Kolmogorov Complexity and Toda's Theorem[1]

J. Ramachandran
Computer and Information Sciences
Ohio State University
Columbus, Ohio, 43210

[1] Please address correspondence to J. Ramachandran, BU CS Dept, 111 Cummington Street, Boston, MA 02215. ramachan@cs.bu.edu

Abstract

Toda's theorem states that the polynomial time hierarchy is contained in P(PP) An important building block in the proof is the lemma: NP is random reducible to $\oplus$P. We present a new proof of a corollary of this lemma, NP $\subseteq$ $\oplus$P/poly, using Kolmogorov complexity. The proof provides a new perspective for understanding Toda's Theorem and adds to the growing corpus of work relating Kolmogorov complexity to Structural complexity.

1 Introduction

Valiant defined #P as the class of functions computable by counting the number of accepting paths of polynomial time bounded nondeterministic machines. Papadimitriou and Zachos first introduced the counting class $\oplus$P derived from #P ([13]), defined as the class of languages L for which there exists a #P function f such that, for all strings x in Σ^*, x is in L if and only if $f(x)$ is odd. They showed $\oplus\mathrm{P}(\oplus\mathrm{P}) = \oplus\mathrm{P}$ and $\oplus\mathrm{P} \subseteq \mathrm{P}(\#\mathrm{P}[1])$. Since then, there has been considerable research on counting classes, the best known of which is Toda's seminal theorem proving that the polynomial time hierarchy (PH) is contained in BP.$\oplus$P (and therefore, in P(PP)).

Toda's proof depends critically on an efficient randomized procedure developed by Valiant and Vazirani [18] to show that NP random reduces to USAT (the set of boolean formulae with unique satisfying assignments). Toda uses this procedure to show:

Lemma 1.1 NP $\subseteq$ BP.$\oplus$P.

Karp and Lipton ([8]) defined *advice functions* as a notion of nonuniform complexity. A language A is in the class $\mathcal{C}/\mathcal{F}$ if, for some choice of B in the language class $\mathcal{C}$ and of function f from the function class $\mathcal{F}$, for all x in Σ^*, x is in A if and only if $\langle x, f(|x|)\rangle$ is in B. The *advice string* $f(|x|)$ depends only on the length of the input string x and is presented at no cost.

Toda and Ogiwara in [17] state the following as a corollary of Lemma 1.1:

Theorem 1.2 NP $\subseteq$ $\oplus$P/poly.

We prove Theorem 1.2 using Kolmogorov complexity. The Kolmogorov complexity of a string x is the length of the shortest program p that produces x from empty input, (denoted $C(x) = |p|$). Informally, a string is said to be Kolmogorov random or *incompressible* if $C(x) > |x| - 2\log|x|$. It is often the case that proofs using counting or probability arguments are simplified by substituting a Kolmogorov complexity argument, where we pick an incompressible string x and develop contradictions of the form $C(x) < |x| - 2\log|x|$.

In addition to providing a novel viewpoint for Toda's Theorem we believe the proof has intuitive appeal, giving a structural interpretation to Theorem 1.2, namely that the advice string is composed of two components, each of appropriate Kolmogorov complexity, where the first component encodes a family of hash functions, and the second encodes a choice of a single path in some nondeterministic computation tree. The proof shows yet another interesting application of Kolmogorov complexity in structural complexity theory.

2 Definitions and Notations

Recall the definitions of standard complexity classes such as P, NP, BPP, PH, #P, etc. as found in [3] and [4].

Definition 2.1 *A language L is in $\oplus$P if there exists #P function f such that for all $x \in \Sigma^*$ $[x \in L \iff f(x) \equiv 1 \bmod 2]$.*

Definition 2.2 *[8]*

1. *A set A belongs to $\mathcal{C}/\mathcal{F}$ if there exists a B in $\mathcal{C}$ and a function $f : \mathcal{N} \to \Sigma^*$ in $\mathcal{F}$, such that, for all $x \in \Sigma^*$, $x \in A$ if and only if $\langle x, f(|x|)\rangle \in B$.*
2. *A function f is in the function class* poly *if there exists a constant c such that, for all $n \in \mathcal{N}$, $|f(n)| \le n^c$ respectively.*

Also recall the definition of the BP operator:

