UPPER BOUNDS ON SATURATION FUNCTIONS FOR SEQUENCES

SHIHAN KANUNGO

Abstract. This paper explores the concept of saturation in combinatorial structures, with a particular focus on sequences. We investigate the saturation and extremal functions $\mathrm{Sat}(n, u)$ and $\mathrm{Ex}(n, u)$ for sequences as well as other varieties such as graphs and 0-1 matrices. We are interested in the growth rate of $\mathrm{Sat}(n, u)$; the main goal being to prove that $\mathrm{Sat}(n, u) = O(n)$ for all sequences u. This would resolve a conjecture posed by Anand, Geneson, Kaustav, and Tsai in 2021, where they show that $\mathrm{Sat}(n, u) = O(n)$ for two-letter sequences. Similar results have already been shown for graphs and for 0-1 matrices. We describe an algorithm to generate u-saturated sequences that can be easily implemented in code. Using this as a tool, we give an example of a three-letter sequence u, which despite having extremal function that grows faster than linear, still satisfies $\mathrm{Sat}(n, u) = O(n)$, and we conjecture that for any u, the sequences produced by this algorithm have length that grows at most linearly with n, which implies the original conjecture.

1. Introduction

The concept of pattern avoidance lies at the heart of extremal combinatorics, which investigates questions such as:

What is the largest possible structure that avoids a given substructure?

Well-known examples include Ramsey theory and the forbidden subgraph problem. Another important example is the study of **Davenport-Schinzel sequences**, which are sequences S over an alphabet of n distinct letters that satisfy the following constraints:

- For any two distinct letters a, b, the alternating subsequence $abab\cdots$ of length $s + 2$ is forbidden, where s is a nonnegative integer.
- No two consecutive letters in S are the same.

A key quantity of interest is the function $\lambda_s(n)$, which denotes the length of the longest Davenport-Schinzel sequence over an n-letter alphabet. For $s \geq 3$, $\lambda_s(n)$ grows as n times a small but nonconstant factor. For instance,

$$\lambda_3(n) = \Theta(n\alpha(n)),$$

where $\alpha(n)$ is the inverse Ackermann function, known for its extremely slow growth (see [9] for a definition of the inverse Ackermann funtion). Originally introduced to analyze linear differential equations, Davenport-Schinzel sequences and the function $\lambda_s(n)$ have since found numerous applications in discrete geometry and geometric algorithms.

A **generalized Davenport-Schinzel sequence** extends this concept by requiring sequences to avoid a given pattern u on r distinct letters while also being r**-sparse**, meaning every consecutive r letters are pairwise distinct. Analogous to $\lambda_s(n)$, we define the **extremal function** $\mathrm{Ex}(n, u)$, which represents the longest r-sparse sequence on n letters that avoids u.

Date: March 8, 2025.

The growth of $\mathrm{Ex}(n, u)$ has been studied in [9], and generalized Davenport-Schinzel sequences have many significant applications.

This project focuses on a related function, the **saturation function** $\mathrm{Sat}(n, u)$. Saturation functions have been extensively studied in other combinatorial settings, including graphs [3, 7], posets [4], and 0-1 matrices [5]. The notion of saturation for sequences was introduced in [1], where several fundamental results were also established. Specifically, given a forbidden sequence u with r distinct letters, we say that a sequence s on a given alphabet is **u-saturated** if s is r-sparse, u-free, and adding any letter from the alphabet to an arbitrary position in s violates r-sparsity or induces a copy of u. Notably, it was shown that if u consists of only two distinct letters, then $\mathrm{Sat}(n, u) = O(1)$ or $\Theta(n)$, leading to the open question of whether this dichotomy extends to arbitrary sequences u. In fact, proving that $\mathrm{Sat}(n, u) = O(n)$ for all sequences u is sufficient to establish this dichotomy. Thus, we aim to address the following conjecture:

Conjecture 1 ([1])**.** *We have* $\mathrm{Sat}(n, u) = O(n)$ *for any sequence* u*.*

Similar dichotomies have been proven in other settings, such as for graphs [7] and for 0-1 matrices [5]. However, proving the $O(n)$ bound in the context of sequence saturation appears to be significantly more challenging.

2. Saturation and Extremal Funtions

We begin by defining the different variations of saturation and extremal functions.

