Scientiae Mathematicae Japonicae Online, **Vol. 5**, (2001), 405–419

COMPUTABILITY ASPECTS OF SOME DISCONTINUOUS FUNCTIONS

Mariko Yasugi *, Vasco Brattka† and Masako Washihara‡

Received August 28 2001; revised October 9, 2001

Abstract. We speculate on computational aspects of certain discontinuous functions by taking the Gaußian function $[x]$ as a typical example. An algorithm how to compute $[x]$ for a single computable real number is first described, followed by a remark that $[x]$ does *not* necessarily preserve sequential computability. Second, $[x]$ is studied in the light of the notions of *upper semi-computability* and of *limiting computation.* Then two Fréchet spaces, $\mathbf{R}^{\mathbf{Z}}$ and $L^1_{loc}(\mathbf{R})$, in which some discontinuous functions will become computable, will be taken up.

1 Introduction The computability of real functions was originally defined for continuous functions, which may be called Grzegorczyk computability (See Chapter 0 in[13], for example). A real function f defined on the compact interval $[0,1]$ is called Grzegorczyk-computable if, for any computable sequence of real numbers $\{x_n\}$, $\{f(x_n)\}$ is a computable sequence of real numbers (f preserves the sequential computability) and if f has a recursive modulus of (uniform) continuity. This is reasonable with continuous functions since one can evaluate their values at some dense points, such as rational points, and then, with these values, one can approximate the value at any point to an arbitrary precision.

If the values at these dense points are computable and the function is smooth enough, then one can expect that the modulus of continuity can be taken to be recursive. In other words, such a function is characterized by some representing values "effectively."

On the other hand, one often deals with a function which jumps at some points such as at integer points and which is continuous on each open interval between two jump points. One would compute the values of such a function at some points which supply information for computation and then approximate values at other points.

It is thus a natural attitude to investigate ways of expressing such computations. There can be various approaches to attain this purpose, some of which will be listed as references. Our interest lies in mathematical treatment rather than foundational.

In [13], Pour-El and Richards proposed to call a function computable if it is nicely approximated by a sequence of rational coefficient polynomials with respect to the norm of a Banach space. Then a function which jumps at some points but is continuous on their complement can be regarded as computable. This was made possible by employing a norm of a function which is not necessarily the maximum norm. On the other hand, in such a theory, a function is regarded computable only as a point in a function space, while we are more interested in computing the *values* of a function.

As a sample of our concern, we will discuss some computability aspects of the Gaußian function $[x]$ (the integer part function). It jumps up at integers, but is continuous in each open interval between two consecutive integers. Although it is an innocent, anti-dramatic function, it is a typical and a simplest function which fits into the category of our concern,

2000 *Mathematics Subject Classification.* 03F60.
Key words and phrases. Discontinuous function, Computability, Gaußian function, Upper semi-computability, Limiting recursive functional, Fréchet space, Computability structure.
*This work is partly supported by Scientific Grant-in-Aid (No.12440031): the corresponding author.

and so is expected to be appropriate to distill the essence of the problem. Due to its simplicity, it does not obstruct us with irrelevant details.

The function $[x]$ can be regarded as a computable point in two Fréchet spaces (the space of integer-indexed sequence of real numbers and the space of locally integrable functions). It can also be characterized as having computational algorithms if some kind of non-recursive principles are admitted. One such notion of algorithm is the upper semi-computability, that is, the value of a function at any point can be approximated from above by a sequence of values, computed from the information on x. Another is the notion of computation in the limit, that is, the value of a function at a point can be computed in terms of some limiting recursive functionals.

In Section 2, we will first demonstrate an attempt to compute $[x]$ for a computable real number x from the information of x as follows.

Let $\{r_l\}$ be a computable sequence of rationals converging to x effectively. Then we find an integer k such that $k < x < k+2$. We will then define a computable sequence of rationals (in fact integers) $\{N_p\}$ which has the following property. $N_p = k+1$ as long as a recursive condition $R(p)$ does not hold: $N_p = k$ once $R(q)$ is attained for a $q \leq p$. There are two cases: (i) $\forall p \neg R(p)$, in which case $N_p = k+1$ for all p, and (ii) $\exists p R(p)$, in which case $N_q = k$ for all $q \geq p$, where p is the first step for which $R(p)$ holds. If we write $\lim_p N_p$ for the number N such that $\exists p \forall q \geq p(N_q = N_p = N)$, then it can be classically established that $\lim N_p = [x]$ holds, and the convergence is effective. *Only it cannot be decided* which of (i) or (ii) indeed holds.

We will view such a situation from two standpoints: *upper semi-computability* (Section 3) in the line of [20] and [6], and *limiting computation* (Section 4). The latter can be defined in terms of the notion of *limiting recursive functionals* in [8].

The treatment in Section 4 suggests that upper or lower semi-computable functions can be computed in terms of limiting recursive functionals. Furthermore, there are some discontinuous functions which are neither upper nor lower semi-computable functions but can be computed in terms of limiting recursive functionals. We will take up several examples of such functions. Among them is the system of Rademacher functions, which is important in Walsh analysis. It is shown to be computable with a limiting recursive functional as a sequence of functions.

Formal treatments of the limiting computation and their interpretations are seen, for example, in [1] and [12].

In [16], Washihara formulated the conditions for the computability structure on a Fréchet space (Axioms 1$\sim$3 in Introduction,[16]). The first example of a Fréchet space with computability structure there is $\mathbf{R}^\infty$, the space of real sequences. ((1) in Section 3,[16]). We can extend this space to $\mathbf{R}^\mathbf{Z}$, the space of real sequences with integer indices.

Taking the advantage of the theory of $\mathbf{R}^\mathbf{Z}$, we will embed some step functions into $\mathbf{R}^\mathbf{Z}$, and will show that they are computable in this space (Section 5). This characterizes the computability of the Gaußian function as a computable sequence of values.

On the other hand, computability structures on the Banach spaces $L^p[a,b]$ and $L^p(\mathbf{R})$ are given in 3, Chapter 2 of [13]. Inspired by this, we define a computability structure on the Fréchet space $L^1_{loc}(\mathbf{R})$ (the space of locally integrable real functions) and show that some examples in Section 4 are computable in this space too (Section 5). This means that the Gaußian function is well approximated by a computable sequence of some continuous functions except at the integer points.

These two results indicate that the Fréchet space is a useful framework for investigations of the computability properties of discontinuous functions. For details, see, for example, [14] for the classical theory of the Fréchet space, and [16] for the definition of the computability structure on a Fréchet space.