Definition 2.3 *A language A is in the class BP.$\mathcal{C}$ if and only if there exists a $B \in \mathcal{C}$, a polynomial p, and an $\epsilon > 0$ such that*

$$\begin{aligned} x \in A &\implies \mathrm{Prob}_{w,|w|=p(|x|)}\,[\langle x, w\rangle \in B] \ge 1-\epsilon, \quad \text{and} \\ x \notin A &\implies \mathrm{Prob}_{w,|w|=p(|x|)}\,[\langle x, w\rangle \in B] \le \epsilon \end{aligned}$$

Recall also the definition of resource bounded Kolmogorov complexity. Let ϕ be a universal partial recursive function and let T_ϕ, a Turing machine that computes ϕ, be a universal transducer. Let x, y, and p be strings in Σ^*, $|x| = n$, and let $t : \mathcal{N} \to \mathcal{N}$ and $s : \mathcal{N} \to \mathcal{N}$ be a time and a space bound respectively. If $T_\phi(p, y) = x$ then $\phi(p, y) = x$, and $\langle p, y\rangle$ is a *description* of x. Let A be an oracle set, and let ϕ^A be a universal partial recursive function, under the assumption that the characteristic function of A is computable in constant time. Let T_ϕ^A, an oracle Turing machine that computes ϕ^A, be a universal oracle transducer. If $T_\phi^A(p, y) = x$ then $\phi^A(p, y) = x$, and $\langle p, y\rangle$ is a *description* of x, relative to A.

Definition 2.4 *[11] The resource bounded Kolmogorov complexity of x conditional to ϕ, y and A, denoted $C_\phi^{t,s}(x|A,y)$, is the length of the shortest program p such that $\phi^A(p,y)$ is x. The simulation of ϕ^A by T_ϕ^A runs in time $t(|x|)$ and space $s(|x|)$.*

Note that in the definition of $C_\phi^{t,s}(x|A,y)$ the information after the "|" is given for free. A may consist of several oracles joined together, and y may consist of several strings paired together. The Invariance Theorem in [11] states that

Theorem 2.5 *There exists a universal partial function ϕ_0 such that for all other partial functions ϕ, oracles A, and all $x, y \in \Sigma^*$, there exists a constant c dependent only on ϕ such that $C_{\phi_0}^{ct\log t, cs}(x|A,y) = C_\phi^{t,s}(x|A,y) + c$.*

The subscript ϕ may be dropped from $C_\phi^{t,s}(x|A,y)$ when the time, space, and length bounds are not sensitive to these changes. The parameters y, A, t or s can be dropped in the following circumstances: y is the empty string; A is the empty set; t is infinity; or s is infinity, respectively. For example $C_\phi^{t,s}(x)$ is the *unconditional resource bounded Kolmogorov complexity* of x, $C(x|y)$ is the Kolmogorov complexity of x conditional to y, and $C(x)$ is the unconditional Kolmogorov complexity of x.

Definition 2.6 $\mathrm{K}[l(n), t(n)] = \{x \in \Sigma^{=n} \mid C_\phi^{t(n),\infty}(x) \leq l(n)\}$.

The class $\mathrm{K}[l(n), t(n)]$ denotes the set of strings $x \in \Sigma^{=n}$ that can be generated by a fixed universal transducer running some $l(n)$ bit program in $t(n)$ time.

Definition 2.7 *[11] The self-delimiting code of a string x is the string $\overline{x}x$ where $\overline{x} = u10$, where u is obtained by doubling each bit of the binary representation of $|x|$.*

Self-delimiting codes allow the easy decoding of a sequence of concatenated strings, if each is replaced by its self-delimiting code.

3 Kolmogorov Complexity and Toda's Theorem

Toda's Theorem $\mathrm{PH} \subseteq \mathrm{P(PP)}$ demonstrates that the counting classes have a surprising amount of power. This seminal theorem has been proved in a number of ways (r.f. [10], [16], [14], [5]). An important lemma in the proof is the inclusion $\mathrm{NP} \subseteq \mathrm{BP.}\oplus\mathrm{P}$.

In this section, we provide a new viewpoint to the above proofs by proving a corollary to NP $\subseteq$ BP. $\oplus$P, using Kolmogorov complexity. We show that NP $\subseteq$ $\oplus$P/poly, a result first stated in by Toda and Ogiwara in [17].

