COMPUTABILITY OVER TOPOLOGICAL STRUCTURES

Vasco Brattka*
Theoretische Informatik 1
FernUniversität Hagen
58084 Hagen
Germany
Vasco.Brattka@FernUni-Hagen.de

Abstract Computable analysis is the Turing machine based theory of computability on the real numbers and other topological spaces. Similarly as Eršov's concept of numberings can be used to deal with discrete structures, Kreitz and Weihrauch's concept of representations can be used to handle structures of continuum cardinality. In this context the choice of representations is very sensitively related to the underlying notion of approximation, hence to topology. In this paper we summarize some basic ideas of the representation based approach to computable analysis and we introduce an abstract and purely set theoretic characterization of this theory which can be considered as a generalization of the classical concept of μ-recursive functions. Together with this characterization we introduce the notion of a perfect topological structure. In particular, these structures are effectively categorical, i.e. they characterize their own computability theory. Important examples of perfect structures are provided by metric spaces and additional attention is paid to their effective subsets.

Keywords: Computable analysis, recursion, representations, topological structures.

1. Introduction to Computable Analysis

As one starting point of computable analysis one can surely consider Turing's famous paper [59], which begins with the words:

> The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means.

*Work supported by DFG Grant BR 1807/4-1
to appear in: S. Barry Cooper and Sergey Goncharov (eds.), *Computability and Models*, Kluwer Academic Publishers, Dordrecht, 2002 (in preparation)

In the same paper Turing introduced his well-known machine model in order to give a precise meaning to the words "calculable by finite means". He already realized that the class of computable real numbers, although countable, is quite large and contains all well-known mathematical constants as π and e. As proved by Rice, the class of computable real numbers even forms a real algebraically closed field [46]. Based on Turing's ideas, Banach and Mazur worked out a first theory of computable real number functions [3]. Grzegorczyk and Rasiowa compiled a book on lectures which have been given on this approach by Mazur [40]. The next major step was achieved independently by Grzegorczyk [24, 25] and Lacombe [36] who defined computable real number functions in an equivalent way as it is done today.

While the notion of a computable real number is quite stable with respect to different representations of the real numbers (such as the decimal representation, the binary representation, the continued fraction representation and others), see Robinson [47], it was observed by Mostowski [43] that the notion of a computable function and even the notion of a computable sequence does very sensitively rely on the choice of the representation. Actually, it was already Turing who noticed in a correction [60] of his famous paper that the decimal representation is not suitable to define computable functions and even multiplication by 3 is not computable with respect to this representation. Turing suggested other representations like the interval representation to overcome this difficulty and actually Grzegorczyk's and Lacombe's definitions are based on equivalent representations.

The first systematic study of representations has been presented by Hauck [26, 27], and following this line Kreitz and Weihrauch developed a unified theory of representations which conclusively shows that computational differences of representations are often caused by topological reasons [32, 33, 61, 65, 34, 64]. An alternative but compatible approach to computability on Banach spaces has been presented by Pour-El and Richards [45] and also Ko's theory of time complexity on the real numbers [30] is polynomially equivalent to Kreitz and Weihrauch's approach. While all the forementioned approaches are based on classical logic, there are also several closely related lines of research which are based on different systems of intuitionistic logic. We only mention Bishop school of constructive analysis [6] and Markov's school of Russian constructive analysis [35]. Aberth's work on computable analysis [1] is based on classical logic but closely related to the Russian approach.

A different line of research was devoted to the generalization of classical abstract characterizations of computable functions. Such generalizations have been presented, for instance, by Moschovakis, Friedman, Shepherdson, Fenstad, Tucker and Zucker and Skordev [42, 22, 50, 21, 56, 52]. However, non of these approaches was designed to fit together with computable analysis.

In the following Sections 2 to 6 we will give a brief introduction to certain aspects of the representation based approach to computable analysis. Among other topics we will discuss admissible representations, special representations of the real numbers, the concept of a computable metric space and several notions of effective subsets of such spaces. In Section 7 and 8 we extend the theory to multi-valued operations which are important since even very elementary problems of analysis can only be solved by multi-valued computations. In Sections 9 and 10 we present an abstract characterization of computability which is similar in spirit to the classical characterization by μ–recursive functions. Combined with this characterization we develop a theory of perfect structures which have several nice properties. Especially, they are effectively categorical in the sense that they characterize their own computability theory. In Section 11 we apply this concept to topological structures and we discuss important metric examples. The final Section 12 is devoted to the abstract characterization of effective subsets over structures.[1]

2. Representations and Computable Functions

Inspired by Eršov's theory of numberings [19, 20], Kreitz and Weihrauch have introduced the notion of a representation, see e.g. [32, 33, 61, 62]. A comprehensive and recent account of this theory can be found in [64]. We present some basic concepts of the representation based approach. Roughly speaking, a representation is a (partial) map which assigns at least one "name" to each point of the represented set.

Definition 2.1 (Representation) A *representation* of a set X is a surjective mapping $\delta :\subseteq \mathbb{N}^{\mathbb{N}} \to X$. In this situation (X, δ) is called a *represented space*.

Here and in the following, $\mathbb{N}^{\mathbb{N}}$ denotes the set of sequences $p : \mathbb{N} \to \mathbb{N}$, where $\mathbb{N} := \{0, 1, 2, ...\}$ and the inclusion symbol "$\subseteq$" is used to denote partial maps. A point $x \in X$ is called δ–*computable*, if there exists some computable $p \in \mathbb{N}^{\mathbb{N}}$ such that $\delta(p) = x$. For the definition of computable functions $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ we will use the *set of finite sequences* or *words*

[1]The material contained in Sections 7 to 12 is mainly based on the author's thesis [9] which contains additional results and proofs. An extended abstract on parts of the material has been published in [10] and preliminary versions of some results have been published in [7, 8]. Interesting related considerations of WHILE programs have recently been presented by Tucker and Zucker [57, 58].

$\mathbb{N}^*$ of natural numbers. For all $v, w \in \mathbb{N}^*$, $p \in \mathbb{N}^\mathbb{N}$ we will write $v \sqsubseteq w$ and $v \sqsubseteq p$, if v is a *prefix* of w or of p, respectively. Occasionally, we will use the notation wp for the *concatenation* of a word $w \in \mathbb{N}^*$ and a sequence $p \in \mathbb{N}^\mathbb{N}$ and we will write $w\mathbb{N}^\mathbb{N}$ for the set $\{p \in \mathbb{N}^\mathbb{N} : w \sqsubseteq p\}$ of all sequences which extend the word $w \in \mathbb{N}^*$. For any sequence $p \in \mathbb{N}^\mathbb{N}$ and $n \in \mathbb{N}$ we denote by $p[n] := p(0)...p(n-1)$ the prefix of p of length n. A function $\varphi : \mathbb{N}^* \to \mathbb{N}^*$ is called *monotone*, if $v \sqsubseteq w \Longrightarrow \varphi(v) \sqsubseteq \varphi(w)$ holds for all $v, w \in \mathbb{N}^*$. Now we can approximate partial functions $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ by monotone functions $\varphi : \mathbb{N}^* \to \mathbb{N}^*$.

Definition 2.2 (Computable functions) A function $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ is called *computable*, if there exists a computable, total and monotone function $\varphi : \mathbb{N}^* \to \mathbb{N}^*$ such that φ *approximates* F, i.e.

$$F(p) = \begin{cases} \sup\limits_{w \sqsubseteq p} \varphi(w) & \text{if the length of the supremum is not finite} \\ \uparrow & \text{else} \end{cases}$$

for all $p \in \mathbb{N}^\mathbb{N}$.

Here we assume that the notion of a computable function $\varphi : \mathbb{N}^* \to \mathbb{N}^*$ is well-known. It could be made precise using some bijective standard numbering $\nu^* : \mathbb{N} \to \mathbb{N}^*$ of the set $\mathbb{N}^*$, defined by $\nu^*(0) := \varepsilon$ and inductively by $\nu^*(\langle\langle n_0, ..., n_k\rangle, k\rangle + 1) := n_0 n_1 ... n_k$ for all $k, n_0, ..., n_k \in \mathbb{N}$ (where ε denotes the *empty word* and $\langle.\rangle$ denotes a tupling function which will be precisely defined below). For any sequence $p \in \mathbb{N}^\mathbb{N}$ and $n \in \mathbb{N}$ we denote by $\overline{p}n := (\nu^*)^{-1} p[n]$ the code of the prefix of length n.

The set $\mathbb{N}^\mathbb{N}$, considered as *Baire's space*, comes equipped with the product topology of the discrete topology on $\mathbb{N}$. It is well-known that computable functions $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ are continuous and their domains are G_δ-sets [62]. Alternatively, one can characterize computable functions by Turing machines which operate on infinite tapes over the infinite[2] alphabet $\mathbb{N}$: a function $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ is computable, if and only if there exists a Turing machine with one-way output tape which transfers each input sequence $p \in \mathrm{dom}(F)$ into the corresponding output sequence $F(p)$ by an infinite computation and which diverges on all other sequences. Using computable functions $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ one can lift the computability notion to represented spaces.

[2]In some references $\mathbb{N}^\mathbb{N}$ is used [62], while in other reference, as [64], $\mathbb{N}^\mathbb{N}$ is replaced by set of infinite words Σ^ω over some finite alphabet Σ. This has some advantages as soon as complexity is considered but computationally it is essentially equivalent.

Definition 2.3 (Computability) Let (X, δ_X), (Y, δ_Y) be represented spaces. Then $f :\subseteq X \to Y$ is called a (δ_X, δ_Y)*–computable function*, if there is a computable function $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ such that

$$\delta_Y F(p) = f\delta_X(p)$$

for all $p \in \mathrm{dom}(f\delta_X)$. If, additionally, $p \notin \mathrm{dom}(F)$ for all $p \in \mathrm{dom}(\delta_X) \setminus \mathrm{dom}(f\delta_X)$ holds, then f is called *strongly* (δ_X, δ_Y)*–computable.* We will also say that f is (strongly) (δ_X, δ_Y)–computable *via* F.

If we substitute continuous functions F for computable functions F in the previous definition, then we obtain the corresponding notion of (δ_X, δ_Y)*–continuity.* Figure 1 illustrates the definition.

$$\begin{tikzcd}
\mathbb{N}^\mathbb{N} \arrow[r, "F"] \arrow[d, "\delta_X"'] & \mathbb{N}^\mathbb{N} \arrow[d, "\delta_Y"] \\
X \arrow[r, "f"'] & Y
\end{tikzcd}$$

Figure 1. Computability with respect to representations

Using the intuition of Turing machines, a function $f :\subseteq X \to Y$ is (δ_X, δ_Y)–computable, if there exists a Turing machine which transfers each sequence p, which represents some $x \in \mathrm{dom}(f)$ with respect to δ_X, into a sequence q, which represents $f(x)$ with respect to δ_Y. The strong notion of computability would additionally require that the machine diverges on sequences p which represent elements outside the domain $\mathrm{dom}(f)$. If we consider the identity $\mathrm{id} : \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ as representation of $\mathbb{N}^\mathbb{N}$, then a function $f :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ is strongly $(\mathrm{id}, \mathrm{id})$–computable, if and only if it is computable. Moreover, f is $(\mathrm{id}, \mathrm{id})$–computable, if and only if it is the restriction of a computable function $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$. Thus, the "weak" standard notion of computability does not require natural domains (this is similar with the notion of continuity in topology) while the strong notion of computability does require natural domains (which is closer to the classical notion of computability in recursion theory).

From given representations of spaces we can easily construct representations of product and function spaces as we will see in the following. For coding purposes we will use *Cantor's pairing function* $\pi : \mathbb{N}^2 \to \mathbb{N}$, defined by

$$\pi(n, k) := \langle n, k \rangle := \frac{1}{2}(n + k)(n + k + 1) + k$$

This definition can be generalized inductively by $\langle n\rangle := n$ and for $i > 1$ by $\langle n_1, ..., n_{i+1}\rangle := \langle\langle n_1, ..., n_i\rangle, n_{i+1}\rangle$. Sometimes we will also use pairing functions of types $\mathbb{N}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, $(\mathbb{N}^{\mathbb{N}})^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$. More precisely, we define

$$\langle p, q\rangle(k) \;:=\; \begin{cases} p(n) & \text{if} \quad k = 2n \\ q(n) & \text{if} \quad k = 2n+1, \end{cases}$$

and $\langle p_0, p_1, ...\rangle\langle n, k\rangle := p_n(k)$. We will also use the notation $\langle p, n\rangle := np$ for $p \in \mathbb{N}^{\mathbb{N}}$ and $n \in \mathbb{N}$. It is well-known that all defined pairing functions, as well as the projections of their inverses, are computable. Now we are already prepared to define product and sequence space representations.

Definition 2.4 (Product and sequence spaces) Let (X, δ_X), (Y, δ_Y) be represented spaces.

(1) The *product representation* $[\delta_X, \delta_Y] :\subseteq \mathbb{N}^{\mathbb{N}} \to X \times Y$ is defined by $[\delta_X, \delta_Y]\langle p, q\rangle := (\delta_X(p), \delta_Y(q))$ for all $p, q \in \mathbb{N}^{\mathbb{N}}$.

(2) The *sequence representation* $\delta_X^\infty :\subseteq \mathbb{N}^{\mathbb{N}} \to X^{\mathbb{N}}$ of the set $X^{\mathbb{N}}$ of sequences $f : \mathbb{N} \to X$ is defined by $\delta_X^\infty\langle p_0, p_1, ...\rangle(n) := \delta_X(p_n)$, for all $p = \langle p_0, p_1, ...\rangle \in \mathbb{N}^{\mathbb{N}}, n \in \mathbb{N}$.

The product of representations can easily be generalized to finite products $[\delta_1, ..., \delta_n]$ by $[\delta] := \delta$ and $[\delta_1, ..., \delta_{n+1}] := [[\delta_1, ..., \delta_n], \delta_{n+1}]$. In case of $\delta = \delta_1 = \delta_2 = ... = \delta_n$ we write for short $\delta^n := [\delta_1, ..., \delta_n]$.

In the next step we want to define a representation $[\delta_X \to \delta_Y]$ of the set of (δ_X, δ_Y)–continuous functions. For this purpose we will first define a representation η of certain functions $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ and then we will use this representation to define $[\delta_X \to \delta_Y]$. For the definition of η we have to encode functions F by sequences $q \in \mathbb{N}^{\mathbb{N}}$. We introduce such an encoding in two steps. First we define a partial *multiplication* $* :\subseteq \mathbb{N}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ as follows[3]:

$$q * p = k :\iff (\exists t)\Big(q(\overline{p}t) = k + 1 \text{ and } (\forall s < t)(q(\overline{p}s) = 0)\Big).$$

In other words $q * p$ is defined, if and only if $t = \mu s[q(\overline{p}s) > 0]$ exists and in this case $q * p = q(\overline{p}t) - 1$. Using this partial function we can define the *application* $(.|.) :\subseteq \mathbb{N}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by $(q|p)(n) := q * \langle p, n\rangle$ for all $p, q \in \mathbb{N}^{\mathbb{N}}$ and $n \in \mathbb{N}$. Obviously, $(.|.)$ is continuous and thus, any $q \in \mathbb{N}^{\mathbb{N}}$ encodes a continuous function $\eta_q :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}, p \mapsto (q|p)$. One can prove that in this way a total representation η of the set of

[3]These definitions have been used for instance by Troelstra [55] and Bauer [4]

continuous functions $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ with G_δ–domain has been defined [4]. Moreover, a function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is computable, if and only if there is a computable $q \in \mathbb{N}^{\mathbb{N}}$ such that $F = \eta_q$. Now we are prepared to define the function space representation.

Definition 2.5 (Function spaces) Let (X, δ_X), (Y, δ_Y) be represented spaces. We define a representation $[\delta_X \to \delta_Y]$ of the set $\mathcal{C}(\delta_X, \delta_Y)$ of total (δ_X, δ_Y)–continuous functions $f : X \to Y$ by

$$[\delta_X \to \delta_Y](q) = f :\iff (\forall p \in \mathrm{dom}(\delta_X))\ \delta_Y \eta_q(p) = f\delta_X(p)$$

for all $q \in \mathbb{N}^{\mathbb{N}}$ and $f : X \to Y$.

The representation η has some nice properties: it fulfills a certain smn- and utm-theorem [64]. Or stated differently, the Baire space $\mathbb{N}^{\mathbb{N}}$ together with the application function $(.|.) :\subseteq \mathbb{N}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ forms a partial combinatory algebra [4]. With respect to $[\delta_X \to \delta_Y]$, these properties imply the following result on evaluation and type conversion.

