IEICE TRANS. INF. & SYST., VOL.E100–D, NO.3 MARCH 2017

PAPER Special Section on Foundations of Computer Science —New Trends in Theoretical Computer Science—

Autoreducibility and Completeness for Partial Multivalued Functions

Shuji ISOBE[†a)], ***Member and*** **Eisuke KOIZUMI**[†b)], ***Nonmember***

SUMMARY In this paper, we investigate a relationship between many-one-like autoreducibility and completeness for classes of functions computed by polynomial-time nondeterministic Turing transducers. We prove two results. One is that any many-one complete function for these classes is metric many-one autoreducible. The other is that any strict metric many-one complete function for these classes is strict metric many-one autoreducible.

key words: *partial multivalued function, autoreduction, many-one-like reduction*

1. Introduction

Many computational problems are formulated as functional problems. This problem asks, for any given input x, to compute a witness of the membership in some specified language. Functional problems form a class of partial multivalued functions. In this paper, we focus on the classes NPMV and NPMV_g of functions computed by polynomial-time non-deterministic Turing transducers. These classes contain the witness functions for NP languages, the function which maps each string x in an NP language L to strings which witness the membership of x in L, and the inverse functions of (possibly) one-way functions such as the integer factoring function and the discrete logarithm function.

It is well known in the complexity theory that there are many cases in which functions can be reduced to some associated languages. For example, the discrete logarithm function $\mathsf{DL}(p, g, y)$ over a prime field $\mathbb{F}_p$ can be reduced to an NP language $\{(p, g, y, k) \mid \mathsf{DL}(p, g, y) \leq k\}$ by a simple binary search method. Another example is the graph isomorphism problem: for given two isomorphic graphs, an isomorphism (permutations of vertices) can be found by using the decisional oracle which recognizes whether or not any given two graphs are isomorphic. Therefore, one may naturally expect that the complexity properties of many functions can be characterized by the complexity of those associated languages.

On the other hand, there are also cases in which the complexity of functions may not be characterized by the "underlying" languages. Let us consider the #P-complete function #SAT: for any given boolean formula ϕ, $\#\mathsf{SAT}(\phi)$ is the number of satisfying assignments of ϕ. If the function #SAT reduces to the NP-complete language SAT, then the polynomial-time hierarchy PH would collapse to the second level by Toda's theorem [10]. This observation suggests that #P functions may not reduce to NP languages, and that computing #P functions may be strictly harder than recognizing the underlying languages.

The complexity-theoretic property we are interested in is the autoreducibility. A language A is said to be autoreducible if, for any string x, the membership of x in A reduces to the membership, in A, of several strings other than x. Studying the autoreducibility of complete languages is quite important since one can lead to characterizations and separations of complexity classes (e.g. [1], [5]). One can similarly define the autoreducibility of functions, which has also been used in the study on classes of counting functions [2], [8].

Glaßer et al. [5] proved that any complete language for NP and PSPACE is many-one autoreducible. Then Faliszewski and Ogihara [2] proved similar results for the classes #P, SpanP and GapP of single-valued functions. Our intention is to show that similar results still hold for the classes NPMV and NPMV_g of partial multivalued functions. We first consider the many-one reduction (Definition 2.3), and show that any many-one complete function is metric many-one autoreducible (Theorem 1). We next consider a new reduction named the strict metric many-one reduction (Definition 2.4). This reduction is motivated by a simple observation of relationships between SAT and other languages in NP (see Sect. 2). We prove that any strict metric many-one complete function is strict metric many-one autoreducible (Theorem 2).

Faliszewski and Ogihara [2] pointed out that their results show that the notions of the length-decreasing self-reducibility (see Definition 2.7 of [2]) and the autoreducibility are different both on complete languages for NP and PSPACE and on complete functions for #P, SpanP and GapP. Even though our result does not directly lead to separations of complexity classes, the results of ours and Huh et al. [6] show that the same point as Faliszewski and Ogihara's one applies for the classes NPMV and NPMV_g. More concretely, the results imply that there exists a complete function for NPMV or NPMV_g which is autoreducible but not length-decreasing self-reducible unless P = NP.

This paper is organized as follows: Definitions and notations are given in Sect. 2. We state our results, and give their proofs in Sect. 3.

Manuscript received March 24, 2016.
Manuscript revised July 8, 2016.
Manuscript publicized December 21, 2016.

[†]The authors are with Department of Computer and Mathematical Sciences, Graduate School of Information Sciences, Tohoku University, Sendai-shi, 980–8576 Japan.
a) E-mail: iso@cite.tohoku.ac.jp
b) E-mail: koizumi@cite.tohoku.ac.jp

DOI: 10.1587/transinf.2016FCP0006

2. Preliminaries

Let $\Sigma = \{0, 1\}$, and let Σ^* be the set of all strings over Σ of finite length. For a subset $X \subseteq \Sigma^*$, let $\#X$ denote the cardinality of X. One can define the standard lexicographic order $\preceq$ on Σ^*. For a string x, succ(x) denotes the successor of x.

