Recursive and Computable Operations over Topological Structures

Dissertation

zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften

dem Fachbereich Informatik
der FernUniversität - Gesamthochschule in Hagen

vorgelegt von

Diplom-Informatiker Vasco Brattka

im Dezember 1998

angenommen aufgrund der Berichte von

Prof. Dr. K. Weihrauch, Hagen
Prof. Dr. J.V. Tucker, Swansea, UK

am 29. Juni 1999

Contents

1 Introduction **3**

2 Recursion Operators **9**

2.1 The classically recursive functions 9
2.2 Recursion operators for analysis 11
2.3 The definition of the operators 12
2.3.1 Projection 15
2.3.2 Juxtaposition 16
2.3.3 Product 17
2.3.4 Composition 17
2.3.5 Iteration 18
2.3.6 Inversion 18
2.3.7 Evaluation 19
2.3.8 Transposition 19
2.3.9 Exponentiation 19
2.3.10 Sequentialization 19
2.4 Elementary properties of the operators 20
2.5 Further operators 23
2.5.1 Substitution 23
2.5.2 Primitive recursion 23
2.5.3 Simultaneous recursion 25
2.5.4 Definition by cases 26
2.5.5 Course-of-value recursion 27
2.5.6 Minimization 28
2.5.7 Section 31
2.5.8 Union 32

3 Recursive and Computable Operations over Structures **33**

3.1 Recursive and computable operations over nat. structures 33
3.1.1 Recursive operations over structures 33
3.1.2 The structure of natural numbers 35
3.1.3 Theory of effectivity 38
3.1.4 Computable operations over structures 41

3.1.5 Computable operations are recursive 42
3.1.6 Recursive operations are computable 45
3.2 Perfect structures and sets over structures 55
3.2.1 Perfect structures 55
3.2.2 Reducibility of structures 61
3.2.3 Domains over structures 64
3.2.4 Recursive sets over structures 65
3.2.5 Complete structures 71
3.2.6 Structures with equality 72
3.2.7 Structures with inequality 74
3.2.8 Countable structures 76

4 Recursive Operations over Topological Structures 79
4.1 Recursive operations over topological structures 79
4.2 Recursive sets over topological structures 83
4.3 Perfect topological structures 85
4.4 Metric structures 87
4.4.1 The structure of real numbers 88
4.4.2 Recursive metric spaces 91
4.4.3 Complete recursive metric spaces 95
4.4.4 Constructions on recursive metric spaces 97
4.4.5 Recursive functions over metric structures 101
4.4.6 Recursive sets over metric structures 104
4.4.7 The structure of compact subsets 107
4.4.8 Lifting of functions 109
4.4.9 Recursively locally compact metric spaces 112
4.4.10 Nice metric spaces 113
4.4.11 Recursive Banach spaces 115
4.4.12 The space of continuous functions 117
4.4.13 Evaluation of continuous functions 120
4.4.14 The structure of continuous functions 123
4.4.15 Recursive partition of unity 124
4.4.16 The structure of closed subsets 126
4.5 Order-free recursion over the real numbers 129

5 Conclusion 135

Appendix: Computable Operations and Relations 137

Tables of Perfect Structures 139

Symbols 145

Index 160

Chapter 1

Introduction

> La topologie récursive apparaît ainsi comme la base (...)
> d'une Mathématique constructive.[1]
> (Daniel Lacombe, 1959)

One of Turing's motivations to introduce his famous machine model was to give a precise definition to the notion of a computable real number. In his sense, a real number is *computable*, if there exists a Turing machine which can produce the decimal expansion of the number [Tur36]. As Rice proved later, the computable real numbers form a countable and algebraically closed field which, besides the rational numbers, includes all those well-known concrete numbers like π and e [Ric54]. Although this gives some evidence of the fact that the decimal representation yields a reasonable class of computable numbers, it has already been realized by Turing that the decimal representation is not appropriate to introduce computability of real-valued functions [Tur37]. Intuitively, a real-valued function is computable by a Turing machine, if each finite portion of the output can be computed from a finite portion of the input. Even such a simple function like multiplication by 3, is not computable with respect to the decimal representation in this sense. Fortunately, other representations of the real numbers, like the representation by rational intervals or by rapidly converging Cauchy sequences, have been shown to yield a suitable class of computable real-valued functions. These computable functions, independently introduced by Grzegorczyk and Lacombe, include the rational polynomials, as well as most concrete functions, like the exponential function and the trigonometric functions [Grz57, Lac55]. It should come as no surprise that each computable real-valued function is continuous, simply, because each finite portion of the output especially depends on a finite portion of the input.

While the class of computable real numbers is invariant under several representations, as noted by Robinson [Rob51], Mostowski observed that even

[1] Recursive topology appears as the base (...) of a constructive mathematics [Lac59].

the class of computable sequences is not invariant in general [Mos57]. Evidently, the choice of the representation is a crucial matter if the notion of computability does so sensitively rely on it. Nevertheless, some time passed until Hauck presented a general representation based theory of computability [Hau71, Hau73, Hau78, Hau80, Hau81].[2] 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 differences [KW84, Wei85, KW87, Wei87, Wei].[3] Their theory goes far beyond real numbers and offers a comprehensive theoretical framework for computable analysis and the investigation of effectivity in certain topological spaces.

In the present investigation we want to focus attention on an abstract characterization of computability in analysis. While the theory of representations offers a very concrete point of view, where Turing machines are used to transform names of real numbers (and other continuous objects), one can ask whether it is possible to take an abstract point of view where one builds up the computable operations from some initial operations by certain closure schemes. Both points of view, the abstract and the concrete one, have an analogy in computer science: while the concrete Turing machine theory could be identified with an assembler language which operates on bits, the abstract theory should be identified with an high-level programming language which directly operates on the objects of interest via suitable data structures (cf. Figure 1.1). In classical recursion theory an abstract characterization of computability is

abstract	**concrete**
high-level language	assembler language
closure schemes	Turing machines
recursive operations	computable operations

Figure 1.1: Abstract versus concrete models

given by the μ–recursive functions which have been developed by Herbrand, Gödel, Kleene and others [Kle36]. In the past, several different generalizations of recursive closure schemes in abstract settings have been suggested and investigated by Moschovakis, Friedman, Shepherdson, Fenstad, Tucker and Zucker, and others [Mos69, Mos71, Fri71, She75, She85, Fen80, TZ88]. Some of these approaches are oriented on the needs of higher order recursion theory or on

[2] Cf. also Klaua [Kla56, Kla61].

[3] Deil has investigated representations of the real numbers in this framework [Dei84].

countable structures, some do require a computable pairing mechanism or computable (equality) tests which are not available in our setting and hence these approaches seem to be less appropriate for computable analysis. Following the line of the classical approach we will present generalized versions of the recursive closure schemes which fit together with the needs of computable analysis. The main idea of our approach is to choose a structure for each data-type, which includes some initial operations, and to define *recursive* operations as those ones which can be finitely generated from the initial ones by applying the closure schemes.

One result of the approach, presented here, is that it seems to be unnatural to prove the equivalence of (concrete) computability and (abstract) recursiveness on the level of functions. It appears to be more appropriate to choose the larger class of *operations* to establish the equivalence and to conclude the coincidence of functions as a special case. Here, an operation could equiva-

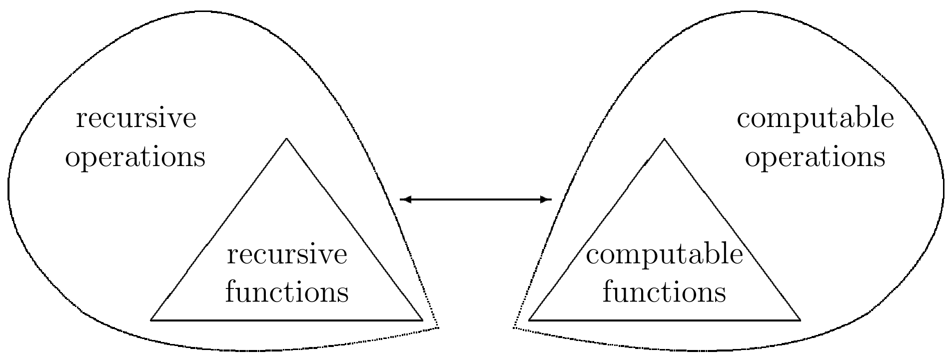

Figure 1.2: Effective functions and operations

lently be considered as a multi-valued function, which possibly offers several outputs for a given input. From a certain point of view these operations can be considered as *indeterministic* operations which always yield some valid result (in contrast to non-deterministic operations which only possibly yield a valid result). However, the technical difficulty to handle operations would be a high price for an abstract model of effectivity if such objects did not appear in practice. But indeed, the necessity to introduce operations is caused by the fact that they inevitably appear in computations. There are several important and natural problems in computable analysis, like the determination of zeros of polynomials, which do offer only many-valued solutions. As a simple example, we note that there are no non-trivial (deterministic) tests on the real numbers, simply, because computable functions are continuous. Here, operations can help since there are non-trivial indeterministic tests which can be considered as finite precision versions of the ordinary tests with a small area of uncertainty. Of course, computable operations are also continuous in a well-defined sense, they are lower semi-continuous as set-valued functions.[4] In a certain sense, *indeterminism is the price for continuity.*

We will precisely introduce and discuss our recursive closure schemes, together with several deduced schemes, in Chapter 2. Altogether, our abstract language will consist of ten different closure schemes listed in Figure 1.3. The

projection	evaluation
juxtaposition	transposition
product	exponentiation
composition	iteration
inversion	sequentialization

Figure 1.3: Recursion operators

operators projection, juxtaposition and product will be used to handle finite product data-types in settings where no pairing mechanism within the structure is available. Composition, iteration and inversion correspond to the classical closure schemes of substitution, primitive recursion and minimization. Evaluation, transposition and exponentiation are infinite versions of projection, juxtaposition and product and will be used to handle sequential data-types. Finally, sequentialization is a special closure scheme which can be used to eliminate indeterminism in certain cases. Common to all these closure schemes is that they easily can be defined by purely set-theoretical means and that the only concrete set involved is the set of natural numbers.

One cannot expect that an abstract language yields reasonable results if it is applied to arbitrary structures. A suitable model of effectivity always has to offer both: effective initial operations, as well as effective closure schemes. If only the closure schemes are effective, then one obtains a model of relative computability and if even the closure schemes are non-effective, then only a definability model is obtained. In the approach presented here, the many-sorted structures that supply the initial operations will typically consist of the Peano structure $(\mathbb{N}, 0, n, n+1)$ on the natural numbers combined with other structures, like the structure of the real numbers $(\mathbb{R}, 0, 1, x+y, -x, x \cdot y, 1/x, \mathrm{Lim}, x < y)$, which will later be defined precisely.

