Texts in Computational Complexity: Proving Yao's XOR Lemma via the Direct Product Lemma

Oded Goldreich
Department of Computer Science and Applied Mathematics
Weizmann Institute of Science, Rehovot, ISRAEL.

November 23, 2005

1 Preliminaries

Our aim is to relate mild and strong notions of inapproximability. These notions are obtained as special cases of the following general definition.

Definition 1 (inapproximability, a general formulation): *We say that $f : \{0,1\}^* \to \{0,1\}$ is (S,ρ)-inapproximable if for every family of S-size circuits $\{C_n\}_{n\in\mathbb{N}}$ and all sufficiently large n it holds that*

$$\Pr[C(U_n) \neq f(U_n)] \geq \frac{\rho(n)}{2} \tag{1}$$

We say that f is T-inapproximable if it is $(T, 1-(1/T))$-inapproximable.

We refer to mildly inapproximable predicate as provided by the following result of [1].

Theorem 2 *Suppose that there exists a Boolean function f in $\mathcal{E}$ having circuit complexity that is almost-everywhere greater than S. Then, there exists an exponential-time computable function $\hat{f} : \{0,1\}^* \to \{0,1\}^*$ that is (S',ρ')-inapproximable for $S'(n') = S(n'/O(1))/\mathrm{poly}(n')$ and $\rho'(n') = (1/n')^2$ such that $|\hat{f}(x)| \leq |x|$. That is, for every family of circuit $\{C'_{n'}\}_{n'\in\mathbb{N}}$ of size $S'(n') = S(n'/O(1))/\mathrm{poly}(n')$ it holds that $\Pr[C'_{n'}(U_{n'}) \neq \hat{f}(U_{n'})] > (1/n')^2$.*

Our aim will be to establish strong inapproximable predicate as stated next.

Theorem 3 *Suppose that for every polynomial p there exists a problem in $\mathcal{E}$ having circuit complexity that is almost-everywhere greater than p. Then there exist polynomial-inapproximable Boolean functions in $\mathcal{E}$; that is, for every polynomial p there exists a p-inapproximable Boolean function in $\mathcal{E}$.*

This will be done by using Yao's XOR Lemma (i.e., Theorem 5), which will be proved by combining a Direct Product Lemma (i.e., Theorem 6) with the following result of [3].

Theorem 4 (a generic hard-core predicate, revisited): *There exists a probabilistic oracle machine that, given parameters n,ε and oracle access to any function $B : \{0,1\}^n \to \{0,1\}$, for every $x \in \{0,1\}^n$, given oracle access to any B_x halts after $\mathrm{poly}(n/\varepsilon)$ steps and with probability at least $1/2$ outputs a list of all strings $x \in \{0,1\}^n$ that satisfy*

$$\Pr_{r\in\{0,1\}^n}[B(r) = b(x,r)] \geq \frac{1}{2} + \varepsilon,$$

where $b(x,r)$ denotes the inner-product mod 2 of x and r.

For motivational purposes, we will refer in a few places to the amplification of one-way functions (see, e.g., [2, Sec. 2.3]), but these references are inessential.

2 Main Text

Having obtained a mildly inapproximable predicate, we wish to obtain a strongly inapproximable one. The information theoretic context provides an appealing suggestion: Suppose that X is a Boolean random variable (representing the mild inapproximability of the aforementioned predicate) that equals 1 with probability ε. Then XORing the outcome of n/ε independent samples of X yields a bit that equals 1 with probability $0.5 \pm \exp(-\Omega(n))$. It is tempting to think that the same should happen in the computational setting. That is, if f is hard to approximate correctly with probability exceeding $1-\varepsilon$ then XORing the output of f on n/ε non-overlapping parts of the input should yield a predicate that is hard to approximate correctly with probability that is non-negligibly higher than $1/2$. The latter assertion turns out to be correct, but (as in the amplification of one-way functions; cf. [2, Sec. 2.3]) the proof of the computational phenomenon is considerably more complex than the analysis of the information theoretic analogue.

