Recognizing Tautology by a Deterministic Algorithm Whose While-loop's Execution Time Is Bounded by Forcing

Toshio Suzuki

Osaka Prefecture University
Sakai, Osaka 599-8531, Japan
suzuki@mi.cias.osakafu-u.ac.jp

Kobe Journal of Mathematics 15, pp. 91–102 (1998)
MR 2000k:68073 68Q25 (03B05 03B35 03D15)

This is a preprint version of the above paper.
©1998 Toshio Suzuki & Kobe University

Recognizing Tautology by a Deterministic Algorithm Whose While-loop's Execution Time is Bounded by Forcing

Toshio Suzuki
Department of Mathematics and Information Sciences
Osaka Prefecture University, Sakai, Osaka 599-8531, Japan
suzuki@mi.cias.osakafu-u.ac.jp

(Received June 9, 1997; Revised September 7, 1998)

Abstract

By Bennet and Gill [4], it is shown that if A is a random oracle then $TAUT^A \notin P^A$ with probability 1, where $TAUT^A$ denotes the collection of all tautologies relative to A. Extending Dowd's work [6], we present a forcing argument to bound execution time of a while-loop of a deterministic algorithm, by which we show that for each positive integer r, if A is an r-generic oracle in the sense of Dowd then $rTAUT^A \equiv_T^P TAUT \oplus A$, where $rTAUT^A$ denotes the collection of all r-query tautologies with respect to A. As a consequence, the following two assertions are equivalent : (i) if A is a random oracle then $rTAUT^A \notin P^A$ with probability 1, (ii) $R \neq NP$.

1 Introduction

The terminology "n-generic oracle" has been traditionally used for Feferman's arithmetical forcing [7]. However, some people have introduced variations of the notion of generic oracles, and consequently there are some concepts also called n-generic oracles, which are completely different from Feferman's n-genericity. The notion of r-generic oracles due to Dowd is an example of such concepts, which plays an important role in this paper. Dowd [6] introduced his r-generic oracles in the study of the relationship between uniform machines and $NP = ?coNP$ question, and he investigated some complexity issues about the relativized propositional calculus. As Dowd showed (Theorem 12 of [6]), if A is a Feferman generic oracle, in other words, if A is a Cohen generic oracle over the dense arithmetical sets, then A cannot be a 1-generic oracle in Dowd's sense. On the other hand, for each positive integer r, the collection of all r-generic oracles in Dowd's sense has Lebesgue measure one in the Cantor space (section 4 of [6]).

Now, to explain our problem, we review the behavior of a $coNP^A$-complete set in the case where A is a random oracle. Suppose that n is a positive integer and $q_1, \ldots, q_n$ are Boolean variables. For each n, we introduce a new connective $\xi^n(q_1, \ldots, q_n)$ which intuitively means that the bit string $q_1 \cdots q_n$ belongs to a given oracle. (In the next section, we shall present a formal definition that is slightly different from this one.) Then, for all natural numbers n, we add connectives ξ^n to the propositional calculus. The resulting system is called the relativized propositional calculus. Suppose that r is a positive integer. A relativized formula that has exactly r occurrences of the additional connectives is called an r-query formula. Suppose that A is an oracle i.e. a set of bit strings of finite lengths. We denote by $TAUT^A$ the collection of all formulas of the relativized propositional calculus which are tautologies with respect to A. $rTAUT^A$ denotes the collection of all r-query formulas which belong to $TAUT^A$. In [4], Bennet and Gill showed that if A is a random oracle then $P^A \neq NP^A$ with probability 1. Since $TAUT^A$ is a $coNP^A$-complete set for an arbitrary oracle A, we obtain the following as its direct corollary.

Fact 1 *If A is a random oracle then $TAUT^A \notin P^A$ with probability* 1.

We consider the problem whether the statement of the above fact remains true when we substitute $rTAUT^A$ for $TAUT^A$. Extending Dowd's theory of r-generic oracles [6], we shall present a forcing argument to bound execution time of a while-loop of a deterministic oracle Turing machine, from which we shall show the following theorem, where $TAUT$ denotes the collection of all tautologies of the usual propositional calculus.

