Chaitin Ω Numbers and Halting Problems

Kohtaro Tadaki

Research and Development Initiative, Chuo University
CREST, JST
1–13–27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan
E-mail: tadaki@kc.chuo-u.ac.jp

Abstract. Chaitin [G. J. Chaitin, *J. Assoc. Comput. Mach.*, vol. 22, pp. 329–340, 1975] introduced his Ω number as a concrete example of random real. The real Ω is defined as the probability that an optimal computer halts, where the optimal computer is a universal decoding algorithm used to define the notion of program-size complexity. Chaitin showed Ω to be random by discovering the property that the first n bits of the base-two expansion of Ω solve the halting problem of the optimal computer for all binary inputs of length at most n. In the present paper we investigate this property from various aspects. It is known that the base-two expansion of Ω and the halting problem are Turing equivalent. We consider elaborations of both the Turing reductions which constitute the Turing equivalence. These elaborations can be seen as a variant of the weak truth-table reduction, where a computable bound on the use function is explicitly specified. We thus consider the relative computational power between the base-two expansion of Ω and the halting problem by imposing the restriction to finite size on both the problems.

Key words: algorithmic information theory, Chaitin Ω number, halting problem, Turing reduction, algorithmic randomness, program-size complexity

1 Introduction

Algorithmic information theory (AIT, for short) is a framework for applying information-theoretic and probabilistic ideas to recursive function theory. One of the primary concepts of AIT is the *program-size complexity* (or *Kolmogorov complexity*) $H(s)$ of a finite binary string s, which is defined as the length of the shortest binary input for a universal decoding algorithm U, called an *optimal computer*, to output s. By the definition, $H(s)$ can be thought of as the information content of the individual finite binary string s. In fact, AIT has precisely the formal properties of classical information theory (see Chaitin [2]). In particular, the notion of program-size complexity plays a crucial role in characterizing the *randomness* of an infinite binary string, or equivalently, a real. In [2] Chaitin introduced the halting probability Ω_U as an example of random real. His Ω_U is defined as the probability that the optimal computer U halts, and plays a central role in the metamathematical development of AIT. The real Ω_U is shown to be random, based on the following fact:

Fact 1 (Chaitin [2]) *The first n bits of the base-two expansion of Ω_U solve the halting problem of U for inputs of length at most n.* $\Box$

In this paper, we first consider the following converse problem:

Problem 1 *For every positive integer n, if n and the list of all halting inputs for U of length at most n are given, can the first n bits of the base-two expansion of Ω_U be calculated ?* $\Box$

As a result of this paper, we can answer this problem negatively. In this paper, however, we consider more general problems in the following forms. Let V and W be arbitrary optimal computers.

Problem 2 *Find a succinct equivalent characterization of a total recursive function $f\colon \mathbb{N}^+ \to \mathbb{N}$ which satisfies the condition: For all $n \in \mathbb{N}^+$, if n and the list of all halting inputs for V of length at most n are given, then the first $n-f(n)-O(1)$ bits of the base-two expansion of Ω_W can be calculated.* $\Box$

Problem 3 *Find a succinct equivalent characterization of a total recursive function $f\colon \mathbb{N}^+ \to \mathbb{N}$ which satisfies the condition: For infinitely many $n \in \mathbb{N}^+$, if n and the list of all halting inputs for V of length at most n are given, then the first $n-f(n)-O(1)$ bits of the base-two expansion of Ω_W can be calculated.* $\Box$

Here $\mathbb{N}^+$ denotes the set of positive integers and $\mathbb{N} = \{0\} \cup \mathbb{N}^+$. Theorem 4 and Theorem 10 below are two of the main results of this paper. On the one hand, Theorem 4 gives to Problem 2 a solution that the total recursive function f must satisfy $\sum_{n=1}^{\infty} 2^{-f(n)} < \infty$, which is the Kraft inequality in essence. Note that the condition $\sum_{n=1}^{\infty} 2^{-f(n)} < \infty$ holds for $f(n) = \lfloor (1+\varepsilon)\log_2 n \rfloor$ with an arbitrary computable real $\varepsilon > 0$, while this condition does not hold for $f(n) = \lfloor \log_2 n \rfloor$. On the other hand, Theorem 10 gives to Problem 3 a solution that the total recursive function f must not be bounded to the above. Theorem 10 also results in Corollary 2 below, which refutes Problem 1 completely.

It is also important to consider whether the bound n on the length of halting inputs given in Fact 1 is tight or not. We consider this problem in the following form:

Problem 4 *Find a succinct equivalent characterization of a total recursive function $f\colon \mathbb{N}^+ \to \mathbb{N}$ which satisfies the condition: For all $n \in \mathbb{N}^+$, if n and the first n bits of the base-two expansion of Ω_V are given, then the list of all halting inputs for W of length at most $n + f(n) - O(1)$ can be calculated.* $\Box$

Theorem 11, which is one of the main results of this paper, gives to Problem 4 a solution that the total recursive function f must be bounded to the above. Thus, we see that the bound n on the length of halting inputs given in Fact 1 is tight up to an additive constant.

