NONLINEARITY OF DAVENPORT—SCHINZEL SEQUENCES AND OF GENERALIZED PATH COMPRESSION SCHEMES

S. HART and M. SHARIR

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 $\Theta(n\alpha(n))$, where $\alpha(n)$ is the functional inverse of Ackermann's function, and is thus very slowly increasing to infinity. This is achieved by establishing an equivalence between such sequences and generalized path compression schemes on rooted trees, and then by analyzing these schemes.

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 $1 \leq i_1 < i_2 < \ldots < i_{s+2} \leq m$ such that $u_{i_1} = u_{i_3} = u_{i_5} = \ldots = a$, $u_{i_2} = u_{i_4} = u_{i_6} = \ldots = b$, and $a \neq b$.

We will write $|U| = m$ for the *length* of the sequence U.

Define $\lambda_s(n) = \max\{|U|: U \text{ is a } DS(n, s) \text{ sequence}\}$. The estimation of $\lambda_s(n)$ is the subject of this paper.

This problem has originally been posed by Davenport and Schinzel [4]. Their interest in it arose from its connection to the analysis of solutions of linear differential equations. Recently, Atallah [2] 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 $g(x) = \min\{f_i(x): i = 1, \ldots, n\}$, for $x \in I$, be the *lower envelope* (i.e. 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 $I_1, ..., I_m$ are arranged in this order from left to right, put $U(f_1, ...,f_n)=$ $=(i_1, ..., i_m)$. It is now easily seen that $U(f_1, ...,f_n)$ is a $DS(n, s)$ sequence. Moreover, it is known (see [2]) that for any $DS(n, s)$ sequence U one can construct a collection $f_1, ...,f_n$ of such functions for which $U(f_1, ...,f_n)=U$. 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 given in [2]; some additional applications will be noted in Section 7.

The problem of estimating $\lambda_s(n)$ has been studied in several papers [2], [3], [4], [12], [15]. It is known (and easy to prove) that $\lambda_1(n)=n$ and $\lambda_2(n)=2n-1$. For $s\geq3$ the deep result of Szemerédi [15] states that $\lambda_s(n)=O(n\log^* n)$, where the constant of proportionality depends on s, and where $\log^* n$ is the smallest $i\geq1$ such that $\varepsilon_i\geq n$, where $\varepsilon_1=2$ and $\varepsilon_{i+1}=2^{\varepsilon_i}$. (We use here the following standard notations. Let f, g be two positive functions defined over the integers. Then $f(n)=$ $=O(g(n))$ (resp. $f(n)=\Omega(g(n))$) if there exists a constant $C>0$ such that $f(n)\leq$ $\leq Cg(n)$ (resp. $f(n)\geq Cg(n)$) for all n. We write $f(n)=\Theta(g(n))$ if both $f(n)=$ $=O(g(n))$ and $f(n)=\Omega(g(n))$ hold, namely, if there exist constants $C_1, C_2>0$ such that $C_1g(n)\leq f(n)\leq C_2g(n)$ for all n. All logarithms are with base 2.)

However, it was not known whether $\lambda_s(n)$, for $s\geq3$, is actually nonlinear in n, i.e. whether Davenport—Schinzel sequences of nonlinear size actually exist.

In this paper we show that $\lambda_s(n)$ is indeed nonlinear for $s\geq3$, and also improve the upper bound of Szemerédi for $s=3$. 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, and 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 definitions of Ackermann's function and its functional inverse, 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, some open problems and a discussion on Ackermann's functions, 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, φ), where V is a finite set of *vertices* (or, *nodes*), $r\in V$ is the *root* of T, and $\varphi: V-\{r\}\to V$ 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 φ has no cycles, so that by repeated applications of φ, every

$x \in V$ is eventually mapped into r. A vertex is called a *leaf* if it has no sons, and *inner* otherwise. 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 $x \neq y$), and *improper* otherwise (i.e. $x = y$).

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 x_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.
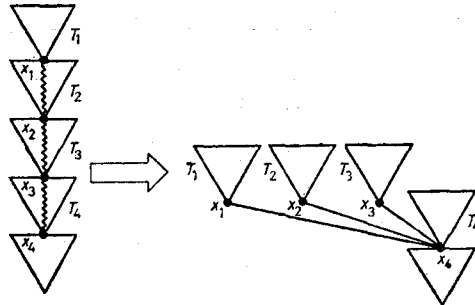

Fig. 2.1. A generalized path compression

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 [16] 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 $\sigma(f)$; 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 (in general, these are not edges of T).

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 $q_1, \ldots, q_l$ and appending r at the end. (Thus T can have many postorders,
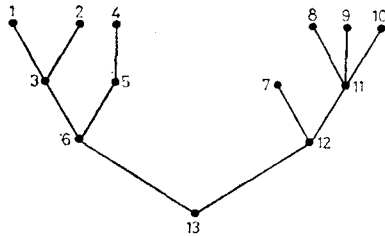

Fig. 2.2. A postordered tree

depending on the order of the enumeration of the sons of each of its nodes.) An example of a postordered tree is given in Fig. 2.2.

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 $(T_1 \equiv T)$;

(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

$$\chi(n, m) = \max \{|F|: F \text{ is an admissible sequence of } m \text{ GPC's on a tree } T \text{ with } n \text{ vertices}\}.$$

(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.)

For our estimation of χ 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 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 every node belongs to at least one GPC; otherwise, remove from the tree such a node (that does not appear at all in F), by connecting its sons (if any) directly to its father; add another node to the tree (so that it still has exactly n nodes) as a son of the starting point of the first GPC in F, and make it a leaf (the first in the postorder); finally, extend the first GPC so as to start from the new node; all these changes yield another tree with n nodes and an admissible sequence F' whose length is greater by 1 than that of F. In particular, each original leaf will now have at least one son (a new leaf).

All these modifications make T into a tree with $n+m$ vertices: n inner vertices and m leaves. Moreover, in the modified sequence F each of the m GPC's starts at a different leaf. We will refer to such a T as an (n, m)-*tree*, and to the corresponding F as a *compression scheme* on T. Thus, a compression scheme on an (n, m)-tree consists of m GPC's, exactly one GPC starting at every one of the m leaves; moreover, the leaves attached to any node succeed all its inner sons in postorder. In what follows we 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 in the transformation above the length of each GPC is increased by 1, thus $|F|$ increases by m. Therefore $\psi(n, m) = \chi(n, m) + m$ for all n, m.

It is known (see [5], [16]) 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 $\psi(n, n) = \Theta(n\alpha(n))$ (cf. [16]). 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 also $\Theta(n\alpha(n))$. Although the same bounds are obtained both here and in [16], there is no obvious relation between the two problems (in particular, our results are not proved by reducing the problem to that studied in [16], although we adapt some of the techniques in [16] to the derivation of our lower bounds; see also Section 6.2).

3. Statement of main results

In this section we state our main results, concerning the functions λ_3 and ψ. For this purpose, we recall first the definition of Ackermann's function ("generalized exponentials" — cf. [1]).

Let N be the set of positive integers $1, 2, \ldots$. Given a function g from a set into itself, denote by $g^{(s)}$ the composition $g \circ g \circ \ldots \circ g$ 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:

$$A_1(n) = 2n$$

$$A_k(n) = A_{k-1}^{(n)}(1), \quad k \geq 2$$

for all $n \in$ N. Note that for all $k \geq 2$, the function A_k satisfies

$$A_k(1) = 2,$$

$$A_k(n) = A_{k-1}(A_k(n-1)), \quad n \geq 2.$$

In particular, $A_2(n) = 2A_2(n-1)$, thus A_2 is the "power function"

$$A_2(n) = 2^n, \quad n \in \text{N}.$$

Then $A_3(n) = 2^{A_3(n-1)}$, thus A_3 is the "tower function"

$$A_3(n) = 2^{2^{\cdot^{\cdot^{\cdot^2}}}},$$

with n 2's in the exponential tower, for $n \in$ N. Finally, put $A(n) = A_n(n)$. This is *Ackermann's function* (actually, there are several variants of this function; their orders of magnitude are essentially the same, and our results do not depend on which one we use; see also Lemma 3.1 below). Ackermann's function A grows very fast; its first values are: $A(1) = 2$, $A(2) = 4$, $A(3) = 16$ and $A(4)$ is a tower of 65536 2's. For basic properties of the functions defined above, the reader is referred to [16, p. 219] (note that his index k is one less than ours).

