PAPER *Special Section on Foundations of Computer Science*

# On the Length-Decreasing Self-Reducibility and the Many-One-Like Reducibilities for Partial Multivalued Functions

**Ji-Won HUH**[†*], ***Nonmember***, **Shuji ISOBE**[†a)], ***Member***, **Eisuke KOIZUMI**[†b)], ***Nonmember***, ***and*** **Hiroki SHIZUYA**[†c)], ***Member***

**SUMMARY** In this paper, we investigate a relationship between the length-decreasing self-reducibility and the many-one-like reducibilities for partial multivalued functions. We show that if any parsimonious (many-one or metric many-one) complete function for NPMV (or $\mathsf{NPMV_g}$) is length-decreasing self-reducible, then any function in NPMV (or $\mathsf{NPMV_g}$) has a polynomial-time computable refinement. This result implies that there exists an NPMV (or $\mathsf{NPMV_g}$)-complete function which is not length-decreasing self-reducible unless P = NP.

***key words:*** *partial multivalued function, length-decreasing self-reduction, many-one-like reduction*

## 1. Introduction

Many computational problems are formulated as functional problems. Such a problem typically asks, for a given instance $x$, to find a witness of the membership in some specified language. Functions induced by these problems above form a class of partial multivalued functions which are computed by nondeterministic Turing transducers. These functions and associated complexity classes have widely been studied in the computational complexity theory. In this paper, we are interested in the notion of the self-reducibility [10] of a function, and investigate the property for some classes of partial multivalued functions.

Intuitively, a language $A$ is said to be self-reducible if, for any string $x$, the membership of $x$ in $A$ reduces to the membership, in $A$, of several strings smaller than $x$ with respect to some specified partial order. The self-reducibility of (multivalued) functions is similarly defined. The notion of self-reducibilities has played an important role in separations and characterizations of classes of languages [3], [5], [8], [9] and counting functions [4], [11]. Therefore, one can naturally expect that self-reducibilities contribute to the development of the computational complexity theory of partial multivalued functions. In this paper, we concentrate on the length-decreasing self-reducibility [1], [4], one of the self-reducibilities.

In the length-decreasing self-reduction, one can query the oracle only about strings which are shorter than the input string. Faliszewski and Ogihara [4] pointed out that many concrete complete languages are length-decreasing self-reducible. For example, the NP-complete language SAT is length-decreasing self-reducible: For a non-trivial Boolean formula $\phi = \phi(x_1, \ldots, x_n)$, $\phi$ is satisfiable if and only if at least one of the two shorter formulas $\phi(0, x_2, \ldots, x_n)$ or $\phi(1, x_2, \ldots, x_n)$ is satisfiable. Similarly, the #P-complete function #SAT and the PSPACE-complete language QBF are also length-decreasing self-reducible [4].

Faliszewski and Ogihara [4] considered whether any complete language for NP (or PSPACE) is length-decreasing self-reducible, and showed that this is unlikely. More precisely, they proved that P = NP (or P = PSPACE) if this statement holds ([4, Corollary 3.4]). It was also proved that a similar result follows for classes #P, SpanP and GapP of counting functions ([4, Corollary 3.8]).

In this paper, we show that their results mentioned above still hold even when each class is replaced with the class NPMV or $\mathsf{NPMV_g}$ of partial multivalued functions computed by polynomial-time nondeterministic Turing transducers (Theorem 1 and Corollary 4). Our result means that there exists an NPMV (or $\mathsf{NPMV_g}$)-complete function which is not length-decreasing self-reducible unless P = NP (Corollary 5).

We note that a witness function of many concrete NP-complete languages is $\mathsf{NPMV_g}$-complete: Let sat be a witness function of SAT, that is, for each $\phi \in$ SAT, sat outputs a satisfying assignment $x$ of $\phi$ which witnesses that $\phi \in$ SAT. Then sat is $\mathsf{NPMV_g}$-complete (see Sect. 3 and [12]). We consider the following two hypotheses: (i) A witness function of any NP-complete language would be $\mathsf{NPMV_g}$-complete. (ii) If a witness function of a language $L$ is length-decreasing self-reducible, then $L$ would be length-decreasing self-reducible. Our results immediately follow from Corollary 3.4 of [4] if both (i) and (ii) hold. However, it is not known whether these hypotheses hold. Hence, our results seem not to be trivial even though our results and proofs look similar to those of [4] (see also Sect. 3).

This paper is organized as follows: In Sect. 2, we give some definitions and notations. In Sects. 3–5, we state our results, and give their proofs. Concluding remarks are given in Sect. 6.

Manuscript received March 21, 2012.
Manuscript revised July 7, 2012.

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

DOI: 10.1587/transinf.E96.D.465

## 2. Preliminaries

Let $\Sigma = \{0, 1\}$, and let $\Sigma^*$ be the set of all strings over $\Sigma$ of finite length. For $x \in \Sigma^*$, $|x|$ denotes the length of $x$.

