Nonlinearity of Davenport-Schinzel Sequences and of a Generalized Path Compression Scheme

Sergiu Hart and Micha Sharir
School of Mathematical Sciences
Tel Aviv University
Ramat Aviv, Tel Aviv 69978, Israel

ABSTRACT

Davenport-Schinzel sequences are sequences that do not contain forbidden subsequences of alternating symbols. They arise in the computation of the envelope of a set of functions. We show that the maximal length of a Davenport-Schinzel sequence composed of \( n \) symbols is \( O(n \alpha(n)) \), where \( \alpha(n) \) is the functional inverse of Ackermann's function, and is thus very slow growing. This is achieved by establishing an equivalence between such sequences and generalized path compression schemes on rooted trees, and then by analyzing these schemes.

The estimation of \( \lambda_s(n) \) is the subject of this paper.

This problem has originally been posed by Davenport and Schinzel [DS]. Their interest in it arose from its connection to the analysis of solutions of linear differential equations. Recently, Atallah [At] has raised it again independently, because of its significance for problems in dynamic computational geometry. These two applications are quite similar, and can be briefly described as follows. Let \( f_1, \ldots, f_n \) be \( n \) real-valued continuous functions defined on a common interval \( I \). Suppose that for each \( i \neq j \) the functions \( f_i \) and \( f_j \) intersect in at most \( s \) points (e.g., this is the case for polynomials of fixed degree, or Chebycheff systems, and so on). Let \( g(z) = \min \{ f_i(z) : i = 1, \ldots, n \} \) for \( z \in I \), be the pointwise minimum of the \( f_i \)'s, and let \( m \) be the smallest number of subintervals \( I_1, \ldots, I_m \) of \( I \) such that for each \( k \) there exists an index \( i_k \) with \( g(x) = f_{i_k}(x) \) for all \( x \in I_k \). In other words, \( m \) is the number of connected portions of the graphs of the \( f_i \)'s which constitute the graph of \( g \). Assuming that \( I_1, \ldots, I_m \) are arranged in this order from left to right, put

\[
U(f_1, \ldots, f_n) = (i_1, \ldots, i_m).
\]

It is now easily seen that \( U(f_1, \ldots, f_n) \) is a \( DS(n,s) \) sequence. Moreover, it is known (see [At]) that for any \( DS(n,s) \) sequence \( U \) one can construct a collection \( f_1, \ldots, f_n \) of such functions for which \( U(f_1, \ldots, f_n) = U \). Therefore the largest possible value of \( m \) is precisely \( \lambda_s(n) \).

Thus, in this setting, Davenport-Schinzel sequences are strongly related to the problem of computing the (lower) envelope of a set of functions which intersect each other in pairs in at most some fixed number of points. This problem has many applications in computational geometry and related areas, many of which are

1. Introduction

Consider the following combinatorial problem: Let \( n, s \) be positive integers. A sequence \( U = (u_1, \ldots, u_m) \) of integers is an \((n,s)\) Davenport-Schinzel sequence (a \( DS(n,s) \) sequence for short), if it satisfies the following conditions:

(i) \( 1 \leq u_i \leq n \) for each \( i \).

(ii) For each \( i < m \) we have \( u_i \neq u_{i+1} \).

(iii) There do not exist \( s+2 \) indices \( 1 \leq i \leq i_2 \leq \cdots \leq i_{s+2} \leq m \) such that

\[
\begin{align*}
& u_{i_1} = u_{i_3} = u_{i_6} = \cdots = a, \\
& u_{i_2} = u_{i_4} = u_{i_5} = \cdots = b, \quad \text{and} \quad a \neq b.
\end{align*}
\]

We will write \( |U| = m \) for the length of the sequence \( U \).

Define

\[
\lambda_s(n) = \max \{ |U| : U \text{ is a } DS(n,s) \text{ sequence} \}.
\]

Work on this paper by the second author has been supported in part by a grant from the U.S.-Israeli Binational Science Foundation.
given in [At]; some additional applications will be noted in Section 7.