This work has been developed from a foregoing article [23]. Although we assume the knowledge of basics of the computability structures developed in [13] as well as upper and lower semi-computability in terms of Type 2 Turing machine in [19] and [20], this article is more or less self-contained. [24] will serve as a quick reference to basics of computability structures in analysis.

The computability problem of another famous object, the δ-function, has been studied in [17] by Washihara. A computability structure is defined for the space of tempered distributions, and it is shown that the δ-function is computable relative to that structure.

The relations between computability structures of Fréchet spaces and those of the metrizations of Fréchet spaces have been discussed in [18] and [11]. We could thus regard the Gaußian function as a computable element in a metric space also, but we will not elaborate upon it now.

There are many related works accomplished on computability problems in function spaces and metric spaces. We will list only a few of them, say [4], [21] and [22], in which one can find further important references.

In [2], the Gaußian function is a simplest example of a computable function in their theory. It is possible because they allow judgement of $x \geq y$, while in our theory this is not decidable. The relation between our approach and that of [2] will be an interesting subject to work on.

If one changes the topology of input data, then one can turn the Gaußian function into a continuous function, so that we can apply the traditional notion of computability to this function. See, for example, [9], [10] and [15].

2 An attempt to compute $[x]$ A sequence of *rational numbers*, $\{r_m\}$ is said to be *recursive* if there are recursive functions β, γ, δ satisfying

$$r_m = (-1)^{\beta(m)} \frac{\gamma(m)}{\delta(m)}$$

for all m. A real number x is *computable* if there is a recursive sequence of rationals $\{r_m\}$ and a recursive function α satisfying that for every $p = 0, 1, 2, \cdots$

$$m \geq \alpha(p) \Rightarrow |x - r_m| \leq \frac{1}{2^p}$$

Such an α is called a modulus of convergence (of $\{r_m\}$ to x).

The notion of computable real numbers can be extended to a *computable sequence* of real numbers. See [13] for details.

Program P We will first *attempt* to compute $[x]$ for a computable x.

Let x be a computable real number represented by a computable sequence of rationals $\{r_m\}$ and a modulus of effective convergence α.

By Proposition 0 in Chapter 0 of [13], there is a program such that, for any computable number x, if $x > 0$ indeed holds, then this is recognized by the program.

Using such a program, we will compose a program $\mathbf{P}$ such that, given a pair of information on x, say $\langle \{r_m\}, \alpha \rangle$, where $\{r_m\}$ is a recursive sequence of rational numbers which converges to x with a recursive modulus of convergence α, $\mathbf{P}$ outputs a recursive sequence $\{s_p\}$ which converges to $[x]$.

1 There is a program $\mathbf{P}_1$ which acts as follows.

Input $\langle \{r_m\}, \alpha, k \rangle$, $\mathbf{P}_1$ checks whether $r_{\alpha(p)} < n - \frac{1}{2^p}$ holds for some $p \leq k$ and $n \leq k$. If the result is *Yes*, then $x < n$ has been determined. Similarly with the other direction, checking the inequality $n + \frac{1}{2^p} < r_{\alpha(p)}$. By increasing k, $\mathbf{P}_1$ eventually hits a number n such that $n < x < n + 2$. Then stop the process.

2 There is a program $\mathbf{P}_2$ such that, input $\langle \{r_m\}, \alpha, n \rangle$, where n has been computed by $\mathbf{P}_1$ so that $n < x < n + 2$, $\mathbf{P}_2$ outputs a sequence of integers $\{N_p\}$ as follows.

Compute the inequality $r_{\alpha(p)} < (n+1) - \frac{1}{2^p}$, for $p = 1, 2, \cdots$. This relation is decidable. As long as the answer is *No*, put $N_p = n + 1$. If the answer is *Yes* at stage p, then put $N_p = n$. (Once the answer is *Yes*, then $N_q = n$ for all $q \geq p$.)

3 There is a program $\mathbf{P}_3$ such that, with input $\{N_p\}$, $\mathbf{P}_3$ outputs a recursive sequence of rational numbers $\{s_p\}$. Namely, regarding N_p as a rational number, we put $s_p = N_p$.

Now, the desired program $\mathbf{P}$ is a composition of $\mathbf{P}_1$,$\mathbf{P}_2$ and $\mathbf{P}_3$.

Note It may be confusing that a sequence $\{N_p\}$ is distinguished as a sequence of integers and as a sequence of rational numbers. This is due to the particular nature of the Gaußian function. If we wished to evaluate, for example, the value of $\frac{1}{2}[x]$, then the situation would be clear. That is, $s_p = \frac{1}{2}N_p$. $\{N_p\}$ is used for case distinctions, while $\{s_p\}$ represents a real number.

Either $N_p = n + 1$ for all p (**Case 1**) or there is a p_0 such that $N_p = n + 1$ for $p < p_0$ and $N_p = n$ for all $p \geq p_0$ (**Case 2**).

$\{s_p\}$ is nonincreasing according to the definition and converges to either $n+1$ (**Case 1**) or to n (**Case 2**). Call this limit ξ.

We can easily claim that, in either case, $\xi = [x]$. Notice that $[x]$ is an integer by definition, which is a computable number.

Case 1 $N_p = n + 1$ for all p. This means that it never happens that $r_{\alpha(p)} < (n + 1) - \frac{1}{2^p}$. This implies that $x \geq n + 1$. Since $x < n + 2$, $x \geq n + 1$ then implies that $[x] = n + 1$. So, $\{s_p\}$ converges to $[x]$.

Case 2 $N_p = n$ for some p and hence for all larger q. This means that $r_{\alpha(p)} < (n+1) - \frac{1}{2^p}$, which implies that $x < n + 1$. Since $n < x$, $x < n + 1$ then implies $[x] = n$.

In either case, the monotone recursive sequence of rationals $\{s_p\}$ converges to a computable number, which is in fact $[x]$.

Furthermore, since the limit number is computable, the convergence is effective by Proposition 2 (Monotone Convergence) in Chapter 0 of [13].

How do we find an effective modulus of convergence? Given p, take the difference $s_q - [x]$, $q = 1, 2, 3, \cdots$, and wait until $|s_q - [x]| < \frac{1}{2^p}$ holds. Take the first such q and put $\beta(p) = q$. β will do.

If Case 1 holds, then $\beta_1(p) = 1$ will suffice. If Case 2 holds, then $\beta_2(p) = \mu q(r_{\alpha(q)} < (n + 1) - \frac{1}{2^p})$ will do, where μq is the minimum number operator.

The program $\mathbf{P}$ can be easily extended to the domain $\mathbf{R}$ of all reals numbers.