At least two essential requirements have to be met by an abstract high-level language: on the one hand, it has to be *correct*, i.e. there has to be an

[4] A type of continuity which is investigated in set-valued analysis (cf. [Bee93]).

implementation of the initial operations, and on the other hand, it has to be *complete*, i.e. everything that can be realized in the concrete language can also be build up from the initial operations in the abstract language. In Chapter 3 we will single out a class of structures, the *perfect structures*, which meet both requirements. Consequently, *over perfect structures computable and recursive operations are in perfect harmony*: they coincide. And more than this, we will prove that perfect structures admit exactly one unique kind of effectivity, characterized by the structure itself.[5]

In Chapter 4 our theory becomes substantial since we will supply a standard way to construct perfect structures. These structures will be derived from recursive metric spaces, which can roughly be characterized as separable metric spaces with computable metric. They have been introduced by Lacombe [Lac59] and investigated in various similar versions by other authors [Mos64a, Mos64b, Wei93]. Some of the spaces which we will investigate are given in Figure 1.4.

set	**recursive points**	**topology**
$\mathbb{N}$	natural numbers	discrete topology
$\mathbb{R}^n$	recursive reals	Euclidean topology
$\mathcal{K}(\mathbb{R}^n)$	recursive compact sets	Vietoris topology
$\mathcal{A}(\mathbb{R}^n)$	recursive closed sets	Fell topology
$\mathcal{C}(\mathbb{R}^n)$	recursive functions	compact open topology

Figure 1.4: Spaces with perfect structures

Especially, perfect structures can be derived from recursive Banach spaces and this is the point where our theory comes in touch with the abstract computability theory of Pour-El and Richards [PER89]. Their theory is based on an axiomatic approach and the notion axiomatized is that of a computable sequence in Banach spaces. Sequences can be used to characterize computable functions since a function is computable, if it admits a computable modulus of continuity and if it is sequentially computable (it transforms computable sequences to computable sequences).[6] Unfortunately, this characterization works only for uniformly continuous functions, it becomes a little more technical for continuous functions with locally compact domain and it seems to be difficult to extend this approach to more general domains. One difference between

[5]Hertling proved directly that the structure of the real numbers admits a unique kind of effectivity [Her99a].

[6]Banach and Mazur's approach to computable analysis has been build upon the notion of sequential computability [BM37, Maz63].

Pour-El and Richards approach and other approaches, like the representation based approach and ours, is that Pour-El and Richards do not offer any special method to construct computable functions or computable Banach spaces. They formulate conditions, such that a given space or a given function is computable, if and only if it meets the condition. But the question how to construct such spaces and functions remains outside the axiomatic theory. What is common to Pour-El and Richards and our approach is that we obtain the same classes of computable sequences over separable Banach spaces. Both approaches consider computability as a further property of classical spaces and objects, analogously to algebraical, topological and other mathematical properties. And these computability properties are studied using classical logic, which distinguishes computable analysis from the constructive analysis of Bishop and Bridges [BB85] and the russian constructive analysis of Markov, Šanin, Ceitin, Kushner and others [Šan68, Cei64, Kus84].[7]

The emphasis in Chapter 4 will not be on structures and spaces which arise from analysis and physics, but on spaces which have been used in computable analysis. Nevertheless, many results can directly be applied to separable Banach and Hilbert spaces. In the same sense as functional analysis investigates spaces which appear in analysis, we will concentrate on function spaces which contain computable real-valued functions as recursive points and on hyperspaces which contain recursive sets of real numbers as computable points. Figure 1.4 shows some of these spaces, together with the corresponding classes of recursive points and the underlying topologies. A precise definition of the initial operations of the corresponding structures can be found in the Appendix. Some of our results have already been presented in preliminary, partly different and less comprehensive settings in [Bra95, Bra96, Bra97].

Acknowledgements

While working on this project, I enjoyed the productive atmosphere of our research group. Implicitly, I have incorporated many ideas of my colleagues Peter Hertling, Matthias Schröder, Xizhong Zheng and others. Fruitful discussions with Martin Hötzel Escardo, John V. Tucker, Viggo Stoltenberg-Hansen and Jeffrey Zucker have influenced my work. Finally, I am indebted to my supervisor Klaus Weihrauch for sharing his knowledge and supporting my research.

[7] A further approach to computable analysis has been made by Aberth [Abe80].

Chapter 2

Recursion Operators

In this chapter we will expose our abstract high-level language. First, we will give a short survey on the classical definition of recursive functions and then we will motivate the definitions of our generalized recursive closure schemes. The definitions of these schemes will be purely set-theoretical and for the following we will call them *recursion operators*. Finally, we will investigate some further closure schemes which can be derived from the recursion operators.

2.1 The classically recursive functions

Historically, Kleene first has defined total recursive functions and later on partial recursive functions by μ–recursion [Odi89]. We will immediately start with partial recursive functions. We denote a (potentially) partial function by $f :\subseteq X \to Y$, where the inclusion sign indicates, that f might be non-defined for some values $x \in X$, and if f is non-defined for a specific x, then we write $f(x) =\uparrow$. The composition $g \circ f(x)$ of two functions $f :\subseteq X \to Y$ and $g :\subseteq Y \to Z$ is defined, if and only if $y = f(x)$ is defined and $g(y)$ is defined. Moreover, we will identify constants $c \in Y$ with the zero-ary constant function $\{()\} \to Y, () \mapsto c$ with value c and we assume $X^0 = \{()\}$ for each set X. We will use the notation $\mathbb{N} := \{0, 1, 2, ...\}$ for the set of *natural numbers*.

Definition 2.1.1 (Classically recursive functions) The class of *classically recursive functions* $f :\subseteq \mathbb{N}^k \to \mathbb{N}$ is defined as the smallest class of functions

(1) which contains the *initial constants and functions*

$$
\begin{aligned}
&0 \in \mathbb{N},\\
&S : \mathbb{N} \to \mathbb{N}, n \mapsto n + 1,\\
&\mathrm{pr}_i^{(n)} : \mathbb{N}^n \to \mathbb{N}, (x_1, ..., x_n) \mapsto x_i,
\end{aligned}
$$

for all $n \geq i \geq 1$,

(2) and which is closed under

(a) *substitution*, that is the scheme that given $g_1, ..., g_m :\subseteq \mathbb{N}^n \to \mathbb{N}$ and $h :\subseteq \mathbb{N}^m \to \mathbb{N}$ produces $f :\subseteq \mathbb{N}^n \to \mathbb{N}$, defined by

$$f(x) := h(g_1(x), ..., g_m(x)),$$

(b) *primitive recursion*, that is the scheme that given $g :\subseteq \mathbb{N}^n \to \mathbb{N}$ and $h :\subseteq \mathbb{N}^{n+2} \to \mathbb{N}$ produces $f :\subseteq \mathbb{N}^{n+1} \to \mathbb{N}$, defined by

$$\left\{ \begin{array}{lcl} f(x,0) & := & g(x) \\ f(x,y+1) & := & h(x,y,f(x,y)) \end{array} \right. ,$$

(c) μ–*recursion*, that is the scheme that given $g :\subseteq \mathbb{N}^{n+1} \to \mathbb{N}$ produces $f :\subseteq \mathbb{N}^n \to \mathbb{N}$, defined by

$$\begin{array}{rcl} f(x) & := & \mu y[g(x,y) = 0 \text{ and } (\forall z < y) g(x,z) > 0] \\ & := & \left\{ \begin{array}{ll} \min(M) & \text{if } M \neq \emptyset \\ \uparrow & \text{otherwise} \end{array} \right. \end{array}$$

with $M := \{y : g(x,y) = 0 \text{ and } (\forall z < y) g(x,z) > 0\}$.

Correspondingly, the class of *primitive recursive* functions $f :\subseteq \mathbb{N}^k \to \mathbb{N}$ is defined as the smallest class of functions which contains the initial constants and functions from (1) and which is closed under substitution and primitive recursion.

It should be noticed that the constant zero function $Z : \mathbb{N} \to \mathbb{N}, n \mapsto 0$ can be generated from 0 and $\mathrm{pr}_2^{(2)}$ by primitive recursion. Occasionally, we can use the following version of Kleene's Normal Form Theorem [Odi89].

Theorem 2.1.2 (Kleene's Normal Form Theorem) *For each classically recursive function $f :\subseteq \mathbb{N}^k \to \mathbb{N}$ there is a primitive recursive total function $t : \mathbb{N}^{k+1} \to \mathbb{N}$ such that $f(x) = \mu y[t(x,y) = 0]$ for all $x \in \mathbb{N}^k$.*

There are many well-known equivalent modifications of the definition of classically recursive functions. For instance, Kleene and Gödel have independently proved that the primitive recursion scheme can be eliminated in presence of the following additional initial functions: sum, product and characteristic function of equality [Kle52, Odi89]. Julia Robinson has proved that inversion can be substituted for μ–recursion in presence of suitable initial functions [Rob50, Rob68]. Recursive closure schemes have also been discussed in detail by Rózsa Péter [Pét67, Pét81]. Recently, Sabadini, Vigna and Walters have given an elegant definition of recursive functions using the language of distributive categories [SVW96]. Recursive functions in abstract settings have been investigated by Tucker and Zucker (cf. [TZ88]). Especially, they have considered functions defined on many-sorted algebras.

2.2 Recursion operators for analysis

Now we want to discuss how we have to modify the classical recursive closure schemes such that they fit together with the needs of computable analysis. First, we want to take a look at some instructive examples of operations that are of interest in analysis:

(1) The *addition* $+ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ on the real numbers.

(2) The partial *limit operator* $\mathrm{Lim} :\subseteq \mathbb{R}^{\mathbb{N}} \to \mathbb{R}, (x_n)_{n\in\mathbb{N}} \mapsto \lim_{n\to\infty} x_n$ (restricted to rapidly converging Cauchy sequences).

(3) The *Mandelbrot function* $f : \mathbb{C} \to \mathbb{C}, z \mapsto z^2 + c$ and its iteration $f^* : \mathbb{C} \times \mathbb{N} \to \mathbb{C}, f^*(z, n) := f^n(z)$.

(4) The *integral operator* $\int : \mathcal{C}[0,1] \to \mathbb{R}, f \mapsto \int_0^1 f(x)\,dx$.

(5) The *Hausdorff metric* $d_{\mathcal{K}} : \mathcal{K}(\mathbb{R}) \times \mathcal{K}(\mathbb{R}) \to \mathbb{R}$.

(6) Partial *zero operators* $Z :\subseteq \mathcal{C}[0,1] \rightrightarrows \mathbb{R}$.

(7) Finite precision *tests* $t : \mathbb{R} \rightrightarrows \mathbb{N}$.

Here, the double arrow $\rightrightarrows$ denotes multi-valued functions. As discussed in the introduction, there are computable multi-valued functions which do not admit a computable single-valued selector. Thus, multi-valued functions are of some importance for computable analysis. Moreover, examples like the integral operator and the Hausdorff metric show that operations in analysis are typically many-sorted, and the limit operator, the iteration of the Mandelbrot function and the Hausdorff metric show that product as well as sequence data types are used. Altogether, our approach to recursion in analysis should offer *multi-valued operations* over *many-sorted structures* and *products and sequences* of data-types. However, the product data-types have to be handled explicitly by the recursion operators since there are no *pairing functions* in analysis available. For instance, there is no continuous and hence no computable injective function $f : \mathbb{R}^2 \to \mathbb{R}$. Several sets have been used in the previous examples of operations and for these sets one could consider structures like the following (without precise definition):

(1) *Natural numbers* $(\mathbb{N}, 0, n, n+1)$,

(2) *Real numbers* $(\mathbb{R}, 0, 1, x+y, -x, x\cdot y, 1/x, \mathrm{Lim}, <)$,

(3) *Compact non-empty subsets* $(\mathcal{K}(\mathbb{R}), \{x\}, A, A \cup B, d_{\mathcal{K}}, \mathrm{Lim})$,

(4) *Continuous functions* $(\mathcal{C}[0,1], 1, f, f+g, x \cdot f, f \cdot g, ||f||, \mathrm{Lim})$.

