Meanders, Ramsey Theory and lower bounds for branching programs

Noga Alon+ *

+Dept. of Math., Tel Aviv University
Ramat Aviv, Tel Aviv, Israel and
Bell Commun. Research, Morristown, N.J. 07960

ABSTRACT
A novel technique for obtaining lower bounds for the time versus space complexity of certain functions in a general input oblivious sequential model of computation is developed. This is demonstrated by studying the intrinsic complexity of the following set equality problem SE(n,m):

Given a sequence x1,x2,...,xn, y1,...,yn of 2n numbers of m bits each, decide whether the sets {x1,...,xn} and {y1,...,yn} coincide. We show that for any log log n<m<n/2 log n and any 1< s< log n, any input oblivious sequential computation that solves SE(n,m) using 2m/s space, takes n(n·s) time. This result is sharp for all admissible values of n,m,s and is the first known nontrivial time space tradeoff lower bound (for space = o(log n) of a set recognition problem on such a general model of computation. Our method also supplies lower bounds on the length of arbitrary (not necessarily input oblivious) branching programs for several natural symmetric functions, improving results of Chandra, Furst and Lipton, of Pudlak and of Ajtai et. al. For example we show that for the majority - function any branching program of width w(n) has length Ω(n·log n/w(n) ·log w(n)), in particular for bounded width we get length Ω(n log n) (independently of our work Babai et. al. [BPRS] have simultaneously proved this last result). Our lower bounds for branching programs imply lower bounds on the number of steps that are needed to pebble arbitrary computation graphs for the same computational problems.

To establish our lower bounds we introduce the new concept of a meander that captures superconcentrator-type properties of sequences. We prove lower bounds on the length of meanders via a new Ramsey theoretic lemma that is of interest in its own right. This lemma has other applications, including a tight lower bound on the size of weak superconcentrators of depth 2 that strengthens the known lower bound of Pippenger [PI]. A surprising new feature of these applications of Ramsey theory in lower bound arguments is the fact that no numbers are required to be unusually large and that several of the resulting superlinear lower bounds are in fact optimal.

1. INTRODUCTION
A branching program that computes a Boolean function f of n Boolean variables x1,...,xn is a model of computation that generalizes decision trees. The program is a directed acyclic graph, with a special vertex, that has no ingoing edges, denoted by S, and some other special vertices (sinks), that have no outgoing edges. All non-sink vertices are labeled with an input variable and all sinks are labeled 0 or 1. Every non-sink vertex has fan-out two, and the two edges leaving it are labeled 0 or 1. Each assignment of values b_i to the input variables defines a unique computation path from S to one of the sinks, which starts at S and leaves every non-sink vertex labeled x_i through the edge labeled b_i. The program computes f if f(b1,...,bn) is the label of the end-vertex of this path, for each possible b1,...,bn.

It is customary to assume, (and for most purposes this can be done without loss of generality), that each vertex has a level, where the level of S is 1, and edges go from each level only to the next one. The width of the program is the maximum number of vertices on a level, and its log corresponding is the space of the computation. The length is the number of levels, and it corresponds to the time of the computation. The size is the total number of vertices in the program. Branching programs describe a general sequential model of computation, when we identify the vertices in each level with all the possible internal states of the computational device. It is desirable to
find functions (in P) that cannot be computed simultaneously in linear time and logarithmic space in a general model, i.e., that do not have linear length and polynomial width branching programs. One of the main problems raised by Borodin and Cook, [BC], who proved a time-space trade-off for sorting in this model, is to obtain a result for a one output bit function in P. Here we obtain such a result for input oblivious branching programs.

A program is input oblivious if all non-sink vertices in each level have the same label. Notice that every program can be made input oblivious by increasing its length by a factor of its width. In particular every branching program of bounded width can be assumed to be input oblivious (unless constant factors are important).

A slightly more powerful model of computation than a branching program is the R-way model, introduced by Borodin and Cook in [BC]. Here we compute a function of n variables x1, ..., xn, each being a number between 0 and R-1. Each non-sink vertex is now labeled by one of the x1's, and has R outgoing edges labeled by 0,1,...,R-1. The program branches in this vertex according to the value of x1.

Obviously any function of the considered type can be computed by an R-way input oblivious branching program of length n and width R^n.

