n = 7\), we can take the family \(\{[7] \setminus Y_i : i \in [7]\}\), where \(Y_1, \ldots, Y_7 \in [7]^{(3)}\) form the Fano plane. Constructions of such designs for \(n = 16, 37, 56,\) and 79 can be found in [9, Table 6.47].

Of course, the non-existence of a symmetric \((n, k, 2)\) design does not imply anything about sat\((n, 3T_2^2)\), since a minimum \(3T_2^2\)-saturated matrix \([T_n^{\leq 3}, M]\) need not have the same number of ones in the rows of \(M\).

Lemma 4.3 is not always optimal for \(l = 3\). One trivial example is \(n = 3\). Another one is \(n = 5\):

**Lemma 4.4** sat\((5, 3T_2^2) = 12.\)

*Proof.* Suppose on the contrary that we have a \(3T_2^2\)-saturated \(5 \times 11\)-matrix \([T_5^{\leq 3}, M]\). Let \(X_1, \ldots, X_5\) be the subsets of \([5]\) encoded by the rows of \(M\). Then \(|X_i \cap X_j| = 2\) for \(1 \leq i < j \leq 5\). If \(X_i \subseteq X_j\) for distinct \(i, j \in [5]\), then \(|X_i| = 2\), every other \(X_h\) contains \(X_i\) as a subset, and \(M\) has two equal columns, a contradiction. In particular, \(3 \leq |X_i| \leq 4\) for every \(i \in [5]\). A simple case analysis gives a contradiction by assuming that each \(|X_i| = 3\). Finally, if some \(|X_i| = 4\), say \(X_1 = [4]\), then each of \(X_2, \ldots, X_5\) contains 5 and some two elements of \([4]\), and we can easily derive a contradiction. ~

**Problem 4.5** Determine sat\((n, 3T_2^2)\) for every \(n\).

**Remark 4.6** It is interesting to note that if we let \(F = [IT_2^2, (0, 1)^T]\) then the sat-function is bounded by a constant. Indeed, complete \(M' = [\chi_{[n]\setminus\{i\}}]_{i \in [n]}\) to an arbitrary \(F\)-saturated matrix \(M\). Clearly, in any added column all entries after the \(l\)th position are either zeros or ones; hence sat\((n, F) \leq 2 \cdot 2^l.\)

It is easy to compute sat\((n, T_2^1)\) by observing that the \(n\)-row matrix \(M_Y\) whose columns encode \(Y \subseteq 2^{[n]}\) is \(T_2^1\)-free if and only if \(Y\) is a chain, that is, of each two
members of $Y$ one is a subset of the other. Thus $M_Y$ is $T^*_2$-saturated if and only if $Y$ is a maximal chain without repeated entries. As all maximal chains in $2^{[n]}$ have size $n + 1$, we conclude that

$$\text{sat}(n, T^*_2) = \text{forb}(n, T^*_2) = n + 1, \quad n \geq 2.$$ 

**Theorem 4.7** Let $F = [T^0_2, T^2_2] = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$. Then $\text{sat}(n, F) = 3, n \geq 2$.

**Proof.** For $n \geq 3$, the matrix $M$ consisting of the columns $(0, 1, (1)^{n-2})^T, (1, 0, (1)^{n-2})^T$ and $(0, 0, (1)^{n-2})^T$ can be easily verified to be $F$-saturated and the upper bound follows.

Since $n = 2$ is trivial, let $n \geq 3$. Any 2-column $F$-free matrix $M \not\subseteq T^*_2$ is, without loss of generality, the following: we have rows $(0, 0), (1, 1), (1, 0)$ and $(0, 1)$ occurring, in this order, $n_{00} \geq 0, n_{11} \geq 1, n_{10} \leq 1$ and $n_{01} \leq 1$ times respectively. But then the addition of a new column $((0)^{n_{00}+1}, 1, 1, \ldots)^T$ cannot create an $F$-submatrix. □

**Theorem 4.8** Let $F = T^*_2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$. Then $\text{sat}(n, F) = \text{forb}(n, F) = n + 1, \quad n \geq 2$.

**Proof.** Clearly, $\text{forb}(n, F) \leq \text{forb}(n, K_2) = n + 1$.

Next, suppose that some $M \in \text{SAT}(n, F)$ has two equal rows, for example, $M(1, i) = M(2, i) = ((1)^i, (0)^m)$. Let $X = [l]$ and $Y = [l + 1, l + m]$. Define

$$A_i = \{j \in [l + m] : M(i, j) = 1\}, \quad i \in [n].$$

(For example, $A_1 = A_2 = X$.) As $M$ is $F$-free, for every $i, j \in [n]$, the sets $A_i$ and $A_j$ are either disjoint or one is a subset of the other. For $i \in [3, n]$, let $b_i = 1$ if $A_i$ strictly contains $X$ or $Y$ and let $b_i = 0$ otherwise (then $A_i$ is contained in $X$ or $Y$). Let $b_1 = 1$ and $b_2 = 0$.

