Hard-core distributions for somewhat hard problems

Russell Impagliazzo*
Computer Science and Engineering
UC, San Diego
9500 Gilman Drive
La Jolla, CA 92093-0114
russell@cs.ucsd.edu

Abstract

Consider a decision problem that cannot be $1-\delta$ approximated by circuits of a given size in the sense that any such circuit fails to give the correct answer on at least a δ fraction of instances. We show that for any such problem there is a specific "hard-core" set of inputs which is at least a δ fraction of all inputs and on which no circuit of a slightly smaller size can get even a small advantage over a random guess. More generally, our argument holds for any non-uniform model of computation closed under majorities. We apply this result to get a new proof of the Yao XOR lemma [Y], and to get a related XOR lemma for inputs that are only k-wise independent.

1 Introduction

If you have a difficult computational problem, is it always the case that several independent instances of the problem are proportionately harder than a single instance? In particular, if any algorithm taking less than R resources has failure probability at least δ for a particular problem on a certain input distribution, does combining several independent instances of the problem make the probability of success proportionally smaller? Here, "combining" can mean asking the algorithm to output answers for each input, or to compute some predicate (e.g., the parity) depending on all the answers. A canonical example of such a result is the Yao exclusive-or lemma ([Y]), which says that if we have a Boolean function f that is δ-hard for circuits of size C, and then the function $\overline{f}(x_1,...x_k) = f(x_1) \oplus f(x_2) .. \oplus f(x_k)$ is $\epsilon + (1-2\delta)^k$-hard-core for circuits of size $O(\epsilon^2 C)$. (For a generalization and exposition of this result, see Levin [L]). Such results are called "direct product theorems".

Here, we give a new proof of a version of this lemma via a result of independent interest. In intuitive terms, the new result is that for any yes/no problem for which any feasible method to solve the problem fails on a non-negligible fraction of inputs, the instances can be divided up into a set of "easy" instances and a "hard-core" of difficult instances for which it is impossible to do significantly better at computing the function than a random guess.

In addition to obtaining the Yao XOR lemma as a corollary (with slightly weaker numerical parameters), we also obtain a new direct product lemma for inputs $x_1, ..x_n$ that are only k-wise independent, rather than completely independent. This gives us, for any fixed polynomials p, and q, a security-preserving reduction from a problem that is hard with probability $1/q(n)$ over random inputs to one that is hard to guess with more than a $1/p(n)$ advantage over a random coin toss. This reduction can be used to improve the simulation time in the determinization of probabilistic algorithms based on somewhat hard problems using the approach from [NW], although in general the deterministic time to simualate BPP will remain quasi-polynomial rather than polynomial.

A more precise statement of the main result is as follows. (Although for simplicity, we give the statement of the result for circuit complexity on the uniform distribution, the technique works for an arbitrary starting distribution, and for any non-uniform model of computation closed under taking majorities.) Definitions are found in the next section.

Theorem 1 *Let f be a Boolean function on n-bit inputs that is δ-hard for circuits of size g on the uniform distribution, and let $\epsilon > 0$. Then there is a set $S \subseteq \{0,1\}^n$ so that $|S| \geq \delta 2^n$ and f is ϵ-hard-core on*

*Research Supported by NSF YI Award CCR-92-570979, Sloan Research Fellowship BR-3311, and USA-Israel BSF Grant 92-00043

S for circuits of size $d\epsilon^2\delta^2 g$, where d is an absolute constant.

2 Basic Definitions

Definition 1 *Let f be a Boolean function on n bit inputs, and D a distribution on n bit strings. Let $1/2 > \delta > 0$ and let $n \leq g \leq 2^n/n$. We say f is δ-hard on D for size g if for any boolean circuit C with at most g gates, and for x chosen according to D, $Prob[C(x) = f(x)] \leq 1 - \delta$. A* measure *on strings of length n is a function M with $M(x) \in [0, 1]$. (Think of a measure as defining a "fuzzy" set of strings, where rather than definitely being in or out of the set, x is in the set with probability $M(x)$.) The* relative size *of a measure M is written $\mu(M)$ and is defined by $\mu(M) = 1/2^n \sum_x M(x)$; the* absolute size *of a measure M is written $|M| = 2^n \mu(M) = \sum_x M(x)$. The distribution D_M induced by M is defined by $D_M(x) = M(x)/|M|$. For a circuit C and an input x define $R_C(x) = 1$ if $f(x) = C(x)$, -1 otherwise. The advantage of C on M is defined by $Adv_C(M) = \sum_x M(x)R_C(x)$. It is easy to see that if x is chosen according to D_M, $Prob[C(x) = f(x)] \geq 1/2(1 + \epsilon)$ if and only if $Adv_C(M) \geq \epsilon|M|$. If for any circuit C of size g, $Adv_C(M) < \epsilon|M|$, we call f ϵ-* hard-core *on M for size g. We call f ϵ-hard-core on S for size g if f is hard-core on the characteristic function of S with the same parameters. We call a function f ϵ-hard-core for size g if f is hard-core on the set of all inputs for the same parameters.*