We can analyze the logical structure of the computation above as follows.

Logical structure The *logical structure* of the preceding argument is the following. Put $R(p) \equiv r_{\alpha(p)} < n - \frac{1}{2^p}$. $R(p)$ is recursive (with respect to p and n). Then, Case 1 is expressed as $\forall p \neg R(p)$, and Case 2 is expressed as $\exists p R(p)$. Classically, $\forall p \neg R(p) \vee \exists p R(p)$ holds. Under each condition, one can construct a modulus of convergence, but we will not know effectively which case holds. It is in general not decidable.

A formula of the form $\forall p \neg R(p) \vee \exists p R(p)$ with a recursive R represents a Σ^0_1-*LEM*, *LEM* denoting *limited excluded middle* according to [12]. We can thus describe the evaluation of $[x]$ above as follows: there is a method of computating $[x]$ which is effective relative to a Σ^0_1 -*LEM*.

Remark 1 Here one uses different types of input and output representations. The input is represented by a pair $(\mathbf{r}, \alpha)$, where $\mathbf{r}$ is an effectively converging Cauchy sequence and α is a recursive modulus of convergence, while the output is represented by a monotonically converging sequence $\mathbf{q}$ of rationals. Although the convergence of the output sequence is effective (since $[x]$ is always an integer and hence computable), the modulus of convergence, called β, cannot be computed from the pair $(\mathbf{r}, \alpha)$ which represents the input. For the computation of β, one needs an a priori knowledge of $[x]$. In other words, the mapping $(\mathbf{r}, \alpha) \rightarrow \beta$ is not computable, although each β is a computable function.

Let us remark that the Gaußian function is *not* sequentially computable. A counter-example is given below. It will imply that the case distinction $\forall p \neg R(p) \vee \exists p R(p)$ is *not* decidable, since otherwise the computation of $[x]$ would be effective.

Counter-example Let a be a recursive injection whose range is not recursive. Let $\{x_n\}$ be the computable sequence of reals in Example 4, Chapter 0 of [13]. That is,

$$x_n = \left\{ \begin{array}{ll} \frac{1}{2^m} & \text{if } n = a(m) \text{ for some } m \\ 0 & \text{otherwise} \end{array} \right\}$$

Consider the computable sequence of reals $\{y_n\}$ where $y_n = 1 - x_n$. Then

$$y_n = \left\{ \begin{array}{ll} 1 - \frac{1}{2^m} & \text{if } n = a(m) \text{ for some } m \\ 1 & \text{otherwise} \end{array} \right\}$$

This implies that

$$[y_n] = \left\{ \begin{array}{ll} 0 & \text{if } n = a(m) \text{ for some } m \\ 1 & \text{otherwise} \end{array} \right\}$$

Now, suppose $\{[y_n]\}$ were a computable sequence. Then, it can be shown, as in Example 4, Chapter 0 of [13], that the range of a would be recursive, yielding a contradiction. So, $\{[y_n]\}$ cannot be a computable sequence.

This counter-example assures us of the following fact: the Gaußian function does not necessarily preserve computability of a sequence of reals.

3 Upper semi-computability In this section, we prove that the Gaußian function is upper semi-computable.

We will use the following notations throughout the rest of this article.
N: the set of natural numbers
Z: the set of integers
R: the set of real numbers

Intuitively, we will call a function $f : \mathbf{R} \rightarrow \mathbf{R}$ upper semi-computable if, for each $x \in \mathbf{R}$, we can compute from a given sequence of open rational intervals whose intersection contains the single point x, a sequence of rational numbers $\{q_n\}_{n \in \mathbf{N}}$ which satisfies $f(x) = \inf_{n \in \mathbf{N}} q_n$.

We will give our definition of upper semi-computability as follows. (See [20] and [6] for details.)

Definition 3.1 (Upper semi-computable functions) $f : \mathbf{R} \rightarrow \mathbf{R}$ is *upper semi-computable* if the following condition holds.

Let x be any real number. Suppose there is a sequence of rational intervals, say $\{(a_n, b_n)\}_n$, all of which contain x and which converges to x, and hence $\{|b_n - a_n|\}_n$ converges to 0. Then there is an algorithm to compute a sequence of rationals $\{q_n\}_n$ from the information afforded by $\{a_n\}$ and $\{b_n\}$, so that $f(x) = \inf_{n \in \mathbf{N}} q_n$.

Remark 2 1. Equivalently, we could replace the sequence of intervals by a sequence of rational numbers which converges to x.
2. In particular, if x is a computable real number, then there is a computable sequence of rationals converging effectively to x. Therefore, $\{q_n\}$ will be a computable sequence.
3. More precisely, the upper semi-computable functions $f : \mathbf{R} \to \mathbf{R}$ are exactly the $(\rho, \rho_>)$–computable functions in the sense of the theory of representations in [20].

With this preparation, we come to our first conclusion.

Theorem 1 The Gaußian function is *upper semi-computable.*

Proof Given a real number $x \in \mathbf{R}$ and a sequence $\{I_n\}_{n\in\mathbf{N}}$ of rational intervals (a_n, b_n) such that $\bigcap_{n\in\mathbf{N}} I_n = \{x\}$. Then simply define $q_n := [b_n]$ for all $n \in \mathbf{N}$. $\{q_n\}_{n\in\mathbf{N}}$ can be *computed* from $\{b_n\}_{n\in\mathbf{N}}$, since the Gaußian function restricted to rational numbers is a (classically) computable rational function.

It is then a mathematical practice to prove $[x] = \inf_{n\in\mathbf{N}}[b_n]$.

First, we note that $x < b_n$ for all $n \in \mathbf{N}$.
Case 1 $x \in \mathbf{Z}$. Then for each $\varepsilon > 0$ there is some $k \in \mathbf{N}$ such that $x < b_k < x + \varepsilon$ and thus $\inf_{n\in\mathbf{N}}[b_n] = x = [x]$.
Case 2 $x \notin \mathbf{Z}$. Let $y := \min\{z \in \mathbf{Z} : x < z\}$. Then there is some $k \in \mathbf{N}$ such that $x < b_k < y$ and thus $\inf_{n\in\mathbf{N}}[b_n] = [b_k] = y - 1 = [x]$.

Remark 3 In our theory, we distinguish between a sequence of rationals (expressed in the form of fractions) and a sequence of real numbers whose entries happen to be rationals. The distinction is especially important in considering computability of such a sequence, since there is an example of a computable sequence of real numbers whose entries are rationals but which is not computable as a sequence of rationals (See Example 4, Chapter 0 of [13].)

