Notre Dame Journal of Formal Logic
Volume 42, Number 2, 2001

Forcing Complexity: Minimum Sizes of Forcing Conditions

Toshio Suzuki

Abstract This note is a continuation of our former paper "Complexity of the r-query tautologies in the presence of a generic oracle." We give a very short direct proof of the nonexistence of t-generic oracles, a result obtained first by Dowd. We also reconstitute a proof of Dowd's result that the class of all r-generic oracles in his sense has Lebesgue measure one.

1 Introduction

In a series of our former papers (Suzuki [4, 6, 7]), by extending Dowd's pioneering work [2], we studied complexity issues on minimum sizes of forcing conditions. This short note is a continuation of Suzuki [6]. In [6], we produced an NP predicate, and by using it as a tool, we gave a (very short) proof of the fact that the class of t-generic oracles has measure zero; in this note, a better chosen NP predicate provides an equally short proof of a more drastic fact: this class is empty. The original proof of this last fact, by Dowd [2], was less direct; moreover, in [6], we showed the original proof's logical gap by presenting a counterexample. We also reconstitute a proof of the fact that for each positive integer r, the class of all r-generic oracles in the sense of Dowd has Lebesgue measure one. The original proof of this fact [2] was difficult to understand. The preliminary version of this note was cited in Suzuki [5] and [7] as "Forcing complexity: Supplement to complexity of the r-query tautologies."

We refer to [6] and [7] for history, motivations, definitions, and notations. We state only a minimum of definitions here. In the sequel X is a symbol for an *oracle*, that is, a set of bit strings, or better, the characteristic function of such a set. A *forcing condition* means a restriction of an oracle to a finite domain. By "forcing complexity" we mean the minimum size of forcing conditions that force a given predicate. More formally, it is stated as follows. Let y be a variable for a bit string. Assume $\varphi(X, y)$ is an arithmetical predicate. Further, we assume φ is *finitely testable*

Received September 25, 1998; accepted April 19, 2002; printed May 22, 2003
2001 Mathematics Subject Classification: Primary, 68Q15; Secondary, 03D15
Keywords: computational complexity, t-generic oracle

(Poizat [3]), that is, there exists a function $f : \mathbb{N} \to \mathbb{N}$ such that for every oracle A and every bit string u, $\varphi(A, u)$ holds if and only if $\varphi(B, u)$ holds, where B is the extension of $A \upharpoonright (\{0, 1\}^{\leq n})$ such that $B(u) = 0$ for all u such that $|u| > n$. Assume A is an oracle and n is a natural number. *The forcing complexity of φ relative to A at n* is the least natural number k of the following property: for any bit string u of length n, if $\varphi(A, u)$ is true then A has a finite portion S of size at most k such that S forces $\varphi(X, u)$: in other words, the cardinality of dom(S) is at most k and for any oracle B extending S, $\varphi(B, u)$ is true. An oracle is called *t-generic* if relative to which the forcing complexity of (the property of being) relativized tautologies is at most polynomial.

2 The Nonexistence of t-generic Oracles

We define an arithmetical (in fact, NP) predicate CS(X, y) as follows.

Definition 2.1 y is not the empty string and, letting $n = |y| - 1$, there exist strings $u_1, \ldots, u_{n+1}$ of length n, which are consecutive in the lexicographic ordering, such that $y = X(u_1), \ldots, X(u_{n+1})$. And coCS$(X, y)$ is the negation of it. The letters CS are for ‘consecutive strings’.

Theorem 2.2 *Assume that A is an oracle and n is a natural number. Then the forcing complexity of* coCS *relative to A at $n + 1$ is at least* $(2^n - n)/(n + 1)$.

Proof Assume for a contradiction that the forcing complexity is less than $(2^n - n)/(n+1)$. Among bit strings of length $n+1$, at most 2^n bit strings vs make the assertion CS(A, v) true, so that we can find one of them, say u, for which coCS(A, u) is true. By hypothesis, coCS(X, u) (where only u is fixed) is forced by a finite portion S of A, whose domain has d elements, where $d < (2^n - n)/(n + 1)$.