It is well known that the base-two expansion of Ω_U and the halting problem of U are Turing equivalent, i.e., $\Omega_U \equiv_T \operatorname{dom} U$ holds, where $\operatorname{dom} U$ denotes the domain of definition of U. This paper investigates an elaboration of the Turing equivalence. For example, consider the Turing reduction $\Omega_U \le_T \operatorname{dom} U$, which partly constitutes the Turing equivalence $\Omega_U \equiv_T \operatorname{dom} U$. The Turing reduction can be equivalent to the condition that there exists an oracle deterministic Turing machine M such that, for all $n \in \mathbb{N}^+$,

$$M^{\operatorname{dom} U}(n) = \Omega_U\!\restriction_n, \tag{1}$$

where $\Omega_U \restriction_n$ denotes the first n bits of the base-two expansion of Ω_U. Let $g\colon \mathbb{N}^+ \to \mathbb{N}$ and $h\colon \mathbb{N}^+ \to \mathbb{N}$ be total recursive functions. Then the condition (1) can be elaborated to the condition that there exists an oracle deterministic Turing machine M such that, for all $n \in \mathbb{N}^+$,

$$M^{\operatorname{dom} U\restriction_{g(n)}}(n) = \Omega_U\!\restriction_{h(n)}, \tag{2}$$

where $\operatorname{dom} U \restriction_{g(n)}$ denotes the set of all strings in $\operatorname{dom} U$ of length at most $g(n)$. This elaboration allows us to consider the asymptotic behavior of h which satisfies the condition (2), for a given g. We might regard g as the degree of the relaxation of the restrictions on the computational resource (i.e., on the oracle $\operatorname{dom} U$) and h as the difficulty of the problem to solve. Thus, even in the context of computability theory, we can deal with the notion of asymptotic behavior in a manner like in computational complexity theory in some sense. Note also that the condition (2) can be seen as a variant of the weak truth-table reduction of the function $\Omega_U\!\restriction_{h(n)}$ of n to $\operatorname{dom} U$, where a computable bound on the use of $M^{\operatorname{dom} U}(n)$ is explicitly specified by the function g. Theorem 4, a solution to Problem 2, is obtained as a result of the investigation in this line, and gives the upper bound of the function h in the case of $g(n) = n$.

The other Turing reduction $\operatorname{dom} U \le_T \Omega_U$, which constitutes $\Omega_U \equiv_T \operatorname{dom} U$, is also elaborated in the same manner as above to lead to Theorem 11, a solution to Problem 4.

Thus, in this paper, we study the relationship between the base-two expansion of Ω and the halting problem of an optimal computer using a more rigorous and insightful notion than the notion of Turing equivalence. The paper is organized as follows. We begin in Section 2 with some preliminaries to AIT. We then present Theorems 4, 10, and 11 in Sections 3, 4, and 5, respectively. Due to the 10-page limit, we omit some proofs, in particular, the proof of Theorem 10. A full paper which describes all the proofs and other related results is in preparation.

2 Preliminaries

We start with some notation about numbers and strings which will be used in this paper. $\#S$ is the cardinality of S for any set S. $\mathbb{N} = \{0, 1, 2, 3, \dots\}$ is the set of natural numbers, and $\mathbb{N}^+$ is the set of positive integers. $\mathbb{Z}$ is the set of integers, and $\mathbb{Q}$ is the set of rationals. $\mathbb{R}$ is the set of reals. Normally, $O(1)$ denotes any function $f\colon \mathbb{N}^+ \to \mathbb{R}$ such that there is $C \in \mathbb{R}$ with the property that $|f(n)| \le C$ for all $n \in \mathbb{N}^+$.

$\{0,1\}^* = \{\lambda, 0, 1, 00, 01, 10, 11, 000, \dots\}$ is the set of finite binary strings where λ denotes the *empty string*, and $\{0,1\}^*$ is ordered as indicated. We identify any string in $\{0,1\}^*$ with a natural number in this order, i.e., we consider $\varphi\colon \{0,1\}^* \to \mathbb{N}$ such that $\varphi(s) = 1s - 1$ where the concatenation $1s$ of strings 1 and s is regarded as a dyadic integer, and then we identify s with $\varphi(s)$. For any $s \in \{0,1\}^*$, $|s|$ is the *length* of s. A subset S of $\{0,1\}^*$ is called *prefix-free* if no string in S is a prefix of another string in S. For any subset S of $\{0,1\}^*$ and any $n \in \mathbb{Z}$, we denote by $S{\restriction_n}$ the set $\{s \in S \mid |s| \le n\}$. Note that $S{\restriction_n} = \emptyset$ for every subset S of $\{0,1\}^*$ and every negative integer $n \in \mathbb{Z}$. For any partial function f, the domain of definition of f is denoted by $\operatorname{dom} f$. We write "r.e." instead of "recursively enumerable."