With these examples in mind we formulate the following concept for the definition of the generalized recursion operators:[1]

- **Initial operations** are exactly those from the underlying structure.
- **Recursion operators** should have simple set-theoretical definitions which generalize the classical closure schemes. They should handle products and sequences, as well as multi-valued operations.
- **Recursive operations** are those that can be generated from the initial operations by finitely many applications of the recursion operators.

The main goal is, of course, that the resulting abstract class of recursive operations coincide with the concrete class of computable operations, defined via Turing machines.

2.3 The definition of the operators

In this section we will introduce the extended and modified operators which result from the requirements discussed in the previous section. First, we have to introduce some technical notations. By $f :\subseteq X \rightrightarrows Y$ we will denote partial multi-valued operations which we will call for short *operations* in the following. Here the symbol $\subseteq$ indicates that the operation f is partial, and $\rightrightarrows$ indicates that f is multi-valued. More precisely, an operation $f :\subseteq X \rightrightarrows Y$ is a correspondence $f = (F, X, Y)$, that is $F \subseteq X \times Y$. 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 call X the *source* and Y the *target* of f. Moreover, we will use some notations for operations:

$$\begin{array}{lcl} \mathrm{dom}(f) & := & \{x \in X : (\exists y \in Y)\ (x,y) \in F\}, \\ \mathrm{range}(f) & := & \{y \in Y : (\exists x \in X)\ (x,y) \in F\}, \\ \mathrm{graph}(f) & := & F \end{array}$$

denotes the *domain*, *range*, and the *graph* of f, respectively and

$$\begin{array}{lcl} f(A) & := & \{y \in Y : (\exists x \in A)\ (x,y) \in F\}, \\ f^{-1}(B) & := & \{x \in X : (\exists y \in B)\ (x,y) \in F\} \end{array}$$

[1] Our former approach, presented in [Bra96], did not strictly distinguish structures and recursion operators and hence it did not meet the requirements formulated here.

denotes the *image*, *preimage* of $A \subseteq X, B \subseteq Y$ under f, respectively. By $f(x) := f\{x\} = \{y \in Y : (x,y) \in F\}$ we denote the *image* of x under f for each $x \in \mathrm{dom}(f)$. The image $f(x)$ is either non-defined in case $x \notin \mathrm{dom}(f)$ (which is denoted by $f(x) =\uparrow$ in the following), or $f(x) \neq \emptyset$ in case $x \in \mathrm{dom}(f)$. In the latter case, $\mathrm{range}(f) = \bigcup_{x \in \mathrm{dom}(f)} f(x)$. 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 $f = (F, X, Y)$ we associate the *inverse operation* $f^{-1} = (F^{-1}, Y, X)$, which is given by $F^{-1} := \{(y,x) : (x,y) \in F\}$. Thus, $f^{-1}(x) = f^{-1}\{x\}$ if $x \in \mathrm{dom}(f^{-1}) = \mathrm{range}(f)$ and $f^{-1}(x) =\uparrow$ else.

We will write $f \subseteq g$ for operations $f, g :\subseteq X \rightrightarrows Y$, if $\mathrm{dom}(f) \subseteq \mathrm{dom}(g)$ and $f(x) \subseteq g(x)$ for all $x \in \mathrm{dom}(f)$ (or equivalently, if $f = (F, X, Y), g = (G, X, Y)$ and $F \subseteq G$). We will say that g is an *extension* of f, if $\mathrm{dom}(f) \subseteq \mathrm{dom}(g)$ and $f(x) = g(x)$ for all $x \in \mathrm{dom}(f)$.

If X is a set then we denote by $X^{\mathbb{N}}$ the *set of all sequences* in X. We will write $\alpha = (x_n)_{n \in \mathbb{N}}$ for the sequence in X defined by $\alpha(n) = x_n$ for all $n \in \mathbb{N}$. We will not distinguish the set product $(X \times Y) \times Z$ from $X \times (Y \times Z)$. For each set X we define $X^0 := \{()\}$ and we assume $X^0 \times Y = Y \times X^0 = Y$ for each set Y.

In the following we will assume that U, V, X, Y, Z are arbitrary sets. All recursion operators 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 2.3.1 (Recursion operators) Define the following *recursion operators*:

(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 $x \in \mathrm{dom}(f_1) := \mathrm{dom}(f)$. Let the second projection $f_2 :\subseteq X \rightrightarrows Z$ be 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

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

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

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

It should be noticed that all operators are defined by purely set-theoretical constructions. The definitions of the first four operators, which will be called the *finite operators* in the following, do no refer to any special structure. The only structure which is involved explicitly in the definitions of the other operators, is the structure of the natural numbers. But this should come as no surprise, since the nature of computation incorporates the natural numbers and operators like iteration are defined canonically with the help of them. Obviously, it would be possible to "hide" or "simulate" the natural numbers by different techniques, but this is not the goal of the present approach.

Of course, some of our operators have well-known counterparts, defined at least for the special case of functions. They appear in various branches of mathematics and computer science.[2]

The operators projection, juxtaposition and product can be used to handle product data-types. If we associate with each operation $f :\subseteq X \times \mathbb{N} \rightrightarrows Y$ the sequence of operations $f_n :\subseteq X \rightrightarrows Y, x \mapsto f(n,x)$ with $n \in \mathbb{N}$, then the transposition $[f]$ corresponds to an infinite juxtaposition $(f_0, f_1, ...)$. Analogously, evaluation and exponentiation are infinite versions of projection and product in a certain sense. In other words, the operators evaluation, transposition and exponentiation can be used to handle sequence data-types. Composition, iteration and inversion correspond to the classical schemes of substitution, primitive recursion and minimization. Finally, sequentialization is an operator which can be used to eliminate indeterminism in certain cases. We will discuss each of these operators in detail in the following subsections (and we will reformulate their definitions for the special case of functions).

2.3.1 Projection

Projection is based on the idea that from a given result which consists of two components one can discard one component and keep the other one. Figure 2.1 illustrates the situation. If $f :\subseteq X \to Y \times Z$ is a function, then the projection

[2] The names "transposition" and "exponentiation" have been borrowed from category theory (cf. [McL92]) and transposition is well-known as "Curry-Operator" in the theory of semantic domains (cf. [GS90]). Juxtaposition of functions f and g is equal to the diagonal $f\Delta g$, sometimes used in topology, and exponentiation of f is equal to the infinite product $\prod_{i=0}^{\infty} f$ of f(cf. [Eng89]).

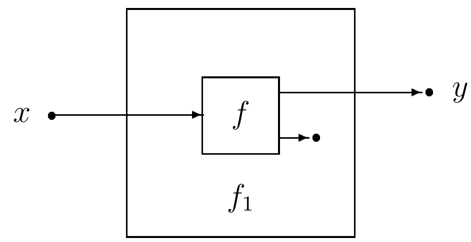

Figure 2.1: The projection operator

$f_1 :\subseteq X \to Y$ is defined by

$$f_1(x) = y :\iff (\exists z) f(x) = (y, z)$$

for all $x \in \mathrm{dom}(f_1) := \mathrm{dom}(f)$. In the classical setting projection is not an operator but an initial operation. From our point of view projection does not belong to an underlying structure but should always be available as a basic construction principle. In presence of identities $\mathrm{id}_X : X \to X, \mathrm{id}_Y : Y \to Y$ we can construct the *projection function* $\mathrm{pr}_1 : X \times Y \to X$ by applying product and projection operator:

$$\mathrm{pr}_1 = (\mathrm{id}_X \times \mathrm{id}_Y)_1.$$

Other projections can be constructed correspondingly. Occasionally, we will also use the notation f_n for the projection of an operation $f :\subseteq X \rightrightarrows Y_1 \times \ldots \times Y_n$ on its n-th component. If one of the sets Y_i itself is a product of sets, then the notation is not unique, but we will always indicate the precise meaning.

2.3.2 Juxtaposition

In a certain sense juxtaposition is the inverse of projection. It is based on the idea that we can combine two computations with one input to a parallel one with the same input and with two output components. Figure 2.2 illustrates the situation. If $f :\subseteq X \to Y$ and $g :\subseteq X \to Z$ are functions, then the juxtaposition $(f, g) :\subseteq X \to Y \times Z$ is defined by

$$(f, g)(x) := (f(x), g(x))$$

for all $x \in \mathrm{dom}(f, g) := \mathrm{dom}(f) \cap \mathrm{dom}(g)$. In the classical setting juxtaposition has not been considered explicitly since in presence of bijective computable pairing functions juxtaposition can be simulated. However, in our setting there are no suitable pairing functions in general and juxtaposition is useful to handle many-sorted products. As we will see later, in presence of juxtaposition the classical substitution scheme can be replaced by the easier composition scheme and the classical primitive recursion scheme can be replaced by the easier iteration scheme.

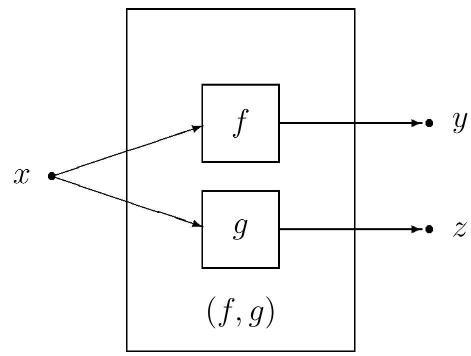

Figure 2.2: The juxtaposition operator

2.3.3 Product

The product operator is quite similar to the juxtaposition operator with the only difference that the parallel computation is performed on two different inputs. Figure 2.3 illustrates the situation. If $f :\subseteq X \to Y$ and $g :\subseteq U \to V$

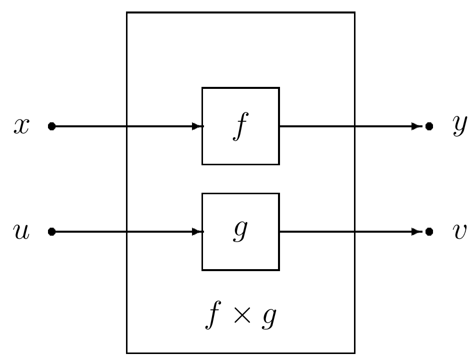

Figure 2.3: The product operator

are functions, then the product $f \times g :\subseteq X \times U \to Y \times V$ is defined by

$$(f \times g)(x, u) := (f(x), g(u))$$

for all $(x, u) \in \mathrm{dom}(f \times g) := \mathrm{dom}(f) \times \mathrm{dom}(g)$. Product and juxtaposition are closely related; one obtains the formulas $f \times g = (f \circ \mathrm{pr}_1, g \circ \mathrm{pr}_2)$, where $\mathrm{pr}_i := (\mathrm{id} \times \mathrm{id})_i$ for $i = 1, 2$ and $(f, g) = (f \times g) \circ \mathrm{in}$, where $\mathrm{in} := (\mathrm{id}, \mathrm{id})$.

