数理解析研究所講究録
第 818 巻 1993 年 99-110

A NOTE ON A POLYNOMIAL TIME REDUCIBILITY

JUICHI SHINODA
篠田寿一
(名古屋大学・人間情報学研究科)

§1. INTRODUCTION

In [2], G. L. Miller studied a version of polynomial time reducibility on the functions which have syntactic polynomial growth, and proved under the Extended Riemann Hypothesis (ERH) that some number theoretic functions such as the Euler function and the Carmichael function are equivalent to the prime factorization with regard to this reducibility.

Let $\Sigma = \{0, 1\}$ and Σ^* denote the set of finite strings on Σ. For an element x of Σ^*, $|x|$ denotes the length of x. We sometimes identify the natural numbers with the elements of Σ^*. A function $f : \Sigma^* \to \Sigma^*$ has *syntactic polynomial growth* if there is a polynomial $p(t)$ such that $|f(x)| \le p(|x|)$ for all $x \in \Sigma^*$. Throghout this note, $x, y, z, \dots$ will denote elments of Σ^* and $f, g, h, \dots$ functions with syntactic polynomial growth. Following [2], f is said to be *polynomial time reducible* (p-reducible) to g, write $f \le_p g$, if there is a polynomial time computable function $\Phi : \Sigma^* \times \Sigma^* \to \Sigma^*$ such that $f(x) = \Phi(x, g(x))$ for all $x \in \Sigma^*$. It is easy to see that this reducibility is reflexive and transitive, and therefore we can define an equivalence relation $\equiv_p$ by

$$f \equiv_p g \iff f \le_p g \ \& \ g \le_p f.$$

The *p-degree* of f is the equivalence class of f and denoted by $\deg_p(f)$. $f <_p g$ iff $f \le_p g$ and $g \not\le_p f$.

Suppose n has the prime factorization $p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$, which we denote by $g(n)$. The Euler function $\varphi(n)$ and the Carmichael function $\lambda(n)$ are p-reducible to g

since they are given by

$$\varphi(n) = p_1^{\alpha_1-1}p_2^{\alpha_2-1}\cdots p_k^{\alpha_k-1}(p_1-1)(p_2-1)\cdots(p_k-1),$$
$$\lambda(n) = \mathrm{lcm}\{p_1^{\alpha_1-1}(p_1-1),\dots,p_k^{\alpha_k-1}(p_k-1)\}.$$

G. L. Miller [2] has shown that the converse holds if we assume ERH. He defined an auxilially function $\lambda'(n)$ by

$$\lambda'(n) = \mathrm{lcm}\{p_1-1,\dots,p_k-1\},$$

and proved assuming ERH that if f is a function with syntactic polynomial growth and for all n $\lambda'(n)$ devides $f(n)$ then $g \le_p f$, which implies φ, λ and λ' are all p-equivalent to g, the prime factorization.

This note is concerned with the structure of the p-degrees of functions with syntactic polynomial growth. In §2, we shall study the basic properties of the reducibility $\le_p$. In §3, we prove the existence of minimal pairs: given $f \not\equiv_p 0$, there is a g such that f and g form a minimal pair. In §4, the density of the p-degrees of low functions are proved: if f and g are low and $f <_p g$, then there is an h such that $f <_p h <_p g$.

§2. Basic properties

Thoroughout this note, let $\{\Phi_e(x,y)\}_{e\in\mathbf{N}}$ be a fixed recusive enumeration of the polynomial time computable functions of two variables. Thus, $f \le_p g$ iff there is an e such that $f(x) = \Phi_e(x, g(x))$ for all x.

The p-reducibility is a special case of the polynomial time 1-*tt* reducibility $\le^p_{1\text{-}tt}$, where $f \le^p_{1\text{-}tt} g$ iff there are polynomial time computable functions $\Phi(x,y)$ and $\varphi(x)$ such that

$$f(x) = \Phi(x, g(\varphi(x))) \quad \text{for all } x \in \Sigma^*.$$

First we remark that the p-reducibility is strictly stronger than the polynomial time 1-*tt* reducibility.