The problem of estimating \( \lambda_s(n) \) has been studied in several papers: [DS], [Da], [RS], [Sz], [At]. It is known (and easy to prove) that \( \lambda_1(n) = n \) and \( \lambda_2(n) = 2n - 1 \). For \( s \geq 3 \) the deep result of Szemerédi [Sz] states that \( \lambda_s(n) = O(n \log^s n) \), where \( (j) \) the constant of proportionality depends on \( s \), and where \( \log^s n \)

\( \) is the smallest \( t \geq 1 \) such that \( \epsilon_t \geq n \), where \( \epsilon_1 = 2 \) and \( \epsilon_{t+1} = 2^{\epsilon_t} \).

However, it was not known whether \( \lambda_s(n) \), for \( s \geq 3 \), is actually nonlinear in \( n \), i.e., whether Davenport-Schinzel sequences of nonlinear size actually exist.

In this paper we show that \( \lambda_3(n) \) is indeed nonlinear, and also improve the upper bound of Szemerédi for this function. Specifically we show that \( \lambda_3(n) = \Theta(n \alpha(n)) \), where \( \alpha(n) \) is the functional inverse of Ackermann’s function; the function \( \alpha(n) \) is very slowly growing, but tends to infinity with \( n \). The proof is based on an interesting equivalence between Davenport-Schinzel sequences with \( s = 3 \) and sequences of certain operations performed on arbitrary rooted trees, called generalized path compressions.

The paper is organized as follows. Section 2 introduces the notion of generalized path compressions and compression schemes on trees. Section 3 reviews the basic properties of Ackermann’s function, and states the main theorems of this paper. Section 4 establishes a linear equivalence between \( DS(n, 3) \) sequences and path compression schemes, so that the problem can be reduced to that of estimating the maximal length of such schemes. Sections 5 and 6 contain the derivations of the upper and lower bounds, respectively. Concluding remarks, including some applications of these results and some open problems, are given in the final Section 7.

2. Generalized Path Compressions on Trees.

Let \( T \) be an arbitrary rooted tree. It is given by a triple \((V, r, \varphi)\), where \( V \) is a finite set of vertices, \( r \in V \) is the root of \( T \), and \( \varphi : V \to r \to V \) is the fatherhood mapping: for each \( x \in V \), \( x \neq r \), \( \varphi(x) \) is the father of \( x \), and \( x \) is a son of \( \varphi(x) \). The mapping \( \varphi \) has no cycles, so that by repeated applications of \( \varphi \), every \( x \in V \) is eventually mapped onto \( r \). We will use the following standard notations for \( x, y \in V \): \( x \) is a descendant of \( y \) (and \( y \) is an ancestor of \( x \)) if there are \( n \geq 1 \) vertices \( x_1, x_2, \ldots, x_n \in V \), such that \( x_1 = x \), \( x_n = y \), and \( \varphi(x_i) = x_{i+1} \) for \( i = 1, \ldots, n-1 \); \( x \) is a proper descendant of \( y \) if \( n > 1 \) (thus \( x \neq y \)).

We define an operation on \( T \), called generalized path compression (GPC for short) as follows. Let \( x_1, x_2, \ldots, x_k \) be a sequence of nodes of \( T \) such that each \( e_i \) is a proper descendant of \( x_{i+1} \), for \( i = 1, \ldots, k-1 \). The generalized path compression \( f = (x_1, x_2, \ldots, x_k) \) is an operation that modifies \( T \) so as to make each \( x_i \), for \( i = 1, \ldots, k-1 \), a son of \( x_k \). More precisely, \( f \) results in making \( \varphi(x_i) = x_k \) for \( i = 1, \ldots, k-1 \), and leaving \( \varphi(x) \) unchanged for all other \( x \); see Fig. 2.1.

![Fig. 2.1. A generalized path compression](image)

This notion generalizes the notion of standard path compression, in that the nodes \( x_1, \ldots, x_k \) are not required to be adjacent along their present path. These standard compressions are used in the efficient implementation of the FIND operation in the set-union algorithm used for processing equivalence relationships (see [Ta] for an extensive analysis and earlier references).