Proposition 2.6 *Let (X, δ_X), (Y, δ_Y), (Z, δ_Z) be represented spaces.*

(1) **(Evaluation)** *Evaluation* $\mathrm{ev} : \mathcal{C}(\delta_X, \delta_Y) \times X \to Y, (f, x) \mapsto f(x)$ *is $([[\delta_X \to \delta_Y], \delta_X], \delta_Y)$–computable,*

(2) **(Type conversion)** $f : Z \times X \to Y$ *is $([\delta_Z, \delta_X], \delta_Y)$–computable, if and only if the transposed function $\check{f} : Z \to \mathcal{C}(\delta_X, \delta_Y)$, defined by $\check{f}(z)(x) := f(z, x)$, is $(\delta_Z, [\delta_X \to \delta_Y])$–computable.*

The evaluation property is already hidden in the definition of η while the "only if"-direction of the type conversion property requires a technical but straightforward proof. The "if"-direction follows from the evaluation property. As a consequence one obtains the following result on the category of represented spaces.

Corollary 2.7 *The category of represented spaces is cartesian closed.*

Here the objects are represented spaces, the morphisms are total functions which are continuous (or computable) with respect to the corresponding representations. Products and exponentials are defined with the help of product and function space representations.

3. Topologically Admissible Representations

Up to now we have considered arbitrary representations of sets. But sets like the real numbers $\mathbb{R}$ admit a large variety of different representations. Some of these representations are useful, other rather useless;

some representations fit together with an a priori defined topology on the represented set, others do not. In this section we will consider the class of *admissible* representations which are well-behaved in a topological sense. As a very useful tool for the comparison of representations we will use the following concept of reducibility.

Definition 3.1 (Reducibility) Let δ, δ' be representations.

(1) δ is *reducible* to δ', or $\delta \leq \delta'$ for short, if there is a computable function $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ with $\delta(p) = \delta' F(p)$ for all $p \in \mathrm{dom}(\delta)$.

(2) δ is *equivalent* to δ', or $\delta \equiv \delta'$ for short, if $\delta \leq \delta'$ and $\delta' \leq \delta$.

In other words, $\delta \leq \delta'$ holds, if and only if the identity id is (δ, δ')–computable. If we replace "computable" by "continuous" in the definition, then we obtain the corresponding concept of *continuous reducibility* which is denoted by "$\leq_t$". It is easy to prove that equivalent representations induce the same (strong) computability on represented sets. Moreover, the product operation on representations is associative up to equivalence, i.e. $[[\delta_1, \delta_2], \delta_3] \equiv [\delta_1, [\delta_2, \delta_3]]$ and equivalent representations have equivalent product, sequence and function space representations.

For the set $\mathbb{N}$ we will use the representation $\delta_\mathbb{N} : \mathbb{N}^\mathbb{N} \to \mathbb{N}$, which is defined by $\delta_\mathbb{N}(p) := p(0)$ for all $p \in \mathbb{N}^\mathbb{N}$. We will use the notation $\hat{n}$ for the constant sequence $p \in \mathbb{N}^\mathbb{N}$ with $p(i) = n$ for all $i \in \mathbb{N}$. In particular we obtain $\delta_\mathbb{N}(\hat{n}) = n$. A function $f :\subseteq \mathbb{N} \to \mathbb{N}$ is $(\delta_\mathbb{N}, \delta_\mathbb{N})$–computable, if and only if it is a restriction of a classically recursive function. For any representation δ we obtain $\delta^\infty \equiv [\delta_\mathbb{N} \to \delta]$. Thus, the sequence space representation is compatible with the function space representation.

A representation of a topological space is called *admissible*, if it is continuous and maximal among all continuous representations with respect to continuous reducibility. More precisely:

Definition 3.2 (Admissibility) A representation δ of a topological space X is called *admissible*, if δ is continuous and if $\delta' \leq_t \delta$ holds for all continuous representations δ' of X.

It is straightforward to define examples of admissible representations of second countable T_0–spaces and we will use the following standard representation for such spaces.

Proposition 3.3 (Standard representation) *Let X be a T_0–space with subbase $\{B_n : n \in \mathbb{N}\}$. The representation δ of X, defined by*

$$\delta(p) = x :\iff \mathrm{range}(p) = \{n \in \mathbb{N} : x \in B_n\}$$

for all $p \in \mathbb{N}^\mathbb{N}$ and $x \in X$, is admissible. It is called the standard representation *of the space X.*

Here, we assume that $\bigcup_{n=0}^{\infty} B_n = X$ for any subbase $\{B_n : n \in \mathbb{N}\}$ of X (such that for any point $x \in X$ there is some n with $x \in B_n$). The following result shows that the admissible representations of second countable T_0–spaces are essentially the continuous representations with surjective and open restrictions. Here, "essentially" means, that this does hold true only for a certain but large class of representations. A representation δ of a topological space X is called *complete*, if the following holds: whenever $\{\delta(p[i]\mathbb{N}^{\mathbb{N}}) : i \in \mathbb{N}\}$ is a neighbourhood base of some point $x \in X$, then $p \in \mathrm{dom}(\delta)$ and $\delta(p) = x$. For complete representations we obtain the following characterization of admissibility, which is due to Brattka and Hertling [13].

Theorem 3.4 *Let X be a second countable T_0–space and let δ be a complete representation of X. Then δ is admissible, if and only if δ is continuous and has a surjective and open restriction.*

Completeness is not a very strong restriction, since any representation can be completed in a way such that the properties continuity, openness and admissibility are preserved [13]. Admissible representations are very useful because of the Theorem of Kreitz and Weihrauch which shows that for admissible representations relative continuity coincides with ordinary continuity. This theorem has been proved in [33] for second countable T_0–spaces. We present a generalized version for arbitrary topological spaces, which is due to Schröder [49]. Here continuity is replaced by sequential continuity (a function is called *sequentially continuous*, if it maps convergent sequences into convergent sequences).

Theorem 3.5 (Kreitz and Weihrauch) *Let δ_X and δ_Y be admissible representations of topological spaces X and Y, respectively, and let $f :\subseteq X \to Y$ be a function. Then f is sequentially continuous, if and only if f is (δ_X, δ_Y)–continuous.*

Proof. Let $f :\subseteq X \to Y$ be sequentially continuous. Then $f\delta_X$ is sequentially continuous as well, since δ_X is continuous. Since the Baire space $\mathbb{N}^{\mathbb{N}}$ has a countable base, it is sequential[4] and hence $f\delta_X$ is even continuous. Hence, δ', defined by

$$\delta'(np) := \begin{cases} f\delta_X(p) & \text{if } n = 0 \\ \delta_Y(p) & \text{else} \end{cases}$$

is continuous as well and surjective by definition. Thus, δ' is a continuous representation of Y and we obtain $\delta' \leq_t \delta_Y$, since δ_Y is admissible.

[4]A topological space X is called *sequential*, if all subsets $U \subseteq X$ with the property that any sequence which converges to an element of U is eventually in U, are open.

Hence, there is a continuous function $G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that $\delta' = \delta_Y G$ and $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, defined by $F(p) := G(0p)$ is continuous as well and we obtain $\delta_Y F(p) = \delta_Y G(0p) = \delta'(0p) = f\delta_X(p)$ for all $p \in \mathrm{dom}(f\delta_X)$. Consequently, f is (δ_X, δ_Y)–continuous.

Now let $f :\subseteq X \to Y$ be (δ_X, δ_Y)–continuous. Then there exists a continuous function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that $f\delta_X(p) = \delta_Y F(p)$ for all $p \in \mathrm{dom}(f\delta_X)$. Let $x \in \mathrm{dom}(f)$ and let $(x_n)_{n\in\mathbb{N}}$ be a sequence in $\mathrm{dom}(f)$ which converges to x. Now we define a representation δ' of X by

$$\delta'(np) := \begin{cases} x & \text{if } n = 0 \text{ and } p = \hat{0} = 000... \\ x_k & \text{if } n = 0 \text{ and } p = 0^k\hat{1} = 0^k 111... \\ \delta_X(p) & \text{if } n \neq 0 \end{cases}.$$

Then δ' is a continuous representation of X and we obtain $\delta' \leq_t \delta_X$ since δ_X is admissible. Hence, there is a continuous function $G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that $\delta' = \delta_X G$. Let U be an open neighbourhood of $f(x)$. Since $\delta_Y FG$ is continuous and $\delta_Y FG(\hat{0}) = f\delta_X G(\hat{0}) = f\delta'(\hat{0}) = f(x) \in U$, there is some $k \in \mathbb{N}$ with $\delta_Y FG(0^k\mathbb{N}^{\mathbb{N}}) \subseteq U$. For all $n \geq k$ we obtain

$$f(x_n) = f\delta'(0^{n+1}\hat{1}) = f\delta_X G(0^{n+1}\hat{1}) = \delta_Y FG(0^{n+1}\hat{1}) \in U.$$

Thus, $(f(x_n))_{n\in\mathbb{N}}$ converges to $f(x)$ and f is sequentially continuous. $\square$

Second countable T_0–spaces are sequential and in this case ordinary continuity and sequential continuity coincide. Especially, $\mathcal{C}(\delta_X, \delta_Y)$ is equal to the set $\mathcal{C}(X, Y)$ of continuous functions $f : X \to Y$ if δ_X, δ_Y are admissible representations of sequential spaces. If one tries to generalize Corollary 2.7 to categories of admissibly represented topological spaces, then one has to face the problem that even many categories of topological spaces are not cartesian closed. One has to choose either a subcategory of the category of all topological spaces (as the category of sequential topological spaces, see Schröder [49]) or a larger category (as the category of limit spaces, see Schröder [48]). We formulate the first result.

Theorem 3.6 (Schröder) *The category of admissibly represented sequential T_0–spaces is cartesian closed.*

Again the morphisms are the total continuous functions and products and exponentials of this category are just given by the corresponding product and function space operations on representations, as defined in the previous section.[5]

[5]The considerations in this section show that the category of admissibly represented sequential T_0–spaces is suitable for semantic purposes. Results of Bauer [4, 5], show that this category is closely related to a subcategory of equilogical spaces, as proposed by Dana Scott.

4. Real Number Representations

In this section we briefly discuss some real number representations. Such representations have implicitly been used for a long time. For instance the identities

$$\pi = 3 + \cfrac{1}{7 + \cfrac{1}{15 + \cfrac{1}{1 + \cfrac{1}{292 + \cfrac{1}{1 + \dots}}}}} = 3.14159265358979323846264338 3...$$

refer to the representations of the number π as *normed continued fraction* and as *decimal expansion*. It is obvious how such ideas can be formalized to define representations $\delta :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{R}$. We define some other representations formally. Therefore, we will use a numbering $\nu_{\mathbb{Q}} : \mathbb{N} \to \mathbb{Q}$ of the rational numbers $\mathbb{Q}$, defined by $\nu_{\mathbb{Q}}\langle i, j, k\rangle := \frac{i-j}{k+1}$. We start with the definition of the Cauchy representation.

Definition 4.1 (Cauchy representation) The *Cauchy representation* ρ of the real numbers $\mathbb{R}$ is defined by

$$\rho(p) := \lim_{n\to\infty} \nu_{\mathbb{Q}} p(n)$$

for all $p \in \mathbb{N}^{\mathbb{N}}$ such that $|\nu_{\mathbb{Q}} p(i) - \nu_{\mathbb{Q}} p(k)| \leq 2^{-k}$ for $i > k$.

Intuitively, $\rho(p) = x$, if p is an (encoded) Cauchy sequence of rational numbers which rapidly converges to x. In the following we will call the (ρ^n, ρ)–computable functions $f :\subseteq \mathbb{R}^n \to \mathbb{R}$ just *computable*. Many well-known real number functions, such as the arithmetic operations, the trigonometric functions and many others, are computable [64]. The $(\delta_{\mathbb{N}}, \rho)$–computable sequences and the ρ–computable real numbers will be called *computable* for short. The Cauchy representation characterizes the most important equivalence class of real number representations. For certain purposes other representations are of interest.[6] We define two of them, the representations by enumeration of left and right cuts.

Definition 4.2 (Left and right representations) We define representations $\rho_<, \rho_>$ of the real numbers $\mathbb{R}$ by

(1) $\rho_<(p) = x :\iff \{\nu_{\mathbb{Q}} p(n) : n \in \mathbb{N}\} = \{r \in \mathbb{Q} : r < x\}$,

(2) $\rho_>(p) = x :\iff \{\nu_{\mathbb{Q}} p(n) : n \in \mathbb{N}\} = \{r \in \mathbb{Q} : r > x\}$.

[6] As mentioned in the introduction, the decimal representation is not very suitable: its final topology is the Euclidean topology, but addition is not continuous with respect to the decimal representation [64]. Hence, the decimal representation is not admissible with respect to any topology.

Intuitively, $\rho_<(p) = x$, if and only if p is an (encoded) list of all rational numbers which are less than x. A corresponding statement holds for $\rho_>$. The $\rho_<$– and $\rho_>$–computable numbers $x \in \mathbb{R}$ are called *left computable* and *right computable*, respectively. The $(\rho^n, \rho_<)$– and $(\rho^n, \rho_>)$–computable functions are called *lower semi-computable* and *upper semi–computable*, respectively. It is easy to see that the lower and upper semi–computable functions are lower and upper semi–continuous, respectively. The defined representations $\rho, \rho_<, \rho_>$ are admissible with respect to certain topologies.

Proposition 4.3 *The representations $\rho, \rho_<$ and $\rho_>$ are admissible with respect to the topologies on $\mathbb{R}$, which are generated by the open intervals (r, s), (r, ∞) and $(-\infty, s)$, respectively.*

A proof can be found in [64]. Figure 2 displays reducibility relations of certain real number representations.

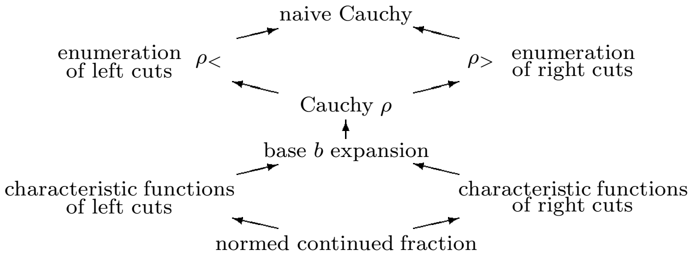

Figure 2. The lattice of some real number representations

Here, any arrow $\delta \to \delta'$ means that δ is computably reducible to δ', i.e. $\delta \leq \delta'$, but δ' is not even continuously reducible to δ, i.e. $\delta' \not\leq_t \delta$. Up to transitivity the diagram is complete.[7] The classes of computable numbers induced by the representations in Figure 2 which are located below the Cauchy representation ρ, coincide with the computable real numbers. This is not the case for the representations above ρ.[8]

[7]A comprehensive comparison of real number representations can be found in [17] and [64]. It is interesting to mention that the representation with base b expansion is reducible to the representation with base a expansion, if and only if all prime divisors of a are also prime divisors of b.

[8]The *naive Cauchy representation* is defined as the Cauchy representation but without any restrictions concerning the speed of convergence. It is quite useless, since no finite prefix of such a Cauchy sequence gives any information on the represented number. For further details, see [13]. The class of computable real numbers induced by the naive Cauchy representation is the class of *recursively approximable real numbers*, see [2, 67].

5. Computable Metric Spaces

In this section we present the concept of a *computable metric space*.[9] First, we introduce some notations. By $B(x, \varepsilon) := \{y \in X : d(x, y) < \varepsilon\}$ we denote the *open balls* and by $\overline{B}(x, \varepsilon) := \{y \in X : d(x, y) \leq \varepsilon\}$ the corresponding *closed balls* of a metric space (X, d). Moreover, we use the abbreviation $\overline{n} := \nu_{\mathbb{Q}}(n)$ for the rational numbers given by $n \in \mathbb{N}$. We will say that a sequence $\alpha : \mathbb{N} \to X$ is *dense* in X, if the topological closure of range(α) is equal to X.

Definition 5.1 (Computable metric space) We call (X, d, α) a *computable metric space*, if

(1) $d : X \times X \to \mathbb{R}$ is a metric on X,

(2) $\alpha : \mathbb{N} \to X$ is dense in X,

(3) $d \circ (\alpha \times \alpha) : \mathbb{N}^2 \to \mathbb{R}$ is a computable (double) sequence.

If (X, d, α) just fulfills (1) and (2), then it is sometimes called an *effective metric space* (which is nothing but a separable metric space together with a numbering of a dense subset). For simplicity, we will simply say that (X, d, α) is a *separable metric space* in this case. The space $(\mathbb{R}, d, \nu_{\mathbb{Q}})$ with the *Euclidean metric* $d(x, y) := |x-y|$ is an example of a computable metric space. Similar as in the real number case, we can introduce the *Cauchy representation* of separable metric spaces in general.