We first refer to the notions of functions and Turing transducers stated in [4] and [6]. Let X and Y be subsets of Σ^*. A (partial multivalued) function from X to Y is a map from X to the power set of Y. Let f be a function from X to Y. Then the set X is called the domain of f, and is denoted by $\mathrm{dom}\, f$. For each string $x \in \mathrm{dom}\, f$, we set

$$\mathrm{graph}\, f = \{(x, y) \in \Sigma^* \times \Sigma^* \mid x \in \mathrm{dom}\, f \text{ and } y \in f(x)\}.$$

A function f is said to be single-valued if $f(x)$ is a singleton set for each $x \in \mathrm{dom}\, f$. When f is single-valued, we regard $f(x)$ as a string of Σ^*.

We use nondeterministic Turing transducers which equip an input tape and an output tape in order to compute functions. We assume that each Turing transducer has a special tape symbol $\perp$ which is not contained in Σ. We also assume that, for any input string x, each Turing transducer always outputs a string y or the symbol $\perp$, and then halts. For a Turing transducer M, we write $M(x) \mapsto y$ if there exists a computation path in M such that M outputs the string y on the input string x. We now define a computation of functions by Turing transducers.

Definition 2.1: A Turing transducer M *computes* a function f if for any pair $(x, y) \in \Sigma^* \times \Sigma^*$, $M(x) \mapsto y$ if and only if $y \in f(x)$.

Let M be a Turing transducer which computes a function f. It follows from this definition that there exists a computation path in M such that M outputs a string $y \in \Sigma^*$ with $(x, y) \in \mathrm{graph}\, f$ for any $x \in \mathrm{dom}\, f$. This means that M nondeterministically recognizes the language $\mathrm{dom}\, f$. Note that M may output $\perp$ even if $x \in \mathrm{dom}\, f$, although the special tape symbol $\perp$ is not contained in Σ. On the other hand, M always outputs $\perp$ whenever $x \notin \mathrm{dom}\, f$.

We briefly refer to some complexity classes of functions [3]. NPMV is the set of all functions which can be computed by a nondeterministic polynomial-time Turing transducer. NPMV_g is the set of functions $f \in \mathsf{NPMV}$ such that $\mathrm{graph}\, f \in \mathsf{P}$. FP is the set of single-valued functions which can be computed by a polynomial-time deterministic Turing transducer.

We now recall two many-one-like reductions: the metric many-one and the many-one reductions [6]. Intuitively, one can make only one query to the oracle in these reductions.

Definition 2.2 ([6]): A function f is *metric many-one* ($\leq^{\mathrm{p}}_{\mathrm{met}}$-) *reducible* to a function g, denoted by $f \leq^{\mathrm{p}}_{\mathrm{met}} g$, if there exist two functions $\psi, \varphi \in \mathsf{FP}$ such that the following conditions hold for any $x \in \Sigma^*$:

(i) if $x \in \mathrm{dom}\, f$, then $\psi(x) \in \mathrm{dom}\, g$ and $\varphi(x, z) \in f(x)$ follows for any $z \in g(\psi(x))$, and
(ii) if $x \notin \mathrm{dom}\, f$, then $(x, z) \notin \mathrm{dom}\, \varphi$ holds for any $z \in g(\psi(x))$.

Definition 2.3 ([6]): A function f is *many-one* ($\leq^{\mathrm{p}}_{\mathrm{m}}$-) reducible to a function g, denoted by $f \leq^{\mathrm{p}}_{\mathrm{m}} g$, if there exist two functions $\psi, \varphi \in \mathsf{FP}$ such that the following conditions hold for any $x \in \Sigma^*$:

(i) if $x \in \mathrm{dom}\, f$, then $\psi(x) \in \mathrm{dom}\, g$ and $\varphi(z) \in f(x)$ follows for any $z \in g(\psi(x))$, and
(ii) if $x \notin \mathrm{dom}\, f$, then $z \notin \mathrm{dom}\, \varphi$ holds for any $z \in g(\psi(x))$.

A function f is $\leq^{\mathrm{p}}_{\mathrm{met}}$-complete for $\mathcal{FC}$ if $f \in \mathcal{FC}$ and $g \leq^{\mathrm{p}}_{\mathrm{met}} f$ holds for any function $g \in \mathcal{FC}$. The completeness is also defined for the many-one reducibility.

In this paper, we define another many-one-like reduction, the strict metric many-one reduction.

Definition 2.4: A function f is *strict metric many-one* ($\leq^{\mathrm{p}}_{\text{s-met}}$-) *reducible* to a function g, denoted by $f \leq^{\mathrm{p}}_{\text{s-met}} g$, if there exist two functions $\psi, \varphi \in \mathsf{FP}$ such that the following conditions hold for any $x \in \Sigma^*$:

(i) $x \in \mathrm{dom}\, f$ if and only if $\psi(x) \in \mathrm{dom}\, g$, and
(ii) if $x \in \mathrm{dom}\, f$, then $\varphi(x, z) \in f(x)$ follows for any $z \in g(\psi(x))$.