A (partial multivalued) function $f : X \to Y$ from a subset $X$ of $\Sigma^*$ to a subset $Y$ of $\Sigma^*$ maps each string $x \in X$ into one or more strings in $Y$. We write $f(x) \mapsto y$ when a string $x$ corresponds to a string $y$ via $f$. The set $X$ is called the domain of $f$, and is denoted by $\mathrm{dom}\, f$. In particular, $f$ is called a total function if $\mathrm{dom}\, f = \Sigma^*$. The graph of $f$ is defined by

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

For each string $x \in \mathrm{dom}\, f$, we set

$$\text{set-}f(x) = \{y \in \Sigma^* \mid f(x) \mapsto y\}.$$

A function $f$ is said to be single-valued if set-$f(x)$ is a singleton set for each $x \in \mathrm{dom}\, f$. When $f$ is single-valued, we write $f(x) = y$ or $y = f(x)$ instead of $f(x) \mapsto y$. A function $g$ is called a refinement of a function $f$ when $\mathrm{dom}\, g = \mathrm{dom}\, f$ and $\mathrm{graph}\, g \subseteq \mathrm{graph}\, f$ hold. Let $\mathcal{FC}$ and $\mathcal{FC}'$ be two classes of functions. For a function $f$, we write $f \in_c \mathcal{FC}$ if there exists a function $g \in \mathcal{FC}$ such that $g$ is a refinement of $f$. If $f \in_c \mathcal{FC}'$ holds for any $f \in \mathcal{FC}$, then we write $\mathcal{FC} \subseteq_c \mathcal{FC}'$.

We note that a function can also be defined by a (binary) relation on $\Sigma^*$. We use this notion in this paper. Let $R \subseteq \Sigma^* \times \Sigma^*$ be a relation on $\Sigma^*$. Then we can define a function $f$ as follows: Set

$$\mathrm{dom}\, f = \{x \in \Sigma^* \mid \exists y \in \Sigma^* \, [(x, y) \in R]\},$$

and for $x \in \mathrm{dom}\, f$, define $f(x) \mapsto y$ so that $(x, y) \in R$. The function $f$ is called the function associated with the relation $R$. By the definition, we see $\mathrm{graph}\, f = R$ and

$$\text{set-}f(x) = \{y \in \Sigma^* \mid (x, y) \in R\}$$

for any $x \in \mathrm{dom}\, f$.

In this paper, 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 $\Sigma$. For a Turing transducer $M$, we write $M(x) \mapsto y$ if there exists a computation 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 $f(x) \mapsto y$.

Let $M$ be a Turing transducer which computes a function $f$. It follows from this definition that there exists a computation 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 $\Sigma$. On the other hand, $M$ always outputs $\perp$ whenever $x \notin \mathrm{dom}\, f$.

We briefly refer to some complexity classes of functions [7]. $\mathsf{NPMV}$ is the set of all functions which can be computed by a nondeterministic polynomial-time Turing transducer. $\mathsf{NPMV_g}$ is the set of functions $f \in \mathsf{NPMV}$ such that $\mathrm{graph}\, f \in \mathsf{P}$. $\mathsf{PF}$ is the set of all functions which can be computed by a polynomial-time deterministic Turing transducer.

We state Turing transducers with oracles, called oracle Turing transducers. We assume that any oracle is a single-valued function. An oracle Turing transducer contains an oracle query tape, an oracle answer tape and an oracle call state. Let $g$ be a single-valued function, and let $M$ be an oracle Turing transducer with the oracle $g$. When a string $w$ is written on the oracle query tape and $M$ enters the oracle call state, $M$ works as follows:

- If $w \in \mathrm{dom}\, g$, then the string $g(w)$ is written on the oracle answer tape, and
- if $w \notin \mathrm{dom}\, g$, then $\perp$ is written on the oracle answer tape.

We shall assume, without loss of generality, that $M$ never makes the same query as before, that is, all the queries are distinct.

We are now ready to define the reducibilities between functions.

**Definition 2.2** ([7]): A function $f$ is polynomial-time *Turing* ($\leq^{\mathrm{p}}_{\mathrm{T}}$-) reducible to a function $g$, denoted by $f \leq^{\mathrm{p}}_{\mathrm{T}} g$, if there exists a polynomial-time deterministic oracle Turing transducer $M$ such that for any single-valued refinement $g'$ of $g$, $M[g']$, the transducer $M$ with the oracle $g'$, computes a single-valued refinement of $f$.

We next define many-one-like reducibilities. Intuitively, one can make only one query to the oracle in many-one-like reductions. The definitions for total single-valued functions, including counting functions, are stated in [4]. In this paper, we formulate the partial multivalued function version of many-one-like reducibilities.

**Definition 2.3** (cf. [6, Definition 2]): 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 $\psi, \varphi \in \mathsf{PF}$ 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 $\varphi(x, y) \in$ set-$f(x)$ follows for any $y \in$ set-$g(\psi(x))$, and
(ii) if $x \notin \mathrm{dom}\, f$, then $(x, y) \notin \mathrm{dom}\, \varphi$ holds for any $y \in$ set-$g(\psi(x))$.

$f$ is *strongly metric many-one* ($\leq^{\mathrm{p}}_{\mathrm{smet}}$-) reducible to $g$, denoted by $f \leq^{\mathrm{p}}_{\mathrm{smet}} g$, if $f \leq^{\mathrm{p}}_{\mathrm{met}} g$ and the condition (iii) below holds:

(iii) for any $(x, z) \in \mathrm{graph}\, f$, there exists $y \in$ set-$g(\psi(x))$ such that $z = \varphi(x, y)$.