Clearly, $C = (b_1, \ldots, b_n)^T$ is not a column of $M$ so its addition creates a forbidden submatrix, say $F \subseteq [M, C](\{i, j\}, )$. Of course, $b_i = b_j = 0$ is impossible as $(0, 0)^T \not\subseteq F$. If $b_i = b_j = 1$ then necessarily $A_i \cap A_j \neq \emptyset$ and $M(\{i, j\}) \supseteq (1, 1)^T$ contains $F$. Finally, if $b_i \neq b_j$, e.g., $b_i = 0, b_j = 1$ and $i < j$, then $A_i \supseteq A_j$ (as $(0, 1)^T$ cannot be a column of $M(\{i, j\})$), which implies $A_i = A_j$; but then we do not have a copy of $F$ as $(1, 0)^T$ is missing.

Thus, no $F$-saturated matrix $M$ cannot have two equal rows and, by Theorem 2.1, $\text{sat}(n, F) \geq n + 1$ for any $n$. □

Trivially, $\text{sat}(n, [(0, 1)^T, T^*_2]) = 2$ so we know the sat-function for any simple 2-row matrix.
4.3 Forbidding 3-Row Matrices

Here we consider some particular 3-row matrices. First we solve completely the case when \( F = [T_3^0, T_3^3] \).

**Theorem 4.9** Let \( F = [T_3^0, T_3^3] = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \). Then

\[
\text{sat}(n, F) = \begin{cases} 
7, & \text{if } n = 3 \text{ or } n \geq 6, \\
10, & \text{if } n = 4 \text{ or } 5.
\end{cases}
\]

**Proof.** For the upper bound we define the following family of matrices.

\[
M_4 = \begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1
\end{bmatrix}
\]

\[
M_5 = \begin{bmatrix}
1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{bmatrix}
\]

\[
M_6 = \begin{bmatrix}
1 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1
\end{bmatrix}
\]

For any \( n \geq 7 \) let \( M_n([6],) = M_6 \) and \( M_n(i, ) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \) for every \( 7 \leq i \leq n \). A computer search [10] showed that \( \text{sat}(n, F) = e(M_n) \) for \( 3 \leq n \leq 10 \). It remains to show that

\[
\text{sat}(n, F) \geq 7 \tag{6}
\]

for \( n \geq 11 \). In order to see this, we show the following result first.

---

**Claim.** If \( M \) is an \( F \)-saturated matrix of size \( n \times m \) with \( n \geq 11 \) and \( m \leq 6 \) then \( M \) contains a row with all zero entries or with all one entries.

Suppose on the contrary that we have a counterexample \( M \). We may assume that the first 6 entries of the first column of \( M \) are equal to 0. Consider a matrix \( A = M([6],\lbrace 2, \ldots, m \rbrace) \). Note that every column of \( A \) contains at most two entries equal to 1, otherwise \( M([6],) \supseteq F \). Hence, the number of 1’s in \( A \) is at most \( 2(m−1) \).
By our assumption, each row of $A$ has at least one entry 1. Since $2(m - 1) < 12$, $A$ has a row with precisely one entry equal to 1. We may assume that $A(1, 1) = 1$ and $A(1, i) = 0$ for $2 \leq i \leq m$. Let $C_2$ be the second column of $M$ (remember that $C_2(1) = A(1, 1) = 1$).

Consider the $n$-column $C_3 = [0, C_2(\{2, \ldots, n\})^T]^T$ which is obtained from $C_2$ by changing the first entry to 0. If it is not in $M$, then $F \subseteq [M, C_3]$. This copy of $F$ has to contain the entry in which $C_3$ differs from $C_2$. But the only non-zero entry in Row 1 is $M(1, 2)$; thus $F \subseteq [C_2, C_3]$, which is an obvious contradiction. Thus we may assume that $C_3$ is the third column of $M$.

We have to consider two cases. First, suppose that $C_2(\{2, \ldots, n\})$ has at least one entry equal to 1. Without loss of generality, assume that $C_2(2) = C_3(2) = 1$.