Definition 5.2 (Cauchy representation) Let (X, d, α) be a separable metric space. The *Cauchy representation* δ_X of X is defined by

$$\delta_X(p) := \lim_{n \to \infty} \alpha p(n)$$

for all $p \in \mathbb{N}^{\mathbb{N}}$ such that $d(\alpha p(i), \alpha p(k)) \leq 2^{-k}$ for $i > k$ and $(\alpha p(n))_{n \in \mathbb{N}}$ is convergent.

In case of $(\mathbb{R}, d, \nu_{\mathbb{Q}})$ we obtain $\delta_{\mathbb{R}} = \rho$. The following result characterizes the equivalence class of the Cauchy representation of a computable metric space with the help of the metric $d : X \times X \to \mathbb{R}$ and the *limit operator*

$$\mathrm{Lim} :\subseteq X^{\mathbb{N}} \to X, (x_n)_{n \in \mathbb{N}} \mapsto \lim_{n \to \infty} x_n,$$

[9]These spaces, sometimes called *recursive metric spaces*, have been used with similar definitions by Lacombe [38], Ceĭtin [16], Moschovakis [41], Weihrauch [63], Spreen [53] and many others. Some of these authors have studied metric spaces restricted to computable points.

which is restricted to rapidly converging sequences, i.e. dom(Lim) contains only convergent sequences $(x_n)_{n\in\mathbb{N}}$ such that $d(x_i, x_k) \leq 2^{-k}$ for all $i > k$.

Proposition 5.3 (Cauchy representation) *Let (X, d, α) be a computable metric space with Cauchy representation δ_X and let δ be a further representation of X. Then $\alpha : \mathbb{N} \to X$ is $(\delta_\mathbb{N}, \delta_X)$–computable and*

(1) $\delta \leq \delta_X \iff d : X \times X \to \mathbb{R}$ is $([\delta, \delta_X], \delta_\mathbb{R})$–computable,

(2) $\delta_X \leq \delta \iff \mathrm{Lim} :\subseteq X^\mathbb{N} \to X$ is $(\delta_X^\infty, \delta)$–computable.

Proof. Define $F : \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ by $F(p)(n) := p(0)$ for all $p \in \mathbb{N}^\mathbb{N}$ and $n \in \mathbb{N}$. Then F is computable and $\delta_X F(p) = \alpha p(0) = \alpha\delta_\mathbb{N}(p)$ for all $p \in \mathbb{N}^\mathbb{N}$. Thus, α is $(\delta_\mathbb{N}, \delta_X)$–computable via F.

(1) Let $\delta \leq \delta_X$. It suffices to prove that d is $([\delta_X, \delta_X], \delta_\mathbb{R})$–computable. Since $d \circ (\alpha \times \alpha) : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$ is computable, we can effectively find some $n_k \in \mathbb{N}$ with $\overline{n_k} < d(\alpha p(k+3), \alpha q(k+3)) < \overline{n_k} + 2^{-k-2}$ for any given $p, q \in \mathrm{dom}(\delta_X)$ and $k \in \mathbb{N}$. Thus, for $x := \delta_X(p)$ and $y := \delta_X(q)$ we obtain

$$\begin{array}{rl} & |d(x,y) - \overline{n_k}| \\ \leq & d(x, \alpha p(k+3)) + d(y, \alpha q(k+3)) + |d(\alpha p(k+3), \alpha q(k+3)) - \overline{n_k}| \\ \leq & 2^{-k-3} + 2^{-k-3} + 2^{-k-2} = 2^{-k-1}. \end{array}$$

Let $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ be the computable function which associates a corresponding value $n_k = F\langle p, q\rangle(k)$ with each $p, q \in \mathrm{dom}(\delta_X)$ and $k \in \mathbb{N}$. Then $\delta_\mathbb{R} F\langle p, q\rangle = \lim_{k\to\infty} \overline{n_k} = d(x, y) = d \circ [\delta_X, \delta_X]\langle p, q\rangle$. Thus, d is $([\delta_X, \delta_X], \delta_\mathbb{R})$–computable via F.

Now let d be $([\delta, \delta_X], \delta_\mathbb{R})$–computable. Since α is $(\delta_\mathbb{N}, \delta_X)$–computable, there is a computable function $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ which is defined for all $p \in \mathrm{dom}(\delta)$ such that $F(p)(k) = n \Longrightarrow d(\delta(p), \alpha(n)) < 2^{-k-1}$ for all $p \in \mathrm{dom}(\delta)$, $k, n \in \mathbb{N}$. Thus $\delta(p) = \lim_{k\to\infty} \alpha F(p)(k) = \delta_X F(p)$ for all $p \in \mathrm{dom}(\delta)$, i.e. $\delta \leq \delta_X$.

(2) Let $\delta_X \leq \delta$. It suffices to prove that Lim is $(\delta_X^\infty, \delta_X)$–computable. Define $F : \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ by $F\langle p_0, p_1, ...\rangle(k) := p_{k+2}(k+2)$. Then F is computable. Now let $p = \langle p_0, p_1, ...\rangle \in \mathrm{dom}(\mathrm{Lim}\,\delta_X^\infty)$. Then we obtain for all $n > k$

$$\begin{array}{rl} & d(\alpha p_{n+2}(n+2), \alpha p_{k+2}(k+2)) \\ \leq & d(\alpha p_{n+2}(n+2), \alpha p_{n+2}(k+2)) + d(\alpha p_{n+2}(k+2), \delta_X(p_{n+2})) \\ & + d(\delta_X(p_{n+2}), \delta_X(p_{k+2})) + d(\delta_X(p_{k+2}), \alpha p_{k+2}(k+2)) \\ \leq & 2^{-k-2} + 2^{-k-2} + 2^{-k-2} + 2^{-k-2} = 2^{-k} \end{array}$$

and thus $\delta_X F(p) = \lim_{k\to\infty} \alpha p_{k+2}(k+2) = \lim_{n\to\infty} \lim_{k\to\infty} \alpha p_n(k) = \lim_{n\to\infty} \delta_X(p_n) = \mathrm{Lim} \circ \delta_X^\infty(p)$ for all $p = \langle p_0, p_1, ...\rangle \in \mathrm{dom}(\mathrm{Lim}\,\delta_X^\infty)$. Consequently, Lim is $(\delta_X^\infty, \delta_X)$–computable via F.

Now let Lim be $(\delta_X^\infty, \delta)$–computable. Since α is $(\delta_\mathbb{N}, \delta_X)$–computable and $\delta_X = \mathrm{Lim} \circ \alpha^\mathbb{N}$, it follows that δ_X is $(\delta_\mathbb{N}^\infty, \delta)$–computable. Since $\delta_\mathbb{N}^\infty \equiv \mathrm{id}_{\mathbb{N}^\mathbb{N}}$, we obtain $\delta_X \leq \delta$. □

This result especially shows that the metric d and the limit operator Lim are computable for any computable metric space (X, d, α). Moreover, the result shows that the limit operation can be used to "synthesize" and the metric can be used to "analyze" a metric space.[10]

If X and Y are computable metric spaces with Cauchy representations δ_X and δ_Y, respectively, then we will call a function $f :\subseteq X \to Y$ *computable*, if it is (δ_X, δ_Y)–computable. Similarly as in the real number case, we will call a function $f :\subseteq X \to \mathbb{R}$ *lower semi-computable* and *upper semi-computable*, if it is $(\delta_X, \rho_<)$– and $(\delta_X, \rho_>)$–computable, respectively. A sequence $s : \mathbb{N} \to X$ is called *computable*, if it is $(\delta_\mathbb{N}, \delta_X)$–computable, and a point $x \in X$ is called *computable*, if the constant sequence with value x is computable. In the real number case it is well-known that the set of computable points cannot be enumerated effectively. The same holds for complete computable metric spaces without isolated points. The result can be derived from an effective version of the Baire Category Theorem [11].

Proposition 5.4 *If X is a computable complete metric space without isolated points, then there exists no computable sequence $(x_n)_{n\in\mathbb{N}}$ in X such that $\{x_n : n \in \mathbb{N}\}$ is the set of computable points of X.*

An alternative approach to computable analysis on Banach spaces has been proposed by Pour-El and Richards [45]. In their approach, the notion of a computable sequence has been axiomatized. Yasugi, Mori and Tsujii have presented a generalization of this approach for metric spaces [66]. The approach presented in this section is compatible with these ideas: the set of computable sequences of a computable metric space fulfills the axioms of a computability structure in sense of [66]. We close this section with some examples of computable metric spaces. For the Cauchy representation $\delta_{\mathbb{R}^n}$, associated with the first example, we obtain $\delta_{\mathbb{R}^n} \equiv \rho^n$.

[10] A corresponding result cannot be obtained for non-separable metric spaces, since one can show that any metric space (X, d) with a representation δ such that d is $([\delta, \delta], \rho_>)$–continuous, necessarily is separable (see Lemma 8.1.1 in [64]).

Example 5.5 (Computable metric spaces)

(1) $(\mathbb{R}^n, d_{\mathbb{R}^n}, \alpha_{\mathbb{R}^n})$ with the *Euclidean metric*

$$d_{\mathbb{R}^n}(x,y) := \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$$

and some standard enumeration $\alpha_{\mathbb{R}^n}$ of all rational points $\mathbb{Q}^n$ is a computable metric space.

(2) $(\mathcal{C}[0,1], d_{\mathcal{C}}, \alpha_{\mathcal{C}})$ with the set $\mathcal{C}[0,1]$ of continuous real-valued functions $f : [0,1] \to \mathbb{R}$ and the *supremum metric*

$$d_{\mathcal{C}}(f,g) := ||f - g|| := \sup_{x \in [0,1]} |f(x) - g(x)|$$

and some standard numbering $\alpha_{\mathcal{C}}$ of the rational polynomials $\mathbb{Q}[x]$ is a computable metric space. The computable points in this space are exactly the computable functions $f : [0,1] \to \mathbb{R}$.

(3) $(\mathcal{K}(X), d_{\mathcal{K}}, \alpha_{\mathcal{K}})$ with the set $\mathcal{K}(X)$ of non-empty compact subsets of a computable metric space (X, d, α) and the *Hausdorff metric*

$$d_{\mathcal{K}}(A,B) := \max\left\{\sup_{a \in A} \inf_{b \in B} d(a,b), \sup_{b \in B} \inf_{a \in A} d(a,b)\right\}$$

and some standard numbering $\alpha_{\mathcal{K}}$ of the non-empty finite subsets of $\mathrm{range}(\alpha)$ is a computable metric space.

6. Computable Subsets of Metric Spaces

In the previous section we have defined computable points, sequences and functions on metric spaces. In this section we want to define computable subsets of metric spaces. Essentially, we want to extend the classical notions of recursively enumerable, co-recursively enumerable and recursive subsets, as they are used in recursion theory [44], to this setting. It turns out that the intuition of decidability is not very fruitful in computable analysis: if (X, d) is a connected metric space, then the *characteristic function*

$$\mathrm{cf}_A : X \to \mathbb{R}, x \mapsto \begin{cases} 0 & \text{if } x \in A \\ 1 & \text{else} \end{cases}$$

of a subset $A \subseteq X$ is continuous (and hence computable) only in the trivial cases $A \in \{\emptyset, X\}$. The solution of this problem is to replace the discontinuous characteristic function by a continuous substitute. Therefore, one can use the *distance function*

$$d_A : X \to \mathbb{R}, x \mapsto \inf_{a \in A} d(x, a),$$

defined for any metric space (X, d) and subsets $A \subseteq X$. Figure 3 displays the characteristic function of the unit interval $A = [0, 1]$ together with its distance function.

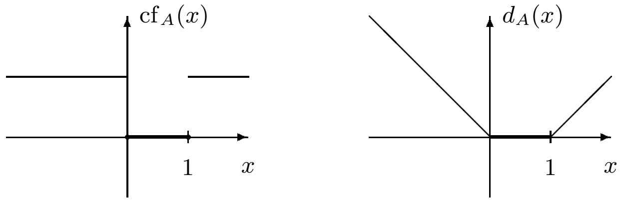

Figure 3. Characteristic function and distance function of the unit interval

In the following definition we use different computability properties of the distance function to define computability notions for subsets.[11]

Definition 6.1 (Located subsets) Let (X, d, α) be a computable metric space and let $A \subseteq X$ be a subset. Then A is called *lower semi-located, upper semi-located* and *located*, if $d_A : X \to \mathbb{R}$ is lower semi-computable, upper semi-computable and computable, respectively.

Since the distance function of a subset $A \subseteq X$ coincides with the distance function of the topological closure $\overline{A}$ of A, we can restrict most considerations about located subsets to closed subsets. For closed subsets (or dually, for open subsets) the following notion of effectivity has already been considered by Lacombe [37].

Definition 6.2 (Co-r.e. closed subsets) Let (X, d, α) be a computable metric space and let $A \subseteq X$ be a closed subset. Then A is called *co-r.e. closed*, if there exists an r.e. subset $C \subseteq \mathbb{N}$ such that

$$X \setminus A = \bigcup_{\langle n,k \rangle \in C} B(\alpha(n), \overline{k})$$

It is easy to show that a subset $A \subseteq X$ is co-r.e. closed, if and only if there exists a computable function $f : X \to \mathbb{R}$ such that $f^{-1}\{0\} = A$ and

[11]The idea to use distance functions in order to define *located* subsets goes back to intuitionistic and constructive analysis, see for instance [6]. They have been used in computable analysis by several authors, see for instance [65, 23, 68].

this is the case, if and only if the characteristic function $\mathrm{cf}_A : X \to A$ is lower semi-computable. Another notion which has been considered in some references is the notion of *effective separability*.

Definition 6.3 (Effective separability) Let (X, d, α) be a computable metric space and let $A \subseteq X$ be a subset. Then A is called *effectively separable*, if there exists a computable sequence $f : \mathbb{N} \to X$ such that $\mathrm{range}(f) = \{f(n) : n \in \mathbb{N}\}$ is dense in A.

Finally, we even define four further notions of effectivity.

Definition 6.4 (Recursive subsets) Let (X, d, α) be a computable metric space and let $A \subseteq X$ be a closed subset.

(1) A is called *recursively enumerable closed* (*r.e. closed* for short), if the set $\{\langle n, k\rangle \in \mathbb{N} : A \cap B(\alpha(n), \overline{k}) \neq \emptyset\}$ is r.e.

(2) A is called *recursive closed*, if it is r.e. closed and co-r.e. closed.

(3) A is called *strongly co-recursively enumerable closed* (*strongly co-r.e. closed* for short), if the set $\{\langle n, k\rangle \in \mathbb{N} : A \cap \overline{B}(\alpha(n), \overline{k}) = \emptyset\}$ is r.e.

(4) A is called *strongly recursive closed*, if A is r.e. closed, as well as strongly co-r.e. closed.

Figure 4 shows the logical relations between all introduced notions of effectivity for closed subsets, see [14] for proofs.

$$\begin{tikzcd}
 & \text{recursive} \arrow[r] \arrow[dl] & \text{co-r.e.} \\
\begin{array}{c}\text{r.e. =}\\ \text{upper semi-located}\end{array} & \text{located} \arrow[l] \arrow[u] \arrow[r] & \text{lower semi-located} \arrow[u] \\
\text{effectively separable} \arrow[u] & \text{strongly recursive} \arrow[u] \arrow[r] & \text{strongly co-r.e.} \arrow[u]
\end{tikzcd}$$

Figure 4. Notions of effectivity for closed subsets of computable metric spaces

Each arrow in the diagram does not only indicate an implication but an effective reducibility: for each notion of effectivity, there exists a corresponding hyperspace representation $\delta :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathcal{A}(X)$ of the set $\mathcal{A}(X)$ of closed subsets $A \subseteq X$. In this case, the arrows in Figure 4 can be read as reducibilities, similarly as the arrows in Figure 2.

In case of Euclidean space (and other "effectively locally compact" computable metric spaces) the vertical arrows in Figure 4 can be reversed and thus the three horizontal layers of the diagram collapse [15]. Moreover, the recursive compact subsets $K \subseteq \mathbb{R}^n$ are exactly the computable points in $\mathcal{K}(\mathbb{R}^n)$. In contrast to that, the three vertical layers represent essentially different notions of effectivity which generalize the classical notions of effectivity, as the following result shows.