We prove NP $\subseteq$ $\oplus$P/poly in three stages:

1. We construct a nondeterministic transducer M_U that takes as input a boolean formula F and a string τ and, assuming τ is sufficiently long and is Kolmogorov random, outputs on a fraction (bounded from below) of its paths a formula F' which is uniquely satisfiable if and only if F is satisfiable.

2. We construct a nondeterministic transducer M_{BP} that uses M_U and, on input F and τ and an integer t, under the assumption that τ is sufficiently long and is Kolmogorov random, outputs with high probability (dependent on t) a formula F'' that has an odd number of satisfying assignments if and only if F is satisfiable.

3. Finally we create a deterministic transducer $M_{\oplus\mathrm{P}}$ that takes as input F, τ, and τ', and assuming τ and τ' are sufficiently long and are Kolmogorov random, chooses a specific computation path of M_{BP} (the choice determined solely by τ') and outputs the formula computed on this path, which has an odd number of satisfying assignments if and only if F is satisfiable.

The following definitions and Lemma 3.1 are derived from [11]. Let $h : \Sigma^n \to \Sigma^m$ be a linear transformation given by a given $m \times n$ matrix $R = r_{ij}$. That is, for each $x \in \Sigma^n$ we have $R(x)$ is a string $y \in \Sigma^m$ where $y_i = (\sum_j r_{ij}.x_j) \bmod 2$. Let H be a collection of such functions. Let $B, C \subseteq \Sigma^n$ and $x \in \Sigma^n$. A mapping h *separates* x *within* B if for every $y \in B$, different from x, $h(y) \neq h(x)$. The mapping h *separates* C *within* B if it separates each $x \in C$ within B. The set of mappings H *separates* C *within* B if for each $x \in C$ some $h \in H$ separates x within B. We require a *coding lemma* from [11]:

Lemma 3.1 (Coding lemma) *Let $B \subseteq \Sigma^{=n}$, where $|B| = k$, and let $N \geq n$. Let $m = 1 + \lceil \log k \rceil$. There is a collection H of Nm linear transformations $\Sigma^n \to \Sigma^m$ such that H separates B within B.*

Proof. Fix a random string z of length Nnm^2 such that $C(z|B, P, m, n, N) \geq |z| - (N - 1)m$, where P is the program for describing z (independent of the actual z) in the following discussion. Cut z into Nm equal pieces. Use the nm bits from each piece to form an $n \times m$ matrix in the obvious way. Thus, we have constructed a set H of Nm Kolmogorov random hash functions. We claim H separates B within B.

Assume the contrary. That is, for some $x \in B$, no $h \in H$ separates x within B. Then, there exist $y_1, y_2, \ldots, y_{Nm} \in B$ such that $h_i(x) = h_i(y_i)$. Hence $h_i(x - y_i) = 0$ (here $(x - y_i)$ means mod 2 subtraction).

Since $x - y_i \neq 0$, the first column of h_i corresponding to a '1' in vector $x - y_i$ can be expressed by the rest of the columns using $x - y_i$. Now we can describe a fixed program P that uses the following data:

1. The index of x in B, using $\lceil \log k \rceil$ bits;
2. The indices of $y_1, y_2, \ldots, y_{Nm}$ in B, using at most $Nm\lceil \log k \rceil$ bits;
3. matrices $h_1, h_2, \ldots, h_{Nm}$ each minus its redundant column, in $Nnm^2 - Nm^2$ bits.

From this description the nonredundant columns of the h_i's and x, and from the y_i's, the program P, given B as an oracle, will construct h_i. The total length of this description is only

$$Nnm^2 - Nm^2 + \lceil \log k \rceil + Nm(\lceil \log k \rceil) \leq Nnm^2 - (N-1)m - 1.$$

Hence $C(z|B, P, m, n, N) < |z| - (N-1)m$ a contradiction. $\square$

Recall that $\oplus$SAT is the set of boolean formulae with an odd number of satisfying assignments.

3.1 Constructing M_U

We construct a nondeterministic polynomial time bounded transducer M_U that takes as input a boolean formula F and an advice string τ, and under the assumption that τ has certain properties (in which case we call τ *good advice*), does the following on a fraction (bounded from below by a function of $|F|$) of its paths: M_U produces a boolean formula F' from F, such that F' is in USAT if and only if F is satisfiable. If the input τ is not good advice the behavior of M_U is unspecified.