Theorem 5 (Yao's XOR Lemma): *Let p be a polynomial and suppose that the Boolean function f is $(T, 1/p)$-inapproximable, for every polynomial T. Then the function $F(x_1, ..., x_{t(n)}) = \oplus_{i=1}^{t(n)} f(x_i)$, where $x_1, ..., x_{t(n)} \in \{0,1\}^n$ and $t(n) = n \cdot p(n)$, is T'-inapproximable for every polynomial T'.*

Combining Theorems 2 and 5 (and Exercise 8), we complete the (first) proof of Theorem 3. Several different proofs of Theorem 5 are known. We choose using a proof that "reduces" the analysis of the exclusive-or of predicates to the analysis of their direct product. That is, the proof proceeds in two steps: First we prove that the corresponding "direct product" function $P(x_1, ..., x_{t(n)}) = (f(x_1), ..., f(x_{t(n)}))$ is difficult to compute in a strong average-case sense, and next we establish the desired result by an application of Theorem 4. In fact, the first step is the main one, and we believe that it is of independent interest (and thus generalize it from Boolean functions to arbitrary ones).

Theorem 6 (The Direct Product Lemma): *Let p be a polynomial and $f : \{0,1\}^* \to \{0,1\}^*$. Suppose that for every family of polynomial-size circuits, $\{C_n\}_{n\in\mathbb{N}}$, and all sufficiently large $n \in \mathbb{N}$, it holds that $\Pr[C_n(U_n) \neq f(U_n)] > 1/p(n)$. Let $P(x_1, ..., x_{t(n)}) = (f(x_1), ..., f(x_{t(n)}))$, where $x_1, ..., x_{t(n)} \in \{0,1\}^n$ and $t(n) = n \cdot p(n)$. Then, for every family of polynomial-size circuits, $\{C'_m\}_{m\in\mathbb{N}}$, it holds that $\Pr[C'_m(U_m) = P(U_m)] < \mu(m)$, where μ is a negligible function.*

Theorem 5 follows from Theorem 6 by considering the function $P'(x_1, ..., x_{t(n)}, r) = b(f(x_1) \cdots f(x_{t(n)}), r)$, where f is a Boolean function, $r \in \{0,1\}^{t(n)}$, and $b(y, r)$ is the inner-product modulo 2 of the $t(n)$-bit long strings y and r. Applying Theorem 4, we infer that P' is T'-inapproximable for every polynomial T'. Lastly, we reduce the approximation of P' to the approximation of F (see Exercise 9), and Theorem 5 follows.

Proof of Theorem 6. As in the proof of the amplification of one-way functions (see [2, Sec2.3]), we show how to converts circuits that violate the theorem's conclusion into circuits that violate the theorem's hypothesis. We note, however, that things were much simpler in the context of the amplification of one-way functions: There we could (efficiently) check whether or not a value contained in the output of the circuit that solves the direct-product problem constitutes a correct answer for the corresponding instance of the basic problem. Lacking such an ability in the current context, we shall have to use such values more carefully. Loosely speaking, we will take a weighted majority vote among various answers, where the weights reflect our confidence in the correctness of the various answers.

We derive Theorem 6 by applying the following lemma that provides quantitative bounds on the feasibility of computing the direct product of two functions. In this lemma, $\{Y_m\}_{m\in\mathbb{N}}$ and $\{Z_m\}_{m\in\mathbb{N}}$ are independent probability ensembles such that $Y_m, Z_m \in \{0,1\}^m$, and $X_n = (Y_{\ell(n)}, Z_{n-\ell(n)})$ for some function $\ell : \mathbb{N} \mapsto \mathbb{N}$. The lemma refers to the success probability of computing the direct product function F defined by $F(yz) = (F_1(y), F_2(z))$, where $|y| = \ell(|yz|)$, when given bounds on the success probability of computing F_1 and F_2 (separately). Needless to say, these probability bounds refer to circuits of certain sizes. We stress that the statement of the lemma is not symmetric with respect to the two functions, guaranteeing a stronger (and in fact lossless) preservation of circuit sizes for one of the functions (which is arbitrarily chosen to be F_1).

