SIAM J. DISCRETE MATH.
Vol. 25, No. 1, pp. 211–233

ORIGINS OF NONLINEARITY IN DAVENPORT–SCHINZEL SEQUENCES[*]

SETH PETTIE[†]

Abstract. A *generalized* Davenport–Schinzel sequence is one over a finite alphabet that excludes subsequences isomorphic to a fixed *forbidden subsequence.* The fundamental problem in this area is bounding the maximum length of such sequences. Following Klazar, we let $\mathrm{Ex}(\sigma, n)$ be the maximum length of a sequence over an n-letter alphabet excluding subsequences isomorphic to σ. It has been proved that for every σ, $\mathrm{Ex}(\sigma, n)$ is either linear or very close to linear. In particular it is $O(n2^{\alpha(n)^{O(1)}})$, where α is the inverse-Ackermann function and $O(1)$ depends on σ. In much the same way that the complete graphs K_5 and $K_{3,3}$ represent the minimal causes of nonplanarity, there must exist a set Φ_{Nonlin} of *minimal* nonlinear forbidden subsequences. Very little is known about the size or membership of Φ_{Nonlin}. In this paper we construct an infinite antichain of nonlinear forbidden subsequences which, we argue, strongly supports the conjecture that Φ_{Nonlin} is itself infinite. Perhaps the most novel contribution of this paper is a succinct, humanly readable code for expressing the structure of forbidden subsequences.

Key words. forbidden sequences, generalized Davenport–Schinzel sequences, infinite antichains

AMS subject classifications. 05D99, 68R05

DOI. 10.1137/080735862

1. Introduction.

A *generalized* Davenport–Schinzel sequence is one over a finite alphabet, none of whose subsequences are isomorphic to a fixed *forbidden subsequence.* Davenport–Schinzel sequences have played a major role in combinatorial and computational geometry and have recently been applied to bounding the running time of self-adjusting data structures [27, 28]. A canonical example of their application is in bounding the complexity of geometric objects, particularly the lower envelopes of functions with a limited number of crossings, such as low degree polynomials or line segments. See Agarwal and Sharir [2] for a survey of the geometric applications of Davenport–Schinzel sequences up to 1995. In order to discuss prior work with any precision we must first define some basic notation concerning sequences.

Notation and terminology. Let $\Sigma(\sigma)$ be the *alphabet* (or set of distinct symbols) of σ. Let $|\sigma|$ be the length of σ and $\|\sigma\| = |\Sigma(\sigma)|$. For the following definitions let $\sigma = (\sigma_j)_{1\le j\le|\sigma|}$ and $\hat{\sigma} = (\hat{\sigma}_j)_{1\le j\le|\hat{\sigma}|}$ be two sequences over possibly different alphabets. We say that equal length σ and $\hat{\sigma}$ are *isomorphic*, or $\sigma \sim \hat{\sigma}$, if there is a bijection $f : \Sigma(\sigma) \to \Sigma(\hat{\sigma})$ such that $f(\sigma_j) = \hat{\sigma}_j$. We say σ is a *subsequence* of $\hat{\sigma}$ if there are indices $j_1 < j_2 < \cdots < j_{|\sigma|}$ such that $\sigma_i = \hat{\sigma}_{j_i}$. We write $\sigma \mathrel{\bar{\prec}} \hat{\sigma}$ and $\sigma \prec \hat{\sigma}$ to mean that σ is, respectively, a subsequence of $\hat{\sigma}$ and isomorphic to a subsequence of $\hat{\sigma}$. A sequence $\hat{\sigma}$ (or class of sequences) is σ-*free* if $\sigma \not\prec \hat{\sigma}$. A sequence $\sigma = (\sigma_j)$ is c-*sparse* if $\sigma_i = \sigma_j$ implies $|j - i| \ge c$. A sequence σ is a *palindrome* if it is isomorphic to its reversal, denoted $\overline{\sigma}$.

DEFINITION 1.1. *$\mathrm{Ex}(\sigma, n)$ is the maximum length of a $\hat{\sigma}$ such that $\sigma \not\prec \hat{\sigma}$, $\|\hat{\sigma}\| = n$, and $\hat{\sigma}$ is $\|\sigma\|$-sparse. (The condition that $\hat{\sigma}$ be $\|\sigma\|$-sparse simply rules out uninteresting*

[*] Received by the editors September 22, 2008; accepted for publication (in revised form) November 15, 2010; published electronically February 10, 2011. This work is supported by NSF CAREER grant CCF-0746673 and a grant from the United States-Israel Binational Science Foundation.
http://www.siam.org/journals/sidma/25-1/73586.html

[†] University of Michigan, Electrical Engineering and Computer Science Department, Ann Arbor, MI 48109 (pettie@umich.edu).

sequences. For instance, the infinite sequence ababababa $\cdots$ is (abc)-free, but in the least interesting way.)

1.1. Standard Davenport–Schinzel sequences. Much of the work in this area follows the original definition of Davenport and Schinzel [8], who considered alternating forbidden subsequences of the form $ababa\ldots$. It is not difficult to prove that $\mathrm{Ex}(aba,n) = n$ and $\mathrm{Ex}(abab,n) = 2n-1$, but for longer alternating forbidden subsequences the problem becomes much harder, even if we are only interested in asymptotic bounds. A celebrated result of Hart and Sharir [13] is that $\mathrm{Ex}(ababa,n) = \Theta(n\alpha(n))$, where α is the slowly growing inverse of Ackermann's function.[1] The best known bounds on $\mathrm{Ex}((ab)^k,n)$ and $\mathrm{Ex}((ab)^k a,n)$ are also slightly nonlinear in n for all $k \geq 3$. However, tight asymptotic bounds are only known for $\mathrm{Ex}(ababab,n)$:

$$\mathrm{Ex}(ababab,n) = \Theta(n \cdot 2^{\alpha(n)}), \tag{1.1}$$

$$\mathrm{Ex}((ab)^k,n) = n \cdot 2^{(1+o(1))\alpha(n)^{k-2}/(k-2)!} \qquad \text{for all } k \geq 4, \tag{1.2}$$

$$\mathrm{Ex}((ab)^k a,n) \leq n \cdot 2^{(1+o(1))\alpha(n)^{k-2}\log\alpha(n)/(k-2)!} \qquad \text{for all } k \geq 3. \tag{1.3}$$

The lower bounds in (1.1)–(1.3) were provided by Agarwal, Sharir, and Shor [3] as well as the upper bound in (1.1). Nivasch [24] provided the upper bounds in (1.2), (1.3). Note that for $\mathrm{Ex}((ab)^k,n)$ the upper and lower bounds are tight up to the lower order terms in the exponent. It has been conjectured [5] that the upper bound (1.3) is tight; however, there are no lower bounds on $\mathrm{Ex}((ab)^k a,n)$ that are stronger than those for $\mathrm{Ex}((ab)^k,n)$.

Much less is known about the behavior of $\mathrm{Ex}(\sigma,n)$ when σ is *not* of the form $ababa\ldots$. Nivasch [24], improving on [19], showed that for any σ with $|\sigma| \geq \|\sigma\|+3$

$$\mathrm{Ex}(\sigma,n) \leq \begin{cases} n \cdot 2^{(1+o(1))\frac{\alpha(n)^t}{t!}}, & t = \frac{|\sigma|-\|\sigma\|-2}{2}, \text{ and } |\sigma|-\|\sigma\| \text{ even,} \\ n \cdot 2^{(1+o(1))\frac{\alpha(n)^t}{t!}\log\alpha(n)}, & t = \frac{|\sigma|-\|\sigma\|-3}{2}, \text{ and } |\sigma|-\|\sigma\| \text{ odd.} \end{cases} \tag{1.4}$$

Aside from (1.1)–(1.2), which are special cases of (1.4), there are no σ for which (1.4) is known to be tight.

1.2. Nonlinearity. Perhaps the most basic problem regarding the functions $\mathrm{Ex}(\sigma,n)$ is to explain the difference between *linear* and *nonlinear* forbidden subsequences, that is, to identify the features of σ that cause $\mathrm{Ex}(\sigma,n)$ to be $O(n)$ or $\omega(n)$.[2]

One can easily see that the set of all nonlinear forbidden subsequences can be characterized by a set of *minimal* such forbidden subsequences, denoted Φ_{Nonlin}.

DEFINITION 1.2. *The set* $\Phi_{Lin} = \{\sigma \mid \mathrm{Ex}(\sigma,n) = O(n)\}$. *Define* Φ_{Nonlin} *to be any minimal set such that*

$$\mathrm{Ex}(\sigma,n) = \omega(n) \qquad \textit{if and only if} \qquad \exists\sigma' \in \Phi_{Nonlin} : \sigma' \prec \sigma \;\; \textit{or} \;\; \sigma' \prec \overline{\sigma}.$$

Note that Φ_{Nonlin} is not unique since we can exchange a $\sigma' \in \Phi_{Nonlin}$ for its reversal $\overline{\sigma'}$ if σ' is not a palindrome. Furthermore, we show that Φ_{Nonlin} contains nonpalindromes.

[1] There are several commonly used definitions of Ackermann's function that are essentially equivalent. It can be defined as $A(1,j) = 2^j$ for $j \geq 1$, $A(i,1) = A(i-1,2)$ for $i > 1$, and $A(i,j) = A(i-1, A(i,j-1))$ for $i > 1, j > 1$. The two and one argument inverses are defined as $\alpha(m,n) = \min\{i \mid A(i, \lceil\frac{m+n}{n}\rceil) \geq n\}$ and $\alpha(n) = \alpha(n,n)$. Most perturbations of this definition have no significant effect on the inverses. For example, if we redefined the base case as $A(1,j) = j+1$, this would increase $\alpha(m,n)$ and $\alpha(n)$ by at most 3.

[2] Recall that $f = \omega(g)$ is short for $\lim_{n\to\infty} g(n)/f(n) = 0$ and $f = \Omega(g)$ is short for $f(n) \geq c \cdot g(n)$ for some c and infinitely many n.

Given the volume of research on (generalized) Davenport–Schinzel sequences, it is rather surprising how little we know about Φ_{Nonlin}. Hart and Sharir's result [13] shows that $ababa \in \Phi_{Nonlin}$, and Adamec, Klazar, and Valtr [1] proved that *ababa* is the *only* 2-letter sequence in Φ_{Nonlin}. Klazar [17] showed that Φ_{Nonlin} contains at least two elements: *ababa* and another which is currently unknown, but is a subsequence of *abcbdadbcd*. In other words, the presence of $ababa \prec \sigma$ is not the *sole* cause of nonlinearity in $\mathrm{Ex}(\sigma, n)$. Klazar's result [17] is actually more general in that he shows that any 2-sparse (repetition free) σ for which a directed graph $G(\sigma)$ is strongly connected has $\mathrm{Ex}(\sigma, n) = \Omega(n\alpha(n))$. The vertex set of $G(\sigma)$ is $\Sigma(\sigma)$, and an edge (x, y) exists if and only if either $xyyx \mathrel{\bar{\prec}} \sigma$ or $yxyx \mathrel{\bar{\prec}} \sigma$. See Figure 1.1.[3]

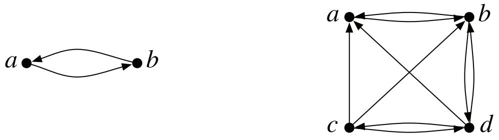

FIG. 1.1. *The digraph $G(\sigma)$ has one vertex for each letter in the alphabet of a repetition-free σ. An edge (x, y) appears in $G(\sigma)$ if σ contains as a subsequence either xyyx or yxyx. (Left) $G(ababa)$; (right) $G(abcbdadbcd)$.*

Klazar [19] posed the intriguing question of whether Φ_{Nonlin} is finite or infinite. The results of [17] raised the possibility that *strong connectivity* of $G(\sigma)$ could be the cause of nonlinearity of $\mathrm{Ex}(\sigma, n)$. This hypothesis, which was never put forward in [17], has some aesthetic appeal. It says that out of the meaningless muck of Φ_{Nonlin}—which is just a set of inert sequences—we could nonetheless explain the true cause *semantically* as a statement about the connectivity of $G(\sigma)$. However, even if this hypothesis were true, it would not resolve the question of whether Φ_{Nonlin} is finite since there is no known infinite antichain of minimal, strongly connected $G(\sigma)$.

