ON NOTIONS OF INFORMATION TRANSFER IN VLSI CIRCUITS

Alfred V. Aho
Bell Laboratories
Murray Hill, New Jersey

and

Jeffrey D. Ullman†
Stanford University
Stanford, California

and

Mihalis Yannakakis
Bell Laboratories
Murray Hill, New Jersey

I. Introduction

Several papers have recently dealt with techniques for proving area-time lower bounds for VLSI computation by "crossing sequence" methods. Yao [Y1] talks about the minimum information transfer being a lower bound on the number of crossing sequences across some boundary on the chip. Lipton and Sedgewick [LS] and Savage [S] talk about "fooling sets," which are particular sets of input assignments that can be used to prove lower bounds on the minimum information transfer needed. A number of natural questions are raised by these definitions.

1. *Is the fooling set approach the most powerful way to get information-transfer-based lower bounds?* We shall show it is not, and offer a candidate for the title "most powerful." Of course, without a precise definition of "information transfer argument," there could be other contenders.
2. *Are the notions of the three papers cited equivalent?* We shall exhibit certain inequivalences among the three notions, although open questions remain. However, we can resolve an open question of Papadimitriou and Sipser [PS] concerning the relationship between nondeterministic and deterministic communication complexity.

† Work partially supported by NSF grant MCS-82-03405 and ARPA contract MDA 903-80-C-0107.

II. Problems, Fooling Sets, and Adversary Arguments

A *problem* is a list $X = x_1, \ldots, x_n$ of boolean input variables and a list $Y = y_1, \ldots, y_m$ of boolean output variables, together with boolean formulas expressing the y's in terms of the x's. Methods have been developed in the references cited and elsewhere to prove that certain problems can only be solved on an integrated circuit of area A and time T if AT^2 is at least a certain function of n, usually $\Omega(n^2)$. Thompson [T], Vuillemin [V], and Kedem [K] present a sequence of advances in this direction, along with [LS] and [Y1]. The basic idea is to identify a notion of "information," which has the property that if a problem has information content I, then there must be at least 2^I distinct crossing sequences at some boundary within the chip, and therefore, the time multiplied by the length of the shorter side of the chip is at least I, from which $AT^2 \geq I^2$ follows. The typical proof can be viewed as an adversary argument with the outline of Fig. 1.

Step (1) models the idea that inputs and outputs occur at fixed locations on the chip, and we can select a boundary across the shorter dimension of the chip. Since we are proving a lower bound, the adversary is the chip designer, who gets to say on which side of a boundary the ports go. The "constraints" mentioned in (1) are typically that no more than 2/3 of the inputs, no more than 2/3 of the outputs, or no more than 2/3 of the total go on one side of the boundary. Our ability to constrain the adversary this way comes from the fact that we can select the boundary after the chip is designed. To argue that a boundary that divides the inputs into roughly equal-sized parts can be found, we first rule out the possibility that very many inputs occur at one point (if they did, then T would have to be large), and then

 $00.75

	Adversary	Lower Bound Prover
(1)	Pick partition of X and Y subject to some constraints.	
(2)		Select fooling set of 2^l input assignments.
(3)	Select two members of fooling set to have the same crossing sequence.	
(4)		Win by showing that from the fact the two inputs have the same crossing sequence, there must be some other input on which the chip makes the wrong output.

Fig. 1. Standard adversary argument.

select a straight line with a single jog if necessary, as in Fig. 2.

In step (2), we evidently have to select the input assignments so that if any two yield the same crossing sequence, then the input assignment we get by combining the portion of one to the left of the boundary with the portion of the other to the right will produce an erroneous output. This idea can be formalized as follows.

Let $P = (X, Y)$ be a problem, and suppose the input set X is partitioned into X_L and X_R, while the output set is partitioned into Y_L and Y_R. An *input assignment* α is a mapping from X to $\{0, 1\}$. We use α_L and α_R for α restricted to X_L and X_R, respectively, and we use $\alpha_L\beta_R$ for the input assignment that agrees with α on X_L and with β on X_R. A set $A = \{ \alpha_1, \ldots, \alpha_k \}$ of input assignments is said to be a *two-way fooling set* for this partition if for any α and β in A, one of the following four conditions holds.