One can easily show that almost all Boolean functions cannot be computed by a branching program of subexponential size. It is much more difficult to find functions in P (or even in NP) that require non-linear size. Nechiporuk [Ne] (see also [Sa]) proved an \( n^{c^2/\log n} \) lower bound for the size of any branching program that computes a certain P-function of size n. A barely nonlinear lower bound for the size of the branching programs of the majority function was proved using Ramsey theory by Pudlák [Pu]. All the other nontrivial known lower bounds deal with programs that are restricted in some sense.

The most popular restriction is the case of bounded width branching programs. The main result of [BDPP] and [Ya] is a super-polynomial lower bound for width-2 branching programs that compute majority. Chandra, Furst and Lipton proved a non-linear lower bound for the length of any width branching program that computes the function of n Boolean variables x1,...,xn whose value is 1 iff \( \sum x_i = n/2 \). Their lower bound is very close to linear, being \( n^{O(nW(n))} \), where W(n) is the inverse of van der Waerden numbers. Pudlák [Pu] established an \( \log n \log \ln n \cdot \log \log n \) lower bound for some symmetric functions and Ajtai et. al. [ABHJKRST] obtained an \( n^{\log n/\log \log n} \) lower bound for some other, not so natural, symmetric functions (it is easy to see that their arguments lose only to natural symmetric functions). Very recently, this lower bound has been improved by [BPRS] to \( n^{\log n} \). Our methods (developed independently of both [ABHJKRST] and [BPRS]) enable us to establish a lower bound of \( n^{\log n / w \cdot \log w} \) on the length of arbitrary (not necessarily input oblivious) branching programs of width w for many natural symmetric functions, including all threshold functions \( T_k \) for \( n^0 \leq k \leq n^n \), and including the function \( \sum x_i = n / 2 \) considered in [CFL] ([BPRS] gives an additional lower bound on the size, but only for matching and for some values of w slightly weaker lower bound on the length of branching programs of unbounded width). In particular for branching programs of bounded width we get a lower bound of \( n^{\log n} \) for the previously mentioned symmetric functions. We now present Barrington's recent surprising result [Ba], which asserts that the class of functions computable on branching programs of width 5 and polynomial length coincides with the class of functions that have log-depth polynomial size Boolean circuits (i.e., nonuniform NC^1). It seems difficult to obtain any nontrivial lower bounds for any function in this class (that contains, of course, all symmetric functions).

All the previously known results supply no nontrivial lower bound for the length of programs whose width is, say, n^2. Since the logarithm of the width of the program corresponds to the space of the computation this corresponds to space \( O(\log n) \) and linear time, which is, of course, not so impressive. As mentioned in [BC] it is desirable to have explicit P-functions whose branching programs have nonlinear length, even when the width is greater than \( n \). Here we obtain nonlinear lower bounds for the length of input oblivious R-way branching programs for several NC^1 functions of n bits, even when the width is much greater than \( n \).

Our bounds hold for several functions. Here we specify two examples. The first is the set equality function \( SE(n,m) \). Its input is a sequence of 2n numbers, \( x_1, ..., x_n, y_1, ..., y_n \), each having \( m \) bits, where \( \log \log n \leq m \leq \log n \), i.e., each in the range \( (0,1,...,2^m - 1) \). The function is 1 if and only if for each \( 1 \leq i \leq n \) there is some \( j, 1 \leq j \leq n \) such that \( x_i = y_j \) and vice versa, i.e., iff the two sets \( X = \{x_1, ..., x_n \} \) and \( Y = \{y_1, ..., y_n \} \) coincide (without counting multiplicities). In Section 4 we prove that if the width of an input oblivious R-way branching program computing \( SE(n,m) \) is at most \( 2^{\sqrt{\log n}} \)
then its length is $\Omega(n \log n)$. This result is optimal. Note that the total number of input bits is $m = 10 \log \log n$, so $m = \Omega(n \log n)$.

The second example is the sequence equality function $Q(n)$. Its input is a sequence of $2n$ numbers $x_1, \ldots, x_n, y_1, \ldots, y_n$, each being 0, 1 or 2. The value of the function is 1 if and only if the sequence obtained from $x_1, \ldots, x_n$ by omitting all occurrences of 2 coincides with the one obtained in the same manner from $y_1, y_2, \ldots, y_n$. In Section 4 we show that for any $1 \leq a \leq \frac{1}{2} \log n$, if the width of an input oblivious 3-way branching program computing $Q(n)$ is at most $2^{n/2^a}$, then its length is $\Omega(n^a)$. Thus the length is $\Omega(n)$. This is, in a sense, best possible since obviously any Boolean function of $n$ bits can be computed by an input oblivious branching program of length $n$ and width $2^n$.