1.3. Linearity. Adamec, Klazar, and Valtr [1] and Klazar and Valtr [20] studied ways in which forbidden subsequences could be combined and manipulated that preserved (or did not significantly affect) their extremal functions. Adamec, Klazar, and Valtr [1] proved that $\mathrm{Ex}(abbaab) = O(n)$ and made several trivial observations on the extremal functions of related forbidden subsequences. Below, a, b are symbols, σ's are sequences, and k is any fixed integer.

$$\mathrm{Ex}(\sigma, n) = \mathrm{Ex}(\overline{\sigma}, n), \tag{1.5}$$

$$\mathrm{Ex}(aa, n) = n, \tag{1.6}$$

$$\mathrm{Ex}(\sigma_1 \prec \sigma_2, n) = O(\mathrm{Ex}(\sigma_2, n)), \tag{1.7}$$

$$\mathrm{Ex}(aa\sigma, n) = O(n + \mathrm{Ex}(a\sigma, n)), \tag{1.8}$$

$$\mathrm{Ex}(\sigma_1 a^k \sigma_2, n) = O(\mathrm{Ex}(\sigma_1 aa \sigma_2, n)). \tag{1.9}$$

It follows from (1.5)–(1.9) and $\mathrm{Ex}(abbaab, n) = O(n)$ that *ababa* is the only 2-letter sequence in Φ_{Nonlin}. Klazar and Valtr [20] proved two nontrivial theorems on forbidden subsequences that are derived from simpler ones. Lines (1.10), (1.11) are

[3] Klazar [17] actually exhibited $G(abcbadadbcd)$ as a strongly connected graph. Nivasch [23] noted that the second "*a*" is redundant, i.e., that $G(abcbdadbcd)$ is also strongly connected.

corollaries that pertain to the set Φ_{Lin}:

$$\hat{\sigma},\ \sigma_1 aa\sigma_2 \in \Phi_{Lin} \text{ implies } \sigma_1 a\hat{\sigma}a\sigma_2 \in \Phi_{Lin}, \quad \text{where } \Sigma(\sigma_1 aa\sigma_2) \cap \Sigma(\hat{\sigma}) = \emptyset, \tag{1.10}$$

$$\sigma_1 aa\sigma_2 a \in \Phi_{Lin} \text{ implies } \sigma_1 abba\sigma_2 ab \in \Phi_{Lin}, \quad \text{where } b \notin \Sigma(\sigma_1\sigma_2). \tag{1.11}$$

We define Φ_{KV} to be the set of all linear sequences that can be derived from lines (1.5)–(1.11). Among others, Φ_{KV} includes all N-shaped sequences of the form $ab\cdots yzy\cdots bab\cdots yz$. Valtr [32] used the linearity of such sequences to prove that geometric graphs with no $k = O(1)$ pairwise crossing edges have size $O(n \log n)$. It remains an open problem whether $\Phi_{KV} = \Phi_{Lin}$.

1.4. New results. We exhibit an infinite antichain Ψ (with respect to $\prec$) of nonlinear forbidden subsequences that constitutes the first plausible candidate for Φ_{Nonlin}. The elements of Ψ are not fundamentally different but naturally divide themselves into 9 classes, where each class has a constant number of *primitive types.* Why 9? This number arises from a new code we use for describing the structure of a forbidden subsequence. Each element in Ψ can be represented as a finite string over {♡, ♤, ♠, ♥, ♢, ♣, ♣, (,)}. Such strings must obey several grammatical rules, and there just happens to be 9 natural classes of grammatical strings. Our result refutes the possibility that strong connectivity of $G(\sigma)$ is *the* cause of nonlinearity of $\mathrm{Ex}(\sigma, n)$ and implies, nonconstructively, that $|\Phi_{Nonlin}| \geq 3$. In the conclusion we discuss why the infinitude of Ψ supports the proposition that Φ_{Nonlin} is also infinite.

1.5. Related work. Davenport–Schinzel sequences are part of a class of problems concerning combinatorial objects with forbidden substructures. Klazar [19] surveys generalizations of Davenport–Schinzel sequences to trees, permutations, hypergraphs, matrices, ordered digraphs, and partitions. Other examples in this vein include graphs avoiding a fixed set of minors (e.g., planar graphs), matrices with the Monge property [7], and (partially defined) monotone matrices [4, 16, 15]. Below we survey the results concerning trees and matrices.

Trees. A *Davenport–Schinzel tree* is a tree whose nodes are assigned one of n labels such that all nodes with a common label lie on a path. If the tree is directed, then edges point toward a specified root. The functions $\mathrm{Ex}^{T}(\sigma, n)$ and $\mathrm{Ex}^{\vec{T}}(\sigma, n)$ are the maximum size of undirected and directed trees, respectively, all of whose paths avoid subsequences isomorphic to σ. Valtr [33] studied both the directed and undirected versions of this problem and, in both cases, fully characterized the nonlinear forbidden subsequences $\sigma \in \{a, b\}^*$ over the 2-letter alphabet. In particular, $\mathrm{Ex}^{T}(\sigma, n) = \omega(n)$ if and only if σ contains either *ababa* or *abbaab*, and $\mathrm{Ex}^{\vec{T}}(\sigma, n) = \omega(n)$ if and only if σ contains either *ababa* or *abbab*. Valtr also exhibited some minimal nonlinear forbidden subsequences over the 3-letter alphabet. The question of whether there are an infinite number of causes of nonlinearity in this context is open *and* independent of whether Φ_{Nonlin} is infinite.

Matrices. Let A and B be two matrices whose entries are either 1 or blank. We write $A \prec B$ if there is a submatrix B' of B with the same dimensions as A such that for each 1 in A there is a corresponding 1 in B'. (In other words, blanks in B are "don't cares.") We let $\mathrm{Ex}^{M}(A, n)$ be the maximum number of 1's in an $n \times n$ matrix avoiding A. Füredi [10] and Bienstock and Györi [6] initiated the study of this problem and showed that $\mathrm{Ex}^{M}(\left(\begin{smallmatrix} 1 & & 1 \\ 1 & 1 & \end{smallmatrix}\right), n) = \Theta(n \log n)$. The original application for this particular pattern was in bounding the number of unit distances in a convex polygon [10]. This pattern and similar ones have since found uses in other geometric problems [25, 9, 26] and analyzing the complexity of several data structures [28]. Füredi and Hajnal [11] and Tardos [30] gave sharp asymptotic bounds on $\mathrm{Ex}^M(A,n)$ for all A's containing at most four 1's, as well as some sharp bounds when there are multiple forbidden submatrices. In general it is known that $\mathrm{Ex}^M(A,n) = O(n)$ for any permutation matrix A [22], that $\mathrm{Ex}^M(A,n) = O(n^{2-\Omega(1)})$ for *any* matrix A [11], and that $\mathrm{Ex}^M(A,n) < n \cdot 2^{\alpha(n)^{O(1)}}$ if A contains one 1 in each column [18]. Results of Keszegh [14] and Geneson [12] imply that there are an infinite number of minimal nonlinear forbidden submatrices whose extremal function is $\Omega(n \log n)$, yet only a constant number of these matrices can actually be identified. See Pettie [29] for a longer discussion of minimal nonlinear forbidden matrices.

1.6. Overview. In section 2 we construct a class of n-letter sequences with length $\Omega(n\alpha(n))$. In section 3 we exhibit the 9 classes of nonlinear forbidden subsequences and show how to generate an infinite number of such sequences. The construction of $\Omega(n\alpha(n))$-length sequences in section 2 can easily be adapted to give $\Omega(n\alpha(n))$-size labeled trees. In section 4 we reprove some of Valtr's results [33] concerning Davenport–Schinzel trees and exhibit an infinite antichain of forbidden subsequences that are nonlinear with respect to Ex^T. In section 5 we conclude with some remarks and conjectures.

2. Constructing nonlinear sequences.

2. Constructing nonlinear sequences. We construct sequences with length $\Omega(n\alpha(n))$ in much the same way as those constructions from [13, 2, 34]. As an intermediate step, our construction builds a *model tree* that is used extensively in the proofs. If u and v are nodes in a rooted tree, $u \lhd v$ means that u is a *strict* descendant of v, and $u \unlhd v$ means $u \lhd v$ or $u = v$. When referring to nodes, the relations *below* and *above* are synonymous with $\unlhd$ and $\unrhd$.

Constructing the model tree. Our nonlinear sequences are associated with full, rooted binary trees $\{T(i,j)\}_{i,j\geq 1}$, where the leaves are assigned a left-to-right order and where each node v is assigned a *label* $\mathscr{L}(v)$, which is a sequence of distinct symbols. Let $|T|$ be the number of leaves in tree T. Our trees will satisfy the following property.

PROPERTY 2.1. *For any $i, j \geq 1$,*

1. *the labels of $T(i,j)$ are drawn from an alphabet of $j \cdot |T(i,j)|$ symbols.*
2. *each leaf's label has length j.*
3. *every symbol appears in the label of exactly one leaf.*
4. *every symbol appears in exactly i labels.*
5. *the root has no label.*

We construct the trees in the following doubly inductive manner. Let $T(1,j)$ be a rooted tree with 3 nodes: a root r and leaves v_1, v_2 with labels $\mathscr{L}(v_1) = (a_j, a_{j-1}, \ldots, a_1)$ and $\mathscr{L}(v_2) = (a_{2j}, a_{2j-1}, \ldots, a_{j+1})$, where the symbols $a_1, \ldots, a_{2j}$ are distinct. Clearly $T(1,j)$ satisfies Property 2.1 list items 1–5.

Before we get to $T(i,j)$, for $i \geq 2$, let us first define the composition of labeled trees. If T and T' are labeled trees, we derive $T \circ T'$ by making $|T'|$ copies of T, each with alphabets disjoint from T' and each other, and by identifying each copy of T with a leaf of T'. The previously unlabeled root of a T inherits the label of its associated leaf from T'. Obviously $|T \circ T'| = |T| \cdot |T'|$. Let $T(i,0)$ be a 3-node, 2-leaf tree with empty labels.

Structurally $T(i,j) = T(i,j-1) \circ T(i-1, |T(i,j-1)|)$, where $i,j \geq 1$. See Figure 2.1. Let v_k be the kth leaf of $T(i-1, |T(i,j-1)|)$ and $v_{k,l}$ be its lth leaf descendant

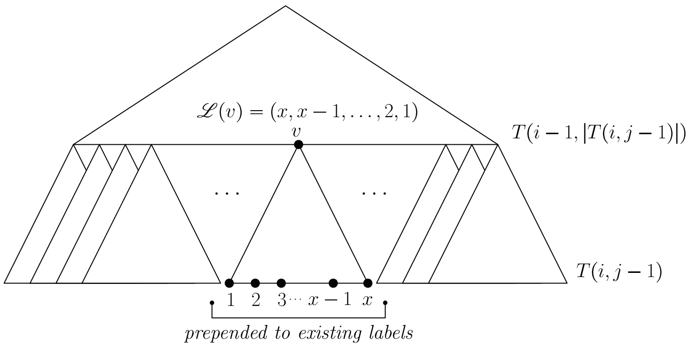
$\mathscr{L}(v) = (x, x-1, \ldots, 2, 1)$
v
$T(i-1, |T(i, j-1)|)$
$\cdots$
$\cdots$
$T(i, j-1)$
$1 \;\; 2 \;\; 3 \cdots x-1 \;\; x$
prepended to existing labels

FIG. 2.1. *Composition of trees in the construction of $T(i, j)$.*

in $T(i, j)$. If $\mathscr{L}(v_k) = (a_{k,|T(i,j-1)|}, \ldots, a_{k,1})$, we prepend $a_{k,l}$ to $\mathscr{L}(v_{k,l})$. All other labels in $T(i, j)$ are kept as is. One may verify that $T(i, j)$ satisfies Property 2.1.

The leaves of any $T(i, j)$ (which may live in a larger tree $T(i', j')$ for $i' > i$) are called *i-nodes*. If a symbol a appears in the label of a leaf v, we say that *a originates* from v.

LEMMA 2.2 (locations of symbols). *Let a be any symbol in the alphabet of $T(i, j)$. Then a appears in the labels of i distinct nodes $v_1, \ldots, v_i$ such that $v_i \lhd v_{i-1} \lhd \cdots \lhd v_1$. Furthermore, v_k is the* first *k-node encountered on the path from v_1 to v_i; that is, v_1 and v_i uniquely determine $v_2, \ldots, v_{i-1}$.*

Proof. The claim follows easily by induction over the construction of $T(i, j)$. □

We adopt the following global ordering for symbols in $T(i, j)$. The alphabet is $1, \ldots, j \cdot |T(i, j)|$, and the label of the kth leaf v_k of $T(i, j)$ is $\mathscr{L}(v_k) = (kj, kj - 1, \ldots, (k-1)j + 1)$. In general, we say that a labeled tree is *sorted* if, for any two leaves v_1, v_2, $\mathscr{L}(v_1)$ is in descending order and if v_1 is to the left of v_2, all symbols in $\mathscr{L}(v_1)$ are smaller than those in $\mathscr{L}(v_2)$.