The notion of the strict metric many-one reduction is motivated by a simple observation of relationships between SAT and other languages in NP: We note that any function in NPMV_g can be expressed as a witness function wit_A of some language $A \in \mathsf{NP}$ (Proposition 2.2 of [4]). Let sat be a witness function of SAT, that is, for each $\phi \in \mathsf{SAT}$, sat outputs a satisfying assignment of ϕ which witnesses that $\phi \in \mathsf{SAT}$. For any language $A \in \mathsf{NP}$, there exists a reduction ψ from A to SAT such that, using the reduction ψ, one can easily extract a string y witnessing the membership $x \in A$ from a satisfying assignment z of the Boolean formula $\psi(x)$ (the proof of Theorem 13 of [9]). Hence, sat is $\leq^{\mathrm{p}}_{\text{s-met}}$-complete for NPMV_g.

This fact also shows that $A \leq^{\mathrm{p}}_{\mathrm{m}} \mathsf{SAT}$ implies $\mathsf{wit}_A \leq^{\mathrm{p}}_{\text{s-met}}$ sat, that is, SAT has a "witness-preserving" property. Many concrete NP-complete languages also have such a property, and hence, their witness functions are $\leq^{\mathrm{p}}_{\text{s-met}}$-complete for NPMV_g. However, it is still open whether the witness-preserving property holds for *any* language L in NP (see also Remark in Sect. 3 of [6]).

We now consider a relationship between the metric many-one and the strict metric many-one reductions. Assume that $f \leq^{\mathrm{p}}_{\text{s-met}} g$ and that $x \notin \mathrm{dom}\, f$. Then we have $\psi(x) \notin \mathrm{dom}\, g$, that is, $g(\psi(x)) = \emptyset$, and the condition (ii) of Definition 2.2 trivially holds. So $f \leq^{\mathrm{p}}_{\text{s-met}} g$ implies $f \leq^{\mathrm{p}}_{\mathrm{met}} g$. Let us consider whether the converse holds. We define two functions f and g by

$$\mathrm{dom}\, f = \mathsf{SAT}, \quad f(x) = \{(1, y) \mid y \in \mathsf{sat}(x)\},$$
$$\mathrm{dom}\, g = \mathsf{BF},$$

$$g(x) = \begin{cases} \{(1, y) \mid y \in \mathsf{sat}(x)\} & \text{if } x \in \mathsf{SAT}, \\ \{0\} & \text{if } x \in \mathsf{BF} \setminus \mathsf{SAT}, \end{cases}$$

where BF be the set of all the Boolean formulas. We also define two single-valued functions ψ and φ by

$$\begin{aligned} &\operatorname{dom}\psi = \Sigma^*, \ \ \psi(x) = x, \\ &\operatorname{dom}\varphi = \Sigma^* \times (\Sigma^* \setminus \{0\}), \ \ \varphi(x, z) = z. \end{aligned}$$

Then we see that $f \leq^{\mathrm{p}}_{\mathrm{met}} g$ via ψ and φ. On the other hand, if $f \leq^{\mathrm{p}}_{\text{s-met}} g$, then $\mathsf{SAT} \leq^{\mathrm{p}}_{\mathrm{m}} \mathsf{BF}$ follows. This means that $\mathsf{P} = \mathsf{NP}$ holds. Hence, it is unlikely that $f \leq^{\mathrm{p}}_{\mathrm{met}} g$ implies $f \leq^{\mathrm{p}}_{\text{s-met}} g$ for any functions f and g.

Huh et al. [6] defined the notion of strong metric many-one reduction: A function f is strong metric many-one reducible to a function g if $f \leq^{\mathrm{p}}_{\mathrm{met}} g$ and the following equation hold for any $x \in \operatorname{dom} f$:

$$\{\varphi(x, z) \mid z \in g(\psi(x))\} = f(x).$$

This means that any string $y \in f(x)$ can be obtained by $\varphi(x, z)$ for some string $z \in g(\psi(x))$. The strong metric many-one reduction bears no immediate relationship to the strict one.

In this paper, we consider the many-one-like autoreducibility and completeness for NPMV and $\mathsf{NPMV}_{\mathrm{g}}$. The informal definition of the autoreducibility is stated in Introduction. One can naturally apply this definition to many-one-like reductions. Here, we only state the definition of metric many-one autoreducibility. The strict metric many-one autoreducibility is similarly defined.

Definition 2.5: A function f is *metric many-one* ($\leq^{\mathrm{p}}_{\mathrm{met}}$-) *autoreducible* if there exist two functions $\psi, \varphi \in \mathsf{FP}$ such that the following conditions hold for any $x \in \Sigma^*$:

(i) $\psi(x) \neq x$,
(ii) if $x \in \operatorname{dom} f$, then $\psi(x) \in \operatorname{dom} f$ and $\varphi(x, z) \in f(x)$ follows for any $z \in f(\psi(x))$, and
(iii) if $x \notin \operatorname{dom} f$, then $(x, z) \notin \operatorname{dom}\varphi$ holds for any $z \in f(\psi(x))$.

3. Autoreducibility and Completeness for Functions

3.1 Statement of the Result

Let $\mathcal{FC}$ denote one of NPMV and $\mathsf{NPMV}_{\mathrm{g}}$ in this section. We first consider the many-one reduction.

Theorem 1: Let f be $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete for $\mathcal{FC}$ with $\#\operatorname{dom} f \geq 2$. Then f is $\leq^{\mathrm{p}}_{\mathrm{met}}$-autoreducible.