Proposition 6.5 *A subset $A \subseteq \mathbb{N}$ is r.e., co-r.e., or recursive in the classical sense, if and only if A is r.e., co-r.e., or recursive, respectively, considered as a closed subset of Euclidean space $\mathbb{R}$.*

7. Computable Multi-Valued Operations

In this section we want to present an extension of the representation based notion of computability to multi-valued operations. There are at least two motivations for such an extension. On the one hand, multi-valued operations occur in practice and examples like the determination of zeroes of polynomials are important and unavoidable: given the coefficients of a non-constant polynomial, we can effectively find a zero, but in general the corresponding computation is non-extensional and therefore it has to be described by a multi-valued function.[12] Such *indeterministic computations*[13] may lead to different results on the same input, but all possible results have to be valid (in contrast to *non-deterministic* computations in complexity theory, where only some computations have to yield a valid result). On the other hand, one can motivate the introduction of multi-valued operations by formal reasons. If the arrows in the category of admissibly represented spaces are chosen to be multi-valued, then the category gains some nice properties, as we will implicitly see below.

Before we define computability of multi-valued operations, we introduce some notations. By $f :\subseteq X \rightrightarrows Y$ we will denote partial multi-valued functions which we will call for short *operations* in the following. Here the symbol "$\rightrightarrows$" indicates that f might be multi-valued. More precisely, an operation $f :\subseteq X \rightrightarrows Y$ is a correspondence $f = (\Phi, X, Y)$, that is $\Phi \subseteq X \times Y$. We will use these objects from an operational point of view, that is X is considered as a space of inputs and Y as a space of outputs. We will use some notations for operations: $\mathrm{graph}(f) := \Phi$, $\mathrm{dom}(f) := \{x \in X : (\exists y \in Y)\ (x, y) \in \Phi\}$, and $\mathrm{range}(f) := \{y \in Y : (\exists x \in X)\ (x, y) \in \Phi\}$ will be called *graph*, *domain*, and *range* of f, respectively. The *image* of $A \subseteq X$ under f will be denoted by $f(A) := \{y \in Y : (\exists x \in A)\ (x, y) \in \Phi\}$, and the *preimage* of $B \subseteq Y$ by $f^{-1}(B) := \{x \in X : (\exists y \in B)\ (x, y) \in \Phi\}$. By $f(x) := f\{x\} = \{y \in Y : (x, y) \in \Phi\}$ we denote the *image* of x un-

[12]The idea to compute with multi-valued operations has been considered by several authors in different settings [42, 18, 52, 31].
[13]This notion of indeterminism has already been used by Shepherdson [50].

der f for each $x \in \mathrm{dom}(f)$. If $f(x)$ is single-valued, i.e. $f(x) = \{y\}$ for some $y \in Y$, then we also write $f(x) = y$, as usual for functions. With each operation $f = (\Phi, X, Y)$ we associate the *inverse operation* $f^{-1} = (\Phi^{-1}, Y, X)$, which is given by $\Phi^{-1} := \{(y, x) : (x, y) \in \Phi\}$.

Computable multi-valued operations $f :\subseteq X \rightrightarrows Y$ are usually defined by substituting "$\delta_Y F(p) \in f\delta_X(p)$" for the condition "$\delta_Y F(p) = f\delta_X(p)$" in Definition 2.3 (see [64]). We will call this kind of computability *weak* (δ_X, δ_Y)*–computability*. Weak computability corresponds to the idea that the realization F on input p with $x = \delta_X(p)$ selects one of the possible results in $f(x)$. For our purposes, however, this definition is too weak, since the realization F does not characterize the whole image $f(x)$. In a second attempt, we could define that f is (δ_X, δ_Y)–computable, if there is a computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that $f(x) = \delta_Y F \delta_X^{-1}\{x\}$ for all $x \in \mathrm{dom}(f)$. But this definition would sensitively rely on the concrete choice of the representation δ_X of X. It is possible that for some representations δ_1 of X the preimage $\delta_1^{-1}\{x\}$ for some specific x is single-valued while for other equivalent representations δ_2 of X the preimage $\delta_2^{-1}\{x\}$ might be very large. Consequently, equivalent representations would induce different kinds of computability of operations. This is one reason why we choose a different definition of computable operations. In our definition the complete image $f(x)$ will be obtained by the help of an additional "oracle" input.

Definition 7.1 (Computable operations) Let (X, δ_X), (Y, δ_Y) be represented spaces. Then $f :\subseteq X \rightrightarrows Y$ is called a (δ_X, δ_Y)*–computable operation*, if there is a computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that

$$\delta_Y F\langle p, \mathbb{N}^{\mathbb{N}}\rangle = f\delta_X(p)$$

and $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}(\delta_Y F)$ for all $p \in \mathrm{dom}(f\delta_X)$. Furthermore, f is called a *strongly* (δ_X, δ_Y)*–computable operation*, if, additionally, the condition $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \not\subseteq \mathrm{dom}(F)$ holds for all $p \in \mathrm{dom}(\delta_X) \setminus \mathrm{dom}(f\delta_X)$.

Here, $\langle p, \mathbb{N}^{\mathbb{N}}\rangle := \{\langle p, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\}$. It is easy to see that the previous definition has the property that equivalent representations induce the same kind of (strong) computability of operations. Moreover, a function $f :\subseteq X \to Y$ is (δ_X, δ_Y)–computable, if and only if it is (δ_X, δ_Y)–computable, considered as an operation. And if f is strongly (δ_X, δ_Y)–computable as a function, then it is also strongly (δ_X, δ_Y)–computable as an operation. However, the converse of the latter statement does not hold true in general. Therefore, in case of doubts, strong computability will always mean strong computability in the sense of operations. Our next goal is to show that weak computability and computability of operations are also closely related notions. The following lemma will be helpful for this purpose.

Lemma 7.2 (Oracle Lemma) *There exists a total computable function $H : \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ such that*

(1) $\mathrm{range}(H\langle p, q\rangle) = \mathrm{range}(p)$ *for all* $p, q \in \mathbb{N}^\mathbb{N}$*,*

(2) for all $r, p \in \mathbb{N}^\mathbb{N}$ *with* $\mathrm{range}(r) = \mathrm{range}(p)$ *there exists a* $q \in \mathbb{N}^\mathbb{N}$ *such that* $H\langle p, q\rangle = r$*.*

Proof. For each $p, q \in \mathbb{N}^\mathbb{N}$ we define $r = H\langle p, q\rangle$ by an inductive construction. In step $n = 0, 1, ...$ let $\langle i, j\rangle := q(n)$.

(a) If there is no k such that $r(k)$ is already defined and equal to $p(n)$, then choose the first $k \geq i$ such that $r(k)$ is still undefined and let $r(k) := p(n)$.

(b) If $r(n)$ is still undefined, then let $r(n) := p(j)$.

The algorithm, described hereby, defines a total computable function $H : \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$. After execution of step n, at least $r(0), ..., r(n)$ are defined by (b). We have to show that (1) and (2) hold.

(1) The construction obviously guarantees $\mathrm{range}(r) \subseteq \mathrm{range}(p)$. On the other hand, for each n there is some k such that $r(k) = p(n)$ after execution of step n, part (a). Hence, $\mathrm{range}(p) \subseteq \mathrm{range}(r)$.

(2) Now let $r, p \in \mathbb{N}^\mathbb{N}$ be such that $\mathrm{range}(r) = \mathrm{range}(p)$. Then define $q \in \mathbb{N}^\mathbb{N}$ for each $n \in \mathbb{N}$ by $q(n) := \langle i, j\rangle$, where $i := \mu k[r(k) = p(n)]$ and $j := \mu k[p(k) = r(n)]$. Then $H\langle p, q\rangle = r$. □

It is obvious that any (δ_X, δ_Y)–computable operation $f :\subseteq X \rightrightarrows Y$ is also weakly (δ_X, δ_Y)–computable. The following result shows that a certain converse version of this statement holds true as well.

Proposition 7.3 *Let X be a second countable T_0–space with standard representation δ_X and let (Y, δ_Y) be a represented space. If an operation $f :\subseteq X \rightrightarrows Y$ is weakly (δ_X, δ_Y)–computable, then there is a (δ_X, δ_Y)–computable operation $g :\subseteq X \rightrightarrows Y$ such that* $\mathrm{dom}(g) = \mathrm{dom}(f)$ *and* $\mathrm{graph}(g) \subseteq \mathrm{graph}(f)$.

Proof. Let $f :\subseteq X \rightrightarrows Y$ be weakly (δ_X, δ_Y)–computable. Then there is a computable function $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ such that $\delta_Y F(p) \in f\delta_X(p)$ for all $p \in \mathrm{dom}(f\delta_X)$. Let $H : \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ be the computable function which exists by the Oracle Lemma 7.2. Define $G :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$

by $G\langle p,q\rangle := FH\langle p,q\rangle$ for all $p,q \in \mathbb{N}^\mathbb{N}$. Then G is a computable function. Now let $g :\subseteq X \rightrightarrows Y$ be defined by $\mathrm{dom}(g) := \mathrm{dom}(f)$ and by $g\delta_X(p) := \{\delta_Y G\langle p,q\rangle : q \in \mathbb{N}^\mathbb{N}\}$ for all $p \in \delta_X^{-1}(\mathrm{dom}(g))$. By definition of H and δ_X we obtain $\delta_X^{-1}\delta_X(p) = \{H\langle p,q\rangle : q \in \mathbb{N}^\mathbb{N}\}$, i.e. g is well-defined and (δ_X,δ_Y)–computable. Now let $x \in \mathrm{dom}(g)$ and $y \in g(x)$. Then there are $p,q \in \mathbb{N}^\mathbb{N}$ such that $\delta_X(p) = x$ and $\delta_Y G\langle p,q\rangle = y$. Hence $y = \delta_Y FH\langle p,q\rangle \in f\delta_X H\langle p,q\rangle = f\delta_X(p) = f(x)$. Thus, $\mathrm{graph}(g) \subseteq \mathrm{graph}(f)$. □

We will close this section with a short discussion of continuity. It turns out that computable multi-valued operations are *lower semi-continuous*. As usual, a multi-valued operation $f :\subseteq X \rightrightarrows Y$ on topological spaces X, Y is called *lower semi-continuous*,[14] if $f^{-1}(U)$ is open in $\mathrm{dom}(f)$ for any open subset $U \subseteq Y$. If f is single-valued, then it is lower semi-continuous, if and only if it is continuous in the ordinary sense. Before we formulate our result we introduce another notation. With each representation $\delta_X :\subseteq \mathbb{N}^\mathbb{N} \to X$ we can associate its *cylindrification* $\delta^{\mathrm{cyl}} :\subseteq \mathbb{N}^\mathbb{N} \to X$, which is defined by $\delta^{\mathrm{cyl}}\langle p,q\rangle := \delta(p)$ for all $p,q \in \mathbb{N}^\mathbb{N}$ (cf. [62]). Obviously, each representation is equivalent to its cylindrification.

Theorem 7.4 (Continuity) *Let (X,δ_X), (Y,δ_Y) be admissibly represented second countable T_0–spaces. Then any (δ_X,δ_Y)–computable multi-valued operation $f :\subseteq X \rightrightarrows Y$ is lower semi-continuous.*

Proof. Without loss of generality, we can assume that δ_X, δ_Y are standard representations. Now let $f :\subseteq X \rightrightarrows Y$ be (δ_X,δ_Y)–computable via some computable function $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$. It is easy to see that δ_Y is continuous and that δ_X is open. The latter fact implies that $\delta_X^{-1} : X \rightrightarrows \mathbb{N}^\mathbb{N}$ is lower semi-continuous. Then $g :\subseteq X \rightrightarrows Y$, defined by $g(x) := \delta_Y \circ F \circ \langle \delta_X^{-1}(x), \mathbb{N}^\mathbb{N}\rangle$ is an extension of f, that is $g(x) = f(x)$ for all $x \in \mathrm{dom}(f)$. Altogether, it follows that g is lower semi-continuous and hence f, as a restriction of g, is lower semi-continuous as well.[15] □

[14] A corresponding notion of *upper semi-continuity* is defined with "closed" instead of "open" and in contrast to the single-valued case, upper semi-continuity is different from lower semi-continuity. Both concepts of semi-continuity should not be confused with the corresponding notions of continuity of real number functions.

[15] Implicitly, we have used that lower semi-continuous operations are closed under composition, see Theorem 8.2, where composition is defined as in Definition 8.1.

8. Recursive Closure Schemes

In this section we discuss some set-theoretical closure schemes of computable multi-valued operations. These closure schemes are generalizations of the classical closure schemes *substitution*, *primitive recursion* and *μ–recursion* [44]. In the following we will assume that U, V, X, Y, Z are arbitrary sets. All closure schemes which will be introduced are defined for arbitrary sets, with exception of those places where the set $\mathbb{N}$ of natural numbers is mentioned explicitly.

Definition 8.1 (Recursive closure schemes) The following closure schemes are called *recursive closure schemes*:

(1) **Projection**: If $f :\subseteq X \rightrightarrows Y \times Z$ is an operation, then the *projection* $f_1 :\subseteq X \rightrightarrows Y$ is defined by

$$f_1(x) := \{y : (\exists z)\ (y, z) \in f(x)\}$$

for all $x \in \mathrm{dom}(f_1) := \mathrm{dom}(f)$. The projection $f_2 :\subseteq X \rightrightarrows Z$ on the second component is defined correspondingly.

(2) **Juxtaposition**: If $f :\subseteq X \rightrightarrows Y$ and $g :\subseteq X \rightrightarrows Z$ are operations, then the *juxtaposition* $(f, g) :\subseteq X \rightrightarrows Y \times Z$ is defined by

$$(f, g)(x) := f(x) \times g(x) = \{(y, z) : y \in f(x) \text{ and } z \in g(x)\}$$

for all $x \in \mathrm{dom}(f, g) := \mathrm{dom}(f) \cap \mathrm{dom}(g)$.

(3) **Product**: If $f :\subseteq X \rightrightarrows Y$ and $g :\subseteq U \rightrightarrows V$ are operations, then the *product* $f \times g :\subseteq X \times U \rightrightarrows Y \times V$ is defined by

$$(f \times g)(x, u) := f(x) \times g(u) = \{(y, v) : y \in f(x) \text{ and } v \in g(u)\}$$

for all $(x, u) \in \mathrm{dom}(f \times g) := \mathrm{dom}(f) \times \mathrm{dom}(g)$.

(4) **Composition**: If $f :\subseteq X \rightrightarrows Y$ and $g :\subseteq Y \rightrightarrows Z$ are operations, then the *composition* $g \circ f :\subseteq X \rightrightarrows Z$ is defined by

$$(g \circ f)(x) := g(f(x)) := \{z : (\exists y \in f(x))\ z \in g(y)\}$$

for all $x \in \mathrm{dom}(g \circ f) := \{x : f(x) \subseteq \mathrm{dom}(g)\}$.

(5) **Iteration**: If $f :\subseteq X \rightrightarrows X$ is an operation, then the *iteration* $f^* :\subseteq X \times \mathbb{N} \rightrightarrows X$ is defined by

$$\begin{cases} f^*(x, 0) & := \quad \{x\}, \\ f^*(x, n+1) & := \quad f \circ f^*(x, n) \end{cases}$$

and abbreviated by $f^n(x) := f^*(x, n)$ for all $x \in X$ and $n \in \mathbb{N}$.

(6) **Inversion**: If $f :\subseteq X \times \mathbb{N} \rightrightarrows Y \times \mathbb{N}$ is an operation, then the (*twisted*) *inversion* $f^{\leftrightarrow} :\subseteq X \times \mathbb{N} \rightrightarrows Y \times \mathbb{N}$ is defined by

$$f^{\leftrightarrow}(x,n) := \{(y,k) : (y,n) \in f(x,k)\}$$

for all $(x,n) \in \mathrm{dom}(f^{\leftrightarrow}) := \{(x,n) : (\forall k)\ (x,k) \in \mathrm{dom}(f)$ and $(\exists k)\ n \in f_2(x,k)\}$.

(7) **Evaluation**: If $f :\subseteq X \rightrightarrows Y^{\mathbb{N}}$ is an operation, then the *evaluation* $f_* :\subseteq X \times \mathbb{N} \rightrightarrows Y$ is defined by

$$f_*(x,n) := \{y : (\exists (y_k)_{k\in\mathbb{N}} \in f(x))\ y_n = y\}$$

for all $(x,n) \in \mathrm{dom}(f_*) := \mathrm{dom}(f) \times \mathbb{N}$.

(8) **Transposition**: If $f :\subseteq X \times \mathbb{N} \rightrightarrows Y$ is an operation, then the *transposition* $[f] :\subseteq X \rightrightarrows Y^{\mathbb{N}}$ is defined by

$$[f](x) := \{(y_n)_{n\in\mathbb{N}} : (\forall n)\ y_n \in f(x,n)\}$$

for all $x \in \mathrm{dom}([f]) := \{x : (\forall n)\ (x,n) \in \mathrm{dom}(f)\}$.