Theorem 1 *Suppose that r is a positive integer. Then, for each A that is an r-generic oracle in Dowd's sense, we have the following.*

$$rTAUT^A \equiv_T^P TAUT \oplus A. \tag{1.1}$$

When A is a Feferman generic oracle, we observe that (1.1) does not hold. Let $L[A]$ be a tally set defined by $L[A] = \{0^n : \neg\exists y \in A\,(|y| = n)\}$ (see [2]). Then, $L[A]$ is polynomial time many-one reducible to $1TAUT^A$. However, it is not polynomial time Turing reducible to $TAUT \oplus A$ whenever A is a Feferman generic oracle. Whereas, we do have the above formula (1.1) for A that is an r-generic oracle in the sense of Dowd.

Our conclusion in this paper is the following theorem.

Theorem 2 *Suppose that r is a positive integer. Then, the following two statements are equivalent.*

(1) If A is a random oracle then $rTAUT^A \notin P^A$ with probability 1.

(2) The unrelativized class R does not equal NP.

We shall prove Theorem 1 in section 4 and Theorem 2 as its corollary in section 5. However, to explain our motivation of proofs, we shall show Theorem 1 in the case where $r = 1$ separately in section 3.

2 Preliminaries

Our alphabet is $\Sigma = \{0,1\}$. For each bit string u, $|u|$ is its binary length and u_i is the ith bit of u $(i \geq 1)$. Thus, $u = u_1 u_2 \dots u_n$, where $n = |u|$. In addition, for each $i \in \{0,1\}$, we denote the concatenation of u and i by ui. The collection of all bit strings is denoted by $\{0,1\}^*$. We order the elements of $\{0,1\}^*$ in the canonical way:

$$\lambda, 0, 1, 00, 01, 10, 11, 000, \dots,$$

where λ is the empty string. $z(n)$ denotes the $(n+1)$st string in this order. A subset of $\{0,1\}^*$ is called either a language or an *oracle*. For an oracle A, the computational complexity class P^A consists of all languages recognized by polynomial time-bounded deterministic oracle Turing machines with the oracle A. For an oracle A and an oracle machine M^X, $\mathrm{Lang}(M^A)$ denotes the language recognized by M^A. In section 4, we use the notation $M[X]$ to denote an oracle machine instead of M^X. For two languages A and B, $A \leq_T^P B$ means $A \in P^B$, and $A \equiv_T^P B$ means the conjunction of $A \leq_T^P B$ and $B \leq_T^P A$. $A \oplus B$ denotes the join of A and B: $\{u0 : u \in A\} \cup \{v1 : v \in B\}$. We identify an oracle with its characteristic function, i.e., $A(u) = 1$ if $u \in A$, and $A(u) = 0$ if $u \notin A$.

In this paper, we call a function S a *finite function* if $\mathrm{dom}(S)$ (the domain of S) is a finite subset of $\{0,1\}^*$ and $\mathrm{ran}(S)$ (the range of S) is a subset of $\{0,1\}$. For a finite function S and an oracle A, $S \sqsubseteq A$ means that for all $u \in \mathrm{dom}(S)$, $S(u) = A(u)$. For a finite set X, $\mathrm{Card}(X)$ denotes its cardinality.

By adding a set $\{\xi^n(q_1, \dots, q_n) : n \geq 1\}$ of new connectives to the language of the propositional calculus, we define *the relativized propositional calculus.* Suppose r is a positive integer. A relativized formula of the following form is called an *r-query formula*,

$$\Big((a^{(1)} \Leftrightarrow \xi^{i_1}(q_1^{(1)}, \dots, q_{i_1}^{(1)})) \wedge \dots \wedge (a^{(r)} \Leftrightarrow \xi^{i_r}(q_1^{(r)}, \dots, q_{i_r}^{(r)}))\Big) \Rightarrow H,$$

where H contains no ξ^i's. Given A and n, we define an n-ary Boolean function A^n as follows. Let $u = u_1 u_2 \cdots u_n$ be an arbitrary bit string of length n. Then, there is a unique $j < 2^n$ such that $u = z(2^n - 1 + j)$. Thus, we set $A^n(u_1, u_2, \dots, u_n) = A(z(j))$. For example,

$$A^3(0,0,0) = A(\lambda), A^3(0,0,1) = A(0), \dots, A^3(1,1,1) = A(000).$$

When ξ^n is interpreted by the Boolean function A^n, $TAUT^A$ denotes the set of all (Gödel numbers of) relativized formulas which are tautologies with respect to A. For each $r \geq 1$, $rTAUT^A$ denotes the set of all r-query formulas which belong to $TAUT^A$. $TAUT$ denotes the set of all tautologies of the propositional calculus (without additional connectives).