In the language introduced above , the sketch of the proof of theorem 1 is as follows: We first find a hard-core measure M with $\mu(M) \geq \delta$. We then use a counting argument to show that a randomly chosen subset S is also hard-core, where S is chosen by putting $x \in S$ with probability $M(x)$. This last step seems just a technicality; the hard-core measure will be sufficient for all applications, but several proofs become simpler if we deal with a set of inputs instead of a measure. The last step is the only one that uses any specific fact about circuits, as opposed to a general non-uniform model of computation closed under taking majorities.

3 Intuition

Consider a problem like inverting a supposedly one-way function, where if we have a correct solution, then it is easy to verify it. Finding a "hard-core " set of instances for such a problem is easy (in a non-uniform model). Either there is no circuit of size $.5\epsilon\delta g$ that solves the problem on a ϵ fraction of instances, or there is. If not, our hard-core distribution is the uniform distribution on all inputs. If so, this circuit weeds out an ϵ fraction of inputs as easy, and we look for a circuit that does well on the remaining inputs. This process continues until either we find a hard-core distribution, or the set of remaining inputs is smaller than δ. Note that, since we weed out at least a $\delta\epsilon$ fraction of inputs each time, this process continues at most $1/(\delta\epsilon)$ iterations. So if we don't find a hard-core distribution, we could piece all of the circuits we found into one circuit that tries them all and outputs the first correct solution. This circuit has size g and solves the problem $1 - \delta$ of the time, contradicting the assumed hardness of the problem.

We give two proofs for the main theorem. The first follows the above outline, except that in general, we won't be able to tell when a circuit solves a particular instance, so we won't be able to just eliminate those instances where our first circuit solves the problem correctly. Instead, we gradually reduce the importance of those inputs where the circuits we have found so far do well, until we have reached a certain "comfort level" where the margin of success is high enough that we don't have to worry about that input for a while. If the margin of success is large for almost all inputs, the circuit that computes the majority of the circuits we have found computes f correctly on almost all inputs.

The second, due to Nisan ([Ni]), uses von Neumann's min-max theorem ([vN]). It constructs a game where one players moves are sets of inputs of size δ and the other player's moves are circuits, and the payoff to the second player is the advantage of the circuit on the uniform distribution on the set. This approach is similar to that taken by Lipton and Young [LY]. They were interested in finding a *small* hard-core distribution for the problem; we want to find a *large* one.

4 First Proof of Existence of Hard-Core Measures

The two proofs give somewhat different and incomparable quantitative bounds, so we state both as distinct lemmas. The quantitative aspect is in the decrease in the size of circuit for which the function is hard-core on the measure compared to the size of circuit for which the function is δ-hard.

Lemma 1 *Let f be δ-hard for size g on the uniform distribution on n-bit strings, and let $1 > \epsilon > 0$. Then there is a measure M with $\mu(M) \geq \delta$ so that f is ϵ-hard-core on M for size $.25\epsilon^2\delta^2 g$.*

Proof: Assume not, i.e., that on every measure M with $\mu(M) \geq \delta$, we can find a circuit C_M of size $g' = .25\epsilon^2\delta^2 g$ so that $Prob[f(x) = C_M(x)] \geq .5(1+\epsilon)$ when x is chosen according to D_M. Let $\gamma = \epsilon\delta$. Then for each such M, $Adv_{(C_M)}(M) \geq \epsilon|M| \geq \gamma 2^n$.

For a sequence of circuits $C_1, ...C_i$, let $N_i(x) = \sum_{1 \leq j \leq i} R_{C_i}(x)$ (i.e., the margin by which we are predcting f on x correctly), and let $M_i(x) = 1$ if $N_i(x) \leq 0$, 0 if $N_i(x) \geq 1/\gamma$ and $1 - \gamma N_i(x)$ otherwise. (In other words, if we've guessed more incorrectly than correctly on x we definitely want to include x in our next candidate hard-core distribution, if we have a comfortable margin on x, we don't, and if we are somewhere in between, include x with a probability decreasing linearly with our margin.) For the empty set of circuits, define $N_0(x) = 0$ and hence $M_0(x) = 1$ (i.e., we start in the uniform distribution on all inputs.)