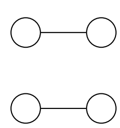
First, consider the setting of **graphs**. Let H be a *forbidden graph* and let G be any graph. We say that G **avoids** H if H is not a subgraph of G. A graph G is **H-saturated** if it avoids H, but adding any new edge to G creates a copy of H.

We define:

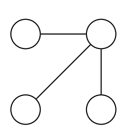
- $\mathrm{Sat}(n, H)$ as the **minimum** number of edges in an H-saturated graph on n vertices.
- $\mathrm{Ex}(n, H)$ as the **maximum** number of edges in a H-saturated graph on n vertices (equivalently, the maximum size of an H-free graph).

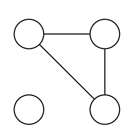
Turan's theorem, a fundamental result in extremal graph theory, evaluates $\mathrm{Ex}(n, H)$ in the case where H is a complete graph.

Example 1. Let $H = P_3$, the path graph on 3 vertices, and let G be the following graph:



Then G does not contain a copy of H, but adding any edge (which must be between one of the top and one of the bottom vertices) will induce a copy of H, so G is H-saturated. One can check that there are only two other H-saturated graphs on 4 vertices (up to isomorphism). These are the following:





It follows that $\mathrm{Sat}(4, H) = 2$ and $\mathrm{Ex}(4, H) = 3$.

The case of **0-1 matrices** is analogous. A 0-1 matrix is a matrix whose entries are either 0 or 1. Given a **forbidden pattern** P (a 0-1 matrix) and another 0-1 matrix A, we say that A **avoids** P if no submatrix of A can be transformed into P by changing some 1's to 0's. A matrix A is P**-saturated** if A avoids P, but changing any 0 in A to a 1 creates a copy of P. We define:

- $\mathrm{Sat}(n, P)$ as the **minimum** number of 1's in a P-saturated $n \times n$ matrix.
- $\mathrm{Ex}(n, P)$ as the **maximum** number of 1's in an $n \times n$ matrix that avoids P (equivalently, the maximum size of a P-free matrix).

Example 2. Consider the following 0-1 matrices:

$$P = \begin{pmatrix} \bullet & \bullet \\ \bullet & \end{pmatrix}, \qquad A = \begin{pmatrix} \bullet & \bullet & \bullet \\ & & \bullet \\ & & \bullet \end{pmatrix}.$$

As per standard convention, when writing 0-1 matrices, we leave an empty space for 0's and dark circles for 1's. Then A avoids P, but adding a one in any of the remaining places will induce a copy of P.

We can attempt to define analogous saturation and extremal functions for sequences, but some challenges arise. Fortunately, these can be addressed with a slight modification.

Given a sequence u of length r and a sequence s over an n-letter alphabet, we say that s **avoids** u if s does not contain a subsequence isomorphic to u under a one-to-one letter mapping. A natural definition of u**-saturation** would be that adding any new letter (from the n-letter alphabet) always induces a copy of u. From this, we could define $\mathrm{Sat}(n, u)$ and $\mathrm{Ex}(n, u)$ similarly to the graph and matrix cases.

However, this approach fails because we can construct arbitrarily long sequences that avoid u. For example, if $u = abc$, the sequence $s = 1, 1, \ldots, 1$ avoids u indefinitely, rendering $\mathrm{Ex}(n, u)$ undefined.

To resolve this, we introduce the notion of r**-sparsity**: a sequence is r**-sparse** if every r consecutive letters are pairwise distinct (where r is the length of u). We then refine our definitions:

- A sequence s is u**-saturated** if it avoids u, is r-sparse, and adding any new letter either induces a copy of u or violates r-sparsity.
- The functions $\mathrm{Sat}(n, u)$ and $\mathrm{Ex}(n, u)$ are then defined as before, now incorporating the r-sparsity condition.

Example 3. Let $u = abca$ and consider the sequence $s = 1, 2, 3, 4$, where $r = 3$ and $n = 4$. Clearly, s avoids u and is 3-sparse. We now check whether adding any letter results in a copy of u or violates 3-sparsity:

- Adding 1: To maintain 3-sparsity, 1 can only be placed after the 3 or at the end. In both cases, we obtain the subsequence $1, 2, 3, 1$, which is a copy of u.
- Adding 2: To maintain 3-sparsity, 2 must be placed at the end, forming the subsequence $2, 3, 4, 2$, which is a copy of u.
- Adding 3: To maintain 3-sparsity, 3 must be placed at the beginning, forming the subsequence $3, 1, 2, 3$, which is a copy of u.