Lemma 7 (Direct Product, a quantitative two argument version): *For $\{Y_m\}$, $\{Z_m\}$, F_1, F_2, ℓ, $\{X_n\}$ and F as in the foregoing, let $\rho_1(\cdot)$ be an upper-bound on the success probability of $s_1(\cdot)$-size circuits in computing F_1 over $\{Y_m\}$. That is, for every such circuit family $\{C_m\}$*

$$\Pr[C_m(Y_m)=F_1(Y_m)] \leq \rho_1(m)$$

Likewise, suppose that $\rho_2(\cdot)$ is an upper-bound on the probability that $s_2(\cdot)$-size circuits compute F_2 over $\{Z_m\}$. Then, for every function $\varepsilon:\mathbb{N}\mapsto\mathbb{R}$, the function ρ defined as

$$\rho(n) \stackrel{\rm def}{=} \rho_1(\ell(n)) \cdot \rho_2(n-\ell(n)) + \varepsilon(n)$$

is an upper-bound on the probability that families of $s(\cdot)$-size circuits correctly compute F over $\{X_n\}$, where

$$s(n) \stackrel{\rm def}{=} \min\left\{s_1(\ell(n))\ ,\ \frac{s_2(n-\ell(n))}{\mathrm{poly}(n/\varepsilon(n))}\right\}$$

Theorem 6 is derived from Lemma 7 by using careful induction, which capitalizes on the asymmetry of Lemma 7. Specifically, we write $P(x_1, x_2, ..., x_{t(n)})$ as $P^{(t(n))}(x_1, x_2, ..., x_{t(n)})$, where $P^{(i)}(x_1, ..., x_i) = (f(x_1), ..., f(x_i))$ and $P^{(i)}(x_1, ..., x_i) \equiv (P^{(i-1)}(x_1, ..., x_{i-1}), f(x_i))$. For every polynomial s and a noticeable function ε (i.e., $\varepsilon(n) > 1/p(n)$ for some positive polynomial p), we prove by induction on i that circuits of size $s(n)$ cannot compute $P^{(i)}(U_{i\cdot n})$ with success probability greater than $(1-(1/p(n))^i + i\cdot\varepsilon(n)$. (The induction basis is guaranteed by the theorem's hypothesis.) The induction step is proved using Lemma 7 with $F_1 = P^{(i-1)}$ and $F_2 = f$ (along with $\rho_1((i-1)n) = (1-(1/p(n))^{i-1} + (i-1)\cdot\varepsilon(n)$, $s_1((i-1)n) = s(n)$, $\rho_2(n) = 1-(1/p(n))$ and $s_2(n) = \mathrm{poly}(n/\varepsilon(n))\cdot s(n)$). In particular, we use again the theorem's hypothesis regarding f, and note that $((1-(1/p(n))^{i-1} + (i-1)\cdot\varepsilon(n))\cdot(1-(1/p(n)) + \varepsilon(n)$ is upper-bounded by $(1-(1/p(n))^i + i\cdot\varepsilon(n)$. Thus, no $s(n)$-size circuit can compute $P^{(t(n))}(U_{t(n)\cdot n})$ with success probability greater than $(1-(1/p(n))^{t(n)} + t(n)\cdot\varepsilon(n) = \exp(-n) + t(n)\cdot\varepsilon(n)$.

Proof of Lemma 7: Proceeding (as usual) by the contrapositive, we consider a family of $s(\cdot)$-size circuits $\{C_n\}_{n\in\mathbb{N}}$ that violates the lemma's conclusion; that is, $\Pr[C_n(X_n) = F(X_n)] > \rho(n)$. We will show how to use such circuits in order to obtain either circuits that violate the lemma's hypothesis regarding F_1 or circuits that violate the lemma's hypothesis regarding F_2. Towards this end, it is instructive to write the success probability of C_n in a conditional form, while denoting the $i^{\rm th}$ output of $C_n(x)$ by $C_n(x)_i$ (i.e., $C_n(x) = (C_n(x)_1, C_n(x)_2)$):

$$\Pr[C_n(Y_{\ell(n)}, Z_{n-\ell(n)})=F(Y_{\ell(n)}, Z_{n-\ell(n)})]$$

$$\begin{aligned} &= \Pr[C_n(Y_{\ell(n)}, Z_{n-\ell(n)})_1 = F_1(Y_{\ell(n)})] \\ &\quad \cdot \Pr[C_n(Y_{\ell(n)}, Z_{n-\ell(n)})_2 = F_2(Z_{n-\ell(n)}) \mid C_n(Y_{\ell(n)}, Z_{n-\ell(n)})_1 = F_1(Y_{\ell(n)})]. \end{aligned}$$