Let $C_1 = C_{M_0}$, and until $\mu(M_i) < \delta$, let $C_{i+1} = C_{M_i}$. Note that,for maj the boolean majority function, $maj(C_1, ..C_i)$ is correct on all inputs except those with $N_i(x) \leq 0$ and hence $M_i(x) = 1$, so $Prob[maj(C_1, ..C_i) = f(i)] \geq 1 - \mu(M_i)$. So if the above process halts before $i_0 = g/2g' = 2\gamma^{-2}$ then this defines a circuit of size at most $g' i_0 + O(i_0) < 2g' i_0 = g$ gates that computes f on $1 - \delta$ of the inputs, a contradiction to the assumed hardness of f.

On the other hand, we claim that the process halts before i_0 steps. Let x be any fixed input. Let $A_i(x) = \sum_{0 \leq j \leq i-1} R_{C_i}(x) M_{i-1}(x)$. We claim $A_i(x) \leq 1/\gamma + .5\gamma i$. To see this, for each $k \geq 0$, match up the times j so that $N_j(x) = k$ and $N_{j+1}(x) = k+1$ with those where $N_j(x) = k+1, N_{j+1}(x) = k$, with possibly one time left out for each $0 \leq k < N_i(x)$. (In other words, if you ride an elevator starting at the ground floor, you will go up from floor k to floor $k+1$ at most one more time than you go down from floor $k+1$ to floor k, and that one time will occur if and only if you get off at a floor greater than k.) Do the analagous matching for $k < 0$, matching the times a when $N_a(x)$ drops from k to $k-1$ with those where it rises from $k-1$ to k. For each such pair of times a, b, $R_{C_{a+1}}(x) M_a(x) + R_{C_{b+1}}(x) M_b(x) = M_a(x) - M_b(x)$. If $0 \leq k < 1/\gamma$, this is $1 - k(\gamma) - (1 - (k+1)\gamma = \gamma$, and otherwise it is 0. Thus, each pair contributes at most γ to the sum, so the at most $.5i$ pairs together contribute at most $.5i\gamma$. Each unmatched time c with $N_c(x) < 0$ contributes -1 to the sum, each unmatched time with $0 \leq N_c(x) < 1/\gamma$ contributes at most 1, and each unmatched time with $N_c(x) \geq 1/\gamma$ contributes 0 since $M_c(x) = 0$, so the total contribution of the unmatched edges is at most $1/\gamma$. Thus, we have proved the claim.

Assume the process continues for $i_0 + 1$ steps. Then $\sum_x A_{i_0+1}(x) = \sum_x \sum_{0 \leq j \leq i_0} R_{C_{j+1}}(x) M_j(x) = \sum_{0 \leq j \leq i_0} Adv_{C_{j+1}}(M_j) \geq 2^n \gamma (i_0 + 1)$. On the other hand, from the claim proved in the above paragraph, we have $\sum_x A_{i_0+1}(x) \leq \sum_x (1/\gamma + .5(i_0+1)\gamma) = 2^n(1/\gamma + .5(i_0+1)\gamma)$, so $.5\gamma(i_0+1) \leq 1/\gamma$, contrary to our choice of $i_0 = 2\gamma^{-2}$. Thus, the process must terminate within i_0 steps, which yields a circuit of size g computing f on all but a δ fraction of inputs, a contradiction which proves the lemma. □

5 The Min-max proof

We give here an alternate existence proof of a hard-core measure due to Nisan. The proof yields an incomparable size bound, which is significantly better as a function of δ, but slightly worse as a function of ϵ. The exact size function is stated in the next lemma.

Lemma 2 *Let f be δ-hard for size g on the uniform distribution on n-bit strings, and let $1 > \epsilon > 0$. Then there is a measure M with $\mu(M) \geq \delta$ so that f is ϵ-hard-core for size $g' = (1/16)\epsilon^2(-log(.5\epsilon\delta))^{-1} g$ on M.*

Proof: Consider the following two player zero-sum game. Player 1 picks a set S of $\delta 2^n$ inputs. Player 2 picks a circuit C of size g'. The payoff for Player 2 is $Adv_C(S)$. By the min-max Theorem ([vN]), either the first player has a mixed strategy so that there is no move for the second that achieves an expected payoff larger than $\epsilon 2^n$, or there is a mixed strategy for the second player which achieves expected payoff $\geq \epsilon 2^n$ for any move from the first player.

In the first case, there is a distribution on sets of size at least $\delta 2^n$ so that the average advantage of any circuit is at most $\epsilon 2^n$. This is the same as the average advantage of the circuit on the distribution D on inputs where one picks a set S according to the mixed strategy, and then picks an input uniformly from S. Since each set has $\delta 2^n$ elements, $D(x) \leq 1/(\delta 2^n)$ for any input x, so the function $M(x) = \delta 2^n D(x)$ is a measure of size $\delta 2^n$ on which no circuit of size g' can achieve advantage $\epsilon\delta 2^n$. So f is ϵ-hard-core for size g' on M.