- Adding 4: To maintain 3-sparsity, 4 must be placed either at the beginning or before the 2, but in both cases, we obtain the subsequence $4, 2, 3, 4$, which is a copy of u.

Thus, s is u-saturated. More generally, the same argument shows that the sequence $s = 1, 2, \ldots, n$ is always u-saturated, leading to the bound

$$\mathrm{Sat}(n, u) \leq n \leq \mathrm{Ex}(n, u).$$

We can also consider the saturation of *families* of forbidden structures. That is, given a family $\mathbf{F} = \{F_1, F_2, \ldots, F_k\}$ of forbidden graphs, patterns, or sequences, we examine the maximal and minimal sizes of a structure that avoids every element of $\mathbf{F}$ but where adding a new edge, 1, or letter, respectively, induces a copy of some element in $\mathbf{F}$. The saturation and extremal functions are defined analogously to the case of a single forbidden pattern.

It is relatively straightforward to show that the saturation function is $O(n)$ for graphs [7] and 0-1 matrices [5]. The proof for graphs involves considering the saturation of graph families and showing that modifying a family $\mathbf{F}$ into a new one, $\mathbf{F}'$, by deleting edges affects the saturation function by at most a linear term. Repeating this process simplifies the family until the saturation function can be directly shown to be $O(n)$.

We now show that the saturation function for 0-1 matrices is $O(n)$.

Proposition 2. *For any forbidden pattern P, the saturation function* $\mathrm{Sat}(n, P) = O(n)$.

Proof. If P consists entirely of 0's, the statement is trivial. Now, suppose P is a $k \times l$ matrix with a 1 entry at the intersection of the i-th row and j-th column. Define M to be a matrix with the following structure:

- The entries in the first $i - 1$ rows and the last $k - i$ rows are all 1's.
- The entries in the first $j - 1$ columns and the last $l - j$ columns are also all 1's.
- All other entries are 0.

As an example, consider $k = 4$, $l = 3$, $i = j = 2$, and $n = 5$. The matrix M is as shown:

$$M = \begin{pmatrix} \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & & & & \bullet \\ \bullet & & & & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \end{pmatrix}.$$

It is easy to verify that M avoids P, but adding a 1 to any position will induce a copy of P. In fact, it will induce a copy of the $k \times l$ all-1's matrix. This proves that M is P-saturated.

The number of 1's in M is given by the following expression:

$$(k - 1)n + (l - 1)n - (k - 1)(l - 1).$$

This is because there are $k - 1$ all-1 rows and $l - 1$ all-1 columns, and $(k - 1)(l - 1)$ ones are counted twice in both an all-1 row and an all-1 column. Therefore, we have the bound:

$$\mathrm{Sat}(n, P) \leq (k - 1)n + (l - 1)n - (k - 1)(l - 1),$$

which implies that $\mathrm{Sat}(n, P) = O(n)$. $\square$

Remark 1. The matrix A in Example 2 is in fact the matrix given by the construction in the proof of Proposition 2.

What makes Conjecture 1 more challenging than its graph and 0-1 matrix counterparts is the sparsity condition.

3. Saturation of Sequences

We are interested in showing that $\mathrm{Sat}(n, u) = O(n)$ for any sequence u. This has been shown for 2-letter sequences u in [1], but the cases for 3-letter sequences and higher remain open.

It is natural to begin by considering 3-letter sequences u, as these represent the simplest unresolved cases. Trivially, we know that $\mathrm{Sat}(n, u) \le \mathrm{Ex}(n, u)$, so if $\mathrm{Ex}(n, u) = O(n)$, we will be done. A **nonlinear sequence** is defined as one for which $\mathrm{Ex}(n, u)$ grows faster than linearly. Therefore, we only need to focus on nonlinear sequences.