Given a strictly increasing function g from N into itself, its *functional inverse* is the function γ from 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 α_k and α denote the functional inverses of A_k and A, respectively. Then, for all $n \in N$,

$$\alpha_1(n) = \left\lceil \frac{n}{2} \right\rceil,$$

$$\alpha_2(n) = \lceil \log n \rceil.$$

The functions α_k are easily seen to satisfy the following recursive formula:

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

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

All the functions α_k are non-decreasing, and converge to infinity with their argument. The same holds for α too, which grows more slowly than any of the α_k. Note that $\alpha(n) \leq 4$ for all $n \leq A(4)$ 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 exist 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 are given in the Remarks at the end of Sections 5 and 6). Note moreover that, in comparison, Szemerédi's result (for λ_3) can be stated as $\lambda_3(n) \leq Cn\alpha_3(n)$.

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* $n, m \geq 1$,

(i) $\lambda_3(n) \leq \psi(2n, n)$,

(ii) $\psi(n, m) \leq \lambda_3(m) + (n + m - 1)$.

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 (n, m)). (Note that $\alpha(2n) \leq \alpha(n) + 1$ for all n.) ∎

The inverse of Ackermann's function has also appeared in the analysis of the union-find algorithm in [16]. The inverse function appearing there is, in our notations

$$\alpha^T(n) = \min \{k \geq 1 : A_{k+1}(4) > \log n\}$$

(cf. [16, p. 221], for $m=n$); recall that our function is

$$\alpha(n) = \min\{k \geq 1: A_k(k) \geq n\}.$$

However, the two functions are of the same order of magnitude:

Lemma 3.1. *For all* $n \geq 1$,

$$\frac{1}{4}\alpha(n) \leq \alpha^T(n) \leq 2\alpha(n).$$

To show this, we use the following

Lemma 3.2. *For all* $k \geq 1$ *and* $s \geq 3$, $A_k(s+1) \leq A_{k+1}(s)$.

Proof. It is easily checked that $A_k(s) \leq A_{k+1}(s)$ for all k, $s \geq 1$. Then

$$A_{k+1}(s) = A_k(A_{k+1}(s-1)) \geq A_k(A_2(s-1)) = A_k(2^{s-1}) \geq A_k(s+1),$$

for all $s \geq 3$. ∎

Proof of Lemma 3.1. If $n \leq 16$, then $\alpha(n) \leq 3$ and $\alpha^T(n)=1$. If $n>16$, then $\alpha(n) \geq \geq 4$, and using Lemma 3.2 we obtain for $k=\alpha(n)$

$$\log n < n \leq A_k(k) \leq A_{2k-4}(4),$$

thus $\alpha^T(n) \leq 2\alpha(n)-5$.

For the converse, it is easily checked that if $\alpha^T(n)=1$ or 2, then $\alpha(n) \leq 4$; if $\alpha^T(n)=3$, then $\alpha(n) \leq 5$. And finally, if $k=\alpha^T(n) \geq 4$, then

$$n < 2^{A_{k+1}(4)} = A_2(A_{k+1}(4)) \leq A_k(A_{k+1}(4)) = A_{k+1}(5) \leq A_{k+1}(k+1),$$

thus $\alpha(n) \leq \alpha^T(n)+1$. ∎

4. Linear equivalence between ψ and λ_3

In this section we prove Theorem A, showing that the two functions ψ and λ_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 $U=(u_1, u_2, ..., u_m)$ be a $DS(n, 3)$ sequence. Define, for each $i=1, ..., n$,

$$\mu_i = \min\{\beta: u_\beta = i\},$$
$$v_i = \max\{\beta: u_\beta = i\};$$

that is, μ_i is the index of the first occurrence of i in U, and v_i is the index of the last such occurrence. Without loss of generality (permuting $1, ..., n$ if necessary) assume that $\mu_1 < \mu_2 < ... < \mu_n$.

A *chain* $c = (u_\beta, \ldots, u_\gamma)$ is a maximal decreasing contiguous subsequence of U, i.e.

$$(u_{\beta-1} <) u_\beta > u_{\beta+1} > \ldots > u_\gamma (< u_{\gamma+1}).$$

Chains are obviously disjoint and their union is U.

Lemma 4.1. *Suppose that* $i = u_{\beta-1} < u_\beta = j$. *Then either* $\beta - 1 = v_i$ *or* $\beta = \mu_j$.

Proof. Suppose the contrary; then $\gamma = \mu_j < \beta - 1$ and $\delta = v_i > \beta$. Also, since $i < j$ we must have $\varepsilon = \mu_i < \mu_j = \gamma$. Overall, we obtain five indices $\varepsilon < \gamma < \beta - 1 < \beta < \delta$ such that $u_\varepsilon = u_{\beta-1} = u_\delta = i$ and $u_\gamma = u_\beta = j$, a contradiction which completes the proof. ∎

Corollary 4.2. *There are at most* $2n$ *distinct chains in* U.

Proof. By the preceding lemma, a chain can end either at v_i or at $\mu_i - 1$, for some $i = 1, \ldots, n$, and there are at most $2n$ such places. ∎

Remark. The correct upper bound on the number of chains is actually $2n - 1$, since $\mu_1 - 1 = 0$; it is attained, for example, by the sequence (of length $5n - 8$)

$$1 \; 2 \; 1 \; 3 \; 1 \ldots 1 \; n-1 \; 1 \; n-1 \; n-2 \ldots 3 \; 2 \; n \; 2 \; n \; 3 \; n \ldots n \; n-1 \; n.$$

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 $1, \ldots, p$ corresponding to the chains $c_1, \ldots, c_p$; they are arranged in a single path, with $q + 1$ the father of q, for $q = 1, \ldots, p - 1$; the n leaves $l_1, \ldots, l_n$ correspond to the symbols $1, \ldots, n$ appearing in the sequence U, and are attached to the p inner nodes as follows. For each i, let $t_1 < t_2 < \ldots < t_{q_i}$ be the indices of the chains that contain i; we then attach the leaf l_i to the node t_1 in T. Define a GPC $f_i = (l_i, t_1, \ldots, t_{q_i})$ on T, and let $F = (f_1, \ldots, f_n)$. The total length of F is

$$|F| = \sum_{i=1}^n |f_i| = \sum_{i=1}^n q_i = \sum_{r=1}^p |c_r| = |U|.$$

Proposition 4.3. *The sequence* F *is a compression scheme on* T.

Proof. For each $i = 1, \ldots, n$, let s_i be the index of the chain containing the first occurrence of i in U (i.e. $\mu_i \in c_{s_i}$); since $\mu_1 < \mu_2 < \ldots < \mu_n$ and the chains are decreasing sequences, we have $s_1 < s_2 < \ldots < s_n$, and thus the leaves $l_i = \sigma(f_i)$, whose fathers are the nodes s_i, are indeed in postorder.

Next, suppose to the contrary that F is not executable, and let f_i be the first GPC in F which cannot be executed. Write $f_i = (l_i, t_1, \ldots, t_q)$ and suppose that when f is to be executed, t_k is no longer a descendant of t_{k+1} (note that the leaf l_i of t_1 has not yet been involved in any GPC, and is still a son of t_1). Then there had to exist a GPC f_j with $j < i$ which has separated t_k from t_{k+1}; thus f_j contains a node u with $t_k \leq u < t_{k+1}$, and its last node v is such that $t_{k+1} < v$. Since $j < i$, we also have $s_j < s_i = t_1 \leq t_k$. Therefore $j \in c_{s_j}$, c_u, c_v, and $i \in c_{t_k}$, $c_{t_{k+1}}$, implying that U contains a forbidden subsequence of the form $jiji$, contrary to assumption (note that this

occurs even if $t_k = u$, because in the decreasing chain c_u the symbol i must precede j). ∎

Proof of Theorem A(i). The transformation described above yields $\lambda_3(n) \leq \psi(p, n) \leq$ $\leq \psi(2n, n)$, since ψ is monotone and $p \leq 2n$. ∎

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 GPC with the set of all vertices of T through which it passes, except for its last vertex. Specifically, for each $f_j = (x_1, \ldots, x_{k-1}, x_k)$, let $f_j^0 = \{x_1, \ldots, x_{k-1}\}$. Enumerate the $n+m$ nodes of T in the given postorder as $1, 2, \ldots$ $\ldots, n+m$, and define, for each $v = 1, \ldots, n+m$, a sequence

$$U_v = [j: 1 \leq j \leq m \quad \text{and} \quad v \in f_j^0],$$

where the elements of each sequenc U_v are arranged in *decreasing* order. Let U be the concatenation

$$U = U_1 \| U_2 \| \cdots \| U_{n+m}.$$

To obtain a Davenport—Schinzel sequence V from U, we proceed through the subsequences U_v in order, erasing the first element of U_v whenever it equals the preceding non-erased element of U; in total, at most $n+m-1$ elements are erased.