LEMMA 2.3 (symbol ordering). *Let v be an i'-node in some $T(i, j)$, for $i' \le i$, and let $\mathscr{L}(v) = (a_l, a_{l-1}, \ldots, a_1)$ be its label. Then*

1. *$a_1 < a_2 < \cdots < a_l$.*
2. *let $v_1, v_2, \ldots, v_l$ be v's nearest $(i'-1)$-node ancestors, in least-to-most ancestral order, and let v_{l+1} be its nearest i'-node ancestor or the root if there is none. Then, for each $k \in [1, l]$, all occurrences of a_k that are strictly above v must appear above v_k and strictly below v_{k+1}.*

Proof. We first prove part 1. Our claim is that if $T(i, j)$ possesses a sorted labeling, then the labels of all nodes are in descending order. If $i = 1$, the claim holds trivially. We may then write $T(i, j) = T(i, j-1) \circ T(i-1, |T(i, j-1)|)$. By induction the claim holds for each individual $T(i, j-1)$. One can quickly check that if $T(i, j)$ is sorted, then the tree $T(i-1, |T(i, j-1)|)$ is also sorted; thus, by induction the claim holds for all nodes in a sorted $T(i, j)$. Part 2 follows easily by induction over the construction of $T(i, j)$. □

LEMMA 2.4 (the trapping lemma). *Let v, w, x be vertices with $v \unlhd w \unlhd x$, and suppose that t appears in $\mathscr{L}(w)$ and s appears in both $\mathscr{L}(v)$ and $\mathscr{L}(x)$. If the symbol t originates at a descendant of v, then $t \in \mathscr{L}(v)$.*

Proof. Suppose v is an i'-node. Lemma 2.2 implies that v is the *first* i'-node on the path from x to the origin of s. Since $w \unlhd x$ and t originates below v, v is also the first i'-node on the path from w to the origin of t. By Lemma 2.2 this implies that t appears in $\mathscr{L}(v)$; see Figure 2.2. □

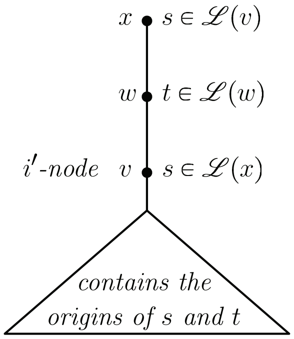

FIG. 2.2. *The situation that causes t to make an* "implied" *appearance in $\mathscr{L}(v)$.*

Lemma 2.4 is a useful tool for generating nonlinear forbidden subsequences. In the terminology developed later in section 3, the symbol s appearing at $\mathscr{L}(v)$ and $\mathscr{L}(x)$ functions as a *trap* for the *captive* symbol t at $\mathscr{L}(w)$. Whenever this situation can be created, we can conclude that t makes an "implied" appearance $\mathscr{L}(v)$; see Figure 2.2.

2.1. Constructing the nonlinear sequence. We form the sequence $\mathscr{S}_{i,j}$ directly from the labeled tree $T(i,j)$. Property 2.1(1) says the alphabet size of $T(i,j)$ is $j \cdot |T(i,j)|$. The effective alphabet size of $\mathscr{S}_{i,j}$ will also be $j \cdot |T(i,j)|$. However, we will introduce some *garbage* symbols to guarantee that $\mathscr{S}_{i,j}$ is c-sparse, for some $c = O(1)$, in order to comply with Definition 1.1. Let $v_1, \ldots, v_{2|T(i,j)|-1}$ be the nodes of $T(i,j)$ listed in postorder,[4] and let $\mathscr{L}'(v_k) = \mathscr{L}(v_k) \cdot (g_{k,1}, \ldots, g_{k,c})$ be v_k's augmented label, where $g_{k,1}, \ldots, g_{k,c}$ are garbage symbols associated only with v_k. The sequence $\mathscr{S}_{i,j}$ is defined as

$$\mathscr{S}_{i,j} = \mathscr{L}'(v_1) \cdot \mathscr{L}'(v_2) \cdots \mathscr{L}'(v_{2|T(i,j)|-1}).$$

LEMMA 2.5 (sparsity). *$\mathscr{S}_{i,j}$ is c-sparse.*

Lemma 2.5 follows from the fact that $\mathscr{L}(v_k)$ contains only distinct symbols and, in $\mathscr{S}_{i,j}$, there are c garbage symbols between $\mathscr{L}(v_k)$ and $\mathscr{L}(v_{k+1})$, all of which occur only once. Lemma 2.6 expresses the length of $\mathscr{S}_{i,j}$ in terms of the standard one- and two-argument versions of the inverse-Ackermann function.

LEMMA 2.6 (nonlinear length). *Let $n = \|\mathscr{S}_{i,j}\|$ and $l = |T(i,j)|$. Then $|\mathscr{S}_{i,j}| = \Theta(n\alpha(n,l))$, and if $j = O(1)$, $|\mathscr{S}_{i,j}| = \Theta(n\alpha(n))$.*

With the addition of garbage symbols we have $n = j\,|T(i,j)| + c(2\,|T(i,j)| - 1) = O(j\,|T(i,j)|)$, where $|T(i,j)|$ satisfies the recurrence

$$\begin{aligned}
|T(1,j)| &= 2 && \text{for } j \geq 1,\\
|T(i,1)| &= 2 \cdot |T(i-1,2)| && \text{for } i \geq 2,\\
|T(i,j)| &= |T(i,j-1)| \cdot |T(i-1,|T(i,j-1)|)| && \text{for } i,j \geq 2.
\end{aligned}$$

It is a simple, but tedious, exercise to show that the row-inverses of T are asymptotically equivalent to the corresponding row-inverses of Ackermann's function, under Tarjan's definition [31] or a similar one. In particular, this implies that $i = \alpha(j\,|T(i,j)|\,,|T(i,j)|) \pm O(1)$. Alon et al. [5] presented a systematic way to prove bounds of this kind.

[4] For a rooted, ordered tree, the unique postorder consists of the postorder of the root's left subtree, followed by the postorder of its right tree, followed by the root.

In the remainder of the paper $\mathscr{S}$ and T refer to $\mathscr{S}_{i,1}$ and $T(i,1)$ for an arbitrary i and some sufficiently large sparsity constant c.

Lemma 2.7 is often invoked to determine the correct order of two symbols. In particular, if $stst \mathrel{\bar{\prec}} \mathscr{S}$, we can infer that $s < t$; i.e., the origin of s is to the left of the origin of t.

Lemma 2.7 (*ababa*-freeness). *For $s < t$, $tsts \mathrel{\not\bar{\prec}} \mathscr{S}$.*

Proof. Let u, v, w, x be the vertices in T associated with the respective occurrences of t and s in the purported subsequence $tsts$ appearing in $\mathscr{S}$. That is, t appears in $\mathscr{L}(u)$ and $\mathscr{L}(w)$, s appears in $\mathscr{L}(v)$ and $\mathscr{L}(x)$. The postordering of vertex labels in $\mathscr{S}$ and the ordering $s < t$ imply that $u \trianglelefteq v \trianglelefteq w \trianglelefteq x$. Lemma 2.4 and the descending order of vertex labels imply that t appears in $\mathscr{L}(v)$ and precedes s in $\mathscr{L}(v)$. Thus $w \triangleleft x$. However, Lemma 2.3(2) then implies that t must follow s in $\mathscr{L}(v)$, a contradiction, since an occurrence of t precedes that of s on the path from v to the root. □

One consequence of Lemma 2.7 is that $\mathrm{Ex}(ababa, n) = \Omega(n\alpha(n))$. This is one half of Hart and Sharir's proof [13] that $\mathrm{Ex}(ababa, n) = \Theta(n\alpha(n))$.

3. Nonlinear forbidden subsequences.

Lemma 2.4 is a trivial but powerful tool for generating a slew of nonlinear forbidden subsequences. It says that under the correct circumstances we can force symbols to appear in undesirable places. Whereas our forbidden subsequence may be ($ababa$)-free, it may contain $a \ldots b \ldots b \ldots a$. If using Lemma 2.4 we could force an *implied* appearance of a between the two b's, we would arrive at a contradiction, by Lemma 2.7. To create the right circumstances we require an ensemble cast of symbols, each of which will play a specific role in effecting the final contradiction. In the first ensemble cast (the *one-trap* cast) there are five distinct roles, which we call the *binder*, the *trap*, the *inner captive*, the *outer captive*, and the *guard*. In order to specify a forbidden subsequence over the alphabet $\{a, b, c, d, e\}$ we simply need to say which symbol plays which of the five roles. Of the 5! role assignments only a small fraction lead to nonlinear, ($ababa$)-free forbidden subsequences. Symbols can play multiple roles (leading to forbidden subsequences with fewer than five symbols), and some roles can be split among many symbols, which lead to arbitrarily long nonlinear forbidden subsequences. Our second ensemble cast is slightly more complicated than the first. Its five roles (two traps, two binders, and a captive) achieve the same ends via slightly different means.

One virtue of our cast system is that it allows us to reveal the *structure* of a forbidden subsequence using a succinct code. By representing roles as suits we can describe a forbidden subsequence *semantically* as a string over $\{\heartsuit, \rotatebox[origin=c]{180}{$\heartsuit$}, \spadesuit, \rotatebox[origin=c]{180}{$\spadesuit$}, \diamondsuit, \clubsuit, \rotatebox[origin=c]{180}{$\clubsuit$}, (,)\}$, where each suit is identified with a specific role. Without the assistance of this coding system we would have found it impossible to fully explore the space of our nonlinear forbidden subsequences. After setting out some grammatical rules for legal encodings, it becomes very simple to enumerate all possibilities. In sections 3.1 and 3.2 we present the one-trap and two-trap systems for designing forbidden subsequences.

3.1. The one-trap cast.

In order to motivate the cast's five roles we will start with some examples of specific forbidden subsequences.

Theorem 3.1. $\mathrm{Ex}(abcaccbc, n) = \Omega(n\alpha(n))$.

Proof. Suppose that $\sigma = abcaccbc$ were to occur in $\mathscr{S}$. Note that $abab, bcbc \mathrel{\bar{\prec}} \sigma$. By Lemma 2.7 we can therefore eliminate all cases except $a < b < c$. Let $v_{x,k}$ be the vertex in T corresponding to the kth occurrence of x in σ. It follows from the postordering of the labels in $\mathscr{S}$ that $v_{a,1}, v_{b,1}, v_{c,1} \trianglelefteq v_{a,2}$ and that $v_{a,2} \triangleleft v_{c,2} \triangleleft v_{c,3} \trianglelefteq v_{b,2} \triangleleft v_{c,4}$. See Figure 3.1. We apply Lemma 2.4 to the symbols b and c occurring in $\mathscr{L}(v_{c,2}), \mathscr{L}(v_{b,2})$, and $\mathscr{L}(v_{c,4})$ and conclude that b must also appear in $\mathscr{L}(v_{c,2})$.

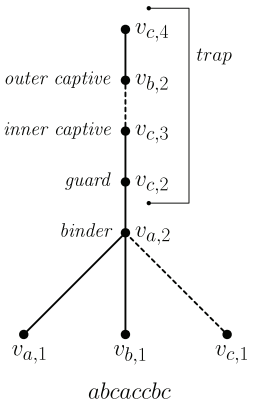

FIG. 3.1. *The vertex $v_{x,k} \in T$ is such that $\mathscr{L}(v_{x,k})$ contains the symbol corresponding to the kth occurrence of x in σ. Dashed lines connect vertices that may be the same.*

In other words, if *abcaccbc* appears in $\mathscr{S}$, then $abcac\underline{\mathbf{b}}cbc$ appears as well. Since, by Lemma 2.7, $\mathscr{S}$ contains no subsequences isomorphic to *bcbcb*, it must also be σ-free. Therefore, $\mathrm{Ex}(\sigma, n) \geq |\mathscr{S}| = \Omega(n\alpha(n))$. □

Let us analyze the functions of a, b, and c in the proof of Theorem 3.1. The symbol a did not appear in the ultimate contradiction (the implied subsequence *bcbcb*), but it did facilitate the contradiction by forcing $v_{b,1}$ and $v_{c,1}$ to be descendants of $v_{c,2}, v_{c,3}, v_{b,2}$, and $v_{c,4}$. In our terminology a is the *binder* (symbolized by ♡) because it binds previous symbols (i.e., vertices in T) under one common ancestor. The locations of b and c were chosen with the preconditions of Lemma 2.4 in mind. For the proof to go through we need c to appear in $\mathscr{L}(v_{c,2})$ and $\mathscr{L}(v_{c,4})$ and b to appear in $\mathscr{L}(v_{b,2})$ and, crucially, that $v_{c,2} \lhd v_{b,2}$. This last condition is enforced by the immediate repetition of c in σ. In our terminology c acts as a *guard*, making sure $v_{b,2}$ is a strict ancestor of $v_{c,2}$, and both b and c are *captives* (one outer, one inner) of the *trap* c, meaning that the symbols b and c appear at vertices that lie strictly above one occurrence of c and strictly below another occurrence of c. The guard, outer captive, inner captive, and trap are represented by ◇, ♠, ♥, and ♣, respectively. Thus, we can represent σ as ♡♠(◇♥♣): a acts as the binder, b as the outer captive, and c as the guard, inner captive, and trap.

Obviously, the discussion above suggests that c's triple role could be replaced by three separate symbols. Before we analyze one-trap casts in their full generality, let us look at one more example.