Let α be an arbitrary real. For any $n \in \mathbb{N}^+$, we denote by $\alpha{\restriction_n} \in \{0,1\}^*$ the first n bits of the base-two expansion of $\alpha - \lfloor \alpha \rfloor$ with infinitely many zeros, where $\lfloor \alpha \rfloor$ is the greatest integer less than or equal to α. For example, in the case of $\alpha = 5/8$, $\alpha{\restriction_6} = 101000$. On the other hand, for any non-positive integer $n \in \mathbb{Z}$, we set $\alpha{\restriction_n} = \lambda$. A real α is called *r.e.* if there exists a total recursive function $f\colon \mathbb{N}^+ \to \mathbb{Q}$ such that $f(n) \le \alpha$ for all $n \in \mathbb{N}^+$ and $\lim_{n\to\infty} f(n) = \alpha$. An r.e. real is also called a *left-computable* real.

2.1 Algorithmic information theory

In the following we concisely review some definitions and results of algorithmic information theory [2, 3]. A *computer* is a partial recursive function $C\colon \{0,1\}^* \to \{0,1\}^*$ such that $\operatorname{dom} C$ is a prefix-free set. For each computer C and each $s \in \{0,1\}^*$, $H_C(s)$ is defined by $H_C(s) = \min\left\{\, |p| \;\middle|\; p \in \{0,1\}^* \,\&\, C(p) = s \,\right\}$ (may be ∞). A computer U is said to be *optimal* if for each computer C there exists $d \in \mathbb{N}$ with the following property; if $p \in \operatorname{dom} C$, then there is $q \in \operatorname{dom} U$ for which $U(q) = C(p)$ and $|q| \le |p| + d$. It is easy to see that there exists an optimal computer. We choose a particular optimal computer U as the standard one for use, and define $H(s)$ as $H_U(s)$, which is referred to as the *program-size complexity* of s or the *Kolmogorov complexity* of s. It follows that for every computer C there exists $d \in \mathbb{N}$ such that, for every $s \in \{0,1\}^*$,

$$H(s) \le H_C(s) + d. \tag{3}$$

Based on this we can show that, for every partial recursive function $\Psi\colon \{0,1\}^* \to \{0,1\}^*$, there exists $d \in \mathbb{N}$ such that, for every $s \in \operatorname{dom} \Psi$,

$$H(\Psi(s)) \le H(s) + d. \tag{4}$$

For any $s \in \{0,1\}^*$, we define s^* as $\min\{\, p \in \{0,1\}^* \mid U(p) = s\}$, i.e., the first element in the ordered set $\{0,1\}^*$ of all strings p such that $U(p) = s$. Then, $|s^*| = H(s)$ for every $s \in \{0,1\}^*$. For any $s, t \in \{0,1\}^*$, we define $H(s,t)$ as $H(b(s,t))$, where $b\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ is a particular bijective total recursive function. Note also that, for every $n \in \mathbb{N}$, $H(n)$ is $H(\text{the } n\text{th element of } \{0,1\}^*)$.

Definition 1 (Chaitin Ω number, Chaitin [2]). *For any optimal computer V, the halting probability Ω_V of V is defined by*

$$\Omega_V = \sum_{p \in \operatorname{dom} V} 2^{-|p|}.$$

□

For every optimal computer V, since $\operatorname{dom} V$ is prefix-free, Ω_V converges and $0 < \Omega_V \leq 1$. For any $\alpha \in \mathbb{R}$, we say that α is *weakly Chaitin random* if there exists $c \in \mathbb{N}$ such that $n - c \leq H(\alpha\restriction_n)$ for all $n \in \mathbb{N}^+$ [2,3]. Based on Fact 1, Chaitin [2] showed that Ω_V is weakly Chaitin random for every optimal computer V. Therefore $0 < \Omega_V < 1$ for every optimal computer V. For any $\alpha \in \mathbb{R}$, we say that α is *Chaitin random* if $\lim_{n\to\infty} H(\alpha\restriction_n) - n = \infty$ [3]. We can then show the following theorem (see Chaitin [3] for the proof and historical detail).

Theorem 1. *For every $\alpha \in \mathbb{R}$, α is weakly Chaitin random if and only if α is Chaitin random.* □

Miller and Yu [7] recently strengthened Theorem 1 to the following form.