Proposition 2.1. *There are recursive functions f and g such that $f \le^p_{1\text{-}tt} g$ but $f \not\le_p g$.*

Proof. We identify Σ^* with $\mathbf{N}$ as usual. Define g by recursion as follows.

$$\begin{cases} g(0) = 0, \\ g(2x+1) = 0, \\ g(2x+2) = \Phi_x(x+1, g(x+1)) + 1. \end{cases}$$

Define f by $f(x) = g(2x)$. Then, obviously $f \leq^p_{1\text{-}tt} g$. However, by definition, for all x we have

$$f(x+1) = g(2x+2) \neq \Phi_x(x+1, g(x+1)),$$

which implies $f \not\leq_p g$. $\square$

Definition 2.2. $(f \oplus g)(x) = \langle f(x), g(x) \rangle$.

The following lemma yields that the p-degrees form an upper semi-lattice.

Lemma 2.3. (1) *$f \leq_p f \oplus g$ and $g \leq_p f \oplus g$.*
(2) *if $f \leq_p h$ and $g \leq_p h$, then $f \oplus g \leq_p h$.*

Proof. (1) $f(x) = ((f \oplus g)(x))_0$ and $g(x) = ((f \oplus g)(x))_1$. (2) Suppose $f(x) = \Phi(x, h(x))$ and $g(x) = \Psi(x, h(x))$ where $\Phi(x, y)$ and $\Psi(x, y)$ are polynomial time computable functions. Define $\Theta(x, y)$ by

$$\Theta(x, y) = \langle \Phi(x, y), \Psi(x, y) \rangle.$$

Then, $\Theta(x, y)$ is also polynomial computable, and $(f \oplus g)(x) = \Theta(x, h(x))$. $\square$

Proposition 2.4. *For every f, there is a function g recursive in f such that $f <_p g$.*

Proof. Define $h(x) = \Phi_x(x, f(x)) + 1$. Then, $h \not\leq_p f$. Let $g = f \oplus h$. It is easy to see that g has the desired property. $\square$

Proposition 2.5. *Given $f \not\equiv_p 0$, there is a function g recursive in f such that f and g are incomparable with regard to $\leq_p$.*

Proof. g is constructed by simple diagonalization. We require g to satisfy

(R_{2e}) $\qquad g(x) \neq \Phi_e(x, f(x))$ for some x,

and

(R_{2e+1}) $\qquad f(x) \neq \Phi_e(x, g(x))$ for some x.

Now define g by recursion.

Stage 0. Let $l_0 = 0$.

Stage $2e+1$. Suppose l_{2e} and $g \restriction l_{2e}$ have been already defined where

$$g \restriction l_{2e} = g \restriction \{z \in \Sigma^* : |z| < l_{2e}\}.$$

Let $l_{2e+1} = l_{2e} + 1$ and $g(x) = \Phi_e(x, f(x)) + 1$ for all x with $|x| = l_{2e}$. Then, the requirement R_{2e} is obviously satisfied.

Stage $2e+2$. Suppose l_{2e+1} and $g \restriction l_{2e+1}$ are given. Since $f \not\equiv_p 0$, there is an $x \in \Sigma^*$ such that $l_{2e+1} \leq |x|$ and $f(x) \neq \Phi_e(x, 0)$. Take the least such x and set $l_{2e+2} = |x| + 1$. Define g on $\{z : l_{2e+1} \leq |z| < l_{2e+2}\}$ by setting $g(z) = 0$. Then the requirement R_{2e+1} is satisfied. □

Proposition 2.6. *Given $\{f_n\}_{n \in \mathbf{N}}$ such that $f_n \not\equiv_p 0$ for all n, there exists a function g such that $g \not\equiv_p 0$ and $f_n \not\leq_p g$ for all n.*

Proof. The proof is similar to that of the preceding proposition. The requirements for g are

$$(R_{2e}) \qquad\qquad g(x) \neq \Phi_e(x, 0) \quad \text{for some } x,$$

and

$$(R_{2e+1}) \qquad\qquad f_n(x) \neq \Phi_i(x, g(x)) \quad \text{for some } x,$$

where $e \mapsto (n, i)$ is a (recursive) bijection between $\mathbf{N}$ and $\mathbf{N} \times \mathbf{N}$.

We construct g in stages.

Stage 0. Set $l_0 = 0$.