Corollary 4.6 of [2] implies that any $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete function for #P is $\leq^{\mathrm{p}}_{\mathrm{met}}$-autoreducible. So Theorem 1 shows that a similar result holds even when the class #P is replaced with the class NPMV or $\mathsf{NPMV}_{\mathrm{g}}$.

It is natural to ask whether a result similar to Theorem 1 holds for $\leq^{\mathrm{p}}_{\mathrm{met}}$-complete functions. We do not have a complete answer to this question. Alternatively, as a partial answer, we show a result for $\leq^{\mathrm{p}}_{\text{s-met}}$-complete functions. We note that the set of $\leq^{\mathrm{p}}_{\text{s-met}}$-complete functions are contained in that of $\leq^{\mathrm{p}}_{\mathrm{met}}$-complete ones.

Theorem 2: Let f be $\leq^{\mathrm{p}}_{\text{s-met}}$-complete for $\mathcal{FC}$ with $\#\operatorname{dom} f \geq 2$. Then f is $\leq^{\mathrm{p}}_{\text{s-met}}$-autoreducible, and hence, f is $\leq^{\mathrm{p}}_{\mathrm{met}}$-autoreducible.

We prove the theorems in Sects. 3.2 and 3.3, respectively.

We use a function version of the left set technique [7] in order to prove these theorems: In brief, we define another function $f_L \in \mathsf{NPMV}_{\mathrm{g}} \subseteq \mathsf{NPMV}$ for any complete function $f \in \mathcal{FC}$, and we show that f is autoreducible by using the fact that f_L reduces to f. We devote the rest of this subsection to constructing f_L from f.

Let f be any function for $\mathcal{FC}$, and let M_f be a nondeterministic polynomial-time Turing transducer which computes f. Without loss of generality, we can assume that on an input string x, all the (computation) paths of M_f are exactly of length $p(|x|)$ for some polynomial p, where $|x|$ denotes the length of x. Namely, M_f halts with some output in just $p(|x|)$ steps. For an input string x, let $M_f(x; w)$ denote the output of M_f along the path w.

We define a function f_L as follows:

$$\begin{aligned} \operatorname{dom} f_L = \{(x, w) \mid\ & |w| = p(|x|), \\ & w \preceq \exists w_0\, [M_f(x; w_0) \neq \perp]\} \end{aligned}$$

and

$$f_L(x, w) = \{(w_0, y) \mid w \preceq w_0,\ M_f(x; w_0) = y\}.$$

Then the following lemma immediately holds from the definition of the function f_L:

Lemma 3.1: The function f_L satisfies the following properties:

(L1) $f_L \in \mathsf{NPMV}_{\mathrm{g}}$,
(L2) $x \in \operatorname{dom} f$ if and only if $(x, 0^{p(|x|)}) \in \operatorname{dom} f_L$,
(L3) if $(w_0, y) \in f_L(x, w)$, then $y \in f(x)$, and
(L4) if $(x, w) \in \operatorname{dom} f_L$ and $M_f(x; w) = \perp$, then $(x, \mathrm{succ}(w)) \in \operatorname{dom} f_L$.

Remark : The left set technique was used in order to clarify a relationship between many-one-like completeness and autoreducibility for the classes NP [5] and #P [2]. We note that our results does not directly follow from their results even though our ones look similar to their ones.

The construction of f_L stated above is inspired by the proof of Theorem 4.5 of [2]. However, our proof is not a simple application of their proof since the function constructed in it is a single-valued function from Σ^* to $\mathbb{N}$, not a partial multivalued function.

We next consider the following statements (see also Sect. 3 of [6]):

(i) If a function f is $\leq^{\mathrm{p}}_{\text{s-met}}$-autoreducible, then $\operatorname{dom} f$ is $\leq^{\mathrm{p}}_{\mathrm{m}}$-autoreducible.
(ii) If a language L is $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete for NP, then its witness

(P0) Input a string x.
(P1) If $x = x_1$, then set $x_0 = x_2$. Otherwise, set $x_0 = x_1$.
(P2) Compute $x' = \psi_1(x, 0^{p(|x|)})$.
(P3) If $x' \neq x$, then output x', and halt.
(P4) Compute $x'' = \psi_1(x, 1^{p(|x|)})$.
(P5) If $x'' = x$, then

- if $M_f(x; 1^{p(|x|)}) \neq \perp$, then output x_0, and halt.
- otherwise, output $\perp$, and halt.

(P6) Find a string w of length $p(|x|)$ such that $\psi_1(x, w) = x$ and $\psi_1(x, \mathrm{succ}(w)) \neq x$ by the standard binary search.
(P7) If $M_f(x; w) \neq \perp$, then output x_0, and halt. Otherwise, output $\psi_1(x, \mathrm{succ}(w))$, and halt.