Since $n(d + 1) < 2^n - d$, the set R of strings of length n which are not in the domain of S contains more than $n(d + 1)$ elements. This set R being composed of a maximum of $d + 1$ intervals for the lexicographic ordering, one of them contains at least $n + 1$ elements. Therefore, we obtain an oracle B extending S and satisfying CS(B, u), a contradiction. □

In [6], §3, by using the predicate CORANGE of Bennett and Gill [1], we presented a short proof of the fact that the class of all t-generic oracles has Lebesgue measure zero. By using coCS in place of CORANGE, we drastically improve it as follows.

Corollary 2.3 ([2], Lemma 7) *t-generic oracles do not exist.*

Proof According to [2], §3, a t-generic oracle should force any coNP predicate with the help of one of its polynomial-sized fragments, contradicting Theorem 2.2. □

3 Reconstitution: r-generic Oracles

For a positive integer r, an oracle is *r-generic in the sense of Dowd [2]* if it satisfies the definition of a t-generic oracle for r-query tautologies in place of relativized tautologies. This r-genericity is completely different from that of arithemtical forcing. In the remaininig part of this note, r-genericity always means that of Dowd, though we do not think it is a good terminology (in [7], in order to avoid confusion, we used the terminology *r-Dowd oracles* instead of r-generic oracles in the sense of Dowd). In this section, we reconstitute a proof of the following fact.

Fact 3.1 ([2], Theorem 10) For every positive integer r, the class of all r-generic oracles has Lebesgue measure one in the Cantor space.

Dowd proved his Theorem 10 by using the following fact.

Fact 3.2 ([2], Lemma 9) If F is a 1-query tautology with respect to some oracle then there is a unique minimal forcing condition S that forces F (to be a tautology).

Here the assumption of 1-query is critical. For example, if G is a 2-query formula asserting that exactly one of two strings 001 and 101 belongs to a given oracle then there are two minimal forcing conditions that force G. Therefore, when $r \geq 2$, we cannot rely on the uniqueness of the minimal forcing condition that forces a given formula. However, Dowd's original proof of his Theorem 10 ([2], p. 70, l. 33–p. 71, l. 14) seems to rely on the uniqueness. And it is difficult to understand the structure of induction used there. In [5], Chapter 4, a proof of Dowd's Theorem 10 was rigorously reconstituted but it was long and complicated. Here we sketch it without detail. There are two important ideas.

The first is a "partial version" of forcing complexity. Suppose $r \geq 2$. For each oracle A, we define a certain subset of r-query tautologies with respect to A; for the time being, we call them *nice r-query tautologies with respect to A*. They have the following property (a revised version of Dowd's Lemma 9): If F is a nice r-query tautology with respect to some oracle X and if T is a (certain special type of) finite portion of X, then there is a unique minimal forcing condition S_T such that the union of S_T and T forces F.

The second important idea is oracles' hierarchy with respect to forcing complexity. Suppose $r \geq 2$. Let $r\mathrm{GEN}_1$ be the class of all oracles D such that for every nice r-query tautology F and every (certain type of) finite portion T of D, the S_T has polynomial size of $|F|$. Let $r\mathrm{GEN}_2$ be the class of all oracles for which the forcing complexity of (the property of being) nice r-query tautologies is at most polynomial. For each $r \geq 1$, let $r\mathrm{GEN}_3$ be the class of all r-generic oracles. We get the following scheme.

$$\begin{array}{ccccccc} & & 2\mathrm{GEN}_1 & \supseteq & 3\mathrm{GEN}_1 & \supseteq \cdots \\ & & \mid\!\bigcup & & \mid\!\bigcup & \\ & & 2\mathrm{GEN}_2 & \supseteq & 3\mathrm{GEN}_2 & \supseteq \cdots \\ & & \mid\!\bigcup & & \mid\!\bigcup & \\ 1\mathrm{GEN}_3 & \supseteq & 2\mathrm{GEN}_3 & \supseteq & 3\mathrm{GEN}_3 & \supseteq \cdots \end{array}$$

Claim 3.3 *For each $r \geq 2$, $r\mathrm{GEN}_1$ has Lebesgue measure one.*

Sketch of proof The proof is similar to Dowd's proof of the existence of 1-generic oracles ([2], p. 70). Instead of Dowd's Lemma 9, we use the revised version of it. See also [7], §4. □

Claim 3.4 *We have* $2\mathrm{GEN}_1 \subseteq 1\mathrm{GEN}_3$.