All lower bounds on the length of input oblivious branching programs imply lower bounds on the number of steps that are needed to pebble arbitrary computation graphs for the same problem. We will discuss this connection in more detail at the end of Section 4.

Our lower bounds for branching programs follow from a combination of a "cut and paste" argument (i.e., cross sequence argument) with a new Ramsey theoretic lemma, interestng in its own right, which is proved in Section 2. Our use of Ramsey theory differs in two aspects from previous applications of Ramsey theory in lower bound arguments. According to "common knowledge" every application of Ramsey theory in lower bound arguments requires the certain parameters (typically the size of the considered input numbers) have to be "very large". Therefore the derived lower bounds tend to be not applicable to ranges of the input parameters that are of practical interest. Furthermore the derived superlinear lower bounds are typically "barely superlinear" and far away from the best known upper bounds. A surprising feature of the applications of the Ramsey-theoretic Lemma 2.1 in the lower bound arguments of Theorems 3.1, 4.1, 4.2 and 4.3 is the fact that the numbers are not required to be "very large" and that the derived lower bounds are optimal or close to optimal (typically in the $n \log n$-range).

Our Ramsey-theoretic lemma has other interesting applications. In Section 3 we define sequences that are useful in lower bound arguments, called *meanders*, and use our lemma to show that each such sequence must be "long". As an application we prove an optimal lower bound for the size of log $x$-superconcentrators (s.c.) of depth 2.

2. A RAMSEY THEORETIC LEMMA.

Put $N = \{1, 2, \ldots, n\}$ and let $X = (x_1, x_2, \ldots, x_r)$ be a sequence of elements of $N$. For an ordered pair $(a, b)$ of distinct elements of $N$, we define $v_X(a, b)$ to be the binary vector obtained from $X$ by replacing each $a$ by 0 and each $b$ by 1, and by omitting all $x_i \in N - \{a, b\}$. We call $v_X(a, b)$ the order type vector of $(a, b)$ in $X$.

**LEMMA 2.1.** Let $X = (x_1, \ldots, x_r)$ be a sequence in which each $a \in N$ appears precisely $k$ times ($r = n \cdot k$), and suppose $N = N_1 \cup N_2$ is a partition of $N$ into two disjoint sets. Then there are two subsets $S \subseteq N_1, T \subseteq N_2$, $|S| \geq \frac{|N_1|}{2^{2k-1}}$ and $|T| \geq \frac{|N_2|}{2^{2k-1}}$, such that all the order type vectors $(v_X(s, t)) : s \in S, t \in T$ are identical.

**REMARK 2.2.** The assertion of Lemma 2.1 is a Ramsey-theoretic result. It is possible to use some known Ramsey-type results to obtain weaker versions of it. Indeed by considering the complete graph on the elements of $N$ in which the edge $(a, b)$ for $a < b$ is colored by $v_X(a, b)$, one can prove a weak version of Lemma 2.1 by applying the standard Ramsey theorem for graphs (see, e.g., [Bo]). A somewhat better result can be proved using the known results about the problem of Zarankiewicz (see [Bo]). Using these, we can obtain the assertion of Lemma 2.1 for $S, T$ of size $\Omega(\log n/2k)$ (if $|N_1| = |N_2|$). Both results are considerably weaker than the one proved above.

**REMARK 2.3.** Lemma 2.1 is not far from being the best possible. For every $k$ and $n$, we can construct a sequence $X$, in which each $a \in N$ appears precisely $k$ times, with no two disjoint sets $S, T$ of size bigger than $\frac{n}{2} k^{2k-1}$ that satisfy the assertion of the lemma. Indeed, put
\[ i = \lfloor k/2 \rfloor. \] For each \( 1 \leq i \leq 4 \), let 
\[ N_{10}(N_{11}) \] be the sequence of all elements of \( N \) whose \( i \)-th least significant bit is 
0 (1, respectively), ordered in an increasing order. Let \( X \) be the concatenation 
of the following 24 permutations of \( N \): 
\[ (N_{10} N_{11}) \] for \( i = 1, \ldots, 4 \) and \( (N_{11} N_{10}) \) 
for \( i = 1, \ldots, 4 \). We claim that if \( S \subseteq N \) and 
there is even a single \( t \in N - S \) such that all vectors 
\[ \{ v_x(s, t) : s \in S \} \] are 
identical, then \( |S| \leq \lfloor n/2 \rfloor^4 = \lfloor n/2 \rfloor^{2 \times \lfloor k/2 \rfloor} \). 
Indeed, otherwise, we have \( s_0, s_1 \in S \) which differ in the \( i \)-th coordinate for 
some \( 1 \leq i \leq 4 \), and one can easily check that 
\[ v_x(s_0, t) \neq v_x(s_1, t). \]