It follows that $C_2(i) = C_3(i) = 0$ for $3 \leq i \leq 6$ (otherwise the first and the second columns of $M$ would contain $F$). Let

$$B = M(\{3, 4, 5, 6\}, \{4, \ldots, m\}).$$

By our assumption, each row of $B$ has at least one 1; in particular $m \geq 4$. Clearly, $B$ contains at most $2(m - 3) < 8$ ones. Thus, by permuting Rows 3, \ldots, 6 and Columns 4, \ldots, $m$, we can assume that $B(1, 1) = 1$ while $B(1, i) = 0$ for $2 \leq i \leq m-3$. Let $C_4$ be the fourth column of $M$ and $C_5$ be such that $C_4$ and $C_5$ differ at the third position only, \textit{i.e.}, $C_4(3) = 1$ and $C_5(3) = 0$. As before, $C_5$ must be in $M$, say it is the fifth column. Since $C_4(\{4, 5, 6\})$ has at most one 1, assume that $C_4(5) = C_4(6) = C_5(5) = C_5(6) = 0$. We need another column $C_6$ with $C_6(5) = C_6(6) = 1$ (otherwise the fifth or the sixth row of $M$ would consist of all zero entries). In particular, $m = 6$. But now the new column $C_7$ which differs from $C_6$ at the fifth position only (\textit{i.e.} $C_7(5) = 0$ and $C_7(i) = C_6(i)$ for $i \neq 5$) should be also in $M$, since $M$ is $F$-saturated. This contradicts $e(M) = 6$. Thus the first case does not hold.

In the second case, we have $C_2(i) = C_3(i) = 0$ for every $2 \leq i \leq 6$. We may define $B$ as in (7) and get a contradiction in the same way as above. This proves the claim.

Suppose on the contrary to the theorem that we can find an $F$-saturated matrix $M$ with $n \geq 11$ rows and $m \leq 6$ columns. By the claim, $M$ has a constant row, say $M^T = [N^T, (0)^{\{m\}}]^T$. If $C$ is an $(n-1)$-column missing from $N$, then the column $Q = (C^T, 1)^T$ is missing in $M$. Moreover, a copy of $F$ in $[M, Q]$ cannot use the $n$-th row. Thus $F \subseteq [N, C]$, which means that $N \in \text{SAT}(n-1, F)$ and $\text{sat}(n-1, F) \leq m \leq 6$. Repeating this argument, we eventually conclude that $\text{sat}(10, F) \leq 6$, a contradiction to the results of our computer search. The theorem is proved. \hfill \Box

**Theorem 4.10** Let $F = [T_3^0, T_3^2, T_3^3] = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}$. Then

$$\text{sat}(n, F) = \begin{cases} 7, & \text{if } n = 3, 6 \text{ or } 7, \\ 9, & \text{if } n = 4 \text{ or } 5. \end{cases}$$

14
Moreover, for any $n \geq 8$, $\text{sat}(n, F) \leq 7$.

Proof. For $n = 3, \ldots, 7$ the statement follows from a computer search \[10\] with the following $F$-saturated matrices.

$$M_4 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$M_5 = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$M_6 = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

For any $n \geq 7$ let $M_n([6],) = M_6$ and $M_n(i,.) = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$ for every $7 \leq i \leq n$ (i.e. the last row of $M_6$ is repeated $(n-6)$ times). It remains to show that $M_n$, $n \geq 8$, is $F$-saturated. Clearly, this is the case, since $M_7$ is $F$-saturated and $F$ contains no pair of equal rows. \[\square\]

**Conjecture 4.11** Let $F = [T_3^0, T_3^2, T_3^3]$. Then $\text{sat}(n, F) = 7$ for every $n \geq 8$.

**Theorem 4.12** Let $F = T_3^{\leq 2} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$. Then

$$\text{sat}(n, F) = \begin{cases} 7, & \text{if } n = 3, \\ 10, & \text{if } 4 \leq n \leq 6. \end{cases}$$

Moreover, for any $n \geq 7$, $\text{sat}(n, F) \leq 10$.

Proof. For $n = 3, \ldots, 6$ the statement follows from a computer search \[10\] with the following $F$-saturated matrices.

$$M_4 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$
Moreover, we assume that every column of matrix \( M \)

\[
M_5 = \begin{bmatrix}
  1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
  0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
  0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\
  0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
  0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 
\end{bmatrix}
\]

For any \( n \geq 6 \) let \( M_n([5],) = M_5 \) and \( M_n(i,) = [1 1 0 0 0 1 0 1 1] \) for every \( 6 \leq i \leq n \). It remains to show that \( M_n, n \geq 7 \), is \( F \)-saturated. Clearly, this is the case, since \( M_6 \) is \( F \)-saturated and \( F \) contains no pair of equal rows.