Proposition 4.4. *The sequence V is a $DS(m, 3)$ sequence of length $|V| \geq |F| - (n+m-1)$.*

Proof. V is composed of m symbols; the erasing procedure and the fact that each sequence U_v contains distinct elements imply that no two consecutive elements of V are equal. For each $f_j \in F$, its index j appears $|f_j|$ times in U (the last vertex of f_j is ignored), thus $|U| = |F|$ and $|V|$ satisfies the required inequality.

Finally, U — and thus V too — does not contain a subsequence of the form $ijij$. Suppose the contrary; then there exist vertices*

$$\beta \leq \gamma \leq \delta \leq \varepsilon \leq \theta$$

in T, and two distinct GPC's $f_i, f_j \in F$ such that

$$i \in U_\beta, U_\delta, U_\theta,$$

$$j \in U_\gamma, U_\varepsilon,$$

and moreover these five occurrences of i and j appear in U in the order $ijij$. Let us denote this illegal subsequence as Q.

Without loss of generality, we can assume that Q is the (lexicographically) leftmost occurrence of such an illegal subsequence in U. In particular, no j appears in U before the occurrence of i in U_β (for otherwise we would have an illegal subsequence of the form $jijij$ lying left of Q). Moreover, we can assume for similar reasons that U_β con-

---

* All inequalities between vertices refer to the given postorder.

tains the first occurrence of i in U and that U_γ contains the first occurrence of j in U. By definition we must have $\beta = \sigma(f_i)$, $\gamma = \sigma(f_j)$, so that β and γ are *distinct* leaves of T. Thus $\beta < \gamma$ and so $i < j$ since the sequence F is postordered.

Furthermore we also have $\delta \neq \varepsilon$, for otherwise U_δ would contain both i and j, and j would have to precede i in U_δ (since $i < j$). But then our subsequence Q would appear in U in the order $ijjij$, contrary to assumption. Moreover, γ is a leaf whereas δ is not (f_i starts at β and then passes through δ), so that we also have $\gamma \neq \delta$.

Now $\beta < \gamma < \delta$ and β is a proper descendant of δ (f_i contains both of them). Since the vertices of T are arranged in postorder, γ must also be a proper descendant of δ. By the same argument, δ must be a proper descendant of ε and ε must be a (possibly improper) descendant of θ.

Let ζ be the least common ancestor of β and γ in T; then ζ is a descendant of δ. Since $\beta \neq \gamma$ are leaves, ζ is not a leaf, and so $\zeta \neq \beta, \gamma$. Let ξ be the son of ζ on the path from γ to ζ; see Figure 4.1.
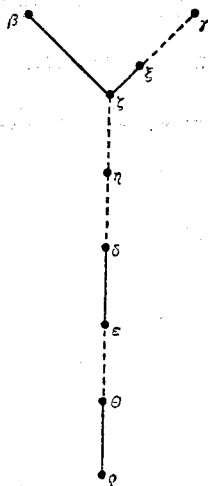

Fig. 4.1. Initial arrangement of the nodes β, γ, ... in T. A solid edge denotes a nonempty path, whereas a dashed edge denotes a possibly empty path

Since $\beta < \gamma$ we also have $\beta < \xi$, and in fact every leaf of the subtree T_ξ of T rooted at ξ succeeds β in postorder. Since the GPC's of F are executed in postorder, it follows that by the time f_i is executed, no GPC has yet passed through any vertex of T_ξ; in particular γ is still a descendant of ζ at that time.

Let η be the first vertex (i.e. furthest from the root of T) in f_i which is a (possibly improper) ancestor of ζ. Note that since $\delta \in f_i$, η must lie on the path between ζ and δ (including ζ, δ).

After f_i is executed, γ remains a descendant of ζ and thus of η too. Let ϱ be the last node of f_i (note that $\theta < \varrho$). Execution of f_i has made η a son of ϱ. Now ε lied originally strictly between η and ϱ (strictly because $\delta \neq \varepsilon$ and $\theta \neq \varrho$). Thus, after

execution of f_i, η, and hence *a fortiori* γ, will no longer be a descendant of ε. But this implies that f_j cannot be executed, a contradiction which completes the proof. ∎

Proof of Theorem A(ii). The above transformation plainly gives the required inequality. ∎

5. The upper bounds

In this section we prove Theorem B, i.e. establish an upper bound on ψ (and hence also on λ_3). To obtain this bound we first derive a recurrence relation for the function ψ.

Proposition 5.1. *Let* $n, m \geq 1$, *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

(*) $$\psi(n, m) \leq \psi(b-1, m^*) + 2n + 2m^* + \sum_{i=1}^{b} \psi\left(\frac{n}{b}, m_i\right).$$

Proof. It suffices to consider only the extreme case of (n, m)-trees all of whose inner nodes lie along a single path; we will refer to such a tree as being *canonical*. Indeed, given any (n, m)-tree T, transform it into a canonical (n, m)-tree T_0 by arranging its inner nodes along a single path, according to their given postorder (the leaves remain attached to the same nodes). It is easily seen that any compression scheme on T is also a scheme on T_0.

Let then T be any canonical (n, m)-tree, and let $b > 1$ be a divisor of n. Partition the tree T into b layers $L_1, \ldots, L_b$ as follows: Let $1, \ldots, n$ denote the inner nodes of T in postorder (n is the root), then the layer L_i consists of the $p = n/b$ inner nodes $(i-1)p+1$, $(i-1)p+2$, ..., ip, together with all the leaves attached to them. 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 $i = 1, \ldots, b$, 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

(1) $$\sum_{i=1}^{b} \psi\left(\frac{n}{b}, m_i\right).$$

(II) GPC's that contain vertices of more than one layer. Let m^* denote the number of GPC's of this type; plainly, $m^* + \sum_{i=1}^{b} m_i = m$. The total length of these GPC's is the total number of their edges; call an edge (x_j, x_{j+1}) *internal* if x_j and x_{j+1} belong to the same layer, and *crossing* otherwise. The layer L to which $\sigma(f)$ belongs will be referred to as the *starting layer* of the GPC f.

The total length of the GPC's of type II can thus be written as $N_1 + N_2 + N_3$, where N_1 is the number of internal edges within the starting layer of their GPC, N_2 is the number of internal edges within non-starting layers of their GPC, and N_3 is the number of crossing edges.

6*

To estimate N_1 we argue as follows. Fix a layer L_i, for $i=1, \ldots, b-1$, and let Q_i denote the set of all inner nodes of L_i whose fathers do not belong to L_i. Initially $|Q_i|=1$, and we clearly always have $|Q_i| \le n/b$. Consider any GPC f which starts in L_i, contains $k_f \equiv k_f^i \ge 1$ internal edges within L_i, and then crosses over to another layer. Then f passes through k_f inner nodes of L_i, all of which become members of Q_i after the execution of f. Note that only the last node of f in L_i could already belong to Q_i before the execution of f; all the other k_f-1 nodes certainly did not. Hence the execution of f adds at least k_f-1 new nodes to Q_i, so that

$$\sum_f (k_f-1) \le \frac{n}{b}-1,$$

where summation is over all f's of type II as above. Let q_i be the number of such GPC's. Then their total contribution to N_1 is

$$\sum_f k_f = q_i + \sum_f (k_f-1) \le q_i + \frac{n}{b}-1.$$

Hence, summing over all layers, and observing that $\sum_{i=1}^{b-1} q_i \le m^*$, we obtain

(2) $$N_1 \le \sum_{i=1}^{b-1} \left(q_i + \frac{n}{b}-1 \right) \le m^* + n.$$

For the estimation of N_2 we need the following lemma, which establishes an important property of compression schemes.