**Proof of Lemma 2.1:** For \( 1 \leq p \leq k, 1 \leq q \leq k \), 
and an ordered pair \( (a, b) \) of distinct elements of \( N \), let 
\[ v(s, b) \] be the subsequence of \( v_x(a, b) \) consisting of the 
initial p-zeros and initial q 1's in \( v_x(a, b) \). Thus \( v(s, b) \) is the order type 
vector of \( (a, b) \) in the sequence obtained from \( X \) by omitting every occurrence of 
a besides the first \( p \), and every occurrence of \( b \) besides the first \( q \).

We claim that there are two sets \( S_1, S_2 \subseteq N \), 
\( T_1, T_2 \subseteq N \), with \( |S_1|^2 \geq |N_1|/2 \), \( |T_2|^2 \geq |N_2|/2 \) 
such that all vectors \( \{ v(s_1, t_1) : s_1 \in S_1, t_1 \in T_2 \} \) are identical and 
their last two coordinates are distinct. Assume, without loss of generality, 
that each such \( v(s_1, t_1) \) ends with a 1.

If \( p = k \), we are done, since for each 
\( s \in S^{P+q}, t \in T^{P+q} \), \( v_x(s, t) \) is just \( v_x(s_1, t_2) \) followed by \( k - q \) 1's. If \( p < k \), we 
claim that there are \( S_1^{P+q} \subseteq S^{P+q} \) and 
\( T_1^{P+q} \subseteq T^{P+q} \), satisfying \( |S_1^{P+q}| \geq |S^{P+q}|/2 \) and 
\( |T_1^{P+q}| \geq |T^{P+q}|/2 \) such that all vectors \( \{ v(s_1, t_1) : s \in S_1^{P+q}, t \in T_1^{P+q} \} \) are identical, and their last 
two coordinates are distinct. Indeed, put 
\( I = \{ i : x_i \) is the \( (p+1) \)-occurrence \} of some \( s \in S^{P+q} \) and 
\( J = \{ i : x_i \) is the \( q \)-occurrence of some \( t \in T^{P+q} \} \). Clearly 
\( |I| = |S^{P+q}| \) and \( |J| = |T^{P+q}| \). Let \( i \) be 
the \( \lceil |I|/2 \rceil \)-smallest number in \( I \) and 
let \( j \) be the \( \lceil |J|/2 \rceil + 1 \)-smallest 
number in \( J \). If \( i < j \), then we define 
\( S^{P+q+1} = \{ s \in S^{P+q} : \text{the \( p+1 \) occurrence of} \ s \ \text{in} \ X \ \text{is not after} \ x_i \} \), and 
\( T^{P+q+1} = \{ t \in T^{P+q} : \text{the} \ q \ \text{occurrence of} \ t \ \text{in} \ X \ \text{is not before} \ x_j \} \). Clearly, in 
this case, for every \( s \in S^{P+q+1} \) and 
\( t \in T^{P+q+1} \), \( v(s_{i+1}, t_j) \) is equal to the 
vector obtained from \( v(s_i, t_{j-1}) \) by replacing 
it's last coordinate (which is 1) by 01. If \( i \geq j \), we define, similarly, 
\( S^{P+q+1} = \{ s \in S^{P+q} : \text{the} \ (p+1) \ \text{occurrence of} \ s \ \text{in} \ X \ \text{is not before} \ x_i \} \), and 
\( T^{P+q+1} = \{ t \in T^{P+q} : \text{the} \ q \ \text{occurrence of} \ t \ \text{in} \ X \ \text{is not after} \ x_j \} \). In this case, 
for every \( s \in S^{P+q+1} \) and \( t \in T^{P+q+1} \), \( v(s_{i+1}, t_j) \) is \( v(s_i, t_{j-1}) \) followed by a 
zero. In both cases, \( |S^{P+q+1}| \geq |S^{P+q}|/2, \) 
\( |T^{P+q+1}| \geq |T^{P+q}|/2 \) and all vectors 
\( \{ v(s_{i+1}, t_j) : s \in S^{P+q+1}, t \in T^{P+q+1} \} \) are 
identical and their last two coordinates are distinct. This proves our claim and 
completes the proof of Lemma 2.1.