We note the following two facts before defining other many-one-like reducibilities:

- When $f \leq^{\mathrm{p}}_{\mathrm{smet}} g$, any string $z \in$ set-$f(x)$ can be obtained through some string $y$ with $g(\psi(x)) \mapsto y$. On the other hand, when $f \leq^{\mathrm{p}}_{\mathrm{met}} g$, there may exist a string $z \in$ set-$f(x)$ which cannot be obtained through the oracle $g$.
- The condition (ii) is redundant when $f$ is a total function, and the condition (iii) immediately follows from the condition (i) when $f$ is single-valued. Namely, when $f$ is single-valued, $f \leq^{\mathrm{p}}_{\mathrm{met}} g$ implies $f \leq^{\mathrm{p}}_{\mathrm{smet}} g$.

**Definition 2.4:** 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 $\psi, \varphi \in \mathsf{PF}$ 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 $\varphi(y) \in$ set-$f(x)$ follows for any $y \in$ set-$g(\psi(x))$, and
(ii) if $x \notin \mathrm{dom}\, f$, then $y \notin \mathrm{dom}\, \varphi$ holds for any $y \in$ set-$g(\psi(x))$.

$f$ is *strongly many-one* ($\leq^{\mathrm{p}}_{\mathrm{sm}}$-) reducible to $g$, denoted by $f \leq^{\mathrm{p}}_{\mathrm{sm}} g$, if $f \leq^{\mathrm{p}}_{\mathrm{m}} g$ and the condition (iii) below holds:

(iii) for any $(x, z) \in \mathrm{graph}\, f$, there exists $y \in$ set-$g(\psi(x))$ such that $z = \varphi(y)$.

**Definition 2.5** (cf. [2, Definition 1]): A function $f$ is *parsimoniously* ($\leq^{\mathrm{p}}_{\mathrm{par}}$-) reducible to a function $g$, denoted by $f \leq^{\mathrm{p}}_{\mathrm{par}} g$, if there exists a function $\psi \in \mathsf{PF}$ 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 set-$g(\psi(x)) \subseteq$ set-$f(x)$ follow, and
(ii) if $x \notin \mathrm{dom}\, f$, then $\psi(x) \notin \mathrm{dom}\, g$ holds.

$f$ is *strongly parsimoniously* ($\leq^{\mathrm{p}}_{\mathrm{spar}}$-) reducible to $g$, denoted by $f \leq^{\mathrm{p}}_{\mathrm{spar}} g$, if $f \leq^{\mathrm{p}}_{\mathrm{par}} g$ and the condition (iii) below holds:

(iii) set-$g(\psi(x)) =$ set-$f(x)$.

We call the reducibilities defined in Definitions 2.3 through 2.5 the many-one-like reducibilities. Note that, in [2], the term "many-one reducibility" is used to denote the strongly parsimonious reducibility defined above.

By definitions of the Turing reducibility and the many-one-like reducibilities, we have the following proposition:

**Proposition 2.6:** Let $f$ and $g$ be functions.

- $f \leq^{\mathrm{p}}_{\mathrm{par}} g \Longrightarrow f \leq^{\mathrm{p}}_{\mathrm{m}} g \Longrightarrow f \leq^{\mathrm{p}}_{\mathrm{met}} g \Longrightarrow f \leq^{\mathrm{p}}_{\mathrm{T}} g$.
- $f \leq^{\mathrm{p}}_{\mathrm{spar}} g \Longrightarrow f \leq^{\mathrm{p}}_{\mathrm{sm}} g \Longrightarrow f \leq^{\mathrm{p}}_{\mathrm{smet}} g$.

A function $f$ is $\leq^{\mathrm{p}}_{\mathrm{spar}}$-hard for a class $\mathcal{FC}$ of functions if $g \leq^{\mathrm{p}}_{\mathrm{spar}} f$ holds for any function $g \in \mathcal{FC}$. A function $f$ is $\leq^{\mathrm{p}}_{\mathrm{spar}}$-complete for $\mathcal{FC}$ if $f \in \mathcal{FC}$ and $f$ is $\leq^{\mathrm{p}}_{\mathrm{spar}}$-hard for $\mathcal{FC}$. These notions are also defined for other reducibilities.

At the end of this section, we define the length-decreasing self-reducibility of functions. The language version of the length-decreasing self-reducibility is similarly defined.

**Definition 2.7:** A function $f$ is (polynomial-time) *length-decreasing self-reducible* if there exists a polynomial-time deterministic oracle Turing transducer $M$ such that, for any single-valued refinement $f'$ of $f$, $M[f']$ computes a single-valued refinement of $f$, where, on any input $x \in \Sigma^*$, $M$ queries the oracle $f'$ only about strings $y \in \Sigma^*$ with $|y| < |x|$.

## 3. Main Result

Let $\mathcal{FC}$ denote one of $\mathsf{NPMV}$ and $\mathsf{NPMV_g}$ in the rest of this paper. We now state our main theorem, which is an extension, to $\mathcal{FC}$, of Corollaries 3.4 and 3.8 of [4].

**Theorem 1:** Assume that $\mathcal{FC}$ has some $\leq^{\mathrm{p}}_{\mathrm{spar}}$-complete function. If all $\leq^{\mathrm{p}}_{\mathrm{spar}}$-complete functions for $\mathcal{FC}$ are length-decreasing self-reducible, then $\mathcal{FC} \subseteq_c \mathsf{PF}$ holds.