2.3.4 Composition

The composition operator is based on the idea that computations can be performed successively such that the output of the first computation is supplied

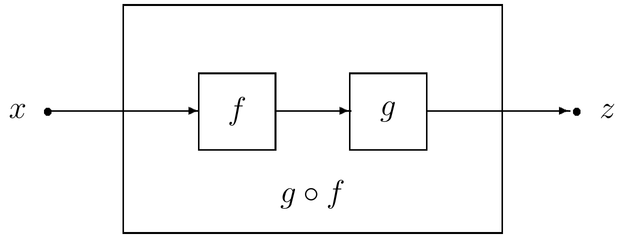

Figure 2.4: The composition operator

as input to the second one. Figure 2.4 illustrates the situation. If $f :\subseteq X \to Y$ and $g :\subseteq Y \to Z$ are functions, then the composition $g \circ f :\subseteq X \to Z$ is defined by

$$(g \circ f)(x) := g(f(x))$$

for all $x \in \mathrm{dom}(g \circ f) := \{x : f(x) \in \mathrm{dom}(g)\}$. In comparison with the classical setting substitution is replaced by composition which is possible in presence of the other operations.

2.3.5 Iteration

The iteration is the first operator which explicitly involves the structure of the natural numbers. It is defined as an inductive repetition of the composition of an operation. If $f :\subseteq X \to X$ is a function, then the iteration $f^* :\subseteq X \times \mathbb{N} \to X$ is defined by

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

for all $x \in X$ and $n \in \mathbb{N}$. In comparison with the classical setting primitive recursion is replaced by iteration.

2.3.6 Inversion

The inversion operator is the only one which does not preserve functionality, that is, the twisted inversion $f^{\leftrightarrow}$ of a function might be a multi-valued operation. If $f :\subseteq X \times \mathbb{N} \to Y \times \mathbb{N}$ is a function, then we will say that f is *injective in the last component*, if $f(x,n) = f(x,m) \Longrightarrow n = m$ holds for all $(x,n), (x,m) \in \mathrm{dom}(f)$. Occasionally, we will call the inversion operator *injective inversion*, if it is restricted to functions which are injective in the last component. In this case the inversion $f^{\leftrightarrow} :\subseteq X \times \mathbb{N} \to Y \times \mathbb{N}$ is a function which is defined by

$$f^{\leftrightarrow}(x,n) := (y,k) :\Longleftrightarrow \ f(x,k) = (y,n) \text{ and } (\forall i)\ (x,i) \in \mathrm{dom}(f)$$

for all $(x, n), (y, k)$. Moreover, for total functions $f : \mathbb{N} \to \mathbb{N}$ we have $f^{\leftrightarrow}(n) = f^{-1}\{n\}$ and if f, additionally, is injective, then twisted inversion is equal to ordinary inversion $f^{\leftrightarrow} = f^{-1} :\subseteq \mathbb{N} \to \mathbb{N}$. In comparison with the classical setting μ-recursion is replaced by inversion.

2.3.7 Evaluation

The evaluation is the first operator which involves sequence sets. It is a natural generalization of the projection operator to infinite products. If $f :\subseteq X \to Y^{\mathbb{N}}$ is a function, then the evaluation $f_* :\subseteq X \times \mathbb{N} \to Y$ is defined by

$$f_*(x, n) := f(x)(n)$$

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

2.3.8 Transposition

The transposition is a natural generalization of the juxtaposition operator to sequence sets. If $f :\subseteq X \times \mathbb{N} \to Y$ is a function, then the transposition $[f] :\subseteq X \to Y^{\mathbb{N}}$ is defined by

$$[f](x)(n) := f(x, n)$$

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

2.3.9 Exponentiation

The exponentiation operator is a natural generalization of the product operator to sequence sets. If $f :\subseteq X \to Y$ is a function, then the exponentiation $f^{\mathbb{N}} :\subseteq X^{\mathbb{N}} \to Y^{\mathbb{N}}$ is defined by

$$f^{\mathbb{N}}(\alpha)(n) := f(\alpha(n))$$

for all $\alpha \in \mathrm{dom}(f^{\mathbb{N}}) := \{\alpha : \mathrm{range}(\alpha) \subseteq \mathrm{dom}(f)\}$.

2.3.10 Sequentialization

The sequentialization operator is an operator which can be used to eliminate indeterminism in the following sense: if $f :\subseteq X \rightrightarrows \mathbb{N}$ is an operation, then $f^{\triangle}(x)$ is the set of all sequences which enumerate the image $f(x)$. Thus, we can pick some sequence $(y_n)_{n \in \mathbb{N}} \in f^{\triangle}(x)$ and we have an enumeration of the complete image $f(x)$. However, $f^{\triangle}$ is multi-valued if f is. The operator of sequentialization will only be used in very special situations. If we consider functions $f :\subseteq X \to \mathbb{N}$, then sequentialization is redundant, since it can be constructed by transposition via $f^{\triangle} = [(f \times \mathrm{id})_1]$.

2.4 Elementary properties of the operators

In this subsection we will study some elementary properties of our operators. Essentially, we will investigate some algebraic properties of the finite operators projection, juxtaposition, product, and composition. The operators juxtaposition, product, and composition are associative. Unfortunately, juxtaposition and composition do only provide a weak kind of distributivity, which could be called *semi-distributivity* (cf. (7) of the following proposition). This deficiency is due to the indeterminism of operations. All proofs are easy and straightforward. For completeness we sketch some examples.

Proposition 2.4.1 (Algebraic properties of operators) *For operations* e, f, g, h *to which the corresponding operators apply, the following holds:*

(1) $((f,g),h) = (f,(g,h))$,

(2) $(f \times g) \times h = f \times (g \times h)$,

(3) $(h \circ g) \circ f = h \circ (g \circ f)$,

(4) $f \supseteq (f,g)_1$ *and* $g \supseteq (f,g)_2$,

(5) $f \subseteq (f_1, f_2)$,

(6) $f \times g = ((f \times g)_1, (f \times g)_2)$,

(7) $(f,g) \circ h \subseteq (f \circ h, g \circ h)$,

(8) $(g \times h) \circ (e \times f) = (g \circ e) \times (h \circ f)$.

If $\mathrm{dom}(f) = \mathrm{dom}(g)$ *in (4), then equality holds; if* f_1 *or* f_2 *is a function in (5), then equality holds; if* h, $f \circ h$, *or* $g \circ h$ *is a function in (7), then equality holds.*

Proof.

(1) Follows since we identify $(X \times Y) \times Z$ with $X \times (Y \times Z)$ and since $(X \cap Y) \cap Z = X \cap (Y \cap Z)$.

(2) Analogously to (1).

(3) Let $f :\subseteq X \rightrightarrows Y, g :\subseteq Y \rightrightarrows Z, h :\subseteq Z \rightrightarrows W$ be operations. Then

$$\begin{aligned} w \in (h \circ (g \circ f))(x) &\iff (\exists z \in (g \circ f)(x)) w \in h(z) \\ &\iff (\exists z)(\exists y \in f(x)) z \in g(y) \text{ and } w \in h(z) \\ &\iff (\exists y \in f(x))(\exists z \in g(y)) w \in h(z) \\ &\iff (\exists y \in f(x)) w \in (h \circ g)(y) \\ &\iff w \in ((h \circ g) \circ f)(x) \end{aligned}$$

and

$$\begin{aligned}
& x \in \mathrm{dom}(h \circ (g \circ f)) \\
\iff & (g \circ f)(x) \subseteq \mathrm{dom}(h) \\
\iff & x \in \mathrm{dom}(g \circ f) \text{ and } (\forall y \in f(x)) g(y) \subseteq \mathrm{dom}(h) \\
\iff & f(x) \subseteq \mathrm{dom}(h \circ g) \\
\iff & x \in \mathrm{dom}((h \circ g) \circ f).
\end{aligned}$$

(4) One obtains $\mathrm{dom}(f,g)_1 = \mathrm{dom}(f,g) = \mathrm{dom}(f) \cap \mathrm{dom}(g) \subseteq \mathrm{dom}(f)$ and for all $x \in \mathrm{dom}(f,g)_1$

$$(f,g)_1(x) = \{y : (\exists z)(y,z) \in (f,g)(x) = f(x) \times g(x)\} = f(x)$$

holds. The projection on the second component is treated analogously.

(5) Obviously, $\mathrm{dom}(f) = \mathrm{dom}(f_1) \cap \mathrm{dom}(f_2) = \mathrm{dom}(f_1, f_2)$ and

$$f(x) \subseteq f_1(x) \times f_2(x) = (f_1, f_2)(x)$$

for all $x \in \mathrm{dom}(f)$.

(6) Is easy to prove.

(7) Let $f :\subseteq X \rightrightarrows Y, g :\subseteq X \rightrightarrows Z, h :\subseteq W \rightrightarrows X$ be operations, then

$$\begin{aligned}
w \in \mathrm{dom}((f,g) \circ h) & \iff h(w) \subseteq \mathrm{dom}(f,g) = \mathrm{dom}(f) \cap \mathrm{dom}(g) \\
& \iff h(w) \subseteq \mathrm{dom}(f) \text{ and } h(w) \subseteq \mathrm{dom}(g) \\
& \iff w \in \mathrm{dom}(f \circ h) \text{ and } w \in \mathrm{dom}(g \circ h) \\
& \iff w \in \mathrm{dom}((f \circ h) \times (g \circ h))
\end{aligned}$$

and

$$\begin{aligned}
(f,g) \circ h(x) &= \bigcup_{y \in h(x)} (f,g)(y) \\
&= \bigcup_{y \in h(x)} (f(y) \times g(y)) \\
&\subseteq \bigcup_{y \in h(x)} f(y) \times \bigcup_{y \in h(x)} g(y) \\
&= ((f \circ h) \times (g \circ h))(x)
\end{aligned}$$

for all $x \in \mathrm{dom}((f,g) \circ h)$

(8) Is easy to prove.

□

By (4) and (5) projection can be considered as inverse to juxtaposition in a restricted sense. As stated before, juxtaposition and composition, applied to operations, do only fulfill a weak kind of distributivity, as expressed in (7) and for a special case in (5). In case of $(f, g) \circ h$ the computation of f and g is performed with the same (indeterministic) result of h (cf. Figure 2.5).