**3. MEANDERS AND WEAK SUPERCONCENTRATORS.** 
For a sequence \( M = x_1 x_2 \ldots x_m \) of numbers 
\( x_i \in \{1, 2, \ldots, n\} = N \) and for disjoint 
sets \( S, T \subseteq N \) we say that an interval 
\( x_1 x_2 \ldots x_j \) is a link between \( S \) and \( T \) 
if \( x_1, x_2, \ldots, x_{j-1} \not\in S U T \) and \( x_j \in S \) or \( x_j \in T \) or \( x_j \in S \) and \( x_j \in T \). The length 
\( M \) (also written \(| M | = M \)) is \( m \). In analogy 
to the definition of a superconcentrator we say that a sequence \( M \) over \( n \) numbers 
is a meander if for any two disjoint sets 
\( S, T \subseteq \{1, \ldots, n\} \) with \( |S| = |T| \) there 
are \( m \) links between \( S \) and \( T \). More generally, for any function 
g : \( N \rightarrow R^+ \), \( M \) is a \( g(x) \)-meander (over \( n \) numbers) 
if for any disjoint sets 
\( S, T \subseteq \{1, \ldots, n\} \) with \( |S| = |T| \) there 
in \( M \) at least \( g(|S|) \) links between \( S \) and \( T \). We call \( M \) a \( g(x) \)-bipartite 
meander if this link property is restricted to sets \( S, T \) with \( S \subseteq \{1, \ldots, n/2\} \) and 
\( T \subseteq \{n/2+1, \ldots, n\} \).

**Theorem 3.1.** If \( M \) is a \( g(x) \)-bipartite 
meander over \( n \) numbers of length \( n \cdot f \), 
then \( f \geq \frac{1}{2} \cdot g(n/2.8^f+1) \). Hence, if \( M \) is 
a \( g(x) \)-meander of length \( n \cdot f \) the same 
inequality holds. In particular, every 
\( \log x \)-meander has length \( \Omega(n \log n) \) and for 
every \( g(x) = \infty \), the length of a \( g(x) \)- 
meander is superlinear.
THEOREM 3.1. Assume \( s(n) \) is some arbitrary function and \( M \) is a sequence over \([1, \ldots, n]\). In order to prove that
\[
|\mathcal{M}(s(n))| = n \cdot f(n)
\]
for some \( k \) such that \( k \leq n/2 \), it is sufficient to show for some \( k \leq n/2 \) that for any two sets \( S \subseteq [1, \ldots, n/2] \) and \( T \subseteq \{n/2 + 1, \ldots, n\} \) of size \( k \) there are \( k \leq n/2 \) links between \( S \) and \( T \).

PROOF: We first observe that if \( k \leq k' \) and \( M \) satisfies the link property of the corollary for sets \( S, T \) of size \( k \), then it also satisfies this link property for sets of size \( k' \). Let \( |M| = n \cdot f(n) \) for some suitable \( f(n) \). Assume \( g(n) + 1 \leq s(n) \) (otherwise we are done).

Let \( g : \mathbb{N} \rightarrow \mathbb{R}^+ \) be any function with
\[
g(n) = s(n) \text{ for some suitable } f(n).
\]
Since \( \frac{g(n)}{s(n)} \leq \frac{n}{2s(n)} \leq \frac{1}{2} \) we know by our observation above that \( M \) satisfies the weaker link property which was actually needed in the proof of Theorem 3.1. Therefore we have \( f(n) \geq \frac{1}{2} \cdot g(n) + 1 \leq \frac{1}{2} \cdot s(n) \), which contradicts our assumption \( g(n) + 1 \leq s(n) \).

Using probabilistic arguments, we next prove the following result, which shows that Theorem 3.1 is sharp for every function \( g(x) \) that satisfies \( \Omega(n \log x) \leq g(n) \leq O(x \log x) \). For each such \( g \), the bound given by Theorem 3.1 for the length of the meander is \( \Theta(n \log n) \).