By a Theorem in [20] a function $f : \mathbf{R} \to \mathbf{R}$ is upper semi-computable, if and only if its open epigraph $\mathrm{epi}(f) := \{(x, y) \in \mathbf{R}^2 : f(x) < y\}$ is a recursively enumerable open set (cf. [6]).

Corollary 1 (Epigraph) The open epigraph $\{(x, y) \in \mathbf{R}^2 : [x] < y\}$ of the Gaußian function is an r.e. open set.

For a later use, we will give the definition of the lower semi- computability of a function $f : \mathbf{R} \to \mathbf{R}$ similarly to the upper semi-computability. Namely, in Definition 3.1, replace inf by sup, so that $f(x) = \sup_{n\in\mathbf{N}} q_n$.

4 Limiting computation In [8], Gold defined the notion of *limiting recursion*, that is, if, for a recursive (number-theoretic) function $g(p, n)$, the value will eventually become stationary with respect to n, then the computation of that stationary value for p is called the limiting recursion. He also extended the notion of the limiting recursion to that of functionals.

Definition 4.1 ([8]:Gold) (1) (Limiting recursive functional) We call a map from tuples of number-theoretic functions and natural numbers to a natural number a *functional.*

Let Σ be a set of total functions.

A *partial* functional, $F(\phi_1, \cdots, \phi_r, p_1, \cdots, p_s)$, where $r, s \geq 0$, will be called *limiting recursive on* Σ if there is a number-theoretic, total recursive function $g(z_1, \cdots, z_r, p_1, \cdots, p_s, n)$ satisfying

$$F(\phi_1, \cdots, \phi_r, p_1, \cdots, p_s) = \lim_n g(\tilde{\phi}_1(n), \cdots, \tilde{\phi}_r(n), p_1, \cdots, p_s, n)$$

for all r-tuples $\langle \phi_1, \cdots, \phi_r \rangle$ of functions in Σ and all $p_1, \cdots, p_s \in \mathbf{N}$, where $\tilde{\phi}(n)$ represents a code of the finite sequence $\langle \phi(0, p_1, \cdots, p_s), \cdots, \phi(n, p_1, \cdots, p_s) \rangle$.

F will be called simply *limiting recursive* if Σ is the class of all total recursive functions.

We will give two useful examples of limiting recursive functionals.

Example 1 (Least value functional) Let ϕ denote a number-theoretic function with $s+1$ arguments, $s \geq 0$. The least value property

$$\exists m \forall n \ \ (\phi(n, p_1, \cdots, p_s) \geq \phi(m, p_1, \cdots, p_s))$$

holds for ϕ. Let $L(\phi, p_1, \cdots, p_s)$ denote this least value $\phi(m, p_1, \cdots, p_s)$. L is limiting recursive, and will be called the *least value functional.*

We will show that the least value functional L is indeed limiting recursive. For the sake of simplicity, we assume $s = 0$. Let ϕ be a number-theoretic function. The computation strategy to evaluate $L(\phi)$ can be described as follows.

At each n, let g_n be a guess of the least value up to step n:

$$g_0 = \phi(0); g_n = \min(g_{n-1}, \phi(n))$$

Define a function g by

$$g(m, 0) = (m)_0; g(m, n+1) = \min(g(m, n), (m)_{n+1})$$

where $(m)_i$ is the i-th entry of m when m is regarded as a code of a finite sequence of numbers. Then g is (primitive) recursive and satisfies

$$g_n = g(\tilde{\phi}(n), n)$$

Now,

$$L(\phi) = \lim_n g(\tilde{\phi}(n), n)$$

L is thus limiting recursive (over recursive functions).

Example 2 (Limit functional) Let ϕ denote a number-theoretic function, and let Lim be a partial functional such that $Lim(\phi, p_1, \cdots, p_s) = \lim_n \phi(n, p_1, \cdots, p_s)$. Then Lim is limiting recursive.

We will assume that $s = 0$. That the limit functional Lim is limiting recursive can be shown as follows. Let ϕ be a number-theoretic function, and let g be a function defined by $g(m, n) = (m)_n$, where $(m)_n$ is as defined above. Then g is recursive and

$$g(\tilde{\phi}(n), n) = \phi(n)$$

So, $Lim(\phi) = \lim_n g(\tilde{\phi}(n), n)$, and Lim is limiting recursive.

As an example of application of L, we will consider $[x]$. Recall that we have made speculation on the computation of this function in Section 2.

Proposition 4.1 (A computation for $[x]$) There is a program to compute $[x]$ using the least value functional L (cf. Example 1 for L).

Proof Let $\langle\{r_n\},\alpha\rangle$ be a representation of x. We follow the computation algorithm in Section 2.

1. Find some n such that $n < x < n+2$.
2. Define a recursive sequence (of rationals) $\{N_p\}$ such that $N_p = n+1$ if $r_{\alpha(p)} \geq (n+1) - \frac{1}{2^p}$; $N_p = n$ if $r_{\alpha(p)} < (n+1) - \frac{1}{2^p}$.
3. If $l = L(\{N_p\}) = n+1$, then put $\beta(p) = 1$ for all p; if $l = L(\{N_p\}) = n$, then put $\beta(p) = \min\{q \in \mathbf{N} : r_{\alpha(q)} < n+1-\frac{1}{2^q}\}$.
4. Output $\langle\{N_p\},\beta\rangle$. This represents $[x]$.

Note The computation process in the proof above can be extended to a *sequence* of real numbers, that is, given a sequence of real numbers $\{x_i\}$, the computation of $\{[x_i]\}$ is carried out uniformly in i by using L.

We can thus conclude the following.

Theorem 2 (Sequential computation of $[x]$) The Gaußian function has a sequential computation using the least value functional. Namely, for any computable sequence of real numbers, the sequence of values $\{[x_i]\}$ can be computed effectively in terms of the least value functional L.

Proof In the proof of Proposition 4.1, the k_i for which $k_i < x_i < k_i + 2$ holds can be computed effectively in i. The definition of N_{ip} is also effective in i, since the computation of

$$N_{ip} = k_i + 1 \text{ if } r_{i\alpha(i,p)} \geq (k_i+1) - \frac{1}{2^p};\ N_{ip} = k_i \text{ if } r_{i\alpha(i,p)} < (k_i+1) - \frac{1}{2^p}$$

is effective in i. Then we can apply L to $\{N_{ip}\}_p$, and we put $l_i = L(\{N_{ip}\}, i)$. $l_i = k_i + 1$ (Case 1 for i) or $l_i = k_i$ (Case 2 for i). For Case 1, put $\beta(i,p) = 1$, and, for Case 2, put $\beta(i,p) = \mu q(r_{i\alpha(i,q)} < (k_i+1) - \frac{1}{2^p})$. Now output $\langle\{N_{ip}\},\beta\rangle$.