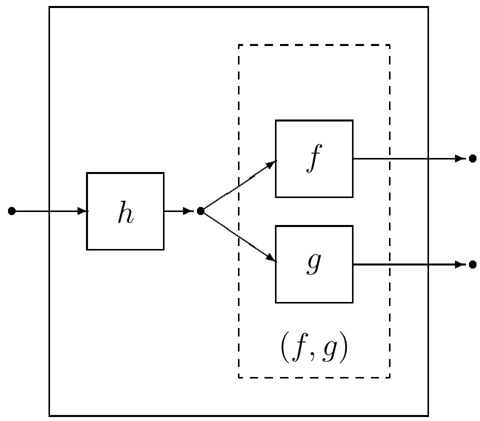

Figure 2.5: $(f, g) \circ h$

In case of $(f \circ h, g \circ h)$ the computation of f and g is performed with two independently determined indeterministic results of h (cf. Figure 2.6). Thus,

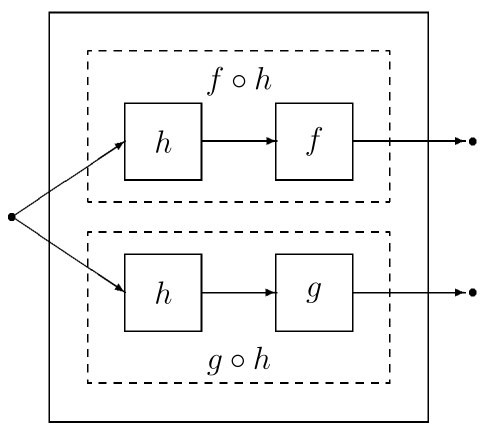

Figure 2.6: $(f \circ h, g \circ h)$

in general $(f, g) \circ h \subseteq (f \circ h, g \circ h)$ and it is easy to construct examples such that "$\supseteq$" does not hold.

2.5 Further operators

In this section we will discuss some further operators which are inspired by classical operators or which are special cases of the other recursion operators. We will compare some of these operators in a *natural setting*, where certain *essential* functions and operators are available: that is the constant 0 (i.e. the function $0 : \{()\} \to \mathbb{N}, () \mapsto 0$), the successor function $S : \mathbb{N} \to \mathbb{N}, n \mapsto n + 1$, and the identities $\mathrm{id}_X : X \to X, x \mapsto x$ for each set X, as well as the finite operators projection, juxtaposition, product, and composition. There is no doubt that everything that can be constructed in a natural setting is constructive in a strong sense. Especially, in natural settings each projection $\mathrm{pr}_1 : X \times Y \to X, (x, y) \mapsto x$ can be constructed by $\mathrm{pr}_1 = (\mathrm{id}_X \times \mathrm{id}_Y)_1$ and each constant function $c : X \to \mathbb{N}, x \mapsto n$ can be constructed by $c = (\mathrm{id}_X \times (S^n \circ 0))_2$.

2.5.1 Substitution

Substitution can be generalized to arbitrary operations by the following definition.

Definition 2.5.1 (Substitution) An operation $f :\subseteq X \rightrightarrows Z$ is defined by *substitution* from operations $g_i :\subseteq X \rightrightarrows Y_i$, $i = 1, ..., m$ and $h :\subseteq Y_1 \times ... \times Y_m \rightrightarrows Z$, if $f = h \circ (g_1, ..., g_m)$.

From the definition one can immediately deduce the following result.

Proposition 2.5.2 (Substitution) *Substitution can be constructed by juxtaposition and composition.*

The following diagram illustrates the situation:

$$\boxed{\text{composition}} + \boxed{\text{juxtaposition}} \to \boxed{\text{substitution}}$$

Occasionally, we will write $f(x) = h(g_1(x), ..., g_m(x))$, which should always be understood as short form of $f(x) = h \circ (g_1, ..., g_m)(x)$.

2.5.2 Primitive recursion

Primitive recursion can also be generalized straightforwardly to arbitrary operations.

Definition 2.5.3 (Primitive recursion) Let $g :\subseteq X \rightrightarrows Y$, $h :\subseteq X \times \mathbb{N} \times Y \rightrightarrows Y$ be operations. Then $f :\subseteq X \times \mathbb{N} \rightrightarrows Y$ is defined by *primitive recursion* from g, h, if

$$\begin{cases} f(x,0) & := & g(x) \\ f(x,n+1) & := & h(x,n,f(x,n)) \end{cases}$$

for all $x \in X, n \in \mathbb{N}$.

The next proposition shows that iteration is a substitute for primitive recursion in our setting.

Proposition 2.5.4 (Primitive recursion) *In natural settings primitive recursion can be constructed by iteration.*

Proof. Let $g :\subseteq X \rightrightarrows Y, h :\subseteq X \times \mathbb{N} \times Y \rightrightarrows Y$ be operations and let $f :\subseteq X \times \mathbb{N} \to Y$ be obtained from g, h by primitive recursion. Define $P :\subseteq X \times \mathbb{N} \times Y \rightrightarrows X \times \mathbb{N} \times Y$ by

$$P := (\mathrm{pr}_1, S \circ \mathrm{pr}_2, h)$$

and $F :\subseteq X \times \mathbb{N} \rightrightarrows Y$ by

$$F := \mathrm{pr}_3 \circ P^* \circ (\mathrm{pr}_1, Z \circ \mathrm{pr}_2, g \circ \mathrm{pr}_1, \mathrm{pr}_2).$$

We claim that $F = f$. First, we prove

$$(\mathrm{pr}_1, \mathrm{pr}_2) \circ P^*(x,n,y,k) = (x, n+k)$$

for all $x \in X, y \in Y, n, k \in \mathbb{N}$ by induction on k. This especially proves that $(\mathrm{pr}_1, \mathrm{pr}_2) \circ P^*$ is a function.

$\underline{k = 0}$ $(\mathrm{pr}_1, \mathrm{pr}_2) \circ P^*(x,n,y,0) = (x,n)$
$\underline{k \to k+1}$ We obtain

$$\begin{aligned} & (\mathrm{pr}_1, \mathrm{pr}_2) \circ P^*(x,n,y,k+1) \\ = \; & (\mathrm{pr}_1, \mathrm{pr}_2) \circ P \circ P^*(x,n,y,k) \\ = \; & (\mathrm{pr}_1, \mathrm{pr}_2) \circ (\mathrm{pr}_1, S \circ \mathrm{pr}_2, h) \circ P^*(x,n,y,k) \\ = \; & (\mathrm{pr}_1, S \circ \mathrm{pr}_2) \circ P^*(x,n,y,k) \\ = \; & (\mathrm{id}_X \times S) \circ (\mathrm{pr}_1, \mathrm{pr}_2) \circ P^*(x,n,y,k) \\ = \; & (\mathrm{id}_X \times S)(x, n+k) \quad \text{by induction hypothesis} \\ = \; & (x, n+k+1). \end{aligned}$$

Now we prove $F(x,n) = f(x,n)$ for all $x \in X, n \in \mathbb{N}$ by induction on n.
$\underline{n = 0}$ We obtain

$$\begin{aligned} F(x,0) &= \mathrm{pr}_3 \circ P^*(x,0,g(x),0) \\ &= \mathrm{pr}_3(x,0,g(x)) \\ &= g(x) \\ &= f(x,0). \end{aligned}$$

$\underline{n \to n+1}$ We obtain

$$\begin{aligned} F(x,n+1) &= \mathrm{pr}_3 \circ P^*(x,0,g(x),n+1) \\ &= \mathrm{pr}_3 \circ P \circ P^*(x,0,g(x),n) \\ &= \mathrm{pr}_3 \circ P \circ (\mathrm{pr}_1,\mathrm{pr}_2,\mathrm{pr}_3) \circ P^*(x,0,g(x),n) \\ &= \mathrm{pr}_3 \circ P \circ ((\mathrm{pr}_1,\mathrm{pr}_2) \circ P^* \circ (\mathrm{pr}_1, Z \circ \mathrm{pr}_2, g \circ \mathrm{pr}_1, \mathrm{pr}_2), \\ &\qquad \mathrm{pr}_3 \circ P^* \circ (\mathrm{pr}_1, Z \circ \mathrm{pr}_2, g \circ \mathrm{pr}_1, \mathrm{pr}_2))(x,n), \\ &\quad \text{since } (\mathrm{pr}_1,\mathrm{pr}_2) \circ P^* \circ (\mathrm{pr}_1, Z \circ \mathrm{pr}_2, g \circ \mathrm{pr}_1, \mathrm{pr}_2) \text{ is a function} \\ &= \mathrm{pr}_3 \circ P \circ (\mathrm{pr}_1,\mathrm{pr}_2,F)(x,n), \\ &\quad \text{since } (\mathrm{pr}_1,\mathrm{pr}_2) \circ P^* \circ (\mathrm{pr}_1, Z \circ \mathrm{pr}_2, g \circ \mathrm{pr}_1, \mathrm{pr}_2) = (\mathrm{pr}_1,\mathrm{pr}_2) \\ &= \mathrm{pr}_3 \circ (\mathrm{pr}_1, S \circ \mathrm{pr}_2, h) \circ (\mathrm{pr}_1,\mathrm{pr}_2,f)(x,n), \\ &\quad \text{by induction hypothesis} \\ &= h \circ (\mathrm{pr}_1,\mathrm{pr}_2,f)(x,n) \\ &= h(x,n,f(x,n)) \\ &= f(x,n+1) \end{aligned}$$

□

The following diagram illustrates the situation:

$$\boxed{\text{essentials}} + \boxed{\text{iteration}} \to \boxed{\text{primitive recursion}}$$

2.5.3 Simultaneous recursion

In classical recursion theory simultaneous recursion is another important construction scheme where several functions are constructed simultaneously via primitive recursion (cf. [Odi89, Pét67]). Simultaneous recursion can also be generalized straightforwardly to arbitrary operations.

Definition 2.5.5 (Simultaneous recursion) Let $g_1, ..., g_m :\subseteq X \rightrightarrows Y$, h_1, ..., $h_m :\subseteq X \times \mathbb{N} \times Y^m \rightrightarrows Y$ be operations with $\mathrm{dom}(g_1) = ... = \mathrm{dom}(g_m)$

and $\mathrm{dom}(h_1) = ... = \mathrm{dom}(h_m)$. Then $f_1, ..., f_m :\subseteq X \times \mathbb{N} \rightrightarrows Y$ are defined by *simultaneous recursion* from $g_1, ..., g_m, h_1, ..., h_m$, if

$$\left\{ \begin{array}{l} \left\{ \begin{array}{lcl} f_1(x,0) & := & g_1(x) \\ & \vdots & \\ f_m(x,0) & := & g_m(x) \end{array} \right. \\ \\ \left\{ \begin{array}{lcl} f_1(x,n+1) & := & h_1(x,n,f_1(x,n),...,f_m(x,n)) \\ & \vdots & \\ f_m(x,n+1) & := & h_m(x,n,f_1(x,n),...,f_m(x,n)) \end{array} \right. \end{array} \right.$$

for all $x \in X, n \in \mathbb{N}$.

It is well-known that classically recursive functions are closed under simultaneous recursion but that simultaneous recursion cannot be replaced by primitive recursion in an abstract setting without pairing operation (cf. [TZ88]). In our setting the pairing operations have been replaced by the juxtaposition operator.