Lemma 5.2. *Let F be a compression scheme on a tree T, and let u and v be two nodes in some $f \in F$. Let w be a node such that $u < w < v$ in the given postorder (thus, in particular, w was initially a descendant of v). Then at the time f is executed, w is still a descendant of v.*

Proof. Suppose the contrary, and assume that f is the first GPC in F for which this property fails to hold for some $u, v \in f$. Then, arguing as in the proof of Proposition 4.3, there had to exist another GPC g preceding f in F which has separated w and v; i.e. g contains a vertex z such that, just before executing g, w has been a (possibly improper) descendant of z, z has been a proper descendant of v, and v has been a proper descendant of the last node r of g. Then $\sigma(g) < \sigma(f) \le u < w \le z$. Since $\sigma(g)$ has been originally a descendant of z, by postorder so was $\sigma(f)$. Let a, b be the two consecutive nodes of g satisfying $\sigma(g) \le a < \sigma(f) \le b \le z$ (they exist because $\sigma(g)$, $z \in g$). Since the assertion of the lemma is assumed to hold for all GPC's in F preceding f, the node $\sigma(f)$ must still be a descendant of b just before executing g. But at that time b is a (possibly improper) descendant of z (because b, $z \in g$). Hence, immediately after g is executed, b, and hence $\sigma(f)$ too, are disconnected from v (b becomes a son of r). But this makes f unexecutable, a contradiction which completes the proof. ∎

The estimation of N_2 now proceeds as follows. We fix a layer L_i, for $i=2, \ldots, b$, and bound the contribution to N_2 due to internal edges within L_i. Let P_i be the set of inner nodes of L_i lying on the path from the first inner node w of L_i to the root of

T. Initially $|P_i| = n/b$, and P_i always contains at least one node, namely w. Consider a GPC f with a crossing edge (u, v) entering L_i and with $l \equiv l_f \geqq 1$ internal edges in L_i. Let $v \equiv v_0, v_1, \ldots, v_l$ denote the nodes of f in L_i. It follows from Lemma 5.2 that, just before f is executed, w is still a descendant of v_0, so that P_i currently contains all the nodes $v_0, \ldots, v_l$. After execution of f all the nodes $v_1, \ldots, v_{l-1}$ will be deleted from P_i; if f terminates in another layer (succeeding L_i), then v_l will also be deleted from P_i. Hence $|P_i|$ is decreased after executing f by l_f^* nodes, where l_f^* is $l_f - 1$ if f terminates in L_i, and is l_f otherwise. Therefore

$$\sum_f l_f^* \leqq \frac{n}{b} - 1,$$

where the summation is over all f's of this sort. Let p_i denote the number of such GPC's which terminate in L_i, Then the total contribution to N_2 within L_i is

$$\sum_f l_f = p_i + \sum_f l_f^* \leqq p_i + \frac{n}{b} - 1.$$

Hence, adding these bounds for $i = 2, \ldots, b$ we obtain

(3)
$$N_2 \leqq \sum_{i=2}^{b} \left(p_i + \frac{n}{b} - 1 \right) \leqq m^* + n,$$

since $\sum_{i=2}^{b} p_i \leqq m^*$.

Finally, for N_3 we use the following construction:

Let T^* be a canonical tree of the following form. It has $b - 1$ inner nodes $y_2 < \ldots < y_b$, each y_i representing the layer L_i; in addition, each y_i has l_i leaves attached to it, where l_i is the number of GPC's of type II in F whose starting layer is L_{i-1}. To simplify notation, we will identify the l_i leaves of y_i with the starting leaves of these GPC's. For each f_j of type II, let g_j be the sequence of layers through which f_j passes, without repetitions (i.e. each such layer appears in g_j just once): $g_j = (i_1, i_2, \ldots, i_r)$ (where $1 \leqq i_1 < i_2 < \ldots < i_r \leqq b$). We transform g_j into a GPC f_j^* on T^*: $f_j^* = (z_j, y_{i_2}, \ldots, y_{i_r})$ where $z_j = \sigma(f_j)$ (which is a leaf of L_{i_1} in T). Note that y_{i_1} does not appear in f_j^*, whose length $|f_j^*| = r - 1$ precisely equals the number of crossing edges of f_j.

Lemma 5.3. *The resulting sequence* $F^* = (f_j^*)$ *is a compression scheme on* T^*.

Proof. First observe that F^* is postordered relative to the postorder on T^* in which all auxiliary leaves of any y_i succeed y_{i-1}. Second, suppose to the contrary that F^* is not executable, and let f^* be the first nonexecutable GPC in F^*: $f^* = (z, y_{i_2}, \ldots, y_{i_r})$, with z a leaf of L_{i_1} in T. We must have one of the following two situations.

(i) y_{i_s} *is not a descendant of* $y_{i_{s+1}}$ *for some* $2 \leqq s \leqq r - 1$.

Then there had to exist a GPC h^* preceding f^*, which contained two inner nodes y_p and y_q of T^*, where $i_s \leqq p < i_{s+1} < q$. Thus the original GPC f contains some nodes $a \in L_{i_s}$, $b \in L_{i_{s+1}}$, and h contained some nodes $c \in L_p$, $d \in L_q$. If $a \leqq c$, then $\sigma(h) < \sigma(f) \equiv z < a \leqq c < b < d$; by Lemma 5.2, a and b are still descendants of c and

d, respectively, when h is executed, after which they certainly become disconnected. But this makes f unexecutable, so we must have $c<a$, and thus $p=i_s$. Moreover, $c<z<a$, otherwise z and a would be disconnected by h (by the same argument as above). Thus f starts in L_{i_s}, hence $s=1$, again a contradiction.

(ii) *z is not a descendant of y_{i_2}.*

Since no other GPC starts at z, it follows that y_{i_1+1}, to which z is attached, is not a descendant of y_{i_2}. Therefore there had to exist a previous h^* which contained two inner nodes y_p and y_q of T^*, where $i_1+1 \leq p < i_2 < q$. Hence h contained some $c \in L_p$, $d \in L_q$. But then $\sigma(h) < z < c$ and h would have disconnected z from L_{i_2}, so that f would not be executable, a contradiction which completes the proof. ∎

Corollary 5.4.

(4) $$N_3 \leq \psi(b-1, m^*).$$

Proof. Immediate from the preceding lemma. ∎

Combining the estimates (1)—(4), we finally obtain the desired recurrence relation (*), and thus complete the proof of Proposition 5.1. ∎

Proposition 5.5. *Let* $m, s \geq 1$, $k \geq 2$, *and suppose that n divides $A_k(s)$. Then*

(**) $$\psi(n, m) \leq (2k-2)ns + (2k-1)m.$$

Proof. We will use (*) repeatedly to obtain the series of upper bounds on ψ, stated in (**) for $k=2, 3, \ldots$. 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 ψ which we solve to obtain a better upper bound on ψ.

Specifically, we proceed by double induction on k and s. To start this iterative process with $k=2$, suppose first that $n=A_2(s)=2^s$. Choose $b=2$ in (*); it is easily checked that $\psi(b-1, m^*)=\psi(1, m^*)=m^*$ for all m^*, so that (*) yields

$$\psi(n, m) \leq 2n + 3m^* + \psi\left(\frac{n}{2}, m_1\right) + \psi\left(\frac{n}{2}, m_2\right).$$

The solution to this recurrence relation, for n a power of 2 and $m=m^*+m_1+m_2$ arbitrary, is

$$\psi(n, m) \leq 2n \log n + 3m.$$

(This is obvious for $n=1$ because $\psi(1, m)=m$. Assuming it holds for $n/2$, we obtain

$$\psi(n, m) \leq 2n + 3m^* + 4\frac{n}{2}\log\frac{n}{2} + 3m_1 + 3m_2 = 2n \log n + 3(m^* + m_1 + m_2)$$

$$= 2n \log n + 3m,$$

thus the claim follows by induction.)

To complete the argument for $k=2$, note that if n divides $A_2(s)=2^s$ then n is a power of 2 and $\log n \leq s$, thus $\psi(n, m) \leq 2n \log n + 3m \leq 2ns + 3m$.

In particular, we have for $n=1,2$

$$\psi(n, m) \leqq 2n+3m \leqq (2k-2)n+(2k-1)m$$

for all $k \geqq 2$. Since $A_k(1)=2$ it follows that (**) holds for each $k \geqq 2$ and $s=1$.

Suppose next that $k>2$ and $s>1$, and that the induction hypothesis is true for all $k'<k$ and $s' \geqq 1$, and for $k'=k$ and all $s'<s$. Observe that $A_k(s-1)$ is a divisor of $A_k(s)$ because they are both powers of 2. Assume first that $n=A_k(s)$; let $t=A_k(s-1)$ and choose $b=n/t$, which is an integer dividing $n=A_k(s)=A_{k-1}(t)$. Hence by the induction hypothesis (for $k-1$ and t) we have

$$\psi(b-1, m^*) \leqq \psi(b, m^*) \leqq (2k-4)bt+(2k-3)m^* = (2k-4)n+(2k-3)m^*.$$

Then (*) becomes

$$\psi(n, m) \leqq (2k-4)n+(2k-3)m^*+2n+2m^*+\sum_{i=1}^{b} \psi(t, m_i).$$

Using the induction hypothesis once more (for k and $s-1$), we obtain

$$\psi(n, m) \leqq (2k-2)n+(2k-1)m^*+\sum_{i=1}^{b}((2k-2)t(s-1)+(2k-1)m_i)$$

$$= (2k-2)ns+(2k-1)\left(m^*+\sum_{i=1}^{b} m_i\right) = (2k-2)ns+(2k-1)m,$$

because $m^*+\sum_{i=1}^{b} m_i=m$.