In the following, we will prove a general proposition: any upper semi-computable function can be computed in terms of Lim, the limit functional. This means that the computation in terms of limiting recursive functionals is a wider notion than upper semi-computability. The same holds for lower semi-computability. On the other hand, there are functions which can be computed in terms of limiting recursive functionals but are neither upper nor lower semi-computable.

Theorem 3 (Upper semi-computability, lower semi-computability and Lim) Each upper (lower) semi-computable function can be computed in terms of the functional Lim.

Proof We will give a proof for the upper semi-computable function. Let $f : \mathbf{R} \to \mathbf{R}$ be an upper semi-computable function. From a given information $\langle\{r_n\},\alpha\rangle$ which represents x, one can compute a sequence $\{q_l\}$ of rational numbers such that $f(x) = \inf q_l$.

For the simplicity, we may assume that q_l are positive. We will first define a sequence $\{t_l\}$ as follows.

$$t_0 = q_0; t_{l+1} = \min\{q_{l+1}, t_l\}$$

If $\{q_l\}$ is recursive, then so is $\{t_l\}$. $\{t_l\}$ is decreasing (non-increasing), and so is monotone converging to $f(x)$. Next define a function g by

$$g(l,p) = \begin{cases} \min\{k : k < l, t_k - t_l < \frac{1}{2^p}\}, & \text{if such } k \text{ exists;} \\ 0, & \text{otherwise.} \end{cases}$$

If $\{t_l\}$ is recursive, then the function g is also recursive. Since $\{t_l\}$ is monotone converging, $\{g(l,p)\}_l$ is non-decreasing and bounded above. So,

$$Lim(g,p) = \lim_{l\to\infty} g(l,p)$$

exists. Furthermore, $Lim(g,p) > 0$ for every p.

For the modulus of convergence, one can take $\beta(p) = Lim(g,p)$. We have thus obtained an output $\langle\{t_l\},\beta\rangle$ in terms of the limit functional, given an information $\langle\{r_n\},\alpha\rangle$.

We will subsequently give further examples of discontinuous functions which can be "computed" in terms of limiting recursive functionals. We asssume that a real number x is represented by $\langle\{r_m\},\alpha\rangle$. When we say an object can be found, we will mean that it can be found effectively from the information $\langle\{r_m\},\alpha\rangle$.

Example 3 Let h be the function $h(x) = x - [x]$. Here we may assume that $r_m > x$. The n satisfying $n < x < n+2$ can be found. Define a sequence of natural numbers $\{N_q\}$ and a sequence of rational numbers $\{s_q\}$ as follows.

Put $R(p) \equiv (r_{\alpha(p)} < (n+1) - \frac{1}{2^p})$.

$$N_q = \begin{cases} n, & \text{if } \exists p \le q\ R(p) \\ n+1 & \text{if } \forall p \le q\ \neg R(p) \end{cases}$$

$$s_q = \begin{cases} r_q - n, & \text{if } \exists p \le q\ R(p) \\ r_q - (n+1) & \text{if } \forall p \le q\ \neg R(p) \end{cases}$$

$\{s_q\}$ converges to $h(x)$. The modulus of convergence, say β, can be obtained as follows. If $L(\{N_q\}) = n$, then $\beta(p) = \max(\alpha(p), p_0)$, where $p_0 = \mu(r_{\alpha(p)} < (n+1) - \frac{1}{2^p})$; if $L(\{N_p\}) = n+1$, then $\beta(p) = \alpha(p)$.

Example 4 Let $]x[$ denote the following real function.

$$]x[= n \text{ if and only if } n < x \le n+1$$

$]x[$ is lower semi-computable but is not upper semi-computable, and can be computed in terms of the least value functional L.

Example 5 Define a function σ as follows: $\sigma(x) = 1(x \in (0,\infty)), = \frac{1}{2}(x = 0), = 0(x \in (-\infty, 0))$. σ is a modified version of the signum function, and is neither upper nor lower semi-computable.

σ can be computed in terms of the limit functional Lim.

Given a real number x with a representation $\langle\{r_n\},\alpha\rangle$, define a sequence of natural numbers $\{M_p\}$ as follows.

$$M_p = \begin{cases} 2, & \text{if } r_{\alpha(p)} > \frac{1}{2^p} \\ 1, & \text{if } |r_{\alpha(p)}| \le \frac{1}{2^p} \\ 0 & \text{if } r_{\alpha(p)} < -\frac{1}{2^p} \end{cases}$$

For each p, it can be decided which condition holds, and hence the sequence $\{M_p\}$ is recursive.

Define $s_p = \frac{1}{2}M_p$. $\{s_p\}$ is a recursive sequence of rational numbers.

We will show that $\sigma(x) = \lim_p s_p$.

Case 1: $\exists p\ (r_{\alpha(p)} > \frac{1}{2^p})$.

Once a p satisfying $r_{\alpha(p)} > \frac{1}{2^p}$ is hit, $M_q = 2$ for all $q \ge p$, and hence $\lim_p M_p = 2$ and

$\lim_p s_p = 1$. Since the condition $\exists p\ (r_{\alpha(p)} > \frac{1}{2^p})$ means that $x > 0$, $\sigma(x) = 1$, and so $\sigma(x) = \lim_p s_p = 1$.

Case 2: $\forall p\ (|r_{\alpha(p)}| \leq \frac{1}{2^p})$
In this case, the limit of $\{M_p\}$ is 1, $x = 0$ and hence $\sigma(x) = \frac{1}{2} = \lim_p s_p$.

Case 3: $\exists p\ (r_{\alpha(p)} < -\frac{1}{2^p})$
Once such a p is hit, then $M_q = 0$ for every $q \geq p$, and the limit of $\{M_q\}$ is 0. Furthermore, $x < 0$ and hence $\sigma(x) = 0$, and $\lim_p s_p = 0$.

In any case, $\lim_p s_p$ exists, and $\sigma(x) = \lim_p s_p$.

Let Lim be the limit functional defined in Example 2. $Lim(\{M_p\}) = 2$ for Case 1, $Lim(\{M_p\}) = 1$ for Case 2, and $Lim(\{M_p\}) = 0$ for Case 3.

The modulus of convergence γ is the following.