For each natural number r, Dowd [6] introduced r-generic oracles, which have their origin in theory of forcing [9]. Suppose S is a finite function and F is a relativized formula. We say "*S forces $TAUT^X(F)$*"if F is a tautology with respect to an arbitrary oracle A such that $S \sqsubseteq A$. An oracle A is called

r-generic (in the sense of Dowd) if there exists a polynomial $p(x)$ such that for each (Gödel number of an) r-query formula F in $TAUT^A$, there exists a finite function $S \sqsubseteq A$ whose domain is of size at most $p(|F|)$, where S forces $TAUT^X(F)$.

A language L belongs to the computational complexity class R if and only if there exists a probabilistic polynomial time Turing machine M and a positive constant $\varepsilon < 1/2$ such that (1) and (2) below hold for every string u :

(1) $u \in L$ if and only if Prob[M accepts u] $\geq (1/2) + \varepsilon$,

(2) $u \notin L$ if and only if Prob[M accepts u] $= 0$.

It is well known that $P \subseteq R \subseteq NP$. See [3] for more about R (and BPP).

In our proof of Theorem 2, we shall identify an oracle A with the infinite binary sequence $A(z(0)), A(z(1)), A(z(2)), \ldots$. Thus the class of all oracles is identified with the *Cantor space* [10]. The statement (1) of Theorem 2 is equivalent to the assertion that the set $\{X : rTAUT^X \notin P^X\}$ has Lebesgue measure 1 in the Cantor space.

3 Proof of Theorem 1 for $r = 1$

In this section, we shall give a proof of Theorem 1 in the case where $r = 1$ to explain our motivation. In this special case, each tautology F has the unique minimal finite function that forces F. Let us recall the following two results by Dowd.

Fact 2 *(Lemma 9 of [6]) If F is a 1-query formula which is a tautology with respect to some X then there is a unique minimal set S of queries which forces F to be a tautology (we say F specifies S).*

Fact 3 *(Theorem 11 of [6]) If $NP = coNP$ then there is a nondeterministic oracle machine M uniformly accepting $TAUT^X$, such that with respect to any r-generic X (in the sense of Dowd), M accepts the members of $rTAUT^X$ in polynomial time.*

We shall show an analogue of Fact 3 for a deterministic oracle machine:

Lemma 3 *There exists a deterministic oracle Turing machine M^X for which* (1) *and* (2) *below hold.*

(1) For every oracle A, Lang$(M^{TAUT \oplus A}) = 1TAUT^A$.

(2) If A is 1-generic in the sense of Dowd, then $M^{TAUT \oplus A}$ accepts the members of $1TAUT^A$ in polynomial time.

In the proof of Fact 3, Dowd used a forcing argument to bound the lengths of bit strings written on a query tape by a nondeterministic oracle machine. However, we want to bound the computing time of a deterministic oracle machine.

To show the above Lemma 3, we shall carefully examine the computational complexity of a function which maps each 1-query formula F that is a tautology with respect to a given oracle X to the unique minimal finite function S such that F specifies S. We shall construct a transducer that has one while-loop so that, for each 1-query formula F, it computes a candidate for (the domain of) the specified finite function. The transducer computes (the domain of) the specified function S itself as long as input F is a tautology with respect to a given oracle. A forcing argument will put the bounds of the execution time of the while-loop. Finally, we shall check whether the candidate for the specified finite function really forces that F is a tautology.

Proof of Lemma 3. Let us define four predicates as follows.

$TAUTALL(F) \equiv_{def.}$ F is a formula of the relativized propositional calculus and F is a tautology with respect to any oracle.