Finally, assume n divides $A_k(s)$, say $A_k(s)=pn$. Let T be a canonical (n, m)-tree and F a compression scheme on T of length $\psi(n, m)$. Take p copies of T, and construct a canonical (pn, pm)-tree T^* by concatenating them in sequence. Let F^* be the concatenation of the corresponding p copies of F. Since $|F^*|=p|F|=p\psi(n, m)$, we obtain

$$p\psi(n, m) \leqq \psi(pn, pm) = \psi(A_k(s), pm) \leqq (2k-2)sA_k(s)+(2k-1)pm,$$

which, divided by p, yields the required inequality. This completes the proof. ∎

Corollary 5.6. *For all* $n, m \geqq 1$ *and* $k \geqq 2$,

$$\psi(n, m) \leqq (4k-4)n\alpha_k(n)+(2k-1)m,$$

where α_k *is the functional inverse of* A_k *as defined in Section 3.*

Proof. Put $s=\alpha_k(n)$, so that $A_k(s-1)<n\leqq A_k(s)$. Let $p=\lfloor A_k(s)/n\rfloor \geqq 1$, then $A_k(s)<(p+1)n\leqq 2pn$. As in the preceding proof we have

$$p\psi(n, m) \leqq \psi(pn, pm) \leqq \psi(A_k(s), pm) \leqq (2k-2)sA_k(s)+(2k-1)pm$$

$$\leqq (4k-4)pns+(2k-1)pm.$$

Dividing by p we obtain the desired inequality. ∎

Proof of Theorem B. We claim that

(B) $$\psi(n, m) \leqq (24n+4m)\alpha(n)$$

for all $n, m \geq 1$. Indeed, assume first $n > 4$ so that $\alpha(n) \geq 3$. Let $k = 2\alpha(n) - 3$, then $n \leq A_{\alpha(n)}(\alpha(n)) \leq A_k(3)$ by a repeated application of Lemma 3.2, hence $\alpha_k(n) \leq 3$. Corollary 5.6 yields

$$\psi(n, m) \leq (4k-4)\cdot 3n + (2k-1)m \leq 24n\alpha(n) + 4m\alpha(n),$$

as required. For $n \leq 4$ the desired inequality can be verified directly from Corollary 5.6. ∎

Remark. Inequality (B) and Theorem A (i) imply

$$\lambda_3(n) \leq \psi(2n, n) \leq (48n+4n)\alpha(2n) \leq 52n\big(\alpha(n)+1\big).$$

6. The lower bounds

In this section we will establish nonlinear lower bounds for ψ (and, *a fortiori*, also for λ_s for all $s \geq 3$) which match closely the upper bounds just obtained. This is achieved using a construction similar to that of [16]. 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.

6.1. The functions B_k and their properties

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

$$B_1(s) = 0, \quad s \geq 0,$$

$$B_k(0) = 1, \quad k \geq 2,$$

$$B_k(s) = B_k(s-1) + B_{k-1}(2^{B_k(s-1)}), \quad k \geq 2, \ s \geq 1.$$

Here are some properties of the functions B_k.

(B1) $\qquad\qquad\qquad\qquad B_2(s) = 1, \quad for \quad s \geq 0.$

(B2) $\qquad\qquad\qquad\qquad B_3(s) = s+1, \quad for \quad s \geq 0.$

(B3) $B_4(s) \geq 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}$, *with s 2's in the exponential tower, for* $s \geq 0$; *i.e.* $B_4(s) \geq A_3(s)$.

Indeed, $B_4(0) = 1$, and $B_4(s) = B_4(s-1) + 2^{B_4(s-1)} + 1 \geq 2^{B_4(s-1)}$, for $s \geq 1$.

(B4) *Each function $B_k(s)$ is strictly increasing in s, for all $k \geq 3$, and each sequence $\{B_k(s)\}_{k \geq 1}$ is strictly increasing, for all $s \geq 1$.*

To see this note first that $B_k(s) \geq 1$ for all $k \geq 2$ and $s \geq 0$. This implies, for $k \geq 3$ and $s \geq 0$, that

$$B_k(s+1) = B_k(s) + B_{k-1}(2^{B_k(s)}) \geq B_k(s) + 1,$$

hence also $B_k(s) \geq s+1$. Finally, we obtain

$$B_{k+1}(s) \geq B_k(s) + 1, \quad k \geq 1, \ s \geq 1.$$

(This is clear for $k=1$. For $k>1$ we have $2^{B_{k+1}(s-1)} \geqq 2^s \geqq s$, hence

$$B_{k+1}(s) = B_{k+1}(s-1) + B_k(2^{B_{k+1}(s-1)}) \geqq 1 + B_k(s).)$$

(B5) $B_k(s) \geqq A_{k-1}(s)$,   for $k \geqq 4$, $s \geqq 1$.

This is proven by double induction on k and s. (B5) holds for $k=4$ by (B3), and holds for $s=1$ since

$$B_k(1) = 1 + B_{k-1}(2) \geqq 2 = A_{k-1}(1).$$

For $k>4$ and $s>1$, assume (B5) to hold for all $4 \leqq k' < k$ and $s' \geqq 1$, and for $k'=k$ and all $1 \leqq s' < s$. Then we have

$$2^{B_k(s-1)} \geqq B_k(s-1) \geqq A_{k-1}(s-1),$$

hence

$$B_k(s) \geqq B_{k-1}(2^{B_k(s-1)}) \geqq B_{k-1}(A_{k-1}(s-1)) \geqq A_{k-2}(A_{k-1}(s-1)) = A_{k-1}(s).$$

(B6) $2^{B_k(s)+2} \leqq A_k(s+3)$,   for $k \geqq 1$, $s \geqq 0$.

This is again proven by double induction on k and s. For $k=1$ we have $2^{B_1(s)+2}=4$ and $A_1(s+3)=2(s+3) \geqq 4$.
For $k=2$, $2^{B_2(s)+2}=8$ and $A_2(s+3)=2^{s+3} \geqq 8$.
For $k \geqq 3$ and $s=0$, we have $2^{B_k(0)+2}=8$ and

$$A_k(3) = A_{k-1}(A_k(2)) = A_{k-1}(4) \geqq A_1(4) = 8.$$

Finally, for $k \geqq 3$ and $s \geqq 1$, put $t=B_k(s-1)$; then

$$B_k(s) = t + B_{k-1}(2^t) \leqq 3 \cdot (2^t-1) + B_{k-1}(2^t)$$

which, by (B4), is

$$\leqq B_{k-1}(2^t + 3 \cdot (2^t-1)) = B_{k-1}(2^{t+2}-3).$$

Thus

$$2^{B_k(s)+2} \leqq 2^{B_{k-1}(2^{t+2}-3)+2} \leqq A_{k-1}(2^{t+2})$$

$$= A_{k-1}(2^{B_k(s-1)+2}) \leqq A_{k-1}(A_k(s+2)) = A_k(s+3),$$

using the induction hypothesis twice.

Note, as a corollary, that

(B7) $A_{k-1}(s) \leqq B_k(s) \leqq A_k(s+3)$,   for $k \geqq 4$, $s \geqq 1$,

so that the sequences of functions B_k and A_k are indeed of the same order of magnitude.

6.2. The derivation of the lower bounds

Although the proof of the upper bounds in the preceding section has dealt exclusively with canonical (n, m)-trees, we will find it more convenient in this section to work with (essentially) symmetric binary trees rather than with single-path ones (of course, as noted in Section 5, any compression scheme on an arbitrary (n, m)-tree can be translated to the corresponding canonical tree).

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. In the sequel, we always consider only such induced postorders on the trees $T(i)$.
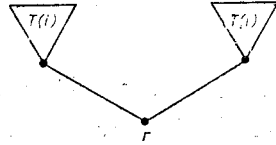

Fig. 6.1. $T(i+1)$

Theorem 6.1. *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.*

Proof. We use an argument similar to that in [16, Theorem 15], based on double induction on k and m. Note that if the assertion of the theorem holds for some i, it also holds for all $j > i$; it thus suffices to prove it for $i = B_k(m)$. Moreover, we remark that the theorem holds for (n, m)-trees with arbitrary $n \geq 1$ as well; to reduce it to the case $n = 1$, ignore the inner structure of the tree, and assume that each leaf is connected directly to the root.

