2013 IEEE 54th Annual Symposium on Foundations of Computer Science

Nondeterministic Direct Product Reductions and the Success Probability of SAT Solvers

Andrew Drucker
School of Mathematics
Institute for Advanced Study
Princeton, NJ 08540
Email: andy.drucker@gmail.com

***Abstract*—We give two nondeterministic reductions which yield new direct product theorems (DPTs) for Boolean circuits. In these theorems one assumes that a target function f is mildly hard against *nondeterministic* circuits, and concludes that the direct product $f^{\otimes t}$ is extremely hard against (only polynomially smaller) probabilistic circuits. The main advantage of these results compared with previous DPTs is the strength of the size bound in our conclusion.**

As an application, we show that if NP is not in $\mathsf{coNP/poly}$ then, for every PPT algorithm attempting to produce satisfying assigments to Boolean formulas, there are infinitely many instances where the algorithm's success probability is nearly-exponentially small. This furthers a project of Paturi and Pudlák [STOC'10].

***Keywords*-hardness amplification; direct product theorems; Satisfiability**

I. Introduction

This work contributes to two central areas of study in complexity theory: *hardness amplification* on the one hand, and *the complexity of* NP *search problems* on the other.

A. Hardness amplification and direct product theorems

In the general hardness amplification project, we assume that we have identified a function f that is "mildly hard" to compute, for some class $\mathcal{C}$ of resource-bounded algorithms. Our goal is to derive a second function f' that is "extremely hard" to compute, for some possibly-different class $\mathcal{C}'$. In our initial discussion we will focus on the case where $f : \{0,1\}^n \to \{0,1\}^d$, $f' : \{0,1\}^{n'} \to \{0,1\}^{d'}$ are finite functions, and $\mathcal{C}, \mathcal{C}'$ are two sets of probabilistic Boolean circuits, but we note that the project can be studied in other models as well.

The notion of difficulty suggested above can be formalized in two ways (both relevant to our work). Let $p \in [0,1]$. In the *average-case* setting, let us say that f is *p-hard for* $\mathcal{C}$ with respect to an input distribution $\mathcal{D}$ if every algorithm in $\mathcal{C}$ computes $f(\mathbf{x})$ correctly with probability at most p on an input $\mathbf{x} \sim \mathcal{D}$. In the *worst-case* setting, we say that f is *worst-case p-hard* for a class $\mathcal{C}$ of randomized algorithms if, for every algorithm $C \in \mathcal{C}$, there is an input x such that $\Pr_C[C(x) = f(x)] \leq p$. In either setting, from a "mild" hardness guarantee $p = 1 - \varepsilon$ for $\mathcal{C}$ in computing f, we want to obtain a much stronger bound $p' \ll 1$ for the second class $\mathcal{C}'$ in computing f'.

There are several motivations to pursue hardness amplification. First, the security of most modern cryptographic primitives, such as one-way functions and public-key cryptosystems, inherently requires the existence of computational problems, solvable in NP, which possess a strong average-case hardness guarantee. While the mere existence of hard-on-average problems in NP is not known to be *sufficient* for doing cryptography, a better understanding of the sources of average-case hardness seems necessary for making progress in the foundations of cryptography. Moreover, hardness-amplification techniques in complexity theory have also helped to inspire tools for the *security-amplification* of cryptographic primitives. (See, e.g., [1] for background on the complexity-theoretic underpinnings of modern cryptography and on ideas of security amplification.)

Second, average-case hardness is also inherent in the fundamental concept of *pseudorandomness*: a pseudorandom source is information-theoretically distinguishable from random bits, yet the distinguishing task must be computationally hard-on-average. Techniques of hardness amplification have played a key role in important constructions of pseudorandom generators [2], [3].

Third, hardness amplification, and in particular the *direct product* approach to amplifying hardness, is interesting in its own right, and helps us to critically examine some of our most basic intuitions about computation, as we will describe.

Given a function f as above and $t \in \mathbb{N}^+$, let $f^{\otimes t} : \{0,1\}^{n \times t} \to \{0,1\}^{d \times t}$, the *$t$-fold direct product of f*, be the function which takes t length-n inputs $(x^1, \ldots, x^t)$ and outputs $(f(x^1), \ldots, f(x^t))$. A *direct product theorem (DPT)* is any result upper-bounding the success bound p' for computing $f^{\otimes t}$ in terms of an assumed success bound p for f (and, possibly, other parameters).

When f is Boolean, the direct product construction can be considered along with the related "t-fold XOR" $f^{\oplus t}(x^1, \ldots, x^t) := \bigoplus f(x^j)$ (i.e., the sum mod 2). An "XOR lemma" is any result upper-bounding the success bound p' for computing $f^{\oplus t}$ in terms of an assumed success bound p for f. The original XOR lemma was stated by

0272-5428/13 $26.00 © 2013 IEEE
DOI 10.1109/FOCS.2013.84

Yao in unpublished work, with the first published proof due to Levin [4]; see [5] for more information on the lemma's history and several proofs. Unlike the direct product, the t-fold XOR $f^{\oplus t}$ is itself a Boolean function, which can be advantageous in applications of hardness amplification, but which can also be a disadvantage since this limits its average-case hardness (an algorithm may compute $f^{\oplus t}$ with success probability 1/2 by guessing a random bit). Several works show how in some settings XOR lemmas may be obtained from DPTs [6], [5] or vice versa [7], [8]. We will not prove or use XOR lemmas in this work; we merely point out that their study is often intimately linked with the study of direct product theorems.

The motivation for the direct product construction is as follows. Let $\mathcal{C}_{\leq s}$ be the class of probabilistic Boolean circuits of size at most s. It would *appear* that, if any circuit $C \in \mathcal{C}_{\leq s}$ has success probability at most $p < 1$ in computing f, then any circuit $C' \in \mathcal{C}_{\leq s}$ should have success probability not much more than p^t in computing $f^{\otimes t}$. The intuition here is that combining the t "unrelated" computations should not help much, and simply trying to solve each instance separately would be a nearly-optimal strategy.

This hypothesis can be considered in both the worst-case and the average-case setting; in the latter, if f is p-hard with respect to inputs drawn from $\mathcal{D}$, then it is natural to study the difficulty of computing $f^{\otimes t}$ with t inputs drawn independently from $\mathcal{D}$.[1]

One might even be tempted to make the bolder conjecture that $f^{\otimes t}$ is p^t-hard against circuits in $\mathcal{C}_{\leq t \cdot s}$, but this was shown by Shaltiel to fail badly in the average-case setting [10]. So what kind of DPT is known to hold in the circuit model? A version of the following theorem, with slightly weaker parameters, was proved in [5, first version in '95] ; a tighter argument leading to the bounds given below is described in [11]. In the form of the DPT stated below, the focus is on getting a success probability bound of at most some ε for $f^{\otimes t}$, where t is chosen accordingly.

Theorem I.1 (see [5], Lemma 7, and [11], Theorems 2.9, 2.10)**.** *Suppose the Boolean function $f : \{0,1\}^n \to \{0,1\}$ is p-hard for circuits of size s with respect to input distribution $\mathcal{D}$, for some $p \in [.5, 1)$. For $\varepsilon \in (0, .25)$ and an appropriate $t = \Theta\left(\frac{\ln(1/\varepsilon)}{1-p}\right)$, the function $f^{\otimes t}$ is ε-hard with respect to $\mathcal{D}^{\otimes t}$ for circuits with size bounded by s', where $s' = \Theta(\frac{\varepsilon \cdot s}{\ln(1/\varepsilon)})$.*

This is a powerful and elegant result, but one whose parameters can be disappointing in many situations. The size bound s' degrades quickly as $\varepsilon \to 0$; if $s \leq \mathrm{poly}(n)$, i.e., if our initial hardness assumption is against circuits of some fixed-polynomial size (and $p = 2/3$, say), then Theorem I.1 cannot give a success bound of $n^{-\omega(1)}$ against any nontrivial class of circuits. In unpublished work, Steven Rudich has observed that this barrier is inescapable for a certain class of "black-box," relativizing, *deterministic* (or probabilistic) reductions (see [5] for more discussion). The limitations of more general black-box hardness-amplification reductions have been extensively studied, particularly for the case of XOR lemmas and other reductions that produce a Boolean function (for an overview, see [12]).

[1] "Derandomized" variants of the scenario, in which the t inputs are not fully independent, have also been studied, and powerful "derandomized" DPTs were obtained, notably in [3], [9]. A new derandomized DPT will appear in the full version of our paper.

B. Our new direct product theorems

In this work we show that, if we are willing to assume that our starting function f is somewhat hard to compute by *nondeterministic* circuits, then we obtain very strong hardness results for $f^{\otimes t}$ against the class of probabilistic circuits. The direct product theorems we prove have quantitative parameters that are far superior to those in Theorem I.1.