THEOREM 3.2. $\mathrm{Ex}(abcdebeadce, n) = \Omega(n\alpha(n))$.

Proof. As before, suppose that $\sigma = abcdebeadce \prec \mathscr{S}$, and let $v_{x,k}$ be the vertex in T corresponding to the kth occurrence of x in σ. Since *acac*, *bcbc*, *adad*, *bdbd*, *cece*, and *dede* appear in σ, we can conclude from Lemma 2.7 that $\{a, b\} < \{c, d\} < e$. From the postordering of the vertex labels in $\mathscr{S}$ we have $v_{c,1}, v_{d,1}, v_{e,1} \trianglelefteq v_{b,2}$, i.e., b *binds* the origins of c, d, and e under a common ancestor $v_{b,2}$; see Figure 3.2. We also have that $v_{b,2} \lhd v_{e,2} \trianglelefteq v_{a,2} \lhd v_{d,2} \trianglelefteq v_{c,2} \lhd v_{e,3}$, where the strict descendent relationships come from the fact that vertex labels are in descending order. The purpose of a is to *guard* e from the *outer captive* c and *inner captive* d and, in particular, to guarantee that $v_{c,2} \neq v_{e,2}$. From Lemma 2.4 we conclude that c makes an implied appearance in $\mathscr{L}(v_{e,2})$ and, consequently, that $cd\mathbf{c}dc \prec \mathscr{S}$, a contradiction. □

In our succinct encoding we would express *abcdebeadce* as ◇♡♠♥♣. Generally speaking, the argument employed in Theorem 3.2 will go through if the binder binds,

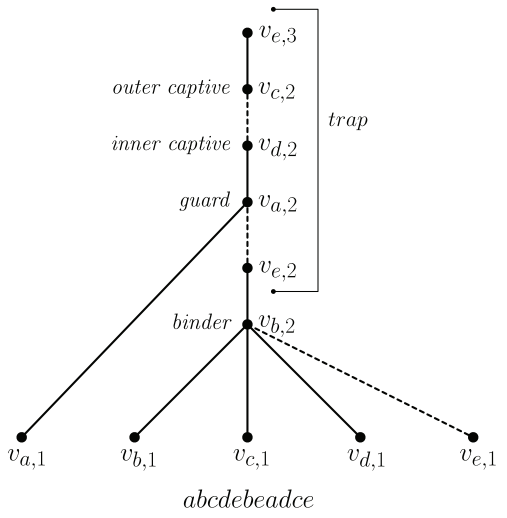

FIG. 3.2. *The vertex $v_{x,k} \in T$ is such that $\mathscr{L}(v_{x,k})$ contains the symbol corresponding to the kth occurrence of x in σ. Dashed lines connect vertices that may be the same.*

the guard guards, and the trap traps. In terms of our encoding system, the binder can only bind if the ♡ precedes ♣, ♠, and ♥, and the guard guards only if the ♢ precedes or is equal to ♥. In order for the forbidden subsequence to be ($ababa$)-free it turns out that the trap ♣ must come last, though it can be equal to the guard and inner captive. What is not obvious is that the binder need not be one symbol. The binding *role*, that is, getting the first occurrences of the captives and trap under a common ancestor, can be played by an arbitrarily large set of *semibinders*. All semibinders are represented by ♡.

3.1.1. One-trap encoding. Definition 3.3 defines the set of legal one-trap encodings in two equivalent ways: as an exhaustive list of regular expressions and as a set of rules. We give a procedure for translating an encoding into a forbidden subsequence, and in Theorem 3.4 we prove that if σ is derived from a legal one-trap encoding, then $\mathrm{Ex}(\sigma, n)$ is nonlinear. Below X^* represents zero or more repetitions of X.

DEFINITION 3.3. *A string over* {♡, ♢, ♣, ♠, ♥, (,)} *is a* legal one-trap encoding *if it appears in* 1–5

1. ♡♠♡*(♢♥♣),
2. ♡♠♡*(♢♥)♡*♣,
3. ♡♠♡*♢♡*♥♡*♣,
4. ♡♢♠♡*♥♡*♣,
5. ♢♡♠♡*♥♡*♣.

or, equivalently, if it satisfies rules (i)–(v)

(i) *It contains one* ♢, ♣, ♠, *and* ♥, *at least one* ♡, *and at most one parenthesized expression, which is either* (♢♥) *or* (♢♥♣).
(ii) ♢ *precedes* ♥.
(iii) ♠ *precedes* ♥.
(iv) *The first* ♡ *precedes* ♠, *which precedes all other* ♡*'s.*
(v) *The last suit is* ♣.

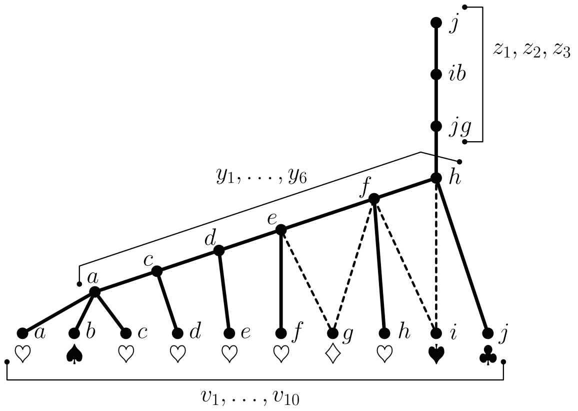
j
ib
jg
z_1, z_2, z_3
$y_1, \ldots, y_6$
h
f
e
d
c
a
a b c d e f g h i j
$\heartsuit$ $\spadesuit$ $\heartsuit$ $\heartsuit$ $\heartsuit$ $\heartsuit$ $\diamondsuit$ $\heartsuit$ $\varheartsuit$ $\clubsuit$
$v_1, \ldots, v_{10}$

FIG. 3.3. *An example τ_λ. The dashed lines represent possible parent-child relationships. We have $\sigma_\lambda = abcadcedf[eg, ge]h[fi, if]jhjgibj$, where the bracketed expressions correspond to the parent options of v_7 and v_9.*

The *illegal* encodings are illegal for one of three reasons: either the corresponding forbidden subsequence contains *ababa* and is trivially nonlinear, or it cannot be proved nonlinear by our method, or it is redundant, i.e., a strict subsequence can be proved nonlinear.

3.1.2. Translating an encoding. Let λ be a legal one-trap encoding. We create a forbidden subsequence in two stages. In the first we translate λ into a rooted vertex labeled tree τ_λ; in the second we translate τ_λ into a sequence σ_λ. There is often a little ambiguity in encoding λ, which lets us choose τ_λ among $O(1)$ possibilities. The sequence σ_λ depends solely on τ_λ.

Let $|\lambda|$ be the number of its symbols, where a parenthesized expression $(\diamondsuit\spadesuit)$ or $(\diamondsuit\spadesuit\clubsuit)$ counts as one symbol, and let $\lambda(j)$ be its jth symbol. Our forbidden subsequence will be over the alphabet $\{1, \ldots, |\lambda|\}$.

If λ contains B (semi)binders, let $\{j_i^\heartsuit\}_{1 \le i \le B}$ be the indices for which $\lambda(j_i^\heartsuit) = \heartsuit$. Similarly, let $j^\diamondsuit, j^\spadesuit, j^\varheartsuit, j^\clubsuit$ be the indices of the guard, outer captive, inner captive, and trap, respectively. If λ contains a parenthesized expression, some of $j^\diamondsuit, j^\varheartsuit, j^\clubsuit$ may be equal. As we describe the general construction of τ_λ, the reader may wish to examine $\tau_{\heartsuit\spadesuit\heartsuit\heartsuit\heartsuit\heartsuit\diamondsuit\heartsuit\varheartsuit\clubsuit}$ depicted in Figure 3.3.

We create $|\lambda|$ leaf nodes $v_1, \ldots, v_{|\lambda|}$ and $B + 3$ internal nodes $y_1, \ldots, y_B, z_1, z_2, z_3$. The nodes are arranged as follows:

1. $y_1, \ldots, y_B, z_1, z_2, z_3$ appear on a path in least-to-most ancestral order; i.e., y_1 and z_3 are the least and most ancestral nodes.
2. $v_{j_1^\heartsuit}$ is the child of y_1, and for $k > 1$, $v_{j_k^\heartsuit}$ is the child of y_{k-1}.
3. $v_{j^\clubsuit} = v_{|\lambda|}$ is the child of y_B. Define the artificial index $j_{B+1}^\heartsuit = |\lambda| + 1$. If an index $j \in \{j^\spadesuit, j^\varheartsuit, j^\diamondsuit\}$ appears strictly between $j_k^\heartsuit$ and $j_{k+1}^\heartsuit$, then v_j can be made the child of either y_{k-1} or y_k. (Note first that if $k = 1$, then y_{k-1} does not exist and second that Definition 3.3(iv) implies $j_1^\heartsuit < j^\spadesuit < j_2^\heartsuit$.) If two such indices $j < j'$ appear between $j_k^\heartsuit$ and $j_{k+1}^\heartsuit$, their choice of parents cannot cross; i.e., v_j cannot be the child of y_k while $v_{j'}$ is the child of y_{k-1}.
4. If $j^\diamondsuit = 1$, i.e., if λ belongs to Definition 3.3(5), v_1 is the child of z_1.

The labels are assigned as follows. Let v_k get the label (k), y_k get the label $(j_k^\heartsuit)$, and z_1, z_2, z_3 get the labels $(j^\clubsuit j^\diamondsuit)$, $(j^\varheartsuit j^\spadesuit)$, and $(j^\clubsuit)$. There is only one exception: if

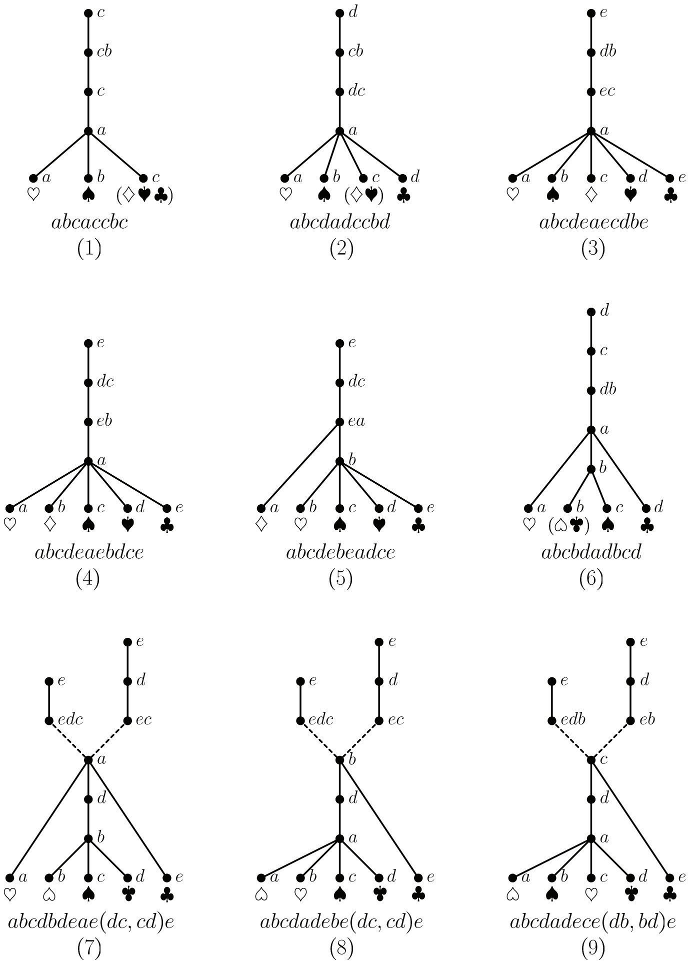

FIG. 3.4. *The primitive forms of our one-trap* (1–5) *and two-trap* (6–9) *encodings. The symbolic encodings* (7–9) *can be translated into two distinct sequences via distinct trees.*

$j^{\clubsuit} = j^{\diamondsuit} = j^{\varheartsuit}$, i.e., λ is in Definition 3.3(1), z_1 gets the label $(j^{\clubsuit})$. This concludes the construction of τ_λ. One can verify that there are at most four ways to map λ to τ_λ; it could be the case that both $v_{j^{\diamondsuit}}$ and $v_{j^{\varheartsuit}}$ have a choice between two parents.

We generate σ_λ by concatenating the vertex labels in the unique postorder traversal of τ_λ in which $v_1, \ldots, v_{|\lambda|}$ appear in that order. The similarity between the construction of σ_λ from τ_λ and that of $\mathscr{S}$ from T is no accident. The tree τ_λ captures the necessary ancestor-descendant relationships between nodes in T that are involved in some (hypothetical) occurrence of $\sigma_\lambda \prec \mathscr{S}$.

Figure 3.4 summarizes the primitive types (those with the fewest semibinders) of our one-trap and two-trap encodings. Some of the two-trap primitive types can be translated into different trees and hence different forbidden subsequences. In one-trap encodings, ambiguity in the encoding-to-tree translation arises only when there are multiple semibinders.