We first define good advice. As we are interested in NP sets, we restrict our attention to the case where the set B of Lemma 3.1 is not arbitrary but is related to SAT in the following manner. Let F be a boolean formula and let B_F be the set $\{y \mid y \text{ satisfies } F\}$. Let $N \geq n \geq 1$. Define S to be the set:

$$S(m,n,N) = \{F \mid |F| = N,\ F \text{ has } n \text{ variables and } 2^{m-2} < |B_F| \leq 2^{m-1}\}.$$

Let $2 \leq m \leq n+1$, as we wish to set aside the special cases when $0 \leq B_F \leq 1$.

Let a string z be *good* if z encodes a family of hash functions H such that H separates B_F within B_F, and be *bad* otherwise. Then, by Lemma 3.1, we have that any z such that $|z| = Nnm^2$ and $C(z|B_F, P, m, n, N) \geq Nnm^2 - (N-1)m$ encodes a family of hash functions H such that H separates B_F within B_F; i.e, z is good. The total number of bad zs is at most $2^{Nnm^2-(N-1)m+1} - 1$. But $|S(m,n,N)| \leq 2^N$ as $F \in S(m,n,N)$ is restricted to be of length N, and therefore the number of Nnm^2 bit strings that are bad for any $F \in S(m,n,N)$ is $2^N \times (2^{Nnm^2-(N-1)m+1} - 1) < 2^{Nnm^2-(N-1)m+N+1}$. Note that $m \geq 2$ implies that $Nnm^2 - (N-1)m + N + 1 < Nnm^2$. Therefore, amoungst all the strings of length Nnm^2, there exists some string z that is good for all $F \in S(m,n,N)$.

Therefore, as a corollary to the proof of Lemma 3.1, we have:

Corollary 3.2 *For all $N \geq n \geq 1$ and $m \geq 2$, a Kolmogorov random string z of length Nnm^2, where $C(z) \geq Nnm^2 - (N-1)m$, exists such that z encodes a collection H of Nm random linear transformations $\Sigma^n \to \Sigma^m$ such that, for all $F \in S(m,n,N)$, H separates B_F within B_F.*

We now define good advice. The string τ is called *good advice for length N* if $\tau = \langle \tau_1, \ldots, \tau_n, \ldots, \tau_N \rangle$ and τ_n is good advice for all formulas with n variables. The string τ_n is good advice for all formulas with n variables if τ_n encodes a collection of n Kolmogorov random strings $\langle H_1, \ldots, H_m, \ldots, H_{n+1} \rangle$, where $H_m = h_{m1} h_{m2} \ldots h_{m \times Ni}$ and h_{mj} within H_m is of length mn and encodes a single hash function from $\Sigma^n \to \Sigma^m$. Note that H_m is of length Nnm^2 for each $m \leq n+1$ respectively. Thus $|\tau_n| < N^5$, implying $|\tau| < N^6$.

Every input $F \in \Sigma^{=N}$ encodes a formula of $n \leq N$ variables, and has less than $2^n + 1$ satisfying assignments. Thus there exists a unique n and m such that $F \in S(m,n,N)$. By the above description τ_n contains the substring H_m which is the string z of Corollary 3.2. Therefore τ is good advice to all formulae with less than $n+1$ variables, for our algorithm reducing SAT to USAT.

Input F with n variables;
Input an advice string $\tau = \langle \tau_1, \ldots, \tau_{|F|} \rangle$
Extract τ_n from τ
1. Guess $m \leq n + 1$
2. Guess an $i \leq Nm$ and a y of length m
if $m \leq 1$ **then**
  Output $F' = F$
else
  Output F' stating : $F \wedge (h_{mi}(x_1, \ldots, x_n) = y)$
end(* **if** *)

M_U is the following algorithm:

F' is unsatisfiable if F is unsatisfiable. Assume F is satisfiable, and τ is good advice; i.e, $\tau_n = \langle H_1, H_2, \ldots, H_{n+1} \rangle$. Let $B_F = B = \{z \mid z$ is a satisfying assignment of $F\}$. Then $0 < |B| \leq 2^n$. If F is already in USAT then with probability $1/(n+1)$ we output $F' = F$. If, on the otherhand $|B| \geq 2$, there exists a unique m such that $2^{m-2} < |B| \leq 2^{m-1}$.