Suppose first that $k = 1$. Since $B_1(m) = 0$, it suffices to prove the assertion for T itself. Indeed, take F to be the sequence of GPC's $((l_1, r), ..., (l_m, r))$, where $l_1, ..., l_m$ are the leaves of T arranged in postorder, and r is the root of T.

Next suppose that $k = 2$ (this step is not essential to our inductive proof, but is given anyway to prepare for the following more complex steps). Consider $T(1) = T(B_2(m))$, and let r_0 denote its root. For each leaf l of $T(1)$ take the GPC (l, r, r_0) of length 2, where r is the root of the copy of T containing l, and let F be the postordered sequence of these GPC's.

Now suppose that $k \geq 3$. Assume that the assertion holds for all $k' < k$ and all $m' \geq 1$, and consider first the case $m = 1$. T consists of just one leaf l and a root r (note that $l \neq r$). Let T^* be the subtree of $T(1)$ obtained by removing its (two) leaves; T^* is a $(1, 2)$-tree. Let $i_0 = B_{k-1}(2)$ and $i = B_k(1)$; then, by definition, $i = 1 + i_0$. By the induction hypothesis, there exists a compression scheme F^* on $T^*(i_0)$, all of whose GPC's are of length $k - 1$. Now $T^*(i_0)$ may be regarded as a subtree of $(T(1))(i_0) = T(i)$, and we will extend each GPC $f^* = (x_1, ..., x_k) \in F^*$ to a GPC $f = (x_0, x_1, ..., x_k)$ in $T(i)$, by adding to f^* the leaf x_0 which is the unique son of x_1. The resulting sequence F is clearly postordered and executable in $T(i)$, and each of its GPC's has length k, thus F is the required compression scheme.

Finally, let $k \geq 3$ and $m > 1$, and assume that the assertion holds for all $k' < k$ and $m' \geq 1$, and for $k' = k$ and all $m' < m$. Let T be a $(1, m)$-tree, and let $l_1, \ldots, l_m$ be its leaves arranged in postorder. Let T^* be the $(1, m-1)$-tree obtained from T by removing its last leaf l_m. Put $i_0 = B_k(m-1)$, $i_1 = B_{k-1}(2^{i_0})$, and $i = B_k(m)$; thus $i = i_0 + i_1$.

Consider $T(i_0)$ and its subtree $T^*(i_0)$. By the induction hypothesis, there exists a compression scheme on $T^*(i_0)$, all of whose GPC's have length k. Thus, starting at each leaf $l_1, \ldots, l_{m-1}$ in each copy of T in $T(i_0)$, we have a GPC of length k. Now $T(i) = (T(i_0))(i_1)$ contains 2^{i_1} copies of $T(i_0)$, and in each of these copies we have a corresponding compression scheme of the above form. Refer to the GPC's in all these schemes as GPC's of *type I*.

Next consider the subtree T^{**} of $T(i_0)$, obtained by removing all its $m \cdot 2^{i_0}$ leaves. Since T is a $(1, m)$-tree, the leaves of T^{**} are precisely the $p = 2^{i_0}$ copies (in $T(i_0)$) of the root of T. Regard T^{**} as a $(1, p)$-tree by ignoring its inner structure, and apply to it the induction hypothesis (recall the note at the beginning of this proof). We obtain a compression scheme F^{**} on $T^{**}(i_1)$, consisting of GPC's of length $k - 1$ each, such that only the root and the leaves of each copy of T^{**} (but none of its intermediate nodes) participate in F^{**}. Now $T^{**}(i_1)$ is a subtree of $(T(i_0))(i_1) = T(i)$; we extend each GPC $f^{**} = (x_1, \ldots, x_k) \in F^{**}$ to a GPC $f = (x_0, x_1, \ldots, x_k)$ in $T(i)$ as follows: x_1 is the root of some copy of T, and we take x_0 to be the last leaf there (i.e. the copy of l_m). We will refer to these GPC's as being of *type II*.

We can now obtain a compression scheme on $T(i)$ by merging the GPC's of types I and II; namely, we proceed through the leaves of $T(i)$ in their induced postorder; for each leaf which is a copy of $l_1, \ldots, l_{m-1}$ (resp. of l_m), we take the corresponding GPC of type I (resp. type II) starting at that leaf.

The sequence F of GPC's obtained in this way is the required compression scheme on $T(i)$. It is clearly postordered, and its executability can be established by proceeding inductively through the sequence, and using the following arguments:

(a) All the postorders considered here (on $T(i)$, $T^*(i_0)$, and $T^{**}(i_1)$) are consistent with one another, since they are all induced by the original postorder on T. This implies the separate executability of the sequence of GPC's of type II, and also of those of type I (in each copy of $T(i_0)$ separately, and also all together).

(b) Let $f = (l', \ldots, z)$ be a GPC of type I, where l' is a non-last leaf, and let r' be the father of l' (i.e. r' is the root of the copy of T containing l'). All the nodes of f are included in the same copy of $T(i_0)$; therefore, if g is any GPC of type II which contains a node on the path from l' to z (excluding z), then this node must be r', and g is thus executed only after f (since $\sigma(g)$ is the last leaf of r'). By (a), the other GPC's of type I also do not affect the executability of f.

(c) Let $f = (l', r', u, \ldots, z)$ be a GPC of type II, where l' is a last leaf and r' is its father; let q' be the root of the copy of $T(i_0)$ which contains l' and r', then q' is a (possibly improper) descendant of u. The GPC f is executable at its turn, because:

(i) l' belongs to no other GPC, thus it still is a son of r'.

(ii) For any GPC g of type I, if it contains a node on the path from l' to z, then all its nodes lie in the copy of $T(i_0)$ with root q', thus g does not disconnect r' from q' and u, and of course does not affect the other nodes of f.

(iii) Finally, the other GPC's of type II do not affect the executability of f (by (a)).

This establishes the assertion of the theorem for k and m too, and thus completes the proof. ∎

Corollary 6.2. *For all* n, $m \geq 1$

$$\psi(n, m) \geq \frac{1}{4}\beta m,$$

where

$$\beta \equiv \beta(n, m) = \max\left\{k \geq 1 : A_k\left(\left\lceil\frac{2m}{n}\right\rceil + 3\right) \leq 2n\right\},$$

and $\beta = 1$ *if there is no such* k.

Proof. Since $\psi(n, m) \geq m$ always, we can assume $\beta > 1$. Consider first the case $2m \geq n$. Let $s = \lfloor 2m/n \rfloor \geq 1$, and apply Theorem 6.1 to a $(1, s)$-tree, to obtain for all $i \geq B_\beta(s)$

$$\psi(2^{i+1} - 1, s \cdot 2^i) \geq \beta s 2^i.$$

Put $b = \frac{1}{2}A_\beta(s+3) \leq n$ and $i = \log b - 1$; then $2b \geq 2^{B_\beta(s)+2}$ by (B6), hence $i \geq \geq B_\beta(s)$ and we have

$$\psi\left(b - 1, \frac{sb}{2}\right) \geq \beta \frac{sb}{2}.$$

Now let $p = \lfloor n/b \rfloor \geq 1$, then $pb \leq n$ and $psb/2 \leq ns/2 \leq m$. Using the same argument as in the proof of Proposition 5.5 we obtain

$$\psi(n, m) \geq \psi\left(pb, \frac{psb}{2}\right) \geq p\psi\left(b, \frac{sb}{2}\right) \geq p\psi\left(b - 1, \frac{sb}{2}\right)$$

$$\geq p\beta\frac{sb}{2} \geq \frac{1}{4}\beta m,$$

since $\lfloor n/b \rfloor \cdot b \geq n/2$ and $\lfloor 2m/n \rfloor \cdot n \geq m$.