(9) **Exponentiation**: If $f :\subseteq X \rightrightarrows Y$ is an operation, then the *exponentiation* $f^{\mathbb{N}} :\subseteq X^{\mathbb{N}} \rightrightarrows Y^{\mathbb{N}}$ is defined by

$$f^{\mathbb{N}}((x_n)_{n\in\mathbb{N}}) := \{(y_n)_{n\in\mathbb{N}} : (\forall n)\ y_n \in f(x_n)\}$$

for all $(x_n)_{n\in\mathbb{N}} \in \mathrm{dom}(f^{\mathbb{N}}) := \{(x_n)_{n\in\mathbb{N}} : (\forall n)\ x_n \in \mathrm{dom}(f)\}$.

(10) **Sequentialization**: If $f :\subseteq X \rightrightarrows \mathbb{N}$ is an operation, then the *sequentialization* $f^{\triangle} :\subseteq X \rightrightarrows \mathbb{N}^{\mathbb{N}}$ is defined by

$$f^{\triangle}(x) := \{(y_n)_{n\in\mathbb{N}} : f(x) = \{y_n : n \in \mathbb{N}\}\}$$

for all $x \in \mathrm{dom}(f^{\triangle}) := \mathrm{dom}(f)$.

The closure schemes projection, juxtaposition and product can be used to handle product spaces and analogously, evaluation, transposition and exponentiation can be used to handle sequence spaces (the latter three schemes are infinite versions of the former three). In presence of projection, juxtaposition and product, the classical schemes of substitution, primitive recursion and minimization can be realized by composition, iteration and inversion. Finally, sequentialization is a scheme which can be used to eliminate indeterminism in certain cases.[16]

[16]The definitions of most of the closure schemes are straightforward, although the details need some care. For instance, a second natural choice for the domain $\mathrm{dom}(g \circ f)$ of the composition of g and f would be the set $\{x : f(x) \cap \mathrm{dom}(g) \neq \emptyset\}$. However, this definition would not satisfy Theorem 8.2.

Theorem 8.2 *On topological spaces, all recursive closure schemes preserve lower semi-continuity of multi-valued operations.*

A proof can be found in [9]. Using the notations of Definition 8.1, we tacitly assume for the previous result that U, V, X, Y, Z are topological spaces and $\mathbb{N}$ is endowed with the discrete topology. Moreover, product and sequence spaces are to be endowed with the corresponding product topologies. Actually, the closure schemes do not only preserve continuity but computability as well. For the following result we assume that (U, δ_U), (V, δ_V), (X, δ_X), (Y, δ_Y), (Z, δ_Z) are represented spaces and that $\mathbb{N}$ is represented by $\delta_{\mathbb{N}}$. Product spaces and sequence spaces are represented by the corresponding product and sequence representations.

Theorem 8.3 *On represented spaces, all recursive closure schemes preserve (strong) computability of multi-valued operations.*

Proof. (1) Let $f :\subseteq X \rightrightarrows Y \times Z$ be (strongly) $(\delta_X, [\delta_Y, \delta_Z])$–computable via a computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$. Define $G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by $G\langle p, q\rangle := \pi_1 F\langle p, q\rangle$ for all $p, q \in \mathbb{N}^{\mathbb{N}}$, where $\pi : \mathbb{N}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is the usual pairing function and $\pi_1 := \mathrm{pr}_1 \pi^{-1}$. Then G is computable and $f_1 :\subseteq X \rightrightarrows Y$ is (strongly) (δ_X, δ_Y)–computable via G.
(2) Let $f :\subseteq X \rightrightarrows Y$, $g :\subseteq X \rightrightarrows Z$ be (strongly) (δ_X, δ_Y)–, (δ_X, δ_Z)–computable via computable functions $F, G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, respectively. Define $H :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by $H\langle p, \langle q, r\rangle\rangle := \langle F\langle p, q\rangle, G\langle p, r\rangle\rangle$ for all $p, q, r \in \mathbb{N}^{\mathbb{N}}$. Then H is computable and $(f, g) :\subseteq X \rightrightarrows Y \times Z$ is (strongly) $(\delta_X, [\delta_Y, \delta_Z])$–computable via H.
(3) Let $f :\subseteq X \rightrightarrows Y$, $g :\subseteq U \rightrightarrows V$ be (strongly) (δ_X, δ_Y)–, (δ_U, δ_V)–computable via computable functions $F, G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, respectively. Define $H :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by $H\langle\langle p, p'\rangle, \langle q, r\rangle\rangle := \langle F\langle p, q\rangle, G\langle p', r\rangle\rangle$ for all $p, p', q, r \in \mathbb{N}^{\mathbb{N}}$. Then H is computable and $f \times g :\subseteq X \times U \rightrightarrows Y \times V$ is (strongly) $([\delta_X, \delta_U], [\delta_Y, \delta_V])$–computable via H.
(4) Let $f :\subseteq X \rightrightarrows Y$, $g :\subseteq Y \rightrightarrows Z$ be (strongly) (δ_X, δ_Y)–, (δ_Y, δ_Z)–computable via computable functions $F, G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, respectively. Define $H :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by $H\langle p, \langle q, r\rangle\rangle := G\langle F\langle p, q\rangle, r\rangle$ for all $p, q, r \in \mathbb{N}^{\mathbb{N}}$. Then H is computable and $g \circ f :\subseteq X \rightrightarrows Z$ is (strongly) (δ_X, δ_Z)–computable via H.
(5) Let $f :\subseteq X \rightrightarrows X$ be (strongly) (δ_X, δ_X)–computable via a computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$. Define $G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by

$$\begin{cases} G\langle\langle p, p'\rangle, \langle q_0, q_1, ...\rangle\rangle := p & \text{if } p'(0) = 0 \\ G\langle\langle p, p'\rangle, \langle q_0, q_1, ...\rangle\rangle := F\langle G\langle\langle p, \hat{n}\rangle, \langle q_0, q_1, ...\rangle\rangle, q_n\rangle & \text{if } p'(0) = n + 1 \end{cases}$$

for all $p, p', q_i \in \mathbb{N}^{\mathbb{N}}, i \in \mathbb{N}$. Then G is computable and $f^* :\subseteq X \times \mathbb{N} \rightrightarrows X$ is (strongly) $([\delta_X, \delta_{\mathbb{N}}], \delta_X)$–computable via G.

(6) Let $f :\subseteq X \times \mathbb{N} \rightrightarrows Y \times \mathbb{N}$ be $([\delta_X, \delta_{\mathbb{N}}], [\delta_Y, \delta_{\mathbb{N}}])$–computable via a computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$. Thus, there is a computable, total, and monotone function $\varphi : \mathbb{N}^* \to \mathbb{N}^*$ which approximates F. Define a predicate $P \subseteq \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N}$ by

$$P := \{(p, n, k, i) : \pi_1'\nu^*(i) \sqsubseteq \langle p, \hat{k}\rangle \text{ and } n \sqsubseteq \pi_2'\varphi\nu^*(i)\}.$$

Here, $\pi_1', \pi_2' : \mathbb{N}^* \to \mathbb{N}^*$ denote computable projections such that $w \sqsubseteq \langle p, q\rangle$ is equivalent to $\pi_1'(w) \sqsubseteq p$ and $\pi_2'(w) \sqsubseteq q$. By π_i we will denote the projections of the inverses of the tuple functions $\mathbb{N}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, as well as of $\mathbb{N}\times\mathbb{N} \to \mathbb{N}$. Since P is decidable it follows that $h :\subseteq \mathbb{N}^{\mathbb{N}}\times\mathbb{N}\times\mathbb{N} \to \mathbb{N}$, defined by

$$\begin{cases} h(p, n, 0) & := & \mu\langle k, i\rangle[(p, n, k, i) \in P] \\ h(p, n, m+1) & := & \mu\langle k, i\rangle[\langle k, i\rangle > h(p, n, m) \text{ and } (p, n, k, i) \in P] \end{cases}$$

for all p, n, m, is computable. Define a function $G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by $G\langle\langle p, p'\rangle, mq\rangle := \langle \pi_1 F\langle\langle p, \hat{k}\rangle, wq\rangle, \hat{k}\rangle$ for all $p, p', q \in \mathbb{N}^{\mathbb{N}}$, $m \in \mathbb{N}$, where $\langle k, i\rangle := h(p, p'(0), m)$ and $w := \pi_2'\nu^*(i)$. Then G is computable and one can show that $f^{\leftrightarrow}$ is computable via G. In case of strong computability one has to restrict G to those inputs $\langle\langle p, p'\rangle, \langle q, q'\rangle\rangle$ such that $\langle\langle p, q\rangle, q'\rangle \in \mathrm{dom}(F)$.

(7) Let $f :\subseteq X \rightrightarrows Y^{\mathbb{N}}$ be (strongly) $(\delta_X, \delta_Y^\infty)$–computable via a computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$. Define a function $G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by $G\langle\langle p, p'\rangle, q\rangle := \eta(F\langle p, q\rangle, p'(0))$ for all $p, p', q \in \mathbb{N}^{\mathbb{N}}$, where the evaluation function $\eta : \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{N}^{\mathbb{N}}$, defined by $\eta(\langle p_0, p_1, ...\rangle, n) := p_n$, is computable. Then G is computable and $f_* :\subseteq X \times \mathbb{N} \rightrightarrows Y$ is (strongly) $([\delta_X, \delta_{\mathbb{N}}], \delta_Y)$–computable via G.

(8) Let $f :\subseteq X \times \mathbb{N} \rightrightarrows Y$ be (strongly) $([\delta_X, \delta_{\mathbb{N}}], \delta_Y)$–computable via a computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$. Define $G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by $G\langle p, \langle q_0, q_1, ...\rangle\rangle := \langle F\langle\langle p, \hat{0}\rangle, q_0\rangle, F\langle\langle p, \hat{1}\rangle, q_1\rangle, ...\rangle$ for all $p, q_i \in \mathbb{N}^{\mathbb{N}}$, $i \in \mathbb{N}$. Then G is computable and $[f] :\subseteq X \rightrightarrows Y^{\mathbb{N}}$ is (strongly) $(\delta_X, \delta_Y^\infty)$–computable via G.

(9) Let $f :\subseteq X \rightrightarrows Y$ be (strongly) (δ_X, δ_Y)–computable via a computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$. Define a function $G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by $G\langle\langle p_0, p_1, ...\rangle, \langle q_0, q_1, ...\rangle\rangle := \langle F\langle p_0, q_0\rangle, F\langle p_1, q_1\rangle, ...\rangle$ for all $p_i, q_i \in \mathbb{N}^{\mathbb{N}}$, $i \in \mathbb{N}$. Then G is computable and $f^{\mathbb{N}} :\subseteq X^{\mathbb{N}} \rightrightarrows Y^{\mathbb{N}}$ is (strongly) $(\delta_X^\infty, \delta_Y^\infty)$–computable via G.

(10) Let $f :\subseteq X \rightrightarrows \mathbb{N}$ be $(\delta_X, \delta_{\mathbb{N}})$–computable via a computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$. Define $G' :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by $G'(p)(n) := \delta_{\mathbb{N}} F\langle p, r_n\rangle = F\langle p, r_n\rangle(0)$ for all $p \in \mathbb{N}^{\mathbb{N}}$, $n \in \mathbb{N}$, where $r_n := \nu^*(n)000... = \nu^*(n)\hat{0}$. Since $\{r_n : n \in \mathbb{N}\}$ is dense in $\mathbb{N}^{\mathbb{N}}$ and F is continuous, it follows

$$f\delta_X(p) = \{\delta_{\mathbb{N}} F\langle p, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\} = \{G'(p)(n) : n \in \mathbb{N}\}$$

for all $p \in \mathrm{dom}(f\delta_X)$. Now define $G :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ by

$$G\langle p,q\rangle\langle n,k\rangle := \begin{cases} H\langle G'(p),q\rangle(n) & \text{if } \langle p,q\rangle \in \mathrm{dom}(F) \\ \uparrow & \text{else} \end{cases}$$

for all $p,q \in \mathbb{N}^\mathbb{N}$ and $n,k \in \mathbb{N}$, where H is the function given by the Oracle Lemma 7.2. Then G is computable. By definition of H it follows

$$\begin{aligned} f^\triangle\delta_X(p) &= \{r \in \mathbb{N}^\mathbb{N} : f\delta_X(p) = \mathrm{range}(r)\} \\ &= \{H\langle G'(p),q\rangle : q \in \mathbb{N}^\mathbb{N}\} \\ &= \{\delta_\mathbb{N}^\infty G\langle p,q\rangle : q \in \mathbb{N}^\mathbb{N}\} \end{aligned}$$

for all $p \in \mathrm{dom}(f^\triangle\delta_X)$, i.e. $f^\triangle :\subseteq X \rightrightarrows \mathbb{N}^\mathbb{N}$ is $(\delta_X, \delta_\mathbb{N}^\infty)$–computable via G. Correspondingly, the case of strong computability can be treated. □

9. Recursive Operations over Structures

In this section we will use the recursive closure schemes to introduce the notion of a recursive operation over a structure. We will consider many-sorted structures $(X_1,...,X_n;f_1,...,f_k)$ with a finite number of pairwise disjoint sets $X_1,...,X_n$ and a finite number of operations $f_1,...,f_k$. We start with the definition of a *prestructure* (prestructures have the property that the operations are not necessarily related to the universes).

Definition 9.1 (Prestructures) $S = (X_1,...,X_n;f_1,...,f_k)$ is called a *many-sorted prestructure* with *universe* $X_1 \times ... \times X_n$, if $X_1,...,X_n$ are pairwise disjoint sets and $f_1,...,f_k$ are arbitrary operations, called the *initial operations* of S.

Usually, we will say for short *prestructure* instead of many-sorted prestructure. In the next step we define the notion of a set over a prestructure $S = (X_1,...,X_n;f_1,...,f_k)$. Roughly speaking, a set over a prestructure S is a set that can be constructed as product or sequence set of the sets $X_1,...,X_n$. This definition takes into account that we want to handle product spaces as well as sequential spaces.

Definition 9.2 (Sets and operations over prestructures) Let $S = (X_1,...,X_n;f_1,...,f_k)$ be a prestructure. Then the *class of sets over* S is the smallest class of sets such that:

(1) $X_1,...,X_n$ are sets over S,

(2) $X \times Y$ and $X^\mathbb{N}$ are sets over S, if X,Y are sets over S.

An operation $f :\subseteq X \rightrightarrows Y$ is called an *operation over* S, if X,Y are sets over S.

Now we will single out those prestructures S whose initial operations are operations over S and we will call them *structures*.

Definition 9.3 (Structures) A prestructure S is called a *structure*, if all initial operations of S are operations over S.

The reason why the universe of a structure S is defined as product is a technical one: it ensures that the universe itself is a set over S. If $S = (X_1, ..., X_n; f_1, ..., f_k)$ and $T = (Y_1, ..., Y_m; g_1, ..., g_l)$ are structures such that $Y_1, ..., Y_m \in \{X_1, ..., X_n\}$ and $g_1, ..., g_l \in \{f_1, ..., f_k\}$, then T is called a *substructure* of S and we write $T \sqsubseteq S$. Now we proceed to define recursive operations over structures.

Definition 9.4 (Recursive operations over structures) The class of *recursive operations over a structure* S is the smallest class of operations which contains all initial operations of S and which is closed under the recursive closure schemes.

From the definition of the closure schemes it is obvious that each recursive operation over a structure is an operation over that structure.

A constant $x \in X$ is called a *recursive constant over a structure* S, if the operation $\{()\} \to X, () \mapsto x$ is recursive over S (we assume $X^0 = \{()\}$ for each set X). If there is a constant $c \in X$ among the initial operations of a structure, then we will consider it as zero-ary constant function $X^0 \to X, () \mapsto c$ with value c. For completeness, we define a representation $\delta : \mathbb{N}^{\mathbb{N}} \to \{()\}$ by $\delta(p) := ()$ for all $p \in \mathbb{N}^{\mathbb{N}}$, but usually we will not use this representation explicitly.

A very important structure is the *structure of natural numbers* which is given by

$$\mathbf{N} := (\mathbb{N}; 0, n, n+1).$$

By "0" we denote the *zero-ary constant function* $\mathbb{N}^0 \to \mathbb{N}, () \mapsto 0$ with value 0, by "n" the *identity* $\mathrm{id}_{\mathbb{N}} : \mathbb{N} \to \mathbb{N}, n \mapsto n$, and by "$n+1$" the *successor function* $S : \mathbb{N} \to \mathbb{N}, n \mapsto n+1$. Here and in the following we will often use bold letters, as "$\mathbf{N}$", to distinguish a structure from its universe. It is routine to prove the following result which shows that the class of recursive operations over $\mathbf{N}$ is quite rich.