In the second case, there is a mixed strategy for the second player, i.e., a distribution on circuits of size g', that achieves average advantage at least ϵ on any set of size $\delta 2^n$. Then the set S of inputs on which this distribution has probability at most $1/2 + \epsilon/2$ of being correct, is of size at most $\delta(1 - \epsilon/2)2^n$. For otherwise, the set S' of those $\delta 2^n$ inputs for which the advantage is smallest would have at most $.5\delta\epsilon 2^n$ nonmembers of S, so the advantage would be less than

$(\delta)(2^n.5\epsilon) + .52^n\delta\epsilon = 2^n\delta\epsilon$, contradicting the average advantage of at least ϵ on this set of size $\delta 2^n$.

Then picking t independent random samples from this distribution and taking their majority, the probability that the output is incorrect for any specific $x \notin S$ is at most $e^{-\epsilon^2 t/8}$ by Chernoff bounds. Setting $t = 8\epsilon^{-2}(-\log(.5\epsilon\delta))$, this probability is at most $\epsilon\delta$. So this gives a probabilistic construction of a circuit of size g whose overall error is at most $(\delta(1-.5\epsilon))+.5\delta\epsilon = \delta$.

Thus, either there is a circuit of size g computing the function within error δ or there is a hard-core measure of size δ for circuits of size g'. □

6 Getting a hard core set from a hard-core measure

Lemma 3 *Let M be an $\epsilon/2$ hard-core measure for f for size $2n < g < (1/8)(2^n/n)(\epsilon\delta)^2$, and assume $\mu(M) \geq \delta$. Then there is a ϵ-hard-core set S for f for size g with $|S| \geq \delta 2^n$.*

Proof: First, note that the number of circuits of size g is at most $(2(2n+g))^{2g} \leq 2^{2ng} \leq< .25e^{2^n\epsilon^2\delta^2/2}$. Let C be any circuit of size g, and pick S by placing $x \in S$ with probability $M(x)$. Let M_S be the characteristic function for S. Then $Adv_C(M) = Exp[Adv_C(M_S)] \leq \epsilon\mu(M)$, and $Adv_C(M_S)$ is the sum of 2^n independent random variables that are in the interval $[0, 2^{-n}]$. Hence, the probability that $Adv_C(M_S) \geq 2\epsilon\mu M$ is at most $e^{-2^n\epsilon^2\delta^2/2}$, by Chernoff bounds. Thus, the probability that there is such a C is at most $1/4$. On the other hand, the probability that $|S| \geq Exp|S| = \mu(M)$ is at least $1/2$ Therefore, there is a set S with $|S| \geq \mu(M)$ and $Adv_C(M_S) \leq 2\epsilon\mu(M)$ for every circuit C with at most g gates. Therefore, $Prob[C(x) = f(x)] \leq 1/2 + \epsilon$ for x uniformly selected from S for any such circuit C, and so f is $1/2 - \epsilon$-hard for size g on M_S. □ Theorem 1 then follows from combining Lemma 1 and Lemma 3. with the observation that if f is δ-hard on any distribution for any δ for size g, then $g < 2^n/n$, since any function can be computed with $2^n/n$ gates. □

7 Yao's XOR Lemma

Here, we use the hard-core distribution to give a simple proof of Yao's XOR Lemma, that follows closer to the intuition that if we ask several questions of a somewhat hard problem, then with high probability, one is a hard instance, and hence the XOR is hard. We state the lemma directly in terms of hard-core sets, so that any quantitative improvement of our results translates directly.

Lemma 4 *If f is ϵ-hard-core for some set of $\delta 2^n$ inputs for size g, then the function $\overline{f}(x_1, ..x_k) = f(x_1) \oplus f(x_2) \oplus .. \oplus f(x_k)$ is $\epsilon + (1-\delta)^k$-hard-core for size g.*

This combined with either proof of the main theorem gives a version of the XOR lemma.

Proof: To get a contradiction, let C be a circuit of size at most g that achieves advantage more than $\epsilon + (1-\delta)^k$ for computing the XOR of k independent instances of the problem. Let A_l be C's advantage given that exactly l instances are from the hard-core set. Then since the probability of having no instances of the set is $(1-\delta)^k$, there must be an $l \geq 1$ with $A_l \geq \epsilon$. Then let C' be a circuit constructed probabilistically as follows : Pick $a_1, ..a_{l-1}$ uniformly from the hard core, and pick $b_1, ..b_{k-l}$ uniformly from the complement of the hard-core. Then C' is the circuit with input x which is C on the inputs $a_1, ..a_{l-1}, b_1, ..b_{k-l}, x$ inserted in a random order, with the output negated if and only if $f(a_1) \oplus f(a_2) \oplus ..f(a_{l-1}) \oplus f(b_1) \oplus ..f(b_{k-l}) = 1$. Then the average advantage (over the choice of $\vec{a}, \vec{b}$ and the order for inputs) of C' for x chosen uniformly from the hard-core set is exactly $A_l \geq \epsilon$, and each C' is of size at most that of C. This contradicts the assumption that the set was hard-core. □