Note that the compression scheme we have obtained has all its GPC's of length β. Thus, for any $m' \leq m$, we can remove $m - m'$ leaves together with their associated GPC's obtaining a compression scheme consisting of m' GPC's of length β each, on the modified tree. Thus

$$\psi(n, m') \geq \frac{1}{4}\beta(n, m) \cdot m'$$

for all $m' \leq m$. In particular, for $m' \leq m = n/2$, we obtain

$$\psi(n, m') \geq \frac{1}{4}\beta\left(n, \frac{n}{2}\right) \cdot m';$$

but by definition of β, in this case we have $\beta(n, m') = \beta(n, n/2)$, and this completes the proof. ∎

Proof of Theorem C. Apply Corollary 6.2 to obtain

$$\psi(2n, n) \geq \frac{1}{4}\beta n,$$

where

$$\beta = \max\{k \geq 1 : A_k(4) \leq 4n\}.$$

If $k \equiv \alpha(n) \geq 4$, then

$$A_k(4) \leq A_k(k) = A(k) \leq n \leq 4n,$$

thus $\beta \geq \alpha(n)$, and

(C) $$\psi(2n, n) \geq \frac{1}{4} n\alpha(n).$$

If $\alpha(n) \leq 3$, then, since we trivially always have $\psi(n, m) \geq m$, (C) holds in this case too. ∎

Remark. Equation (C) together with Theorem A (ii) imply

$$\lambda_3(n) \geq \psi(2n, n) - 3n \geq \frac{1}{4} n\alpha(n) - 3n.$$

Remark. The construction in Theorem 6.1 and Proposition 4.4 can be combined to provide an explicit construction of $DS(n, 3)$ sequences of nonlinear size. Thus the tree duplication operation $T(i)$ corresponds to a similar sequence duplication operation, and the inductive construction of compression schemes on the trees $T(i)$ can be translated to an inductive construction of $DS(n, 3)$ sequences. However, it seems that this explicit construction becomes much less intuitive than the implicit one through GPC's on trees.

7. Concluding remarks

7.1. Applications

Besides the numerous applications noted in [2], 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 line segments. 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 I_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 $O(n\alpha(n))$.

Proof. Let K be a real number greater than the absolute value of the slope of each of the line segments I_k. Extend each line segment I_k to a continuous piecewise linear function over the whole x-axis, by making it linear with slope $+K$ on the right of I_k and linear with slope $-K$ on its left. It is easily checked that each pair of these extended functions intersect in at most 3 points, so that the required number m of x-intervals is at most $\lambda_3(n)$, thus $O(n\alpha(n))$. ∎

Remark. It is an open problem whether $\lambda_3(n)$ can actually be obtained in this set-up.

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

Proof. Take each of the n line segments appearing in the graphs of the f_i's, and extend it to a continuous piecewise linear function as in (1) above. It is plain that the lower envelope of the f_i's is identical to the lower envelope of these extended functions, from which the first part of our assertion follows. Calculation of this lower envelope within the stated time bound can be accomplished using a straightforward "divide and conquer" algorithm, which partitions the collection of the n extended functions into two equal subsets, computes recursively the lower envelope of the functions in each of these subsets, and finally merges the two lower envelopes into the lower envelope of all the n functions, in time that is linear in the number of line segments forming these two envelopes.

(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 $1, ..., n$. 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 n elements into S, then swap them, and finally delete them all) then plainly $C(n) = \Theta(n)$. In the dynamic case however we have

Proposition. $C(n) = \lambda_3(n)$; thus $C(n) = \Theta(n\alpha(n))$.

Proof. Let U be the sequence of elements as they appear in the first place of S. It is easy to see that U is a $DS(n, 3)$ sequence: Plainly no two adjacent elements of U are equal. Suppose that there exist five indices $i < j < k < l < m$ such that $u_i = u_k = u_m = a$, $u_j = u_l = b$ and $a \neq b$. Then the (middle) appearance of a as u_k must have been preceded by a swap of a and b, and must later be followed by a similar swap. This however is impossible, because two elements can be swapped at most once. Therefore $C(n) \leq \lambda_3(n)$.

For the converse statement, let $U = (u_1, ..., u_m)$ be any $DS(n, 3)$ sequence. We obtain from U a dynamic sorting process (with U its corresponding sequence of first elements), as follows. As in Section 4.1, assume without loss of generality that for any two symbols a and b appearing in U, we have $a < b$ if and only if the index μ_a of the first occurrence of a in U is less than the index μ_b of the first occurrence of b. We start with an empty sequence S, and for each $k = 1, ..., m$ we do one of the following (sequences of) operations on S, where $a = u_k$.

(i) If u_k is the first occurrence of a in U, we insert a as the first element of S.

(ii) Otherwise, a presently appears somewhere in S (but not as its first element). We then move a to the first place of S by repeated swapping of a with all elements lying in S before it.

(iii) If u_k is the last occurrence of a in U, we then (after (ii)) delete a from the first place of S.

We claim that we can always perform the swaps in (ii). For suppose the contrary, and let k be the first place in U where one of the corresponding swaps in (ii) cannot be executed. Let $a=u_k$, and let $b<a$ be an element appearing in S before a (so that the required swap of a with b cannot be done). Since $b<a$ we have $\mu_b<\mu_a$, and moreover $\mu_a<k<\nu_b$, where ν_b is the index of the last occurrence of b in U (since b has not yet been deleted). We claim that b must have appeared in U also at some place l between μ_a and k. Indeed, just after μ_a, b lies in S after a, whereas just before k they have changed their order in S. Since $b<a$ that must have occured by swapping b towards the first place in S, and by definition that could have happened only due to an intermediate appearance of b in U. This however implies that U contains the illegal subsequence $babab$ at the five indices $\mu_b<\mu_a<l<k<\nu_b$, contrary to assumption. This shows that $\lambda_3(n)\leqq C(n)$, and hence these two functions are equal. ∎

7.2. Open Problems

(1) For $s>3$, 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 is whether the upper bound of [15] on $\lambda_s(n)$ can be improved for $s>3$. This problem was settled recently, after the original submission of this paper, in [13], where Szemerédi's bound is improved for $s>3$ to bounds which are roughly of the form

$$\lambda_s(n) = O\big(n\alpha(n)^{O(\alpha(n)^{s-3})}\big).$$

However, no lower bounds, better than the $\Omega\big(n\alpha(n)\big)$ bound obtained in this paper, are known as yet for $s>3$.

(2) We have already raised the question whether $\lambda_3(n)$ can be attained for the pointwise minimum of n line segments. A related open problem is whether $\lambda_3(n)$ can be attained for the minimum of n cubic polynomials (it is easy to check that both $\lambda_1(n)$ and $\lambda_2(n)$ are attained for the minimum of n linear and quadratic functions, respectively).

(3) We have obtained the bounds on $\lambda_3(n)$ 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 linear or 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 [16], 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 [16] are special instances?

(5) Consider the following two-dimensional generalization of our problem: Let $f_1(x, y), \ldots, f_n(x, y)$ be a collection of n continuous real valued functions defined over some two-dimensional rectangular region R. Suppose that for each triple of distinct indices i, j, k the surfaces of $z = f_i(x, y)$, $z = f_j(x, y)$, $z = f_k(x, y)$ intersect in at most s points. What is the maximal number of connected portions of the surfaces of the f_i's which appear in their lower envelope? More precisely, since we are interested in the overall complexity of the lower envelope, what is the maximal number $\lambda_s^{(2)}(n)$ of "corners" of this lower envelope (i.e. points where the minimum of the functions f_i is attained simultaneously by at least three of these functions)? An obvious estimate is $\lambda_s^{(2)}(n) \leq sn^3/6$. On the other hand, $\lambda_s^{(2)}(n) = \Omega(n^2)$. Indeed, such a lower bound is obtained for the following collection of $2n$ functions

$$f_i(x, y) = \left(x - \frac{i}{n}\right)^2$$

$$g_i(x, y) = 2\left(y - \frac{i}{n}\right)^2$$

for $i = 1, \ldots, n$. We leave details to the reader.

Some recent progress on this generalized 2-D problem was made by Livne and Sharir [9] (after the original submission of this paper), who have shown that $\lambda_2^{(2)}(n) = \Theta(n^2)$, and that $\lambda_6^{(2)}(n) = \Omega(n^2\alpha(n))$. Another recent result [14] yields an upper bound of the form $O(n\lambda_s(n))$ on the number of corners on the minimum of n bivariate continuous functions, under some additional assumptions concerning these functions, the most constraining of which is that, in each cross section $x = \text{const}$, each pair of the graphs of these functions intersect in at most two points.

An interesting special case of this two-dimensional problem is where the functions in question $f_i(x, y)$ are all polynomials of some maximal degree d (so that in particular, assuming non-degeneracy, each triple of them intersect in at most $s = d^3$ points). In this special case, every cross-section of the graphs of these polynomials at $x = \text{const}$ or $y = \text{const}$ yields a sequence of univariate functions each pair of which intersecting in at most d^2 points, so that the lower envelope of every such cross section behaves as a standard Davenport—Schinzel sequence.