The basic idea is that if the first factor is greater than $\rho_1(\ell(n))$ then we derive a circuit contradicting the lemma's hypothesis regarding F_1, whereas if the second factor is significantly greater than $\rho_2(n - \ell(n))$ then we derive a circuit contradicting the lemma's hypothesis regarding F_2. The basic idea for the latter case is that a sufficiently large sample of $(Y_{\ell(n)}, F_1(Y_{\ell(n)}))$, which may be hard-wired into the circuit, allows using the conditional probability space (in such a circuit) towards an attempt to approximate F_2. This may work provided the condition holds with noticeable probability. The last caveat motivates a separate treatment of z's with noticeable $\Pr[C_n(Y_{\ell(n)}, z)_1 = F_1(Y_{\ell(n)})]$ and of the rest.

Let us first simplify the notations by fixing a generic n and using the abbreviations $C = C_n$, $\varepsilon = \varepsilon(n)$, $\ell = \ell(n)$, $Y = Y_\ell$, and $Z = Y_{n-\ell}$. We call z good if $\Pr[C(Y, z)_1 = F_1(Y)] \geq \varepsilon/2$ and let G be the set of good z's. Then, we upper-bound the success probability of C by $\Pr[C(Y, Z) = F(Y, Z) \wedge Z \in G] + \varepsilon/2$, where the bound follows by observing that for any $z \notin G$:

$$\Pr[C(Y, z) = F(Y, z)] \leq \Pr[C(Y, z)_1 = F_1(Y)] < \varepsilon/2 .$$

Thus, using $\Pr[C(Y, z) = F(Y, z)] > \rho(n) = \rho_1(\ell) \cdot \rho_2(n - \ell) + \varepsilon$, we have

$$\Pr[C(Y, Z) = F(Y, Z) \wedge Z \in G] > \rho_1(\ell) \cdot \rho_2(n - \ell) + \frac{\varepsilon}{2}. \tag{2}$$

We proceed according to the forgoing outline, first showing that if $\Pr[C(Y, Z)_1 = F_1(Y)] > \rho_1(\ell)$ then we derive circuits violating the hypothesis concerning F_2. Actually, we prove something stronger (which we will actually need for the other case).

Claim 7.1: For every z, it holds that $\Pr[C(Y, z)_1 = F_1(Y)] \leq \rho_1(\ell)$.

Proof: Otherwise, using any $z \in \{0,1\}^{n-\ell}$ that satisfies $\Pr[C(Y, z)_1 = F_1(Y)] > \rho_1(\ell)$, we obtain a circuit $C'(y) \stackrel{\text{def}}{=} C(y, z)_1$ that contradicts the lemma's hypothesis concerning F_1. □

Using Claim 7.1, we show how to obtain a circuit that violates the lemma's hypothesis concerning F_2, and doing so we complete the proof of the lemma.

Claim 7.2: There exists a circuit C'' of size $s_2(n - \ell)$ such that

$$\begin{aligned} \Pr[C''(Z) = F_2(Z)] &\geq \frac{\Pr[C(Y, Z) = F(Y, Z) \wedge Z \in G]}{\rho_1(\ell)} - \frac{\varepsilon}{2} \\ &> \rho_2(n - \ell) \end{aligned}$$

Proof: The second inequality is due to Eq. (2), and thus we focus on establishing the first one. We construct the circuit C'' as suggested in the foregoing outline. Specifically, we take a poly(n/ε)-large sample, denoted S, from the distribution $(Y, F_1(Y))$ and let $C''(z) \stackrel{\text{def}}{=} C(y, z)_2$, where (y, v) is a uniformly selected among the elements of S for which $C(y, z)_1 = v$ holds. Details follow.