One needs to assume the existence of $\leq^{\mathrm{p}}_{\mathrm{spar}}$-complete functions in this theorem since it is not known whether such functions in fact exist.

The following lemma, which is an extension of Theorem 3.7 of [4], plays an important role in order to prove the theorem:

**Lemma 2:** For any $f_1 \in \mathcal{FC}$, there exists a function $f_2$ which satisfies the following properties:

(i) $f_1 \leq^{\mathrm{p}}_{\mathrm{spar}} f_2$, $f_2 \leq^{\mathrm{p}}_{\mathrm{spar}} f_1$, and
(ii) if $f_2$ is length-decreasing self-reducible, then $f_2 \in_c \mathsf{PF}$.

We also use the closure property of the class $\mathcal{FC}$ under the $\leq^{\mathrm{p}}_{\mathrm{spar}}$-reducibility, as stated in the following proposition:

**Proposition 3:** The class $\mathcal{FC}$ is closed under $\leq^{\mathrm{p}}_{\mathrm{spar}}$-reducibility, that is, if $f \leq^{\mathrm{p}}_{\mathrm{spar}} g$ and $g \in \mathcal{FC}$, then $f \in \mathcal{FC}$ follows.

We prove Lemma 2 and Proposition 3 in Sects. 4 and 5, respectively.

*Proof of Theorem* 1. Let $f_1$ be an arbitrary $\leq^{\mathrm{p}}_{\mathrm{spar}}$-complete function for $\mathcal{FC}$. Then, there exists a function $f_2$ which satisfies the condition (i) of Lemma 2. Proposition 3 implies that $f_2 \in \mathcal{FC}$, and we see that $f_2$ is also $\leq^{\mathrm{p}}_{\mathrm{spar}}$-complete for $\mathcal{FC}$. By the assumption and the condition (ii) of Lemma 2, we have $f_2 \in_c \mathsf{PF}$. Since $f_2$ is $\leq^{\mathrm{p}}_{\mathrm{spar}}$-complete for $\mathcal{FC}$, each function in $\mathcal{FC}$ also has a single-valued refinement contained in $\mathsf{PF}$. □

Theorem 1 still holds if we replace $\leq^{\mathrm{p}}_{\mathrm{spar}}$ with $\leq^{\mathrm{p}}_{\mathrm{sm}}$ or $\leq^{\mathrm{p}}_{\mathrm{smet}}$.

**Corollary 4:** Let $\leq$ denote one of $\leq^{\mathrm{p}}_{\mathrm{sm}}$ and $\leq^{\mathrm{p}}_{\mathrm{smet}}$. If all $\leq$-complete functions for $\mathcal{FC}$ are length-decreasing self-reducible, then $\mathcal{FC} \subseteq_c \mathsf{PF}$ holds.

We note that one does not have to assume the existence of $\leq$-complete functions in this corollary: Let $\mathsf{sat}$ be a witness function of $\mathsf{SAT}$, that is, for each $\phi \in \mathsf{SAT}$, $\mathsf{sat}$ outputs a satisfying assignment $x$ of $\phi$ which witnesses that $\phi \in \mathsf{SAT}$. Then $\mathsf{sat}$ is $\leq$-complete for $\mathcal{FC}$ (see the proof of [12, Theorem 13]).

*Proof of Corollary* 4. Let $f_1$ be a $\leq$-complete function for $\mathcal{FC}$. Since $f_1 \in \mathcal{FC}$, there exists a function $f_2$ specified by Lemma 2. We have $f_2 \leq^{\mathrm{p}}_{\mathrm{spar}} f_1$ by the condition (i) of the lemma. (Note that $f_2 \leq^{\mathrm{p}}_{\mathrm{spar}} f_1$ follows although $\leq$ is not $\leq^{\mathrm{p}}_{\mathrm{spar}}$.) Since $f_1 \in \mathcal{FC}$ and $f_2 \leq^{\mathrm{p}}_{\mathrm{spar}} f_1$, we have $f_2 \in \mathcal{FC}$ by Proposition 3. Further, it follows from $f_1 \leq^{\mathrm{p}}_{\mathrm{spar}} f_2$ and Proposition 2.6 that $f_1 \leq f_2$ holds. Hence, we see that $f_2$ is $\leq$-complete for $\mathcal{FC}$. By the assumption and the condition (ii) of Lemma 2, we have $f_2 \in_c \mathsf{PF}$. Since $f_2$ is $\leq$-complete for $\mathcal{FC}$, each function in $\mathcal{FC}$ also has a single-valued refinement contained in $\mathsf{PF}$. □

Since it is shown, in [12], that $\mathsf{P} = \mathsf{NP}$ holds if and only if $\mathcal{FC} \subseteq_c \mathsf{PF}$, we have the following corollary:

**Corollary 5:** If all strongly parsimonious (many-one or metric many-one) complete functions for $\mathcal{FC}$ are length-decreasing self-reducible, then $\mathsf{P} = \mathsf{NP}$.