If $Lim(\{M_p\}) = 2$ or $Lim(\{M_p\}) = 0$, then $\gamma(p) = \min\{q \in \mathbf{N} : |r_{\alpha(q)}| > \frac{1}{2^q}\}$ will do.

If $Lim(\{M_p\}) = 1$, then $\gamma(p) = 1$ will do.

Now output $\langle \{s_p\}, \gamma \rangle$.

As an example of a function sequence which is computable in terms of limiting recursive functionals, let us deal with the Rademacher function system. Some computational properties of this function sequence has been studied in [25].

Example 6 Let n denote $0, 1, 2, 3, \cdots$. Then the nth *Rademacher function* $\phi_n(x)$ is defined as follows.

$$\phi_0(x) = 1, \quad x \in [0, 1)$$

$$\phi_n(x) = \begin{cases} 1, & x \in [\frac{2i}{2^n}, \frac{2i+1}{2^n}) \\ -1, & x \in [\frac{2i+1}{2^n}, \frac{2i+2}{2^n}) \end{cases}$$

where $n \geq 1$ and $i = 0, 1, 2, \cdots, 2^{n-1} - 1$.

The sequence $\{\phi_n(x)\}$ is called the *system of Rademacher functions*, or the *Rademacher system*. The Rademacher system has an important role in Walsh analysis (cf. [7] and [26]).

Rademacher functions can be computed as a sequence of functions in terms of the least value functional as follows.

For an n, the k satisfying $\frac{k}{2^n} < x < \frac{k+2}{2^n}$ can be found. Write k_n for such k. $\{k_n\}$ can be effectively defined. Define $\{N_q^n\}$ as follows.

$$N_q^n = \begin{cases} k_n, & \text{if } \exists p \leq q\ (r_{\alpha(p)} < \frac{k_n+1}{2^n} - \frac{1}{2^p}) \\ k_n + 1, & \text{if } \forall p \leq q\ (r_{\alpha(p)} \geq \frac{k_n+1}{2^n} - \frac{1}{2^p}) \end{cases}$$

Define $\{t_q^n\}$ as follows. $t_q^n = 1$ if N_q^n is even; $t_q^n = -1$ if N_q^n is odd. $\{t_q^n\}_q$ converges to $\phi_n(x)$. The modulus of convergence is the following. $\beta_1(n, p) = \min\{p \in \mathbf{N} : r_{\alpha(p)} < k_{n+1} + 1 - \frac{1}{2^p}\}$ if $L(\{N_q^n\}, n) = k$; $\beta_2(n, p) = 1$ if $L(\{N_q^n\}, n) = k + 1$.

The following example is a partial function on the real line. Namely, we will consider the function $\tan x$.

Example 7 In order to make the discussion simple, we define a function τ as follows. $\tau(x) = \tan x$ if $\frac{2n+1}{2}\pi < x < \frac{2n+3}{2}\pi$; $\tau(x) = 0$ if $x = \frac{2n+1}{2}\pi$ for all n. Unlike the preceding examples, the function τ is unbounded on appropriately large compact sets.

An integer n satisfying $\frac{2n+1}{2}\pi < x < \frac{2n+5}{2}\pi$ can be found. Define $\{N_p\}$ as follows.

$$N_p = \begin{cases} n, & \text{if } r_{\alpha(p)} < \frac{2n+3}{2}\pi - \frac{1}{2^p} \\ n+2, & \text{if } r_{\alpha(p)} > \frac{2n+3}{2}\pi + \frac{1}{2^p} \\ n+1, & \text{if } \frac{2n+3}{2}\pi - \frac{1}{2^p} \leq r_{\alpha(p)} \leq \frac{2n+3}{2}\pi + \frac{1}{2^p} \end{cases}$$

Define $\{t_q\}$ as follows.

$$t_q = \begin{cases} \tan r_q, & \text{if } N_q = n \text{ or } N_q = n+2 \\ 0, & \text{if } N_q = n+1 \end{cases}$$

$\{t_q\}$ converges to $\tau(x)$. The modulus of convergence can be defined similarly to that of Example 5 according as $Lim(\{N_q\}) = n$, $n+2$ or $n+1$.

Note In Theorems 2 and 3 as well as in each of Examples 3 through 7 above, a sequence of rational numbers approximating a function value $f(x)$ can be obtained in a uniform manner from the information on x, and is recursive given that x is computable, while a limiting recursive functional is used to determine a modulus of convergence. In all the cases except Theorem 3, a modulus of convergence is recursive, although which function it is cannot be known effectively. In Theorem 3, a modulus of convergence itself requires a limiting computation.

5 Discontinuous functions in Fréchet spaces We will first give a brief review concerning classical theory of Fréchet spaces and the notion of computability thereof for the *reader's convenience*. See, for example, [14] for the former and [16] for the latter.

Some definitions concerning the computability structure on a Fréchet space will be cited from [16] and [23].

A Fréchet space is defined as a locally convex topological vector space that is metrizable and complete. We will work on a Fréchet space accompanied by a sequence of seminorms defining its topology.

Let $\langle X, \{p_m\}\rangle$ be a Fréchet space with a sequence of semi-norms $\{p_m\}$.

Definition 5.1 (Computability structure on a Fréchet space: Definitions 1 and 2 of [16] and Definition 5.1 in [23]) (i) Let $\{x_{nk}\}$ and $\{x_n\}$ be respectively a double sequence and a sequence from X. $\{x_{nk}\}$ is said to *converge* to $\{x_n\}$ *effectively* in k and n if there exists a recursive function d such that

$$k \geq d(m,n,l) \text{ implies } p_m(x_{nk} - x_n) \leq \tfrac{1}{2^l}$$

(ii) A *nonempty* family of sequences from X, say $\mathcal{S}$, is called a *computability structure* on X if the following three axioms are satisfied.

Axiom 1 (Linear forms) For $\{x_n\}, \{y_n\} \in \mathcal{S}$, for any computable sequences of reals $\{\alpha_{nk}\}$ and $\{\beta_{nk}\}$, and for a recursive function d, the sequence $\{s_n\}$ defined by

$$s_n = \sum_{k=1}^{d(n)} (\alpha_{nk}x_k + \beta_{nk}y_k)$$

belongs to $\mathcal{S}$.

Axiom 2 (Limits) Let $\{x_{nk}\} \in \mathcal{S}$ satisfy that x_{nk} converges to x_n as $k \to \infty$ effectively in k and n. Then $\{x_n\} \in \mathcal{S}$.

Axiom 3 (Seminorms) If $\{x_n\} \in \mathcal{S}$, then $\{p_m(x_n)\}$ is a computable double sequence of reals.