Proposition 2.5.6 (Simultaneous recursion) *Simultaneous recursion can be constructed by projection, juxtaposition and primitive recursion.*

Proof. If we define $f :\subseteq X \times \mathbb{N} \rightrightarrows Y^m$ by

$$\left\{ \begin{array}{lcl} f(x,0) & := & (g_1,...,g_m)(x) \\ f(x,n+1) & := & (h_1,...,h_m)(x,n,f(x,n)) \end{array} \right. ,$$

then the projections $f_1, ..., f_m$ of f are just the operations which are defined from $g_1, ..., g_m, h_1, ..., h_m$ by simultaneous recursion (provided that $\mathrm{dom}(g_1) = ... = \mathrm{dom}(g_m)$ and $\mathrm{dom}(h_1) = ... = \mathrm{dom}(h_m)$). $\square$

Consequently, the following diagram illustrates the situation:

$$\boxed{\text{projection}} + \boxed{\text{juxtaposition}} + \boxed{\begin{array}{c}\text{primitive}\\ \text{recursion}\end{array}} \rightarrow \boxed{\begin{array}{c}\text{simultaneous}\\ \text{recursion}\end{array}}$$

2.5.4 Definition by cases

In classical recursion theory it is usual to define recursive functions by cases which can be distinguished by recursive predicates. We can formulate a generalized version of this property (and we will prove that it is a special case of primitive recursion).

Definition 2.5.7 (Definition by cases) Let $f, g :\subseteq X \rightrightarrows Y$, $t : X \rightrightarrows \mathbb{N}$ be operations with $\mathrm{dom}(g) \subseteq \mathrm{dom}(f)$. Then $(f|_t\, g) :\subseteq X \rightrightarrows Y$ is defined by

$$(f|_t\, g)(x) := \begin{cases} f(x) & \text{if} \quad \{0\} = t(x) \\ g(x) & \text{if} \quad \{0\} \not\subseteq t(x) \\ f(x) \cup g(x) & \text{else} \end{cases}$$

for all $x \in \mathrm{dom}(f|_t\, g) := \{x : (\{0\} \subseteq t(x) \Longrightarrow x \in \mathrm{dom}(f))$ and $(\{0\} \neq t(x) \Longrightarrow x \in \mathrm{dom}(g))\}$.

We will call $(.|.)$ the *definition-by-cases operator*. The next proposition shows that the definition-by-cases operator can be constructed by primitive recursion.

Proposition 2.5.8 (Definition by cases) *In natural settings the definition-by-cases operator can be constructed by primitive recursion.*

Proof. Let $f, g :\subseteq X \rightrightarrows Y, t : X \rightrightarrows \mathbb{N}$ be operations with $\mathrm{dom}(g) \subseteq \mathrm{dom}(f)$. Define $h :\subseteq X \times \mathbb{N} \rightrightarrows Y$ by

$$\begin{cases} h(x, 0) & := f(x) \\ h(x, n+1) & := g(x) \end{cases}$$

for all $x \in X, n \in \mathbb{N}$. Then $(f|_t\, g)(x) = h(x, t(x))$ for all $x \in X$. $\square$

2.5.5 Course-of-value recursion

In classical recursion theory course-of-value recursion is considered as further recursive definition scheme. It generalizes primitive recursion in the sense that the definition of the value $f(x, n+1)$ does not only depend on the predecessor value $f(x, n)$ but possibly on the complete "history" $f(x, 0), ..., f(x, n)$ of values. One can prove that classically recursive functions are closed under course-of-value recursion since the pairing function allows to code sequences of numbers of arbitrary finite length. Of course, we cannot simulate this idea by juxtaposition since only finite juxtapositions of fixed arity are allowed. Nevertheless, we can simulate course-of-value recursion with the help of sequences in the following sense.

Definition 2.5.9 (Course-of-value recursion) Let $g :\subseteq X \rightrightarrows Y$, $h :\subseteq X \times \mathbb{N} \times Y^{\mathbb{N}} \rightrightarrows Y$ be operations. Then $f :\subseteq X \times \mathbb{N} \rightrightarrows Y$ is defined by *course-of-value recursion* from g, h, if

$$\begin{cases} f(x, 0) & := & g(x) \\ f(x, n+1) & := & h(x, n, (f(x, 0), ..., f(x, n), f(x, n), ...)) \end{cases}$$

for all $x \in X$, $n \in \mathbb{N}$.

Proposition 2.5.10 (Course-of-value recursion) *In natural settings course-of-value recursion can be constructed by evaluation, transposition, and primitive recursion.*

Proof. Let $g :\subseteq X \rightrightarrows Y$, $h :\subseteq X \times \mathbb{N} \times Y^{\mathbb{N}} \rightrightarrows Y$ be operations and let $f :\subseteq X \times \mathbb{N} \rightrightarrows Y$ be defined by course-of-value recursion from g, h. Define $\check{g} :\subseteq X \times \mathbb{N} \rightrightarrows Y$, $\check{h} :\subseteq X \times \mathbb{N} \times Y^{\mathbb{N}} \times \mathbb{N} \rightrightarrows Y$ by

$$\check{g}(x,n) := g(x), \quad \check{h}(x,n,y,k) := \begin{cases} (\mathrm{id}_Y^{\mathbb{N}})_*(y,k) & \text{if } k \leq n \\ h(x,n,y) & \text{else} \end{cases}$$

and $\hat{f} :\subseteq X \times \mathbb{N} \rightrightarrows Y^{\mathbb{N}}$ by

$$\begin{cases} \hat{f}(x,0) & := & [\check{g}](x) \\ \hat{f}(x,n+1) & := & [\check{h}](x,n,\hat{f}(x,n)) \end{cases}$$

for all $x \in X, n \in \mathbb{N}$. Then $\hat{f}(x,n) = (f(x,0),...,f(x,n),f(x,n),...)$ and $f(x,n) = \hat{f}_*(x,n,n)$ for all $x \in X, n \in \mathbb{N}$. $\square$

Consequently, the following diagram illustrates the situation:

$$\boxed{\text{essentials}} + \boxed{\begin{array}{c}\text{primitive}\\ \text{recursion}\end{array}} + \boxed{\begin{array}{c}\text{evaluation,}\\ \text{transposition}\end{array}} \rightarrow \boxed{\begin{array}{c}\text{course-of-value}\\ \text{recursion}\end{array}}$$

2.5.6 Minimization

In this subsection we show that we can generalize the classical least number operator (i.e. μ-recursion) almost straightforwardly to arbitrary operations. First we generalize the operator only to functions.

Definition 2.5.11 (μ-recursion) If $f :\subseteq X \times \mathbb{N} \to \mathbb{N}$ is a function, then the *μ-recursion* $\mu f :\subseteq X \to \mathbb{N}$ is defined by

$$\mu f(x) := \min\{k : f(x,k) = 0 \text{ and } f(x,0),...,f(x,k-1) > 0\}$$

for all $x \in \mathrm{dom}(\mu f) := \{x : (\exists k) f(x,k) = 0 \text{ and } (x,0),...,(x,k-1) \in \mathrm{dom}(f)\}$.

If μ-recursion is applied only to total functions, then we will call it *total μ-recursion*. The following proposition essentially shows that total μ-recursion can be constructed by (injective) inversion. We will use the *arithmetical difference* $\dot{-} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, defined by $x \dot{-} y := \max\{0, x-y\}$ for all $x, y \in \mathbb{N}$. It is well-known that the arithmetical difference is primitive recursive.

Proposition 2.5.12 (μ-recursion) *In natural settings total μ-recursion can be constructed by primitive recursion and (injective) inversion.*

Proof. Let $f : X \times \mathbb{N} \to \mathbb{N}$ be a function. Define $g : X \times \mathbb{N} \to \mathbb{N}$ by

$$\begin{cases} g(x,0) & := & 1 \dot{-} f(x,0) \\ g(x,n+1) & := & (1 \dot{-} f(x,n+1)) + 2 \cdot g(x,n) \end{cases}$$

for all $x \in X$ and $n \in \mathbb{N}$. Then $g(x,n) = 1 \iff n = \mu f(x)$ for all x, n. Define $h : X \times \mathbb{N} \to \mathbb{N}$ by definition by cases:

$$h(x,n) := \begin{cases} n+1 & \text{if } g(x,n) \neq 1 \\ 0 & \text{if } g(x,n) = 1 \end{cases}$$

for all $(x,n) \in \mathrm{dom}(h) := \mathrm{dom}(g)$. Then h is injective in the last component since $|g^{-1}\{1\}| \leq 1$. Moreover,

$$\begin{aligned} (x,0) \in \mathrm{dom}(h^{\leftrightarrow}) & \iff & (\exists n) g(x,n) = 1 \\ & \iff & (\exists n) f(x,n) = 0 \\ & \iff & x \in \mathrm{dom}(\mu f) \end{aligned}$$

and

$$h^{\leftrightarrow}(x,0) = \min\{n : f(x,n) = 0\} = \mu f(x)$$

for all $x \in \mathrm{dom}(\mu f)$. Consequently, in natural settings total μ-recursion can be obtained from primitive recursion and injective inversion. $\square$

Vice versa, it is easy to see that in natural settings injective inversion can be obtained by μ-recursion. Now we will generalize μ-recursion to general operations. For this we have to take into account that in the case of general operations juxtaposition and composition are not distributive but only semi-distributive.

Definition 2.5.13 (Minimization) If $f :\subseteq X \times \mathbb{N} \rightrightarrows Y \times \mathbb{N}$ is an operation, then the *minimization* $f_{\min} :\subseteq X \rightrightarrows Y$ is defined by

$$f_{\min}(x) := \{y : (\exists n)\ (y,0) \in f(x,n) \text{ and } f(x,0), ..., f(x,n-1) \not\subseteq Y \times \{0\}\}$$

for all $x \in \mathrm{dom}(f_{\min}) := \{x : (\exists n)\ f(x,n) \subseteq Y \times \{0\}$ and $(x,0), ..., (x,n-1) \in \mathrm{dom}(f)\}$.

It is easy to see that minimization and μ-recursion are equivalent on functions. Minimization is even closer to while loops in programming. Consider an operation $f :\subseteq X \times \mathbb{N} \rightrightarrows Y \times \mathbb{N}$ and let $(y,k) \in f(x,n)$. Then y is one

```
input x;
n := 0;
repeat
    choose (y, k) ∈ f(x, n);
    n := n + 1
until k = 0;
output y.
```

Figure 2.7: Minimization program

possible result of the computation with input x and loop index n. The result of the minimization $f_{\min}(x)$ simply consists of all valid results y which are indicated by $k = 0$. Consider the program given in Figure 2.7. In this situation $f_{\min}(x)$ contains all possible results y of the program, in contrast to μ-recursion which would yield the corresponding indices n. While in the functional case the results y can be retrieved from the index n, the information contained in the index does not suffice in the multi-valued case. Here some indeterministic choices of $(y, k) \in f(x, n)$ may yield valid results while others yield invalid results. So, it is necessary to collect the valid results y directly. More formally, the reason that retrieving is impossible is that $f \subseteq (f_1, f_2)$ but "$\supseteq$" does not hold in general. If minimization is only applied to total operations, then we will call it *total minimization.* The next proposition generalizes the previous proposition in case of general operations.