Stage $2e+1$. Suppose l_{2e} and $g \restriction l_{2e}$ are given. Set $l_{2e+1} = l_{2e} + 1$, and define g on $\{z : |z| = l_{2e}\}$ by

$$f(x) = \Phi_e(x, 0) + 1.$$

Then, R_{2e} is met in this stage.

Stage 2e+2. Suppose l_{2e+1} and $g \restriction l_{2e+1}$ have been already defined. Let (n, i) be the e-th element of $\mathbf{N} \times \mathbf{N}$. Since $f_n \not\equiv_p 0$, there is an x such that $l_{2e+1} \leq |x|$ and $f_n(x) \neq \Phi_i(x, 0)$. Take the least such x and set $l_{2e+2} = |x| + 1$. We extend g to $\{z : |z| < l_{2e+2}\}$ by setting $g(z) = 0$ for all z with $l_{2e+1} \leq |z| < l_{2e+2}$. Then, the requirement R_{2e+1} is satisfied. □

Corollary 2.7. *There exists a non-recursive g such that for every recursive f if $f \not\equiv_p 0$ then $f \not\leq_p g$.*

§3. Minimal pairs

Definition 3.1. f and g form a *minimal pair* if

(i) $f \not\equiv_p 0$ and $g \not\equiv_p 0$,

(ii) for every h, if $h \leq_p f$ and $h \leq_p g$, then $h \equiv_p 0$.

Theorem 3.2. *For every f with $f \not\equiv_p 0$, there is a function g recursive in f such that f and g form a minimal pair.*

Proof. The following are the requirements for g.

$$(R_{2e}) \qquad g(x) \neq \Phi_e(x, 0) \quad \text{for some } x,$$

$$(R_{2e+1}) \qquad (\forall x)[h(x) = \Phi_{e_1}(x, f(x)) = \Phi_{e_2}(x, g(x)] \implies h \equiv_p 0,$$

where $e \mapsto (e_1, e_2)$ is a recursive bijection between $\mathbf{N}$ and $\mathbf{N} \times \mathbf{N}$.

The requirement R_{2e} will be met by simple diagonalization. We only define $g(x)$ to be different from $\Phi_e(x, 0)$. For the requirement R_{2e+1}, first we try to invalidate the equality $\Phi_{e_1}(x, f(x)) = \Phi_{e_2}(x, g(x))$, and if it fails then the function $\Phi_{e_1}(x, f(x))$ will be computed in polynomial time. To accomplish the construction consistently, however, we need a simple priority argument. We say that the requirement R_n is given priority over R_m, or that R_n has higher priority than R_m, if $n < m$.

Definition 3.3.

(1) R_{2e} is satisfied before stage $s + 1$ iff there is an x such that $|x| < s$ and $g(x) \neq \Phi_e(x, 0)$.

(2) R_{2e+1} is satisfied before stage $s + 1$ iff there is an x such that $|x| < s$ and $\Phi_{e_1}(x, f(x)) \neq \Phi_{e_2}(x, g(x))$.

It is easy to see that if R_i is satisfied before stage $s+1$ then it is met, furthermore in the case of $i = 2e + 1$ it is met by invalidating the premise of the requirement.

Definition 3.4. R_{2e} requires attention at stage $s + 1$ iff $e \leq s$ and it is not satisfied before $s + 1$.

Definition 3.5. R_{2e+1} requires attention at stage $s + 1$ iff $e \leq s$ and

(i) it is not satisfied before stage $s + 1$,

(ii) there are x, y such that $|x| = |y| = s$ and that $\Phi_{e_1}(x, f(x)) \neq \Phi_{e_2}(x, y)$.

With these definitions we give a detail of the construction of g.

Stage 0. Do nothing.

Stage $s+1$. Suppose $g \restriction \{z : |z| < s\}$ has been already defined. At this stage, we extend g on $\{z : |z| \leq s\}$. If no requirement requires attention, then we simply set $g(z) = 0$ for all z with $|z| = s$. Otherwise take the requirement R_i $(i \leq s)$ with highest priority which requires attention. We attack the requirement R_i in this stage. If $i = 2e$, then for all z with $|z| = s$ we define $g(z)$ to be different from the value $\Phi_e(z, 0)$. Then, it is easy to see that R_{2e} is met at this stage. Suppose $i = 2e+1$. Let x_0, y_0 be the least x, y such that $|x| = |y| = s$ and $\Phi_{e_1}(x, f(x)) \neq \Phi_{e_2}(x, y)$. Then, we extend g on $\{z : |z| \leq s\}$ by setting $g(z) = y_0$ for all z with $|z| = s$. Thus, the requirement R_{2e+1} is satisfied. This completes the construction.