Fig. 1 Construction of M_ψ

(Q0) Input a tuple (x, z).
(Q1) If $z = \perp$, then output $\perp$, and halt.
(Q2) Compute $x' = \psi_1(x, 0^{p(|x|)})$.
(Q3) If $x' \neq x$, then output $\varphi_1^2(z)$, and halt.
(Q4) Compute $x'' = \psi_1(x, 1^{p(|x|)})$.
(Q5) If $x'' = x$, then output $M_f(x; 1^{p(|x|)})$, and halt.
(Q6) Find a string w of length $p(|x|)$ such that $\psi_1(x, w) = x$ and $\psi_1(x, \mathrm{succ}(w)) \neq x$ by the standard binary search.
(Q7) If $M_f(x; w) \neq \perp$, then output $M_f(x; w)$, and halt. Otherwise, output $\varphi_1^2(z)$, and halt.

Fig. 2 Construction of M_φ

function wit_L is $\leq^{\mathrm{p}}_{\text{s-met}}$-complete for $\mathsf{NPMV}_{\mathrm{g}}$.

Theorem 2 follows from Theorem 3.1 of [5] if both the statements hold. Conversely, their theorem follows from our one if the converses of the statements (i) and (ii) hold. However, it is not known whether these statements (and their converses) hold.

3.2 Proof of Theorem 1

Let f be $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete for $\mathcal{FC}$ with $\#\operatorname{dom} f \geq 2$, and let $x_1, x_2 \in \operatorname{dom} f$ be two distinct strings. Since $f_L \leq^{\mathrm{p}}_{\mathrm{m}} f$, there exist two functions $\varphi_1, \psi_1 \in \mathsf{FP}$ such that the following two conditions hold:

(C1) If $(x, w) \in \operatorname{dom} f_L$, then $\psi_1(x, w) \in \operatorname{dom} f$ and $\varphi_1(z) = (\varphi_1^1(z), \varphi_1^2(z)) \in f_L(x, w)$ follows for any $z \in f(\psi_1(x, w))$, and
(C2) if $(x, w) \notin \operatorname{dom} f_L$, then $z \notin \operatorname{dom} \varphi_1$ holds for any $z \in f(\psi_1(x, w))$.

In order to define two functions ψ and φ, we construct Turing transducers M_ψ and M_φ. These transducers are depicted in Figs. 1 and 2, respectively.

Note that Steps (P6) and (Q6) are concretely executed as follows:

(B1) Set $w_1 = 0^{p(|x|)}$ and $w_2 = 1^{p(|x|)}$.
(B2) While $\mathrm{succ}(w_1) \neq w_2$, repeat the following procedure:

- Let w' be the middle string between w_1 and w_2.
- If $\psi_1(x, w') = x$, then set $w_1 = w'$. Otherwise, set $w_2 = w'$.

(B3) Set $w = w_1$.

By the definition, we see that $\psi, \varphi \in \mathsf{FP}$ and that $\psi(x) \neq x$ for any $x \in \Sigma^*$.

Lemma 3.2: If $x \in \operatorname{dom} f$, then $\psi(x) \in \operatorname{dom} f$ follows.

Proof. **(I)** Assume that M_ψ halts in Step (P3). We have

$$\begin{aligned} & x \in \operatorname{dom} f \\ \Longrightarrow\ & (x, 0^{p(|x|)}) \in \operatorname{dom} f_L && \text{(by (L2))} \\ \Longrightarrow\ & \psi(x) = x' = \psi_1(x, 0^{p(|x|)}) \in \operatorname{dom} f. && \text{(by (C1))} \end{aligned}$$

(II) Assume that M_ψ halts in Step (P5). We note that $x = x' = x''$.

If $M_f(x; 1^{p(|x|)}) \neq \perp$, then $x \in \operatorname{dom} f$ immediately follows.

Assume that $M_f(x; 1^{p(|x|)}) = \perp$. Since $(x, 1^{p(|x|)}) \notin \operatorname{dom} f_L$, we have $z \notin \operatorname{dom} \varphi_1$ for any $z \in f(x) = f(\psi_1(x, 1^{p(|x|)})) = f(\psi_1(x, 0^{p(|x|)}))$ by the condition (C2). We see $(x, 0^{p(|x|)}) \notin \operatorname{dom} f_L$ from the condition (C1), and $x \notin \operatorname{dom} f$ follows. Consequently, in this case, we have

- if $x \in \operatorname{dom} f$, then $M_f(x; 1^{p(|x|)}) \neq \perp$ and $\psi(x) = x_0 \in \operatorname{dom} f$ hold, and
- if $x \notin \operatorname{dom} f$, then $M_f(x; 1^{p(|x|)}) = \perp$ and $\psi(x) = \perp$ hold.

(III) Assume that M_ψ halts in Step (P7). If $x \in \operatorname{dom} f$ and $M_f(x; w) \neq \perp$, then $\psi(x) = x_0 \in \operatorname{dom} f$ holds.

Assume that $x \in \operatorname{dom} f$ and that $M_f(x; w) = \perp$. Then $\psi_1(x, w) = \psi_1(x, 0^{p(|x|)}) = x \in \operatorname{dom} f$ follows. Since $(x, 0^{p(|x|)}) \in \operatorname{dom} f_L$, we have $z \in \operatorname{dom} \varphi_1$ for any $z \in f(\psi_1(x, 0^{p(|x|)})) = f(\psi_1(x, w))$ by the condition (C1). So, $(x, w) \in \operatorname{dom} f_L$ follows from the condition (C2). We have $(x, \mathrm{succ}(w)) \in \operatorname{dom} f_L$ by the property (L4), and hence, $\psi(x) = \psi_1(x, \mathrm{succ}(w)) \in \operatorname{dom} f$ follows. □

Lemma 3.3: If $x \in \operatorname{dom} f$, then $\varphi(x, z) \in f(x)$ follows for any $z \in f(\psi(x))$.