1. Suppose the choice of m in step 1 is such that $2^{m-2} < |B| \leq 2^{m-1}$. By Corollary 3.2, H_m is a family of hash functions that separate B within B.

2. For any fixed $z \in B$, there exists i_z and an $h_{mi_z} \in H_m$ such that $\forall z' \in B$, if $z' \neq z$ then $h_{mi_z}(z') \neq h_{mi_z}(z)$. Fix a satisfying assignment z. Suppose the choice of $i = i_z$ in step 1 is such that h_{mi} separates z within B.

3. Suppose the choice of y in step 2 is such that $y = h_{mi}(z)$. Note that no other assignment z' that satisfies F can map to y, i.e. $h_{mi}(z') \neq y$.

For these choices, the formula, $F' = F \wedge (h_{mi}(x_1, \ldots, x_n) = y)$, is uniquely satisfiable if $F \in$ SAT, as all assignments in $B - \{z\}$ fail to satisfy the second clause of F', and unsatisfiable otherwise. The probability that a randomly guessed h_{mi} will separate an assignment z within B, by mapping it to a randomly guessed y, is therefore the number of satisfying assignments multiplied by the probability of success in hitting the appropriate h_{mi} and y for each satisfying assignment, which is $|B|/(Nm \times 2^m) > 1/(4Nm)$.

The different computations of M_U correspond to different choices of i and y. If τ is good advice and given that we choose the correct value of m with probability $1/(n+1)$, we have that $1/(4N(n+1)m) > 1/(4N^3)$ of the paths of M_U compute a reduction from NP to USAT; i.e., on these paths, if F is satisfiable then F' is in USAT, and if F is not satisfiable then all paths of M_U produce unsatisfiable formulae.

3.2 Constructing $M_{\rm BP}$

Assume τ is good advice and that the input formula F has n variables. In this case, the probability of error in converting F to a USAT formula by the algorithm M_U is 0 if $F \notin$ SAT but is $1 - 1/(4N^3)$ otherwise. We define an nondeterministic transducer $M_{\rm BP}$ which, under the same assumption, boosts the probability of an arbitrary path transforming F into an F' such that F is satisfiable if and only if F' is in $\oplus$SAT. We use the well-known technique of amplifying the probability by OR-ing repeated trials. For each trial we rename the variables of F, choose a new m and y, and reuse τ_n (amplification does not increase the advice length).

The following construction is from [10]. Given t boolean formulas $F_1, \ldots, F_t$, each pairwise over a distinct set of variables, we construct a formula $\text{OR}(F_1, \ldots, F_t) = F$, and F has an odd number of satisfying assignments if and only if for some choice of $i \leq t$, F_i has an odd number of satisfying assignments. Let F be a boolean formula, and let y_F be a variable not in F. Then, we define functions f and h:

$$f(F) = (F \wedge \neg y_F) \vee (x_1 \wedge x_2 \wedge \ldots \wedge x_n \wedge y_F)$$

and let $h(F_1, F_2, \ldots, F_t) = F_1 \wedge F_2 \wedge \ldots \wedge F_t$. Then the required formula is

$$\text{OR}(F_1, \ldots, F_t) = f(h(f(F_1), \ldots, f(F_t))).$$

The transducer $M_{\rm BP}$ is as follows

Input F with n variables, and integer t
Input an advice string τ
for $i = 1$ **to** t **do**
  Rename the variables of F
  Guess a computation path of $M_U(F, \tau)$
  Let the formula produced on this path be F_i
end($*$ **for** $*$)
Output the formula $F'' = \text{OR}(F_1, \ldots, F_t)$

Under the assumption that τ is good advice, the probability that F' has an even number of assignments even though F is satisfiable is $(1 - 1/4N^3)$. This inverse polynomial probability of error can be made less than a constant, say $1/4$, by repeating $t > 4N^3$ trials and OR-ing them together. Then the probability that F'' has an odd number of accepts is greater than $3/4$. Thus we can boost the error probability of the above algorithm to $3/4$ and by further OR-ing N^2 trials, to inverse exponential probability: $1 - 1/2^{2N^2}$.