For each GPC \( f = (x_1, \ldots, x_k) \) call \( x_1 \) the starting node of \( f \), and denote it by \( s(f) \); the length of \( f \) is \( |f| = k-1 \) (this is the number of edges \((x_1, x_2), \ldots, (x_{k-1}, x_k)\) in \( f \)).

Another notion we need is that of a postorder on \( T \). It is a linear order of the nodes of \( T \) obtained recursively as follows: Suppose the root \( r \) of \( T \) has \( l \) sons \( q_1, \ldots, q_l \). Then a postorder on \( T \) is obtained by concatenating postorders of the subtrees of \( T \) rooted at

\( (f) \) In this paper all logarithms are with base 2.
$g_1, \ldots, g_k$ and appending $r$ at the end. (Thus $T$ can have many postorders, depending on the order of the enumeration of the sons of each of its nodes.)

Finally, a sequence $F = (f_1, \ldots, f_m)$ of GPC’s on a rooted tree $T$ is admissible if

(i) it is executable, meaning that each $f_i$ is a GPC on the tree $T_i$ obtained from $T$ after the compressions $f_1, \ldots, f_{i-1}$ have been executed ($T_i \equiv T$);

(ii) it is postordered, meaning that there exists a postorder on $T$ such that the starting nodes $\sigma(f_1), \ldots, \sigma(f_m)$ are arranged in (weak) postorder (thus more than one GPC may start at the same node; however, if $\sigma(f_i) \neq \sigma(f_{i+1})$ then $\sigma(f_{i+1})$ succeeds $\sigma(f_i)$ in the postorder).

The length $|F|$ of such a sequence is defined as $\sum_{i=1}^{m} |f_i|$. We are concerned here with the maximal possible value of $|F|$, or, more precisely, with the quantity

$$\chi(n, m) = \max \{|F| : F \text{ is an admissible sequence of GPC's on a tree with } n \text{ vertices} \}.$$ (Note that $\chi(n, m)$ is well defined, e.g. $\chi(n, m) \leq (n-1)m$, since $|f| \leq n-1$ for any GPC $f$ on $T$.)

In our estimation of $\chi$ it is useful to make the following simplifications: Let a tree $T$ with $n$ nodes and an admissible sequence $F$ of $m$ GPC’s on $T$ be given. For each $f = (x_1, \ldots, x_k) \in F$, we add to $T$ a new leaf (i.e. a vertex with no sons) $x_0$, make it a son of $x_1$, and change $f$ to $(x_0, x_1, \ldots, x_k)$. Furthermore, we extend the postorder on $T$ so that all new leaves of some node $x$ succeed all its original sons, and are arranged according to the order of the GPC’s they correspond to. Since we want to maximize $|F|$, we can assume without loss of generality that at least one GPC starts at each of the $n$ original nodes of $T$ (add trivial GPC’s of the form $(x, r)$ otherwise, with $r$ the root of $T$). In particular, each original leaf will now have at least one (new) leaf.

All these modifications make $T$ into a tree with $n+m$ vertices, exactly $m$ of which are leaves, such that each GPC in $F$ starts at a different leaf. (We note that the length of each GPC is increased by 1, thus $|F|$ increases by $m$.) We will refer to such a $T$ as an $(n, m)$-tree, and to the corresponding modified $F$ as a compression scheme on $T$. In what follows we will only consider $(n, m)$-trees $T$ and corresponding compression schemes $F$ (with exactly one GPC in $F$ starting at every leaf of $T$), and will aim at the estimation of the associated quantity

$$\psi(n, m) = \max \{|F| : F \text{ is a compression scheme on an } (n, m)-\text{tree} \}.$$ Note that $\psi(n, m) = \chi(n, m) + m$ for all $n, m$.