8 An amplification lemma for k-wise independent inputs

Nisan and Wigderson showed that a problem in EXPTIME for which no circuit of exponential-size could achieve even an exponentially small advantage, then $P = BPP$ [NW]. They use hard instances of the problem of logarithmic size as the seed for a pseudorandom generator. However, the candidates for such a problem are more likely to be shown to be hard some fraction of the time than to be directly shown hard to even get a small advantage on. For example, Lipton [Li] gives a self-reduction for the permanent problem that shows that if there is a circuit for this problem that succeed on a $1 - 1/2n$ fraction of random $n \times n$ matrices, then there is one not much bigger that compute the permanent for all such matrices. Thus, the worst-case and average-case complexities of this problem are almost identical, and since the permanent is $\#P$-complete, the worst case complexity is probably quite high. We could then use Yao's XOR lemma to get a related problem that is hard to get a small advantage on, but this would increase the size of the inputs by a factor of n. This would still be enough to put BPP in deterministic quasi-polynomial time, but not in P.

Actually, since the $n \times n$ permanent problem requires $O(n^2 \log n)$ bits to specify a single instance, and it can be solved by brute force in $O(n^n)$ time, it is not really a candidate for a Nisan-Wigderson type pseudo-random generator that achieves $P = BPP$ anyway. However, it still illustrates that the efficiency of the simulation of probabilistic time is determined by the *input size* required for the problem to have a certain level of difficulty. Thus, it would be nice to have a version of the XOR lemma that did not increase the input size significantly. An analagous result for amplifying the hardness of one-way functions without substantially increasing the input length appears in [GILVZ].

We are not quite able to achieve the full length-preserving reduction from a problem that is hard to solve $1 - \delta$ fraction of the time by exponential size circuits to one that is hard to approximate with some exponential advantage . However, we are able to reduce from any polynomially small failure probability to any polynomially small advantage with only a constant factor increase in input size.

Theorem 2 *Let f be a decision problem on n-bit strings that is $1/n^c$-hard for circuits of size g. Then for any c', there is a decision problem f' on $O(n)$ bit strings which is polynomial-time truth-table reducible to f and is $1/n^{c'}$-hard-core for circuits of size $g/n^{O(1)}$.*

Definition 2 *We say that a function $p(r)$ outputting sequences of n n-bit strings, $x_1, ..x_n$ is* k-wise independent *iff for every $i_1, ..i_k, y_1, ..y_k$, there are the same number of r's so that for $\vec{x} = p(r)$, $x_{i_j} = y_j, j = 1..k$. We say that p is* constructively *k-wise independent if in addition p is polytime computable and there is a poly-time algorithm that on input $i_1, ..i_k, y_1, ..y_k$ uniformly generates such r's.*

There are known methods of computing constructively k-wise independent functions for any fixed k with $|r| = O(n)$. For example, over the finite field of 2^n elements, we can use $p(a_1, ..a_k)$ which sets $x_i = a_1 + c_i a_2 + c_i^2 a_3 + ...c_i^{k-1} a_k$ for any n distinct elements $c_1, ..c_n$ of the field. The required algorithm is then interpolating a degree $k-1$ polynomial from its values at k points; it follows that there is one and only one solution for any set of i_j's and y_j's.

For $s, r \in \{0,1\}^n$, let $< s, r >$ denote the inner product mod 2 of vectors s and r, i.e., $< s, r >= s_1 r_1 \oplus s_2 r_2 \oplus .. \oplus s_n r_n$. The proof of theorem 2 makes extensive use of the following theorem due to Goldreich and Levin ([GL]):

Theorem 3 *There is a probabilistic oracle machine $M^O(n, \gamma)$ running in time polynomial in n and $1/\gamma$ with the following property: Let $v \in \{0,1\}^n$ and B be a function from $\{0,1\}^n$ to $\{0,1\}$. If $B(s) =< s, v >$ with probability at least $1/2 + \gamma$ for $s \in_U \{0,1\}^n$, then $M^B(n, \gamma) = v$ with probability at least $O(\gamma^2)$.*

Corollary 1 *There is a probabilistic oracle machine $N^O(n)$ running in time polynomial in n with the following property: Let $v \in \{0,1\}^n$ and B be a function from $\{0,1\}^n$ to $\{0,1\}$. If $B(s) =< s, v >$ with probability at least $.8$ for $s \in_U \{0,1\}^n$, then $N^B(n) = v$ almost certainly.*