1. $\alpha_L\beta_R$ disagrees with α on some bit of Y_L.
2. $\alpha_L\beta_R$ disagrees with β on some bit of Y_R.
3. $\beta_L\alpha_R$ disagrees with α on some bit of Y_R.
4. $\beta_L\alpha_R$ disagrees with β on some bit of Y_L.

The argument in each case is the same. In (1), for example, we claim that if α and β have the same crossing sequence, then the left side of the chip will give the same response to α and $\alpha_L\beta_R$, and therefore, all bits of Y_L will assume the same value. At least one of those bits is incorrect for one of α or $\alpha_L\beta_R$.

Let us define $I_{2FS}(P, \pi)$, where π is a partition of the inputs and outputs for problem P, to be the logarithm, base 2, of the size of the largest two-way fooling set for π. We use $I_{2FS}(P)$ for the minimum over all "legal" partitions π, of $I_{2FS}(P, \pi)$. While the definition of "legal" could vary, here we take it to mean that neither X_L nor X_R are larger than 2/3 the size of X. We also drop the problem name P when it is obvious, and speak of I_{2FS}, the *two-way fooling set information*.

Another kind of fooling set plays an important role in the theory. A *left-going fooling set* is a pair (A, β_R), where β_R is an assignment to the right side inputs X_R, and $A = \{ \alpha_{1L}, \ldots, \alpha_{kL} \}$ is a list of assignments to X_L, such that for all $i \neq j$, the input assignments $\alpha_{iL}\beta_R$ and $\alpha_{jL}\beta_R$ differ on some bit of Y_R. A *right-going fooling set* can be defined analogously, with left and right interchanged.

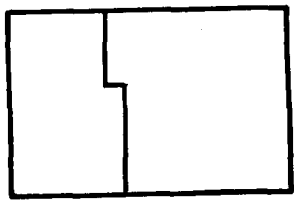

Fig. 2. Boundary selected in a chip.

For problem P and partition of its inputs and outputs π, define $I_{1FS}(P, \pi)$ to be the logarithm, base 2, of the size of the largest list A in a left-going or right-going fooling set for P and π. Let $I_{1FS}(P)$, or just I_{1FS}, be the minimum over all legal partitions π, of $I_{1FS}(P, \pi)$; we call this value the *one-way fooling set information* for P.

The following relationships are easy to prove.

Theorem 1. $I_{1FS}(P, \pi) \leq I_{2FS}(P, \pi)$ and $I_{1FS} \leq I_{2FS}$. □

It might therefore appear that there is no point in considering I_{1FS}. However, the following theorem shows that they do play a role, when we consider probabilistic algorithms (in the sense that there are random number generators on the chip, and the output is correct with probability greater than 1/2 on any run).

Theorem 2. If problem P is solved by a probabilistic chip of area A and time T, then $AT^2 \geq \Omega(I_{1FS}(P)^2)$.

Proof. The proof is a generalization of the proof in [LS] that probabilism is no help in transmitting strings. □

Theorem 3. There is a problem P for which $I_{1FS}(P) < I_{2FS}(P)$.

Proof. Let P be any hard predicate, such as one of those proved in [LS]. $I_{1FS}(P) \leq 1$, because a one-way fooling set cannot exceed size 2 if there is only one output bit. □

III. Better Adversary Arguments

We now take up the question of what is the most powerful way to prove lower bounds on AT^2. One direction in which we could improve the fooling set arguments has already been considered by Yao [Y1, Y2]. There, lower bounds on the number of required crossing sequences are obtained by arguing about the number of bits that must be exchanged by two sides of a chip. We shall consider this approach in Sections IV and V, but we do not regard Yao's methods as comparable to fooling set arguments because Yao's definition is significantly less constructive (of a lower bound proof) than fooling set arguments.