$1QFORMULA(i, H, F) \equiv_{def.}$ i is a positive natural number, H is a query free formula and F is the 1-query formula "$(a \Leftrightarrow \xi^i(q_1, \ldots, q_i)) \Rightarrow H$ ".

$CRITICAL(u, i, H, F) \equiv_{def.}$ $|u| = i$, $1QFORMULA(i, H, F)$ is true and the following formula is not a tautology : "$(\bigwedge_{j=1}^{i}(q_j \Leftrightarrow u_j)) \Rightarrow H$ ".

$SEGMENT(u, i, H, F, \langle v^{(1)}, \ldots, v^{(m)} \rangle) \equiv_{def.}$ There exists w such that u is an initial segment of w, $CRITICAL(w, i, H, F)$ is true and no $v^{(j)}$ equals w $(j = 1, \ldots, m)$.

Hereafter, whenever we talk about a 1-query formula F, we assume that i and H satisfy $1QFORMULA(i, H, F)$.

$TAUTALL$ belongs to $coNP$ (see [6]) and $SEGMENT$ belongs to NP. Hence there are two deterministic polynomial time-bounded oracle Turing machines N_{ALL}^X and N_{SEG}^X such that Lang(N_{ALL}^{TAUT}) = $TAUTALL$ and Lang(N_{SEG}^{TAUT}) = $SEGMENT$. Using these machines, we construct two oracle machines T^X and M^X as follows.

```
**transducer** $T^Y$ (**input** $F$ : 1-query formula)
**begin**
  $List := \langle\rangle$; /* the empty list */
  **while** $N_{SEG}^Y$ accepts $\langle \lambda, i, H, F, List \rangle$ **do**
    $u := \lambda$;
    **for** $i$ **times do**
      **if** $N_{SEG}^Y$ accepts $\langle u0, i, H, F, List \rangle$
        **then** $u := u0$;
        **else** $u := u1$;
      **end-if**;
    **end-for**;
    Add $u$ to $List$;
  **end-while**;
  **output**( $List$ )
**end** $\{T^Y\}$
```

```
machine $M^{Y\oplus Z}$ (input $F$ : 1-query formula)
begin
  $List := T^{Y}(F)$;
  if $List$ is empty then accept;
  $\langle v^{(1)},\ldots,v^{(m)}\rangle := List$;
  $u_j := Z^i(v_1^{(j)},\ldots,v_i^{(j)})$, for each $j = 1,\ldots,m$;
  $G :=$ the formula "$(\bigwedge_{j=1}^{m}(u_j \Leftrightarrow \xi^i(v_1^{(j)},\ldots,v_i^{(j)}))\ ) \Rightarrow F$ ";
  if $N_{ALL}^{Y}$ accepts $G$
    then accept;
    else reject;
  end-if;
end $\{M^{Y\oplus Z}\}$
```

Suppose that F is a 1-query formula and u is a string whose length is i. When T^{TAUT} runs on input F, u is added to $List$ if and only if $CRITICAL(u,i,H,F)$ is true. Hence if F is a tautology with respect to an oracle A then it is accepted by $M^{TAUT\oplus A}$. Conversely, the final test in M^X guarantees that any formula accepted by $M^{TAUT\oplus A}$ is a tautology with respect to A. Hence (1) of Lemma 3 holds.

Next, we show (2). Note that if a finite function forces a given 1-query formula, then the function should be an extension of the finite function specified by the 1-query formula. Thus, if F is a 1-query formula and a finite function S forces $TAUT^X(F)$, then for each $j \le 2^n - 1$ such that $CRITICAL(z(2^i - 1 + j), i, H, F)$ is true, $z(j)$ belongs to dom(S). Therefore, for every A that is 1-generic in the sense of Dowd, there exists a polynomial $p(x)$ such that the while-loop of T^{TAUT} terminates within $p(|F|)$ steps whenever input F is a tautology with respect to A. □

Proof of Theorem 1 for $r = 1$. Now, for each A that is a 1-generic oracle in Dowd's sense, we construct a machine by adding an appropriate clock to M^X in Lemma 3. This machine witnesses $1TAUT^A \le_T^P TAUT \oplus A$. □

4 Proof of Theorem 1 in general case

In this section, we present a proof of Theorem 1 without the assumption of $r = 1$.