Proof: First use the machine from theorem 3 to get a polynomial-size list that almost certainly contains v. For each $w \neq v$ on the list, $Prob[< s, w >=< s, v >] = .5$, so $Prob[B(s) =< s, w >] \leq Prob[< s, w >=< s, v >] + Prob[B(s) \neq < s, v >] \leq .5 + .2 = .7$. Thus, by testing B for many random values of s and outputting the probably unique element of the list that matches on more than a .75 fraction of the samples, we almost certainly output v. □

Corollary 2 *Let $h : \{0,1\}^n \rightarrow \{0,1\}^n$. If there is a circuit C of size g that given random $s, r \in \{0,1\}^n$ computes $< s, h(r) >$ with probability at least $1 - \gamma$, then there is a circuit of size $n^{O(1)}g$ that given a random $r \in \{0,1\}^n$ computes $h(r)$ with probability at least $1 - 5\gamma$.*

Proof: The fraction of r's on which C outputs $< s, h(r) >$ with probability less than .8 is at most 5γ. We can use the machine from corollary 1 with oracle $B(s) = C(r, s)$ to almost certainly compute $h(r)$ for all but this set. □

Corollary 3 *Let $h : \{0,1\}^n \rightarrow \{0,1\}^n$. If there is a circuit C of size g that given random $s, r \in \{0,1\}^n$ computes $< s, h(r) >$ with probability at least $1/2 + 2\gamma$, then there is a circuit of size $(n/\gamma)^O(1)g$ that given a random $r \in \{0,1\}^n$ computes $h(r)$ with probability at least $O(\gamma^3)$.*

Proof: The fraction of r's on which C outputs $< s, h(r) >$ with probability greater than $1/2 + \gamma$ is at least γ. We can use the machine from corollary 3 with oracle $B(s) = C(r, s)$ to output $h(r)$ with probability at least $O(\gamma^2)$ for each such r, for a total probability of $O(\gamma^3)$ over random r's. □

We are now ready to prove Theorem 2. First, we apply the following lemma c times to amplify a function that is $1/n^c$-hard to one that is $\Omega(1)$-hard:

Lemma 5 *Let f be a δ-hard decision problem for circuits of size g, where $\delta < 1/(16n)$. Let p be a constructively pairwise independent function. Let $f'(r,s) =< s, f(x_1), ..f(x_n) >$ where x_i are obtained from r using p Then f' is $.05\delta n$-hard for circuits of size $\delta^{O(1)}n^{-O(1)}g$,*

Proof: From corollary 2, it suffices to prove that no circuit of some size which is $\delta^{O(1)}n^{-O(1)}g$ can compute $\overline{f}(r) = f(x_1)..f(x_n)$ with probability $1 - .25\delta$. Let f be .2-hard-core for a set H of size $\delta 2^n$ for circuits of size $g_1 = d_1\delta^2 g$, for some constant d_1, as is guaranteed to exist by theorem 1. Assume that there were a circuit C_1 of size g_1 so that for all but a $.25\delta n$ fraction of seeds r output $f(x_1)...f(x_n)$. Now, by the pairwise independence of the x_i's, for any i the probability that $x_i \in H$ and for all $j \neq i$, $x_j \notin H$ is at least $\delta(1 - n\delta) \geq (15/16)\delta$. Assume there were no i so that conditioned on this event occurring for i, the probability of outputting $\overline{f}(r)$ was at least 2/3. Then since the events are exclusive, there would be a total of at least $(1/3)(15/16)\delta n > .25\delta n$ total failure probability, contrary to assumption. Thus there must be one i so that conditioned on x_i being the sole element of H, the failure probability is at most 1/3. Then since conditioned on $x_i \in H$, there is at least a 15/16 probability that x_i is the sole element in H, the overall failure probability given that $x_i \in H$ is at most $(1/3) + (1/16) \leq .4$. Thus, using the i'th bit of the output of C_1 as a guess for $f(x_i)$ gives advantage at least $.6 - .4 = .2$, which contradicts the assumption that f is .2-hard-core for H. $\Box$.

We now want to amplify from $O(1)$-hardness to get a $1/poly$-hard-core problem.

Lemma 6 *Let $1 > \delta = \omega(1)$. Let f be a δ-hard decision problem for circuits of size g. Let p be a constructively $2k$-wise independent function. Let $f'(r,s) =< s, f(x_1), ..f(x_n) >$ where the x_i are obtained from r using p. Then f' is $O((n)^{-k/3})$-hard-core for circuits of size $gn^{-O(1)}$.*

Proof: From Corollary 3, it suffices to prove that no circuit of some size which is $n^{-O(1)}\delta^{O(1)}g$ can compute $\overline{f}$ for some $O(n^{-k})$ fraction of inputs. Let $\gamma = .5(4n)^{-k}$

Let H be a γ-hard-core set for f of size δ for circuits of size $g_1 = gn^{-O(1)}$, , as given by theorem 1. Assume without loss of generality that the interpolation algorithm for p can be implemented by a circuit of size at most $.25g_1$.