Lemma 3.6. *Each requirement requires attention only finitely often.*

Proof. Suppose lemma is proved for all $j < i$. Take a sufficiently large s_0 so that any of R_j $(j < i)$ does not require attention at any stage after s_0. If R_i requires attention at some stage $s+1 > s_0$, then it must be attacked and does not require attention any more. $\square$

Lemman 3.7. *Every requirement is met.*

Proof. By the previous lemma, there is an s_0 such that any of the requirements R_j $(j \leq i)$ does not require attention after s_0. If R_i is satisfied at some stage, then it must be met. So, suppose it is never satisfied. Since any even requirement is eventually satisfied, i must be $2e+1$ for some e. Suppose $s+1 > s_0$. Since R_{2e+1} does not require attention at $s+1$, we have

$$(\forall x, y)[|x| = |y| = s \longrightarrow \Phi_{e_1}(x, f(x)) = \Phi_{e_2}(x, y)].$$

Therefore, we see that

$$(\forall x)[|x| \geq s_0 \longrightarrow \Phi_{e_1}(x, f(x)) = \Phi_{e_2}(x, 0^{|x|})],$$

which implies that the function $x \mapsto \Phi_{e_1}(x, f(x))$ is computable in polynomial time. $\square$

§4. Density

Ladner [1] applied delayed diagonalizations first in the proofs of the density and splitting theorems for the polynomial time Turing degrees (p-T degrees). It is not difficult to apply his method to the p-degrees of recursive functions.

Theorem 4.1. *Given recursive f, g such that $f <_p g$, there is an h such that $f <_p h <_p g$.*

Proof. We require h to satisfy the following.

$$(R_{2e}) \qquad h(x) \neq \Phi_e(x, f(x)) \quad \text{for some } x,$$

and

$$(R_{2e+1}) \qquad g(x) \neq \Phi_e(x, h(x)) \quad \text{for some } x.$$

We will construct h so that $h(x)$ agrees with $\langle f(x), g(x)\rangle$ on some long interval $\{x : l_{2e} \leq |x| < l_{2e+1}\}$ in which there is an x witnessing the requirement R_{2e}, and likewise agrees with $\langle f(x), 0\rangle$ on the next long interval $\{x : l_{2e+1} \leq |x| < l_{2e+2}\}$, in which there is an x witnessing the requirement R_{2e+1}. To ensure that $f \leq_p h \leq_p g$, some delay will be put before changing stages. Now we give the detail of the construction.

Stage 0. We set $l_0 = 0$.

Stage $2e+1$. Suppose l_{2e} is given. Since $f \oplus g \not\leq_p f$, there is an x such that $l_{2e} \leq |x|$ and $\langle f(x), g(x)\rangle \neq \Phi_e(x, f(x))$. We find the least such x by successively computing $f(0^{l_{2e}})$, $g(0^{l_{2e}})$, $\Phi_e(0^{l_{2e}}, f(0^{l_{2e}}))$; $f(0^{l_{2e}-1}1)$, $g(0^{l_{2e}-1}1)$, $\Phi_e(0^{l_{2e}-1}1, f(0^{l_{2e}-1}1))$; ... until we encounter the first x such that

$$\langle f(x), g(x)\rangle \neq \Phi_e(x, f(x)).$$

Let m be the number of steps needed to accomplish these computations. We set $l_{2e+1} = l_{2e} + m$.

Stage $2e+2$. Suppose l_{2e+1} has been already defined. We search for the first x such that $l_{2e+1} \leq |x|$ and $g(x) \neq \Phi_e(x, \langle f(x), 0\rangle)$. Since $g \not\leq_p f \oplus 0$, such an x exists. The definition of l_{2e+2} is similar to the previous stage. Namely, l_{2e+2} is l_{2e+1} plus the number of steps needed to find the first x which satisfies the inequality $g(x) \neq \Phi_e(x, \langle f(x), 0\rangle)$.