More importantly, there is a strategic issue that none of the references cited addresses. The standard fooling set or information-theoretic argument gives away too much to the adversary. In particular, what is really going on is that the adversary designs the whole chip, after which the lower

bound prover is privileged to focus on any boundary that he chooses. The prover must respond to any design, but not to every possible legal partition; perhaps there is some other partition on the same chip that yields a better lower bound. Figure 3 summarizes what we believe is a more powerful form of adversary argument for VLSI lower bounds.

	Adversary	Lower Bound Prover
(1)	Pick design of the chip.	
(2a)		Pick a partition of input and output bits that can be drawn with a boundary of limited length on the chip.
(2b)		Select a fooling set of 2^I input assignments for this partition.
(3)	Same as Fig. 1.	
(4)		Same as Fig. 1.

Fig. 3. More powerful adversary argument.

The Linear Argument

One simple way to implement the idea of Fig. 3 is to realize that the points of the chip can be ordered, column by column (assuming the chip is wider than high), with columns taken from the left, and each column taken from top to bottom. Associated with each point is a set of input and output bits that are read or written at that point.

We can therefore represent the layout of the chip by a list of sets of input and output bits. We assume that no one set contains more than 1/3 of the input bits, or else we must use another argument to get a lower bound on time alone, as for the previous methods. An example of such a list is $\{ x_1, x_5, y_2 \}, \{ x_2, x_6 \}, \{ x_3 \}, \{ y_1 \}, \{ x_2, x_4, y_3 \}$.

The legal moves for the prover in step (2a) of Fig. 3 are selections of a cut in the list; e.g., he might choose to put the cut between the first and second sets above, so $X_L = \{x_1, x_5\}$, $X_R = \{ x_2, x_3, x_4, x_6 \}$, $Y_L = \{ y_2 \}$, and $Y_R = \{ y_1, y_3 \}$. Each such cut in the list corresponds to a boundary in the chip with one jog, across the shorter dimension, as in Fig. 2; the reason is that we may view the list as representing the column-by-column ordering discussed above.

Let us define $I_{LFS}(P)$ to be the minimum, over all linear orderings, of the logarithm of the maximum possible fooling set size for any partition of that ordering. Then:

Theorem 4. There are problems P for which $I_{2FS}(P)$ is strictly less than $I_{LFS}(P)$.

Proof. Let P_1 and P_2 be two instances, each with n inputs, of some problem such as cyclic shift for which $I_{2FS} = \Omega(n)$ [V]. Consider the problem P that is the union of P_1 and P_2. $I_{2FS}(P) = 0$, because of the partition where all the bits of P_1 are on the left and all those of P_2 are on the right.

However, we can use the linear information argument to show an $\Omega(n)$ lower bound. Given any linear ordering, ignore the bits from P_2, and select a cut that divides the input bits of P_1 with at least $n/6$ on either side, which we may do because $(2n)/3$ is an upper bound on the number of input bits of P in any set of the list. The lower bound argument then proceeds as for P_1. □

The Ultimate Argument

In what appears to be the most general use of the argument of Fig. 3, we require the adversary to lay out the entire chip. We, in step (2a), pick an arbitrary boundary of length l to divide the chip into "left" and "right" halves, which may not even be connected "halves."

The information $I_U(P)$ for a given problem P is the minimum over all layouts of the maximum over all boundaries B, of the minimum amount of information in the sense of [Y1] that must flow across the boundary, divided by the length (in units of $\sqrt{A}$) of the boundary B, where A is the area of the chip.

The relationship between I_{LFS} and I_U is open, although undoubtedly the inequality is proper. However, we can make the following observation about the boundaries used in proofs involving I_U.

Theorem 5. $I_U(P)$ is not diminished if we restrict ourselves to boundaries that divide the chip into two connected regions. □

IV. Information-Theoretic Arguments

As mentioned, Yao [Y1] has considered the number of bits that must flow across boundaries as a lower bound on the number of crossing sequences required. Since [Y1] dealt only with single-bit outputs, we must make some changes in detail, but not in spirit, to his definitions.

We say problem P has *one-way information* $I_1(P)$ if that is a lower bound over all "legal" partitions on the minimum number of bits we must transmit from one side to the other, so that the second side will know the correct value for each output bit, regardless of the side on which the bit appears. As before, we take "legal" to mean at least one third of the input bits are on either side, but other definitions of "legal" might be considered.