It is known (see [Fi], [Ta]) that if no restrictions on the order of the starting nodes of the GPC’s or on the structure of $T$ are imposed, then (even for standard path compressions) $\psi(n, n) = \Theta(n \log n)$ (for easier comparison, we state all results in the case $m = n$; usually $m = \Theta(n)$, which of course yields the same bounds). If the tree $T$ is required to be balanced and one uses only standard path compressions, then $\psi(n, n) = \Theta(n a(n))$ (cf. [Ta]). Here we show that, using generalized path compressions, and imposing no restrictions on the structure of $T$, but requiring that the GPC’s be postordered, $\psi(n, n)$ is $\Theta(n a(n))$. Although the same bounds are obtained both here and in [Ta], there is no obvious relation between the two problems (in particular, we do not prove our results by reducing the problem to that studied in [Ta], although we adapt some of the techniques in [Ta] to the derivation of our lower bounds; see also Section 6).

The equivalence between $\psi$ and $\lambda_3$, to be established in the Section 4, yields similar upper and lower bounds for $\lambda_3$.

3. Statement of Main Results

In this section we state our main results, concerning the functions $\lambda_3$ and $\psi$. For this purpose, we recall first the definition of Ackermann’s function ("generalized exponentials" - cf. [Ac]).

Let $N$ be the set of positive integers $1, 2, \ldots$. Given a function $g$ from $N$ into itself, denote by $g^{(s)}$ the composition $g \circ g \circ \cdots \circ g$ of $g$ with itself $s$ times, for $s \in N$. Define inductively a sequence $\{A_k\}_{k=1}^{\infty}$ of functions from $N$ into itself as follows:

$$A_1(n) = 2n,$$

$$A_k(n) = A_k(n_1)(1), \quad k \geq 2,$$

for all $n \in N$. Note that for all $k \geq 2$,

$$A_k(1) = 2,$$

$$A_k(n) = A_{k-1}(A_k(n-1)), \quad n \geq 2.$$ In particular, $A_2(n) = 2A_2(n-1)$, thus $A_2$ is the “power function”

$$A_2(n) = 2^n, \quad n \in N.$$ Also, $A_3(n) = 2A_3(n-1)$, thus $A_3$ is the “tower function”
\[ A_3(n) = 2^{2^{n-2}}, \]

with \( n \) 2's, for \( n \in \mathbb{N} \). Finally, put

\[ A(n) = A_n(n). \]

This is Ackermann’s function (actually, there are several variants of this function; their order of magnitudes are essentially the same, and our results do not depend on which one we use). The first values of Ackermann’s function are: \( A(1) = 2, A(2) = 4, A(3) = 16 \) and \( A(4) = 65536 \) 2’s. For basic properties of the functions defined above, the reader is referred to [Ta, p. 219] (note that his index \( k \) is one less than ours).

Given a strictly increasing function \( g \) from \( \mathbb{N} \) into itself, its functional inverse is the function \( \gamma \) from \( \mathbb{N} \) into itself given by

\[ \gamma(n) = \min \{ s \geq 1 : g(s) \geq n \}; \]

thus, \( \gamma(n) = s \) if and only if \( g(s - 1) < n \leq g(s) \).

In particular, let \( \alpha_k \) and \( \alpha \) denote the functional inverses of \( A_k \) and \( A \), respectively. Then, for all \( n \in \mathbb{N} \),

\[ \alpha_1(n) = \left\lfloor \frac{n}{2} \right\rfloor, \]
\[ \alpha_2(n) = \lceil \log n \rceil. \]

The functions \( \alpha_k \) are easily seen to satisfy the following recursive formula:

\[ \alpha_k(n) = \min \{ s \geq 1 : \alpha_{k-1}(s) = 1 \}; \]

that is, \( \alpha_k(n) \) is the number of iterations of \( \alpha_{k-1} \) needed to go from \( n \) to 1. In particular, \( \alpha_3(n) \) is precisely \( \log^* n \), as defined in Section 2.