We begin with rewriting a given r-query formula so that the resulting formula is longer than the original one but it has fewer occurrences of query symbols. Then we shall investigate a fast algorithm for the rewriting. Let r and i be positive integers and H a query free formula. $\natural\langle r,i,H\rangle$ denotes the following r-query formula.

$$\Big(\bigwedge_{j=1}^{r}(a^{(j)} \Leftrightarrow \xi^i(q_1^{(j)},\ldots,q_i^{(j)}))\Big) \Rightarrow H.$$

$\sharp\langle r, i, H\rangle$ denotes the r-query formula below :

$$\Big[\textstyle\bigwedge_{j=1}^{r} \quad (a^{(j)} \Leftrightarrow \xi^{i}(q_1^{(j)}, \ldots, q_i^{(j)}))\Big] \Rightarrow$$
$$\Big[[\textstyle\bigwedge_{j=1}^{r} \quad \Big((\displaystyle\bigwedge_{k=1}^{i}(q_k^{(j)} \Leftrightarrow q_k^{(r+1)})) \Rightarrow (a^{(j)} \Leftrightarrow a^{(r+1)})\Big)] \Rightarrow H\Big].$$

Note that the length of $\natural\langle r, i, H\rangle$ is given by a polynomial of $r + i +$the length of H and the length of $\sharp\langle r, i, H\rangle$ is similarly bounded.

From now on, we fix a positive integer r throughout this section. For an oracle A, $(r+1)CRITICAL^{A}$ denotes the collection of all triples $\langle w, i, H\rangle$ for which the following two conditions hold.
(1) i is a positive integer, H is a query free formula and w is a bit string whose length is i.
(2) $\sharp\langle r, i, \bigwedge_{k=1}^{i}(w_k \Leftrightarrow q_k^{(r+1)}) \Rightarrow H\rangle$ does not belong to $rTAUT^{A}$.

Proposition 4 *Let i and m be positive integers and H a query free formula. And, let A be an oracle and $u, v^{(1)}, \ldots, v^{(m)}$ bit strings. Assume the following four conditions hold.*

(i) $|u| = m$,

(ii) $|v^{(l)}| = i$ *and* $u_l = A^{i}(v_1^{(l)}, \ldots, v_i^{(l)}) \quad (l = 1, \ldots, m)$,

(iii) G *is the query free formula below:*

$$\bigwedge_{l=1}^{m}\Big((\bigwedge_{k=1}^{i}(v_k^{(l)} \Leftrightarrow q_k^{(r+1)})) \Rightarrow (u_l \Leftrightarrow a^{(r+1)})\Big),$$

(iv) $\{v^{(1)}, \ldots, v^{(m)}\} = \{w : \langle w, i, H\rangle \in (r+1)CRITICAL^{A}\}$.

Then the following two assertions are equivalent.

(a) $\sharp\langle r, i, G \Rightarrow H\rangle \in rTAUT^{A}$.

(b) $\natural\langle r+1, i, H\rangle \in (r+1)TAUT^{A}$.

Proof. First, we show that (a) implies (b). Assume (a). Then, the formula $\natural\langle r+1, i, G \Rightarrow H\rangle$ belongs to $(r+1)TAUT^{A}$. However, by (ii) and (iii), $\natural\langle r+1, i, G\rangle$ also belongs to $(r+1)TAUT^{A}$. Hence (b) holds.

Next, we show that the negation of (a) implies the negation of (b). Assume $\sharp\langle r, i, G \Rightarrow H\rangle$ does not belong to $rTAUT^{A}$. Then, for some truth assignment $\mathcal{V}$, the following four formulas are true :

(1) $\bigwedge_{j=1}^{r}(a^{(j)} \Leftrightarrow A^{i}(q_1^{(j)}, \ldots, q_i^{(j)}))$,

(2) $\bigwedge_{j=1}^{r}\Big((\bigwedge_{k=1}^{i}(q_k^{(j)} \Leftrightarrow q_k^{(r+1)})) \Rightarrow (a^{(j)} \Leftrightarrow a^{(r+1)})\Big)$,

(3) G,

(4) $\neg H$.