We say P has *two-way information* $I_2(P)$ if that is a lower bound over all "legal" partitions on the minimum number of bits that must be transmitted in either direction, so that each side knows the value of each output bit on that side. Note that the sequence of bit transmissions can include transmission in both directions, and can be adaptive, depending on the bits sent previously.

Similarly, we can define $I_L(P)$ to be the analog of I_{LFS}, but with general information arguments permitted, rather than being restricted to a fooling set argument. The following is obvious.

Theorem 6. $I_{2FS}(P) \leq I_2(P)$, $I_{1FS}(P) \leq I_1(P)$, and $I_{LFS}(P) \leq I_L(P)$. □

It is expected that the inequalities are proper. For the one-way case, at least, we have.

Theorem 7. The ratio $I_1(P)/I_{1FS}(P)$ can be as high as $\Omega(n)$.

Proof. As in Theorem 3, let P be any hard predicate. Then $I_{1FS} \leq 1$, but $I_1 \geq I_2 = \Omega(n)$. □

Nondeterministic Complexity

The notion of computation on a chip has been extended naturally to nondeterministic computation of predicates [LS, MS, PS]. If L is a language, we may define $I_{2N}(L)$ to be the minimum over all legal partitions of the number of bits that must cross the boundary in a nondeterministic computation accepting L, and we may define $I_{2co\text{-}N}(L)$ to be the same for the complement of L.

Papadimitriou and Sipser showed that the language L = "graph given by its adjacency matrix has a triangle" has $I_2 = I_{2co\text{-}N} = \Omega(n)$, but $I_{2N} = O(\log n)$. They asked whether there is a language with $I_2 = \Omega(n)$, but I_{2N} and $I_{2co\text{-}N}$ both $O(\log n)$. The question is important because it exposes something about the power of lower bound arguments, just as the results in the present paper attempt to do. That is, fooling set, and similar crossing-sequence arguments apply to nondeterministic computation as well as deterministic. We can put a lower bound on the deterministic information required for a predicate by arguing about either that problem or its complement, but if both the problem and its complement have low nondeterministic complexity, other techniques are needed.

Theorem 8. There is a predicate P with $I_{2N}(P) = I_{2co\text{-}N}(P) = O(\log n)$, but $I_2 = \Omega(n)$.

Proof (sketch). Let Q be the predicate "has a triangle" from [PS]. Suppose we divide the nodes of a graph into two equal-sized sets V_1 and V_2, and suppose that the input bits to Q are ordered so that the edges between V_1 and V_2 follow the other edges. Let m be the size of V_1 and V_2, $n = \binom{2m}{2}$. Let $P(abcd)$, where a and c are strings of $2\binom{m}{2}$ bits representing the edges within V_1 and within V_2, and b and d are strings of m^2 bits representing edges between V_1 and V_2. Note that these bits do not all represent one graph of $2m$ nodes. Then the predicate P is

$$P(abcd) = Q(ab) \textbf{ and not } Q(cb) \textbf{ and } Q(cd) \textbf{ and not } Q(ad)$$

For I_{2N} choose a partition π that places a and d in one side and b and c in the other side. With this partition $I_{2N}(P,\pi) = O(\log n)$. For I_{2co-N} choose a partition σ that places a and b in one side, and c and d in the other side; $I_{2co-N}(P,\sigma) = O(\log n)$. It can be shown, however, that for any "legal" partition τ, either $I_{2N}(P,\tau)$ or $I_{2co-N}(P,\tau)$ is $\Omega(n)$, and consequently $I_2(P) = \Omega(n)$. □

We note that the usual fooling set argument suffices to prove that $I_2(P) = \Omega(n)$ in Theorem 8. The proof does not introduce a new technique but rather exposes a weakness of the partition argument (here with respect to nondeterministic computations) in a way similar to Theorem 4. So, for example, if we let I_{LN}, I_{Lco-N} be the analogs of I_{2N}, I_{2co-N} with the linear argument in place of the partition argument, then the predicate P of Theorem 8 has $I_{LN}(P) = I_{Lco-N}(P) = \Omega(n)$.