THEOREM 3.3. For every sufficiently large \( n \), there is an \( \frac{1}{2} x \log n \)-meander \( M_n \) of length \( 3n \log n \). In fact, almost all sequences containing \( 3n \log n \) occurrences of each \( 1 \in [1, \ldots, n] \) are \( \frac{1}{2} x \log n \) (and hence also \( \frac{1}{2} x \log x \))-meanders.

PROOF: Define a function \( g(x) = \frac{1}{2} x \log x \) and let \( M \) be a random sequence in which each \( 1 \in [1, \ldots, n] \) occurs \( \frac{1}{2} n \log n \) times. We show that the probability that \( M \) is a \( g(x) \)-meander tends to 1 as \( n \) tends to infinity.

Fix a number \( s \), \( 1 \leq s \leq n/2 \) and fix two arbitrary disjoint sets \( S, T \subseteq N \) with \( |S| = |T| = s \). An easy combinatorial argument shows that the probability that \( M \) has exactly \( 2j + 1 \) links between \( S \) and \( T \) is
\[
\frac{2 \cdot (3s \log n - 1)^2}{(3s \log n)^2}.
\]
Indeed, the denominator here counts the total number of binary sequences consisting of \( 3s \log n \) 0's and 1's and the numerator counts the number of those sequences with \( j \) blocks of 0's and \( j \) blocks of 1's. As the links between \( S \) and \( T \) depend only on the occurrences of elements from \( S \cup T \) in \( M \), the above expression for the probability the \( 2j + 1 \) links can be given analogously. By a standard estimate, for every \( 1 \leq g(s)/2 \)
\[
(3s \log n)^2 \leq (3s \log n)^2 \leq (42e)^g(s)/2
\]
and hence the probability that there are less than \( g(s) \) links between \( S \) and \( T \) can be bounded by
\[
\frac{2 \cdot (g(s) \cdot (42e)^g(s))}{n^{3s}}.
\]
(Here we used the trivial estimate \( \frac{6s \log n}{(3s \log n)^2} \geq n^{3s} \)). Therefore, the probability that there are two disjoint \( S, T \subseteq N \) with \( |S| = |T| = s \) and with less than \( g(|S|) \) links between them is
\[
\sum_{n/2}^{3n/2} \frac{(en/2)^2}{s(2s)} \cdot 2^s \cdot 2s \log n (42e)^g(s) \leq s \log n
\]
\[
\leq \sum_{s=1}^{\infty} \frac{(en/2)^2}{s(2s)} = \frac{7}{n^3} \log n
\]
\[
\leq \sum_{s=1}^{\infty} \frac{(en/2) \cdot 2s \log n (42e)(\log n)^7}{s^7} \leq S \log n
\]
\[
\leq \sum_{s=1}^{\infty} \frac{(en/2) \cdot 2s \log n (42e)(\log n/7)^s}{s^7} \leq \frac{\log n \cdot n^{6.87/7}}{s^7}
\]

414
which tends to 0 as n tends to infinity.

Recall the definition of a \( g(x) \)-s.c. of depth 2 given in Section 1. The following lemma is due to Pippenger.

**Lemma 3.4.** If there is a \( g(x) \)-superconcentrator \( G \) of depth 2 with \( n \) inputs, \( n \) outputs, and \( e \) edges, then there is a \( g(x) \)-meander \( M \) over \( n \) numbers of length \( e \).

**Proof:** Let \( B_1 : I \rightarrow N \) and \( B_2 : O \rightarrow N \) be two fixed bijections from the set of inputs \( I \) of \( G \) and the set of outputs \( O \) of \( G \) to \( N = \{1, 2, \ldots, n\} \). For any interior vertex \( v \) (i.e., \( v \notin I \cup O \)) of \( G \), let \( I_v \) be the set of inputs adjacent to \( v \) and let \( O_v \) be the set of outputs adjacent to \( v \). Let \( M \) be the concatenation of the sequences \( B_1(I_v), B_2(O_v) \) for all \( v \). One can easily check that if \( G \) is a depth 2 \( g(x) \)-s.c. with \( e \) edges then \( M \) is a \( g(x) \)-meander over \( n \) numbers of length \( e \).

Theorem 3.1 and the last lemma imply that any depth-2 \( \log x \)-s.c. with \( n \) inputs and \( n \) outputs has \( \Omega(n \log n) \) edges. One can easily check that this is optimal (simply take \( \log n \) interior vertices, each adjacent to all inputs and all outputs.