Theorem 2 (Ample Excess Lemma, Miller and Yu [7]). *For every $\alpha \in \mathbb{R}$, α is weakly Chaitin random if and only if $\sum_{n=1}^{\infty} 2^{n-H(\alpha\restriction_n)} < \infty$.* □

The following is an important result on random r.e. reals.

Theorem 3 (Calude, et al. [1], Kučera and Slaman [6]). *For every $\alpha \in (0,1)$, α is r.e. and weakly Chaitin random if and only if there exists an optimal computer V such that $\alpha = \Omega_V$.* □

3 Elaboration I of the Turing reduction $\Omega_U \leq_T \operatorname{dom} U$

Theorem 4 (main result I). *Let V and W be optimal computers, and let $f\colon \mathbb{N}^+ \to \mathbb{N}$ be a total recursive function. Then the following two conditions are equivalent:*

(i) There exist an oracle deterministic Turing machine M and $c \in \mathbb{N}$ such that, for all $n \in \mathbb{N}^+$, $M^{\operatorname{dom} V\restriction_n}(n) = \Omega_W\restriction_{n-f(n)-c}$.
(ii) $\sum_{n=1}^{\infty} 2^{-f(n)} < \infty$. □

Theorem 4 follows from Theorem 5 and Theorem 6 below, and Theorem 3.

Theorem 5. *Let α be an r.e. real, and let V be an optimal computer. For every total recursive function $f\colon \mathbb{N}^+ \to \mathbb{N}$, if $\sum_{n=1}^{\infty} 2^{-f(n)} < \infty$, then there exist an oracle deterministic Turing machine M and $c \in \mathbb{N}$ such that, for all $n \in \mathbb{N}^+$, $M^{\operatorname{dom} V\restriction_n}(n) = \alpha\restriction_{n-f(n)-c}$.* □

Theorem 6. *Let α be a real which is weakly Chaitin random, and let V be an optimal computer. For every total recursive function $f\colon \mathbb{N}^+ \to \mathbb{N}$, if there exists an oracle deterministic Turing machine M such that, for all $n \in \mathbb{N}^+$, $M^{\operatorname{dom} V\restriction_n}(n) = \alpha\restriction_{n-f(n)}$, then $\sum_{n=1}^{\infty} 2^{-f(n)} < \infty$.* □

The proofs of Theorem 5 and Theorem 6 are given in the next two subsections, respectively.

3.1 The proof of Theorem 5

In order to prove Theorem 5, we need Theorem 7 and Corollary 1 below.

Theorem 7 (Kraft-Chaitin Theorem, Chaitin [2]). *Let $f\colon \mathbb{N}^+ \to \mathbb{N}$ be a total recursive function such that $\sum_{n=1}^{\infty} 2^{-f(n)} \le 1$. Then there exists a total recursive function $g\colon \mathbb{N}^+ \to \{0,1\}^*$ such that (i) g is an injection, (ii) the set $\{\, g(n) \mid n \in \mathbb{N}^+ \}$ is prefix-free, and (iii) $|g(n)| = f(n)$ for all $n \in \mathbb{N}^+$.* □

Let M be a deterministic Turing machine with the input and output alphabet $\{0,1\}$, and let C be a computer. We say that M *computes* C if the following holds: for every $p \in \{0,1\}^*$, when M starts with the input p, (i) M halts and outputs $C(p)$ if $p \in \operatorname{dom} C$; (ii) M does not halt forever otherwise. We use this convention on the computation of a computer by a deterministic Turing machine throughout the rest of this paper. Thus, we exclude the possibility that there is $p \in \{0,1\}^*$ such that, when M starts with the input p, M halts but $p \notin \operatorname{dom} C$. For any $p \in \{0,1\}^*$, we denote the running time of M on the input p by $T_M(p)$ (may be ∞). Thus, $T_M(p) \in \mathbb{N}$ for every $p \in \operatorname{dom} C$ if M computes C.

Theorem 8. *Let V be an optimal computer. Then, for every computer C there exists $d \in \mathbb{N}$ such that, for every $p \in \{0,1\}^*$, if p and the list of all halting inputs for V of length at most $|p| + d$ are given, then the halting problem of the input p for C can be solved.*

Proof. Let M be a deterministic Turing machine which computes a computer C. We consider the computer D such that (i) $\operatorname{dom} D = \operatorname{dom} C$ and (ii) $D(p) = T_M(p)$ for every $p \in \operatorname{dom} C$. Recall here that we identify $\{0,1\}^*$ with $\mathbb{N}$. It is easy to see that such a computer D exists. Then, since V is an optimal computer, from the definition of optimality there exists $d \in \mathbb{N}$ with the following property; if $p \in \operatorname{dom} D$, then there is $q \in \operatorname{dom} V$ for which $V(q) = D(p)$ and $|q| \le |p| + d$.