Proposition 9.5 *Each computable function* $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ *is recursive over* $\mathbf{N}$.

One can even prove this result without using the closure scheme of sequentialization, see [9]. In the following we will often need structures which include the natural numbers and some essential operations. These structures will be called *natural*. By $\mathrm{id}_X : X \to X, x \mapsto x$ we denote the *identity* of X.

Definition 9.6 (Natural structures) A structure S with universe X is called *natural*, if $\mathbf{N}$ is a substructure of S, id_X is recursive over S and there is a recursive constant $c \in X$ over S.

It is easy to see that for natural structures $S = (X_1, ..., X_n; f_1, ..., f_k)$ the identity of each set over S is recursive. Especially, the identities $\mathrm{id}_{X_1}, ..., \mathrm{id}_{X_n}$ are recursive, and each set $X_1, ..., X_n$ contains at least one recursive constant.

Sometimes it is also suitable to consider structures where a special constant operation is available. For each set $A \subseteq X$ we define the *omnipotent operation* $\Omega_A : \{()\} \rightrightarrows X, () \mapsto A$ of A, i.e. Ω_A is the constant operation which yields as result $\Omega_A() = A$.

Definition 9.7 (Complete structure) A structure S with universe X is called a *complete structure*, if Ω_X is recursive over S.

Since $\Omega_{\mathbb{N}} = z^{\leftrightarrow} \circ 0$, where $z = (0 \times \mathrm{id}_{\mathbb{N}})_1$ denotes the constant zero function $z : \mathbb{N} \to \mathbb{N}, n \mapsto 0$ and $\Omega_{\mathbb{N}^{\mathbb{N}}} = [(\Omega_{\mathbb{N}} \times \mathrm{id}_{\mathbb{N}})_1]$, we obtain that $\Omega_{\mathbb{N}}$ and $\Omega_{\mathbb{N}^{\mathbb{N}}}$ are recursive over each natural structure. Especially, the structure $\mathbf{N}$ is complete.

Our next goal is to prove that recursive operations are also computable. Therefore, we will first define representations of structures and computability on structures.

Definition 9.8 (Representations of structures) Consider a structure $S = (X_1, ..., X_n; f_1, ..., f_k)$ and let $\delta_1, ..., \delta_n$ be representations of $X_1, ..., X_n$, respectively. Then $\delta := [\delta_1, ..., \delta_n]$ is called a *representation* of S and (S, δ) is called a *represented structure*.

Using this definition we can extend the notion of computability to structures.

Definition 9.9 (Computability on structures) Consider a structure $S = (X_1, ..., X_n; f_1, ..., f_k)$ with representation $\delta = [\delta_1, ..., \delta_n]$. Now, let $f :\subseteq Y \rightrightarrows Z$ be an operation over S and let δ_Y, δ_Z be representations of Y, Z, respectively, which are finitely generated from $\delta_1, ..., \delta_n$, correspondingly as Y, Z are finitely generated from $X_1, ..., X_n$. Then f is called (*strongly*) *computable* with respect to δ, if it is (strongly) (δ_Y, δ_Z)–computable.

It should be clear how the representations δ_Y, δ_Z have to be constructed correspondingly to Y, Z. If, for instance, $Y = (X_1 \times X_2)^{\mathbb{N}}$, then the corresponding representation is $\delta_Y = [\delta_1, \delta_2]^{\infty}$. In the next step we define the notion of an effective structure.

Definition 9.10 (Effective structures) A structure S is called a (*strongly*) *effective structure*, if there is a representation δ of S such that all initial operations of S are (strongly) computable with respect to δ. In this situation S is also called (*strongly*) *effective via* δ.

It is easy to see that the structure $\mathbf{N}$ is strongly effective via $\delta_{\mathbb{N}}$. In the following we will use effective structures where the effectivity on natural numbers is fixed. We will say that $\delta = [\delta_1, ..., \delta_n]$ is a *natural representation* of a natural structure $S = (X_1, ..., X_n; f_1, ..., f_k)$, if $X_i = \mathbb{N}$ implies $\delta_i \equiv \delta_{\mathbb{N}}$. By structural induction we obtain the following corollary of Theorem 8.3.

Corollary 9.11 (Recursive operations over effective structures) *If S is a natural structure which is (strongly) effective via a natural representation δ, then each operation which is recursive over S is also (strongly) computable with respect to δ.*

10. A Stability Theorem for Perfect Structures

In this section we introduce the class of recursive structures and we will prove that over recursive structures each strongly computable operation is also recursive. This leads us to the definition of perfect structures which are both: effective and recursive. Perfect structures have several nice properties, especially they are categorical in the sense that they characterize their own computability theory. In order to define recursive structures, we will use the notion of a recursive retraction.

Definition 10.1 (Recursive retraction) An operation $f :\subseteq X \rightrightarrows Y$ over a structure S is called a *recursive retraction* over S, if it is recursive and if it admits a recursive right inverse operation $f^- : Y \rightrightarrows X$ over S, i.e. $f \circ f^- = \mathrm{id}_Y$.

If f is a recursive retraction, then the restriction of f to range(f^-) is a surjective function. A structure will be called strongly recursive, if it admits a representation which is a recursive retraction.

Definition 10.2 (Recursive structure) Let S be a structure. Then

(1) S is called a *recursive structure*, if there is a representation δ of S, which admits a recursive extension, as well as a recursive right inverse operation over S,

(2) S is called a *strongly recursive structure*, if there is a representation δ of S which is a recursive retraction over S.

In these cases S is also called *(strongly) recursive via* δ.

Intuitively, a strongly recursive structure is a structure with a representation which can be "synthesized" as well as "analyzed" within the structure.By the following lemma $\delta = [\delta_1, ..., \delta_n]$ is a recursive retraction, if $\delta_1, ..., \delta_n$ are.

Lemma 10.3 *Let S be a natural structure and let $\delta_i :\subseteq \mathbb{N}^{\mathbb{N}} \to X_i$, $i = 1, ..., n$, $\delta :\subseteq \mathbb{N}^{\mathbb{N}} \to X$ be representations. If $\delta_1, ..., \delta_n, \delta$ are recursive retractions over S, then so are $[\delta_1, ..., \delta_n]$ and δ^∞.*

Vice versa, it is easy to see that $\delta_1, ..., \delta_n$ are recursive retractions over S, if $\delta = [\delta_1, ..., \delta_n]$ is a recursive retraction over S. Now we prove that the structure of natural numbers is recursive.

Proposition 10.4 *The structure* $\mathbf{N}$ *is strongly recursive via* $\delta_{\mathbb{N}}$.

Proof. The representation $\delta_{\mathbb{N}} = (\mathrm{id}_{\mathbb{N}^{\mathbb{N}}})_* \circ (\mathrm{id}_{\mathbb{N}^{\mathbb{N}}} \times 0)$ is recursive over $\mathbf{N}$. Define $f : \mathbb{N} \times \mathbb{N} \rightrightarrows \mathbb{N}$ by $f(k, 0) := k$ and $f(k, n+1) := \Omega_{\mathbb{N}}()$. Then f is recursive over $\mathbf{N}$ and so is $\delta_{\mathbb{N}}^{-1} = [f]$. $\square$

Since the structure of natural numbers is recursive it makes sense to apply the notion of recursiveness to natural structures. Now we can prove the announced converse version of Corollary 9.11.

Theorem 10.5 (Comp. operations over recursive structures) *If S is a natural structure which is strongly recursive via a representation δ, then each operation over S which is strongly computable with respect to δ is also recursive over S.*

Proof. Let $S = (X_1, ..., X_n; f_1, ..., f_k)$, let $\delta = [\delta_1, ..., \delta_n]$, and let $f :\subseteq X \rightrightarrows Y$ be an operation over S. Then X, Y are finitely generated by product and sequence constructions from the sets $X_1, ..., X_n$. Let δ_X, δ_Y be the corresponding representations, constructed from $\delta_1, ..., \delta_n$. By structural induction one can deduce from Lemma 10.3 that δ_X, as well as δ_Y, are recursive retractions. Now let $\delta_X^- : X \rightrightarrows \mathbb{N}^{\mathbb{N}}$ be a right inverse of δ_X, such that δ_X^- is recursive over S. Let $f :\subseteq X \rightrightarrows Y$ be strongly (δ_X, δ_Y)–computable via a computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$. Then one obtains $f = \delta_Y \circ F \circ \langle \delta_X^- \times \Omega_{\mathbb{N}^{\mathbb{N}}} \rangle$ and it follows that f is recursive over S: since F is computable, it is recursive over $\mathbf{N}$ by Proposition 9.5 and $\langle \delta_X^- \times \Omega_{\mathbb{N}^{\mathbb{N}}} \rangle$ is recursive over S, since the tupling function $\langle . \rangle : \mathbb{N}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, δ_X^- and $\Omega_{\mathbb{N}^{\mathbb{N}}}$ are recursive over S. Altogether, f is recursive over S. $\square$

From the proof we can also conclude the following property of computable operations over recursive structures.

Corollary 10.6 (Comp. operations over recursive structures) *If S is a natural structure which is recursive via a representation δ, then each operation over S which is computable with respect to δ also admits a recursive extension over S.*

We have seen that over effective structures recursive operations are computable and over recursive structures computable operations are recursive. Thus, it makes sense to consider structures which are both, recursive and effective. We will see that such structures have very nice properties. The first result states that for such structures there is a unique effectivity which is characterized by the structure itself.

Theorem 10.7 (Stability theorem) *If S is a natural structure which is recursive via a representation δ and which is effective via a natural representation δ', then $\delta \equiv \delta'$.*

Proof. Let S be a natural structure which is recursive via a representation δ and which is effective via a natural representation δ'. Without loss of generality, let $X = \mathbb{N} \times X_2 \times ... \times X_n$ be the universe of S and let $\delta' = [\delta'_1, ..., \delta'_n]$. Since δ' is a natural representation $\delta_1'^{\infty} \equiv \delta_{\mathbb{N}}^{\infty} \equiv \mathrm{id}_{\mathbb{N}^{\mathbb{N}}}$. Let δ_+ be an extension of δ and let δ^- be a right inverse operation of δ which are recursive over S. Then $\delta_- := (\delta^-)^{-1}$ is a representation of S too. We prove $\delta \equiv \delta_+ \equiv \delta' \equiv \delta_-$.

"$\delta \leq \delta_+$" Since δ_+ is an extension of δ, we obtain $\delta \leq \delta_+$.

"$\delta_+ \leq \delta'$" By Corollary 9.11 it follows that δ_+ is computable with respect to δ', i.e. δ_+ is $(\delta_1'^{\infty}, \delta')$–computable. Thus, δ_+ is $(\mathrm{id}_{\mathbb{N}^{\mathbb{N}}}, \delta')$–computable. But this means $\delta_+ \leq \delta'$.

"$\delta' \leq \delta_-$" Again we can deduce by Corollary 9.11 that δ^- is $(\delta', \delta_1'^{\infty})$–computable and thus $(\delta', \mathrm{id}_{\mathbb{N}^{\mathbb{N}}})$–computable, i.e. there is a computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that $\delta^-\delta'(p) = \{F\langle p, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\}$ for all $p \in \mathrm{dom}(\delta^-\delta') = \mathrm{dom}(\delta')$. Then $G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, defined by $G(p) := F\langle p, p\rangle$ for all $p \in \mathbb{N}^{\mathbb{N}}$ is computable and $\delta'(p) = \delta_- G(p)$ for all $p \in \mathrm{dom}(\delta')$, i.e. $\delta' \leq \delta_-$.

"$\delta_- \leq \delta$" Since δ^- is a right inverse operation of δ, it follows that δ_- is a restriction of δ and thus $\delta_- \leq \delta$. □

We can deduce that, if a structure S is recursive, then all natural representations δ such that S is effective via δ are equivalent. Thus, if S is recursive and effective, then it is *effectively categorical*, a notion that has been used in a similar setting by Hertling [29]. But even more, if S is effective via a natural representation, then all representations δ which are recursive retractions over S are also equivalent. Thus, if S

is recursive and effective, then it has a further property which could be called *recursively categorical*.

It should be mentioned that there is no hope for a corresponding result without the restriction to natural representations. As long as we do not demand any evaluation property for the output, we can effectivize a structure just by using terms and their evaluations. The situation changes if we do fix effectivity for at least one "output sort". For instance, in the classical setting this can be done by fixing effectivity for the boolean sort, i.e. the "output sort" of predicates.[17] In our setting, we have fixed effectivity on the natural numbers. The Stability Theorem gives rise to the following definition.

Definition 10.8 (Perfect structures) A natural structure S is called (*strongly*) *perfect*, if it is (strongly) recursive and (strongly) effective via a natural representation. In this situation each natural representation δ such that S is (strongly) recursive via δ is called a (*strong*) *standard representation* of S.

It is easy to see that $\mathbf{N}$ is a strongly perfect structure with strong standard representation $\delta_{\mathbb{N}}$. The Stability Theorem states that perfect structures uniquely characterize their effectivity. Especially, all standard representations of a perfect structure and all natural representations which make this structure effective belong to the same equivalence class. Therefore, we can define computability over perfect structures without mentioning any special representation.

Definition 10.9 (Comp. operations over perfect structures) An operation f over a perfect structure S is called (*strongly*) *computable over* S, if it is (strongly) computable with respect to a standard representation δ of S.

Now we can formulate a corollary of Theorem 10.5 and Corollary 9.11 which characterizes recursive operations over strongly perfect structures. The corresponding statement for perfect structures follows from Corollary 10.6 and Corollary 9.11

Theorem 10.10 (Operations over perfect structures) *An operation over a strongly perfect structure is strongly computable, if and only if it is recursive. An operation over a perfect structure is computable, if and only if it admits a recursive extension.*

[17] This is the case in Mal'cev's Stability Theorem for finitely generated algebras [39, 54], which can be considered as a special version of the Stability Theorem 10.7, see [10].

Finally, we obtain an Extension Theorem for operations over strongly perfect structures as a combination of both results.

Corollary 10.11 (Extension Theorem) *Each computable operation over a strongly perfect structure admits a strongly computable extension.*

Another nice property of strongly perfect structures is the property of *conservative extension.* Consider a natural structure S with an *extension* S', i.e. a structure S' such that $S \sqsubseteq S'$. In general, the additional initial operations of S' could increase the class of recursive operations even over the universe of S. The following theorem states that this cannot happen over perfect structures.

Theorem 10.12 (Conservation Theorem) *Let $S \sqsubseteq S'$ be strongly perfect structures and let f be an operation over S. Then f is recursive over S, if and only if f is recursive over S'.*

Proof. Without loss of generality, we assume $S = (X_1, ..., X_n; f_1, ..., f_k)$ and $S' = (X_1, ..., X_{n+1}; f_1, ..., f_m)$ with $m \geq k$. The general case can be deduced by induction. Let $\delta = [\delta_1, ..., \delta_n]$ and $\delta' = [\delta'_1, ..., \delta'_{n+1}]$ be standard representations of S, S', respectively. Then S' is effective via δ' and hence S is effective via $\delta'' := [\delta'_1, ..., \delta'_n]$. On the other hand, S is effective via δ and since S is perfect we can conclude $\delta \equiv \delta''$ by the Stability Theorem 10.7. If f is recursive over S, then f obviously is recursive over S' too. If f is an operation over S which is recursive over S', then f is strongly computable over S' by Theorem 10.10, since S' is strongly perfect. Thus, f is strongly computable with respect to δ' and thus with respect to δ''. Consequently, f is strongly computable with respect to δ too and thus it is recursive over S by Theorem 10.10, since S is strongly perfect. $\square$

If we revisit the proof, then we obtain the following version of the Conservation Theorem for perfect structures.

Corollary 10.13 *Let $S \sqsubseteq S'$ be perfect structures and let f be an operation over S. Then f admits a recursive extension over S, if and only if it admits a recursive extension over S'.*

We close this section with a short meta-analysis of our results. We have already mentioned that for the proof of Proposition 9.5 it suffices to use all recursive closure schemes besides sequentialization. Proposition 9.5 has been used for the proof of Theorem 10.5. But neither for this theorem nor for the other results of this section we have used the sequentialization operator. Indeed, all results on perfect structures remain true, if we exclude sequentialization from our recursive closure schemes. However, the class of perfect structures could be smaller in this case and this is the reason why we have included sequentialization.

11. Topological and Metric Structures

In the previous section we have seen that perfect structures have some nice properties but we have not proved that there are any interesting perfect structures. In this section we will show that T_0–spaces with countable bases admit perfect topological structures and we will define a perfect standard structure for separable metric spaces. We start with the definition of the notion of a topological structure. It is natural to assume that the universes of such structures are topological spaces and that the initial operations are continuous.