We summarize the relationships between our result and the result of Faliszewski and Ogihara [4]. Let $\leq$ denote one of $\leq^{\mathrm{p}}_{\mathrm{spar}}$, $\leq^{\mathrm{p}}_{\mathrm{sm}}$ and $\leq^{\mathrm{p}}_{\mathrm{smet}}$. We now consider the following four statements:

- (i) If a function $f$ is length-decreasing self-reducible, then $\mathrm{dom}\, f$ is length-decreasing self-reducible.
- (ii) For a function $f$, if $\mathrm{dom}\, f$ is length-decreasing self-reducible, then $f$ is length-decreasing self-reducible.
- (iii) A witness function of any $\mathsf{NP}$-complete language is $\leq$-complete for $\mathsf{NPMV_g}$.
- (iv) Any $\leq$-complete function for $\mathsf{NPMV_g}$ is formed as a witness function of some $\mathsf{NP}$-complete language.

As stated in Introduction, our results follow from Corollary 3.4 of [4] if the statements (i) and (iii) hold. Conversely, the corollary follows from our results if the statements (ii) and (iv) hold. However, it is not known whether these four statements hold. Although our results and proofs are similar to those of [4], our results do not imply those of [4], and their results do not imply our results, either.

**Remark :** The statement (iii) says that for any $\mathsf{NP}$-complete language $L$ and any language $A \in \mathsf{NP}$, there exists a "witness-preserving" reduction $\psi$ from $A$ to $L$ in a way that, using the reduction $\psi$, one can easily extract a string $y$ witnessing the membership $x \in A$ from a string $w$ witnessing the membership $\psi(x) \in L$. Many concrete $\mathsf{NP}$-complete languages, including $\mathsf{SAT}$, have such a "witness-preserving" property. However, it is still open whether *any* $\mathsf{NP}$-complete language has the property.

In general, the complexity of computing a function can be much harder than that of recognizing its domain. Therefore, for any functions $f$ and $g$, the reduction $\mathrm{dom}\, f \leq \mathrm{dom}\, g$ of domains does not necessarily imply the reduction $f \leq g$ of functions and vice versa when $\leq$ is one of $\leq^{\mathrm{p}}_{\mathrm{spar}}$, $\leq^{\mathrm{p}}_{\mathrm{sm}}$ and $\leq^{\mathrm{p}}_{\mathrm{smet}}$.

## 4. Proof of Lemma 2

We now return to Lemma 2. Our proof is not a simple application of the proof of Theorem 3.7 of [4] since the function constructed in it is a total single-valued function. Our construction of the function $f_2$ is motivated by the proof of Theorem 3.3 of [4].

We first define a function $\rho : \mathbb{N} \cup \{0\} \to \mathbb{N}$ by

$$\rho(n) = \min\left\{2^{2^i} \mid i \geq 0,\ 2^{2^i} > n\right\}.$$

We have $\rho(0) = \rho(1) = 2$ and $\rho(|x|)^{1/2} \leq |x| < \rho(|x|)$ for $|x| \geq 2$. This implies that $\rho(|x|) \leq |x|^2 + 2$ for any $x \in \Sigma^*$. In addition, we can compute $\rho(|x|)$ by at most $i = O(\log\log|x|)$ times successively calculating such as $2, 2^2, (2^2)^2, \ldots, (2^{2^{i-1}})^2$. Hence, $\rho(|x|)$ can be computed in time polynomial in $|x|$.

Let $f_1 \in \mathcal{FC}$. We define a subset $X_2$ of $\Sigma^*$ and a relation $R_2$ on $\Sigma^*$ as follows:

$$X_2 = \{x10^m \mid x \in \mathrm{dom}\, f_1,\ 1 + |x| + m = \rho(|x|)\}$$

and

$$R_2 = \{(y, z) \mid y = x10^m \in X_2,\ z \in \text{set-}f_1(x)\}.$$

Let $f_2$ denote the function associated with the relation $R_2$. Note that $\mathrm{dom}\, f_2 = X_2$. We prove that $f_2$ satisfies the properties (i) and (ii) of Lemma 2.

We show that the property (i) holds. Define a function $\psi$ as follows: The domain $\mathrm{dom}\,\psi$ is $\Sigma^*$, and for each $x \in \mathrm{dom}\,\psi$, $\psi(x)$ is defined by $\psi(x) \mapsto x10^m$, where $m = \rho(|x|) - 1 - |x|$. Since $\rho(|x|)$ is computable in time polynomial in $|x|$, $\psi \in \mathsf{PF}$ follows. By the definitions of $X_2$ and $\psi$, we see that $x \in \mathrm{dom}\, f_1$ if and only if $\psi(x) \in \mathrm{dom}\, f_2$. Let $x \in \mathrm{dom}\, f_1$. Noting that $\psi(x) \in X_2$, we have

$$\begin{aligned} z \in \text{set-}f_1(x) &\iff (\psi(x), z) \in R_2 = \mathrm{graph}\, f_2 \\ &\iff z \in \text{set-}f_2(\psi(x)). \end{aligned}$$

This implies that $f_1 \leq^{\mathrm{p}}_{\mathrm{spar}} f_2$.