Conjecture 4.13 Let \( F = T_3^{\leq 2} \). Then \( sat(n, F) = 10 \) for every \( n \geq 7 \).

Theorem 4.14 Let \( F_1 = T_3^2 \) and \( F_2 = [T_3^2, T_3^3] \). Then \( sat(n, F_1) = sat(n, F_2) = 3n - 2 \) for any \( 3 \leq n \leq 6 \). Moreover, for any \( n \geq 7 \), \( sat(n, F_1) \leq 3n - 2 \) and \( sat(n, F_2) \leq 3n - 2 \) as well.

Proof. Let \( M_n = [T_n^0, T_n^1, T_n^2, T_n^3] \), where \( T_n^2 \subseteq T_n^2 \) and consists of all those columns of \( T_n^2 \) which have precisely one entry equals 1 either in the first or in the \( n \)th row, e.g., for \( n = 5 \) we obtain

\[
M_5 = \begin{bmatrix}
  0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
  0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
  0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
  0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 
\end{bmatrix}
\]

Clearly, \( e(M_n) = e(T_n^0) + e(T_n^1) + e(T_n^2) + e(T_n^3) = 1 + n + 1 + 2n - 4 = 3n - 2 \). Moreover, since \( T_n^2 \) is \( F_1 \)-admissible we get that \( M_n \) is both \( F_1 \) and \( F_2 \) admissible. Now we show that \( M_n \) is \( F_1 \)-saturated (consequently, \( M_n \) is also \( F_2 \)-saturated). Indeed, pick any column \( C \) which is not present in \( M_n \). Such a column must contain at least 2 ones and 1 zero. Let \( 1 \leq i, j, k \leq n \) be the indices of \( C \) so that \( c_i = 0, c_j = c_k = 1 \). If \( i = 1 \) or \( i = n \), then the matrix \( M_n(i,j,k) \) contains \( F_1 \). Otherwise, \( c_1 = c_n = 1 \), and there also exists \( 1 < i < n \) such that \( c_i = 0 \). Here \( M_n((1,i,n)) \) contains \( F_1 \). Hence, \( sat(n, F_1) \leq 3n - 2 \) and \( sat(n, F_2) \leq 3n - 2 \) for any \( n \geq 3 \). A computer search \([10]\) yields that these inequalities are equalities when \( n = 3, \ldots, 6 \).

Conjecture 4.15 Let \( F_1 = T_3^2 \) and \( F_2 = [T_3^2, T_3^3] \). Then \( sat(n, F_1) = sat(n, F_2) = 3n - 2 \) for every \( n \geq 7 \).

Remark 4.16 It is not hard to see that \( sat(n, F_1) \geq n + c \sqrt{n} \) for some absolute constant \( c \) and all \( n \geq 3 \). Indeed, let \( M \) be an \( n \times (n + 2 + \lambda) \) \( F_1 \)-saturated matrix of size \( sat(n, F_1) \) for some \( \lambda = \lambda(n) \). We may assume that \( M([n + 2]) = [T_n^0, T_n^1, T_n^2, T_n^3, \ldots, T_n^{[\lambda]} \]}. Moreover, we assume that every column of matrix \( M([\lambda],\{n + 3, \ldots, n + 2 + \lambda\}) \)
contains at least one entry equal to 1 (there must be a permutation of the rows of $M$ satisfying this requirement). We claim that all rows of $M(\{\lambda+1, \ldots, n\}, \{n+3, \ldots, n+2+\lambda\})$ are different. Suppose not. Then, there are indices $\lambda+1 \leq i, j \leq n$ such that $M(i, \{n+3, \ldots, n+2+\lambda\}) = M(j, \{n+3, \ldots, n+2+\lambda\})$. Now consider a column $C$ in which the only nonzero entries correspond to $i$ and $j$. Clearly, $C$ is not present in $M$, since the first $\lambda$ entries of $C$ equal 0. Moreover, since $M$ is $F_1$-saturated, the matrix $[M, C]$ contains $F_1$. In other words, there are three rows in $M$ which form $F_1$ as a submatrix. Note that the $i$th and $j$th row must be among them. But this is not possible since $F_1$ has no pair of equal rows.

Let $M_0 = M(\{\lambda+1, \ldots, n\}, \{n+3, \ldots, n+2+\lambda\})^T$. Clearly, $M_0$ is $F_1$ admissible. Anstee and Sali showed (see Theorem 1.3 in [3]) that $\text{forb}(\lambda, F_1) = O(\lambda^2)$. That means that $n - \lambda = O(\lambda^2)$, and consequently, $\lambda = \Omega(\sqrt{n})$. Hence, $\text{sat}(n, F_1) = e(M) \geq n + \Omega(\sqrt{n})$, as required.