All the functions \( \alpha_k \) are non-decreasing, and converge to infinity with their argument. The same holds for \( \alpha \), which also grows more slowly than any of the \( \alpha_k \). Note that \( \alpha(n) \leq 4 \) for all \( n \leq 4 \) which is a tower with 65536 2's, thus \( \alpha(n) \leq 4 \) for all practical purposes.

We can now state our results.

**MAIN THEOREM:**

\[ \lambda_3(n) = \Theta(n \alpha(n)). \]

Thus, there exists constants \( C_1, C_2 > 0 \) such that

\[ C_1 n \alpha(n) \leq \lambda_3(n) \leq C_2 n \alpha(n) \]

for all \( n \geq 1 \); we remark that the constants are of reasonable magnitude (very coarse estimates can be easily computed from the inequalities at the end of Sections 5 and 6). Note moreover that, in comparison, Szemeredi’s result (for \( \lambda_2 \)) can be stated as \( \lambda_3(n) \leq n \alpha_3(n) \).

The Main Theorem is a consequence of the following Theorems, the first establishing connections between Davenport-Schinzel sequences and compression schemes, and the other two yielding (upper and lower) bounds for the latter.

**Theorem A:** For all \( n, m \geq 1 \),

\[ \lambda_3(n) \leq \psi(2n, n), \]

\[ \psi(n, m) \leq \lambda_3(m) + (n + m - 1). \]

**Theorem B:**

\[ \psi(n, m) = O((n + m) \alpha(n)). \]

**Theorem C:**

\[ \psi(2n, n) = \Omega(n \alpha(n)). \]

Theorems A, B and C will be proved in the following three sections.

**Proof of Main Theorem:** Theorem B (with \( (2n, n) \) for \( (n, m) \)) together with Theorem A(i) imply \( \lambda_3(n) = O(n \alpha(n)) \). The other inequality \( \lambda_3(n) = \Omega(n \alpha(n)) \) follows from Theorem C and Theorem A(ii) (with \( (2n, n) \) for \( (m, m) \)). Note that \( \alpha(2n) \leq \alpha(n) + 1 \) for all \( n \). Q.E.D.

**Remark:** The inverse of Ackermann’s function has also appeared in the analysis of the union-find algorithm (see [Ta]). The inverse function appearing there is, in our notations,

\[ \alpha^T(n) = \min \{ k \geq 1 : A_k(4) > \log n \}; \]

(cf. [Ta, p. 221], for \( m = n \)); recall that our function is

\[ \alpha(n) = \min \{ k \geq 1 : A_k(k) > n \}. \]

However, the two functions are of the same order of magnitude: indeed, it can be easily shown that, for all \( n \geq 1 \),

\[ \alpha(n) \leq \alpha^T(n) \leq 2 \alpha(n). \]

4. Linear equivalence between \( \psi \) and \( \lambda_3 \)

This section is devoted to the proof of Theorem A, showing that the two functions \( \psi \) and \( \lambda_3 \) are of the same order of magnitude. This will follow by using two transformations, from DS(\( n, 3 \)) sequences to compression schemes and vice versa.
4.1. Transforming Davenport-Schinzel sequences into compression schemes.

Let \( U = (u_1, u_2, \ldots, u_m) \) be a \( DS(n, 3) \) sequence. Define, for each \( i = 1, \ldots, n \),
\[
\mu_i = \min \{ b : u_b = i \};
\]
that is, \( \mu_i \) is the index of the first occurrence of \( i \) in \( U \). Without loss of generality (permuting \( 1, \ldots, n \) if necessary) assume that
\[
\mu_1 < \mu_2 < \cdots < \mu_n.
\]
A chain \( \langle u_{\mu}, u_{\mu+1}, \ldots, u_{\gamma} \rangle \) is a maximal decreasing contiguous subsequence of \( U \), i.e.
\[
(u_{\mu-1} < u_{\mu} > u_{\mu+1} > \cdots > u_\gamma \langle u_{\gamma+1} \rangle).
\]
Chains are obviously disjoint and their union is \( U \); moreover, it can be easily shown that there are at most \( 2n \) distinct chains.