We define h by

$$h(x) = \begin{cases} \langle f(x), g(x) \rangle & \text{if } l_{2e} \leq |x| < l_{2e+1}, \\ \langle f(x), 0 \rangle & \text{if } l_{2e+1} \leq |x| < l_{2e+2}. \end{cases}$$

Then, h satisfies all the requirements R_{2e} and R_{2e+1}, and therefore we obtain $h \not\leq_p f$ and $g \not\leq_p h$. It is clear that $f \leq_p h$ since $f(x) = (h(x))_0$ for all x. To see that $h \leq_p f \oplus g$, suppose x is given. We can find an n such that $l_n \leq |x| < l_{n+1}$ by performing the construction of the sequence $\{l_n\}_n$ in $|x|$ steps. If $n = 2e$ for some e, then $h(x) = \langle f(x), g(x) \rangle$; if $n = 2e + 1$ for some e, then $h(x) = \langle f(x), 0 \rangle$. Thus, $h(x)$ is calculated from x and $(f \oplus g)(x)$ in polynomial time of $|x|$. $\square$

For the non-recursive functions, it is not known whether the density theorem holds or not. At present, we can prove that if f and g are low then Theorem 4.1 holds, where f is said to be low if the Turing jump of f has the same Turing degree as $0'$, *i.e.*, $f' \equiv_T 0'$.

Lemma 4.2. (Limit Lemma [3]). *If f is recursive in $0'$, then there is a recursive sequence $\{f_s\}_{s \in \mathrm{N}}$ such that*

$$\lim_{s \to \infty} f_s(x) = f(x) \quad \textit{for all } x.$$

Theorem 4.3. *If f and g are low and $f <_p g$, then there is an h such that $f <_p h <_p g$.*

Proof. Suppose f and g are low. By the limit lemma, there are recursive sequences $\{f_s\}_s$ and $\{g_s\}_s$ such that

$$\lim_{s \to \infty} f_s(x) = f(x) \qquad \text{and} \qquad \lim_{s \to \infty} g_s(x) = g(x).$$

Let $U = \{e : (\exists \langle x, y, z \rangle \in W_e)[f(x) = y \ \ \& \ \ g(x) = z]\}$ where W_e is the e-th recusively enumerable set. Then, U is recursively enumerable in g, and hence is recursive in $0'$ since g is low. By the limit lemma, there is a recursive sequence $\{u_s\}_s$ such that $u_s(e) \leq 1$ and $\lim_s u_s(e) = U(e)$ for all e. We define h as in the proof of Theorem 4.1:

$$h(x) = \begin{cases} \langle f(x), g(x) \rangle & \text{if } l_{2e} \leq |x| < l_{2e+1}, \\ \langle f(x), 0 \rangle & \text{if } l_{2e+1} \leq |x| < l_{2e+2}. \end{cases}$$

The increasing sequence $\{l_n\}_n$ will be so constructed that h satisfies the same requirements R_{2e} and R_{2e+1} in the proof of Theorem 4.1. Further, we will build a recursive sequence $\{V_{i,s}\}_{i,s\in\mathbf{N}}$ during the construction. Let $V_i = \bigcup_s V_{i,s}$. Then, V_i is recursive enumerable. By the recursion theorem we may assume that we have in advance an index of V_i with some recursive function θ, *i.e.*, $V_i = W_{\theta(i)}$.

Definition 4.4. Suppose i and s are given. The requirement R_i is *U-certified* at s if $u_s(\theta(i)) = 1$ and there is a $\langle x, y, z\rangle \in V_{i,s}$ such that $f_s(x) = y$ and $g_s(x) = z$.

Now, we give the construction of $\{l_n\}_n$. In the construction, no elements are enumerated in V_i unless explicitly mentioned.

Stage 0. Set $l_0 := 0$.

Stage $2e+1$. Take the least $i \le e$ such tha R_{2i} is not U-certified at l_{2e}. We say that R_{2i} is *attacked.* Our construction in this stage consists of one main routine with 3 subroutines.

Main routine. We set $s := l_{2e}$. Go to Subroutine 1.