Let Ψ_1 be the set of all sequences that can be generated from a legal one-trap encoding.

THEOREM 3.4. *For all* $\sigma \in \Psi_1$, $\mathrm{Ex}(\sigma, n) = \Omega(n\alpha(n))$.

A proof of Theorem 3.4 appears in Appendix A.

3.2. The two-trap cast. A two-trap cast also has five roles: the *inner* and *outer* traps, the *inner* and *outer* binders, and a single captive, which are symbolized by $\rotatebox[origin=c]{180}{$\clubsuit$}$, ♣, ♤, ♡, and ♠, respectively. As before, we motivate these roles by first analyzing some specific examples from scratch.

THEOREM 3.5. $\mathrm{Ex}(abcdbdeaedce, n) = \Omega(n\alpha(n))$.

Proof. Let $v_{x,k}$ be defined as in Theorems 3.1 and 3.2. It follows from the construction of $\mathscr{S}$ that $\{v_{b,1}, v_{c,1}, v_{d,1}\} \trianglelefteq v_{b,2} \triangleleft v_{d,2}$ and that $\{v_{a,1}, v_{d,2}, v_{e,1}\} \trianglelefteq v_{a,2} \triangleleft v_{e,2} \trianglelefteq v_{d,3} \trianglelefteq v_{c,2} \triangleleft v_{e,3}$. See Figure 3.5. Lemma 2.4 applied to the pairs c, e and d, e shows that c and d both appear in $\mathscr{L}(v_{e,2})$. We can then apply Lemma 2.4 to the pair c, d, which shows that c also appears in $\mathscr{L}(v_{d,2})$. Thus, if $abcdbdeaedce \prec \mathscr{S}$, then, $cdd\mathbf{c}dc \prec \mathscr{S}$ as well, a contradiction. □

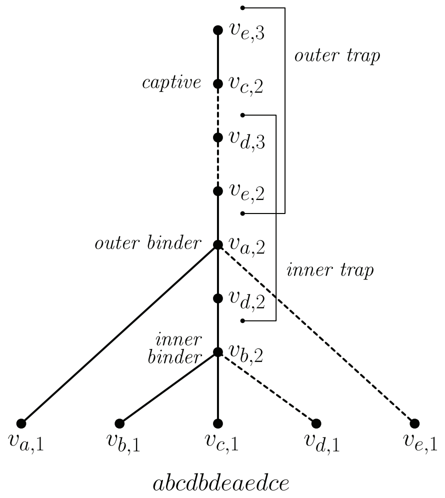

FIG. 3.5. *The vertex* $v_{x,k} \in T$ *is such that* $\mathscr{L}(v_{x,k})$ *contains the symbol corresponding to the kth occurrence of* x *in* σ. *Dashed lines connect vertices that may be the same.*

In the proof of Theorem 3.5 a and b simply acted as binders, with two different functions. It was b's role to ensure that $v_{d,2}$ was ancestral to both $v_{c,1}$ and $v_{d,1}$, and it was a's role to ensure that $v_{e,2}$ was ancestral to both $v_{d,2}$ and $v_{e,1}$. The symbol c was a captive in e's trap, and, after a couple applications of Lemma 2.4, an implied occurrence of c (in $\mathscr{L}(v_{e,2})$) pops up in d's trap. We call e and d the outer and inner trap, and a and b the outer and inner binder. Thus, $abcdbdeaedce$ is represented as ♡♤♠$\rotatebox[origin=c]{180}{$\clubsuit$}$♣.

As in the one-trap system, it is possible for one symbol to play more than one of the five roles. Let us briefly look at one such example and the resulting encoding before moving on.

THEOREM 3.6. $\mathrm{Ex}(abcbdadbcd) = \Omega(n\alpha(n))$.

Proof. Suppose that $abcbdadbcd \prec \mathscr{S}$. It follows from $\mathscr{S}$'s construction that $\{v_{c,1}, v_{b,1}\} \trianglelefteq v_{b,2}$ and that $\{v_{b,2}, v_{d,1}\} \trianglelefteq v_{a,2} \triangleleft v_{d,2} \trianglelefteq v_{b,3} \triangleleft v_{c,2} \trianglelefteq v_{e,3}$. See Figure 3.6. Lemma 2.4 shows that b and c appear in $\mathscr{L}(v_{d,2})$, and another application of Lemma 2.4 shows that c appears in $\mathscr{L}(v_{b,2})$. Thus, if $abcbdadbcd \prec \mathscr{S}$, then $bcb\mathbf{c}bc \prec \mathscr{S}$ as well, a contradiction. □

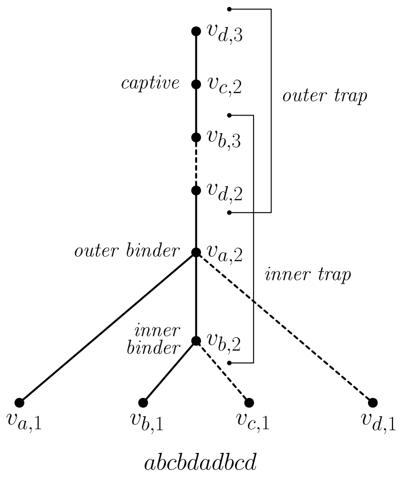

FIG. 3.6. *Solid lines represent strict ancestor-descendant relationships. Dashed lines connect vertices that may be the same.*

In Theorem 3.6 d played the role of the outer trap, a the outer binder, and c the captive. The inner trap (into which the implied occurence of the captive c fell) was played by b, which also doubled as the inner binder. Thus, the encoding for this forbidden subsequence is ♡(♤✤)♠♣.

3.2.1. Two-trap encoding. As in the one-trap system, there are several stringent rules for generating legal encodings. With the exception of ♡(♤✤)♠♣ the captive must precede the inner trap, which precedes the outer trap. There are several ways the inner and outer binder can be positioned, and, furthermore, the duties of *both* the inner and outer binders can be split among many symbols. Definition 3.7 gives all legal encodings in the two-trap system. Below we show how to translate a legal encoding into a sequence.

DEFINITION 3.7. *A string over* {♡, ♤, ♣, ✤, ♠, (,)} *is a* legal two-trap encoding *if it appears in* 6–9

6. ♡(♤✤)♠♡*♣,
7. ♡♤♠♤*✤♡*♣,
8. ♤♡♤*♠♤*✤♡*♣,
9. ♤♠♤*♡♤*✤♡*♣.

or, equivalently, if it satisfies rules (vi)–(x)

(vi) *It contains one* ♠, ♣, *and* ✤, *at least one* ♡, *and at least one* ♤.
It is either in ♡(♤♣)♠♡*♣ *or satisfies* (vii)–(x).
(vii) *The first* ♤ *precedes* ♠, *which precedes* ✤.
(viii) *Exactly one* ♡ *precedes* ✤; *all* ♤*'s precede* ✤.
(ix) *The first two* ♤*'s are not adjacent.*
(x) *The last suit is* ♣.

3.2.2. Translating an encoding. Let λ be a legal two-trap encoding. As before we translate λ into a labeled tree τ_λ and then into a sequence σ_λ. The slight ambiguity found in the one-trap system also arises here; there are often two or more equally good topologies for τ_λ. In the two-trap system there is another degree of ambiguity. As we will see, the nodes of τ_λ can usually be labeled in two slightly different ways. One may wish to follow along with the two specific examples of τ_λ depicted in Figure 3.7 for $\lambda = ♤♠♤^2♡♤✤♡^2♣$ and $♡♤♠♤✤♡^2♣$.

Let $j^{♣}, j^{✤}, j^{♠}$ be the indices in λ of the outer trap, inner trap, and captive. Let $\{j_k^{♡}\}_{1\le k\le B}$ and $\{j_k^{♤}\}_{1\le k\le B'}$ be the indices of the outer and inner (semi)binders, respectively.

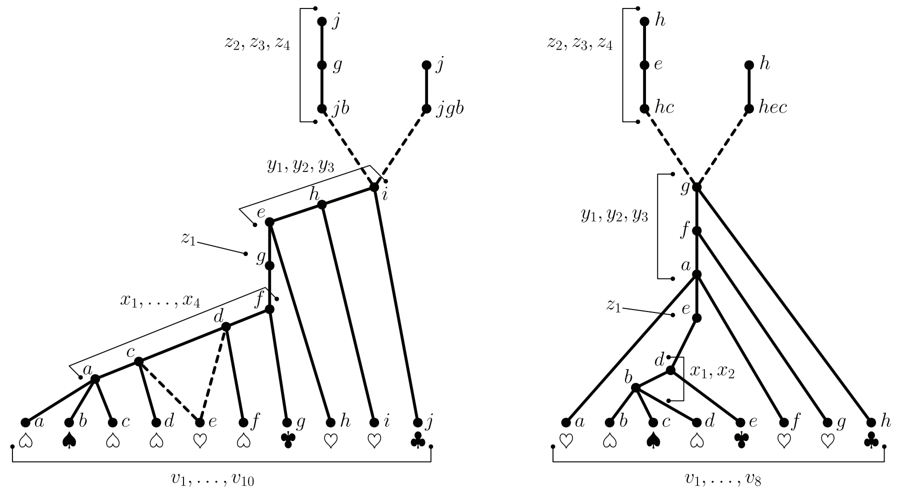

FIG. 3.7. *(Left) an example τ_λ for which $\sigma_\lambda = abcad[ce, ec]fdgfgheihjij[gb, bg]j$. The bracketed expressions reflect the possible parents of v_5 and y_3. (Right) an example τ_λ for which $\sigma_\lambda = abcdbedefagfhgh[ec, ce]h$.*

Our model tree is made up of leaves $v_1, \ldots, v_{|\lambda|}$ and up to $B + B' + 4$ internal nodes $x_1, \ldots, x_{B'}, y_1, \ldots, y_B, z_1, \ldots, z_4$. The inner binder nodes are arranged in a path $x_1, \ldots, x_{B'}$, from least-to-most ancestral order, as are the outer binder nodes $y_1, \ldots, y_B$ and outer trap nodes z_2, z_3, z_4. The inner trap node z_1 is the parent of $x_{B'}$, and the child of y_1 and y_B is the child of z_2. There is one exception; if $\lambda \in ♡(♤✤)♠♡^*♣$ (Definition 3.7(6)), then $j_1^{♤} = j_{B'}^{♤} = j^{✤}$. In this case z_1 does not exist; i.e., $x_{B'} = x_1$ is the child of y_1. The parentage of the leaves are assigned as follows:

1. $v_{j_1^{♤}}$ is a child of x_1.
2. $v_{j_k^{♤}}$ is a child of x_{k-1} for $1 < k \le B'$.
3. $v_{j^{✤}}$ is a child of $x_{B'}$.
4. If a symbol j appears between $j_k^{♤}$ and $j_{k+1}^{♤}$, v_j can be the child of either $x_{j_{k-1}^{♤}}$ or $x_{j_k^{♤}}$. (Note that for $k = 1$, $j_{k-1}^{♤}$ does not exist.) If it appears between $j_{B'}^{♤}$ and $j^{✤}$, its parent can be either $x_{B'-1}$ or x_B.
5. If $j_1^{♡} < j_1^{♤}$, i.e., the parent of $v_{j_1^{♡}}$ has not already been accounted for in (4), we let $v_{j_1^{♡}}$ be the child of y_1.
6. $v_{j_k^{♡}}$ is the child of $y_{j_{k-1}^{♡}}$ for $1 < k \le B$.
7. $v_{j^{♣}}$ is the child of y_B.

The labeling of nodes is as follows. Each leaf v_k gets the label (k), and each x_k gets the label $(j_k^{♤})$ (for $k \in [1, B']$), and each y_k gets the label $(j_k^{♡})$ (for $k \in [1, B]$). There are two ways to label $z_1, \ldots, z_4$. In the first $z_1, \ldots, z_4$ get the labels $(j^{✤}), (j^{♣}j^{♠}), (j^{✤})$, and $(j^{♣})$, respectively. In the second they get the labels $(j^{✤}), (j^{♣}j^{✤}j^{♠}), (j^{♣})$, and $()$ (empty label), respectively. This concludes the construction of τ_λ. The sequence σ_λ

is derived by concatenating the vertex labels of τ_λ in the unique postorder in which $v_1, \ldots, v_{|\lambda|}$ appear in that order.

Let Ψ_2 be the set of all sequences that can be generated from a legal two-trap encoding, and let $\Psi = \Psi_1 \cup \Psi_2$. A proof of Theorem 3.8 appears in the Appendix B.

THEOREM 3.8. *For all* $\sigma \in \Psi_2$, $\mathrm{Ex}(\sigma, n) = \Omega(n\alpha(n))$.

3.3. A lower bound on the size of $\boldsymbol{\Phi_{Nonlin}}$. Klazar [17] proved that Φ_{Nonlin} contains at least two elements, though he could only identify one of them, namely, *ababa*. The second is some subsequence of *abcbdadbcd*. In the same vein we prove, nonconstructively, that $|\Phi_{Nonlin}| \geq 3$.