Proof. **(I)** Assume that M_φ halts in Step (Q3). Then we have $\psi(x) = x' = \psi_1(x, 0^{p(|x|)})$. If $x \in \operatorname{dom} f$, since $(x, 0^{p(|x|)}) \in \operatorname{dom} f_L$, we have $\varphi_1(z) \in f_L(x, 0^{p(|x|)})$ for any $z \in f(\psi_1(x, 0^{p(|x|)})) = f(\psi(x))$. This implies that $\varphi(x, z) = \varphi_1^2(z) \in f(x)$.
(II) Assume that M_φ halts in Step (Q5). In this case, $M_f(x; 1^{p(|x|)}) \neq \perp$ holds if $x \in \operatorname{dom} f$ by the argument (II) of the proof of Lemma 3.2. So we have $\varphi(x, z) = M_f(x; 1^{p(|x|)}) \in f(x)$.
(III) Assume that M_φ halts in Step (Q7). If $x \in \operatorname{dom} f$ and $M_f(x; w) \neq \perp$, then we have $\varphi(x, z) = M_f(x; w) \in f(x)$.

Assume that $x \in \operatorname{dom} f$ and that $M_f(x; w) = \perp$. By the argument (III) of the proof of Lemma 3.2, we have $(x, \mathrm{succ}(w)) \in \operatorname{dom} f_L$ and $\psi(x) = \psi_1(x, \mathrm{succ}(w)) \in \operatorname{dom} f$. Then $\varphi_1(z) \in f_L(x, \mathrm{succ}(w))$ follows for any $z \in$

$f(\psi_1(x, \mathrm{succ}(w))) = f(\psi(x))$. This implies that $\varphi(x, z) = \varphi_1^2(z) \in f(x)$. □

Lemma 3.4: If $x \notin \mathrm{dom}\, f$, then $(x, z) \notin \mathrm{dom}\, \varphi$ holds for any $z \in f(\psi(x))$.

Proof. We first note that $(x, w) \notin \mathrm{dom}\, f_L$ holds for any string w of the length $p(|x|)$ if $x \notin \mathrm{dom}\, f$.
(I) Assume that M_ψ halts in Step (P3). Since $(x, 0^{p(|x|)}) \notin \mathrm{dom}\, f_L$, we have $z \notin \mathrm{dom}\, \varphi_1$ for any $z \in f(\psi_1(x, 0^{p(|x|)})) = f(\psi(x))$. This shows that $\varphi(x, z) = \varphi_1^2(z) = \perp$.
(II) Assume that M_ψ halts in Step (P5). Since $x \notin \mathrm{dom}\, f$, $\psi(x) = \perp$ follows from the argument (II) of the proof of Lemma 3.2.
(III) Assume that M_ψ halts in Step (P7). Since $M_f(x; w) = \perp$, $\psi(x) = \psi_1(x, \mathrm{succ}(w))$ follows. By the same argument as (I), we see that $z \notin \mathrm{dom}\, \varphi_1$ for any $z \in f(\psi_1(x, \mathrm{succ}(w))) = f(\psi(x))$, and hence, $\varphi(x, z) = \varphi_1^2(z) = \perp$ follows. □

This completes the proof of Theorem 1.

Remark : On an input tuple (x, z), the Turing transducer M_φ first computes the string $\psi(x)$, and then outputs an appropriate string. We can show that f is $\leq_{\mathrm{m}}^{\mathrm{p}}$-autoreducible if one can efficiently compute the appropriate string only from z, without knowing x. In general, it is not known whether computing x from $z \in f(\psi(x))$ is easy. If $\psi(x) = \psi_1(x, 0^{p(|x|)})$ or $\psi_1(x, \mathrm{succ}(w))$, then $\varphi_1^2(z) \in f(x)$ follows for any $z \in f(\psi(x))$. However, when $\psi(x) = x_0$, $\varphi_1^2(z) \in f(x)$ does not necessarily hold for any $z \in f(x_0)$. Hence, it seems hard to show that f is $\leq_{\mathrm{m}}^{\mathrm{p}}$-autoreducible.

3.3 Proof of Theorem 2

Let f be any $\leq_{\text{s-met}}^{\mathrm{p}}$-complete function for $\mathcal{FC}$ with $\#\,\mathrm{dom}\, f \geq 2$, and let $x_1, x_2 \in \mathrm{dom}\, f$ be two distinct strings. Since $f_L \leq_{\text{s-met}}^{\mathrm{p}} f$ follows, there exist two functions $\varphi_2, \psi_2 \in \mathsf{FP}$ such that the following two conditions hold:

(C3) $(x, w) \in \mathrm{dom}\, f_L$ if and only if $\psi_2(x, w) \in \mathrm{dom}\, f$, and
(C4) if $(x, w) \in \mathrm{dom}\, f_L$, then

$$\varphi_2((x, w), z) = (\varphi_2^1((x, w), z), \varphi_2^2((x, w), z)) \in f_L(x, w)$$

follows for any $z \in f(\psi_2(x, w))$.