Proposition 2.5.14 (Minimization) *In natural settings total minimization can be constructed by primitive recursion and inversion.*

Proof. Let $f : X \times \mathbb{N} \rightrightarrows Y \times \mathbb{N}$ be an operation and define $g : X \times \mathbb{N} \rightrightarrows \mathbb{N}$ by primitive recursion:

$$\left\{ \begin{array}{lcl} g(x, 0) & := & 1 \dot{-} f_2(x, 0) \\ g(x, n+1) & := & (1 \dot{-} f_2(x, n+1)) + 2 \cdot g(x, n) \end{array} \right.$$

for all $x \in X$ and $n \in \mathbb{N}$. Define $g' : Y \times \mathbb{N} \times X \times \mathbb{N} \rightrightarrows Y \times \mathbb{N}$ by

$$\left\{ \begin{array}{lcl} g'(y, k, x, 0) & := & (y, 1 \dot{-} k) \\ g'(y, k, x, n+1) & := & (y, (1 \dot{-} k) + 2 \cdot g(x, n)) \end{array} \right.$$

for all $y \in Y, x \in X$ and $k, n \in \mathbb{N}$ and define $h : X \times \mathbb{N} \rightrightarrows Y \times \mathbb{N}$ by $h(x,n) := g'(f(x,n),x,n)$ for all $x \in X, n \in \mathbb{N}$. Then

$$0 \in g(x,n) \iff f(x,0),...,f(x,n) \not\subseteq Y \times \{0\}$$

for all (x,n) and

$$\begin{aligned} f_{\min}(x) &= \{y : (\exists n)(y,0) \in f(x,n) \text{ and } f(x,0),...,f(x,n-1) \not\subseteq Y \times \{0\}\} \\ &= \{y : (\exists n)(y,0) \in f(x,n) \text{ and } 0 \in g(x,0),...,g(x,n-1)\} \\ &= \{y : (\exists n)(y,1) \in h(x,n)\} \\ &= (h^{\leftrightarrow})_1(x,1) \end{aligned}$$

for all $x \in \mathrm{dom}(f_{\min})$. $\square$

Consequently, the following diagram illustrates the situation:

$$\boxed{\text{essentials}} + \boxed{\begin{array}{c}\text{primitive}\\\text{recursion}\end{array}} + \boxed{\text{inversion}} \to \boxed{\begin{array}{c}\text{total}\\\text{minimization}\end{array}}$$

2.5.7 Section

In the previous subsection we have seen that inversion can be used to construct μ-recursion (at least for total functions). Now we will show that inversion can be used to construct further interesting and natural operators. The first example is the section operator, which is just a special case of inversion.

Definition 2.5.15 (Section) If $f :\subseteq X \rightrightarrows Y \times \mathbb{N}$ is an operation, then the *section* $f_0 :\subseteq X \rightrightarrows Y$ is defined by

$$f_0(x) := \{y : (y,0) \in f(x)\}$$

for all $x \in \mathrm{dom}(f_0) := \{x : 0 \in f_2(x)\}$.

The next proposition shows that the section operator can be constructed by inversion.

Proposition 2.5.16 (Section) *In natural settings section can be constructed by inversion.*

Proof. Let $f :\subseteq X \rightrightarrows Y \times \mathbb{N}$ be an operation. Define $g :\subseteq X \times \mathbb{N} \rightrightarrows Y \times \mathbb{N}$ by $g := f \circ \mathrm{pr}_1$ and $h :\subseteq X \rightrightarrows Y$ by $h := (g^{\leftrightarrow})_1 \circ (\mathrm{id}_X \times 0)$ for all $x \in \mathbb{N}$. Then

$$h(x) = (g^{\leftrightarrow})_1(x,0) = \{y : (\exists k)(y,0) \in g(x,k)\} = \{y : (y,0) \in f(x)\} = f_0(x)$$

for all $x \in \mathrm{dom}(h) = \{x : (\forall k)(x,k) \in \mathrm{dom}(g) \text{ and } (\exists k) 0 \in g_2(x,k)\} = \{x : 0 \in f_2(x)\} = \mathrm{dom}(f_0)$. $\square$

2.5.8 Union

The union operator is another quite natural operator which can be constructed from inversion.

Definition 2.5.17 (Union) If $f :\subseteq X \times \mathbb{N} \rightrightarrows Y$ is an operation, then the *union* $\cup f :\subseteq X \rightrightarrows Y$ is defined by

$$\cup f(x) := \bigcup_{n=0}^{\infty} f(x,n)$$

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

Occasionally, we will also use the *finite union* $f \cup g :\subseteq X \rightrightarrows Y$ of two operations $f, g :\subseteq X \rightrightarrows Y$, defined by $(f \cup g)(x) := f(x) \cup g(x)$ for all $x \in \mathrm{dom}(f \cup g) := \mathrm{dom}(f) \cap \mathrm{dom}(g)$. The next proposition shows that the union operator can be constructed from inversion.

Proposition 2.5.18 (Union) *In natural settings union can be constructed by inversion.*

Proof. Let $f :\subseteq X \times \mathbb{N} \rightrightarrows Y$ be an operation. Define $g :\subseteq X \times \mathbb{N} \rightrightarrows Y \times \mathbb{N}$ by $g := (f, 0)$ and $h :\subseteq X \rightrightarrows Y$ by $h := (g^{\leftrightarrow})_2 \circ (\mathrm{id}_X \times 0)$ for all $x \in \mathbb{N}$. Then

$$\begin{aligned} h(x) &= (g^{\leftrightarrow})_2(x,0) = \{y : (\exists k)(y,0) \in g(x,k)\} \\ &= \{y : (\exists k) y \in f(x,k)\} = \bigcup_{k=0}^{\infty} f(x,k) = \cup f(x) \end{aligned}$$

for all $x \in \mathrm{dom}(h) = \{x : (\forall k)(x,k) \in \mathrm{dom}(g) \text{ and } (\exists k) 0 \in g_2(x,k)\} = \{x : (\forall k)(x,k) \in \mathrm{dom}(f)\} = \mathrm{dom}(\cup f)$. □

Chapter 3

Recursive and Computable Operations over Structures

In the previous chapter we have studied the closure schemes of our abstract high-level language. In this chapter we will add initial operations to our language which will be provided by structures. The chapter will be divided into two parts: in the first part we will define recursive operations over structures and we will compare them with computable operations. In the second part we will study the special class of *perfect* structures, which have the nice property that recursive and computable operations coincide over them. Moreover, we will investigate recursive sets over structures.

3.1 Recursive and computable operations over natural structures

In this section we will define recursive operations over structures with the help of the recursion operators which have been introduced and investigated in the previous chapter. We will introduce effective and recursive structures and we will show that over effective structures each recursive operation is computable. Vice versa, over recursive structures each computable operation is recursive.

3.1.1 Recursive operations over structures

In the following we want to extend the notion of recursiveness to mathematical structures. We will consider structures $(X, f_1, ..., f_k)$ with a set X and a finite number of operations $f_1, ..., f_k$. If there is a constant $c \in X$ among the operations, then we will consider it in a canonical way as zero-ary constant function $X^0 \to X, () \mapsto c$ with value c (and we assume $X^0 = \{()\}$ for each set

X). Later, we will propose a standard way to consider sets and sequences of sets as parts of structures. For the moment, we will only assume that we can describe theses further objects by operations too. The reader should notice that we allow so-called *prestructures* $(X, f_1, ..., f_k)$ with arbitrary operations $f_i :\subseteq X_i \rightrightarrows Y_i$ in the following definition.

Definition 3.1.1 (Prestructures) $S = (X, f_1, ..., f_k)$ is called a *prestructure* with *universe* X, if X is a set and $f_1, ..., f_k$ are arbitrary operations, called the *initial operations* of S. If $S_1, ..., S_n$ are prestructures, then $S = (S_1, ..., S_n)$ is called a *many-sorted prestructure* and an operation f is called an *initial operation* of S, if it is an initial operation of some S_i, $i = 1, ..., n$. The prestructures $S_1, ..., S_n$ are called *prestructures of* S.

Intuitively, a prestructure is a data-type with certain associated operations. Since these associated operations do not necessarily have the universe of the prestructure as target and source, we allow arbitrary operations for the moment. Nevertheless, we will encapsulate operations in a prestructure which have a semantical coherence.

In the next step we define the notion of a set over a many-sorted prestructure. Roughly speaking, a set over a many-sorted prestructure $S = (S_1, ..., S_n)$ is a set that can be constructed as product or sequence set of the universes of $S_1, ..., S_n$. This definition pays attention to the fact that we want to handle product as well as sequential data-types.

Definition 3.1.2 (Sets and operations over prestructures) Let $S = (S_1, ..., S_n)$ be a many-sorted prestructure and let X_i be the universe of the prestructure S_i for $i = 1, ..., n$. 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, provided that 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 many-sorted prestructures S whose initial operations are operations over S and we will call them *structures*.

Definition 3.1.3 (Structures) A many-sorted prestructure $S = (S_1, ..., S_n)$ is called a *(many-sorted) structure*, if all initial operations of S are operations over S. If X_i is the universe of S_i for $i = 1, ..., n$, then $X = X_1 \times ... \times X_n$ is called the *universe* of 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.

In the following we will say for short *structure* instead of many-sorted structure. If $S = (S_1, ..., S_n)$ is a structure and some (S_i) is a structure too, then we will also say, that S_i is a *substructure* of S and that S_i is a structure itself. For the following we will always assume that the prestructures of a structure are pairwise different. If there are prestructures with the same universe, then we will denote the universes differently such that no confusion should occur. Now we proceed to define recursive operations over structures.

Definition 3.1.4 (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 projection, juxtaposition, product, composition, iteration, inversion, evaluation, transposition, exponentiation and sequentialization.

From the definition of the closure schemes it is obvious that each recursive operation over a structure is an operation over that structure. The previous definition and our convention about constants allow us to define recursive constants in the following way.

Definition 3.1.5 (Recursive constants over structures) 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.

3.1.2 The structure of natural numbers

In this section we want to introduce one of the most important structures of mathematics. Let the *structure of natural numbers* be given by

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

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* $S : \mathbb{N} \to \mathbb{N}, n \mapsto n+1$. This structure and several further structures which we will investigate are listed with precise definitions in the Appendix. Here and in the following we will often use bold letters, as $\mathbf{N}$, to distinguish a structure from its universe. Now we can prove that the classically recursive functions are recursive over $\mathbf{N}$.

Theorem 3.1.6 (Classically recursive functions) *Each classically recursive function* $f :\subseteq \mathbb{N}^k \to \mathbb{N}$ *is recursive over* $\mathbf{N}$.