Given $p \in \{0,1\}^*$ and the list $\{q_1, \dots, q_L\}$ of all halting inputs for V of length at most $|p| + d$, one first calculates the finite set $S = \{\, V(q_i) \mid i = 1, \dots, L \,\}$, and then calculates $T_{\max} = \max S$ where S is regarded as a subset of $\mathbb{N}$. One then simulates the computation of M with the input p until at most the time step $T_{\max}$. In the simulation, if M halts until at most the time step $T_{\max}$, one knows that $p \in \operatorname{dom} C$. On the other hand, note that if $p \in \operatorname{dom} C$ then there is $q \in \operatorname{dom} V$ such that $V(q) = T_M(p)$ and $|q| \le |p| + d$, and therefore $q \in \{q_1, \dots, q_L\}$ and $T_M(p) \le T_{\max}$. Thus, in the simulation, if M does not yet halt at the time step $T_{\max}$, one knows that M does not halt forever and therefore $p \notin \operatorname{dom} C$. □

As a corollary of Theorem 8 we obtain the following.

Corollary 1. *Let V be an optimal computer. Then, for every computer C there exist an oracle deterministic Turing machine M and $d \in \mathbb{N}$ such that, for all $n \in \mathbb{N}^+$, $M^{\operatorname{dom} V \restriction_{n+d}}(n) = \operatorname{dom} C\restriction_n$, where the finite subset $\operatorname{dom} C\restriction_n$ of $\{0,1\}^*$ is represented as a finite binary string in a certain format.* □

Based on Theorem 7 and Corollary 1, Theorem 5 is proved as follows.

Proof (of Theorem 5). Let α be an r.e. real, and let V be an optimal computer. For an arbitrary total recursive function $f\colon \mathbb{N}^+ \to \mathbb{N}$, assume that $\sum_{n=1}^{\infty} 2^{-f(n)} < \infty$. In the case of $\alpha \in \mathbb{Q}$, the result is obvious. Thus, in what follows, we assume that $\alpha \notin \mathbb{Q}$ and therefore the base-two expansion of $\alpha - \lfloor \alpha \rfloor$ is unique and contains infinitely many ones.

Since $\sum_{n=1}^{\infty} 2^{-f(n)} < \infty$, there exists $d_0 \in \mathbb{N}$ such that $\sum_{n=1}^{\infty} 2^{-f(n)-d_0} \leq 1$. Hence, by the Kraft-Chaitin Theorem, i.e., Theorem 7, there exists a total recursive function $g\colon \mathbb{N}^+ \to \{0,1\}^*$ such that (i) the function g is an injection, (ii) the set $\{\, g(n) \mid n \in \mathbb{N}^+ \}$ is prefix-free, and (iii) $|g(n)| = f(n) + d_0$ for all $n \in \mathbb{N}^+$. On the other hand, since α is r.e., there exists a total recursive function $h\colon \mathbb{N}^+ \to \mathbb{Q}$ such that $h(k) \leq \alpha$ for all $k \in \mathbb{N}^+$ and $\lim_{k\to\infty} h(k) = \alpha$.

Now, let us consider the following computer C. For each $n \in \mathbb{N}^+$, $p, s \in \{0,1\}^*$ and $l \in \mathbb{N}$ such that $U(p) = l$, $g(n)ps \in \operatorname{dom} C$ if and only if (i) $|g(n)ps| = n - l$ and (ii) $0.s < h(k) - \lfloor \alpha \rfloor$ for some $k \in \mathbb{N}^+$. It is easy to see that such a computer C exists. Then, by Corollary 1, there exist an oracle deterministic Turing machine M and $d \in \mathbb{N}$ such that, for all $n \in \mathbb{N}^+$, $M^{\operatorname{dom} V\restriction_{n+d}}(n) = \operatorname{dom} C\restriction_n$, where the finite subset $\operatorname{dom} C\restriction_n$ of $\{0,1\}^*$ is represented as a finite binary string in a certain format. We then see that, for every $n \in \mathbb{N}^+$ and $s \in \{0,1\}^*$ such that $|s| = n - |g(n)| - d - |d^*|$,

$$g(n)d^*s \in \operatorname{dom} C \text{ if and only if } s \leq \alpha\restriction_{n-|g(n)|-d-|d^*|}, \tag{5}$$

where s and $\alpha\restriction_{n-|g(n)|-d-|d^*|}$ are regarded as a dyadic integer. Then, by the following procedure, we see that there exist an oracle deterministic Turing machine M_1 and $c \in \mathbb{N}$ such that, for all $n \in \mathbb{N}^+$, $M_1^{\operatorname{dom} V\restriction_n}(n) = \alpha\restriction_{n-f(n)-c}$. Note here that $|g(n)| = f(n) + d_0$ for all $n \in \mathbb{N}^+$ and also $H(d) = |d^*|$.