7.3. On the role of Ackermann's functions in combinatorial problems

Although Ackermann's function A is very fast-growing, one can obtain functions which grow much faster than A by extending the recursive definitions from Section 3 as follows. For each countable ordinal γ we define a *generalized Ackermann's function* A_γ from the set $\mathbf{N}$ of positive integers into itself in the following transfinite inductive manner:

(1) $A_1(n)=2n$, $n\in\mathbf{N}$.

(2) If γ is not a limit ordinal, say $\gamma=\beta+1$, then $A_\gamma(n)=A_\beta^{(n)}(1)$, $n\in\mathbf{N}$.

(3) If γ is a limit ordinal, we associate with γ some fixed increasing sequence $\{\gamma(n)\}$ of ordinals less than γ whose limit is γ. Then $A_\gamma(n)=A_{\gamma(n)}(n)$, $n\in\mathbf{N}$.

Thus, $A=A_\omega$(if we put $\omega(n)=n$, for $n\in\mathbf{N}$).

The rate of growth of the functions A_γ increases with γ. For example, A_ω is not primitive recursive, and in fact grows faster than any primitive recursive function (see [11]). The function A_{ε_0}, where ε_0 is the first *inaccessible* ordinal (it is the limit of the sequence ω, ω^ω, ω^{ω^ω}, ...), has the following remarkable property (cf. [7]): Consider statements Q of the form $\forall n\,\exists m\,P(n, m)$, where $P(n, m)$ is a provably recursive first-order statement in Peano arithmetic with two free variables m, n (i.e. there is an algorithm for deciding if $P(n, m)$ is true for given n, m, and a proof — in Peano arithmetic — that the algorithm always terminates). Suppose that Q is a true statement, and associate with it a function $\varphi_Q: \mathbf{N}\to\mathbf{N}$, defined as follows: for each $n\in\mathbf{N}$, $\varphi_Q(n)$ is the smallest m satisfying $P(n, m)$. Then Q is provable from the Peano axioms (i.e. is a theorem of first order formal number theory) if and only if there exists an ordinal $\gamma<\varepsilon_0$ such that $\varphi_Q(n)<A_\gamma(n)$ for all sufficiently large n. A recent result of Paris and Harrington [10] yields a variant of Ramsey's Theorem which can be expressed in the above form Q, but for which the associated function φ_Q grows as fast as A_{ε_0}, so that Q is not provable from the Peano's axioms (see [8] for details).

As a matter of fact, even for the first infinite ordinal ω, there exist very few "natural" statements Q of the form $\forall n\,\exists m\,P(n, m)$ whose associated function φ_Q grows like A_ω or faster. Our results on Davenport—Schinzel sequences may well be one of the most natural statements yet known that can be expressed in this form. Indeed, let

$$Q \equiv \forall k\,\exists m\,\lambda_3(m) \geq km.$$

Our lower bound on λ_3 implies that Q is a true statement; moreover, both lower and upper bounds imply that φ_Q grows as A_ω (note that for fixed k and m the statement $\lambda_3(m)\geq km$ is a provably recursive statement in Peano arithmetic: simply enumerate all sequences of length km composed of m symbols, and check each of them for being a $DS(m, 3)$ sequence).

Tarjan's bounds [16] on the complexity of the union-find algorithm can also be translated into such a statement Q with φ_Q growing as fast as A_ω. Another example of this sort arises in the proof of Van der Waerden's theorem on the existence of arbitrarily long monochromatic arithmetic progressions in any finite coloring of the integers (cf. [6]); however, in this case A_ω only serves as an upper bound for the associated function φ_Q, and its actual rate of growth is not known.

It is also interesting to note that our inductive solution of the recurrence equation (*) for ψ as given in Section 5 can be extended in a transfinite inductive manner to obtain upper bounds for ψ which involve the inverse functions α_γ of A_γ for ordinals $\gamma>\omega$. Indeed, recall that formula (B) there yields

$$\psi(n, m) \leq C_1 n\alpha_\omega(n)+C_2 m\alpha_\omega(n)$$

for some constants C_1, $C_2>0$. Now choose $b=n/\alpha_\omega(n)$ in (*) (we finesse here details which handle the case in which $\alpha_\omega(n)$ does not divide n; the methods used in Section 5 can be applied here too). Then

$$\psi(b-1, m^*) \leq C_1 b\alpha_\omega(b)+C_2 m^*\alpha_\omega(b) \leq C_1 n+C_2 m^*\alpha_\omega(n),$$

7

so that

$$\psi(n, m) \leq (C_1+2)n+(C_2\alpha_\omega(n)+2)m^*+\sum_{i=1}^{b} \psi(\alpha_\omega(n), m_i),$$

from which we readily obtain

$$\psi(n, m) \leq (C_1+2)n\alpha_{\omega+1}(n)+(C_2\alpha_\omega(n)+2)m.$$

This process can be carried still further to yield, for each $k \geq 1$,

$$\psi(n, m) \leq (C_1+2k)n\alpha_{\omega+k}(n)+(C_2\alpha_\omega(n)+2k)m,$$

and by appropriate choice of k we can also obtain

$$\psi(n, m) \leq D_1 n\alpha_{2\omega}(n)+(C_2\alpha_\omega(n)+D_2\alpha_{2\omega}(n))m.$$

We can further extend this transfinite process to obtain upper bounds of this sort involving still larger ordinals. However, for general values of n and m, these new bounds do not improve the bounds obtained in Section 5, because the coefficients of m keep increasing. Nevertheless, if one assumes, say,

$$m = O\left(\frac{n\alpha_{2\omega}(n)}{\alpha_\omega(n)}\right)$$

then the above bound gives

$$\psi(n, m) = O(n\alpha_{2\omega}(n)),$$

which obviously improves upon $O(n\alpha_\omega(n))$. Further restrictions on m will yield still better upper bounds for ψ.

References

[1] W. ACKERMANN, Zum Hilbertschen Aufbau der reellen Zahlen, *Math. Ann.* 99 (1928), 118—133.
[2] M. ATALLAH, Dynamic Computational Geometry, *Proc. 24th Symp. on Foundations of Computer Science*, 1983, 92—99.
[3] H. DAVENPORT, A Combinatorial Problem Connected with Differential Equations, II, *Acta Arithmetica* 17 (1971), 363—372.
[4] H. DAVENPORT and A. SCHINZEL, A Combinatorial Problem Connected with Differential Equations, *Amer. J. Math.* 87 (1965), 684—694.
[5] M. J. FISHER, Efficiency of Equivalence Algorithms, *in: Complexity of Computer Computations*, (R. E. Miller and J. W. Thatcher, Eds.), Plenum Press, New York, 1972, 153—168.
[6] R. L. GRAHAM, B. L. ROTHSCHILD and J. H. SPENCER, *Ramsey Theory*, Wiley-Interscience, New York, 1980.
[7] G. KREISEL, On the Interpretation of Nonfinitistic Proofs, II, *J. Symbolic Logic* 17 (1952), 43-58.
[8] J. KETONEN and R. M. SOLOVAY, Rapidly Growing Ramsey Functions, *Ann. of Math.* 113 (1981), 267—314.
[9] R. LIVNE and M. SHARIR, On Minima of Functions, Intersection Patterns of Curves, and Davenport—Schinzel Sequences, *Proc. 26th IEEE Symposium on Foundations of Computer Science*, Portland, Ore., October 1985, 312—320.
[10] J. PARIS and L. HARRINGTON, A Mathematical Incompleteness in Peano Arithmetic, *in: Handbook of Mathematical Logic*, (ed: J. Barwise), North-Holland 1977, 1133—1142.
[11] H. ROGERS, *Theory of Recursive Functions and Effective Computability*, McGraw-Hill, 1967.
[12] D. P. ROSELLE and R. G. STANTON, Some Properties of Davenport—Schinzel Sequences, *Acta Arithmetica* 17 (1971), 355—362.

[13] M. SHARIR, Almost Linear Upper Bounds on the Length of General Davenport—Schinzel Sequences, *Combinatorica* 7 (1987), *to appear*.

[14] M. SHARIR, On the Two-dimensional Davenport—Schinzel Problem, *Techn. Rep. 193*, Comp. Sci. Dept., Courant Institute, 1985.

[15] E. SZEMERÉDI, On a Problem by Davenport and Schinzel, *Acta Arithmetica* 25 (1974), 213—224.

[16] R. E. TARJAN, Efficiency of a Good but not Linear Set-union Algorithm, *J. Assoc. Computing Machinery* 22 (1975), 215—225.

Sergiu Hart and Micha Sharir

School of Mathematical Sciences
Tel Aviv University
Tel Aviv 69978, Israel

7*