THEOREM 3.9. $|\Phi_{Nonlin}| \geq 3$.

Proof. Consider the one-trap encodings ♡♠(♢♥♣) and ♡♠♡(♢♥♣) in Definition 3.3(1), which correspond to the nonlinear forbidden subsequences $\sigma_1 = abcaccbc$ and $\sigma_2 = abcadcddbd$, both in Ψ_1. One consequence of [20] is that any sequence over $\{a, b, c\}$ is linear (specifically, in Φ_{KV}) unless it contains $ababa, abcacbc, abcbcac$, or their reversals. That is, the only strict subsequence of σ_1 not known to be linear is $\sigma_1' = abcacbc$. One may check that σ_1' is not a subsequence of σ_2 or $\overline{\sigma_2}$. Thus, Φ_{Nonlin} must contain at least three elements: *ababa* and two subsequences of σ_1 and σ_2. ◻

Every sequence in Ψ contains either *abcacbc* or *abcbcac*, so as long as the status of these remain open, it will be very difficult to improve our lower bounds on $|\Phi_{Nonlin}|$.

4. Davenport–Schinzel trees. As an interim step in our construction of the nonlinear sequences $\mathscr{S}_{i,j}$, we constructed a set of model trees $\{T(i,j)\}_{i,j}$. The only salient difference between our model trees and the Davenport–Schinzel trees studied in [33] is in the labeling. The node labels in our model tree were sequences of symbols, whereas in Davenport–Schinzel trees they are single symbols. By replacing every node $v \in T(i,j)$ with a path of $|\mathscr{L}(v)|$ nodes, each labeled with one symbol from $\mathscr{L}(v)$, we can rederive some of Valtr's results [33] using the machinery developed in section 2.

In section 4.3 we introduce a simple coding scheme for nonlinear forbidden subsequences (with respect to Davenport–Schinzel trees) that is similar in spirit to the one-trap and two-trap encodings from section 3. By enumerating valid encodings we are able to generate an infinite antichain of nonlinear forbidden subsequences.

4.1. Notation and definitions. For a tree Z with node set $V(Z)$, a function $\mathscr{L}^T : V(Z) \to \{1, \ldots, n\}$ is a valid, c-sparse labeling if (i) for $1 \leq j \leq n$, the nodes in ${\mathscr{L}^T}^{-1}(j)$ lie along a path in Z, and (ii) if $\mathscr{L}^T(v_1) = \mathscr{L}^T(v_2)$, then v_1 and v_2 are at distance at least c.

The number of nodes in Z and the alphabet size of Z (the number of distinct labels) are $|Z|$ and $\|Z\|$, respectively. An *in-tree* is one whose edges are directed toward a distinguished root vertex. If σ is a sequence and Z a labeled (in-)tree, the notation $\sigma \prec^T Z$ ($\sigma \prec^{\vec{T}}$) means that for some sequence σ' corresponding to a (directed) path in Z, $\sigma \prec \sigma'$. The relations $\precsim^T$ and $\precsim^{\vec{T}}$ are defined analogously. If $\sigma \not\prec^T Z$ (or $\sigma \not\prec^{\vec{T}}$), then Z is *σ-free.*

DEFINITION 4.1. *Analogous to the extremal function* Ex *for sequences,* Ex^T *and* $\mathrm{Ex}^{\vec{T}}$ *are defined as*

$$\begin{aligned}
\mathrm{Ex}^T(\sigma, n) = \max\big\{\, |Z| \;:\;\; & \textit{Z is a } \sigma\textit{-free, } \|\sigma\|\textit{-sparse, validly labeled} \\
& \textit{undirected tree with } \|Z\| = n \big\}, \\
\mathrm{Ex}^{\vec{T}}(\sigma, n) = \max\big\{\, |Z| \;:\;\; & \textit{Z is a } \sigma\textit{-free, } \|\sigma\|\textit{-sparse, validly labeled} \\
& \textit{in-tree with } \|Z\| = n \big\}.
\end{aligned}$$

Valtr [33] defined $\mathrm{Ex}^{\vec{T}}$ with respect to out-trees, not in-trees, but the two are obviously equivalent. We prefer in-trees because it is consistent with our construction of $\mathscr{S}$ from section 2.

Definition 4.2. *Let Φ^{T}_{Nonlin} and $\Phi^{\vec{T}}_{Nonlin}$ be minimal sets such that*

$$\mathrm{Ex}^{T}(\sigma, n) = \omega(n) \quad \textit{if and only if} \quad \exists \mu \in \Phi^{T}_{Nonlin} : \mu \prec \sigma \quad \textit{or} \quad \overline{\mu} \prec \sigma,$$
$$\mathrm{Ex}^{\vec{T}}(\sigma, n) = \omega(n) \quad \textit{if and only if} \quad \exists \mu \in \Phi^{\vec{T}}_{Nonlin} : \mu \prec \sigma.$$

Note that, unlike with sequences and undirected trees, the identity $\mathrm{Ex}^{\vec{T}}(\sigma, n) = \mathrm{Ex}^{\vec{T}}(\overline{\sigma}, n)$ does not necessarily hold. If the in-tree Z is σ-free, we can only say that the out-tree version of Z is $\overline{\sigma}$-free.

4.2. Construction of trees. Let $\mathscr{T}$ be a labeled tree derived from T through the following transformations. The vertex set of $\mathscr{T}$ is a superset of T, and the number of nodes in $\mathscr{T}$ is equal to total length of all vertex labels in T, i.e., $|\mathscr{T}| = |\mathscr{S}|$. If v is a vertex in T with parent p and label $\mathscr{L}(v) = (a_1, \ldots, a_k)$, there exist vertices $v_1 = v, v_2, \ldots, v_k$ in $\mathscr{T}$, where v_j is the child of v_{j+1}, v_k is the child of p, and $\mathscr{L}^T(v_j) = a_j$. Just as we referred to T when analyzing $\mathscr{S}$, we still identify every $\mathscr{T}$-vertex v_j with the T-vertex v. We use $\mathscr{T}$ and $\vec{\mathscr{T}}$ to refer to its undirected and directed versions.

Lemma 4.3 (relations between $\mathscr{S}, \mathscr{T}, \vec{\mathscr{T}}$).

1. *If $\sigma \prec^{T} \vec{\mathscr{T}}$, then $\sigma \prec \mathscr{S}$.*
2. *If σ cannot be written as $\sigma_1\sigma_2$ where σ_1 and σ_2 have disjoint alphabets, then $\sigma \prec^{T} \mathscr{T}$ implies that $\sigma \prec \mathscr{S}$ or $\overline{\sigma} \prec \mathscr{S}$.*

Proof. Part (1) follows from the fact that $\mathscr{S}$ was generated from T by concatenating labels in postorder, which is consistent with the direction of paths in $\vec{\mathscr{T}}$. For part (2), let P be a minimal (oriented) path in $\mathscr{T}$ containing an occurrence of σ. If P connects a node to one of its ancestors, then $\sigma \prec \mathscr{S}$, and if it connects an ancestor to a descendent, then $\overline{\sigma} \prec \mathscr{S}$. If neither of the above hold, let u be the least common ancestor of the endpoints of P. Then u divides σ into two pieces σ_1, σ_2. From the construction of T (and by extension, $\mathscr{T}$), no symbol can appear in the labels of two unrelated nodes. Thus σ_1 and σ_2 have disjoint alphabets. □

Lemma 4.3(1) implies that $\mathrm{Ex}^{\vec{T}}(ababa, n) \geq |\mathscr{T}| = \Omega(n\alpha(n))$, where $n = \|\mathscr{T}\|$. Since $ababa$ is a palindrome, Lemma 4.3(2) implies that $\mathrm{Ex}^{T}(ababa, n) = \Omega(n\alpha(n))$. We treat $ababa$ as the prototypical nonlinear forbidden subsequence for Ex^{T} and $\mathrm{Ex}^{\vec{T}}$ and derive many other nonlinear sequences from it. Lemma 4.4 is analogous to Lemma 2.4

Lemma 4.4. *Suppose $abab \prec^{\vec{T}} \vec{\mathscr{T}}$ and let u be the T-node corresponding to the first occurrence of b in $abab$. Then a appears after b in $\mathscr{L}(u)$, i.e., $a < b$.*

Proof. Let $v_{x,k}$, $x \in \{a, b\}, k \in \{1, 2\}$ be the T-node corresponding to the kth occurrence of x in $abab$. We have, trivially, that $v_{a,1} \trianglelefteq v_{b,1} \trianglelefteq v_{a,2} \trianglelefteq v_{b,2}$. If $u = v_{b,1}$ is an i-node, then by Lemma 2.4 it must be the first i-node on the path from $v_{b,2}$ to the origin of b. Since $v_{a,2} \trianglelefteq v_{b,2}$ and a originates below $v_{b,1}$, u is also the first i-node on the path from $v_{a,2}$ to the origin of a. By Lemma 2.4, a appears in $\mathscr{L}(u)$ and, by Lemma 2.3, $a < b$. □

Theorem 4.5 reproves some results of Valtr [33].

Theorem 4.5. *For $\sigma \in \{abbab, abcabc, abcbac\}$, $\mathrm{Ex}^{\vec{T}}(\sigma, n) = \Omega(n\alpha(n))$.*

Proof. Define $v_{x,k}$ as usual. If $\sigma = abbab \prec^{\vec{T}} \vec{\mathscr{T}}$, by Lemma 4.4 a must appear in $\mathscr{L}(v_{b,1})$ and $\mathscr{L}(v_{b,2})$, implying that $ababab \prec^{\vec{T}} \vec{\mathscr{T}}$, a contradiction. For $\sigma = abcabc$ or $\sigma = abcbac$, Lemma 4.4 implies that a and b appear in $\mathscr{L}(v_{c,1})$. Since a originates below $v_{b,1}$, Lemma 2.4 then implies that a also appears in $\mathscr{L}(v_{b,1})$. Thus, whether $\sigma = abcabc$ or $abcbac$, if $\sigma \prec^{\vec{T}} \vec{\mathscr{T}}$, then $ababa \prec^{\vec{T}} \vec{\mathscr{T}}$ as well, a contradiction. □

Since *abcabc* is a palindrome, Lemma 4.3 implies that $\mathrm{Ex}^T(abcabc, n) = \Omega(n\alpha(n))$. However, the same conclusion cannot be drawn for *abcbac* or *abbab* since their reversals *do* appear in $\vec{\mathscr{T}}$ and hence $\mathscr{T}$ as well. Theorem 4.6 provides a general way to construct new nonlinear forbidden subsequences. To show that $\mathrm{Ex}^T(\sigma, n) = \Omega(n\alpha(n))$ we just have to make sure that σ cannot appear on a path directed toward the root, a path directed away from the root, or a path connecting two unrelated vertices.

THEOREM 4.6. *Let $\sigma' \in \{abbab, abcbac\}$ and $\sigma'' \in \{abaab, abcacb\}$. Let σ be any minimal sequence such that* (i) *$\sigma', \sigma'' \prec \sigma$, and* (ii) *$\sigma$ cannot be written as $\sigma_1\sigma_2$ where σ_1 and σ_2 have disjoint alphabets and $\sigma'' \sim \sigma_1, \sigma' \sim \sigma_2$. Then $\mathrm{Ex}^T(\sigma, n) = \Omega(n\alpha(n))$.*

Proof. Suppose an occurrence of σ in $\mathscr{T}$ lies on an oriented path from v to w, and let u be the least common ancestor of v and w. If $u = w$ (w is ancestral to v), then $\sigma' \prec \sigma \prec^{\vec{T}} \vec{\mathscr{T}}$, a contradiction by Theorem 4.5. If $u = v$ (v is ancestral to w), then $\overline{\sigma''} \prec \overline{\sigma} \prec^{\vec{T}} \vec{\mathscr{T}}$, also a contradiction by Theorem 4.5. Otherwise u breaks σ into pieces $\sigma_1\sigma_2$ with disjoint alphabets. Because σ is minimal there are only two possibilities: $\sigma_1 \sim \sigma'$ and $\sigma_2 \sim \sigma''$ or the reverse. In the first case $\sigma_1 \sim \sigma' \prec^{\vec{T}} \vec{\mathscr{T}}$, a contradiction; the second case is precluded by the assumptions. Thus, in all cases $\mathrm{Ex}^T(\sigma, n) \geq |\mathscr{T}| = \Omega(n\alpha(n))$. □

4.3. Encoding nonlinear forbidden subsequences. Let Ψ^T be the set consisting of $\{ababa, abcabc\}$ and the *minimal* nonlinear forbidden subsequences implied by Theorem 4.6; i.e., Ψ^T is an antichain with respect to $\prec$. In order to systematically explore Ψ^T it is useful to have a succinct code that expresses how a nonlinear subsequence σ is selected in Theorem 4.6. We have two ways to choose σ', two ways to choose σ'', some number of ways to decide how the alphabets of σ' and σ'' intersect, and, finally, to decide how many new symbols (not appearing in $\Sigma(\sigma') \cup \Sigma(\sigma'')$) to introduce. After these choices are made there are numerous ways to select a minimal sequence σ that contains both σ' and σ'' and satisfies Theorem 4.6(ii). Nearly all of the sequences generated in this way will be redundant, inasmuch as they contain a strict subsequence already known to be nonlinear.