We define a bit string w whose length is i as follows. For each $k \quad (k = 1, \ldots, i)$, let $w_k = \mathcal{V}(q_k^{(r+1)})$ i.e. the truth value of the atom $q_k^{(r+1)}$. Then, the following formula is true with respect to $\mathcal{V}$:

(5) $\bigwedge_{k=1}^{i}(w_k \Leftrightarrow q_k^{(r+1)})$.

Since (1), (2), (4) and (5) are true with respect to $\mathcal{V}$, $\langle w, i, H \rangle$ belongs to $(r+1)CRITICAL^A$. Therefore, by (iv), there is an integer $n \quad (1 \leq n \leq m)$ such that $w = v^{(n)}$. Then, the following formulas are true with respect to $\mathcal{V}$:

(6) $(\bigwedge_{k=1}^{i}(v_k^{(n)} \Leftrightarrow q_k^{(r+1)})) \Rightarrow (u_n \Leftrightarrow a^{(r+1)})$ (by (3)),

(7) $\bigwedge_{k=1}^{i}(v_k^{(n)} \Leftrightarrow q_k^{(r+1)})$ (by (5)),

(8) $a^{(r+1)} \Leftrightarrow A^i(q_1^{(r+1)}, \ldots, q_i^{(r+1)})$ (by (ii), (6) and (7)),

(9) $\bigwedge_{j=1}^{r+1}(a^{(j)} \Leftrightarrow A^i(q_1^{(j)}, \ldots, q_i^{(j)}))$ (by (1) and (8)).

Since (4) and (9) are true with respect to $\mathcal{V}$, $\natural\langle r+1, i, H \rangle \notin (r+1)TAUT^A$. Thus we have shown that the negation of (a) implies the negation of (b). □

Remark. We did not use (iv) to show that (a) implies (b). The statements (1), (8) and (9) of the above proof are not formulas of the relativized propositional calculus in the strict sense but the interpretations of formulas with respect to the particular oracle. However, we abuse terminology. □

Our next problems for a given formula $F = \natural\langle r+1, i, H \rangle$, are the following two.

How long is the rewritten formula $\sharp\langle r, i, G \Rightarrow H \rangle$?

How fast can we rewrite F by a deterministic algorithm ?

The length of the rewritten formula is determined by the cardinality of the following set:

$$B = \{w : \langle w, i, H \rangle \in (r+1)CRITICAL^A\}. \tag{4.1}$$

For the second problem, it is enough to construct a deterministic transducer that outputs a list of all the members of the above set B in polynomial time when it runs on input F.

Proposition 5 *Suppose i is a positive integer and H a query free formula. Let A be an oracle, S a finite function, and B the set given by (4.1). Assume $S \sqsubseteq A$, and assume that S forces $TAUT^X(\natural\langle r+1, i, H \rangle)$. Then,* $\mathrm{Card}(B) \leq \mathrm{Card}(\mathrm{dom}(S))$*.*

Proof. Let C be the following set :

$$\{z(j) : \langle z(2^i - 1 + j), i, H\rangle \in (r+1)CRITICAL^A\}.$$

Clearly, $\mathrm{Card}(B) = \mathrm{Card}(C)$. Therefore it is sufficient to show that $C \subseteq \mathrm{dom}(S)$.

Assume for a contradiction that $z(n) \in C \setminus \mathrm{dom}(S)$ for some n. Fix such an integer n and put $w = z(2^i - 1 + n)$. Recall that $X^i(w_1, \ldots, w_i) = X(z(n))$ for each oracle X. Since $w \in B$, there is a truth assignment $\mathcal{V}$ for which the following four formulas are true :

(1) $\bigwedge_{j=1}^{r}(a^{(j)} \Leftrightarrow A^i(q_1^{(j)}, \ldots, q_i^{(j)}))$,

(2) $\bigwedge_{j=1}^{r}\Big((\bigwedge_{k=1}^{i}(q_k^{(j)} \Leftrightarrow q_k^{(r+1)})) \Rightarrow (a^{(j)} \Leftrightarrow a^{(r+1)})\Big)$,

(3) $\bigwedge_{k=1}^{i}(w_k \Leftrightarrow q_k^{(r+1)})$,

(4) $\neg H$.