Assume there were a circuit C of size $.25g_1$ that computed $\overline{f}(r)$ with probability greater than $2\delta^{-2k}(4k)^k(n)^{-k}$. Let $C_i = C_i(r)$ denote the i'th output of C. From now on, we omit specific references to r, and view the choice of a random r as determining a distribution on $\vec{x}, \vec{C}$.

Let E_i be the event that $x_i \in H$ and $C_i \neq f(x_i)$. Let F_i be the event that $x_i \in H$ and $C_i = f(x_i)$. Let G_i be the event that $x_i \notin H$.

We will show, by induction on m, that for any distinct $i_1, ..i_l$, any j, m with $0 \leq j \leq m \leq l \leq 2k$, and any values $y_{i_{m+1}}, ..y_l$ the probability that $E_{i_1}, ..E_{i_j}$, $F_{i_{j+1}}, ..F_{i_m}$ given that $x_{i_{m+1}} = y_{i_{m+1}}, ..x_{i_l} = y_{i_l}$, is $(\delta/2)^m(1+\kappa)$, for some $|\kappa| \leq 82^m\gamma$. We will simultaneously prove that, if $l < 2k$, for any i_0, the conditional probability that E_{i_0} given $E_{i_1}, ..E_{i_j}, F_{i_{j+1}}, ..F_{i_m}$, and given that $x_{i_{m+1}} = y_{i_{m+1}}, ..x_{i_l} = y_{i_l}$, is $\delta/2(1+\lambda)$ where $|\lambda| \leq 2(2)^m\gamma$.

For $m = 0$, the first claim follows immediately from the $2k$-wise independence.

Assume the first claim holds for m. If the second claim fails for some i_0, then without loss of generality assume that the above conditional probability for E_{i_0} is at least $\delta/2(1 + 2(2)^m\gamma)$. Consider the folowing probabilistic construction of a circuit C' to guess $f(x)$ given x: Choose $b_{i_1}, ..b_{i_m}$ uniformly from H. C' interpolates an r so that $x_{i_t} = y_{i_t}$, for $m+1 \leq t \leq l$, $x_{i_t} = b_{i_t}$ for $1 \leq t \leq m$ and $x_{i_0} = x$. If $C_{i_t} = f(b_{i_t})$ for all $1 \leq t \leq j$ and $C_{i_t} \neq f(b_{i_t})$ for all $j+1 \leq t \leq m$, then C outputs C_{i_0}; otherwise, C outputs a random bit. By the first claim for m, (and letting $i_{l+1} = i_0$) the conditional probability, for any fixed value x for x_{i_0}, that $E_{i_1}, ..E_{i_j}, F_{i_{j+1}}, ..F_{i_m}$ is at least $(\delta/2)^m(1+\kappa)$ where $\kappa \geq -8(2)^m\gamma > -.5$. So the conditional probability of these given $x \in H$ and each $x_{i_t} \in H$ for $1 \leq t \leq m$ is at least $.5(2)^{-m}$. Thus, the probability that C' outputs C_{i_0} on a random $x \in H$ is at least $.52^{-m}$. On the other hand, by the assumption, the conditional probability (over both the choice of the parameters $\vec{b}$ and a random $x \in H$) that C' outputs $f(x)$ if it outputs C_{i_0} is at least $1/2(1 + 2(2)^m\gamma)$. This means the expectation of the advantage of C' for a random $x \in H$ is greater than γ. On the other hand, each instantiation of C' has at most $.25g_1 + .25g_1 + O(k) \leq g_1$ gates, (to do the interpolation, simulate C, and then check at most $2k$ values against a table) contradicting the assumption that f is γ-hard-core on H for size g_1.

Assume both claims hold for m. Shifting the names of indices for the first claim for $m+1$ down by 1, and applying both claims, we see that the probability that $E_{i_2}, ..E_{i_j}, F_{i_{j+1}}, ..F_{i_{m+1}}$, given that $x_{i_t} = y_{i_t}$ for $m+2 \leq t \leq l$ is $(\delta/2)^m(1+\kappa)$ where $|\kappa| \leq 8(2)^m\gamma$, and the conditional probability that E_{i_1} given all of the above is $\delta/2(1+\lambda)$ where $|\lambda| \leq 2(2)^m\gamma$. Hence the probability that $E_{i_1}, ..E_{i_j}, F_{i_{j+1}}, ..F_{i_{m+1}}$, given that $x_{i_t} = y_{i_t}$ for $m+2 \leq t \leq l$ is $(\delta/2)^{m+1}(1+\kappa)(1+\lambda)$, and $|(1+\kappa)(1+\lambda)-1| \leq |\lambda|+|\kappa|+|\lambda\kappa| \leq 2|\lambda|+|\kappa| \leq 4(2^m\gamma + 8(2)^m\gamma \leq 8(2)^{m+1}\gamma$, so the first claim holds for $m+1$.