We construct Turing transducers $M_{\widetilde{\psi}}$ and $M_{\widetilde{\varphi}}$, depicted in Figs. 3 and 4, which compute $\widetilde{\psi}$ and $\widetilde{\varphi}$, respectively.

In Steps (P̃6) and (Q̃6), the transducers execute the procedure similar to that stated in the previous subsection.

By the definition, we see that $\widetilde{\psi}, \widetilde{\varphi} \in \mathsf{FP}$ and that $\widetilde{\psi}(x) \neq x$ for any $x \in \Sigma^*$.

Lemma 3.5: $x \in \mathrm{dom}\, f$ if and only if $\widetilde{\psi}(x) \in \mathrm{dom}\, f$.

Proof. **(I)** Assume that $M_{\widetilde{\psi}}$ halts in Step (P̃3). Then we have

$$\begin{aligned} & x \in \mathrm{dom}\, f \\ & \Longleftrightarrow (x, 0^{p(|x|)}) \in \mathrm{dom}\, f_L \qquad \text{(by (L1))} \\ & \Longleftrightarrow \widetilde{\psi}(x) = x' = \psi_2(x, 0^{p(|x|)}) \in \mathrm{dom}\, f. \qquad \text{(by (C3))} \end{aligned}$$

(P̃0) Input a string x.
(P̃1) If $x = x_1$, then set $x_0 = x_2$. Otherwise, set $x_0 = x_1$.
(P̃2) Compute $x' = \psi_2(x, 0^{p(|x|)})$.
(P̃3) If $x' \neq x$, then output x', and halt.
(P̃4) Compute $x'' = \psi_2(x, 1^{p(|x|)})$.
(P̃5) If $x'' = x$, then

- if $M_f(x; 1^{p(|x|)}) \neq \perp$, then output x_0, and halt.
- otherwise, output $\perp$, and halt.

(P̃6) Find a string w of length $p(|x|)$ such that $\psi_2(x, w) = x$ and $\psi_2(x, \mathrm{succ}(w)) \neq x$ by the standard binary search.
(P̃7) If $M_f(x; w) \neq \perp$, then output x_0, and halt. Otherwise, output $\psi_2(x, \mathrm{succ}(w))$, and halt.

Fig. 3 Construction of $M_{\widetilde{\psi}}$

(Q̃0) Input a tuple (x, z).
(Q̃1) If $z = \perp$, then output $\perp$, and halt.
(Q̃2) Compute $x' = \psi_2(x, 0^{p(|x|)})$.
(Q̃3) If $x' \neq x$, then output $\varphi_2^2((x, 0^{p(|x|)}), z)$, and halt.
(Q̃4) Compute $x'' = \psi_2(x, 1^{p(|x|)})$.
(Q̃5) If $x'' = x$, then output $M_f(x; 1^{p(|x|)})$, and halt.
(Q̃6) Find a string w of length $p(|x|)$ such that $\psi_2(x, w) = x$ and $\psi_2(x, \mathrm{succ}(w)) \neq x$ by the standard binary search.
(Q̃7) If $M_f(x; w) \neq \perp$, then output $M_f(x; w)$, and halt. Otherwise, output $\varphi_2^2((x, \mathrm{succ}(w)), z)$, and halt.

Fig. 4 Construction of $M_{\widetilde{\varphi}}$

(II) Assume that $M_{\widetilde{\psi}}$ halts in Step (P̃5). We have

$$\begin{aligned} & M_f(x; 1^{p(|x|)}) \neq \perp \\ & \Longleftrightarrow (x, 1^{p(|x|)}) \in \mathrm{dom}\, f_L \\ & \Longleftrightarrow x = \psi_2(x, 1^{p(|x|)}) \in \mathrm{dom}\, f. \qquad \text{(by (C3))} \end{aligned}$$

Hence, if $x \in \mathrm{dom}\, f$, then $\widetilde{\psi}(x) = x_0 \in \mathrm{dom}\, f$ follows. On the other hand, we have $\widetilde{\psi}(x) = \perp$ when $x \notin \mathrm{dom}\, f$.
(III) Assume that $M_{\widetilde{\psi}}$ halts in Step (P̃7). If $x \notin \mathrm{dom}\, f$, then $M_f(x; w) = \perp$ and $(x, \mathrm{succ}(w)) \notin \mathrm{dom}\, f_L$ hold. So we have $\widetilde{\psi}(x) = \psi_2(x, \mathrm{succ}(w)) \notin \mathrm{dom}\, f$ by the condition (C3).