Define an oracle D as follows. Let $D(z(n)) = \mathcal{V}(a^{(r+1)})$. For strings $u \neq z(n)$, let $D(u) = A(u)$. Then, the following two are true with respect to $\mathcal{V}$:

(5) $(a^{(r+1)} \Leftrightarrow D^i(q_1^{(r+1)}, \ldots, q_i^{(r+1)}))$ (by (3)),

(6) $\bigwedge_{j=1}^{r}(a^{(j)} \Leftrightarrow D^i(q_1^{(j)}, \ldots, q_i^{(j)}))$ (by (1), (2) and (3)).

Since (4), (5) and (6) are true with respect to $\mathcal{V}$, $\natural\langle r+1, i, H\rangle \notin TAUT^D$.

Since $z(n)$ does not belong to $\mathrm{dom}(S)$, we have $S \sqsubseteq D$. Therefore, S does not force $TAUT^X(\natural\langle r+1, i, H\rangle)$; thus we get a contradiction. □

Given an oracle A, let $(r+1)SEGMENT^A$ denote the collection of all sequences of the form $\langle u, i, H, \langle v^{(1)}, \ldots, v^{(m)}\rangle\rangle$ such that for some bit string w the following three conditions hold.

(1) u is an initial segment of w.

(2) $\langle w, i, H\rangle \in (r+1)CRITICAL^A$.

(3) $w \neq v^{(l)}$ $(l = 1, \ldots, m)$.

Proposition 6 *For each oracle A, $(r+1)SEGMENT^A \leq_T^P rTAUT^A$.*

Proof. Suppose that i is a positive integer, H is a query free formula and $|u| \leq |v^{(1)}| = \cdots = |v^{(m)}| = i$. Then, the sequence $\langle u, i, H, \langle v^{(1)}, \ldots, v^{(m)}\rangle\rangle$ belongs to $(r+1)SEGMENT^A$ if and only if the following formula does not belong to $rTAUT^A$.

$$\sharp\langle r, i, [(\bigwedge_{k=1}^{|u|}(u_k \Leftrightarrow q_k^{(r+1)})) \wedge (\bigwedge_{l=1}^{m} \neg \bigwedge_{k=1}^{i}(v_k^{(l)} \Leftrightarrow q_k^{(r+1)}))] \Rightarrow H\rangle.$$

□

Lemma 7 *There exists a deterministic oracle Turing machine $M_{r+1}[X]$ such that* (1) *and* (2) *below hold.*

(1) For every oracle A, $\mathrm{Lang}(M_{r+1}[rTAUT^A]) = (r+1)TAUT^A$*.*

(2) If A is an $(r+1)$-generic oracle in the sense of Dowd, then $M_{r+1}[rTAUT^A]$ accepts the members of $(r+1)TAUT^A$ in polynomial time.

Proof. As in our proof of Lemma 3, we construct a deterministic oracle transducer $T_{r+1}[X]$ that works as follows. Running on input $F = \natural\langle r+1, i, H\rangle$, $T_{r+1}[rTAUT^A]$ outputs the list of all members of the set B as in (4.1). Further, if A is $(r+1)$-generic in the sense of Dowd, then $T_{r+1}[rTAUT^A]$ terminates within polynomial steps of the length of input F, where such a polynomial depends upon A. Such a construction is possible by Proposition 5 and 6. By using the transducer $T_{r+1}[X]$, again as in our proof of Lemma 3, we construct a deterministic oracle machine $M_{r+1}[X]$ as follows. When $M_{r+1}[rTAUT^A]$ runs on input $F = \natural\langle r+1, i, H\rangle$, it checks whether $\sharp\langle r, i, G \Rightarrow H\rangle$ belongs to $rTAUT^A$, where G is the formula given in (iii) of Proposition 4. By the fact that A is polynomial time Turing reducible to $rTAUT^A$, we can construct such a machine $M_{r+1}[X]$. Thus by Proposition 4 and by our construction of $T_{r+1}[X]$, $M_{r+1}[X]$ satisfies the requirements of the lemma. □

Proof of Theorem 1. Suppose that r is a positive integer and A is an r-generic oracle in Dowd's sense. By Lemma 7, for any positive integer $s < r$, $(s+1)TAUT^A$ is polynomial time Turing equivalent to $sTAUT^A$, and hence we have $rTAUT^A \equiv^P_T 1TAUT^A$. Thus, by Theorem 1 for $r = 1$, we have Theorem 1 for all $r \geq 1$. □

5 Conclusion

For each positive integer r, the following two statements are equivalent (Theorem 2).