Let us remark that, since we assume nonemptiness of $\mathcal{S}$, $\{0, 0, \cdots\}$ belongs to $\mathcal{S}$.

(iii) A sequence $\{x_n\}$ is called computable if $\{x_n\} \in \mathcal{S}$. A point $x \in X$ is called computable if $\{x, x, \cdots\}$ is computable.

We will first discuss the computability of the Gaußian function in the Fréchet space of integer indexed real sequences.

The notations **N**, **Z** and **R** defined in Section 3 will be used.

$\mathbf{R}^{\mathbf{N}}$(which is denoted by $\mathbf{R}^\infty$ in the preceding section and in **2** of [16]) will denote the Fréchet space of all sequences of reals $x = (\xi_k)$. Its seminorms are defined by

$$p_m(x) = \max\{|\xi_j| : 0 \leq j \leq m\}$$

A computability structure on $\mathbf{R}^{\mathbf{N}}$ is the family of all sequences $\{x_n\}$, where $x_n = (\xi_k^n)$ and the sequence $\{\xi_k^n\}$ is a computable double sequence of reals.

Definition 5.2 (Integer indexed sequences) (i) $\mathbf{R}^{\mathbf{Z}}$ will denote the space of all sequences of the form

$$x = (\cdots, \xi_{-k}, \cdots, \xi_{-2}, \xi_{-1}, \xi_0, \xi_1, \xi_2, \cdots, \xi_k, \cdots)$$

where $k = 1, 2, \cdots$ and each ξ_l is a real number.

(ii) Define a system of real seminorms $\{p_m\}$ from $\mathbf{R}^{\mathbf{Z}}$ by

$$p_m(x) = \sum_{|k| \leq m} |\xi_k| \tag{1}$$

for $m = 1, 2, \cdots$.

By simply examining the axioms on a Fréchet space, we can show the proposition below.

Proposition 5.1 (Fréchet space of integer indexed real sequences) $\mathbf{R}^{\mathbf{Z}}$ forms a real vector space with respect to addition and scalar multiplication over reals. $\{p_m\}$ serves as a separating system of seminorms, and $\mathbf{R}^{\mathbf{Z}}$ is complete with respect to this seminorm system. That is, $\langle \mathbf{R}^{\mathbf{Z}}, \{p_m\} \rangle$ is a Fréchet space.

Proposition 5.2 (Computability structure for $\mathbf{R}^{\mathbf{Z}}$) Let $\mathcal{S}$ be a family of sequences from $\mathbf{R}^{\mathbf{Z}}$, and let $\{x_n\}$ denote a sequence from $\mathbf{R}^{\mathbf{Z}}$, where $x_n = (\xi_k^n), n = 0, 1, 2, \cdots$ and $k \in \mathbf{Z}$. Suppose $\mathcal{S}$ satisfies the following.

$\{x_n\} \in \mathcal{S}$ if and only if $\{\xi_{\gamma(i)}^n\}$ is a computable double sequence of reals, where $\gamma(2i) = i$ and $\gamma(2i+1) = -i, i = 0, 1, 2, \cdots$.

(As usual, $\mathcal{S}$ may be assumed to contain also multiple-indexed sequences. By a standard manipulation of coding, we can regard γ as a recursive function.)

Then $\mathcal{S}$ satisfies the axioms of the computability structure (See Definition 5.1.) on the Fréchet space $\mathbf{R}^{\mathbf{Z}}$.

This can be shown similarly to Example (1) in **2** of [16]. The proof is straightforward.

Let Σ denote the set of right continuous step functions from **R** to **R** with integer jump points. A function f in Σ is determined by its values $f(k)$ for integers k. So, we can identify f with its value sequence $\{f(k)\}$.

The correspondence between a function f in Σ and the sequence $\{f(k)\}$ induces an isomorphism between Σ and $\mathbf{R}^{\mathbf{Z}}$ with respect to linear combinations over reals, and hence we can identify Σ and $\mathbf{R}^{\mathbf{Z}}$. We will thus call a sequence of functions from Σ, say $\{f_n\}$ computable if the double sequence $\{f_n(k)\}$ is a computable sequence in $\mathbf{R}^{\mathbf{Z}}$.

As a special case, a function $f \in \Sigma$ is computable if $\{f(k)\} \in \mathcal{S}$.

Now take f to be the Gaußian function $[x]$. $[x] \in \Sigma$ and $\{f(k)\} = \{k\}$, which is certainly a computable element of $\mathbf{R}^{\mathbf{Z}}$. From this fact, we are led to the following.

Proposition 5.3 (Σ-computability) The Gaußian function is computable as a function in Σ.

We will next discuss the computability of the functions in Σ as functions in the Fréchet space of locally integrable real functions. See Definition 5.1 for the effective convergence and the computability structure on a Fréchet space.

Definition 5.3 (Locally integrable functions) (i) A function $f : \mathbf{R} \to \mathbf{R}$ is called locally integrable if f is integrable on every compact interval. $L^1_{loc}(\mathbf{R})$ will denote the set of such functions.

(ii) Define a system of real valued functions $\{p_m\}$ from $L^1_{loc}(\mathbf{R})$ by

$$p_m(f) = \int_{[-m,m]} |f| dx$$

for $m = 1, 2, \cdots$.

From Definition 5.3, it can be easily verified that $\langle L^1_{loc}(\mathbf{R}), \{p_m\} \rangle$ is a Fréchet space.

Proposition 5.4 (Computability structure for $L^1_{loc}(\mathbf{R})$) Let $\mathcal{S}$ be a family of sequences from $L^1_{loc}(\mathbf{R})$. Suppose $\mathcal{S}$ satisfies the following.

$\{f_n\} \in \mathcal{S}$ if and only if there is a computable double sequence of real functions in the sense of Definition A$''$ of [13], say $\{g_{nk}\}$, such that $\{g_{nk}\}$ converges to $\{f_n\}$ (with respect to the seminorm system $\{p_m\}$ above) effectively as $m \to \infty$. That is, $\{f_k\}$ can be approximated (with respect to $\{p_m\}$) *effectively* by a double sequence of *continuous* and *computable* functions over reals on every compact interval $[-k, k]$, $m = 1, 2, \cdots$. (See Definition 5.1 for effective seminorm convergence.)

Then $\mathcal{S}$ is a computability structure for the space $\langle L^1_{loc}(\mathbf{R}), \{p_m\} \rangle$.

The step functions with compact supports $[-k, k], k = 1, 2, \cdots$, and with rational values and jump points form (if enumerated effectively) a computable sequence in $L^1_{loc}(\mathbf{R})$.