V. Lower Bound Techniques for a Fixed Partition

Regardless of the type of the argument used to prove a lower bound on AT^2 (partition, linear or the ultimate), we need techniques for proving lower bounds on the information transfer for a given partition (step 2 in Fig. 1, step 2b in Fig. 3). In this section we will examine the power of such techniques. For simplicity we will restrict our discussion to the case that both sides of the partition output the same function; the results can be extended to the general case where we have different left and right outputs.

Fooling sets is one convenient lower bound technique. It rests on the following two properties of crossing sequences. (1) If two input assignments $\alpha = \alpha_L\alpha_R$ and $\beta = \beta_L\beta_R$ have the same crossing sequence, then also the input assignments $\alpha_L\beta_R$ and $\beta_L\alpha_R$ must have the same crossing sequence with them, and (2) the output on the left (respectively right) side depends only on the crossing sequence and the left (resp. right) part of the input.

In [Y2] we see proposed another technique which rests also on the same properties. Let f be the function which is computed on both sides. Consider a matrix $M(f)$ whose rows correspond to the possible left input assignments, the columns correspond to the possible right input assignments, and whose (α,β) entry is $f(\alpha\beta)$. A *rectangle* is the cross product $A \times B$ of a set A of left input assignments with a set B of right input assignments. Fix a protocol that computes the function f on both sides. From property (1) of crossing sequences, the set of inputs which have the same crossing sequence is a rectangle.

A rectangle is said to be *monochromatic* if f has the same value for every input in the rectangle. From property (2), the rectangle that corresponds to each crossing sequence is monochromatic. (This need not be the case if the two sides output different functions; in that case, in a rectangle that corresponds to a crossing sequence, the left output is constant across each row and the right output is constant across each column.) Each pair (α,β) of left and right input assignments has exactly one crossing sequence. Therefore, the monochromatic rectangles that correspond to the crossing sequences are disjoint and cover the matrix $M(f)$.

Define the *tiling complexity* of f, $I_T(f)$, to be the logarithm of the minimum number of disjoint monochromatic rectangles that cover the matrix of f. Then, $I_2 \geq I_T$ [Y2]. Yao showed also a bound in the opposite direction: $I_2 = O(2^{I_T/2})$; we will prove later that there is no exponential gap between I_2 and I_T.

From the definition of a fooling set, it follows that no two elements of a fooling set belong to the same monochromatic rectangle. Therefore, $I_T \geq I_{2FS}$. As Lipton and Sedgewick observed in [LS], the fooling set technique applies also to nondeterministic protocols computing the function f. A *nondeterministic protocol* is one in which the two sides can take guesses in the course of the computation. It *computes* a function f on both sides if for every pair (α,β) of left and right input assignments, (i) there is a sequence of guesses for which both sides output $f(\alpha\beta)$† and (ii) all sequences of guesses that lead to the production of outputs give the correct outputs. (It is not necessary to have outputs for every sequence of guesses.) We let $I_{2N}(f)$ denote the minimum communication complexity of a nondeterministic protocol

† One may give a slightly different definition of a nondeterministic protocol which does not require the two sides to output the value of the function for the same sequence of guesses. It is not hard to see that this definition can decrease the communication complexity by at most a factor of 2. However, as is customary, we will ignore constant factors.

computing f on both sides. Note the difference between a nondeterministic protocol *accepting* a language L and one computing the characteristic function f of L: In the case of an accepting protocol an output has to be produced only if the input is in the language, whereas in the case of a protocol computing f an output has to be produced for every input. In terms of the communication complexity, we have $I_{2N}(f) \simeq \max\{I_{2N}(L), I_{2co-N}(L)\}$. Every sequence of guesses in a nondeterministic protocol leads to a crossing sequence. As in the deterministic case, the set of inputs which have the same crossing sequence and which output the value of f on this crossing sequence form a monochromatic rectangle. Therefore, a nondeterministic protocol determines also a set of monochromatic rectangles which cover the matrix of f; the rectangles need not be disjoint however. Since no two elements of a fooling set belong to the same monochromatic rectangle, $I_{2N}(f) \geq I_{2FS}(f)$.