On the other hand, set

$$X_2' = \{y \mid y = x10^m,\ m = \rho(|x|) - |x| - 1\}, \qquad (1)$$

and define a function $\psi'$ as follows: The domain $\mathrm{dom}\,\psi'$ is the set $X_2'$, and for each $y = x10^m \in \mathrm{dom}\,\psi'$, $\psi'(y)$ is defined by $\psi'(y) \mapsto x$. We note that the following facts hold: $\mathrm{dom}\, f_2 = X_2 \subseteq X_2'$, $X_2' \in \mathsf{P}$ and $\psi' \in \mathsf{PF}$. If $y = x10^m \in \mathrm{dom}\, f_2$, then we have $\psi'(y) = x \in \mathrm{dom}\, f_1$. Assume that $y \notin \mathrm{dom}\, f_2$. If $y \notin X_2'$, then $\psi'(y) \notin \mathrm{dom}\, f_1$ immediately follows since $y \notin \mathrm{dom}\,\psi'$. When $y = x10^m \in X_2' \setminus \mathrm{dom}\, f_2$, we have $\psi'(y) = x \notin \mathrm{dom}\, f_1$ by the definition of $\mathrm{dom}\, f_2$. Let $y = x10^m \in \mathrm{dom}\, f_2$. Then we see

$$\begin{aligned} z \in \text{set-}f_2(y) &\iff (y, z) \in \mathrm{graph}\, f_2 = R_2 \\ &\iff z \in \text{set-}f_1(x) = \text{set-}f_1(\psi'(y)), \end{aligned}$$

proving $f_2 \leq^{\mathrm{p}}_{\mathrm{spar}} f_1$.

We show that $f_2$ satisfies the condition (ii). The proof is based on those of Theorem 3.3 of [4]. Assume that $f_2$ is length-decreasing self-reducible. There exists a polynomial-time deterministic oracle Turing transducer $M$ which satisfies the following properties:

(M-1) $M[f_2']$ computes a single-valued refinement of $f_2$ for any single-valued refinement $f_2'$ of $f_2$, and
(M-2) on any input $x$, $M$ queries the oracle only about strings which are shorter than $x$.

Since the set $X_2'$ defined by (1) is in $\mathsf{P}$, without loss of generality, we may assume that $M$ works as follows:

(M-3) On an input $x \in \Sigma^*$, $M$ first checks whether $x$ is an element of $X_2'$. If so, then $M$ works on the input $x$. Otherwise, $M$ outputs $\perp$, and halts.
(M-4) When a string $z$ is written on the oracle query tape, $M$ checks that $z \in X_2'$. If so, then $M$ enters the oracle call state. Otherwise, considering that $\perp$ is written on the oracle answer tape, $M$ continues to work.

We construct a deterministic Turing transducer $M'$ as follows:

(1) $M'$ takes as an input $x$.
(2) Simulate $M$ on the input $x$.
(3) If $M$ calls the oracle with a query $w$, then simulate $M$ on the input $w$ by a recursive call in order to obtain the answer of the oracle.
(4) If the simulation of $M$ on the input $x$ is completed with an output $y$, then output $y$, and halt.

Let $p(n)$ and $P(n)$ denote the running times of $M$ and $M'$ on any input of the length $n$, respectively. By the definition of $M$, $p(n)$ is a polynomial in $n$. In order to prove the property (ii) of Lemma 2, it is sufficient to show the following statements:

(S-1) $M'$ computes a single-valued refinement of $f_2$, and
(S-2) $P(n)$ is a polynomial in $n$.

We set

$$D_1 = \{x \mid |x| \neq 2^{2^i} \text{ for any } i \geq 0\}$$

and

$$D_2 = \Sigma^* \setminus D_1 = \{x \mid |x| = 2^{2^i} \text{ for some } i \geq 0\}.$$

By the definition of $f_2$, we have $\operatorname{dom} f_2 \cap D_1 = \emptyset$ and $\operatorname{dom} f_2 \subseteq D_2$.

When $x \in D_1$, by the property (M-3), $M$ outputs $\perp$ in Step (2), and hence, $M'$ outputs $\perp$ and halts in Step (4). This implies that the statement (S-1) holds for any element in $D_1$. In addition, we see that $P(|x|)$ is at most $p(|x|)$ when $x \in D_1$.

We prove the statement (S-1) by induction on $i$ when $x \in D_2$. When $|x| = 2 = 2^{2^0}$, $M$ does not query the oracle by the properties (M-2) and (M-4). So the statement follows from the construction of $M'$.

Set $|x| = 2^{2^i}$, and inductively assume that the statement follows when an input string $x'$ satisfies $|x'| = 2^{2^t}$ for $t = 0, \ldots, i-1$. When $M$ calls the oracle with a query $z$ in Step (3), $z$ is of the form $|z| = 2^{2^t}$ for some $t < i$ by the properties (M-2) and (M-4). Comparing Step (3) with the construction of $M'$, by the induction hypothesis, we see that $M'$ can obtain an element of set-$f_2(z)$ or $\perp$ in Step (3). Hence, $M'$ outputs an element of set-$f_2(x)$ or $\perp$ in Step (4), and the statement (S-1) follows.

We prove the statement (S-2). When $x \in D_1$, we have already shown that $P(|x|)$ is at most $p(|x|)$. Assume that $|x| = 2^{2^i}$. Then $M$ makes at most $p(2^{2^i})$ queries in Step (2). Since the length of each query is at most $2^{2^{i-1}}$, for each query, $M'$ can make the answer of the oracle in time at most $P(2^{2^{i-1}})$ in Step (3). Hence we have

$$\begin{aligned} P(2^{2^i}) &= p(2^{2^i})P(2^{2^{i-1}}) \\ &= \cdots = p(2^{2^i})p(2^{2^{i-1}})\cdots p(2^{2^0}). \end{aligned}$$

Set $d = \deg p + 1$. Then there exist integers $i_0$ and $C$ such that the following conditions hold:

- $p(n) < n^d$ for any $n > 2^{2^{i_0}}$, and
- $p(n) \leq C$ for any $n \leq 2^{2^{i_0}}$.