In [9], the extremal function for sequences, particularly 3-letter sequences, was studied. It was shown that the sequence $u = abcacbc$ is nonlinear and "minimally nonlinear," meaning that no subsequence of u is nonlinear. To demonstrate that $\mathrm{Sat}(n, u) = O(n)$, we must construct a u-saturated sequence on n letters for every n, where the length of the sequence grows at most linearly with n. To find such sequences, we consider the following algorithm, where we say "x can be properly inserted into s" as a shorthand to mean "x can be inserted into s without violating r-sparsity and without inducing a copy of u".

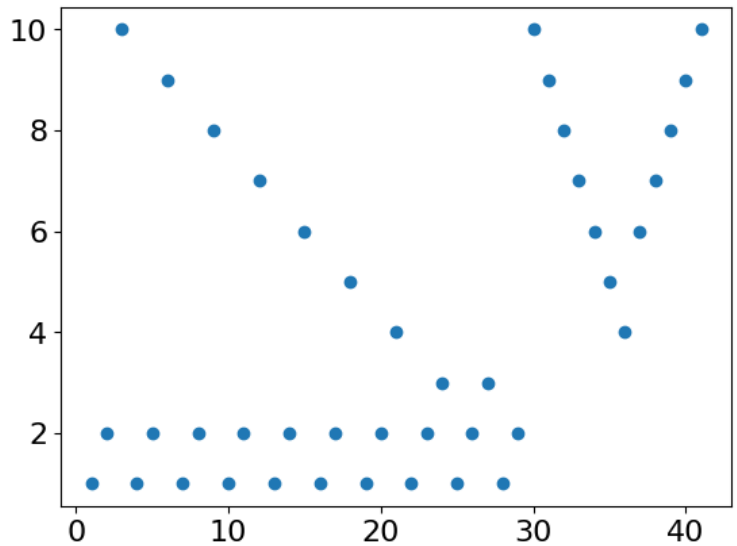
Algorithm 1 Constructing a u-Saturated Sequence

Input: Alphabet $A = \{1, \dots, n\}$, forbidden sequence u
Output: u-saturated sequence
Initialize the sequence: $s \leftarrow 1, 2, \dots, r-1$ ▷ Initial sequence avoids u
while it is possible to extend the sequence **do**
  for each letter $x \in A$ **do**
    if x can be *properly inserted* into s **then**
      Insert x appropriately into s to form s' ▷ Smallest x, leftmost position
      Update $s \leftarrow s'$ ▷ New sequence
      break ▷ Exit loop after the first valid insertion
    end if
  end for
end while
Return s ▷ Final sequence is u-saturated

We can implement this algorithm in code and run it for different values of n to observe if any patterns emerge. If such patterns appear, and if they can be easily generalized to arbitrary n, then it is highly probable that these are the sequences we are seeking. The only remaining step is to verify that the general pattern is indeed a u-saturated sequence for all values of n.

However, to better visualize long sequences and identify patterns, we use the following method. Given a sequence $s = s_1 \cdots s_k$, we plot the points (i, s_i) for $i = 1, \dots, n$. For instance, the sequence $1, 2, \dots, n$ forms a line with a slope of 1, while the sequence $1, 1, \dots, 1$

results in a flat line. By implementing the above algorithm in code for $u = abcacbc$ and running it for various values of n, we observe the following pattern:



This example is the $n = 10$ case. This corresponds to the sequence

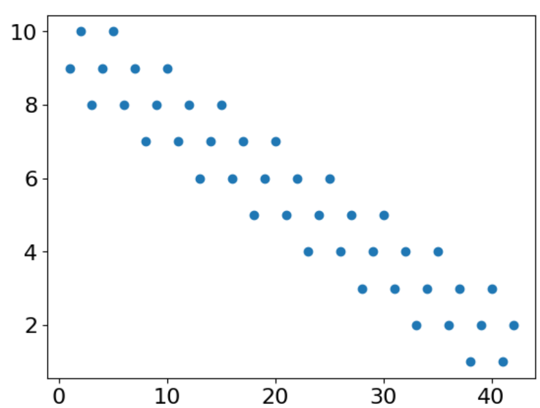
$$1, 2, n, 1, 2, n-1, \ldots, 1, 2, 3, 1, 2, 3, 1, 2, n, n-1, \ldots, 6, 5, 4, 6, 7, \ldots, n.$$

From here it is straightforward (albeit quite tedious) to verify that this is a u-saturated sequence, for any n. The length of this sequence clearly grows linearly with n, and thus we have shown that