Given n and $\operatorname{dom} V\restriction_n$ with $n > d$, one first checks whether $n - |g(n)| - d - H(d) \leq 0$ holds. If this holds then one outputs λ. If this does not hold, one then calculates the finite set $\operatorname{dom} C\restriction_{n-d}$ by simulating the computation of M with the input $n - d$ and the oracle $\operatorname{dom} V\restriction_n$. Then, based on (5), one determines $\alpha\restriction_{n-|g(n)|-d-H(d)}$ by checking whether $g(n)d^*s \in \operatorname{dom} C$ holds or not for each $s \in \{0,1\}^*$ with $|s| = n - |g(n)| - d - H(d)$. This is possible since $|g(n)d^*s| = n - d$ for every $s \in \{0,1\}^*$ with $|s| = n - |g(n)| - d - H(d)$. Finally, one outputs $\alpha\restriction_{n-|g(n)|-d-H(d)}$. □

3.2 The proof of Theorem 6

In order to prove Theorem 6, we need Theorem 9 below and the Ample Excess Lemma (i.e., Theorem 2).

Let M be an arbitrary deterministic Turing machine with the input and output alphabet $\{0,1\}$. We define $L_M = \min\{\, |p| \mid p \in \{0,1\}^* \ \&\ M \text{ halts on input } p\}$ (may be ∞). For any $n \geq L_M$, we define I_M^n as the set of all halting inputs p for M with $|p| \leq n$ which take longest to halt in the computation of M, i.e., as the set $\{\, p \in \{0,1\}^* \mid |p| \leq n \ \&\ T_M(p) = T_M^n \}$ where T_M^n is the maximum running time of M on all halting inputs of length at most n. We can slightly strengthen the result presented in Chaitin [3] to obtain the following (see Note in Section 8.1 of Chaitin [3]).

Theorem 9. *Let V be an optimal computer, and let M be a deterministic Turing machine which computes V. Then $n = H(n,p) + O(1) = H(p) + O(1)$ for all (n,p) with $n \geq L_M$ and $p \in I_M^n$.* □

Proof (of Theorem 6). Let α be a real which is weakly Chaitin random. Let V be an optimal computer, and let M be a deterministic Turing machine which computes V. For each $n \geq L_M$, we choose a particular p_n from I_M^n. For an arbitrary total recursive function $f\colon \mathbb{N}^+ \to \mathbb{N}$, assume that there exists an oracle deterministic Turing machine M_0 such that, for all $n \in \mathbb{N}^+$, $M_0^{\operatorname{dom} V\restriction_n}(n) = \alpha\restriction_{n-f(n)}$. Note that, given (n, p_n) with $n \geq L_M$, one can calculate the finite set $\operatorname{dom} V\restriction_n$ by simulating the computation of M with the input q until at most the time step $T_M(p_n)$, for each $q \in \{0,1\}^*$ with $|q| \leq n$. This can be possible because $T_M(p_n) = T_M^n$ for every $n \geq L_M$. Thus, we see that there exists a partial recursive function $\Psi\colon \mathbb{N} \times \{0,1\}^* \to \{0,1\}^*$ such that, for all $n \geq L_M$, $\Psi(n, p_n) = \alpha\restriction_{n-f(n)}$. It follows from (4) that $H(\alpha\restriction_{n-f(n)}) \leq H(n, p_n) + O(1)$ for all $n \geq L_M$. Thus, by Theorem 9 we have

$$H(\alpha\restriction_{n-f(n)}) \leq n + O(1) \tag{6}$$

for all $n \in \mathbb{N}^+$.

In the case where the function $n - f(n)$ of n is bounded to the above, there exists $c \in \mathbb{N}$ such that, for every $n \in \mathbb{N}^+$, $-f(n) \leq c - n$, and therefore $\sum_{n=1}^{\infty} 2^{-f(n)} \leq 2^c$. Thus, in what follows, we assume that the function $n - f(n)$ of n is not bounded to the above.