Subroutine 1. Suppose the construction enters this routine with s.

While true do

If there exists an x such that

(i) $l_{2e} \le |x| \le s$ and

(ii) $\langle f_s(x), g_s(x)\rangle \ne \Phi_i(x, f_s(x))$,

Then take the least such x and

$y := f_s(x)$,

$z := g_s(x)$,

$V_{2i,s+1} := V_{2i,s} \cup \{\langle x, y, z\rangle\}$,

$s := s + 1$,

Exit from Subroutine 1 and go to Subroutine 2;

Else $s := s + 1$;

End if

End while;

End Subroutine 1.

The following claim ensures that we eventually exit from the while-loop and enters into Subroutine 2.

Claim. *Given s, there is a $t \geq s$ and x such that $l_{2e} \leq |x| \leq t$ and $\langle f_t(x), g_t(x)\rangle \neq \Phi_i(x, f_t(x))$.*

Proof. Since $f \oplus g \not\leq_p f$, there exists an x such that $l_{2e} \leq |x|$ and $\langle f(x), g(x)\rangle \neq \Phi_i(x, f(x))$. Take a sufficiently large t so that $t \geq \max\{s, |x|\}$, $f_t(x) = f(x)$ and $g_t(x) = g(x)$. $\square$

Subroutine 2. Suppose the construction enters this routine with s.

> **While** true do
>> **If** R_{2i} is U-certified at s,
>>> **Then** exit from Subroutine 2 and go to Subroutine 3;
>>> **Else**
>>>> **If** $u_s(\theta(2i)) = 0$ and for all $\langle x, y, z\rangle \in V_{2i,s}$, either $f_s(x) \neq y$ or $g_s(x) \neq z$,
>>>>> **Then** exit from Subroutine 2 and go to Subroutine 1;
>>>>> **Else** $s := s + 1$;
>>>> **End** if;
>>> **End** else
>> **End** if;
> **End** while;

End Subroutine 2;

Claim. *Given s, suppose $V_{2i,s} = V_{2i,t}$ for all $t \geq s$. Then, there is a $t \geq s$ such that either*

(1) *R_{2i} is U-certified at t or*
(2) *$u_t(\theta(2i)) = 0$, and for all $\langle x, y, z\rangle \in V_{2i,t}$ either $f_t(x) \neq y$ or $g_t(x) \neq z$.*

Proof. Take a sufficiently large $s_0 \geq s$ so that

$$(\forall t \geq s_0)\left[\begin{array}{c} u_t(\theta(2i)) = U(\theta(2i)) \quad \text{and} \\ (\forall\langle x, y, z\rangle \in V_{2i,s})[f_t(x) = f(x) \;\;\&\;\; g_t(x) = g(x)]\end{array}\right].$$

Such an s_0 exists since $V_{2i,s}$ is finite. If $U(\theta(2i)) = 1$, then (1) holds for all $t \geq s_0$. If $U(\theta(2i)) = 0$, then (2) holds for all $t \geq s_0$. $\square$

Subroutine 3. Suppose the construction enters this routine with s. Let l_{2e+1} be l_{2e} plus the number of steps performed so far, and exit from the main routine.

Claim. *l_{2e+1} is defined.*

Proof. Suppose not. Then we always exit from Subroutine 2 with

$$(*) \qquad (\forall \langle x, y, z\rangle \in V_{2i,s})[f_s(x) \neq y \vee g_s(x) \neq z]$$

and enters Subroutine 1. Since $f \oplus g \not\leq_p f$, there is an x with $l_{2e} \leq |x|$ such that $\langle f(x), g(x)\rangle \neq \Phi_i(x, f(x))$. Let $y = f(x)$ and $z = g(x)$. Take a sufficiently large s_0 so that

$$(\forall s \geq s_0)[f_s(x) = f(x) \ \ \& \ \ g_s(x) = g(x)].$$

We may assume that $|x| \leq s_0$. If $\langle x, y, z\rangle$ is not enumerated into V_{2i} up to s_0, then $\langle x, y, z\rangle$ is witnessed each time Subroutine 1 is executed after s_0. Therefore, if we enter Subroutine 1 infinitly often, then $\langle x, y, z\rangle$ must be enumerated into V_{2i}, and thus $U(\theta(2i)) = 1$ by definition, which contradicts (*). □