Definition 11.1 (Topological structure) A *topological structure* is a structure $S = (X_1, ..., X_n; f_1, ..., f_k)$ where $X_1, ..., X_n$ are endowed with topologies and all initial operations $f_1, ..., f_k$ are lower semi-continuous with respect to the corresponding product topologies. Moreover, S is called *natural topological structure*, if S is a natural structure and the natural numbers come equipped with the discrete topology.

If Y is a set over a structure S, then it inherits a product topology which is generated from the topologies of $X_1, ..., X_n$, correspondingly as Y is generated from the sets $X_1, ..., X_n$. In this situation we will say that Y is a *topological space over* S. In the following we will sometimes use short names for topological structures with additional properties. For instance, a "metric structure" will be a topological structure where the universe is endowed with a metric. As we have seen in Theorem 8.2, all recursive closure schemes preserve lower semi-continuity. Therefore, we obtain immediately the following corollary.

Corollary 11.2 (Continuity of recursive operations) *All recursive operations over natural topological structures are lower semi-continuous.*

In the following, by $S \oplus T := (X_1, ..., X_n, Y_1, ..., Y_m; f_1, ..., f_k, g_1, ..., g_l)$ we will denote the *union* of a prestructure $S = (X_1, ..., X_n; f_1, ..., f_k)$ with a prestructure $T = (Y_1, ..., Y_m; g_1, ..., g_l)$. If X is a second countable T_0–space with standard representation δ, then δ is open and continuous, hence δ, as well as $\delta^{-1} : X \rightrightarrows \mathbb{N}^{\mathbb{N}}$, are lower semi-continuous. If there is some computable point $p \in \mathrm{dom}(\delta)$, then $\mathbf{X} = \mathbf{N} \oplus (X; \delta, \delta^{-1})$ is a natural topological structure. Obviously, $\delta' := [\delta_{\mathbb{N}}, \delta]$ is a representation of $\mathbf{X}$ and $\mathbf{X}$ is recursive via δ'. Moreover, $\mathbf{X}$ is also effective via δ' since δ is $(\mathrm{id}_{\mathbb{N}^{\mathbb{N}}}, \delta)$–computable and δ^{-1} is $(\delta, \mathrm{id}_{\mathbb{N}^{\mathbb{N}}})$–computable by the Oracle Lemma 7.2. Altogether we obtain the following result.

Theorem 11.3 (Perfect topological structures) *If X is a second countable T_0–space with standard representation δ such that there is some computable point in* $\mathrm{dom}(\delta)$*, then $\mathbf{X} = \mathbf{N} \oplus (X; \delta, \delta^{-1})$ is a perfect topological structure with standard representation $[\delta_{\mathbb{N}}, \delta]$.*

Here, second countability is not only a sufficient but even a necessary condition. If $\mathbf{X}$ is a natural topological structure which is recursive via some representation δ, then there exists some right inverse δ^- of δ which is recursive over $\mathbf{X}$. Then $\{(\delta^-)^{-1}(w\mathbb{N}^{\mathbb{N}}) : w \in \mathbb{N}^*\}$ is a countable base of X. Thus, we have proved the following result.

Proposition 11.4 *If S is a recursive topological structure with universe X, then the topology of X has a countable base.*

By Theorem 11.3 we know that any second countable T_0–space admits a perfect topological structure, but the structure given in Theorem 11.3 is quite artificial because it contains a representation and its inverse as initial operations. In the following we will see that separable metric spaces admit far more natural structures. As a special case we first investigate a structure for the real numbers. As usual, 0 and 1 denote the corresponding real constants, $+, -, \cdot, /$ the usual arithmetic operations on the real numbers with their natural domains. By $<$ we denote the *semi-characteristic operation* of the usual order relation:

$$c_< : \mathbb{R} \times \mathbb{R} \rightrightarrows \mathbb{N}, (x, y) \mapsto \begin{cases} \{0, 1\} & \text{if} \quad x < y \\ \{1\} & \text{else} \end{cases}$$

This definition will be generalized and discussed below. As before, $\mathrm{Lim} :\subseteq \mathbb{R}^{\mathbb{N}} \to \mathbb{R}$ denotes the *limit operator* (restricted to rapidly converging sequences) with respect to the Euclidean distance.

Theorem 11.5 (The structure of the real numbers) *The structure*

$$\mathbf{R} := \mathbf{N} \oplus (\mathbb{R}; 0, 1, x + y, -x, x \cdot y, 1/x, \mathrm{Lim}, x < y)$$

is a strongly perfect complete topological structure and the Cauchy representation yields a standard representation of this structure.

Proof. Obviously, all arithmetic operations are continuous with respect to the Euclidean topology. The limit operation is continuous (with respect to the product topology on $\mathbb{R}^{\mathbb{N}}$), since it is restricted to rapidly converging sequences and $c_<$ is lower semi-continuous since $<$ is open. Altogether, $\mathbf{R}$ is a natural topological structure. It is easy to see that the notation of rational numbers $\nu_{\mathbb{Q}}$ is recursive over $\mathbf{R}$ and thus the Cauchy representation $\delta_{\mathbb{R}} = \mathrm{Lim} \circ \nu_{\mathbb{Q}}^{\mathbb{N}}$ is recursive over $\mathbf{R}$ as well. By the Heron algorithm one can prove that the square root function $x \mapsto \sqrt{x}$ is recursive over $\mathbf{R}$ and it follows that the absolute value function and hence the Euclidean metric $d_{\mathbb{R}}(x,y) = |x-y|$ are recursive over $\mathbf{R}$. Using these facts, one can show that there is a recursive function $f : \mathbb{R} \times \mathbb{N} \rightrightarrows \mathbb{N}$ over $\mathbf{R}$ such that $\mathrm{graph}(f) = \{(x,k,n) \in \mathbb{R} \times \mathbb{N} \times \mathbb{N} : d_{\mathbb{R}}(\nu_{\mathbb{Q}}(n), x) < 2^{-k-1}\}$. Hence, $\delta^- := [f] : \mathbb{R} \rightrightarrows \mathbb{N}^{\mathbb{N}}$ is recursive over $\mathbf{R}$ and it is a right inverse of $\delta_{\mathbb{R}}$. Altogether, it follows that $\mathbf{R}$ is strongly recursive via $\delta_{\mathbb{R}}$. It remains to prove that $\mathbf{R}$ is strongly effective via $\delta_{\mathbb{R}}$ as well. It is routine to prove that the arithmetic operations are strongly computable with respect to the Cauchy representation. By Proposition 5.3 it follows that the limit operation is computable as well and using completeness one can show that Lim is even strongly computable. Finally, the semi-characteristic operation $c_< : \mathbb{R} \times \mathbb{R} \rightrightarrows \mathbb{N}$ is strongly computable via the function $F : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, defined by

$$F\langle\langle p, p'\rangle, q\rangle := \begin{cases} \hat{0} & \text{if} \quad \nu_{\mathbb{Q}} pq(0) < \nu_{\mathbb{Q}} p'q(0) - 2^{-q(0)+1} \text{ and } q(1) = 0 \\ \hat{1} & \text{else} \end{cases}$$

for all $p, p', q \in \mathbb{N}^{\mathbb{N}}$. □

More precisely, $[\delta_{\mathbb{N}}, \delta_{\mathbb{R}}]$ is a standard representation of $\mathbf{R}$. As a direct consequence of this theorem and Theorem 10.10, we can deduce that the recursive functions over $\mathbf{R}$ are exactly the strongly computable ones. As a special case we obtain that the total recursive functions $f : \mathbb{R}^n \to \mathbb{R}$ over $\mathbf{R}$ are exactly the classically computable functions (according to Grzegorczyk's and Lacombe's definitions [24, 36]).

Corollary 11.6 *The recursive functions $f : \mathbb{R}^n \to \mathbb{R}$ over $\mathbf{R}$ are exactly the classically computable real functions.*

As long as we are only interested in total functions, we can obtain the same class of recursive functions even without using the order at all. More precisely, we will consider the order-free structure of the real numbers $\mathbf{R}^\star := \mathbf{N} \oplus (\mathbb{R}; 0, 1, x+y, -x, x \cdot y, 1/x, \mathrm{Lim})$. Shepherdson observed that the effective version of the Weierstraß Approximation Theorem implies that computable functions $f : [0,1] \to \mathbb{R}$ have straight-line programs [51]. Generalizing this idea one can prove the following theorem on order-free recursion. The proof uses special properties of $\mathbb{R}$, such as local compactness (see [8, 9] for details).

Theorem 11.7 (Order-free recursion) *A function $f : \mathbb{R}^n \to \mathbb{R}$ is recursive over $\mathbf{R}^\star$, if and only if it is recursive over $\mathbf{R}$.*

However, one cannot expect that the structure $\mathbf{R}^\star$ is perfect since otherwise the semi-characteristic operation $c_<$ of the order relation would have to be recursive over $\mathbf{R}^\star$ by the Conservation Theorem, i.e. Corollary 10.13. As we will see in the following, the perfectness result on $\mathbf{R}$ can be generalized to recursive metric spaces. As usual, we denote by $\mathrm{Lim} :\subseteq X^{\mathbb{N}} \to X$ the limit operation (restricted to rapidly converging sequences).

Theorem 11.8 (Metric structures) *If (X, d, α) is a (complete) recursive metric space, then*

$$\mathbf{X} := \mathbf{R} \oplus (X; \alpha, \mathrm{id}, d, \mathrm{Lim})$$

is a (strongly) perfect (complete) topological structure. The Cauchy representation yields a standard representation of this structure.

More precisely, if δ_X is the Cauchy representation of the space X, then $[\delta_{\mathbb{N}}, \delta_{\mathbb{R}}, \delta_X]$ is a standard representation of $\mathbf{X}$. The proof is based on Proposition 5.3 and a generalization of the proof of Theorem 11.5. The structure $\mathbf{X}$ is already a bit more natural than the structure given in Theorem 11.3. However, in many concrete cases it is even possible to "decompose" α with the help of the algebraic structure of the given space. We have seen this already in case of the real number structure $\mathbf{R}$. We give two further examples. Again, we will use a λ-like notation for operations as injection $X \to \mathcal{K}(X), x \mapsto \{x\}$, union of compact subsets $\mathcal{K}(X) \times \mathcal{K}(X) \mapsto \mathcal{K}(X), (A, B) \mapsto A \cup B$, and the scalar product of the function space $\mathbb{R} \times \mathcal{C}[0, 1] \to \mathcal{C}[0, 1], (y, f) \mapsto y \cdot f$ etc.

Example 11.9 (Perfect metric structures)

(1) If (X, d) is a (complete) recursive metric space, then the structure

$$\mathcal{K}(\mathbf{X}) := \mathbf{X} \oplus (\mathcal{K}(X); \{x\}, A, A \cup B, d_{\mathcal{K}}, \mathrm{Lim})$$

is a (strongly) perfect topological structure with respect to the topology generated by the Hausdorff metric $d_{\mathcal{K}}$.

(2) The structure

$$\mathcal{C}[\mathbf{0}, \mathbf{1}] := \mathbf{R} \oplus (\mathcal{C}[0, 1]; 1, f, y \cdot f, f + g, f \cdot g, ||f||, \mathrm{Lim})$$

is a strongly perfect topological structure with respect to the topology of uniform convergence generated by the supremum norm $||\ ||$.

Here "1" denotes the constant function in $\mathcal{C}[0, 1]$ with value 1 and Lim denotes the limit operation on the corresponding space. Proofs and a detailed discussion of further examples can be found in [9]. By the Stability Theorem 10.7 the results of this section especially show that the structures **R** and **X** are effectively categorical. This has already been proved in a slightly different setting by Hertling [29]. By results of Hemmerling [28] it follows that effectively categorical structures for recursive metric spaces necessarily include infinitary operations (as Lim). Since second countable T_0–spaces are quasi-metrizable, it is promising to transfer the considerations of this section to computable quasi-metric spaces [12].

12. Recursive Sets over Structures

In Section 6 we have discussed effective subsets of recursive metric spaces. In this section we want to characterize these notions by some canonical abstract definitions for sets over structures which are presented in the following definition. If $A \subseteq X$, then we sometimes write $A^c := X \setminus A$ for the *complement* of A.

Definition 12.1 (Recursive sets over structures) Let S be a natural structure, let X be a set over S and let $A \subseteq X$.

(1) A is *initially semi-recursive* over S, if there exists a total recursive operation $f : X \rightrightarrows \mathbb{N}$ over S such that $A = f^{-1}\{0\}$.

(2) A is *finally semi-recursive* over S, if there exists a partial recursive operation $f :\subseteq \mathbb{N} \rightrightarrows X$ over S such that $A = f\{0\}$.

(3) A is *recursive* over S, if A is finally semi-recursive and $X \setminus A$ is initially semi-recursive over S.

(4) A is *decidable* over S, if there exists a total recursive function $f : X \to \mathbb{N}$ over S such that $A = f^{-1}\{0\}$.

In the following we will say for short *semi-recursive* instead of initially semi-recursive. Obviously, the empty set $\emptyset$, considered as a subset of a set X over a natural structure S, is always recursive and decidable. Furthermore, the set X, considered as a subset of itself, is always initially semi-recursive and decidable over S and X is also finally semi-recursive and hence recursive over S, if and only if Ω_X is recursive over S. In general, it is easy to see that over a natural structure S a non-empty subset A is finally semi-recursive, if and only if Ω_A is recursive over S. Especially, all sets over complete structures are finally semi-recursive and a point $x \in X$ is recursive over S, if and only if $\{x\}$ is finally semi-recursive over X. Over natural structures S a subset A is decidable, if and only if its *characteristic function* cf_A is recursive over S. We introduce the *semi-characteristic operation* to characterize semi-recursive sets A in a corresponding way.

Definition 12.2 (Semi-characteristic operation) For each set X and each subset $A \subseteq X$ we define the *semi-characteristic operation* $c_A : X \rightrightarrows \mathbb{N}$ of A by

$$c_A(x) := \begin{cases} \{0,1\} & \text{if } x \in A \\ \{1\} & \text{else} \end{cases}$$

With this notation a subset A over a natural structure S is semi-recursive, if and only if c_A is recursive over S. The intuition behind a semi-characteristic operation c_A is that it operates as follows: for any input $x \in X$ the operation can answer "no" (indicated by the value 1), but exactly for all inputs $x \in A$ there has to be a computation such that the operation yields the answer "yes" (indicated by the value 0).

It is easy to see that semi-recursive and decidable sets are closed under intersection and semi-recursive, finally semi-recursive, recursive and decidable sets are closed under union. Decidable subsets are obviously closed under complement as well and, as in the classical case we obtain that a subset $A \subseteq X$ of a set X over a natural structure S is decidable over S, if and only if A and A^c are semi-recursive over S. Moreover, semi-recursive sets are closed under preimage of recursive operations and finally semi-recursive sets under the image (provided they are contained in the domain). Over complete natural structures, semi-recursiveness implies final semi-recursiveness.

Proposition 12.3 *Over complete natural structures each semi-recursive set is also finally semi-recursive and each decidable subset is recursive.*

For any operation $f :\subseteq X \rightrightarrows Y \times \mathbb{N}$ we call $f_0 :\subseteq X \rightrightarrows Y$, defined by $f_0 := ((f \circ \mathrm{pr}_1)^{\leftrightarrow})_1 \circ (\mathrm{id}_X \times 0)$, the *section* of f. This name is justified since one obtains $f_0(x) = \{y \in Y : (y,0) \in f(x)\}$. The proof of the proposition follows immediately from $\Omega_A = ((\mathrm{id}_X, c_A) \circ \Omega_X)_0$. The next important property is an "uniformization property" which states that semi-recursive sets are graphs of recursive operations.

Theorem 12.4 (Uniformization Theorem) *Let X, Y be sets over a natural structure S and let $A \subseteq X \times Y$. If Y is finally semi-recursive and A semi-recursive over S, then there is a recursive operation $f :\subseteq X \rightrightarrows Y$ such that* $\mathrm{graph}(f) = A$.

Proof. Define $g : X \times Y \rightrightarrows Y \times \mathbb{N}$ by $g(x,y) := (y, c_A(x,y))$ for all $(x,y) \in X \times Y$. Then g is recursive over S and so is $f :\subseteq X \rightrightarrows Y$, defined by $f := (g \circ (\mathrm{id}_X \times \Omega_Y))_0$. Finally, $\mathrm{graph}(f) = A$. $\square$

Implicitly, the Uniformization Theorem has already been applied in the proof of Theorem 11.5. For a special class of structures we can formulate further implications among the introduced notions of effectivity. If X is a set over a structure S, then the *equality on* X is the predicate $=_X$, that is the set $\{(x,x) : x = x\} \subseteq X \times X$. Over natural structures with semi-recursive equality each finally semi-recursive subset is also semi-recursive. The proof immediately follows from the formula $c_A = c_{=_X} \circ (\Omega_A \times \mathrm{id}_X)$ for all $A \subseteq X$. Together with Proposition 12.3 we get the following corollary.