We define a function $g\colon \mathbb{N}^+ \to \mathbb{Z}$ by $g(n) = \max\{k - f(k) \mid 1 \leq k \leq n\}$. It follows that the function g is non-decreasing and $\lim_{n\to\infty} g(n) = \infty$. Thus we can choose an enumeration $n_1, n_2, n_3, \ldots$ of the countably infinite set $\{n \in \mathbb{N}^+ \mid n \geq 2 \ \&\ 0 \leq g(n-1) < g(n)\}$ with $n_j < n_{j+1}$. It is then easy to see that $g(n_j) = n_j - f(n_j)$ and $1 \leq n_j - f(n_j) < n_{j+1} - f(n_{j+1})$ hold for all j. On the other hand, since α is weakly Chaitin random, using the Ample Excess Lemma, i.e., Theorem 2, we have $\sum_{n=1}^{\infty} 2^{n - H(\alpha\restriction_n)} < \infty$. Thus, using (6) we see that

$$\sum_{j=1}^{\infty} 2^{-f(n_j)} \leq \sum_{j=1}^{\infty} 2^{n_j - f(n_j) - H(\alpha\restriction_{n_j - f(n_j)}) + O(1)} \leq \sum_{n=1}^{\infty} 2^{n - H(\alpha\restriction_n) + O(1)} < \infty. \tag{7}$$

On the other hand, it is easy to see that (i) $g(n) \geq n - f(n)$ for every $n \in \mathbb{N}^+$, and (ii) $g(n) = g(n_j)$ for every j and n with $n_j \leq n < n_{j+1}$. Thus, for each $k \geq 2$, it is shown that

$$\begin{aligned}\sum_{n=n_1}^{n_k-1} 2^{-f(n)} &\leq \sum_{n=n_1}^{n_k-1} 2^{g(n)-n} = \sum_{j=1}^{k-1} \sum_{n=n_j}^{n_{j+1}-1} 2^{g(n)-n} = \sum_{j=1}^{k-1} 2^{g(n_j)} \sum_{n=n_j}^{n_{j+1}-1} 2^{-n} \\ &= \sum_{j=1}^{k-1} 2^{n_j - f(n_j)} 2^{-n_j+1} \left(1 - 2^{-n_{j+1}+n_j}\right) < 2 \sum_{j=1}^{k-1} 2^{-f(n_j)}.\end{aligned}$$

Thus, using (7) we see that $\lim_{k\to\infty} \sum_{n=n_1}^{n_k-1} 2^{-f(n)} < \infty$. Since $2^{-f(n)} > 0$ for all $n \in \mathbb{N}^+$ and $\lim_{j\to\infty} n_j = \infty$, we have $\sum_{n=1}^{\infty} 2^{-f(n)} < \infty$. □

4 Elaboration II of the Turing reduction $\Omega_U \leq_T \operatorname{dom} U$

Theorem 10 (main result II). *Let V and W be optimal computers, and let $f\colon \mathbb{N}^+ \to \mathbb{N}$ be a total recursive function. Then the following two conditions are equivalent:*

(i) There exist an oracle deterministic Turing machine M and $c \in \mathbb{N}$ such that, for infinitely many $n \in \mathbb{N}^+$, $M^{\operatorname{dom} V\restriction_n}(n) = \Omega_W\restriction_{n-f(n)-c}$.
(ii) The function f is not bounded to the above. □

In a similar manner to the proof of Theorem 4 we can prove Theorem 10. The implication (ii) ⇒ (i) of Theorem 10 follows from Lemma 1 below and Corollary 1. On the other hand, the implication (i) ⇒ (ii) of Theorem 10 follows from Theorem 9 and Theorem 1. By setting $f(n) = 0$ and $W = V$ in Theorem 10, we obtain the following.

Corollary 2. *Let V be an optimal computer. Then, for every $c \in \mathbb{N}$, there does not exist an oracle deterministic Turing machine M such that, for infinitely many $n \in \mathbb{N}^+$, $M^{\operatorname{dom} V\restriction_{n+c}}(n) = \Omega_V\restriction_n$.* □

5 Elaboration of the Turing reduction $\operatorname{dom} U \leq_T \Omega_U$

Theorem 11 (main result III). *Let V and W be optimal computers, and let $f\colon \mathbb{N}^+ \to \mathbb{N}$ be a total recursive function. Then the following two conditions are equivalent:*

(i) There exist an oracle deterministic Turing machine M and $c \in \mathbb{N}$ such that, for all $n \in \mathbb{N}^+$, $M^{\{\Omega_V\restriction_n\}}(n) = \operatorname{dom} W\restriction_{n+f(n)-c}$, where the finite subset $\operatorname{dom} W\restriction_{n+f(n)-c}$ of $\{0,1\}^*$ is represented as a finite binary string in a certain format.
(ii) The function f is bounded to the above. □

The implication (ii) ⇒ (i) of Theorem 11 follows immediately from Fact 1 and Corollary 1. On the other hand, in order to prove the implication (i) ⇒ (ii) of Theorem 11, we need the following lemma first.