If $i \leq i_0$ and $n = 2^{2^i}$, then we have

$$P(n) \leq \prod_{j=0}^{i} p(2^{2^j}) \leq \prod_{j=0}^{i_0} p(2^{2^j}) = C^{i_0+1}.$$

Assume $i > i_0$ and $n = 2^{2^i}$. Noting that $2^{2^{i-k}} = (2^{2^i})^{1/2^k} = n^{1/2^k}$, we have

$$\begin{aligned} P(n) &\leq \prod_{j=0}^{i_0} p(2^{2^j}) \prod_{j=i_0+1}^{i} p(2^{2^j}) \\ &\leq C^{i_0+1} \prod_{j=0}^{i-i_0+1} p(n^{1/2^j}) \\ &\leq C^{i_0+1} \prod_{j=0}^{i-i_0+1} n^{d/2^j} \leq C^{i_0+1} \prod_{j=0}^{\infty} n^{d/2^j} \\ &= C^{i_0+1} n^{\sum_{j=0}^{\infty} d/2^j} = C^{i_0+1} n^{2d}. \end{aligned}$$

Hence, we see that $P(n) \leq C^{i_0+1} n^{2d}$ for any $n$, proving the statement (S-2).

## 5. Proof of Proposition 3

We first show that $\mathsf{NPMV}$ is closed under $\leq^{\mathrm{p}}_{\mathrm{spar}}$-reducibility. For functions $f$ and $g$, we assume that $f \leq^{\mathrm{p}}_{\mathrm{spar}} g$ and $g \in \mathsf{NPMV}$. Then there exists a function $\psi \in \mathsf{PF}$ which satisfies the conditions stated in Definition 2.5. Let $M_g$ be a polynomial-time nondeterministic Turing transducer which computes $g$. Using $M_g$, we construct a Turing transducer $M_f$ as follows: on any input $x$,

(1) Compute $\psi(x)$.
(2) Simulate $M_g$ on the input $\psi(x)$.
(3) If $M_g$ halts with an output $y$, then output $y$ and halt. Otherwise, output $\perp$ and halt.

It follows from the definitions of $\psi$ and $M_g$ that $M_f$ is a polynomial-time nondeterministic Turing transducer. Assume that $M_f(x) \mapsto y$. Then there exists a computation in $M_g$ such that $M_g(\psi(x)) \mapsto y$ by the construction of $M_f$. Since $f \leq^{\mathrm{p}}_{\mathrm{spar}} g$, we have $y \in$ set-$g(\psi(x)) =$ set-$f(x)$. On the other hand, let $y$ be any element of set-$f(x)$. Since $f \leq^{\mathrm{p}}_{\mathrm{spar}} g$, we have $y \in$ set-$f(x) =$ set-$g(\psi(x))$. So there exists a computation in $M_g$ such that $M_g(\psi(x)) \mapsto y$, which implies that $M_f(x) \mapsto y$. Hence, $M_f$ computes $f$, and $f \in \mathsf{NPMV}$ follows.

We next show that $\mathsf{NPMV_g}$ is closed under $\leq^{\mathrm{p}}_{\mathrm{spar}}$-reducibility. For functions $f$ and $g$, we assume that $f \leq^{\mathrm{p}}_{\mathrm{spar}} g$ and $g \in \mathsf{NPMV_g}$. Since we have already seen $f \in \mathsf{NPMV}$ by the argument above, it suffices to prove only graph $f \in \mathsf{P}$. Since $f \leq^{\mathrm{p}}_{\mathrm{spar}} g$, set-$f(x)$ = set-$g(\psi(x))$ holds for any $x \in \mathrm{dom}\, f$. Hence, we have

$$\begin{aligned}
&(x, y) \in \mathrm{graph}\, f \\
&\qquad \Longleftrightarrow x \in \mathrm{dom}\, f \text{ and } y \in \text{set-}f(x) \\
&\qquad \Longleftrightarrow \psi(x) \in \mathrm{dom}\, g \text{ and } y \in \text{set-}g(\psi(x)) \\
&\qquad \Longleftrightarrow (\psi(x), y) \in \mathrm{graph}\, g.
\end{aligned}$$

Noting that $\psi \in \mathsf{PF}$, we see that the language graph $f$ is many-one reducible to the language graph $g$. Since graph $g \in \mathsf{P}$, we have graph $f \in \mathsf{P}$. This completes the proof.