Our first direct product theorem holds for the worst-case setting. We show:

Theorem I.2. *Let $f = \{f_n\}$ be a family of Boolean functions on n input bits, and suppose that $f \notin \mathsf{NP/poly} \cap \mathsf{coNP/poly}$.*

Now let $1 < t(n) \leq \mathrm{poly}(n)$ be a parameter, and let $\{C_n\}_{n>0}$ be any family of polynomial-size, probabilistic circuits outputting $t(n)$ bits. Then for infinitely many choices of n and $x \in \{0,1\}^{n \times t(n)}$, $\Pr[C_n(x) = f_n^{\otimes t(n)}(x)] < \exp(-\Omega(t(n)))$.

Like all known DPTs in the circuit setting, this result is proved in its contrapositive form, using a *nondeterministic direct product reduction*—a method for transforming a probabilistic circuit that weakly approximates $f_n^{\otimes t(n)}$ into a *nondeterministic* circuit that computes f_n with much greater confidence. In the reduction used to prove Theorem I.2, we get a nondeterministic circuit that computes f_n exactly. This transformation incurs only a polynomial blowup in circuit size. The reduction is black-box but, due to its use of nondeterminism, is not subject to the limitations identified by Rudich.

We also prove a DPT for the average-case setting, for input distributions that are efficiently sampleable.

Theorem I.3. *Let $\{f_n\}$ be a family of Boolean functions on n input bits. Let $\mathcal{D} = \{\mathcal{D}_n\}$ be a family of input distributions, sampleable by a polynomial-size family of circuits. Suppose that, for all families $\{G_n\}$ of polynomial-size nondeterministic circuits, for suff. large n, $\Pr_{\mathbf{x} \sim \mathcal{D}_n}[G_n(\mathbf{x}) =$*

$f_n(\mathbf{x})] < 2/3$.[2] *Now let* $1 < t(n) \leq \mathrm{poly}(n)$ *be a parameter, and let* $\{C_n\}_{n>0}$ *be any family of polynomial-size, probabilistic circuits outputting* $t(n)$ *bits. Then for suff. large* n *and* $\mathbf{x} \sim \mathcal{D}_n^{\otimes t(n)}$, *we have* $\Pr[C_n(\mathbf{x}) = f_n^{\otimes t(n)}(\mathbf{x})] < \exp(-\Omega(\sqrt{t(n)}))$. [3]

Our nondeterministic direct product reductions are not the first use of nondeterministic reductions in the study of average-case complexity. In particular, previous authors have exploited nondeterminism to give worst-case to average-case reductions for computational problems. In [13, first version in '92], Feige and Lund gave a nondeterministic worst-case to average-case reduction for computing the Permanent over large fields, showing that this problem is exponentially hard. Earlier Amir, Beigel, and Gasarch [14] had shown related results for #SAT. We also mention an extractor construction of Trevisan and Vadhan which used a nondeterministic worst-case to average-case reduction for computing low-degree polynomials over finite fields [16, Lem. 4.1]. None of these works give a direct product reduction in our sense. However, [14] considers t-fold direct products $f^{\otimes t}$ and gives complexity upper bounds for functions f such that $f^{\otimes t}$ can be computed using few queries to an oracle; some of these reductions use nondeterminism.[4]

C. Application to the success probability of PPT SAT solvers

In a recent work, Paturi and Pudlák [17] asked about the achievable worst-case success probability of PPT heuristics for producing satisfying assignments to Boolean circuits. They argue for the importance of this question by observing that many of the known exact algorithms for NP search problems, which achieve exponential runtimes with an improved exponent over naive search, can be converted to *polynomial-time* search heuristics with a success probability attaining nontrivial advantage over random guessing. Thus, exploring the limitations of PPT search heuristics also illuminates the limitations of a very natural paradigm for exponential-time computation.

Paturi and Pudlák show the following powerful result (a special case of a more general theorem): Suppose there is a $\gamma < 1$ and a PPT algorithm P_{solver} that, when given as input a description of any satisfiable Circuit-SAT instance Ψ with r variables, outputs a satisfying assignment to Ψ with probability at least $q(r) := 2^{-\gamma r}$. Then, there is a deterministic circuit family $\{C_r\}_r$ that succeeds at the same task on r-variable instances of size $n \leq \mathrm{poly}(r)$ with probability 1, and C_r is of size at most 2^{r^μ} for some $\mu < 1$.

In this work, we exploit a simple connection between direct product computations for the satisfiability decision problem for Boolean formulas, and the task of producing satisfying assignments to such formulas:

($\star$) If $\psi^1, \ldots, \psi^t$ are formulas, and $s \in [0, t]$ is a correct guess for the number of satisfiable ψ^js, then the 0/1 values $[\psi^1 \in SAT], \ldots, [\psi^t \in SAT]$ can all be inferred given any satisfying assignment to the formula $\Psi^{(s)}$ which asks for satisfying assignments to at least s of the ψ^js.

(This observation has been used earlier in [18] to prove the result $\mathsf{P}_{||}^{\mathsf{NP}} = \mathsf{P}^{\mathsf{NP}[\log]}$.) Using this connection together with our worst-case DPT, we prove the following result, bounding the achievable success probability of polynomial-time heuristics for SAT solvers under the standard complexity-theoretic assumption that NP is not contained in coNP/poly.

Theorem I.4. *Let* $\gamma \in (0, 1)$. *Suppose there is a PPT algorithm* P_{solver} *that, when given as input a description of a satisfiable 3-CNF formula* Φ, *of description length* $|\langle\Phi\rangle| = N$, *outputs a satisfying assignment to* Φ *with probability at least* $q(N) := 2^{-N^\gamma}$.

Then, NP $\subseteq$ coNP/poly *(and the Polynomial Hierarchy collapses to* Σ_3^p*).*

This theorem is incomparable to the result of Paturi and Pudlák. Their result implies a stronger upper bound for the success probability of Circuit-SAT solvers, at the cost of assuming that NP has nearly-exponential circuit complexity. (Also, their work does not give strong limitations on solvers for the special case of 3-CNF inputs.)

We are not aware of past work on hardness amplification using ($\star$); but many works have used connections between direct product theorems and the difficulty of NP search problems. Hardness amplification has also been previously applied to NP *decision problems* to obtain hard-on-average NP *search problems*, e.g., in our previous work [22, Thm. 5]. Also, a great deal of work aims at amplifying average-case hardness *within* the class of NP decision problems (see [24]).

D. Our techniques

While there are major technical differences between the reductions used to prove Theorems I.2 and I.3, both begin with the same high-level intuition, which we'll describe here. Let f be a Boolean function on n input bits, and $\mathcal{D}$ a distribution over inputs to f. Suppose C is a probabilistic circuit that computes $f^{\otimes t}$ with some success probability $q > 2^{-ct}$ on inputs $\overline{\mathbf{x}} = (\mathbf{x}^1, \ldots, \mathbf{x}^t) \sim \mathcal{D}^{\otimes t}$, for some small constant $0 < c \ll 1$. We would like to obtain from C a nondeterministic circuit computing f with high success probability with respect to $\mathcal{D}$. (To prove Theorem I.2, it will

[2] In the most common definition, a nondeterministic circuit $G_n(x)$ is said to *compute* a function $g(x)$ if the following condition holds: $g(x) = 1$ iff there exists some setting of the nondeterministic gates of G_n causing it to accept input x. Our Theorem I.3 is valid if our initial hardness assumption on f_n is with respect to this definition. However, for our results we actually only require hardness with respect to a more restrictive model of nondeterministic computation in which, to compute $g(x)$ correctly, G_n must (a) output the correct value on some nondeterministic branch on input x, and (b) on every branch on x, always output either $g(x)$ or "fail."

[3] In the full version we anticipate a better dependence on t in the conclusion.

[4] As we will discuss in the full version, results from [14] can be used to derive DPTs, but much weaker ones than we give.

suffice to show how to do this for every $\mathcal{D}$. We can then use the minimax theorem and majority-voting in a standard way, to build a nondeterministic circuit that is correct on *every* input.)