In our code we use male and female (♂ and ♀) to indicate the symbols belonging to the alphabets of σ' and σ'', respectively. If there are two ♂'s, then σ' will be isomorphic to *abbab*; if there are three ♂'s, then σ' will be isomorphic to *abcbac*. In a symmetric fashion two ♀'s and three ♀'s correspond to σ'' being isomorphic to *abaab* and *abcacb*. A symbol can take part in both σ' and σ''; in this case it would be a hermaphrodite ⚥. Let us look at a few examples. The code ⚥⚥ represents a set of forbidden subsequences over two letters, say, a and b. There are two ♂'s and two ♀'s, each identified with the letters a and b, so σ' must equal (not be isomorphic to) *abbab* and σ'' must equal *abaab*. For sequences X, Y let $X \oplus Y$ be the set of minimal supersequences containing both X and Y. (We consider only normalized sequences in which the first appearance of a precedes the first appearance of b, and so on.) By Theorem 4.6 every member $\sigma \in abbab \oplus abaab = \{abbaab, ababab\}$ has $\mathrm{Ex}^{\vec{T}}(\sigma, n) = \Omega(n\alpha(n))$. We already know *ababab* is nonlinear, but *abbaab* is new. The code ⚥♀♂ is over a three letter alphabet, say, $\{a, b, c\}$, in which σ' (corresponding to the ♂'s) is *accac* and σ'' (the ♀'s) is *abaab*. The sequences in $abaab \oplus accac = \{abccaacb, abaabccac, abcaacacb, \ldots\}$ do not add anything to our repertoire of minimal nonlinear forbidden subsequences because they all contain a substring isomorphic to *ababa*, *abcabc*, or *abbaab*. For one last example, consider ⚥♀⚥. Here there is a three letter alphabet: two ♂'s, so $\sigma' = accac$, 2-letter Sequences

Code	Translation	New Seqs.
⚥⚥	$abbab \oplus abaab$	$abbaab$

3-letter Sequences

Code	Translation	New Seqs.
♂♀⚥	$accac \oplus bcbbc$	—
♂⚥♀	$abbab \oplus bcbbc$	$abbacbbc$
♂⚥⚥	$abcbac \oplus bcbbc$	$abcbbac$
♀⚥♂	$abaab \oplus bccbc$	$abaaccbc$
♀♂⚥	$acaac \oplus bccbc$	—
♀⚥⚥	$abcacb \oplus bccbc$	—
⚥♂♀	$abbab \oplus acaac$	—
⚥♂⚥	$abcbac \oplus acaac$	$abcbaac$
⚥♀♂	$abaab \oplus accac$	—
⚥♀⚥	$accac \oplus abcacb$	$abccacb$
⚥⚥♂	$abaab \oplus abcbac$	—
⚥⚥♀	$abbab \oplus abcacb$	$abbcacb$
⚥⚥⚥	$abcacb \oplus abcbac$	—

4-letter Sequences

Code	Translation	New Seqs.
♂♂♀♀	$abbab \oplus cdccd$	$abbabcdccd$, $abbacbdccd$, etc.
♂♂♀⚥	$abdbad \oplus cdccd$	—
♂♂⚥♀	$abcbac \oplus cdccd$	$abcbadccd$
♂♀♂♀	$accac \oplus bdbbd$	—
♂♀♂⚥	$acdcad \oplus bdbbd$	—
♂♀♀♂	$addad \oplus bcbbc$	—
♂♀♀⚥	$addad \oplus bcdbdc$	—
♂♀⚥♀	$accac \oplus bcdbdc$	—
♂♀⚥♂	$acdcad \oplus bcbbc$	$abcbbdcad$
♂♀⚥⚥	$acdcad \oplus bcdbdc$	—
♂⚥♂♀	$abcbac \oplus bdbbd$	—
♂⚥♀♂	$abdbad \oplus bcbbc$	—
♂⚥♀♀	$abbab \oplus bcdbdc$	$abbabcdbdc$, $abbacdbdc$

4-letter Sequences, cont.

Code	Translation	New Seqs.
♂⚥♀⚥	$abdbad \oplus bcdbdc$	—
♂⚥⚥♀	$abcbac \oplus bcdbdc$	$abcdbdac$, $abcdbadc$
♀♂♂♀	$adaad \oplus bccbc$	$abccbcdaad$, $abccbdcaad$
♀♂♂⚥	$adaad \oplus bcdcbd$	—
♀♂♀♂	$acaac \oplus bddbd$	—
♀♂♀⚥	$acdadc \oplus bddbd$	—
♀♂⚥♂	$acaac \oplus bcdcbd$	$abcaadcbd$
♀♂⚥♀	$acdadc \oplus bccbc$	$abccbdadc$
♀♂⚥⚥	$acdadc \oplus bcdcbd$	—
♀♀♂♂	$abaab \oplus cddcd$	$abaacbddcd$, $abaacdbdcd$, etc.
♀♀♂⚥	$abdadb \oplus cddcd$	—
♀♀⚥♂	$abcacb \oplus cddcd$	$abcaddcdb$
♀⚥♂♂	$abaab \oplus bcdcbd$	$abaacdcbd$
♀⚥♂♀	$abdadb \oplus bccbc$	—
♀⚥♂⚥	$abdadb \oplus bcdcbd$	—
♀⚥♀♂	$abcacb \oplus bddbd$	$abcacddbd$
♀⚥⚥♂	$bcdcbd \oplus abcacb$	$abcadcbd$
⚥♂♂♀	$abcbac \oplus adaad$	—
⚥♂♀♂	$abdbad \oplus acaac$	—
⚥♂♀♀	$abbab \oplus acdadc$	—
⚥♂♀⚥	$abdbad \oplus acdadc$	—
⚥♂⚥♀	$abcbac \oplus acdadc$	$abcbdadc$
⚥♀♂♂	$abaab \oplus acdcad$	—
⚥♀♂♀	$abdadb \oplus accac$	—
⚥♀♂⚥	$abdadb \oplus acdcad$	—
⚥♀♀♂	$abcacb \oplus addad$	—
⚥♀⚥♂	$abcacb \oplus acdcad$	—
⚥⚥♂♀	$abcbac \oplus abdadb$	—
⚥⚥♀♂	$abcacb \oplus abdbad$	—

FIG. 4.1. *Encodings for forbidden subsequences that are nonlinear for* Ex^T.

and three ♀'s, so $\sigma'' = abcacb$. Among the sequences in $accac \oplus abcacb$ only $abccacb$ is not already known to be nonlinear. Of course, this implies that $\overline{abccacb} \sim abcbbac$ is also nonlinear, which corresponds to the encoding ♂⚥⚥.

The encodings and consequences for 2-, 3-, and 4-letter forbidden subsequences are given in Figure 4.1. One could continue to generate 5- and 6-letter forbidden subsequences in the same way. However, it is not clear how Theorem 4.6 could generate sequences over larger alphabets. To see how it does, consider the encoding ♀♀♂♂, which corresponds to sequences in $S = \sigma_1 \oplus \sigma_2$, where $\sigma_1 = abaab$ and $\sigma_2 = cddcd$. All sequences in S are nonlinear with the exception of $\sigma = \sigma_1\sigma_2$, which violates Theorem 4.6(ii). (Because the alphabets of σ_1 and σ_2 do not intermingle, it is possible for σ to appear on a path in $\mathscr{T}$ between unrelated nodes.) One way to correct this problem is to link σ_1 and σ_2 together using an auxiliary symbol, say, $\mathbf{x}$. The string $abaa\mathbf{x}bc\mathbf{x}ddcd$

satisfies the criteria of Theorem 4.6(i) and 4.6(ii) and is therefore nonlinear. In the same way that semibinders could be daisy chained in our one- and two-trap encodings, any number of auxiliary symbols can be daisy chained. For example, using auxiliary symbols $\mathbf{x}$ and $\mathbf{y}$ we can obtain the nonlinear sequence $abaa\mathbf{x}b\mathbf{yx}c\mathbf{y}ddcd$. It seems as though auxiliary symbols are only useful for encodings where the ♀'s precede the ♂'s.

5. Conclusion and open problems.

Let us briefly summarize what is known about linear and nonlinear forbidden subsequences. There is a large class Φ_{KV} of linear forbidden subsequences [20] though the containment $\Phi_{KV} \subseteq \Phi_{Lin}$ is not known to be strict. The minimal set of nonlinear sequences Φ_{Nonlin} must contain at least three elements (Theorem 3.9); however, the only specific sequence known to be in Φ_{Nonlin} is *ababa*. Our set Ψ forms an infinite antichain of nonlinear forbidden subsequences and, together with *ababa*, forms a possible candidate for the set Φ_{Nonlin}.

The extremal functions for the nonlinear sequences in Ψ are all $\Omega(n\alpha(n))$. It is worthwhile distinguishing Φ_{Nonlin} from the set of minimal nonlinear sequences Φ_α with growth rate $\Omega(n\alpha(n))$. Given what we know about $\text{Ex}(\mu, n)$ for specific μ and the *near* total absence [21] of natural nonlinear functions $o(n\alpha(n))$, we are compelled to make the following conjecture.

CONJECTURE 5.1. $\Phi_{Nonlin} = \Phi_\alpha$.

A good way to prove that Φ_{Nonlin} is infinite is to start proving that $\text{Ex}(\mu, n) = O(n)$ for some μ that are contained in an infinite number of members in Ψ. Consider those sequences $\Psi(1)$ whose codes lie in $\heartsuit\spadesuit\heartsuit\heartsuit^*(\diamondsuit\varheartsuit\clubsuit)$, i.e., Definition 3.3(1), excluding $\heartsuit\spadesuit(\diamondsuit\varheartsuit\clubsuit)$. The first few elements of $\Psi(1)$ are *abcadcddbd*, *abcadcedeebe*, and *abcadcedfeffbf*. According to our often misguided intuition, none of the *strict* subsequences of the members of $\Psi(1)$ are substantially more complex than *abcbccac*. For example, in *bcdcedeebe*($\precneq$ *abcadcedeebe*) the c does not seem to add much to the complexity of the sequence. It "sandwiches" the first occurrence of d but does not mingle with other symbols. It is quite plausible that in this situation, splicing c out of the forbidden subsequence would not affect its complexity. It may be possible to prove Conjecture 5.2 by extending the techniques of Klazar and Valtr [20].

CONJECTURE 5.2. *If a and b do not appear in σ_1 and a does not appear in σ_2, then* $\text{Ex}(\sigma_1 aba\sigma_2, n) = O(n + \text{Ex}(\sigma_1 b\sigma_2, n))$.

One immediate consequence of Conjecture 5.2 is that $\text{Ex}(abcbccac, n) = O(n)$. Furthermore, it would imply the infinitude of Φ_{Nonlin} since no nonlinear forbidden subsequence could be contained in an infinite number of the $\Psi(1)$ sequences.

Appendix A. One-trap lower bounds.

In this section we prove that all forbidden subsequences that could be generated from a one-trap legal compact encoding are nonlinear.

THEOREM 3.4. *For all $\sigma \in \Psi_1$, $\text{Ex}(\sigma, n) = \Omega(n\alpha(n))$.*

Proof. Let λ be the legal compact encoding that generated σ. For simplicity we omit those encodings with parenthesized expressions, e.g., $\heartsuit\spadesuit(\diamondsuit\varheartsuit)\clubsuit$. Let $j^\clubsuit, j^\spadesuit, j^\varheartsuit$, and $j^\diamondsuit$ be the indices of the trap, two captives, and guard. Assuming that $\sigma \prec \mathscr{S}$ let $v_{j,k}$ be the node in T corresponding to the kth occurrence of j in σ. The argument from Theorems 3.1 and 3.2 used to obtain a contradiction will go through for *any* σ, provided we can establish the following ancestor-descendant relationships:

$$\text{(A.1)} \qquad \{v_{j^\spadesuit,1}, v_{j^\varheartsuit,1}\} \lhd v_{j^\clubsuit,2} \qquad \textit{the binder must bind,}$$

$$\text{(A.2)} \qquad v_{j^\clubsuit,2} \lhd \{v_{j^\spadesuit,2}, v_{j^\varheartsuit,2}\} \qquad \textit{the guard must guard,}$$

$$\text{(A.3)} \qquad \{v_{j^\spadesuit,2}, v_{j^\varheartsuit,2}\} \unlhd v_{j^\clubsuit,3} \qquad \textit{the trap must trap.}$$

Given (A.1), line (A.2) follows from the ordering $j^{\diamondsuit} < j^{\rotatebox{180}{$\spadesuit$}}$ (by the fact that $j^{\diamondsuit}j^{\rotatebox{180}{$\spadesuit$}}j^{\diamondsuit}j^{\rotatebox{180}{$\spadesuit$}} \overline{\prec} \sigma$ and Lemma 2.7) and that vertex labels are in descending order. Given (A.1), line (A.3) follows trivially from the construction of $\mathscr{S}$. Thus, our only task is to show (A.1): that the set of binders does, in fact, bind.