The proposition can be proved in a manner similar to Examples in 3 of Chapter 2 in [13].

As a corollary of Proposition 5.4, we obtain the following.

Proposition 5.5 ($L^1_{loc}(\mathbf{R})$ -computability) The Gaußian function $[x]$ is computable in the space of locally integrable functions $\langle L^1_{loc}(\mathbf{R}), \{p_m\} \rangle$.

Note We could have considered the function $[x]$ as a point in the Banach space $L^1(\mathbf{R}, \mu)$, where μ denotes the *Gaußian measure*, that is, $d\mu = e^{-x^2} dx$. We have not taken that approach, since we believe that the behavior of the Gaußian function is better understood in $L^1_{loc}(\mathbf{R})$.

At the end, we will briefly explain that some examples in Section 4 can be regarded as computable objects in some function spaces. Examples 3,4 and 5 are computable in $L^1_{loc}(\mathbf{R})$. As for Example 7, we can define a Fréchet space $\mathcal{F}$ of real functions as follows. Put $\Omega = \mathbf{R} - \{\frac{2n+1}{2}\pi : n \in \mathbf{Z}\}$, and let $L^1_{loc}(\Omega)$ be the set of functions which are integrable on every compact set in Ω. Define a system of seminorms $\{p_m\}$ by

$$p_m(f) = \sum_{|n| \le m} \int_{I_{n,m}} |f|,$$

where $I_{n,m} = [\frac{2n+1}{2}\pi + \frac{1}{2^m}, \frac{2n+3}{2} - \frac{1}{2^m}]$. Put $\mathcal{F} = \langle L^1_{loc}(\Omega), \{p_m\}\rangle$. The function τ in Example 7 is an element of this space. We can define a computability structure for $\mathcal{F}$ with effective approximations by computable continuous functions in $\mathcal{C}(\mathbf{R})$, and τ will be computable in this sense.

Address of gratitude: We are grateful to Y. Akama, B. Buchburger, S. Hayashi, C. Paulin and B. Werner for their attention and valuable discussions while preparing this article. In particular, Hayashi introduced to us the notion of limiting recursion, with which we have been able to improve this article.

References

[1] Y. Akama, *Limiting Partial combinatory algebras*, (2001); available at http://www.math.tohoku.ac.jp/~akama/lcm/.

[2] L. Blum, M. Shub and S. Smale, *On a theory of computation and complexity over the real numbers*: *NP*-completeness, recursive functions and universal machines, Bulletin of American Mathematical Society, vol.21, no.1 (1989), 1-46.

[3] V. Brattka, *Recursive characterization of computable real-valued functions and relations*, Theoretical Computer Science 162 (1996), 45-77.

[4] V. Brattka, *A Stability Theorem for Recursive Analysis*, Combinatorics, Computation & Logic (Proceedings of DMTCS'99 and CATS'99), ed. by C.S. Calude et al, Springer (1999), 144-158

[5] V. Brattka and P. Hertling, *Feasible Real Random Access Machines*, Journal of Complexity, 14 (1998), 490-526.

[6] V. Brattka and K. Weihrauch, *Computability on subsets of Euclidean space I: Closed and compact subsets*, Theoretical Computer Science, 219 (1999), 65-93.

[7] N.J. Fine, *On the Walsh functions*, Trans. Amer. Math. Soc., 65 (1949), 372-414.

[8] E.M. Gold, *Limiting recursion*, JSL, 30-1(1965), 28-48.

[9] T. Mori, *On the computability of Walsh functions*, to appear in TCS.

[10] T. Mori, *Computabilities of functions on effectively separable metric spaces*, Proceedings of CCA2000, (2000), 299-317.

[11] T. Mori, Y. Tsujii and M. Yasugi, *Computability structure on metric spaces,Combinatorics*, Complexity and Logic (Proceedings of DMTCS'96), ed. by Bridges et.al, Springer (1996), 351-362

[12] M. Nakata and S. Hayashi, *Limiting first order realizability interpretation*, manuscript (2000).

[13] M. Pour-El and I. Richards, *Computability in analysis and physics*, Springer, 1989.

[14] W. Rudin, *Functional Analysis*, McGraw-Hill, Inc., 1973.

[15] Y. Tsujii, M. Yasugi and T. Mori, *Some properties of the effective uniform topological space*, Proceedings of CCA2000(2000), 421-440; available also at http://www.kyoto-su.ac.jp/∼yasugi/Recent.

[16] M. Washihara, *Computability and Fréchet space*, Mathematica Japonica, 42 (1995), 1-13.

[17] M. Washihara, *Computability and tempered distributions*, ibid., 50, 1 (1999), 1-7.

[18] M. Washihara and M. Yasugi, *Computability and metrics in a Fréchet space*, ibid., 43 (1996), 1-13

[19] K. Weihrauch, *Computable Analysis*, Springer, 2000.

[20] K. Weihrauch and X. Zheng, *Computability on continuous, lower semi-continuous and upper semi-continuous real functions*, Theoretical Computer Science 219 (1999) 439-450.

[21] K. Weihrauch and N. Zhong, *The Wave Propagator is Turing Computable*, Automata, Languages and Programming (Proceedings of ICALP'99), ed. by J. Wiedermann et al, Springer (1999), 697-706.

[22] M. Yasugi, T. Mori and Y. Tsujii, *Effective Properties of Sets and Functions in Metric Spaces with Computability Structure*, Theoretical Computer Science, 219 (1999), 467-486.

[23] M. Yasugi and M. Washihara, *Computability problem of Gaußian function*, Proceedings of CCA'98 (Brno), Informatik Berichte 235-8/1998 (FernUniversität), Computability and Complexity in Analysis, ed. by K-I. Ko et. al, (1998), 157-165.

[24] M. Yasugi and M. Washihara, *Computability structures in analysis*, Sugaku Expositions(AMS), 13(2000),215-235.

[25] M. Yasugi and M. Washihara, *A note on Rademacher functions and computability*, addressed at ICALW 2000; available at http://www.kyoto-su.ac.jp/~yasugi/Recent.

[26] K. Yoneda, *On application of Walsh Fourier analysis to weakly stationary processes*, Acta. Math. Hungar., 76 (1997), 303-335.

*Faculty of Science, Kyoto Sangyo University:
yasugi@cc.kyoto-su.ac.jp.
†Theoretische Informatik I, FernUniversität:
Vasco.Brattka@FernUni-Hagen.de
‡Faculty of Science, Kyoto Sangyo University:
wasihara@cc.kyoto-su.ac.jp