Say that an execution of C is *j-valid*, for $0 \leq j \leq t$, if it correctly computes f on its first j inputs. We obviously have

$$\Pr\left[C(\overline{\mathbf{x}}) = f^{\otimes t}(\overline{\mathbf{x}})\right] = \prod_{j \in [t]} \Pr[j\text{-valid}|(j-1)\text{-valid}] > 2^{-ct}.$$

Thus, for a *typical* choice of index j, $\Pr[j\text{-valid}|(j-1)\text{-valid}] \approx 2^{-c} \approx 1$. This motivates us to choose such an index $j = j^*$, and to *fix* some settings $y^1, \ldots, y^{j^*-1}$ to the first (j^*-1) inputs. Then, by storing the values $f(y^1), \ldots, f(y^{j^*-1})$, we can easily *recognize* a (j^*-1)-valid execution (as specified by a setting to the remaining inputs and to the random bits used by C). Now, given a single input $\mathbf{x} \sim \mathcal{D}$ on which we wish to compute f, we will try to obtain a (j^*-1)-valid execution of C on an input-tuple whose first j^* elements are $y^1, \ldots, y^{j^*-1}, \mathbf{x}$. The idea is that, by our choice of j^*, a "typical" such execution (in which the remaining inputs are drawn from $\mathcal{D}^{\otimes(t-j^*)}$) should also be j^*-valid, and give us the value $f(\mathbf{x})$. This basic strategy can be seen in [5] and other previous direct product reductions. In our reduction, however, nondeterminism will allow us to obtain a (j^*-1)-valid execution of C very efficiently, even when such executions are extremely rare; we simply need to "guess and check."

This approach requires care, however, because an execution obtained by nondeterministic guessing need not be a representative sample of the population from which it was drawn. Thus, even if we successfully fix values of $y^1, \ldots, y^{j^*-1}$ and receive an input $\mathbf{x} \sim \mathcal{D}$ such that most (j^*-1)-valid executions are also j^*-valid, we need a way to "kill off" the "atypical" (j^*-1)-valid executions which fail to be j^*-valid.

For this task, a natural idea is to try to apply *random hashing*, a well-established tool for reducing the size of a set and culling atypical elements. The use of randomly selected hash functions for such purposes, in conjunction with nondeterministic guessing, was pioneered by Goldwasser and Sipser [25] (with related techniques found in [26]).[5] This technique has an important requirement, however: to kill all the "atypical" (j^*-1)-valid executions while simultaneously leaving at least one j^*-valid execution alive, we need to know a good approximation to the *probability* of a (j^*-1)-valid execution, conditioned on $y^1, \ldots, y^{j^*-1}, \mathbf{x}$. We want this probability to be predictable with high accuracy based on $y^1, \ldots, y^{j^*-1}$ alone, without any foreknowledge of the input $\mathbf{x}$, so that a good approximation can be encoded into as helpful advice. To summarize, we hope to find and fix inputs $y^1, \ldots, y^{j^*-1}$, such that with high probability over $\mathbf{x} \sim \mathcal{D}$, we have the "stability" conditions:

(i) $\Pr[j^*\text{-valid}|y^1, \ldots, y^{j^*-1}, \mathbf{x}] \approx \Pr[(j^*-1)\text{-valid}|y^1, \ldots, y^{j^*-1}, \mathbf{x}]$;

(ii) $\Pr[(j^*-1)\text{-valid}|y^1, \ldots, y^{j^*-1}, \mathbf{x}] \approx \Pr[(j^*-1)\text{-valid}|y^1, \ldots, y^{j^*-1}]$.

(We will tolerate a $(1 \pm .02)$ multiplicative error above.)

So how do we choose the values $y^1, \ldots, y^{j^*-1}$? The obvious idea is to choose them as independent samples from $\mathcal{D}$, after selecting j^* uniformly at random. However, this approach may *fail* to guarantee condition (i) above, if the successful direct-product computations of C are "concentrated" in a pathological way. For example, it may be that $C(x^1, \ldots, x^t)$ always outputs $f^{\otimes t}(x^1, \ldots, x^t)$ iff the first input x^1 lies in some "good" set $G \subset \{0,1\}^n$ of probability mass $\approx 2^{-ct}$, while if $x^1 \notin G$, then C simply outputs random (or false) guesses. In this case, conditions (i) and (ii) can fail for a typical setting $x^1 := y^1$.

We address these difficulties in two distinct ways in our two direct product reductions. In our worst-case reduction (Theorem I.2), we assume from the start that our C has a probability $\geq q$ of computing $f^{\otimes t}$ under *every* input. This assumption turns out to be very useful in analyzing the effect of conditioning on on $y^1, \ldots, y^{j^*-1} \sim \mathcal{D}$ and showing that, for randomly chosen j^*, we obtain conditions (i) and (ii) above.

In our average-case reduction, we use two additional ideas. First, we re-index the t inputs according to a *randomly chosen* permutation, which helps to "smooth out" the effects of conditioning. Second, we choose $y^1, \ldots, y^{j^*-1}$, not independently from $\mathcal{D}$ as before, but according to the distribution induced by conditioning on (j^*-1)-validity. These choices help ensure conditions (i) and (ii), at the cost of a more complicated (information-theoretic) analysis.

After fixing $y^1, \ldots, y^{j^*-1}$, in our direct product reduction for *sampleable* distributions, we can perform hashing over all possible outcomes to $x^{j^*+1}, \ldots, x^t$, weighted by their likelihood under $\mathcal{D}$. In our *worst-case* direct product reduction, $\mathcal{D}$ may not be efficiently sampleable, which poses an additional challenge. In this setting we show that in fact, it is adequate to draw $x^{j^*+1}, \ldots, x^t$ independently at random from multisets $S_{j^*+1}, \ldots, S_t$, each obtained by sampling $\mathrm{poly}(n)$ times from $\mathcal{D}$. These "sparsified" versions of $\mathcal{D}$ can be coded into our circuit. The idea of this sparsification and its analysis are somewhat similar to (and inspired by) a step from our previous paper [28, Lem. 6.3].

Due to space limitations, in this extended abstract we omit full proofs, and focus only on describing how the stability conditions (i)-(ii) are obtained in the proof of Theorem I.2.

[5] We are aiming to build a nondeterministic circuit, not a probabilistic one; but the eventual plan will be to fix a polynomial number of representative hash functions as non-uniform advice. Let us also mention that hash families were used in Paturi and Pudlák's work [17] as well, but in a very different way. Those authors used hash functions to *reduce the amount of randomness* used by a SAT solver having a worst-case success probability guarantee, as a step toward transforming a Circuit-SAT instance into an equivalent instance with *fewer variables*.

II. Stage-based analysis of direct-product computations

Throughout the rest of the paper, fix a function $f : \{0,1\}^n \to \{0,1\}^d$ (no longer assumed Boolean, as in the Introduction—our techniques apply to the non-Boolean case as well) and a value $t > 1$. We say that a probabilistic circuit C on $n \times t$ input bits is a *q-worst-case direct product solver for $f^{\otimes t}$* if, for all input tuples $\overline{x}$, $\Pr[C(\overline{x}) = f^{\otimes t}(\overline{x})] \geq q$. In this section we analyze the behavior of worst-case direct-product solvers on inputs $(x^1, \ldots, x^t)$ drawn from a known probability distribution $\overline{\mathcal{D}}$ over $\{0,1\}^{n \times t}$. The case where $\overline{\mathcal{D}}$ is a t-fold product distribution will be of primary interest to us, although some of our lemmas will apply to non-product input distributions. Some notation: we use $a \in_r A$ to denote that a is uniformly sampled from multiset A.

Next we make some definitions that will be of central importance.

Definition II.1 (j-valid outputs)**.** *Let $(x^1, \ldots, x^t) \in \{0,1\}^{n \times t}$, and let $z = (z_1, \ldots, z_t) \in \{0,1\}^{d \times t}$. For $j \in [t]$, say that z is j-*valid for $(x^1, \ldots, x^t)$, with respect to $f^{\otimes t}$, *if $z_\ell = f(x^\ell)$, for all $\ell \leq j$. When the reference strings $x^1, \ldots, x^t$ are clear from the context, we will simply say that z is j-*valid*. Note that if z is j-valid for $(x^1, \ldots, x^t)$ then it is also j'-valid for $j' < j$. By convention, every $z \in \{0,1\}^{d \times t}$ is said to be 0-valid.*

If C is a probabilistic circuit taking a tuple of strings $(x^1, \ldots, x^t)$ as input (with each x^j of equal, predetermined length) and outputting a t-bit string, we say that a particular execution of C on input $(x^1, \ldots, x^t)$ is j-valid if it outputs some bitstring that is j-valid with respect to the inputs. We denote this event simply as $[C(x^1, \ldots, x^t)$ is j-valid $]$.