How good a bound on I_{2N} is I_{2FS}? F. Chung (private communication) can show that in a random $m \times m$ Boolean matrix the largest fooling set has size $O(\log m)$ and thus $I_{2FS} = O(\log \log m)$; however, the largest monochromatic rectangle has size $O(m \log m)$ which implies that at least $O(m/\log m)$ monochromatic rectangles are needed to cover the matrix and thus $I_{2N} = \Omega(\log m)$. The gap between I_{2FS} and I_{2N} (in fact between I_{2FS} and I_2, I_1) is never more than exponential.

Theorem 9. Let *nrow* be the number of different rows of the matrix $M(f)$ and *ncol* the number of different columns. $I_{2FS} \geq \log \log \max\{nrow, ncol\}$. □

The quantities $\log nrow$, $\log ncol$ are equal to the complexity of one-way communication from the left to the right side and from the right to the left respectively [Y2]. Thus, $I_{2FS} \geq \log I_1 \geq \log I_2$. Nondeterministic complexity is never that far however from deterministic complexity.

Theorem 10. $I_2 = O(I_{2N}^2)$.

Proof. An optimal nondeterministic protocol determines a covering of the matrix $M(f)$ with a set V of monochromatic rectangles (corresponding to its crossing sequences) with $I_{2N} \geq \log|V|$. Form two graphs G_L, G_R with the set V of rectangles as nodes. There is an edge in G_L (resp. G_R) between two rectangles if they have a row (resp. column) in common. Let $deg_L(u)$, $deg_R(u)$ be the degree of a node u in G_L, resp. G_R.

Suppose that the left side receives an input α and the right side an input β. The deterministic protocol works in stages. In each stage we have sets A, B of left, right input assignments such that the left side knows (from the communication that has already taken place) that $\beta \in B$ and the right side knows that $\alpha \in A$. Initially, A and B are the sets of all possible left and right input assignments. We have also a covering of $A \times B$ with a set V of monochromatic rectangles and the two corresponding graphs G_L, G_R known to both sides.

A stage is carried out as follows. The left side looks for a rectangle $u = A_u \times B_u$ in V such that $\alpha \in A_u$ and $deg_L(u) \leq 3|V|/4$. If it finds such a rectangle u, it sends u to the right side; otherwise it communicates that there is no such rectangle. In the first case the right side communicates whether $\beta \in B_u$. If $\beta \in B_u$, then the protocol finishes: both sides know that $f(\alpha\beta)$ is the (constant) value of f in the rectangle u. If $\beta \notin B_u$, then A is replaced by $A \cap A_u$, B remains the same, V is replaced by the set of nonempty rectangles in $\{(A_v \cap A_u) \times B_v \mid v = A_v \times B_v \in V\}$, and the stage finishes. Clearly, $A_v \cap A_u \neq \emptyset$ iff v is adjacent to u in G_L. Thus, the size of the new set V is at most 3/4 of the old size. If the left side does not find an appropriate rectangle u, then the right side looks for a rectangle $w = A_w \times B_w$ in V with $\beta \in B_w$ and $deg_R(w) \leq 3|V|/4$. If it finds such a w it sends it to the left side; otherwise it communicates that there is no such w. In the first case the left side acts symmetrically. That is, either $\alpha \in A_w$ in which case the protocol finishes, or $\alpha \notin A_w$ in whish case the stage finishes with the sets A, B, V updated.

In the second case, when the right side does not find an appropriate w, we claim that both sides have enough information to determine the value of f, and thus, the protocol finishes. Both sides know that every rectangle u containing α has $deg_L(u) > 3|V|/4$, and every rectangle w containing β has $deg_R(w) > 3|V|/4$. Therefore, every rectangle covering (α,β) has degree $> 3|V|/4$ in both graphs G_L, G_R, and thus has degree $> |V|/2$ in their intersection $G = G_L \cap G_R$. We claim now that the value of f is the same in all rectangles with this property. Let x, y be two rectangles with degree $> |V|/2$ in G. Then, there is a node z which is adjacent to both x and y in G. But two rectangles are adjacent in G iff they have a nonempty intersection. Since the rectangles are monochromatic it follows that x, y and z have the same value of f.