Sketch of proof This is shown by adding dummy symbols to a given 1-query formula. □

Claim 3.5 *For each $r \geq 1$, we have* $r\mathrm{GEN}_3 \cap (r+1)\mathrm{GEN}_1 \subseteq (r+1)\mathrm{GEN}_2$.

Sketch of proof This is similar to Dowd's original proof of Theorem 10 for $r \geq 2$ ([2], p. 71, ll. 6–13). We use the revised version of Dowd's Lemma 9. □

Claim 3.6 *For each $r \geq 1$, we have $r\mathrm{GEN}_3 \cap (r+1)\mathrm{GEN}_2 \subseteq (r+1)\mathrm{GEN}_3$.*

Sketch of proof An $(r+1)$-query tautology is equivalent to a certain "simultaneous equation" consisting of r-query tautologies and nice tautologies where the number of these tautologies is at most polynomial because r is fixed. □

Now we show that each vertical hierarchy collapses. By induction on r with Claim 3.4, 3.5, and 3.6, we have $(r+1)\mathrm{GEN}_1 \subseteq r\mathrm{GEN}_3$ and $(r+1)\mathrm{GEN}_1 = (r+1)\mathrm{GEN}_2 = (r+1)\mathrm{GEN}_3$, for each $r \geq 1$. By this fact and Claim 3.3, for each $r \geq 1$, $r\mathrm{GEN}_3$ has Lebesgue measure one. Thus, we have shown Dowd's Theorem 10.

A problem that we still leave open is whether or not the horizontal hierarchy collapses. See [7] for a partial solution to this problem.

References

[1] Bennett, C. H., and J. Gill, "Relative to a random oracle A, $\mathbf{P}^A \neq \mathbf{NP}^A \neq \mathrm{co}-\mathbf{NP}^A$ with probability 1," *SIAM Journal on Computing*, vol. 10 (1981), pp. 96–113. Zbl 0454.68030. MR 83a:68044. 118

[2] Dowd, M., "Generic oracles, uniform machines, and codes," *Information and Computation*, vol. 96 (1992), pp. 65–76. Zbl 0755.68052. MR 93c:68022. 117, 118, 119

[3] Poizat, B., "$\mathcal{Q} = \mathcal{NQ}$?" *The Journal of Symbolic Logic*, vol. 51 (1986), pp. 22–32. Zbl 0592.03027. MR 88g:68040. 118

[4] Suzuki, T., "Recognizing tautology by a deterministic algorithm whose while-loop's execution time is bounded by forcing," *Kobe Journal of Mathematics*, vol. 15 (1998), pp. 91–102. Zbl 0937.68052. MR 2000k:68073. 117

[5] Suzuki, T., *Computational Complexity of Boolean Formulas with Query Symbols*, Ph.D. thesis, University of Tsukuba, Tsukuba-City, 1999. 117, 119

[6] Suzuki, T., "Complexity of the r-query tautologies in the presence of a generic oracle," *Notre Dame Journal of Formal Logic*, vol. 41 (2000), pp. 142–151. Zbl 01881631. MR 1932226. 117, 118

[7] Suzuki, T., "Degrees of Dowd-type generic oracles," *Information and Computation*, vol. 176 (2002), pp. 66–87. Zbl 01853272. MR 2003f:68045. 117, 118, 119, 120

Acknowledgments

I would like to thank Professor Hisao Tanaka and Professor Katsuya Eda for their encouragement and useful discussions. I would also like to thank the anonymous referee and editors for their kind advice. This research was partially supported by Grant-in-Aid for Scientific Research (Nos. 09440072, 11740073, 14740082), Ministry of Education, Science, Sports, and Culture of Japan.

Department of Mathematics and Information Sciences
Osaka Prefecture University
Sakai Osaka 599-8531
JAPAN
suzuki@mi.cias.osakafu-u.ac.jp
http://wwwmi.cias.osakafu-u.ac.jp/~suzuki/index.html