Let S be a sequence of $m \stackrel{\text{def}}{=} \text{poly}(n/\varepsilon)$ pairs, generated by taking m independent samples from the distribution $(Y, F_1(Y))$. We stress that we do not assume here that such a sample can be produced by an efficient (uniform) algorithm (but, jumping ahead, we remark that such a sequence can be fixed non-uniformly). For each $z \in G \subseteq \{0,1\}^{n-\ell}$, we denote by S_z the set of pairs $(y, v) \in S$ for which $C(y, z)_1 = v$. Note that S_z is a random sample for the residual probability space defined by $(Y, F_1(Y))$ conditioned on $C(Y, z)_1 = F_1(Y)$. Also, with overwhelmingly high probability, $|S_z| = \Omega(n/\varepsilon^2)$, because $z \in G$ implies $\Pr[C(Y, z)_1 = F_1(Y)] \geq \varepsilon/2$ and $m = \Omega(n^2/\varepsilon^3)$.

Thus, for each $z \in G$, with overwhelming probability taken over the choices of S, the sample S_z provides a good approximation to the conditional probability space. In particular, with probability greater than $1 - 2^{-n}$, it holds that

$$\frac{|\{(y,v) \in S_z : C(y,z)_2 = F_2(z)\}|}{|S_z|} \geq \Pr[C(Y,z)_2 = F_2(z) \mid C(Y,z)_1 = F_1(Y)] - \frac{\varepsilon}{2} \tag{3}$$

Thus, with positive probability, Eq. (3) holds for all $z \in G \subseteq \{0,1\}^{n-\ell}$. The circuit C'' computing F_2 is now defined as follows. A set $S = \{(y_i, v_i) : i = 1, ..., m\}$ satisfying Eq. (3) for all good z's is "hard-wired" into the circuit C''. (In particular, S_z is not empty for any good z.) On input z, the circuit C'' first determines the set S_z, by running C for m times and checking, for each $i = 1, ..., m$, whether or not $C(y_i, z) = v_i$. In case S_z is empty, the circuit returns an arbitrary value. Otherwise, the circuit selects uniformly a pair $(y, v) \in S_z$ and outputs $C(y, z)_2$. (The latter random choice can be eliminated by a standard averaging argument.) Using the definition of C'', Eq. (3), and Claim 7.1, we have:

$$\begin{aligned}
\Pr[C''(Z) = F_2(Z)] &\geq \sum_{z \in G} \Pr[Z = z] \cdot \Pr[C''(z) = F_2(z)] \\
&= \sum_{z \in G} \Pr[Z = z] \cdot \frac{|\{(y,v) \in S_z : C(y,z)_2 = F_2(z)\}|}{|S_z|} \\
&\geq \sum_{z \in G} \Pr[Z = z] \cdot \left(\Pr[C(Y,z)_2 = F_2(z) \mid C(Y,z)_1 = F_1(Y)] - \frac{\varepsilon}{2} \right) \\
&= \sum_{z \in G} \Pr[Z = z] \cdot \left(\frac{\Pr[C(Y,z)_2 = F_2(z) \wedge C(Y,z)_1 = F_1(Y)]}{\Pr[C(Y,z)_1 = F_1(Y)]} - \frac{\varepsilon}{2} \right) \\
&\geq \left(\sum_{z \in G} \Pr[Z = z] \cdot \frac{\Pr[C(Y,z) = F(Y,z)]}{\rho_1(\ell)} \right) - \frac{\varepsilon}{2}
\end{aligned}$$

where the last inequality is due to Claim 7.1. The claim follows. □

This completes the proof of the lemma. ■

Comments. Firstly, we wish to call attention to the care with which an inductive argument needs to be carried out in the computational setting, especially when a non-constant number of inductive steps is concerned. Indeed, our inductive proof of Theorem 6 involves invoking a quantitative lemma that allows to keep track of the relevant quantities (e.g., success probability and circuit size) throughout the induction process. Secondly, we mention that Lemma 7 (as well as Theorem 6) has a uniform complexity version that assumes that one can efficiently sample the distribution $(Y_{\ell(n)}, F_1(Y_{\ell(n)}))$ (resp., $(U_n, f(U_n))$). For details see [4]. Indeed, a good lesson from the proof of Lemma 7 is that non-uniform circuits can "effectively sample" any distribution. Lastly, we mention that Theorem 5 (Yao's XOR Lemma) also has a (tight) quantitative version (see, e.g., [4, Sec. 3]).