Therefore, the protocol computes correctly f and finishes in at most $\log|V|/\log(4/3)$ stages with a total communication of (roughly) $I_{2N}^2/\log(4/3)$. With a slightly more careful protocol, the coefficient $1/\log(4/3)$ can be replaced by 2. □

Corollary. $I_2 = O(I_T^2)$. □

The transitivity property used to infer the value of f in the last step of the proof of Theorem 10 does not hold in the case that the two sides output different functions. However, the theorem can still be proved (proof omitted). Of course, the same quadratic relationship holds also when the information transfer is calculated with respect to the best partition as in Section IV. Thus, for example, for any language L we have, $I_2(L) = \min_{\pi} I_2(L,\pi)$ $= O([\min_{\pi} \max\{I_{2N}(L,\pi), I_{2co-N}(L,\pi)\}]^2)$ (compare with Theorem 8). The bound is tight up to a logarithmic factor: Melhorn and Schmidt present in [MS] a Boolean function with $I_2 = n$ and $I_{2N} = O(\sqrt{n}\log n)$. The technique they use to prove the bound on I_2 for this function is based on the rank in $GF(2)$ of the matrix. The minimum size of a tiling with disjoint monochromatic rectangles is bounded from below by the rank of the matrix (in any field) [MS, O]. If we let I_R be the logarithm of the rank (in some field) of the matrix, then $I_T \geq I_R$. The rank technique can be used to show bounds that do not apply to nondeterministic protocols. In general however, an exponential gap between I_2 and I_R is possible.

Theorem 11. There is a function f with $I_R(f) = \log n$ and $I_2(f) = I_{2N}(f) = n$.

Proof. For the rank in $GF(2)$ take f to be the inner product mod 2 of two n-dimensional 0-1 vectors. The rank of the matrix in $GF(2)$ is n, and $I_R = \log n$. It is not hard to see that the largest monochromatic rectangle has size 2^n. Since the matrix has size 2^{2n}, at least 2^n monochromatic rectangles are needed to cover it, and therefore $I_2 = I_{2N} = n$. For the rank in the ring of integers, take f to be the inner product without the mod 2 reduction. □

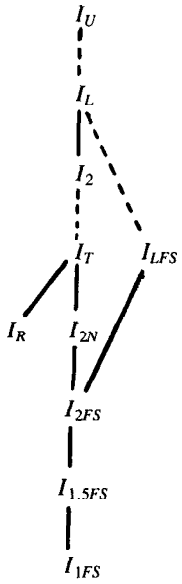

Fig. 4. Diagram of inclusions.

We have seen thus three lower bound techniques: I_{2FS}, I_R, I_T with the last one always larger than the other two. The first two are more constructive in general, in the sense than one can exhibit a large fooling set or can use linear algebra methods to prove that the rank in $GF(2)$ (or another field) is large; they may be in general, however, exponentially far from I_2. The tiling method comes within a square root of I_2 but is less constructive: exhibiting a tiling does not provide a bound on I_T (a difficult *minimization* problem). Still, in some cases, it might be possible (and easier) to argue about I_T. (For example, what is a large fooling set for the function of Theorem 11?)

Open Problem. Find a "constructive" lower bound technique that comes always within a constant factor of I_2.

VI. Asymmetric Fooling Set Rules

One might suppose that the definition of two-way fooling set given in Section II was redundant, and that we could do with only the first two clauses. Put another way, our definition of *2FS* implies that the adversary gives us a set $\{ \alpha, \beta \}$ of inputs, and we can derive a contradiction in any way we can. Could we not allow the adversary to give us an ordered pair (α, β), and derive a contradiction involving $\alpha_L\beta_R$, that is, using rule (1) or rule (2) in the definition of *2FS*?

For want of a better term, let us define $I_{1.5FS}(P, \pi)$ to be the logarithm of the size of the largest set of input assignments that for each pair satisfies clause (1) or clause (2) in the definition of two-way fooling set. Let $I_{1.5FS}(P)$ be the minimum over all "legal" π of $I_{1.5FS}(P, \pi)$.