Definition II.2 (α,β sequences)**.** *Let $C : \{0,1\}^{n \times t} \to \{0,1\}^{d \times t}$ be a probabilistic circuit. Let $\overline{\mathcal{D}}$ be a distribution over $\{0,1\}^{n \times t}$, and let $(\mathbf{x}^1, \ldots, \mathbf{x}^t) \sim \overline{\mathcal{D}}$.*

Define two sequences of random variables $\alpha_0, \alpha_1, \ldots, \alpha_t$, $\beta_0, \beta_1, \ldots, \beta_{t-1}$, as follows. For $j \in [0,t]$, we let

$$\alpha_j \ := \ \Pr[C(\mathbf{x}^1, \ldots, \mathbf{x}^t) \text{ is } j\text{-valid } \big| \mathbf{x}^1, \ldots, \mathbf{x}^j] \ ,$$

with validity defined with respect to f and where the probability is taken over the randomness in $\mathbf{x}^1, \ldots, \mathbf{x}^t$ and in C's randomness. Similarly, for $j \in [0, t-1]$, define

$$\beta_j \ := \ \Pr[C(\mathbf{x}^1, \ldots, \mathbf{x}^t) \text{ is } j\text{-valid } \big| \mathbf{x}^1, \ldots, \mathbf{x}^{j+1}] \ .$$

We have $\alpha_0 = \beta_0 = 1$, all $\alpha_j, \beta_j \in [0,1]$, and $\alpha_{j+1} \leq \beta_j$. The following claim is proved in the full version.

Claim II.3. *1) The function $\ln(1+x)$, defined on $(-1, \infty)$, satisfies*

$$\ln(1+x) \ \leq \ \begin{cases} x - \frac{x^2}{6} & \text{if } x \in (-1,1) \ , \\ (\ln 2)x & \text{if } x \geq 1 \ . \end{cases}$$

Consequently, $\ln(1+x) \leq x - \min\{x^2/6, .3\}$.

Lemma II.4. *Let $Y_1, Y_2, \ldots, Y_T$ be (possibly dependent) nonnegative random variables satisfying $\mathbb{E}[Y_i] \leq 1$ for $i \in [T]$. Suppose that there is some $q \in (0,1]$ such that $Y_{\text{prod}} \geq q$ with probability 1. Let $Y_{\text{prod}} := \prod_{i \in [T]} Y_i$. Let $\mathbf{i} \in_r [T]$ be chosen independently of $Y_1, \ldots, Y_t$. Then,*

$$\Pr[Y_{\mathbf{i}} \in [.99, 1.01]] \ \geq \ 1 - 2^{16} \ln(1/q)/T \ .$$

Proof: Let $Z_1, \ldots, Z_T$ be defined by $Z_i := \ln(Y_i)$. Note that Z_i is well-defined since $Y_i > 0$, under our assumption $Y_{\text{prod}} \geq q$. Letting $Z_{\text{sum}} := \sum_{i \in [T]} Z_i$, note that $Z_{\text{sum}} = \ln(Y_{\text{prod}})$. Then $Z_{\text{sum}} \geq \ln q$. Thus,

$$\sum_{i \in [T]} \mathbb{E}[Z_i] \ = \ \mathbb{E}[Z_{\text{sum}}] \ \geq \ \ln q \ . \tag{1}$$

On the other hand, by applying Claim II.3, we find

$$Z_i \ \leq \ (Y_i - 1) - \min\left\{(Y_i - 1)^2/6, .3\right\} \ .$$

Taking expectations and using that $\mathbb{E}[Y_i] \leq 1$,

$$\mathbb{E}[Z_i] \ \leq \ -\mathbb{E}\left[\min\left\{(Y_i - 1)^2/6, .3\right\}\right] \ .$$

Let $p_i := \Pr[Y_i \notin [.99, 1.01]]$. Then

$$\mathbb{E}\left[\min\left\{(Y_i - 1)^2/6, .3\right\}\right] \ \geq \ p_i \cdot (.01)^2/6 \ ,$$

so that $\mathbb{E}[Z_i] \leq -p_i/2^{16}$. Combining this with Eq. (1) gives

$$\sum_{i \in [T]} p_i \ \leq \ -2^{16} \ln q \ = \ 2^{16} \ln(1/q) \ ,$$

which implies that

$$\Pr_{\mathbf{i} \in_r [T]}[Y_{\mathbf{i}} \notin [.99, 1.01]] \ = \ (1/T) \sum_{i \in [T]} p_i \ \leq 2^{16} \ln(1/q)/T \ .$$

■

Lemma II.5. *Let $C : \{0,1\}^{n \times t} \to \{0,1\}^{d \times t}$ be a probabilistic circuit. Suppose that C is a q-worst-case direct-product solver for $f^{\otimes t}$, for some $q \in (0,1]$.*

1) Let $\overline{\mathcal{D}}$ be a distribution over $\{0,1\}^{n \times t}$. Let $\overline{\mathbf{x}} = (\mathbf{x}^1, \ldots, \mathbf{x}^t) \sim \overline{\mathcal{D}}$. Let $\alpha_0, \ldots, \alpha_t, \beta_0, \ldots, \beta_{t-1}$ be as in Definition II.2, defined with respect to C and $\overline{\mathcal{D}}$. Let $\mathbf{j} \in_r [t]$ be sampled independently of $\overline{\mathbf{x}}$. Then with probability at least $1 - \frac{2^{16} \ln(1/q)}{t}$, we have

$$\alpha_{\mathbf{j}-1}, \beta_{\mathbf{j}-1} > 0 \ , \ \frac{\beta_{\mathbf{j}-1}}{\alpha_{\mathbf{j}-1}} \in [.99, 1.01] \ , \ \frac{\alpha_{\mathbf{j}}}{\beta_{\mathbf{j}-1}} \in [.99, 1] \ . \tag{2}$$

2) Let V be a finite set and let $\{\overline{\mathcal{D}}_v\}_{v \in V}$ be a set of distributions indexed by V, with each distribution over $\{0,1\}^{n \times t}$.

Let $\mathfrak{D}, \mathfrak{D}'$ be two distributions over $V \times [t]$, with the following properties:

(a) *If $(\mathbf{v}, \mathbf{j}) \sim \mathfrak{D}$, then $\mathbf{v}, \mathbf{j}$ are independent and $\mathbf{j}$ is uniform over $[t]$;*
(b) *$||\mathfrak{D} - \mathfrak{D}'|| \leq \ \gamma$, for some $\gamma \in [0,1)$.*

Consider the following experiment $\mathbf{Expt}(\mathfrak{D}')$:

(i) *Sample* $(\mathbf{v}', \mathbf{j}') \sim \mathfrak{D}'$*;*
(ii) *Sample* $\overline{\mathbf{x}} = (\mathbf{x}^1, \ldots, \mathbf{x}^t) \sim \overline{\mathcal{D}}_{\mathbf{v}'}$*;*
(iii) *Let the sequence* $\alpha_0, \ldots, \alpha_t, \beta_0, \ldots, \beta_{t-1}$ *as in Definition II.2 be defined with respect to* $\overline{\mathcal{D}}_{\mathbf{v}'}$ *and* $\overline{\mathbf{x}}$*.*

Then with probability at least $1 - \frac{2^{16}\ln(1/q)}{t} - \gamma$ *over* $\mathbf{Expt}(\mathfrak{D}')$*, we have*

$$\frac{\beta_{\mathbf{j}-1}}{\alpha_{\mathbf{j}-1}} \in [.99, 1.01] \quad \textit{and} \quad \frac{\alpha_{\mathbf{j}}}{\beta_{\mathbf{j}-1}} \in [.99, 1] \ . \tag{3}$$

Proof: **(1)** Let $T := 2t$, and define random variables $Y_1, \ldots, Y_T$ as follows: for each $\ell \in [T]$, if $\ell = 2k+1$ then let

$$Y_\ell := \beta_k/\alpha_k \ ,$$

and if $\ell = 2k$, let

$$Y_\ell := \alpha_k/\beta_{k-1} \ .$$

By our worst-case guarantee on C, the random variables $\alpha_0, \ldots, \alpha_{t-1}, \beta_0, \ldots, \beta_{t-1}$ are all positive, so the Y_ℓ are well-defined. To illustrate the pattern, we have

$$Y_1 = \left(\frac{\beta_0}{\alpha_0}\right) = 1 \ , \ Y_2 = \left(\frac{\alpha_1}{\beta_0}\right) \ , \ \ldots \ , \ Y_T = \left(\frac{\alpha_t}{\beta_{t-1}}\right) \ .$$

Now the product $Y_{\text{prod}} := \prod_{\ell \in [T]} Y_\ell$ equals $\alpha_t/\alpha_0 = \alpha_t$. By the definition of α_t and the guarantee on C, it follows that $Y_{\text{prod}} \geq q$.

For the even indices, we have $Y_{2k} \leq 1$ always. Also, we claim that $\mathbb{E}[Y_{2k+1}] = 1$; to see this, just observe that for $k \in [t(n)]$ we have the identity $\mathbb{E}[\beta_k | \alpha_k] = \alpha_k$, so that Y_{2k+1} has expected value 1 conditioned on any possible value of α_k.