3.3 Constructing $M_{\oplus \mathrm{P}}$

Recall the definition of positive reducibility from [15]:

Definition 3.3 *[15] For two sets A and B, we write $A \in \mathrm{P}_{pos}(B)$ if and only if $A \in \mathrm{P}(B)$ is witnessed by some oracle machine M with the additional property that for all oracle sets X and Y, $X \subseteq Y \Longrightarrow L(M,X) \subseteq L(M,Y)$. A class $\mathcal{C}$ is closed under P_{pos} if $A \in \mathrm{P}_{pos}(B)$ and $B \in \mathcal{C}$ then $A \in \mathcal{C}$.*

It is well known that P, PSPACE, NP, $\oplus$P, and PH closed under positive reducibility. Recall the definition of the BP operator applied to a class $\mathcal{C}$ [15].

Lemma 3.4 (Amplification lemma) *[15] Let the class $\mathcal{C}$ be closed under P_{pos}. For every A in $\mathrm{BP}.\mathcal{C}$ and every polynomial q, there exists a $B \in \mathcal{C}$ and a polynomial p such that,*

$$\forall n, \forall x \in \Sigma^{=n} \ \mathrm{Prob}_{y,|y|=p(|x|)} \left[x \in A \Longleftrightarrow \langle x, y \rangle \in B \right] > 1 - 2^{-q(|x|)}$$

We now prove another familiar fact from structural complexity using Kolmogorov complexity (the proof idea is by Gasarch[11]) :

Lemma 3.5 *If the class $\mathcal{C}$ is closed under positive reductions, then* $\mathrm{BP}.\mathcal{C} \subseteq \mathcal{C}/\mathrm{poly}$

Proof. From Lemma 3.4, for all $A \in \mathrm{BP}.\mathcal{C}$ and polynomials q, there exists a $B \in C$, a nondeterministic oracle Turing machine M, and a polynomial p, defining a $\#\mathrm{P}^{\mathcal{C}}$ function $f(x) = \#\mathrm{acc}_M{}^B(x)$, length-bounded by p such that:

$$\forall x \in \Sigma^{=n} \ [x \in A \Longrightarrow f(x) > 2^{p(n)-q(|x|)}] \ \text{ and } \ [x \notin A \Longrightarrow f(x) < 2^{q(|x|)}]$$

Let $q(n) = n^2$ and $p(n) = n^k$ in the above, i.e, let f be computed by a NDTM M, bounded by polynomial n^k, and let the probability of error be bounded by 2^{-n^2}. Let $R(x,y)$ be the P predicate "M on input x accepts along path y", and let $\#R(x, y)$ be the function computing "the number of accepts to the left of and including path y". Note $\#R$ is in #P.

Claim: *Let τ be a n^k bit string such that $C(\tau) > n^k - 1$. Then $\forall x \in \Sigma^{=n}$ $[x \in A \iff$* $R(x, \tau)$ is an accept].

Proof of claim: Assume there exists an x, $|x| = n$, such that the claim fails. Assume $R(x, \tau)$ is an accept and $x \notin A$ (the converse case is similar). Then the number of accepts of $M(x)$ is at most 2^{n^2}, implying $|\#R(x, \tau)| \leq n^2$. Then the following is a short description of τ:

1. One bit to tell us that $R(x, \tau)$ is an accept.
2. M and k, requiring a constant number of bits c.
3. x, requiring n bits.
4. $\#R(x, \tau)$, requiring n^2 bits.
5. a constant number of bits to describe the following procedure: Run M on x for n^k steps on every path $y \in \Sigma^{=n^k}$ from left to right, counting the number of accepts seen. Output the path y when the count reaches $\#R(x, \tau)$.

Clearly this program outputs τ yet needs only $n^2 + n + c$ bits which, for $n^k \notin \mathcal{O}(n^2)$, leads to contradiction.

Hence Lemma 3.5 follows. $\square$

Theorem 3.6 NP $\subseteq \oplus$P/poly

Proof. The algorithm M_U originally uses less than $(N+1)\log N + N$ random bits to achieve an error probability of $1 - 1/(4N^3)$. For amplification M_{BP} needs $((N+1)\log N + N)(4N^3)N^2 < cN^7$ random bits.