Let V_i be the random variable with value $1-\delta/2$ if E_i and value $-\delta/2$ otherwise. Then from the above, for any $l \leq 2k$,and any distinct $i_1, ..i_l$, the conditional expectation of V_{i_l} given for each $1 \leq t \leq l-1$ whether E_{i_t}, F_{i_t}. or G_{i_t}, is of absolute value at most $2(2)^l\gamma$. It follows that the same is true given $V_{i_1}..V_{i_{l-1}}$.

Consider the expectation of $(\sum_{i=1}^{i=n} V_i)^{2k}$. There are n^{2k} different terms when we expand out the product and do not combine like terms. We can bound the expectation of the sum by the expectation of each term. From the above, the expectation of any term containing a single power of V_t is at most $22^{2k}\gamma$, since the product of the other variables in the term less than 1 in absolute value, and since the aforementioned number bounds the absolute value of the conditional expectation of V_t given the other values. All terms not containing a single power of some variable involve at most k variables. For each set of k variables, the terms only involving those k are those with value 1 instead of 0 when we do the formal substitution $V_i = 1$ for those i in the set, and $V_i = 0$ otherwise. Since then the product has value k^k, there are k^k such terms for each such set. Thus, there are at most $k^k n^k$ terms with no single power of a variable (overcounting considerably). Each contributes at most 1 to the expectation. Thus, the total expectation is less than $(nk)^k + 2n^{2k}(2)^{2k}\gamma = (nk)^k + n^k \leq 2(nk)^k$ by our choice of γ.

However, on any input where C is correct, in particular there are no mistakes on elements from H, so E_i is always false, and $V_i = -\delta/2$ for each i. Thus, the $2k$'th power of the sum of the V_i's is $(.5\delta n)^{2k} = 2(nk)^k(.5(.25\delta^2/k)^k(n)^k)$. So by Markov's inequality, the probability that C succeeds is at most $2(4k/\delta^2)^k(n)^{-k} = O(n^{-k})$. contrary to assumption. □.

9 Open Questions

Can you, starting from a hard problem f for exponentially large circuits, obtain a problem that is hard-core by an exponentially small amount without increasing the input size? In particular, if you pick $x_1, ..x_n$ via a walk on a constructive expander, can any small circuit compute $f(x_1), ..f(x_n)$ and be right on all inputs more than an exponentially small fraction of the time?

Why is the size of the hard-core δ rather than 2δ? In fact, you can easily boot-strap the argument to find a hard-core of size $(2-r)\delta$ for any constant r.

Why in all Yao-style arguments is there a trade-off between resources and probability, rather than a real increase in the hardness in the problem? If f is hard for resources R, the parity of many copies of f should still be hard for resources R, not just some slightly smaller bound.

Acknowledgements

We would like to thank Noam Nisan, Avi Wigderson, Arvind Gupta, Richard Beigel, Steven Rudich, and Mike Saks for helpful conversations.

References

[GILVZ] Goldreich, O., Impagliazzo, R., Levin, L., Venketesan, R., Zuckerman, D., "Security Preserving Amplification of Harness", 31^{st} *FOCS*, 1990, pp. 318-326.

[GL] Goldreich, O., and L.A. Levin, "A Hard-Core Predicate for any One-way Function", 21^{st} *STOC*, 1989, pp 25-32.

[L] Levin, L.A., "One-way Function and Pseudorandom Generators", *Combinatorica*, Vol. 7, No. 4, 1987, pp. 357-363.

[Li] R. Lipton, "New directions in testing", In J. Fegenbaum and M. Merritt, editors, *Distributed Computing and Cryptography*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science Volume 2, pp. 191-202. American Mathematical Society, 1991.

[LY] R. Lipton and N. Young, "Simple Strategies for Large Zero-Sum Games with Applications to Complexity Theory" , 26^{th} *STOC*, 1994, pp.734-740.

[Ni] N. Nisan, personal communication, 1994.

[NRS] N. Nisan, S. Rudich M. Saks, "Products and Help Bits in Decision Trees", 35^{th} *FOCS* 1994, pp. 318-329.

[NW] N. Nisan and A. Wigderson, Hardness vs. Randomness, Journal of Computer and System Sciences, , vol. 49, No. 2 (Oct., 1994), pp. 149-67.

[vN] J. von Neumann, "Zur Theorie der Gesellschaftspiel", Mathematishe Annalen, 100(295-320),1928.

[Y] Yao, A.C., "Theory and Applications of Trapdoor Functions", 23^{rd} *FOCS*, 1982, pp. 80-91.