Enumerate the chains of \( U \) in the order they occur as \( c_1, c_2, \ldots, c_p \), where \( p \leq 2n \). Let \( T \) be a \( (p, n) \)-tree, with the \( p \) inner nodes corresponding to the chains \( c_1, \ldots, c_p \); they are arranged in a single path, with \( \xi+1 \) the father of \( q \), for \( q = 1, \ldots, p-1 \); the \( n \) leaves \( L_1, \ldots, L_n \) correspond to the symbols 1, \ldots, \( n \) in the sequence \( U \), and are attached to the \( p \) inner nodes as follows. For each \( i \), let \( t_1 < t_2 < \cdots < t_q \) be the (indices of the) chains that contain \( i \); we then attach the leaf \( L_i \) to the node \( t_q \) in \( T \). Define a GPC \( f_i = (t_q, t_1, \ldots, t_q) \) on \( T \), and let \( F = (f_1, \ldots, f_p) \). The total length of \( F \) is
\[
|F| = \sum_{i=1}^{p} |f_i| = \sum_{i=1}^{p} q_i = \sum_{r=1}^{n} |c_r| = |U|.
\]
Finally, it can be shown

**Proposition:** The sequence \( F \) is a compression scheme on \( T \).

This transformation therefore yields \( \lambda_3(n) \leq \psi(p, n) \leq \psi(2n, n) \), proving Theorem A(i).

4.2. Transforming compression schemes into Davenport-Schinzel sequences.

Let \( T \) be an \( (n, m) \)-tree, and let \( F = (f_1, f_2, \ldots, f_m) \) be a compression scheme on \( T \). We will identify each \( f_j \) with the set of all vertices of \( T \) through which it passes, except for its last vertex. Enumerate the \( n+m \) nodes of \( T \) in the given postorder as \( 1, 2, \ldots, n+m \), and define, for each \( \nu = 1, \ldots, n+m \), a sequence
\[
U_\nu = \{ j : 1 \leq j \leq m \text{ and } \nu \in f_j \}.
\]
where the elements of each sequence \( U_\nu \) are arranged in decreasing order. Let \( U \) be the concatenation
\[
U = U_1 \parallel U_2 \parallel \cdots \parallel U_{n+m}.
\]
To obtain a Davenport-Schinzel sequence \( V \) from \( U \), we proceed through the subsequences \( U_\nu \) in order, erasing the first element of \( U_\nu \) whenever it equals the preceding non-erased element of \( U \); in total, at most \( n+m-1 \) elements are erased. The resulting sequence \( V \) has the following properties (the proof of (4) is quite complex, and is omitted from this version).

1. \( V \) is composed of \( m \) symbols.
2. The length of \( V \) is at least \( |F| - (n+m-1) \).
3. No two consecutive elements of \( V \) are equal.
4. \( V \) does not contain a subsequence of the form \( 1 \overline{j} \overline{i} \).

This completes the proof of Theorem A(ii).

5. The upper bounds

This section is devoted to the proof of Theorem B, establishing an upper bound on \( \psi \) (and hence also on \( \lambda_3 \)). To obtain this we derive a recurrence relation for \( \psi \).

**Proposition:** Let \( m, n \geq 1 \), and let \( b > 1 \) be a divisor of \( n \). Then there exist integers \( m^*, m_1, m_2, \ldots, m_b \geq 0 \) such that
\[
m^* + \sum_{i=1}^{b} m_i = m,
\]
and
\[
\psi(n, m) \leq \psi(b-1, m^*) + 2n + 2m^* + (*) + \sum_{i=1}^{b} \psi\left(\frac{n}{b^i}, m_i\right).
\]

**Proof:** Let \( T \) be any \( (n, m) \)-tree. Partition the \( n \) inner nodes of \( T \) into \( b \) equal layers \( L_1, \ldots, L_b \) (following the given postorder), each having \( \frac{n}{b^i} \) inner nodes. Let \( F \) be any compression scheme of \( m \) GPC's starting at the leaves of \( T \). Classify the GPC's of \( F \) into the following two types:

(I) GPC's that start and end in the same layer. For each \( i = 1, \ldots, \), let \( m_i \) denote the number of GPC's that start and end in layer \( L_i \). The total length of these GPC's is at most
\[
\sum_{i=1}^{b} \psi\left(\frac{n}{b^i}, m_i\right).
\]

(II) GPC's that contain vertices of more than one layer. Let \( m^* \) denote the number of GPC's
of this type; plainly, \( m^* + \sum_{i=1}^{b} m_i = m \). We then bound from above the total length of these GPC's, yielding in total (*) (see the full version of the paper for the very complex arguments involved).

**Corollary:** For all \( n, m, k \geq 1 \),
\[
\psi(n, m) \leq (4k - 4)n a_k(n) + (2k - 1)m,
\]
where \( a_k \) is the functional inverse of \( A_k \) as defined in Section 3.

To obtain this sequence of upper bounds on \( \psi \), stated above for \( k = 2, 3, \ldots \), we use (*) repeatedly. At each step we choose \( b \) in an appropriate manner, and estimate \( \psi(b-1, m^*) \) using the bound obtained in the preceding step. This yields a recurrence relation on \( \psi \) which we solve, to obtain a better upper bound on \( \psi \).

Finally, we complete the proof of Theorem B by choosing \( k \) to be of the order of \( a(n) \).

6. The lower bounds

In this section we will establish nonlinear lower bounds for \( \psi \) (and, a fortiori, also for \( \lambda_3 \)) which match closely the upper bounds just obtained. This is achieved using a construction similar to that of [Ta]. In this derivation we use a sequence of functions \( B_k \) which are similar to the functions \( A_k \), but which are easier to use in our construction.

Define inductively a sequence \( \{B_k\}_{k=1}^\infty \) of functions from the set \( N_0 = N_U(0) \) into itself as follows.

\[
B_1(s) = 0 \quad s \geq 0,
B_0(0) = 1 \quad k \geq 2,
B_k(s) = B_k(s-1) + B_{k-1}(sB_k(s-1)), \quad k \geq 2, s \geq 1.
\]

It can then be shown that

\[
A_{k-1}(s) \leq B_k(s) \leq A_k(s + 3), \quad k \geq 4, s \geq 1,
\]

thus the functions \( B_k \) and \( A_k \) are indeed of the same order of magnitude.

Let \( T \) be an arbitrary \((n, m)\)-tree. Define a sequence of trees \( T(i) \), for \( i \geq 0 \), as follows: \( T(0) = T \); to construct \( T(i+1) \), take two disjoint copies of \( T(i) \), introduce a new node \( r \) as the root of \( T(i+1) \), and make the roots of the two copies of \( T(i) \) sons of \( r \) (see figure 6.1). Note that \( T(i) \) is an \((n_i, m_i)\)-tree, where

\[
n_i = (n+1)2^i - 1,
m_i = m \cdot 2^i.
\]

Moreover, for any given postorder on \( T \), we obtain inductively an induced postorder on \( T(i+1) \) by taking in each of the two copies of \( T(i) \) the same (induced) postorder.

**Proposition:** Let \( T \) be a \((1, m)\)-tree with \( m \geq 1 \) and let \( k \geq 1 \). Then for each \( i \geq B_k(m) \) there exists a compression scheme \( F \) on \( T(i) \) such that each of the \( m_i \) GPC's in \( F \) is of length \( k \).

To prove this we use an argument similar to that in [Ta, Theorem 15], based on double induction on \( k \) and \( m \) (the lengthy details are omitted from this version).

**Corollary:** For all \( n, m \geq 1 \),
\[
\psi(n, m) \geq \frac{1}{8} \beta m,
\]
where
\[
\beta = \beta(n, m) = \max \{ k \geq 1 : A_k \left( \frac{2m}{n} + 3 \right) \leq 2n \},
\]
and \( \beta = 0 \) if there is no such \( k \).