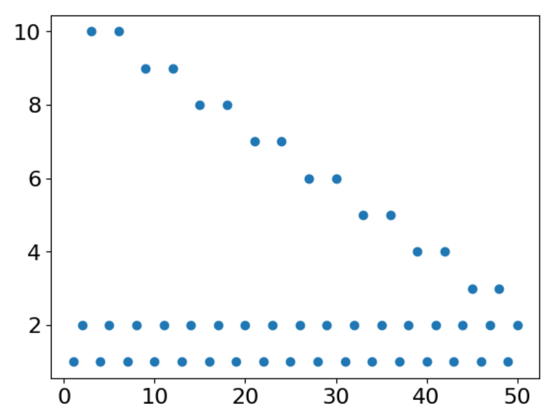
$$\mathrm{Sat}(n, abcacbc) = O(n)$$

despite the fact that $abcacbc$ is nonlinear.

We can do this for different sequences. For every case we tried, we got a repeating pattern like the one above (however some had minor variations at the end that depended on the value of $n \pmod d$ for some d; nevertheless we still had a pattern that worked). Some more examples are shown below:



$u = abcabcab$



$u = abcabcabc$

The fact that we always get a pattern prompts us to posit the following.

Conjecture 3. *Let $s(n, u)$ be the sequence of produced by the algorithm above. Then the length of $s(n, u)$ grows at most linearly in n, for any u.*

There is one important point to note about this algorithm: since it always prefers to insert smaller letters, in the process of generating $s(n, u)$, right before we add the first "n", the sequence must be exactly $s(n-1, u)$. Thus, $s(n-1, u)$ is a subsequence of $s(n, u)$. This gives further evidence that this algorithm will always produce a pattern.

A further direction to investigate, after Conjecture 1 is resolved, would be be to classify when $\mathrm{Sat}(n, u) = O(1)$ or $\Theta(n)$. The corresponding question for 0-1 matrices was investigated by Geneson [6], who showed that almost all permutation matrices have bounded saturation function, and by Berendsohn [2], who completely resolved the classification for permutation matrices. However, the general question remains open.

Acknowledgements

I am grateful to my mentor, Jesse Geneson, for proposing this project and for his guidance during the preparation of this report. While working on this report, I was part of PRIMES-USA, a year-long math research program hosted by MIT. I would like to express my gratitude to the directors and organizers of MIT PRIMES for making this experience possible.

References

[1] Anand, Jesse Geneson, Suchir Kaustav, and Shen-Fu Tsai, *Sequence saturation*, Discrete Applied Mathematics **360** (2025), 382–393.

[2] Benjamin Aram Berendsohn, *An exact characterization of saturation for permutation matrices*, Comb. Theory **3** (2021).

[3] Gábor Damásdi, Balázs Keszegh, David Malec, Casey Tompkins, Zhiyu Wang, and Oscar Zamora, *Saturation problems in the ramsey theory of graphs, posets and point sets*, European Journal of Combinatorics **95** (2021), 103321.

[4] Michael Ferrara, Bill Kay, Lucas Kramer, Ryan R. Martin, Benjamin Reiniger, Heather C. Smith, and Eric Sullivan, *The saturation number of induced subposets of the boolean lattice*, Discret. Math. **340** (2017), 2479–2487.

[5] Radoslav Fulek and Balázs Keszegh, *Saturation problems about forbidden 0-1 submatrices*, SIAM J. Discret. Math. **35** (2020), 1964–1977.

[6] Jesse Geneson, *Almost all permutation matrices have bounded saturation functions*, Electron. J. Comb. **28** (2020), P2.16.

[7] László Kászonyi and Zs. Tuza, *Saturated graphs with minimal number of edges*, Journal of Graph Theory **10** (1986), 203 –210.

[8] Gabriel Nivasch, *Improved bounds and new techniques for davenport-schinzel sequences and their generalizations*, Journal of the ACM **57** (2008), 1–44.

[9] Seth Pettie, *Generalized davenport–schinzel sequences and their 0–1 matrix counterparts*, Journal of Combinatorial Theory, Series A **118** (2011), no. 6, 1863–1895.

San José State University San José, CA 95192
Email address: shihan.kanungo@sjsu.edu