If $x \in \mathrm{dom}\, f$ and $M(x; w) \neq \perp$, then we have $\widetilde{\psi}(x) = x_0 \in \mathrm{dom}\, f$. Assume that $x \in \mathrm{dom}\, f$ and that $M_f(x; w) = \perp$. Since $\psi_2(x, w) = x \in \mathrm{dom}\, f$, we have $(x, w) \in \mathrm{dom}\, f_L$. Hence, $(x, \mathrm{succ}(w)) \in \mathrm{dom}\, f_L$ follows from the property (L4), and we see that $\widetilde{\psi}(x) = \psi_2(x, \mathrm{succ}(w)) \in \mathrm{dom}\, f$. □

By arguments similar to the proof of Lemma 3.3, we have the following lemma:

Lemma 3.6: If $x \in \mathrm{dom}\, f$, then $\widetilde{\varphi}(x, z) \in f(x)$ follows for any $z \in f(\widetilde{\psi}(x))$.

This completes the proof of Theorem 2.

Remark : Assume that f is $\leq_{\mathrm{met}}^{\mathrm{p}}$-complete and that $M_{\widetilde{\psi}}$

halts in Step ($\tilde{\text{P}}5$). If one can show that $x \in \operatorname{dom} f$ implies that $M_f(x; 1^{p(|x|)}) \neq \perp$, then we can prove that f is $\leq^{\text{p}}_{\text{met}}$-autoreducible by the argument similar to the proof of Theorem 1. Let us assume that $M_f(x; 1^{p(|x|)}) = \perp$. Since $(x, 1^{p(|x|)}) \notin \operatorname{dom} f_L$, we have $((x, 1^{p(|x|)}), z) \notin \operatorname{dom} \varphi_2$ for any $z \in f(\psi_2(x, 1^{p(|x|)})) = f(x) = f(\psi_2(x, 0^{p(|x|)}))$. In order to prove $x \notin \operatorname{dom} f$, we try to show either of the following two statements only from x:

(i) $(x, 0^{p(|x|)}) \notin \operatorname{dom} f_L$,
(ii) $((x, 0^{p(|x|)}), z) \notin \operatorname{dom} \varphi_2$ for some $z \in f(x)$.

In general, we need to compute $M_f(x; w)$ for all w, and it seems hard to efficiently do this.

We finally note that one can avoid this difficulty in the proof of Theorem 1: Since $f_L \leq^{\text{p}}_{\text{m}} f$, $z \notin \operatorname{dom} \varphi_1$ holds for any $z \in f(\psi_1(x, 1^{p(|x|)})) = f(\psi_1(x, 0^{p(|x|)})) = f(x)$ by the condition (C1). We therefore have $(x, 0^{p(|x|)}) \notin \operatorname{dom} f_L$.

References

[1] H. Buhrman, L. Fortnow, D. van Melkebeek and L. Torenvliet, "Separating complexity classes using autoreducibility," SIAM J. Comput., vol.29, pp.1497–1520, 2000.
[2] P. Faliszewski and M. Ogihara, "On the Autoreducibility of Functions," Theory Comput. Syst., vol.46, pp.222–245, 2010.
[3] S. Fenner, S. Homer, M. Ogihara, and A. Selman, "Oracles that compute values," SIAM J. Comput., vol.26, no.4, pp.1043–1065, 1997.
[4] K. Fleszar, C. Glaßer, F. Lipp, C. Reitwießner, and M. Witek, "The Complexity of Solving Multiobjective Optimization Problems and its Relation to Multivalued Functions," Electronic Colloquium on Computational Complexity, TR11-053, 2011.
[5] C. Glaßer, M. Ogihara, A. Pavan, A.L. Selman, and L. Zhang, "Autoreducibility, mitoticity, and immunity," J. Comput. Syst. Sci., vol.73, no.5, pp.735–754, 2007.
[6] J.-W. Huh, S. Isobe, E. Koizumi, and H. Shizuya, "On the Length-Decreasing Self-Reducibility and the Many-One-Like Reducibilities for Partial Multivalued Functions," IEICE Trans. Inf. and Syst., vol.E96-D, no.3, pp.465–471, 2013.
[7] M. Ogiwara and O. Watanabe, "Polynomial-Time Bounded Truth-Table Reducibility of NP Sets to Sparse Sets," SIAM J. Comput., vol.20, no.3, pp.471–483, 1991.
[8] A. Pagourtzis, and S. Zachos, "The complexity of counting functions with easy decision version," Proc. 31st International Symposium on Mathematical Foundations of Computer Science, LNCS, vol.4162, pp.741–752, Springer, Berlin, 2006.
[9] A.L. Selman, "A Taxonomy of Complexity Classes of Functions," J. Comput. Syst. Sci., vol.48, no.2, pp.357–381, 1994.
[10] S. Toda, "PP is as Hard as the Polynomial-Time Hierarchy," SIAM J. Comput., vol.20, no.5, pp.865–877, 1991.

Shuji Isobe received the B.Eng. degree from Tohoku University, Japan, in 1997, and M.S. and Ph. D. degrees in information science from Graduate School of Information Sciences, Tohoku University, Japan, in 1999 and 2002, respectively. He has been with Tohoku University since 2002, and has been Associate Professor since 2009. His research interests include information security theory, computational complexity theory and discrete mathematics.

Eisuke Koizumi received the B.Sci. degree from Tohoku University, Japan, in 1998, and M.S. and Ph. D. degrees in science from Tohoku University, Japan, in 2000 and 2005, respectively. He has been with Tohoku University since 2005, where he is Assistant Professor. His research interests include information security theory, computational complexity theory and function theory.