References

[1] R. P. Anstee, Forbidden configurations: Induction and linear algebra, Europ. J. Combin. 16 (1995), 427–438.

[2] R. P. Anstee, J. R. Griggs, and A. Sali, Small forbidden configurations, Graphs Combin. 13 (1997), 97–118.

[3] R. P. Anstee and A. Sali, Small forbidden configurations IV: The 3 rowed case, Combinatorica 25 (2005), no. 5, 503–518.

[4] R.P. Anstee and Z. Füredi, Forbidden submatrices, Discrete Math. 62 (1986), 225–243.

[5] B. Bollobás, Combinatorics, Set Systems, Families of Vectors, and Combinatorial Probability, Cambridge Univ. Press, 1986.

[6] B. Bollobás, Extremal graph theory, Handbook of Combinatorics (R. L. Graham, M. Grötschel, and L. Lovász, eds.), Elsevier Science B.V., 1995, pp. 1231–1292.

[7] J. A. Bondy, Induced subsets, J. Combin. Theory (B) 12 (1972), 201–202.

[8] R. C. Bose, A note on Fisher’s inequality for balanced incomplete block designs, Ann. Math. Statist. 20 (1949), 619–620.

[9] C. J. Colbourn and J. H. Dinitz (eds.), Handbook of combinatorial designs, second ed., Discrete Mathematics and its Applications (Boca Raton), Chapman & Hall/CRC, Boca Raton, FL, 2007.

[10] A. Dudek, O. Pikhurko, and A. Thomason, On minimum saturated matrices (it includes the C source code of satmat.c), e-print arXiv:math/xxxxxxx.
[11] P. Erdős, A. Hajnal, and J. W. Moon, *A problem in graph theory*, Amer. Math. Monthly 71 (1964), 1107–1110.

[12] R. A. Fisher, *An examination of the possible different solutions of a problem in incomplete blocks*, Ann. Eugenics (London) 10 (1940), 52–75.

[13] P. Frankl, *On the trace of finite sets*, J. Combin. Theory (A) 34 (1983), 41–45.

[14] P. Frankl, Z. Füredi, and J. Pach, *Bounding one-way differences*, Graphs Combin. 3 (1987), 341–347.

[15] Z. Füredi, *Turán type problems*, Surveys in Combinatorics, London Math. Soc. Lecture Notes Ser., vol. 166, Cambridge Univ. Press, 1991, pp. 253–300.

[16] S. Jukna, *Extremal combinatorics with applications to computer science*, Springer Verlag, 2001.

[17] O. Pikhurko, *The minimum size of saturated hypergraphs*, Combin. Prob. Computing 8 (1999), 483–492.

[18] N. Sauer, *On the density of families of sets*, J. Combin. Theory (A) 13 (1973), 145–147.

[19] S. Shelah, *A combinatorial problem: Stability and order for models and theories in infinitary languages*, Pac. J. Math 4 (1972), 247–261.

[20] A. Sidorenko, *What we know and what we do not know about Turán numbers*, Graphs Combin. 11 (1995), 179–199.

[21] V. N. Vapnik and A. Chervonenkis, *The uniform convergence of frequencies of the appearance of events to their probabilities (in Russian)*, Teor. Veroyatn. Primen. 16 (1971), 264–279.
C source code of satmat

/***************************************************************************/
*/
* Copyright (C) 2009 Andrzej Dudek, Oleg Pikhurko and Andrew Thomason
*/
* This program is free software: you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation, either version 3 of the License, or
* (at your option) any later version.
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
* You should have received a copy of the GNU General Public License
* along with this program. If not, see <http://www.gnu.org/licenses/>.
*
* satmat.c: this program finds the smallest number of columns (saturated
* matrix) with n rows without having an r-row matrix F
* compile:
*   % gcc -Wall -ansi -O4 -o satmat satmat.c
* compile w/ debug option:
*   % gcc -Wall -ansi -O4 -o satmat -DDEBUG satmat.c
* usage:
*   % ./satmat
*--------------------------------------------------------
* Authors: Andrzej Dudek   adudek@andrew.cmu.edu
*           Oleg Pikhurko   pikhurko@andrew.cmu.edu
*           Andrew Thomason A.G.Thomason@dpmms.cam.ac.uk
*--------------------------------------------------------
/******************************************************************************/
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include <string.h>