Consider the algorithm $M_{\oplus\mathrm{P}}$:

Input F with n variables and advice strings τ and τ'
Use τ' to select a single computation path of $M_{\mathrm{BP}}(F, N^5, \tau)$
Output the formula F'' produced on this path

Thus we need $\mathcal{O}(N^6)$ bits of advice for the coding lemma's hash functions and $\mathcal{O}(N^7)$ bits of advice to choose a path that always produces a formula F'' in $\oplus$P if F is in SAT.

Note that, under the assumption that τ is good advice, M_{BP} produces a $\oplus$SAT formula from a SAT formula with very high probability. As mentioned earlier, M_{BP} uses randomness, and can be stripped of its random bits, by substituting a sufficiently long Kolmogorov hard string as advice (by sufficiently long we mean "of the same length as the number of random bits guessed"). This is τ'. As the random bits are chosen independent of the advice τ, we can append the two Kolmogorov random strings as a single piece of polynomially length bounded advice.

Hence Theorem 3.6 follows. $\square$

The structure of the above proof illustrates how Kolmogorov complexity arguments can be substituted in place of probability arguments, and gives a simple interpretation to the advice needed by the $\oplus$P/poly algorithm for SAT above, namely that the advice need only be incompressible to be good advice.

Acknowledgments

I thank Tim Long for his many helpful comments and corrections.

References

[1] Allender, E., and Hemachandra, L. Lower bounds for the low hierarchy. *J. ACM 39*, 1 (1992), 234–251.

[2] Balcázar, J., and Book, R. Sets with small generalized Kolmogorov complexity. *Acta Informatica 23* (1986), 679–688.

[3] Balcázar, J., Diaz, J., and Gabarró, J. *Structural Complexity I.* Springer-Verlag, Berlin, 1988.

[4] Balcázar, J., Diaz, J., and Gabarró, J. *Structural Complexity II.* Springer-Verlag, Berlin, 1990.

[5] Chari, S., and Rohatgi, P. An alternate proof of Toda's theorem. submitted to IPL.

[6] Gavaldá, R. Bounding the complexity of advice functions. In *Proc. Seventh Annual Conference Structure in Complexity Theory* (1992), IEEE Computer Society Press, pp. 249–254.

[7] Gavaldá, R. *Kolmogorov Randomness and Its applications to Structural Complexity Theory.* PhD thesis, Universitat Politecnica de Catalunya, April 1992.

[8] Karp, R., and Lipton, R. Some connections between nonuniform and uniform complexity classes. In *Proc. 12th ACM Symp. on Theory of Computing* (1980), pp. 302–309.

[9] Köbler, J. Locating P/poly optimally in the extended low hierarchy. Tech. Rep. Nr. 92-05, Ulmer Informatik-Berichte, 1992.

[10] Köbler, J., Schöning, U., and Toran, J. *The Graph Isomorphism Problem: Its Structural Complexity.* Birkhauser, 1993.

[11] Li, M., and Vitanyi, P. *An Introduction to Kolmogorov Complexity and its applications.* Springer-Verlag, 1993.

[12] Long, T. On restricting the size of oracles compared with restricting access to oracles. *SIAM J. of Comput. 14*, 3 (1985), 585–597.

[13] Papadimitriou, C., and Zachos, S. Two remarks on the power of counting. In *Proc. sixth GI conf. on Theoretical Comp. Sc.* (1982), pp. 269–276. Springer-Verlag,LCNS, #145.

[14] Regan, K. W., and Royer, J. S. A simpler proof of PH $\subseteq$ BP[$\oplus$P]. preprint.

[15] Schöning, U. Probabilistic complexity classes and lowness. In *Proc. Conference on Structure in Complexity Theory* (1987), A. Selman, Ed., Springer-Verlag, pp. 2–8.

[16] Toda, S. On the computational power of PP and $\oplus$P. In *Proc. 30th IEEE Symp. on Foundations of Computer Science* (1989), pp. 514–519.

[17] Toda, S., and Ogiwara, M. Counting classes are at least as hard as the the polynomial-time hierarchy. Tech. Rep. CSIM 90-09, University of Electro-Comm., Department of Computer Science and Information Mathematics, Chofu-shi, Tokyo 182, Japan, 1990.

[18] Valiant, L., and Vazirani, V. NP is as easy as detecting unique solutions. In *Proc. 17th ACM Symp. Theory of Computing* (1986), pp. 458–463.