**Remark :** One can similarly show that $\mathsf{NPMV}$ is closed under $\leq^{\mathrm{p}}_{\mathrm{sm}}$ and $\leq^{\mathrm{p}}_{\mathrm{smet}}$-reducibilities. However, it is not known whether $\mathsf{NPMV_g}$ is closed under these two reducibilities: Assume that $f \leq^{\mathrm{p}}_{\mathrm{sm}} g$ and $g \in \mathsf{NPMV_g}$, and let $\psi, \varphi \in \mathsf{PF}$ be functions which satisfy the conditions (i) and (ii) of Definition 2.4. We have

$$\begin{aligned}
&(x, z) \in \mathrm{graph}\, f \\
&\qquad \Longleftrightarrow \psi(x) \in \mathrm{dom}\, g \text{ and } z = \varphi(y) \\
&\qquad\qquad \text{for some } y \in \text{set-}g(\psi(x)) \qquad (2) \\
&\qquad \Longleftrightarrow (\psi(x), y) \in \mathrm{graph}\, g.
\end{aligned}$$

In order to prove that graph $f \in \mathsf{P}$, it suffices to show that one can efficiently find the string $y$ stated in (2). However, it is not straightforward to show whether the statement holds. The similar argument applies to $\leq^{\mathrm{p}}_{\mathrm{smet}}$-reducibility.

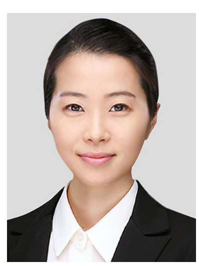
We assume that $f \leq^{\mathrm{p}}_{\mathrm{par}} g$ and $g \in \mathsf{NPMV}$. Then the Turing transducer $M_f$ constructed in the proof above computes some refinement of $f$ since set-$g(\psi(x)) \subseteq$ set-$f(x)$. However, one does not know whether $M_f$ can output all strings of set-$f(x)$ on the input $x$. Hence, one can only prove $f \in_c \mathsf{NPMV}$ on the assumptions that $f \leq^{\mathrm{p}}_{\mathrm{par}} g$ and $g \in \mathsf{NPMV}$. This statement is true if we replace $\leq^{\mathrm{p}}_{\mathrm{par}}$ with $\leq^{\mathrm{p}}_{\mathrm{m}}$ or $\leq^{\mathrm{p}}_{\mathrm{met}}$.

## 6. Concluding Remarks

In this paper, we have extended a part of the results of Faliszewski and Ogihara [4] to cover the classes $\mathsf{NPMV}$ and $\mathsf{NPMV_g}$. We have shown that if any parsimonious complete function for $\mathsf{NPMV}$ (or $\mathsf{NPMV_g}$) is length-decreasing self-reducible, then any function in $\mathsf{NPMV}$ (or $\mathsf{NPMV_g}$) has a refinement which is polynomial-time computable (Theorem 1). We have also shown that Theorem 1 still holds when the term "parsimonious" is replaced with "many-one" or "metric many-one" (Corollary 4). Our result means that there exists an $\mathsf{NPMV}$ (or $\mathsf{NPMV_g}$)-complete function which is not length-decreasing self-reducible unless $\mathsf{P} = \mathsf{NP}$ (Corollary 5).

### References


[1] J.L. Balcázar, "Self-reducibility," J. Comput. Syst. Sci., vol.41, pp.367–388, 1990.
[2] O. Beyersdorff, J. Köbler, and J. Messner, "Nondeterministic functions and the existence of optimal proof systems," Theoret. Comput. Sci., vol.410, pp.3839–3855, 2009.
[3] H. Buhrman and L. Torenvliet, "P-selective self-reducible sets: A new characterization of P," J. Comput. Syst. Sci., vol.53, pp.210–217, 1996.
[4] P. Faliszewski and M. Ogihara, "On the autoreducibility of functions," Theory Comput. Syst., vol.46, pp.222–245, 2010.
[5] J. Feigenbaum and L. Fortnow, "Random-self-reducibility of complete sets," SIAM J. Comput., vol.22, pp.994–1005, 1993.
[6] S. Fenner, F. Green, S. Homer, A.L. Selman, T. Thierauf, and H. Vollmer, "Complements of multivalued functions," Chicago J. Theoret. Comput. Sci., vol.3, 1999.
[7] S. Fenner, S. Homer, M. Ogihara and A. Selman, "Oracles that compute values," SIAM J. Comput., vol.26, pp.1043–1065, 1997.
[8] R. Karp and R. Lipton, "Turing machines that take advice," Enseign. Math. (2), vol.28, pp.191–209, 1982.
[9] S. Mahaney, "Sparse complete sets for NP: Solution of a conjecture of Berman and Hartmanis," J. Comput. Syst. Sci., vol.25, pp.130–143, 1982.
[10] A. Meyer and M. Paterson, "With what frequency are apparently intractable problems difficult?," Technical Report MIT/LCS/TM-126, MIT, 1979.
[11] 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.
[12] A.L. Selman, "A taxonomy of complexity classes of functions," J. Comput. Syst. Sci., vol.48, pp.357–381, 1994.


**Ji-Won Huh** received the B.Eng. degree from Tohoku University, Japan, in 2009, and M.S. degree in information science from Graduate School of Information Sciences, Tohoku University, Japan, in 2011. She is presently with Korea Exchange Bank. Her research interests include information security theory, computational complexity theory and discrete mathematics.

**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.

**Hiroki Shizuya** was born in Sendai, Japan, in September 1957. He received the B.E., M.E., and Dr. Eng. degrees from Tohoku University, Japan, in 1981, 1984, and 1987, respectively. He joined Tohoku University in 1987, and has been Professor since 1995. He is currently with both Center for Information Technology in Education and Department of Computer and Mathematical Sciences, Graduate School of Information Sciences. His current interests are on Cryptology and Computational Complexity Theory. Dr. Shizuya is a member of ACM, IACR and IEEE.