We have verified that the assumptions of Lemma II.4 are satisfied by $(Y_1, \ldots, Y_T)$; we infer that if $\mathbf{i} \in_r [T]$,

$$\Pr\left[Y_{\mathbf{i}} \notin [.99, 1.01]\right] \leq 2^{16}\ln(1/q)/T = 2^{15}\ln(1/q)/t \ .$$

Recall that $T = 2t$ is even. If we instead choose $\hat{\mathbf{i}} \in_r \{1, 3, 5, \ldots, T-1\}$, and then select $\hat{\mathbf{i}}' \in_r \{\hat{\mathbf{i}}, \hat{\mathbf{i}}+1\}$, then $\hat{\mathbf{i}}'$ is uniform over $[T]$ and we get the same bound for $\Pr\left[Y_{\hat{\mathbf{i}}'} \notin [.99, 1.01]\right]$. It follows that

$$\Pr\left[Y_{\hat{\mathbf{i}}} \notin [.99, 1.01] \ \vee \ Y_{\hat{\mathbf{i}}+1} \notin [.99, 1.01]\right] \leq 2^{16}\ln(1/q)/t \ .$$

Now, note that $\mathbf{j} := \hat{\mathbf{i}}/2$ is distributed as a uniform element in $[t]$, and we have the relations

$$Y_{\hat{\mathbf{i}}} = \frac{\beta_{\mathbf{j}-1}}{\alpha_{\mathbf{j}-1}} \ , \quad Y_{\hat{\mathbf{i}}+1} = \frac{\alpha_{\mathbf{j}}}{\beta_{\mathbf{j}-1}} \leq 1 \ .$$

So Eq. (2) holds with probability at least $1 - \frac{2^{16}\ln(1/q)}{t}$.

(2) First, suppose we run the alternative experiment $\mathbf{Expt}(\mathfrak{D})$, which samples $(\mathbf{v}', \mathbf{j}')$ according to $\mathfrak{D}$ rather than $\mathfrak{D}'$. Now, after conditioning upon any outcome $[\mathbf{v}' = v]$ of the first component, the index $\mathbf{j}$ remains uniform over $[t]$ (by property (a) of $\mathfrak{D}$). Thus we may set $\overline{\mathcal{D}} := \overline{\mathcal{D}}_v$ and apply Lemma II.5, part 1 to find that the probability that Eq. (3) holds is at least $1 - \frac{2^{16}\ln(1/q)}{t}$. Thus in $\mathbf{Expt}(\mathfrak{D})$, Eq. (3) holds with probability at least $1 - \frac{2^{16}\ln(1/q)}{t}$.

Now let $\mathbf{I}, \mathbf{I}'$ be the indicator variables for the events that Eq. (3) holds in $\mathbf{Expt}(\mathfrak{D}), \mathbf{Expt}(\mathfrak{D}')$ respectively. Note that the two experiments are identically defined, except that the first draws a single sample from $\mathfrak{D}$ while the second draws a sample from $\mathfrak{D}'$. Thus, $||\mathbf{I} - \mathbf{I}'||_{\text{stat}} \leq ||\mathfrak{D} - \mathfrak{D}'||_{\text{stat}} \leq \gamma$, using property (b). This proves part 2 of the Lemma. ■

Lemma II.6. *Fix* $M \in \mathbb{N}^+$ *with* $M \geq 2$*, and* $q \in (0, 1]$*. Suppose the probabilistic circuit* C *is a worst-case* q*-direct-product solver for* $f^{\otimes t}$*. Let* $\mathcal{D}$ *be a distribution over* $\{0,1\}^n$*, and consider the following experiment* $\mathbf{Expt}^*(\mathcal{D})$*:*

1) *Let* $\mathbf{u} \in \{0,1\}^n$ *be sampled according to* $\mathcal{D}$*, and let* $\mathbf{j} \in_r [t]$*;*
2) *For each* $j \in [t]$*:*
   (i) *let* $s_j \in_r \{M, M+1, M+2, \ldots, 2M-1\}$*;*
   (ii) *Define a multiset* S_j *over* $\{0,1\}^n$*, obtained by drawing* s_j *independent samples from* $\mathcal{D}$*;*
   (iii) *Let* $\hat{S}_j := S_j \cup \{\mathbf{u}\}$ *if* $j = \mathbf{j}$*; otherwise let* $\hat{S}_j := S_j$*;*
   (iv) *Let* $\mathbf{y}^j \in_r S_j$*;*
   (v) *Let* $\hat{\mathbf{y}}^j := \mathbf{u}$ *if* $j = \mathbf{j}$*; otherwise let* $\hat{\mathbf{y}}^j := \mathbf{y}^j$*.*
3) *Define the random variables* $\alpha_0, \alpha_1, \ldots, \alpha_t$*,* $\beta_0, \beta_1, \ldots, \beta_{t-1}$*,* $\hat{\alpha}_0, \hat{\alpha}_1, \ldots, \hat{\alpha}_t$*,* $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_{t-1}$*, by*

$$\alpha_j := \Pr\left[C(\mathbf{y}^1, \ldots, \mathbf{y}^t) \textit{ is } j\textit{-valid} \mid S_1, S_2, \ldots, S_t, \mathbf{y}^1, \ldots, \mathbf{y}^j\right],$$

$$\hat{\alpha}_j := \Pr\left[C(\hat{\mathbf{y}}^1, \ldots, \hat{\mathbf{y}}^t) \textit{ is } j\textit{-valid} \mid \hat{S}_1, \hat{S}_2, \ldots, \hat{S}_t, \hat{\mathbf{y}}^1, \ldots, \hat{\mathbf{y}}^j\right],$$

$$\beta_j := \Pr\left[C(\mathbf{y}^1, \ldots, \mathbf{y}^t) \textit{ is } j\textit{-valid} \mid S_1, S_2, \ldots, S_t, \mathbf{y}^1, \ldots, \mathbf{y}^{j+1}\right],$$

$$\hat{\beta}_j := \Pr\left[C(\hat{\mathbf{y}}^1, \ldots, \hat{\mathbf{y}}^t) \textit{ is } j\textit{-valid} \mid \hat{S}_1, \hat{S}_2, \ldots, \hat{S}_t, \hat{\mathbf{y}}^1, \ldots, \hat{\mathbf{y}}^{j+1}\right].$$

Then with probability at least $1 - \frac{2^{16}\ln(1/q)}{t} - \frac{1}{M}$*, we have*

$$\frac{\hat{\beta}_{\mathbf{j}-1}}{\alpha_{\mathbf{j}-1}} \in [.98, 1.02] \quad \textit{and} \quad \frac{\hat{\alpha}_{\mathbf{j}}}{\hat{\beta}_{\mathbf{j}-1}} \in [.99, 1] \ . \tag{4}$$

Proof: We aim to apply part 2 of Lemma II.5 to the sequences $\hat{\alpha}_0, \ldots, \hat{\alpha}_t, \hat{\beta}_0, \ldots, \hat{\beta}_t$. Let $\mathbb{S}, \hat{\mathbb{S}}$ denote the random tuples $(S^1, \ldots, S^t)$ and $(\hat{S}^1, \ldots, \hat{S}^t)$ respectively. Let $\mathfrak{D}$ denote the distribution governing the tuple $(\mathbb{S}, \mathbf{j})$, and let $\mathfrak{D}'$ denote the distribution governing $(\hat{\mathbb{S}}, \mathbf{j})$. Each t-tuple $\overline{B} = (B_1, \ldots, B_t)$ of multisets over $\{0,1\}^n$ naturally defines a distribution $\overline{\mathcal{D}}_{\overline{B}}$ over elements $(\mathbf{z}^1, \ldots, \mathbf{z}^t) \in \{0,1\}^{n \times t}$, namely, the product distribution that independently chooses $\mathbf{z}^j \in_r B_j$ for each j.

The random variable $\mathbf{j}$ is uniform over $[t]$ and independent of $\mathbb{S}$, so condition (a) of Lemma II.5, part 2 is satisfied by $\mathfrak{D}$. For condition (b), note that $\hat{\mathbb{S}}$ is not fully independent of $\mathbf{j}$. However, we will show that $(\hat{\mathbb{S}}, \mathbf{j})$ is quite close in distribution to $(\mathbb{S}, \mathbf{j})$:

Claim II.7. $\left|\left|(\hat{S}_1, \ldots, \hat{S}_t, \mathbf{j}) - (S_1, \ldots, S_t, \mathbf{j})\right|\right|_{\text{stat}} \leq \frac{1}{M}$.