Corollary 12.5 *Over complete natural structures with semi-recursive equality a subset is semi-recursive, if and only if it is finally semi-recursive and a subset is decidable, if and only if it is recursive.*

This situation is similar to classical recursion theory. Analogously, we can prove the following result.

Theorem 12.6 (Graph Theorem with equality) *Let S be a complete natural structure, $f :\subseteq X \rightrightarrows Y$ an operation over S, and let the equality $=_Y$ be semi-recursive over S. Then f is recursive and* $\mathrm{dom}(f)$ *is semi-recursive over S, if and only if* $\mathrm{graph}(f)$ *is semi-recursive.*

Proof. Let f be recursive and $\mathrm{dom}(f)$ semi-recursive. Since we obtain $\Omega_{\mathrm{graph}(f)} = (\mathrm{id}_X, f) \circ \Omega_{\mathrm{dom}(f)}$ it follows by Corollary 12.5 that $\mathrm{graph}(f)$ is semi-recursive. Now let $\mathrm{graph}(f)$ be semi-recursive. Then f is recursive over S by the Uniformization Theorem 12.4. Moreover, $c_{\mathrm{dom}(f)} = c_{\mathrm{graph}(f)} \circ (\mathrm{id}_X \times \Omega_Y)$, i.e. $\mathrm{dom}(f)$ is semi-recursive. □

However, perfect structures with semi-recursive equality have necessarily a countable universe (which follows from an easy cardinality argument, see [9]). Thus, structures with semi-recursive equality are not very relevant for spaces which occur in analysis. The next best property which one could expect is semi-recursiveness of inequality. Corresponding to equality, *inequality on* X is the predicate $\neq_X$, that is the set $\{(x,y) : x \neq y\} \subseteq X \times X$. Over complete natural structures semi-recursiveness of the inequality already implies recursiveness of equality. This follows since $=_X$ is equal to $\mathrm{range}((\mathrm{id}_X, \mathrm{id}_X) \circ \Omega_X)$. Next, we will prove that over structures with inequality a point is recursive, if and only if the corresponding single-valued set is recursive.

Proposition 12.7 *Let S be a natural structure, X a set over S such that inequality $\neq_X$ is semi-recursive over S, and let $x \in X$. Then x is recursive over S, if and only if $\{x\}$ is recursive over S.*

Proof. Obviously, x is recursive over S, if and only if $\{x\}$ is finally semi-recursive over S. Thus we only have to prove that the complement $\{x\}^c$ is semi-recursive, if $\{x\}$ is finally semi-recursive over S. This follows immediately from $c_{\neq_X}(\Omega_{\{x\}}, y) = c_{\{x\}^c}(y)$ for all $y \in X$. $\square$

Finally, we mention that for perfect structures with inequality we can at least obtain one direction of the Graph Theorem.

Theorem 12.8 (Graph Theorem with inequality) *Let S be a perfect structure, $f :\subseteq X \to Y$ a function over S and let $\neq_Y$ be semi-recursive over S. If $\mathrm{dom}(f)$ and f are recursive over S, then $\mathrm{graph}(f)$ is recursive over S.*

The proof can be found in [9]. Next we want to consider recursive, semi-recursive and decidable sets over topological structures. Let X be a topological space and $A \subseteq X$. Since $\{0\}$ is an open subset of $\mathbb{N}$ and $c_A^{-1}\{0\} = A$, the characteristic operation $c_A : X \rightrightarrows \mathbb{N}$ is lower semi-continuous, if and only if A is open. Thus we can immediately deduce some facts about (semi-)recursive sets over topological structures.

Proposition 12.9 *Over natural topological structures semi-recursive subsets are open, recursive subsets are closed and decidable subsets are open and closed.*

We can directly conclude that there are no non-trivial decidable subsets over connected topological structures.

Corollary 12.10 *Let S be a topological structure, X a connected topological space over S. If $A \subseteq X$ is a decidable subset over S then $A = \emptyset$ or $A = X$.*

As a further corollary we can derive necessary topological properties of topological structures with equality and inequality.

Corollary 12.11 *Let S be a natural topological structure and let X be a topological space over S.*

(1) If $\neq_X$ is semi-recursive over S, then X is a Hausdorff space.

(2) If $=_X$ is semi-recursive over S, then X is a discrete space.

The proof follows directly, since in the first case, $=_X$ is a closed subset of $X \times X$ and in the second case an open subset. If, in the second case, the structure S, additionally, is effective via a natural representation, then we know that X is even a countable discrete space. Our metric standard structures actually have semi-recursive inequalities which is easy to see.

Proposition 12.12 *If X is a recursive metric space, then $\neq_X$ is semi-recursive over $\mathbf{X}$ and if X, additionally, is complete, then $\neq_X$ is even recursive over $\mathbf{X}$.*

Here, the second part of the statement follows directly from Theorem 11.3 and Proposition 12.3. We close this section with a characterization of effective subsets over metric structures.

Theorem 12.13 (Recursive sets over metric structures) *Let X be a recursive metric space and $A \subseteq X$. Then A is semi-recursive over $\mathbf{X}$, if and only if A^c is co-r.e. closed. If, additionally, X is complete and A closed, then A is finally semi-recursive over $\mathbf{X}$, if and only if A is r.e. closed and A is recursive over $\mathbf{X}$, if and only if A is recursive.*

This result shows that the notions which we have introduced in Definition 12.1 are compatible with those of Section 6. The details are left to the reader and can be found in [9].

Acknowledgement

The author would like to thank Klaus Weihrauch, Matthias Schröder and the anonymous referee for several helpful comments, suggestions and corrections.

References

[1] O. Aberth, *Computable Analysis.* McGraw-Hill, New York 1980.

[2] K. Ambos-Spies, K. Weihrauch, and X. Zheng, Weakly computable real numbers, *J. Complexity* **16** (2000) 676–690.

[3] S. Banach and S. Mazur, Sur les fonctions calculables, *Ann. Soc. Pol. de Math.* **16** (1937) 223.

[4] A. Bauer, *The Realizability Approach to Computable Analysis and Topology*, PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh 2000.

[5] A. Bauer, A relationship between equilogical spaces and type two effectivity, in: S. Brooks and M. Mislove (eds.), *Seventeenth Conference on the Mathematical Foundations of Programming Semantics*, vol. 45 of *Elect. Not. Theoret. Comp. Sci.*, Elsevier, Amsterdam 2001.

[6] E. Bishop and D. S. Bridges, *Constructive Analysis.* Springer, Berlin 1985.

[7] V. Brattka, Recursive characterization of computable real-valued functions and relations, *Theoret. Comp. Sci.* **162** (1996) 45–77.

[8] V. Brattka, Order-free recursion on the real numbers, *Math. Log. Quart.* **43** (1997) 216–234.

[9] V. Brattka, Recursive and computable operations over topological structures, Informatik Berichte 255, FernUniversität Hagen, 1999.

[10] V. Brattka, A stability theorem for recursive analysis, in: C. S. Calude and M. J. Dinneen (eds.), *Combinatorics, Computation & Logic*, Springer, Singapore 1999, 144–158.

[11] V. Brattka, Computable versions of Baire's category theorem, in: J. Sgall, A. Pultr, and P. Kolman (eds.), *Mathematical Foundations of Computer Science 2001*, vol. 2136 of *Lect. Not. Comp. Sci.*, Springer, Berlin 2001, 224–235.

[12] V. Brattka, Recursive quasi-metric spaces, *Theoret. Comp. Sci.* (accepted for publication).

[13] V. Brattka and P. Hertling, Topological properties of real number representations, *Theoret. Comp. Sci.* (to appear).

[14] V. Brattka and G. Presser, Computability on subsets of metric spaces, *Theoret. Comp. Sci.* (accepted for publication).

[15] V. Brattka and K. Weihrauch, Computability on subsets of Euclidean space I: Closed and compact subsets, *Theoret. Comp. Sci.* **219** (1999) 65–93.

[16] G. Ceĭtin, Algorithmic operators in constructive metric spaces, *Tr. Mat. Inst. Steklov* **67** (1962) 295–361.

[17] T. Deil, Darstellungen und Berechenbarkeit reeller Zahlen, Informatik Berichte 51, FernUniversität Hagen, 1984.

[18] S. Eilenberg and C. C. Elgot, *Recursiveness.* Academic Press, New York 1970.

[19] J. L. Eršov, Theorie der Numerierungen I–III, *Zeitschr. Math. Logik Grundl. Math.* **19,21,23** (1973,1975,1977) 289–388,473–584,289–371.

[20] J. L. Eršov, Theory of numberings, in: E. R. Griffor (ed.), *Handbook of Computability Theory*, Elsevier, Amsterdam 1999, 473–503.

[21] J. E. Fenstad, *General Recursion Theory.* Springer, Berlin 1980.

[22] H. Friedman, Algorithmic procedures, generalized Turing algorithms, and elementary recursion theory, in: R. Gandy and C. Yates (eds.), *Logic Colloquium'69*, vol. 61 of *Stud. Logic Found. Math.*, North-Holland, Amsterdam 1971, 361–389.

[23] X. Ge and A. Nerode, On extreme points of convex compact Turing located sets, in: A. Nerode and Y. V. Matiyasevich (eds.), *Logical Foundations of Computer Science*, vol. 813 of *Lect. Not. Comp. Sci.*, Springer, Berlin 1994, 114–128.

[24] A. Grzegorczyk, Computable functionals, *Fund. Math.* **42** (1955) 168–202.

[25] A. Grzegorczyk, On the definitions of computable real continuous functions, *Fund. Math.* **44** (1957) 61–71.

[26] J. Hauck, Berechenbare reelle Funktionen, *Zeitschr. Math. Logik Grundl. Math.* **19** (1973) 121–140.

[27] J. Hauck, Konstruktive Darstellungen reeller Zahlen und Folgen, *Zeitschr. Math. Logik Grundl. Math.* **24** (1978) 365–374.

[28] A. Hemmerling, Standard representations of effective metric spaces, in: J. Blanck, V. Brattka, and P. Hertling (eds.), *Computability and Complexity in Analysis*, vol. 2064 of *Lect. Not. Comp. Sci.*, Springer, Berlin 2001, 48–68.

[29] P. Hertling, A real number structure that is effectively categorical, *Math. Log. Quart.* **45** (1999) 147–182.

[30] K.-I. Ko, *Complexity Theory of Real Functions.* Birkhäuser, Boston 1991.

[31] M. Konečný, *Many-Valued Real Functions Computable by Finite Transducers using IFS-Representations*, PhD thesis, School of Computer Science, University of Birmingham, 2000.

[32] C. Kreitz and K. Weihrauch, A unified approach to constructive and recursive analysis, in: M. Richter, E. Börger, W. Oberschelp, B. Schinzel, and W. Thomas (eds.), *Computation and Proof Theory*, vol. 1104 of *Lect. Not. Math.*, Springer, Berlin 1984, 259–278.

[33] C. Kreitz and K. Weihrauch, Theory of representations, *Theoret. Comp. Sci.* **38** (1985) 35–53.

[34] C. Kreitz and K. Weihrauch, Compactness in constructive analysis revisited, *Ann. Pure Appl. Logic* **36** (1987) 29–38.

[35] B. A. Kušner, *Lectures on Constructive Mathematical Analysis.* American Mathematical Society, Providence 1984.

[36] D. Lacombe, Extension de la notion de fonction récursive aux fonctions d'une ou plusieurs variables réelles I-III, *Comptes Rendus* **240,241** (1955) 2478–2480,13–14,151–153.

[37] D. Lacombe, Les ensembles récursivement ouverts ou fermés, et leurs applications à l'Analyse récursive, *Comptes Rendus* **246** (1958) 28–31.

[38] D. Lacombe, Quelques procédés de définition en topologie récursive, in: A. Heyting (ed.), *Constructivity in mathematics*, North-Holland, Amsterdam 1959, 129–158.

[39] A. I. Mal'cev, Constructive algebras. I, in: *The Metamathematics of Algebraic Systems*, North-Holland, Amsterdam 1971, 148–214.

[40] S. Mazur, *Computable Analysis.* Razprawy Matematyczne, Warsaw 1963.

[41] Y. N. Moschovakis, Recursive metric spaces, *Fund. Math.* **55** (1964) 215–238.

[42] Y. N. Moschovakis, Abstract first order computability I, II, *Transactions of the American Mathematical Society* **138** (1969) 427–464,465–504.

[43] A. Mostowski, On computable sequences, *Fund. Math.* **44** (1957) 37–51.

[44] P. Odifreddi, *Classical Recursion Theory.* North-Holland, Amsterdam 1989.

[45] M. B. Pour-El and J. I. Richards, *Computability in Analysis and Physics.* Springer, Berlin 1989.

[46] H. Rice, Recursive real numbers, *Proc. Amer. Math. Soc.* **5** (1954) 784–791.

[47] R. Robinson, Review of "Peter, R., Rekursive Funktionen", *J. Symbolic Logic* **16** (1951) 280–282.

[48] M. Schröder, Admissible representations of limit spaces, in: J. Blanck, V. Brattka, and P. Hertling (eds.), *Computability and Complexity in Analysis*, vol. 2064 of *Lect. Not. Comp. Sci.*, Springer, Berlin 2001, 273–295.

[49] M. Schröder, Extended admissibility, *Theoret. Comp. Sci.* (to appear).

[50] J. Shepherdson, Computation over abstract structures: serial and parallel procedures and Friedman's effective definitional schemes, in: H. Rose and J. Shepherdson (eds.), *Logic Colloquium'73*, vol. 80 of *Stud. Logic Found. Math.*, North-Holland, Amsterdam 1975, 445–513.

[51] J. Shepherdson, On the definition of computable functions of a real variable, *Zeitschr. Math. Logik Grundl. Math.* **22** (1976) 391–402.

[52] D. G. Skordev, *Computability in Combinatory Spaces.* Kluwer Academic Publishers, Dordrecht 1992.

[53] D. Spreen, On effective topological spaces, *J. Symbolic Logic* **63** (1998) 185–221.

[54] V. Stoltenberg-Hansen and J. V. Tucker, Effective algebras, in: S. Abramsky, D. Gabbay, and T. Maibaum (eds.), *Handbook of Logic in Computer Science, Volume 4*, Clarendon Press, Oxford 1995, 357–527.

[55] A. Troelstra, Comparing the theory of representations and constructive mathematics, in: E. Börger, G. Jäger, H. Kleine Büning, and M. Richter (eds.), *Computer Science Logic*, vol. 626 of *Lect. Not. Comp. Sci.*, Springer, Berlin 1992, 382–395.

[56] J. Tucker and J. Zucker, *Program Correctness over Abstract Data Types, with Error-State Semantics.* North-Holland, Amsterdam 1988.

[57] J. Tucker and J. Zucker, Computation by 'While' programs on topological partial algebras, *Theoret. Comp. Sci.* **219** (1999) 379–420.

[58] J. Tucker and J. Zucker, Abstract versus concrete computation on metric partial algebras, Technical Report CAS-01-01-JZ, McMaster University, Hamilton 2001.

[59] A. M. Turing, On computable numbers, with an application to the "Entscheidungsproblem", *Proc. London Math. Soc.* **42** (1936) 230–265.

[60] A. M. Turing, On computable numbers, with an application to the "Entscheidungsproblem". A correction., *Proc. London Math. Soc.* **43** (1937) 544–546.

[61] K. Weihrauch, Type 2 recursion theory, *Theoret. Comp. Sci.* **38** (1985) 17–33.

[62] K. Weihrauch, *Computability.* Springer, Berlin 1987.

[63] K. Weihrauch, Computability on computable metric spaces, *Theoret. Comp. Sci.* **113** (1993) 191–210.

[64] K. Weihrauch, *Computable Analysis.* Springer, Berlin 2000.

[65] K. Weihrauch and C. Kreitz, Representations of the real numbers and of the open subsets of the set of real numbers, *Ann. Pure Appl. Logic* **35** (1987) 247–260.

[66] M. Yasugi, T. Mori, and Y. Tsujii, Effective properties of sets and functions in metric spaces with computability structure, *Theoret. Comp. Sci.* **219** (1999) 467–486.

[67] X. Zheng and K. Weihrauch, The arithmetical hierarchy of real numbers, *Math. Log. Quart.* **47** (2001) 51–65.

[68] Q. Zhou, Computable real-valued functions on recursive open and closed subsets of Euclidean space, *Math. Log. Quart.* **42** (1996) 379–409.