Let $\{j_k^{\heartsuit}\}_{1\le k\le B}$ be the set of B binders. It follows from the steps 2 and 3 of the construction of τ_λ (see section 3.1.2) that the subsequence of σ restricted to the binders and trap is

$$\sigma^{\heartsuit} = j_1^{\heartsuit}j_2^{\heartsuit}j_1^{\heartsuit}j_3^{\heartsuit}j_2^{\heartsuit}j_4^{\heartsuit}\cdots j_B^{\heartsuit}j_{B-1}^{\heartsuit}j^{\clubsuit}j_B^{\heartsuit}j^{\clubsuit}j^{\clubsuit}.$$

Obviously $v_{j_1^{\heartsuit},2}$ is ancestral to $v_{j_1^{\heartsuit},1}$ and $v_{j_2^{\heartsuit},1}$. Assume inductively that $v_{j_i^{\heartsuit},2}$ is ancestral to $v_{j_1^{\heartsuit},1}, v_{j_2^{\heartsuit},1}, \ldots v_{j_{i+1}^{\heartsuit},1}$, where $j^{\clubsuit}$ goes by the pseudonym $j_{B+1}^{\heartsuit}$. The next two symbols in $\sigma^{\heartsuit}$ to appear are $j_{i+2}^{\heartsuit}$ (for the first time) and then $j_{i+1}^{\heartsuit}$ (for the second time). It follows from the postordering of the vertex labels that $v_{j_{i+1}^{\heartsuit},2}$ is ancestral to $v_{j_i^{\heartsuit},2}$ and $v_{j_{i+2}^{\heartsuit},1}$. Thus, when $i = B$ we have that $v_{j_B^{\heartsuit},2}$ is ancestral to $v_{j_1^{\heartsuit},1}, \ldots v_{j_B^{\heartsuit},1}, v_{j^{\clubsuit},1}$. To establish (A.1) we need to show that $v_{j_B^{\heartsuit},2}$ is also ancestral to *both* $v_{j^{\spadesuit},1}$ and $v_{j^{\rotatebox{180}{$\spadesuit$}},1}$. By Definition 3.3(iii)–3.3(v), both $j^{\spadesuit}$ and $j^{\rotatebox{180}{$\spadesuit$}}$ appear between $j_1^{\heartsuit}$ and $j_{B+1}^{\heartsuit} = j^{\clubsuit}$. From the postordering of the vertices, it follows that if $j = j^{\spadesuit}$ or $j^{\rotatebox{180}{$\spadesuit$}}$ and j is sandwiched between $j_i^{\heartsuit}$ and $j_{i+1}^{\heartsuit}$, $v_{j,1}$ must be a descendant of $v_{j_i^{\heartsuit},2}$. □

Appendix B. Two-trap lower bounds.

THEOREM 3.8. *For all $\sigma \in \Psi_2$, $\mathrm{Ex}(\sigma, n) = \Omega(n\alpha(n))$.*

Proof. Let λ be the legal compact encoding that generated σ. For simplicity we omit those encodings with parenthesized expressions, e.g., $\heartsuit(\rotatebox{180}{$\heartsuit$}\rotatebox{180}{$\clubsuit$})\spadesuit\heartsuit\clubsuit$. Let $j^{\clubsuit}, j^{\rotatebox{180}{$\clubsuit$}}, j^{\spadesuit}$ be the indices of the outer trap, inner trap, and captive. Assuming that $\sigma \prec \mathscr{S}$ let $v_{j,k}$ be the node in T corresponding to the kth occurrence of j in σ. The argument from Theorems 3.5 and 3.6 used to obtain a contradiction will go through for *any* σ, provided we can establish the following ancestor-descendant relationships:

$$v_{j^{\spadesuit},1} \trianglelefteq v_{j^{\rotatebox{180}{$\clubsuit$}},2} \qquad \textit{the inner binder must bind,} \tag{B.1}$$

$$v_{j^{\rotatebox{180}{$\clubsuit$}},2} \lhd v_{j^{\clubsuit},2} \qquad \textit{the outer binder must bind,} \tag{B.2}$$

$$v_{j^{\clubsuit},2} \trianglelefteq \{v_{j^{\spadesuit},2}, v_{j^{\rotatebox{180}{$\clubsuit$}},3}\} \trianglelefteq v_{j^{\clubsuit},3} \qquad \textit{the outer trap must trap.} \tag{B.3}$$

Given (B.1) and (B.2), line (B.3) follows trivially from the postordering of the vertex labels. Thus, our two tasks are to show that the inner binders and outer binders do, in fact, bind as promised.

Let $\{j_k^{\heartsuit}\}_{1\le k\le B}$ be the set of B outer binders and $\{j_k^{\rotatebox{180}{$\heartsuit$}}\}_{1\le k\le B'}$ be the B' inner binders. The restriction of σ to the symbols $\rotatebox{180}{$\heartsuit$}$ and $\rotatebox{180}{$\clubsuit$}$ is *precisely* the same as that found in the proof of Theorem 3.4. Similarly, the restriction of σ to the symbols $\heartsuit$ and $\clubsuit$ will also take precisely this form. In other words, the exact same argument shows that $v_{j^{\spadesuit},1}$ (where $j^{\spadesuit}$ must lie between $j_1^{\rotatebox{180}{$\heartsuit$}}$ and $j^{\rotatebox{180}{$\clubsuit$}}$, by Definition 3.7(viii)) will be a descendant of $v_{j^{\rotatebox{180}{$\clubsuit$}},2}$. Furthermore, it implies the descendant relationships $v_{j_1^{\heartsuit},2} \lhd v_{j_2^{\heartsuit},2} \lhd \cdots \lhd v_{j_B^{\heartsuit},2}$ and $v_{j^{\clubsuit},1} \trianglelefteq v_{j_B^{\heartsuit},2}$.

To show that the outer binder binds we need only to show that $v_{j^{\rotatebox{180}{$\clubsuit$}},2} \trianglelefteq v_{j_i^{\heartsuit},2}$ for *some* i. This will actually hold for $i = 1$. Consider the restriction of σ to the first two $\heartsuit$s and $\rotatebox{180}{$\clubsuit$}$, or if $B = 1$, the restriction to $\heartsuit$, $\rotatebox{180}{$\clubsuit$}$, and $\clubsuit$. It will be either

$$j_1^{\heartsuit}j^{\rotatebox{180}{$\clubsuit$}}j^{\rotatebox{180}{$\clubsuit$}}j_2^{\heartsuit}j_1^{\heartsuit} \qquad \text{or} \qquad j_1^{\heartsuit}j^{\rotatebox{180}{$\clubsuit$}}j^{\rotatebox{180}{$\clubsuit$}}j^{\clubsuit}j_1^{\heartsuit}.$$

In either case, the postordering of the vertex labels implies that $v_{j^{\rotatebox{180}{$\clubsuit$}},2} \trianglelefteq v_{j_1^{\heartsuit},2}$. □

REFERENCES

[1] R. Adamec, M. Klazar, and P. Valtr, *Generalized Davenport–Schinzel sequences with linear upper bound*, Discrete Math., 108 (1992), pp. 219–229.

[2] P. Agarwal and M. Sharir, *Davenport–Schinzel Sequences and Their Geometric Applications*, Cambridge University Press, Cambridge, 1995.

[3] P. Agarwal, M. Sharir, and P. Shor, *Sharp upper and lower bounds on the length of general Davenport-Schinzel sequences*, J. Combin. Theory Ser. A, 52 (1989), pp. 228–274.

[4] A. Aggarwal, M. Klawe, S. Moran, P. Shor, and R. Wilber, *Geometric applications of a matrix-searching algorithm*, Algorithmica, 2 (1987), pp. 195–208.

[5] N. Alon, H. Kaplan, G. Nivasch, M. Sharir, and S. Smorodinsky, *Weak ϵ-nets and interval chains*, J. ACM, 55 (2008).

[6] D. Bienstock and E. Györi, *An extremal problem on sparse $0-1$ matrices*, SIAM J. Discrete Math., 4 (1991), pp. 17–27.

[7] R. E. Burkard, B. Klinz, and R. Rudolf, *Perspectives of Monge properties in optimization*, Discrete Appl. Math., 70 (1996), pp. 95–161.

[8] H. Davenport and A. Schinzel, *A combinatorial problem connected with differential equations*, Amer. J. Math., 87 (1965), pp. 684–694.

[9] A. Efrat and M. Sharir, *A near-linear algorithm for the planar segment center problem*, in Proceedings of the 5th ACM-SIAM Symposium on Discrete Algorithms, 1994, pp. 87–97.

[10] Z. Füredi, *The maximum number of unit distances in a convex n-gon*, J. Combin. Theory Ser. A, 55 (1990), pp. 316–320.

[11] Z. Füredi and P. Hajnal, *Davenport-Schinzel theory of matrices*, Discrete Math., 103 (1992), pp. 233–251.

[12] J. T. Geneson, *Extremal functions of forbidden double permutation matrices*, J. Combin. Theory Ser. A, 116 (2009), pp. 1235–1244.

[13] S. Hart and M. Sharir, *Nonlinearity of Davenport–Schinzel sequences and of generalized path compression schemes*, Combinatorica, 6 (1986), pp. 151–177.

[14] B. Keszegh, *On linear forbidden submatrices*, J. Combin. Theory Ser. A, 116 (2009), pp. 232–241.

[15] M. M. Klawe, *Superlinear bounds for matrix searching problems*, J. Algorithms, 13 (1992), pp. 55–78.

[16] M. M. Klawe and D. J. Kleitman, *An almost linear time algorithm for generalized matrix searching*, SIAM J. Discrete Math., 3 (1990), pp. 81–97.

[17] M. Klazar, *Two results on a partial ordering of finite sequences*, Comment. Math. Univ. Carolin., 34 (1993), pp. 697–705.

[18] M. Klazar, *The Füredi–Hajnal conjecture implies the Stanley–Wilf conjecture*, in Formal Power Series and Algebraic Combinatorics, Springer, Berlin, 2000, pp. 250–255.

[19] M. Klazar, *Generalized Davenport–Schinzel sequences: Results, problems, and applications*, Integers, 2 (2002), A11.

[20] M. Klazar and P. Valtr, *Generalized Davenport-Schinzel sequences*, Combinatorica, 14 (1994), pp. 463–476.

[21] M. Loebl, *Unprovable combinatorial statements*, Discrete Math., 108 (1992), pp. 333–342.

[22] A. Marcus and G. Tardos, *Excluded permutation matrices and the Stanley–Wilf conjecture*, J. Combin. Theory Ser. A, 107 (2004), pp. 153–160.

[23] G. Nivasch, *personal communication*, 2008.

[24] G. Nivasch, *Improved bounds and new techniques for Davenport–Schinzel sequences and their generalizations*, J. ACM, 57 (2010).

[25] J. Pach and M. Sharir, *On vertical visibility in arrangements of segments and the queue size in the Bentley–Ottmann line sweeping algorithm*, SIAM J. Comput., 20 (1991), pp. 460–470.

[26] J. Pach and G. Tardos, *Forbidden paths and cycles in ordered graphs and matrices*, Israel J. Math., 155 (2006), pp. 359–380.

[27] S. Pettie, *Splay trees, Davenport–Schinzel sequences, and the deque conjecture*, in Proceedings of the 19th ACM-SIAM Symposium on Discrete Algorithms, 2008, pp. 1115–1124.

[28] S. Pettie, *Applications of forbidden $0-1$ matrices to search tree- and path compression-based data structures*, in Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms, 2010, pp. 1457–1467.

[29] S. Pettie, *On nonlinear forbidden $0-1$ matrices: A refutation of a Füredi–Hajnal conjecture*, in Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms, 2010, pp. 875–885.

[30] G. Tardos, *On* 0-1 *matrices and small excluded submatrices*, J. Combin. Theory Ser. A, 111 (2005), pp. 266–288.

[31] R. E. Tarjan, *Efficiency of a good but not linear set merging algorithm*, J. ACM, 22 (1975), pp. 215–225.

[32] P. Valtr, *Graph drawings with no k pairwise crossing edges*, in Proceedings of the 5th International Symposium on Graph Drawing, 1997, pp. 205–218.

[33] P. Valtr, *Davenport-Schinzel trees*, Combinatoria, 23 (2003), pp. 151–184.

[34] A. Wiernik and M. Sharir, *Planar realizations of nonlinear Davenport–Schinzel sequences by segments*, Discrete Comput. Geom., 3 (1988), pp. 15–47.