#define MAXN 10  /* max value of n */
#define MAXR 5   /* max value of r */
#define MAXNCR 252  /* should be MAXN choose MAXR */
#define MAXS 50   /* max size of strong sat set */
int sub_mask[1 << MAXN][MAXNCR]; /* the sub-column mask indexed by some r rows,
   e.g. n=5 r=3 subset 4 in colex is 10011
30th column is 11110 subcolumn = 110 = 6
so subcol[30][4] = ..111011111 */

int subs_left[MAXS][MAXNCR]; /* allowed sub-columns indexed by rows found so
far; e.g. if the first 18 columns under
consideration contain, in rows indexed by
subset 4, sub-columns of types 0,1,4,5
then subs_left[18][4] = ~110011 = ~51 */

int poss_cols[(1 << MAXN) * MAXS]; /* list of columns to try next */

int T[MAXR][1 << MAXR]; /* family of T_k^l matrices */
int F[MAXR][1 << MAXR]; /* forbidden matrix */
int colT[MAXR+2]; /* colT[l] is the first column of T_k^l in T */
int irrel[1 << MAXN]; /* no sub-column is in F */
int mask; /* column representation of matrix F */

void gen_T(int r)
{
    int rows[MAXR];
    register int i, l, col;

    /* init T */
    memset((void *)T, 0, MAXR*(1 << MAXR)*sizeof(int));
    colT[0] = 0;
    col = 1;
    for (i = 0; i < r; i++) { /* precisely (i+1) 1s per column */
        colT[i+1] = col;
        for (l = 0; l < i+1; l++) {
            rows[l] = l;
        }

        do {
            for (l = 0; l < i+1; l++) {
                T[rows[l]][col] = 1;
            }
            col++;
        } while (/* find next subset of i 1s */)
for (l = 0; l < i; l++) {
    if (rows[l] + 1 != rows[l + 1]) {
        break;
    }
}

rows[l] += 1;
while (--l >= 0) {
    rows[l] = l;
}
}
while (rows[i] < r);
}
colT[i+1] = col;
}

void gen_mask(int r, int c) {
    register int i, j, col;

#ifdef DEBUG
    printf("\n F mask\n");
#endif
    mask = 0;
    for (j = 0; j < c; j++) {
        for (col = 0, i = 0; i < r; i++) {
            if (F[i][j] == 1) {
                col |= (1<<i);
            }
        }
        mask |= (1 << col);
    }

#ifdef DEBUG
    printf(" mask[\%d] --> \%d\n", j, col);
#endif
}

void gen_irrel(int n, int min0, int min1, int *irr) {
    register int i,j,ones;

#ifdef DEBUG
    printf(" \n min0 --> \%d\n", min0);
    printf(" min1 --> \%d\n", min1);
#endif
}
for (*irr = 0, i = 0; i < (1 << n); i++) {
    for (ones = 0, j = 0; j < n; j++) { /* count 1s */
        if (i & (1 << j)) {
            ones++;
        }
    }

    if (ones < min1 || ones > n-min0) {
        irrel[(*irr)++] = i;
    }
}

#ifdef DEBUG
    printf("\n irrelevant columns --> ");
    for (i = 0; i < *irr; i++) {
        printf("%2d ", irrel[i]);
    }
    printf("\n");
#endif

int is_irrel(int col, int irr) {
    register int i;

    /* binary search could be more effective */
    for (i = 0; i < irr; i++) {
        if (irrel[i] == col) {
            return 1;
        }
    }

    return 0;
}

void input(int *n, int *target, int *r, int *c, int *irr) {
    int i,j,k,col,min0,min1;
    char ans;

    printf("\n****************************************************** Wellcome to SATMAT! ******************************************************\n");
    printf("This program checks if sat(n,F) <= target, for a given an r-row matrix F.\n");
    printf("If so, an n-row F-saturated matrix is produced.\n");
    printf("******************************************************\n");

    printf("\n input n --> ");
    scanf("%d", n);
    if (*n < 2 || *n > MAXN) {
        printf(" n is not in the range 2 to %d (MAXN)\n", MAXN);
        exit(0);
    }
printf("\n input r --> ");
scanf("%d", r);
if (*r < 2 || *r > MAXR) {
    printf(" r is not in the range 2 to %d (MAXR)\n\n", MAXR);
    exit(0);
}
if (*r > *n) {
    printf(" r is not in the range 2 to %d \n\n", *n);
    exit(0);
}

printf("\n target --> ");
scanf("%d", target);
if (*target < 2 || *target >= MAXS) {
    printf(" target is not in the range 2 to %d (MAXS-1)\n\n", MAXS - 1);
    exit(0);
}