Stage $2e+2$. Similar to Stage $2e+1$. Take the least $i \leq e$ such that R_{2i+1} is not U-certified at l_{2e+1}. The requirement R_{2i+1} is attacked in this stage. In Subroutine 1, we search for $s \geq l_{2e+1}$ and x such that $l_{2e+1} \leq |x| \leq s$ and $g_s(x) \neq \Phi_i(x, \langle f_s(x), 0\rangle)$, and enumerate $\langle x, y, z\rangle$ into V_{2i+1} where $y = f_s(x)$ and $z = g_s(x)$, then goto Subroutine 2. Other subroutines are defined similarly. We leave the detail to the reader.

This completes the construction. We will show that $\{l_n\}_n$ and h so constructed satisfy the conditions of the theorem.

Lemma 4.5. *For each i, the requirement R_i is attacked only finitely often.*

Proof. We show the lemma for R_{2i}. Suppose R_{2i} is attacked infinitely often. Let e be arbitrary and suppose R_{2i} is attacked at stage $2e+1$. Then, since R_{2i} is U-certified during stage $2e+1$, there is an s such that $l_{2e} \leq s < l_{2e+1}$ and $u_s(\theta(2i)) = 1$. Therefore, $\{s : u_s(\theta(2i)) = 1\}$ is infinite and hence we must have $U(\theta(2i)) = 1$. Thus, by the definition of U, there is a $\langle x, y, z\rangle \in V_{2i}$ such that $f(x) = y$ and $g(x) = z$. Take a sufficiently large s_0 so that

$$(\forall s \geq s_0)[u_s(\theta(2i)) = 1 \ \ \& \ \ \langle x, y, z\rangle \in V_{2i,s} \ \ \& \ \ f_s(x) = f(x) \ \ \& \ \ g_s(x) = g(x)].$$

Then, R_{2i} is U-certified at any s after s_0, which is a contradiction. □

Lemma 4.6. *Every requirement R_i is satisfied.*

Proof. We prove this for R_{2i}. Take a sufficiently large n_0 so that no requirements R_{2j} $(j \leq i)$ are attacked after any stage after n_0. First we show that $U(\theta(2i)) = 1$. If $U(\theta(2i)) = 0$, then there is an s_0 such that for all $s \geq s_0$,

$$(\forall \langle x, y, z\rangle \in V_{2i,s})[f_s(x) \neq y \vee g_s(x) \neq z],$$

which implies the requirement R_{2i} is not U-certified at any s with $s \geq s_0$, and therefore R_{2i} must be attacked at any stage $2e+1 \geq n_0$ with $l_{2e} \geq s_0$, a contradiction. Since $U(\theta(2i)) = 1$, there is a $\langle x, y, z\rangle \in V_{2i}$ such that $f(x) = y$ and $g(x) = z$. Suppose $\langle x, y, z\rangle \in V_{2i,s+1} - V_{2i,s}$. Then, by the construction, we have $y = f_s(x)$, $z = g_s(x)$ and $\langle f_s(x), g_s(x)\rangle \neq \Phi_i(x, f_s(x))$. It follows that $h(x) = \langle f(x), g(x)\rangle \neq \Phi_i(x, f(x))$, and thus the requirement R_{2i} is satisfied. □

Given x, performing the construction of $\{l_n\}_n$ in $|x|$ steps, we can calculate the unique n such that $l_n \leq |x| < l_{n+1}$. Then, we can calculate $h(x)$ from x and $(f \oplus g)(x)$ as before in polynomial time of $|x|$, and we obtain $h \leq_p f \oplus g$. The requirements R_{2e} and R_{2e+1} ensures that $h \not\leq_p f$ and $g \not\leq_p h$. This completes the proof of Theorem 4.3. □

References

1. R. E. Ladner, *On the structure of polynomial time reducibilities*, J. Assoc. Comput. Mech. **22** (1975), 155–171.
2. G. L. Miller, *Riemann's hypothesis and tests for primarity*, J. Computer and System Sciences **13** (1976), 300–317.
3. R. I. Soare, *Recursively Enumerable Sets and Degrees*, Springer-Verlag, New York, 1987.