3 Notes

Like several other fundamental insights attributed to Yao's paper [5], Yao's XOR Lemma (Theorem 5) is not even stated in [5] but is rather due to Yao's oral presentations of his paper. The first published proof of Yao's XOR Lemma was given by Levin (see [4, Sec. 3]). Levin's proof is the only one known giving a tight quantitative analysis (on the decrease in the level of approximability), and the interested reader is referred to it (via the non-laconic presentation of [4, Sec. 3]). The proof presented in Section 2 is due to Goldreich, Nisan and Wigderson [4, Sec. 5].

Exercise 8 Let $\hat{f}$ be as in the conclusion of Theorem 2. Prove that there exists a Boolean function g in $\mathcal{E}$ that is (p, ε)-inapproximable for every polynomial p and for $\varepsilon(n) = 1/n^3$.
(Hint: consider the function g defined such that $g(x, i)$ equals the i^{th} bit of $\hat{f}(x)$.)

Exercise 9 Let f be a Boolean function, and $b(y, r)$ denote the inner-product modulo 2 of the equal-length strings y and r. Suppose that $F'(x_1, ..., x_{t(n)}, r) \stackrel{\text{def}}{=} b(f(x_1) \cdots f(x_{t(n)}), r)$, where $x_1, ..., x_{t(n)} \in \{0,1\}^n$ and $r \in \{0,1\}^{t(n)}$, is T-inapproximable for every polynomial T. Assuming that $n \mapsto t(n) \cdot n$ is 1-1, prove that $F(x) \stackrel{\text{def}}{=} F'(x, 1^{t'(|x|)})$, where $t'(t(n) \cdot n) = t(n)$, is T-inapproximable for every polynomial T.

Guideline: Reduce the approximation of F' to the approximation of F. An important observation is that for any $x = (x_1, ..., x_{t(n)})$, $x' = (x'_1, ..., x'_{t(n)})$, and $r = r_1 \cdots r_{t(n)}$ such that $x'_i = x_i$ if $r_i = 1$, it holds that $F'(x, r) = F(x') \oplus \oplus_{i:r_i=0} f(x'_i)$. Note that the equality holds regardless of the choice of the string $x'_i \in \{0,1\}^n$ for which $r_i = 0$. Also note that the suggested reduction requires knowledge of $\sigma = \oplus_{i:r_i=0} f(x'_i)$, but in our context the reduction may be performed by a small non-uniform circuit, which may incorporate the values of $f(z)$'s for a small number of z's. Indeed, for uniformly chosen $z_1, ..., z_{t(n)} \in \{0,1\}^n$, we use these z_i's as well as the $f(z_i)$'s as advice to the reduction. On input $x_1, ..., x_{t(n)}, r_1 \cdots r_{t(n)}$, the reduction sets $x'_i = x_i$ if $r_i = 1$ and $x'_i = z_i$ otherwise, makes the query $x' = (x'_1, ..., x'_{t(n)})$ to F, and returns $F(x') \oplus_{i:r_i=0} f(z_i)$.

References

[1] L. Babai, L. Fortnow, N. Nisan and A. Wigderson. BPP has Subexponential Time Simulations unless EXPTIME has Publishable Proofs. *Complexity Theory*, Vol. 3, pages 307–318, 1993.

[2] O. Goldreich. *Foundation of Cryptography: Basic Tools.* Cambridge University Press, 2001.

[3] O. Goldreich and L.A. Levin. Hard-core Predicates for any One-Way Function. In *21st ACM Symposium on the Theory of Computing*, pages 25–32, 1989.

[4] O. Goldreich, N. Nisan and A. Wigderson. On Yao's XOR-Lemma. *ECCC*, TR95-050, 1995.

[5] A.C. Yao. Theory and Application of Trapdoor Functions. In *23rd IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982.