printf("\n input F as a union of Ti matrices\n\n");
gen_T(*r);
for (i = 0; i < *r; i++) {
    for (j = 0; j <= *r; j++) {
        if (i == (*r-1)/2) {
            printf(" T%d=|", j);
        } else {
            printf(" |");
        }
        for (k = colT[i]; k < colT[i+1]; k++) {
            printf("%d", T[i][k]);
        }
        printf("|");
    }
    printf("\n");
}
col = 0;
min0 = min1 = *r;
for (i = 0; i <= *r; i++) {
    printf("\n add T%d (y/n) --> ", i);
    scanf(" %c", &ans);
    if (ans == 'y') { /* copy Ti into F */
        for (j = 0; j < *r; j++) {
            for (k = colT[i]; k < colT[i+1]; k++) {
                F[j][col+k-colT[i]] = T[j][k];
            }
        }
        col += (colT[i+1]-colT[i]);
    }
}
/ at least this number of 0s in very column*/ 
if (*r-i < min0) {
    min0 = *r-i;
}

/* at least this number of 1s in very column*/ 
if (i < min1) {
    min1 = i;
}
}
}
*c = col;

/* print F */
printf("\n");
for (i = 0; i < *r; i++) {
    if (i == (*r-1)/2) {
        printf(" F=|");
    } else {
        printf(" |");
    }
    for (j = 0; j < col; j++) {
        printf("%d", F[i][j]);
    }
    printf("|\n");
}

/* find F mask */
gen_mask(*r, *c);

/* find irreleavan columns */
gen_irrel(*n, min0, min1, irr);
}

void setup(int n, int r, int c, int *ncr)
{
    int rows[MAXR];
    register int l, col, subr, subcol;
    
    /* subset of r rows */
    for (l = 0; l < r; l++) {
        rows[l] = 1;
    }
    
    /* index of subset in colex */
    subr = 0;

do {
  /* for each column, extract subcolumn indexed by rows*/
  for (col = 0; col < (1 << n); col++) {
    for (subcol = 0, l = 0; l < r; l++) {
      if (col & (1 << rows[l])) {
        subcol |= (1 << l);
      }
    }
    sub_mask[col][subr] = (1 << (1 << r)) - 1 - (1 << subcol);
    sub_mask[col][subr] &= mask;
  }

  /* find next colex subset of rows */
  for (l = 0; l < r - 1; l++) {
    if (rows[l] + 1 != rows[l + 1]) {
      break;
    }
  }
  rows[l] += 1;
  while (--l >= 0) {
    rows[l] = l;
  }
  subr += 1;
}

while (rows[r - 1] < n); /* don’t do colex beyond n */

*ncr = subr; /* we stopped at subset n choose r */
fflush(stdout);
}

void print_sol(int *cols, int n, int l, int irr)
{
  int matrix[MAXN][MAXS];
  int i, j;

  /* columns from the solution */
  for (j = 0; j < l; j++) {
    for (i = n-1; i >= 0; i--) {
      matrix[i][j] = ((1<<i) & cols[j]) ? 1 : 0;
    }
  }

  /* irrelevant columns */
  for (j = 0; j < irr; j++) {
    for (i = n-1; i >= 0; i--) {
      matrix[i][l+j] = ((1<<i) & irrel[j]) ? 1 : 0;
    }
  }

  l += j;
}
for (i = 0; i < n; i++) {
    if (i == (n-1)/2) {
        printf(" M_{%d,%2d}=|", n, l);
    } else {
        printf(" |");
    }
    for (j = 0; j < l; j++) {
        printf("%d", matrix[i][j]);
    }
    printf("|\n");
} flush(stdout);
}

void find_sat(register int n, register int r, register int ncr, register int target, int irr) {
    int cols[MAXS];  /* current set of columns */
    int *ptr[MAXS], *to[MAXS];  /* pointers into poss_cols */
    register int l, m, subr, col;
    register int *p, *q, *s, *t, *u;

    register int strong = target + 1;  /* smallest strong set so far */
    register int examples = 0;

    int gone_critical[MAXNCR];  /* sub-columns which became critical */
    int *gc[MAXNCR];  /* pointers into gone_critical */
    int crit_at[MAXNCR];  /* when a sub-column went critical */

    time_t start, stop;
    double diff;
    #ifdef DEBUG
        long long counter = 0;
    #endif

    start = time(NULL);

    for (m = 0, col = 0; col < (1 << n); col++) {
        if (is_irrel(col,irr) == 0) {
            poss_cols[m++] = col;
        }
    }

    ptr[0] = poss_cols;
    to[0] = poss_cols + m;