This now easily implies Theorem C.

7. Concluding Remarks

7.1. Applications

Besides the numerous applications noted in [At], whose complexity bounds can now be given more concrete forms (using our estimate for \( s = 3 \), or Szemerédi's upper bound for \( s > 3 \)), we note here some additional applications.

(1) **Pointwise minima/maxima of intervals.** Let \( I_1, \ldots, I_n \) be \( n \) line segments in the plane, none of which is vertical. For each real \( x_0 \) let
\( J(x_0) \) be the line segment \( f_k \) whose intersection with the vertical line \( x = x_0 \) is lowest. Then the smallest number \( m \) of intervals on the \( x \)-axis, over each of which \( J \) is constant, is at most \( \lambda_d(n) \), thus \( O(n \alpha(n)) \). (It is not known whether \( \lambda_d(n) \) can actually be obtained in this setup).

(2) Pointwise minima/maxima of piecewise linear functions. Let \( f_1, \ldots, f_l \) be \( l \) continuous piecewise linear functions defined over a common interval \( I \), and let \( n \) be the total number of linear segments constituting the graphs of \( f_1, \ldots, f_l \). Then the maximal number of linear segments constituting the graph of the lower envelope of these functions is \( O(n \alpha(n)) \), and this envelope can be computed in time \( O(n \alpha(n) \log n) \). (This is an easy corollary of (1) above).

(3) Dynamic sorting. Consider the following "dynamic" sorting problem, in which we want to sort a sequence \( S \) of \( n \) numbers by repeated swaps of unsorted adjacent pairs of elements, but with the added difficulty that elements are being added and removed from \( S \) during our sorting process. More precisely, suppose that the elements to be sorted are the integers \( 1, \ldots, n \). Initially \( S \) is empty. The sorting process consists of a sequence of operations on \( S \), where each operation is one of the following:

(i) Insert a new element as the first element of \( S \) (each integer is inserted into \( S \) just once).
(ii) Delete the first element of \( S \).
(iii) Swap any two adjacent elements \( s_i, s_{i+1} \) of \( S \) for which \( s_i > s_{i+1} \).

The problem at hand is to estimate the maximal number \( C(n) \) of changes in the first place of \( S \). Clearly, if no insertions or deletions from \( S \) are applied (or rather, if we first insert all elements into \( S \), then swap them, and finally delete them all) then plainly \( C(n) = O(n) \). In the dynamic case however we have

Proposition: \( C(n) = \lambda_d(n) = \Theta(n \alpha(n)) \).

7.2. Open Problems

(1) For \( s > 0 \), we do not know if there is any simple connection between \( DS(n, s) \) sequences and tree operations like GPC's. An obvious open problem here is whether the upper bound of \( \lceil \frac{x}{2} \rceil \) on \( \lambda_d(n) \) can be improved for \( s > 3 \).

(2) We have already raised the question whether \( \lambda_d(n) \) can be attained for the pointwise minimum of \( n \) extended intervals. A related open problem is whether \( \lambda_d(n) \) can be attained for the minimum of \( n \) cubic polynomials (it is easy to check that both \( \lambda_1(n) \) and \( \lambda_d(n) \) are attained for the minimum of \( n \) linear (respec-

(3) We have obtained the bounds on \( \lambda_d(n) \) by transforming \( DS(n, 3) \) sequences into compression schemes of GPC's. However, generalized path compressions on trees may be interesting to study for their own sake. Various open problems arise in connection with GPC's, such as that of finding other restricted classes of sequences of GPC's for which almost linear upper bounds can be established. For example, suppose we allow only standard path compressions (i.e., path compressions whose nodes are adjacent to one another along their current path), and still require them to be executed in postorder. Is the maximal total length of such a sequence linear in the number of nodes of the corresponding tree?

(4) There is a similarity between the two problems studied here and in [Ta], in that they both involve path compressions on trees and they both attain similar upper and lower bounds. Is there some general problem of this kind, of which both our problem and that of [Ta] are special instances?

References