Lemma 1. *Let $f\colon \mathbb{N}^+ \to \mathbb{N}$ be a total recursive function. If the function f is not bounded to the above, then the function $f(n) - H(n)$ of $n \in \mathbb{N}^+$ is not bounded to the above.*

Proof. Contrarily, assume that there exists $c \in \mathbb{N}$ such that, for every $n \in \mathbb{N}^+$, $f(n) \leq H(n) + c$. Then, since f is not bounded to the above, it is easy to see that there exists a total recursive function $\Psi\colon \mathbb{N}^+ \to \mathbb{N}^+$ such that, for every $k \in \mathbb{N}^+$, $k \leq H(\Psi(k))$. It follows from (4) that $k \leq H(k) + O(1)$ for all $k \in \mathbb{N}^+$. On the other hand, using (3) we can show that $H(k) \leq 2\log_2 k + O(1)$ for all $k \in \mathbb{N}^+$. Thus we have $k \leq 2\log_2 k + O(1)$ for all $k \in \mathbb{N}^+$. However, we have a contradiction on letting $k \to \infty$ in this inequality, and the result follows. □

Proof (of (i) ⇒ (ii) of Theorem 11). Let V and W be optimal computers. For an arbitrary total recursive function $f\colon \mathbb{N}^+ \to \mathbb{N}$, assume that there exist an oracle deterministic Turing machine M and $c \in \mathbb{N}$ such that, for all $n \in \mathbb{N}^+$, $M^{\{\Omega_V\restriction_n\}}(n) = \operatorname{dom} W\restriction_{n+f(n)-c}$. Then, by considering the following procedure, we see that $n + f(n) < H(\Omega_V\restriction_n) + O(1)$ for all $n \in \mathbb{N}^+$.

Given $\Omega_V\restriction_n$, one first calculates the finite set $\operatorname{dom} W\restriction_{n+f(n)-c}$ by simulating the computation of M with the input n and the oracle $\Omega_V\restriction_n$. Then, by calculating the set $\{\, W(p) \mid p \in \operatorname{dom} W\restriction_{n+f(n)-c}\}$ and picking any one finite binary string s which is not in this set, one can obtain $s \in \{0,1\}^*$ such that $n+f(n)-c < H_W(s)$.

Thus, there exists a partial recursive function $\Psi\colon \{0,1\}^* \to \{0,1\}^*$ such that, for all $n \in \mathbb{N}^+$, $n + f(n) - c < H_W(\Psi(\Omega_V\restriction_n))$. It follows from the optimality of W and (4) that $n + f(n) < H(\Omega_V\restriction_n) + O(1)$ for all $n \in \mathbb{N}^+$, as desired. On the other hand, using (3) we can show that $H(s) \leq |s| + H(|s|) + O(1)$ for all $s \in \{0,1\}^*$. Therefore we have $f(n) < H(n) + O(1)$ for all $n \in \mathbb{N}^+$. Thus, it follows from Lemma 1 that f is bounded to the above. □

Acknowledgments. The proof of Theorem 8 was originally based on the computation history of M with input p in place of $T_M(p)$. Inspired by the suggestion of the anonymous referee to use $(p, T_M(p))$ instead, we can reach the current form of the proof using $T_M(p)$. In addition, the proof of the implication (i) ⇒ (ii) of Theorem 11 is significantly shortened by the suggestion of the same referee. Thus the author is grateful to the referee for the insightful suggestions. This work was supported by KAKENHI, Grant-in-Aid for Scientific Research (C) (20540134), by SCOPE from the Ministry of Internal Affairs and Communications of Japan, and by CREST from Japan Science and Technology Agency.

References

1. C. S. Calude, P. H. Hertling, B. Khoussainov, and Y. Wang, "Recursively enumerable reals and Chaitin Ω numbers," *Theoret. Comput. Sci*, vol. 255, pp. 125–149, 2001.
2. G. J. Chaitin, "A theory of program size formally identical to information theory," *J. Assoc. Comput. Mach.*, vol. 22, pp. 329–340, 1975.
3. G. J. Chaitin, *Algorithmic Information Theory.* Cambridge University Press, Cambridge, 1987.
4. G. J. Chaitin, "Program-size complexity computes the halting problem," *Bulletin of the European Association for Theoretical Computer Science*, vol. 57, p. 198, October 1995.
5. R. G. Downey and D. R. Hirschfeldt, *Algorithmic Randomness and Complexity.* Springer-Verlag, To appear.
6. A. Kučera and T. A. Slaman, "Randomness and recursive enumerability," *SIAM J. Comput.*, vol. 31, No. 1, pp. 199–211, 2001.
7. J. Miller and L. Yu, "On initial segment complexity and degrees of randomness," *Trans. Amer. Math. Soc.*, vol. 360, pp. 3193–3210, 2008.
8. A. Nies, *Computability and Randomness.* Oxford University Press, New York, 2009.
9. R. M. Solovay, "Draft of a paper (or series of papers) on Chaitin's work ... done for the most part during the period of Sept.–Dec. 1974," unpublished manuscript, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, May 1975, 215 pp.