    #ifdef DEBUG
        printf("\n number of relevant columns --> %d\n", m);
    #endif

fflush(stdout);
#endif

/* there is only one relevant column */
if (m == 1) {
    print_sol((int*)0, n, 0, irr);
    strong = 0;
    examples = 1;
}

gc[0] = gone_critical;
for (m = 0; m < ncr; m++) {
    crit_at[m] = -1;
}

l = 0;

/* we have set of l + 1 columns */
do { /* main loop */
    #ifdef DEBUG
        counter++;
    #endif

    cols[l] = *ptr[l]++;

    if (l + 1 >= strong || ptr[l] > to[l]) {
        l -= 1;
        if (l < 0) {
            break;
        }
        goto undo;
    }

    #ifdef DEBUG
        /* info */
        if (l < 5) {
            for (m = 0; m <= l; m++) {
                printf("%3x", cols[m]);
            }
            putchar('r');
            fflush(stdout);
        }
    #endif

    #ifdef DEBUG
        /* make up subs_left mask after column l */
        if (l == 0) {
            for (subr = 0; subr < ncr; subr++) {
                subs_left[0][subr] = sub_mask[cols[l]][subr];
            }
            p = subs_left[0];
        }
    #endif

    #endif
}
} else {
    p = subs_left[l - 1];
}

q = subs_left[l];
s = sub_mask[cols[l]];
t = p + ncr;
u = gc[l];

while (p < t) {
    m = *q++ = *p++ & *s++;
    if ((m & (m - 1)) == 0) { /* only bad subcolumn */
        m = ncr - 1 - (t - p);
        if (crit_at[m] < 0) { /* subcolumn not critical so far */
            crit_at[m] = l;
            *u++ = m;
        }
    }
}

gc[l + 1] = u;

p = ptr[l]; /* find columns which extend this set */
q = s = to[l];
t = gc[l + 1];
while (p < s) {
    m = *p++;
    u = gc[l]; /* check only subcolumns which went critical */
    while (u < t) {
        subr = *u++;
        if (((subs_left[l][subr] & sub_mask[m][subr]) == 0) {
            m = 0;
            break;
        }
    }
    if (m) {
        *q++ = m;
    }
}

ptr[l + 1] = to[l];
to[l + 1] = q;

/* if possible, go round again with larger l */
if (ptr[l + 1] < to[l + 1]) {
    1 += 1;
    continue;
}

/* if not, set is inextendible upwards; is it strongly saturated? */
subr = 0;

for (col=0; col < cols[0]; col++) {
    if (is_irrel(col, irr) == 1) {
        continue;
    }
    for (subr = 0; subr < ncr; subr++) {
        if ((subs_left[l][subr] & sub_mask[col][subr]) == 0) {
            break;
        }
    }
    if (subr == ncr) {
        break;
    }
}

if (subr < ncr) {
    for (m = 0; m < l; m++) {
        for (col = cols[m] + 1; col < cols[m + 1]; col++) {
            if (is_irrel(col, irr) == 1) {
                continue;
            }
            for (subr = 0; subr < ncr; subr++) {
                if ((subs_left[l][subr] & sub_mask[col][subr]) == 0) {
                    break;
                }
            }
            if (subr == ncr) {
                break;
            }
        }
        if (subr == ncr) {
            break;
        }
    }
    if (subr == ncr) {
        break;
    }
}

if (col == cols[l]) { /* we have a strong set size l + 1*/
    if (strong < l + 1) {
        continue; /* not best possible */
    }
    if (strong > l + 1) {
        strong = l + 1; /* best so far */
        examples = 0;
    }
    examples += 1;
    if (examples == 1) {
        
    }
print_sol(cols, n, strong, irr);

stop = time(NULL);
diff = difftime(stop, start);
printf("\n after --> %.1f min\n", diff/60);
fflush (stdout);
}
}

/* now go again with same l */
undo:

u = gc[l]; /* sub-cols critical at l no longer are */
t = gc[l + 1];
while (u < t) {
    crit_at[*u++] = -1;
}
}
while (l >= 0); /* main loop */

#endif DEBUG
printf("\n number of iterations --> %lld\n", counter);
#endif

stop = time(NULL);
diff = difftime(stop, start);
printf("\n time taken --> %.1f min\n", diff/60);

printf("\n******************* Solution *******************\n");
if (examples > 0) {
    printf("%28s sat(%d,F) = %d\n", ",n, strong+irr);
}
else {
    printf("%28s sat(%d,F) > %d\n", ", n, target+irr);
}
printf("******************************* Solution ********************\\n");

int main(int argc, char *argv[])
{
    int n, r, c, ncr, target, irr;

    input(&n, &target, &r, &c, &irr);
    setup(n, r, c, &ncr);
    find_sat(n, r, ncr, target-irr, irr);
    exit(0);
}