Proof: We use a coupling argument. First, we generate a random multiset S_0 consisting of M independent samples from $\mathcal{D}$. Now define a random multiset $\tilde{S}$ by letting

$$\tilde{S} \;:=\; \begin{cases} \hat{S}_{\mathbf{j}} & \text{if } s_{\mathbf{j}} \;<\; 2M-1 \;, \\ S_0 & \text{if } s_{\mathbf{j}} \;=\; 2M-1 \;. \end{cases}$$

Note that $|\tilde{S}|$ is uniform over $\{M, M+1, \ldots, 2M-1\}$, and its elements are distributed as independent samples from $\mathcal{D}$. Thus, if we form the random tuple

$$(S_1, \ldots, S_{\mathbf{j}-1}, \tilde{S}, S_{\mathbf{j}+1}, \ldots, S_t, \mathbf{j}) \;,$$

we see that it is identically distributed to $(S_1, \ldots, S_{\mathbf{j}}, \ldots, S_t, \mathbf{j})$. Also, we have $\tilde{S} = \hat{S}_{\mathbf{j}}$ unless $s_{\mathbf{j}} = 2M-1$, which happens only with probability $\frac{1}{M}$; and we always have $\hat{S}_j = S_j$ for all $j \neq \mathbf{j}$. This proves our Claim. ∎

Thus condition (b) is satisfied by $\mathfrak{D}, \mathfrak{D}'$ with $\gamma := \frac{1}{M}$.

Observe that, after conditioning on $\mathbb{S}$, the sequence $(\mathbf{y}^1, \ldots, \mathbf{y}^t)$ sampled in $\mathbf{Expt}^*(\mathcal{D})$ is distributed precisely according to $\overline{\mathcal{D}}_{\mathbb{S}}$. Similarly, after conditioning on $\hat{\mathbb{S}}$, the sequence $(\hat{\mathbf{y}}^1, \ldots, \hat{\mathbf{y}}^t)$ is distributed according to $\overline{\mathcal{D}}_{\hat{\mathbb{S}}}$. We can therefore combine part 2 of Lemma II.5 with Claim II.7 to find that, with probability at least $1 - \frac{2^{16}\ln(1/q)}{t} - \frac{1}{M}$ over $\mathbf{Expt}^*(\mathcal{D})$, we have

$$\frac{\hat{\beta}_{\mathbf{j}-1}}{\hat{\alpha}_{\mathbf{j}-1}} \;\in\; [.99, 1.01] \quad \text{and} \quad \frac{\hat{\alpha}_{\mathbf{j}}}{\hat{\beta}_{\mathbf{j}-1}} \;\in\; [.99, 1] \;.$$

Next we will need the following simple but important claim:

Claim II.8. *With probability 1 we have*

$$\hat{\alpha}_{\mathbf{j}-1} \;=\; \frac{1}{s_{\mathbf{j}}+1} \cdot \hat{\beta}_{\mathbf{j}-1} + \left(\frac{s_{\mathbf{j}}}{s_{\mathbf{j}}+1}\right) \cdot \alpha_{\mathbf{j}-1} \;.$$

Proof: Consider any outcome of $\mathbf{Expt}^*(\mathcal{D})$, which is fully determined by the values of the random variables

$$(S_1, \ldots, S_t, \mathbf{y}^1, \ldots, \mathbf{y}^t, \mathbf{j}, \mathbf{u}) \;=\; (S_1^*, \ldots, S_t^*, y^1, \ldots, y^t, j, u) \;.$$

Under these conditionings, observe that $\hat{\alpha}_{\mathbf{j}-1}$ equals the probability that the output of the computation $C(y^1, \ldots, y^{j-1}, \mathbf{z}^j, \mathbf{z}^{j+1}, \ldots, \mathbf{z}^t)$ is $(j-1)$-valid, where

$$(\mathbf{z}^j, \mathbf{z}^{j+1}, \ldots, \mathbf{z}^t) \;\in_r\; ((S_j^* \cup \{u\}) \times S_{j+1}^* \times \ldots \times S_t^*) \;.$$

This distribution on $(\mathbf{z}^j, \mathbf{z}^{j+1}, \ldots, \mathbf{z}^t)$ can be equivalently realized as follows:

1) First, let $\mathbf{a} \in_r [s_j + 1]$ (noting here that $s_j = |S_j^*|$);
2) If $\mathbf{a} = s_j + 1$, let $\mathbf{z}^j := u$ and sample $(\mathbf{z}^{j+1}, \ldots, \mathbf{z}^t) \in_r (S_{j+1}^* \times \ldots \times S_t^*)$; otherwise choose $(\mathbf{z}^j, \mathbf{z}^{j+1}, \ldots, \mathbf{z}^t) \in_r (S_j^* \times S_{j+1}^* \times \ldots \times S_t^*)$.

On the other hand, $\hat{\beta}_{\mathbf{j}-1}$ equals the probability that $C(y^1, \ldots, y^{j-1},\ u, \mathbf{z}^{j+1}, \mathbf{z}^{j+2}, \ldots, \mathbf{z}^t)$ is $(j-1)$-valid, where $(\mathbf{z}^{j+1}, \ldots, \mathbf{z}^t) \in_r (S_{j+1}^* \times \ldots \times S_t^*)$; and $\alpha_{\mathbf{j}-1}$ equals the probability that $C(y^1, \ldots, y^{j-1}, \mathbf{z}^j, \mathbf{z}^{j+1}, \ldots, \mathbf{z}^t)$ is $(j-1)$-valid, where $(\mathbf{z}^j, \ldots, \mathbf{z}^t) \in_r (S_j^* \times \ldots \times S_t^*)$. Combining these facts with our observations about $\hat{\alpha}_{\mathbf{j}-1}$ yields the Claim. ∎

Now, suppose that $\hat{\beta}_{\mathbf{j}-1} \;\geq\; .99 \cdot \hat{\alpha}_{\mathbf{j}-1}$, which, as we have seen, occurs with high probability. Using Claim II.8, this implies that

$$\frac{\hat{\beta}_{\mathbf{j}-1}}{.99} \;\geq\; \frac{1}{s_{\mathbf{j}}+1} \cdot \hat{\beta}_{\mathbf{j}-1} + \left(\frac{s_{\mathbf{j}}}{s_{\mathbf{j}}+1}\right) \cdot \alpha_{\mathbf{j}-1} \;,$$

which simplifies to

$$\frac{\hat{\beta}_{\mathbf{j}-1}}{\alpha_{\mathbf{j}-1}} \;\geq\; \frac{99 s_{\mathbf{j}}}{100 s_{\mathbf{j}} + 1} \;.$$

This is greater than .98. By a similar calculation, if $\hat{\beta}_{\mathbf{j}-1} \;\leq\; 1.01 \cdot \hat{\alpha}_{\mathbf{j}-1}$ then

$$\frac{\hat{\beta}_{\mathbf{j}-1}}{\alpha_{\mathbf{j}-1}} \;\leq\; \left(\frac{101 \cdot s_{\mathbf{j}}}{100 s_{\mathbf{j}} - 1}\right) \;,$$

which is less than 1.02 since $s_{\mathbf{j}} \geq 2$. Combining our work, we conclude that with probability at least $1 - \frac{2^{16}\ln(1/q)}{t} - \frac{1}{M}$, Eq. (4) is satisfied. This completes the proof of Lemma II.6. ∎

Now let C be a circuit of size s that is a worst-case q-direct product solver for $f^{\otimes t}$, for $q > \exp(-t/10^7)$. Fix any distribution $\mathcal{D}$ over $\{0,1\}^n$. Below, we will describe part of the construction (using C) of a polynomially larger non-deterministic circuit $C' = C'_{\mathcal{D}}$, computing f[6] with success probability $> .6$ on inputs from $\mathcal{D}$. This step forms the bulk of the proof of (the contrapositive form of) Theorem I.2. First, refer to the experiment $\mathbf{Expt}^*(\mathcal{D})$ of Lemma II.6, defined with respect to $\mathcal{D}, C$, and with $M := \lceil t/\ln(1/q) \rceil$. With the random variables $\alpha_j, \hat{\alpha}_j, \beta_j, \hat{\beta}_j$ as defined in that experiment, let us fix outcomes to the random variables

$$s_1, \ldots, s_t,\ S_1, \ldots, S_t,\ \mathbf{j},\ \mathbf{y}^1, \ldots, \mathbf{y}^t$$

that maximize the probability that Eq. (4) holds, where the probability is now taken over $\mathbf{u} \sim \mathcal{D}$. Let Λ denote the collection of random variables whose values we are fixing, and let $[\Lambda = \lambda]$ denote the particular setting we are making. Let $j^* \in [t]$ denote the fixed outcome to $\mathbf{j}$. (In our circuit construction, we will ignore the values $\mathbf{y}^j$ for $j > j^*$.)

When in $\mathbf{Expt}^*(\mathcal{D})$ we condition on $[\Lambda = \lambda]$, Eq. (4) holds with probability at least $1 - \frac{2^{16}\ln(1/q)}{t} - \frac{1}{M} \geq \frac{(2^{16}+1)\ln(1/q)}{t}$, which is $> .75$ by our guarantee on C under the assumptions of Theorem I.2. Also, under our conditioning, $\mathbf{u}$ remains undetermined and is distributed according to $\mathcal{D}$. Our input to C' will play the role of $\mathbf{u}$ in our construction.