Theorem 12. $I_{1FS}(P) \leq I_{1.5FS}(P) \leq I_{2FS}(P)$, and at least the first inequality is proper for some P.

Proof. The argument for Theorem 3 applies here too. □

At least for a fixed partition the second inequality is also proper.

Theorem 13. There is a problem P and an input partition π such that the ratio $I_{2FS}(P, \pi)/I_{1.5FS}(P, \pi)$ is $\Omega(n)$.

Proof. Consider the problem of determining of input bits $x_1, \ldots, x_n$ whether u, the numerical value of $x_1, \ldots, x_{n/2}$ is equal to or less than v, the numerical value of $x_{n/2+1}, \ldots, x_n$. Let π place $x_1, \ldots, x_{n/2}$ on the left and the other bits on the right. Then $\{ (0, 1), (1, 2), \ldots, (2^n-2, 2^n-1) \}$ is a two-way fooling set for P and π, where (a, b) means the input assignment that assigns values to the inputs so that $u=a$ and $v=b$. Thus, $I_{2FS}(P, \pi) = \Omega(n)$.

However, it is easy to show that no 1.5-way fooling set can have more than two members. Suppose (a, b), (c, d), and (e, f) were in one fooling set, and assume without loss of generality that $a \leq b$ and $c \leq d$. Also, assume $b \leq d$ and let the lone output bit appear on the right. If $\alpha = (a, b)$ and $\beta = (c, d)$, then $\alpha_L\beta_R = (a, d)$ agrees with α on the left, because there are no output bits there, and agrees with β on the right because $a \leq d$. □

VII. Summary

Figure 4 diagrams the relationships we have proved, along with some other obvious relationships. Solid lines show proper containment; dashed lines show (possibly improper) containment, although we have taken the liberty of assuming that a proof of proper containment for a fixed partition is adequate justification to distinguish between two information measures. I_{2N} in the figure stands for the complexity of nondeterministic protocols computing the function.

References

[K] Kedem, Z. M., "Optimal allocation of computational resources in VLSI," *Proc. Twenty-Third Annual IEEE Symposium on Foundations of Computer Science*, pp. 379-386, 1982.

[LS] Lipton, R. J. and R. Sedgewick, "Lower bounds for VLSI," *Proc. Thirteenth Annual ACM Symposium on the Theory of Computing*, pp. 300-307, 1981.

[MS] Melhorn, K. and E. M. Schmidt, "Las Vegas is better than determinism in VLSI and distributed computing," *Proc. Fourteenth Annual ACM Symposium on Theory of Computing*, pp. 330-337, 1982.

[O] Orlin, J., "Contentment in graph theory: covering graphs with cliques," *Proc. Koninklijke Nederlandse Akademie van Wetenschappen*, Amsterdam series A **80**:5, pp. 406-424, 1977.

[PS] Papadimitriou, C. H. and M. Sipser, "Communication complexity," *Proc. Fourteenth Annual ACM Symposium on the Theory of Computing*, pp. 196-200, 1982.

[S] Savage, J. E., "Planar circuit complexity and the performance of VLSI algorithms," in Kung, Sproull, and Steele (eds.), *VLSI Systems and Computations*, Computer Science Press, Rockville, Md., 1981.

[T] Thompson, C. D., "Area-time complexity for VLSI," *Proc. Eleventh Annual ACM Symposium on*

Theory of Computing, pp. 81-88, 1979.

[V] Vuillemin, J. E., "A combinatorial limit to the computing power of VLSI circuits," *Proc. Twenty-First Annual IEEE Symposium on Foundations of Computer Science*, pp. 294-300, 1980.

[Y1] Yao, A. C., "The entropic limitations of VLSI computations," *Proc. Thirteen Annual ACM Symposium on the Theory of Computing*, pp. 308-311, 1981.

[Y2] Yao, A. C., "Some complexity questions related to distributive computing," *Proc. Eleventh Annual ACM Symposium on the Theory of Computing*, pp. 209-213, 1979.