4. **Lower Bounds for Branching Programs.**

Recall the R-way branching programs defined in Section 1. Our lower bounds follow by proving that the labels of the levels in any input oblivious R-way branching program of small width for the considered functions satisfy certain meandertype conditions. The bound follows then from Corollary 3.2. Our first example is the well known set equality function \( SE(n,m) \) defined in Section 1.

One can easily check that a RAM with \( 2^m \) registers can compute this function in time \( O(n) \) (by writing each number \( x_i \) in the register whose address is \( x_i \)), whereas a RAM with \( n^{O(1)} \) registers can solve it in time \( O(n \log n) \) (via sorting). Lower bounds for a RAM with \( n^{O(1)} \) registers and for algebraic computation trees (for the case \( m \gg n \)) appear in [Ma] and [Be], respectively. To the best of our knowledge no lower bound exists for the (realistic) case \( m \ll n \). Here we obtain lower bounds for \( m \ll n \) on R-way input oblivious branching programs.

**Theorem 4.1.** Suppose \( \log \log n \leq m \leq \frac{1}{2} \log n \), \( 1 \leq s \leq \log n \) and \( R = 2^m \). Then any R-way input oblivious branching program of width \( 2^{m/s} \) computing \( SE(n,m) \) has length \( n(s \cdot h(n)) \). This bound is sharp, i.e., for all \( n, m, s \) in this range there is an R-way input oblivious branching program of width \( 2^{m/s} \) and length \( 0(n \cdot s) \) computing \( SE(n,m) \).

**Proof:** The upper bound is straightforward (partition \([0, \ldots, R-1]\) into \( s+1 \) intervals and check separately for each interval which elements of it occur in the input). To prove the lower bound we argue as follows. Let \( m \) be the length of an input oblivious R-way branching program \( B \) computing \( SE(n,m) \), of width \( w \leq 2^{m/s} \).

Let \( M \) be a sequence of length \( m \) over \( \{1, 2, \ldots, 2^m\} \) whose \( i \)-th element is \( j \) if the \( i \)-th level vertices of \( B \) are labeled \( x_j \), and is \( n+j \) if they are labeled \( y_j \). We claim that for any \( S \subseteq \{1, 2, \ldots, n\} \) and \( T \subseteq \{n+1, \ldots, 2n\} \) with \( |S| = |T| = 2^{m-1} \) there are \( \Omega(s) \) links between \( S \) and \( T \) in \( M \). This, together with Corollary 3.2 implies that \( m = \Omega(n \cdot s) \) (for \( s(n) \leq \frac{1}{2} \log n \) we have \( 2^{m-1} \leq n/s \) for all \( n, m, s \) in this range).

Fix sets \( S, T \) as above. Consider inputs \( I_A = \langle z_1, \ldots, z_{2n} \rangle \in SE(n,m) \), where \( z_i = 0 \) for all \( i \notin S \cup T \) and \( A = [z_i : i \in S] = [z_j : j \in T] \), where \( A \) is a set of cardinality at most \( |S| = 2^{m-1} \) of elements from \( \{1, \ldots, 2^m\} \). Let \( L \) be the set of links between \( S \) and \( T \) in \( M \). A standard "cut and paste" argument (= "crossing sequence" argument) implies that for any two inputs \( I_A \) and \( I_A' = \langle z_1', \ldots, z_{2n} \rangle \) with \( A \neq A' \) there is a link \( \ell \) in \( L \), such that the computation path in \( B \) for \( I_A \) differs from that of \( I_A' \) on that level of the branching program \( B \) that corresponds to the last element of the link. Otherwise \( B \) would also accept an amalgamated input \( I = \langle z_1, \ldots, z_{2n} \rangle \notin SE(n,m) \) given by \( z_i = z_i' \) for \( i \leq n \) and \( z_i = z_i' \) for \( 2^{m-1} \leq i > n \).

There are \( \sum_{i=0}^{2^{m-1}} \binom{2^{m-1}}{i} \geq 2^{2m-2} \) different choices for \( A \) and thus \( |L| \geq 2^{2m-2} \). Since \( w \leq 2^{m/s} \) this implies that \( |L| = \Omega(s) \).

Our second example is the sequence equality function \( Q(n) \) defined in Section 1.