[6] A nondeterministic C' *computes* f on input x if $C'(x)$ has some branch outputting a correct guess for $f(x)$, and $C'(x)$ never outputs an incorrect guess—although C' may output "fail" on any number of branches for input x, and may output multiple distinct values on other "bad" inputs $x' \neq x$. Note that this definition makes sense for $d > 1$.

Note that our settings determine outcomes to $\alpha_0, \ldots, \alpha_t, \beta_0, \ldots, \beta_{t-1}$. The value $\alpha_{j^*-1} > 0$ in particular will be useful to us in defining our circuit. Abusing notation somewhat, we now let $s_1, \ldots, s_t, S_1, \ldots, S_t, \alpha_0, \ldots, \alpha_t, \beta_0, \ldots, \beta_{t-1}$ denote the fixed outcomes to these variables. For each $j \in [t]$, let

$$S_j \;=\; \{\; y^{j,\ell} \;\}_{\ell \in [s_j]}$$

be an indexing of S_j (with some elements possibly appearing multiple times, according to their multiplicity in S_j). We define a mapping $\ell^* : [t] \to \mathbb{N}^+$ by the relation that, for our outcomes to $\mathbf{y}^1, \ldots, \mathbf{y}^t$, we have

$$\mathbf{y}^j \;=\; y^{j,\ell^*(j)} \;.$$

As another important piece of non-uniform data in our construction, we will need to know the values taken by f on $y^{1,\ell^*(1)}, \ldots, y^{j^*-1,\ell^*(j^*-1)}$. For $j \in [j^*-1]$, we let

$$\hat{z}^j \;:=\; f(y^{j,\ell^*(j)}) \;.$$

Suppose C uses $R > 0$ bits of randomness. Let $C^{\det}(x^1, \ldots, x^t; r) : \{0,1\}^{n \times t + R} \to \{0,1\}$ denote C considered as a deterministic circuit with random string r as part of its input. For any string $u \in \{0,1\}^n$, we define a *viable certificate for* u as a tuple

$$w = (m_{j^*+1}, m_{j^*+2}, \ldots, m_t, r) \in [s_{j^*+1}] \times \ldots \times [s_t] \times \{0,1\}^R$$

for which the first (j^*-1) length-d output blocks of the computation

$$C^{\det}\left(y^{1,\ell^*(1)}, \ldots, y^{j^*-1,\ell^*(j^*-1)}, u, y^{j^*+1,m_{j^*+1}}, \ldots, y^{t,m_t};\; r\right) \tag{5}$$

equal $(\hat{z}^1, \ldots, \hat{z}^{j^*-1})$. For such a w and $z \in \{0,1\}^d$, we say that w is a *viable z-certificate for* u if the $(j^*)^{th}$ output block of the computation in Eq. (5) equals z, i.e., if in this computation C makes the "guess" that $f(u)$ equals z.

We fix some natural encoding of $[s_{j^*+1}] \times \ldots \times [s_t] \times \{0,1\}^R$ in which each element w has a unique representation as a binary string in $\{0,1\}^N$; here, we may take $N \leq R + O(t \log_2 M) \leq \mathrm{poly}(\mathrm{size}(C))$. Let $V_u \subseteq \{0,1\}^N$ denote the set of viable certificates for u, and for $z \in \{0,1\}^d$, let $V_u^z \subseteq V_u$ denote the viable z-certificates for u. The sets V_u^z form a partition of V_u.

Claim II.9. *Let us condition on $[\Lambda = \lambda]$ as above in $\mathbf{Expt}^*(\mathcal{D})$. Then,*

1) *For the random variable $\mathbf{u}$ over $\{0,1\}^n$, the equality*

$$|V_{\mathbf{u}}| = \hat{\beta}_{j^*-1} \cdot 2^R \prod_{j=j^*+1}^{t} s_j \tag{6}$$

holds with probability 1.

2) *Also, we have the equality*

$$\left|V_{\mathbf{u}}^{f(\mathbf{u})}\right| = \hat{\alpha}_{j^*} \cdot 2^R \prod_{j=j^*+1}^{t} s_j, \tag{7}$$

and therefore

$$\frac{\left|V_{\mathbf{u}}^{f(\mathbf{u})}\right|}{|V_{\mathbf{u}}|} \;=\; \frac{\hat{\alpha}_{j^*}}{\hat{\beta}_{j^*-1}} \;. \tag{8}$$

Proof: **(1)** Condition further on any possible outcome $[\mathbf{u} = u]$ in $\mathbf{Expt}^*(\mathcal{D})$. Together with our prior conditioning $[\Lambda = \lambda]$, this determines the values of $\hat{S}_1, \ldots, \hat{S}_t, \hat{\alpha}_1, \ldots, \hat{\alpha}_t, \hat{\beta}_1, \ldots, \hat{\beta}_t$. Under this conditioning, we see from the definition that

$$\hat{\beta}_{j^*-1} = \Pr\left[C(\hat{\mathbf{y}}^1, \ldots, \hat{\mathbf{y}}^t) \text{ is } (j^*-1)\text{-valid} \middle| \hat{S}_1, \hat{S}_2, \ldots, \hat{S}_t, \hat{\mathbf{y}}^1, \ldots, \hat{\mathbf{y}}^{j^*}\right] \tag{9}$$

$$= \Pr\left[C(y^{1,\ell^*(1)}, \ldots, y^{j^*-1,\ell^*(j^*-1)}, u, \mathbf{v}^{j^*+1}, \ldots, \mathbf{v}^t) \text{ is } (j^*-1)\text{-valid }\right] \tag{10}$$

where we sample $(\mathbf{v}^{j^*+1}, \ldots, \mathbf{v}^t) \in_r S_{j^*+1} \times \ldots \times S_t$ (and where validity is with respect to f). Here, $S_{j^*+1}, \ldots, S_t$ are our fixed values under $[\Lambda = \lambda]$, and the probability in Eq. (10) is taken over $\mathbf{v}^{j^*+1}, \ldots, \mathbf{v}^t$ and over the random bits r used by C.

Let us calculate the probability in Eq. (10). The selection of $(\mathbf{v}^{j^*+1}, \ldots, \mathbf{v}^t)$ may be equivalently performed by choosing $(m_{j^*+1}, \ldots, m_t) \in_r [s_{j^*+1}] \times \ldots \times [s_t]$, and setting $\mathbf{v}^j := y^{j,m_j}$ for $j \in \{j^*+1, \ldots, t\}$. There are $\left(\prod_{j=j^*+1}^{t} s_j\right) \cdot 2^R$ possible outcomes to $(m_{j^*+1}, \ldots, m_t, r)$, each one equally likely. The outcomes that cause the computation indicated in Eq. (10) to be (j^*-1)-valid are, under our definition, precisely those for which

$$(m_{j^*+1}, \ldots, m_t, r) \;\in\; V_u \;.$$

Thus, under our conditioning $[\mathbf{u} = u]$ we have

$$|V_u| \;=\; \hat{\beta}_{j^*-1} \cdot 2^R \prod_{j=j^*+1}^{t} s_j \;.$$

As u was an arbitrary outcome to $\mathbf{u}$, we have proved part 1 of the Claim.

(2) Condition again on any possible outcome $[\mathbf{u} = u]$ in $\mathbf{Expt}^*(\mathcal{D})$. Then we have

$$\hat{\alpha}_{j^*} = \Pr\left[C(\hat{\mathbf{y}}^1, \ldots, \hat{\mathbf{y}}^t) \text{ is } j^*\text{-valid} \;\middle|\; \hat{S}_1, \hat{S}_2, \ldots, \hat{S}_t, \hat{\mathbf{y}}^1, \ldots, \hat{\mathbf{y}}^{j^*}\right] \tag{11}$$

$$= \Pr\left[C(y^{1,\ell^*(1)}, \ldots, y^{j^*-1,\ell^*(j^*-1)}, u, \mathbf{v}^{j^*+1}, \ldots, \mathbf{v}^t) \text{ is } (j^*-1)\text{-valid}\right], \tag{12}$$

where again $(\mathbf{v}^{j^*+1}, \ldots, \mathbf{v}^t) \in_r S_{j^*+1} \times \ldots \times S_t$. Let these $\mathbf{v}^j$ be generated by $(m_{j^*+1}, \ldots, m_t)$ just as in part 1. The outcomes that cause the computation in Eq. (12) to be j^*-valid are exactly those for which the following two conditions hold:

1) $(m_{j^*+1}, \ldots, m_t, r) \in V_u$;
2) The $(j^*)^{th}$ output block of $C^{\det}(y^{1,\ell^*(1)}, \ldots, y^{j^*-1,\ell^*(j^*-1)},\; u,\; \mathbf{v}^{j^*+1}, \ldots, \mathbf{v}^t; r)$ equals $f(u)$.