(1) If A is a random oracle then $rTAUT^A \notin P^A$ with probability 1.

(2) The unrelativized class R does not equal NP.

Proof of Theorem 2. Consider the following five subsets of the Cantor space;

$$\begin{aligned}
D_1 &= \{X : TAUT \leq^P_T X\},\\
D_2 &= \{X : X \text{ is } r\text{-generic in the sense of Dowd}\},\\
D_3 &= \{X : rTAUT^X \equiv^P_T TAUT \oplus X\},\\
D_4 &= \{X : rTAUT^X \leq^P_T X\},\\
D_5 &= \{X : rTAUT^X \notin P^X\}.
\end{aligned}$$

First, $R = NP$ if and only if $NP \subseteq BPP$ (by Ko [8]). The latter is equivalent to the assertion $coNP \subseteq BPP$. Next, $TAUT \in BPP$ if and only if $\mu(D_1) = 1$ (by Bennet and Gill [4], see also [1]), where μ means Lebesgue measure. Now, $\mu(D_2) = 1$ (see section 4 of Dowd [6]). Therefore, by Theorem 1, $\mu(D_3) = 1$. Hence $\mu(D_1) = 1$ if and only if $\mu(D_4) = 1$. Moreover, the set D_4 is closed under finite changes i.e. if $A \in D_4$ and $\{x : A(x) \neq B(x)\}$ is a finite set then $B \in D_4$. Consequently, if $\mu(D_4) > 0$ then $\mu(D_4) = 1$. Hence D_5 has Lebesgue measure 1 if and only if $R \neq NP$. □

Acknowledgement

I would like to thank Professor H. Tanaka for his encouragement and useful discussions. I would also like to thank the referee for a list of corrections.

This research was partially supported by Grant-in-Aid for Scientific Research (No. 09440072, 09440078), Ministry of Education, Science, Sports and Culture of Japan.

References

[1] K. Ambos-Spies, Randomness, relativizations, and polynomial reducibilities, in: *Structure in Complexity Theory*, Lecture Notes in Computer Science **223** (Springer, Berlin, 1986) 23-34.

[2] T. Baker, J. Gill and R. Solovay, Relativizations of $P =?NP$ question, *SIAM J. Comput.* **4** (1975) 431-442.

[3] J. L. Balcázar, J. Díaz and J. Gabarró, Structural complexity I, II (Springer, Berlin, 1988, 1990).

[4] C. Bennett and J. Gill, Relative to a random oracle A, $P^A \neq NP^A \neq$ *co-*NP^A with probability 1, *SIAM J. Comput.* **10** (1981) 96-113.

[5] M. Dowd, Forcing and the P-hierarchy, *Rutgers University Laboratory for Computer Science Research Technical Report* **LCSR-TR-35** (1982).

[6] M. Dowd, Generic oracles, uniform machines, and codes, *Information and Computation* **96** (1992) 65-76.

[7] S. Feferman, Some applications of the notions of forcing and generic sets, *Fund. Math.* **56** (1965), 325-345.

[8] Ker-I. Ko, Some observations on the probabilistic algorithms and NP-hard problems, *Inform. Process. Lett.* **14** (1982) 39-43.

[9] K. Kunen, Set theory, North-Holland, Amsterdam, 1980.

[10] H. Rogers, Jr., Theory of recursive functions and effective computability, (MacGraw-Hill, New York, 1967).

[11] H. Tanaka and M. Kudoh, On relativized probabilistic polynomial time algorithms, *J. Math. Soc. Japan* **49** (1997), 15-30.