**Theorem 4.2.** Any (3-way) input oblivious branching program of width \( 2^{n/\log n} \) computing \( Q(n) \) has length \( N(n \cdot h(n)) \). In particular, if the width is \( 2^{o(n)} \) then the length is superlinear.

415
PROOF: The proof is similar to the previous one. Let $B$ be an input oblivious 3-way branching program for $Q(n)$ of length $m$ and width $w \leq 2^{n/2}$, Let $M$ be a sequence of length $m$ over $\{1,2,\ldots,2n\}$ whose $i$-th element is $j$ if the $i$-th level vertices of $B$ are labeled $x_j$ and is $n+j$ if they are labeled $y_j$.

Set $s := h/2$ and suppose $S \subseteq \{1,\ldots,n\}$ and $T \subseteq \{n+1,\ldots,2n\}$ satisfy $|S| = |T| = 2n/2^s$. By Corollary 3.2 it is sufficient to show that there are in $M$ at least $s$ links between $S$ and $T$. To bound the number $\lambda$ of links between $S$ and $T$ one considers inputs $I_A = \langle z_1, \ldots, z_{2n} \rangle$, where $z_i = 2$ for $i \not \in \text{S} \cup \text{T}$ and $A$ is a binary sequence of length $|S|$ which coincides with the two sequences $\langle z_i \rangle_{i \in S}$ and $\langle z_j \rangle_{j \in T}$. The standard crossing sequence argument implies that $w^4 \geq 2 |S| = 2^{2n/2^s}$, i.e., $\lambda \geq \frac{2n}{2^s} \geq 2^{s+1} \geq s$. $$\blacksquare$$

Finally we consider lower bounds for some symmetric functions.

**Theorem 4.1**. Let $T_k = T_k(x_1,\ldots,x_n)$ be the Boolean function of $n$ variables whose value is 1 if and only if $\sum x_i \geq k$.

Fix any constant $\delta > 0$. Then any input oblivious branching program of width $w$ that computes $T_k$ for some $k$ with $n^6 \leq k \leq n - n^6$ has length $\Omega(n \log n \log \log w)$.

**Proof**: The proof is similar to the two previous ones. For each pair of disjoint subsets $S$ and $T$ of $\{1,\ldots,n\}$ of size $\delta^6$ we consider for each $i \leq n^6$ an input $I_i$ that has $k - n^6$ many 1's on some fixed subset of $\{x_j \mid j \not \in \text{S} \cup \text{T}\}$, many 1's in $\{x_j \mid j \in \text{S}\}$ and $n^5$ many 1's in $\{x_j \mid j \in \text{T}\}$. These inputs will show via a crossing sequence argument that any input oblivious branching program of width $w$ for $T_k$ the number $\lambda$ of links between $S$ and $T$ satisfies $w^4 \geq n$, thus $\lambda \geq \frac{n}{\log n} \log \text{log} \text{log} w$. Therefore we can apply Corollary 3.2 with $s := \delta \cdot \log n / \log \text{log} \text{log} w$ (we have $n/2^s \geq n/n^6 \geq n^6 = |S|$).

Analogous results for other symmetric functions can be proved similarly. In particular, we get an $\Omega(n \log n / \log \text{log} w)$ bound for the function $f$ (considered in [CFL]) of $n$ Boolean variables $x_1,\ldots,x_n$ whose value is 1 if $\sum x_i = n/2$. It is not too difficult to show that this is sharp. Indeed, for say $w = \Theta(\log n)$ one can compute $f$ in length $\Theta(n \log n / \log \log w)$ by computing $\sum x_i$ modulo each prime $p$ satisfying $p < \log n$ and by using the Chinese Remainder Theorem. Similarly, the above bound for this function can be shown to be sharp for all $\log n \leq w \leq n$.

Finally we would like to point out that our lower bounds on the length of input oblivious branching programs imply lower bounds on the number of steps that are needed to pebble any computation graph for the same problem. Assume $G$ is a computation graph for a function $f(x_1,\ldots,x_n)$ where all arguments and intermediate results that are computed at nodes of $G$ are from $\{0,\ldots,R-1\}$. Then any pebbling of $G$ with $p$ pebbles in $T$ steps defines an input oblivious branching program for $f$ of width $R^p$ and length $T$.

**Acknowledgement**

We would like to thank A. Borodin, N. Pippenger and Gyorgy Turan for fruitful discussions.

**References**