These outcomes are exactly those for which $(m_{j^*+1}, \ldots, m_t, r) \in V_u^{f(u)}$. Then by a calculation following that in part 1, we can verify that Eq. (7) holds under $[\mathbf{u} = u]$. As u was arbitrary, Eq. (7) holds identically. Combining this with part 1 gives Eq. (8). ∎

We have chosen the settings $[\Lambda = \lambda]$ so that Eq. (4) holds with high probability over $\mathbf{u} \sim \mathcal{D}$. Using Claim II.9, this implies that $|V_{\mathbf{u}}^{f(u)}| \approx |V_{\mathbf{u}}|$ with high probability, i.e., almost all viable certificates for u correctly guess $f(u)$. Furthermore, by both parts of Claim II.9 and the first condition of Eq. (4), both of $|V_{\mathbf{u}}^{f(\mathbf{u})}|, |V_{\mathbf{u}}|$ are (with high probability) approximately equal to $\rho := 2^R \left(\prod_{j=j^*+1}^{t} s_j\right) \alpha_{j^*-1}$; this latter quantity is determined by the setting $[\Lambda = \lambda]$, and does not depend on $\mathbf{u}$.

This motivates our strategy for building a nondeterministic mapping circuit to compute f on an input u sampled from $\mathcal{D}$. First, we choose a random hash function h from a strongly universal hash family with domain $U := \{0,1\}^N$, and with a range space of size determined by ρ. We consider an element of U "dead" unless it maps to the all-0 vector under h; our aim is to "kill off" all of the viable certificates making incorrect guesses for $f(u)$, while leaving alive some viable certificate that makes a correct guess. Our control on the sizes of $|V_{\mathbf{u}}|, |V_{\mathbf{u}}^{f(\mathbf{u})}|$ makes it possible to ensure this outcome with good success probability. Nondeterminism will then allow us to guess and verify a live, viable certificate.

To make this strategy succeed with the required probability, we perform $\mathrm{poly}(n)$ repeated trials, selecting multiple hash functions and taking a majority vote over the trials. In the end we fix the randomness in these trials to obtain our final nondeterministic circuit $C'_{\mathcal{D}}$. As mentioned earlier, from the fact that we can do this for *any* input distribution $\mathcal{D}$, we are able to use a standard boosting/majority vote technique to obtain a single (polynomially larger) nondeterministic circuit computing f on all length-n inputs.

ACKNOWLEDGMENT

I thank Avi Wigderson for helpful comments, and Robin Moser for a discussion of related questions.

This material is based upon work supported by the National Science Foundation under agreements Princeton University Prime Award No. CCF-0832797 and Sub-contract No. 00001583.

REFERENCES

[1] O. Goldreich, *Foundations of Cryptography: Volume 1, Basic Tools*, 1st ed. Cambridge University Press, 2007.

[2] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby, "A pseudorandom generator from any one-way function," *SIAM J. Comput.*, vol. 28, no. 4, pp. 1364–1396, 1999.

[3] R. Impagliazzo and A. Wigderson, "*P = BPP* if *E* requires exponential circuits: Derandomizing the XOR lemma," in *29th ACM STOC*, 1997, pp. 220–229.

[4] L. A. Levin, "One-way functions and pseudorandom generators," *Combinatorica*, vol. 7, no. 4, pp. 357–363, 1987.

[5] O. Goldreich, N. Nisan, and A. Wigderson, "On Yao's XOR-lemma," in *Studies in Complexity and Cryptography*, ser. Lecture Notes in Computer Science. Springer, 2011, vol. 6650, pp. 273–301, earlier version on ECCC (TR95-050, 1995).

[6] O. Goldreich and L. A. Levin, "A hard-core predicate for all one-way functions," in *21st ACM STOC*, 1989, pp. 25–32.

[7] N. Nisan, S. Rudich, and M. E. Saks, "Products and help bits in decision trees," *SIAM J. Comput.*, vol. 28, no. 3, pp. 1035–1050, 1999, earlier version in FOCS '94.

[8] E. Viola and A. Wigderson, "Norms, XOR lemmas, and lower bounds for polynomials and protocols," *Theory of Computing*, vol. 4, no. 1, pp. 137–168, 2008, earlier version in CCC '07.

[9] R. Impagliazzo, R. Jaiswal, V. Kabanets, and A. Wigderson, "Uniform direct product theorems: Simplified, optimized, and derandomized," *SIAM J. Comput.*, vol. 39, no. 4, pp. 1637–1665, 2010, earlier version in STOC '08.

[10] R. Shaltiel, "Towards proving strong direct product theorems," *Computational Complexity*, vol. 12, no. 1-2, pp. 1–22, 2003, earlier version in CCC '01.

[11] A. Wigderson, "Derandomizing BPP," 1997, lecture notes prepared by Ronen Shaltiel. http://www.math.ias.edu/∼avi/BOOKS/rand.pdf.

[12] R. Shaltiel and E. Viola, "Hardness amplification proofs require majority," *SIAM J. Comput.*, vol. 39, no. 7, pp. 3122–3154, 2010.

[13] U. Feige and C. Lund, "On the hardness of computing the permanent of random matrices," *Computational Complexity*, vol. 6, no. 2, pp. 101–132, 1997.

[14] A. Amir, R. Beigel, and W. I. Gasarch, "Some connections between bounded query classes and non-uniform complexity," *Inf. Comput.*, vol. 186, no. 1, pp. 104–139, 2003, orig. 1990.

[15] S. Toda, "PP is as hard as the polynomial-time hierarchy," *SIAM J. Comput.*, vol. 20, no. 5, pp. 865–877, 1991.

[16] L. Trevisan and S. P. Vadhan, "Extracting randomness from samplable distributions," in *41st IEEE FOCS*, 2000, pp. 32–42.

[17] R. Paturi and P. Pudlák, "On the complexity of circuit satisfiability," in *STOC*, L. J. Schulman, Ed. ACM, 2010, pp. 241–250.

[18] S. R. Buss and L. Hay, "On truth-table reducibility to SAT and the difference hierarchy over NP," in *3rd Structure in Complexity Theory Conference*, 1988, pp. 224–233.

[19] C. H. Papadimitriou, *Computational Complexity*. Addison-Wesley, 1994.

[20] A. C.-C. Yao, "Theory and applications of trapdoor functions (extended abstract)," in *FOCS*. IEEE Computer Society, 1982, pp. 80–91.

[21] N. Nisan and A. Wigderson, "Hardness vs randomness," *J. Comput. Syst. Sci.*, vol. 49, no. 2, pp. 149–167, 1994, earlier version in FOCS '88.

[22] A. Drucker, "A PCP characterization of AM," in *ICALP*, 2011, pp. 581–592, full version at http://eccc.hpi-web.de/report/2010/019/.

[23] R. O'Donnell, "Hardness amplification within NP," in *34th ACM STOC*, 2002, pp. 751–760.

[24] A. Bogdanov and L. Trevisan, "Average-case complexity," *Foundations and Trends in Theoretical Computer Science*, vol. 2, no. 1, 2006.

[25] S. Goldwasser and M. Sipser, "Private coins versus public coins in interactive proof systems," in *18th ACM STOC*, 1986, pp. 59–68.

[26] L. G. Valiant and V. V. Vazirani, "NP is as easy as detecting unique solutions," *Theor. Comput. Sci.*, vol. 47, no. 3, pp. 85–93, 1986, earlier version in STOC '85.

[27] R. Raz, "A parallel repetition theorem," *SIAM J. Comput.*, vol. 27, no. 3, pp. 763–803, 1998, earlier version in STOC '95.

[28] A. Drucker, "New limits to classical and quantum instance compression," in *53rd IEEE FOCS*, 2012, pp. 609–618, full version at http://eccc.hpi-web.de/report/2012/112/.

[29] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[30] C.-K. Yap, "Some consequences of non-uniform conditions on uniform classes," *Theor. Comput. Sci.*, vol. 26, pp. 287–300, 1983.

[31] R. V. Book, T. J. Long, and A. L. Selman, "Quantitative relativizations of complexity classes," *SIAM J. Comput.*, vol. 13, no. 3, pp. 461–487, 1984.

[32] ——, "Qualitative relativizations of complexity classes," *J. Comput. Syst. Sci.*, vol. 30, no. 3, pp. 395–413, 1985.

[33] L. A. Hemaspaandra and M. Ogihara, *The Complexity Theory Companion*, ser. Texts in Theoretical Computer Science. Springer, 2002.