Proof. The constant 0 and the successor function S belong to the structure $\mathbf{N}$. The projections pr_i^n can be generated by virtue of the identity $\mathrm{id}_{\mathbb{N}}$ (which also belongs to the structure $\mathbf{N}$) and by the operators product and projection. Thus, these operations are recursive over $\mathbf{N}$. By the results of the previous chapter in natural settings the operators substitution, primitive recursion, and total μ-recursion can be constructed by iteration and inversion. By Kleene's Normal Form Theorem 2.1.2 substitution, primitive recursion and total μ-recursion are sufficient to generate all classically recursive functions. Altogether, by structural induction one can show that each classically recursive function $f :\subseteq \mathbb{N}^k \to \mathbb{N}$ is recursive over $\mathbf{N}$. □

An inverse statement of the previous theorem does also hold: each function $f :\subseteq \mathbb{N}^k \to \mathbb{N}$ which is recursive over $\mathbf{N}$ and which has a recursively enumerable domain is also classically recursive. This will follow from the more general Theorem 3.1.25. We could also prove this result directly since the structure $\mathbf{N}$ admits a pairing function which allows to "simulate" tuples of natural numbers by single natural numbers. We will not present this proof, but we will use the pairing mechanism for other purposes. More precisely, we will use *Cantor's pairing function* $\pi : \mathbb{N}^2 \to \mathbb{N}$, defined by $\pi(n,k) := \frac{1}{2}(n+k)(n+k+1)+k$. As usual we will write $\langle n,k\rangle := \pi(n,k)$ and more general

$$\begin{cases} \langle n\rangle & := & \pi^{(1)}(n) := n, \\ \langle n_1,...,n_{i+1}\rangle & := & \pi^{(i+1)}(n_1,...,n_{i+1}) := \langle\langle n_1,...,n_i\rangle, n_{i+1}\rangle. \end{cases}$$

It is well-known that all pairing functions $\pi^{(i)}$ as well as the projections of its inverses $\pi_j^{(i)} := \mathrm{pr}_j(\pi^{(i)})^{-1}$ for $j = 1,...,i$ are classically recursive. We will often write $\pi_j := \pi_j^{(i)}$ for short.

In the following we will often need structures which include the natural numbers and our essentials from the previous chapter. Especially, these structures offer a natural setting and hence these structures will be called *natural*. By $\mathrm{id}_X : X \to X, x \mapsto x$ we denote the *identity* of X.

Definition 3.1.7 (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 $S = (S_1,...,S_n)$ with universe $X = X_1 \times ... \times X_n$ the identity of each set over S is recursive. Especially, the identities $\mathrm{id}_{X_1},...,\mathrm{id}_{X_n}$ are recursive, and each (non-empty) set $X_1,...,X_n$ contains at least one recursive constant.

Now we will investigate functions of type $f :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$. We will say that such a function f is *classically computable*, if it is a *general recursive operator* in the sense of Rogers (cf. [Rog67]), or, equivalently, a *computable operator* in the sense of Weihrauch (cf. [Wei87]). For the definition we will use the *set of finite sequences* or *words* $\mathbb{N}^*$ of natural numbers. For all $v, w \in \mathbb{N}^*$, $p \in \mathbb{N}^{\mathbb{N}}$ we will write $v \sqsubseteq w$, $v \sqsubseteq p$, if v is a *prefix* of w, 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 $w \in \mathbb{N}^*$. A function $\varphi : \mathbb{N}^* \to \mathbb{N}^*$ is called *monotone*, if

$$v \sqsubseteq w \Longrightarrow \varphi(v) \sqsubseteq \varphi(w)$$

holds for all $v, w \in \mathbb{N}^*$. Hence, we can approximate functions $f :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by monotone functions $\varphi : \mathbb{N}^* \to \mathbb{N}^*$.

Definition 3.1.8 (Classically computable operators) A function (or operator) $f :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is called *classically computable*, if there is 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 $p \in \mathbb{N}^{\mathbb{N}}$.

Let $\nu^* : \mathbb{N} \to \mathbb{N}^*$ be a bijective *standard numbering of finite sequences*. Then computability of functions $\varphi : \mathbb{N}^* \to \mathbb{N}^*$ is understood as computability w.r.t. ν^*. Especially, the *length function* $\mathrm{lg} : \mathbb{N}^* \to \mathbb{N}$, which maps each word $w \in \mathbb{N}^*$ to its length, is computable w.r.t. ν^* (cf. [Wei87]). For short, we will write $\overline{w} := (\nu^*)^{-1}(w)$.

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 classically computable functions $f :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ are continuous and their domains are G_δ-sets (cf. [Wei87]). If a subset $A \subseteq \mathbb{N}^{\mathbb{N}}$ is a domain of a classically computable function $f :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, then we will call A a *computable G_δ-set*. Now we can prove that each classically recursive function is recursive over $\mathbf{N}$.

Theorem 3.1.9 (Classically computable operators) *Each classically computable function (or operator) $f :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is recursive over $\mathbf{N}$.*

Proof. Let f be classically computable. Then there is a computable total and monotone function $\varphi : \mathbb{N}^* \to \mathbb{N}^*$ such that φ approximates f. First, we note that the identity $\mathrm{id}_{\mathbb{N}^{\mathbb{N}}} = \mathrm{id}_{\mathbb{N}}^{\mathbb{N}}$ is recursive over $\mathbf{N}$ and so is the evaluation function $(\mathrm{id}_{\mathbb{N}^{\mathbb{N}}})_* : \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{N}$ with $(\mathrm{id}_{\mathbb{N}^{\mathbb{N}}})_*(p, k) := p(k)$ for all p, k. Thus, the functions $r : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ and $s : \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{N}$, defined by

$$r(\overline{a_0 ... a_k}, n) := \begin{cases} a_n & \text{if } n \leq k \\ 0 & \text{else} \end{cases}$$

for all $a_i \in \mathbb{N}, n, k \in \mathbb{N}$ and $s(p,k) := \overline{p(0)...p(k)}$ for all p, k are recursive over $\mathbf{N}$. Finally, $t : \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N} \times \mathbb{N}$, defined by

$$t(p,n,k) := (r(\overline{\varphi\nu^* s(p,k)}, n), (n+1) \dot{-} \lg \varphi\nu^* s(p,k))$$

for all p, n, k is recursive over $\mathbf{N}$ too. Now, $t_{\min}(p,n) = f(p)(n)$ for all $p \in \mathrm{dom}(f)$, $n \in \mathbb{N}$ and $\{p\} \times \mathbb{N} \subseteq \mathrm{dom}(t_{\min})$, if and only if $p \in \mathrm{dom}(f)$. Hence, $f = [t_{\min}]$ is recursive over $\mathbf{N}$ (since total minimization is available in natural settings). $\square$

An inverse statement of the previous theorem does also hold: each function $f :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ which is recursive over $\mathbf{N}$ and which has a computable G_δ-domain is also classically computable. This will follow from the more general Theorem 3.1.25. Again one could prove this result directly with the help of suitable pairing functions which we will not do here. But we will introduce pairing functions of types $\mathbb{N}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{N}^{\mathbb{N}}$, $\mathbb{N}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, $(\mathbb{N}^{\mathbb{N}})^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$. More precisely, we define

$$\begin{aligned}
\langle p, n\rangle(k) &:= \begin{cases} n & \text{if} \quad k = 0 \\ p(k-1) & \text{else} \end{cases}, \\
\langle p, q\rangle(k) &:= \begin{cases} p(n) & \text{if} \quad k = 2n \\ q(n) & \text{if} \quad k = 2n+1, \end{cases} \\
\langle p_0, p_1, ...\rangle\langle n, k\rangle &:= p_n(k)
\end{aligned}$$

for all $p, q, p_i \in \mathbb{N}^{\mathbb{N}}, i, k, n \in \mathbb{N}$. As in case of Cantor's pairing function, and by abuse of notation, we denote the projections of the inverses of these pairing functions by π_i. For instance $\pi_1\langle p, q\rangle = p$ and $\pi_i\langle p_0, p_1, ...\rangle = p_i$. From the context it will always be clear which meaning of π_i is intended. It is well-known that all defined pairing functions, as well as the projections of the inverses are classically computable. In the following we will say for short that f is *computable*, if it is classically computable.

Sometimes it is suitable to consider structures where a special constant operation is available. For each set $A \subseteq X$ we define the *omnipotent operation* of A

$$\Omega_A : \{()\} \rightrightarrows X, () \mapsto A,$$

i.e. Ω_A is the constant operation with yields as result $\Omega_A() = A$. Since $\Omega_{\mathbb{N}} = \bigcup \mathrm{id}_{\mathbb{N}}$ 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.

3.1.3 Theory of effectivity

The aim of this section is to briefly recall some facts from type 2 theory of effectivity which will be used in the following. All definitions in this section essentially are standard definitions and can be found in the references ([KW84, Wei95, Wei97]). Type 2 theory of effectivity studies computability w.r.t. representations of the involved spaces.

Definition 3.1.10 (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*.

Now we can use the classical notion of computability of functions (operators) $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ to lift the notion of computability to represented spaces. Figure 3.1 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 3.1: Computability w.r.t. representations

Definition 3.1.11 (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

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

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.

From given representations of spaces we can easily construct representations of finite and countably infinite product spaces according to the following definition.

Definition 3.1.12 (Product and sequence representation) 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* $\delta_X^\infty :\subseteq \mathbb{N}^{\mathbb{N}} \to X^{\mathbb{N}}$ 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 $[\delta_1, ..., \delta_{n+1}] := [[\delta_1, ..., \delta_n], \delta_{n+1}]$. A suitable tool for the comparison of representations is reducibility.

Definition 3.1.13 (Reducibility) Let δ, δ' be representations.

(1) δ is *reducible* to δ', or $\delta \le \delta'$ for short, if there is a computable operator $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that

$$\delta(p) = \delta' F(p)$$

for all $p \in \mathrm{dom}(\delta)$.

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

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 and sequence 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 sequence $p \in \mathbb{N}^{\mathbb{N}}$ with $p(i) = n$ for all $i \in \mathbb{N}$, i.e. $\delta_{\mathbb{N}}(\hat{n}) = n$. Especially, 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 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.

3.1.4 Computable operations over structures

In this section we will define computable operations over structures. First we have to extend the notion of computability to operations. In general, computable operations $f :\subseteq X \rightrightarrows Y$ are many-valued. Thus, in order to compute f we have to characterize the whole set $f(x)$, effectively in x. One could simply define that f is (δ_X, δ_Y)–computable, if there is a computable operator $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ 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 have chosen a different definition of computability of operations. In our definition the complete image $f(x)$ will be obtained by the help of an additional "oracle" input. This corresponds to the fact that computations of operations can be considered as indeterministic computations. Of course, in contrast to the definition of nondeterministic computations in complexity theory, here we demand that the output is an element of $f(x)$ for each oracle input.

Definition 3.1.14 (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

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

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 *strongly* (δ_X, δ_Y)*–computable operation*, if, additionally, $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \not\subseteq \mathrm{dom}(F)$ for all $p \in \mathrm{dom}(\delta_X) \setminus \mathrm{dom}(f\delta_X)$. We will also say that f is (strongly) (δ_X, δ_Y)–computable *via* F.

Here, $\langle p, \mathbb{N}^{\mathbb{N}}\rangle := \{\langle p, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\}$. We can associate with each representation $\delta_X :\subseteq \mathbb{N}^{\mathbb{N}} \to X$ 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. [Wei87]). Obviously, each representation is equivalent to its cylindrification. With the help of cylindrifications we can express the following: if an operation f is strongly (δ_X, δ_Y)–computable via $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, then

$$f = \delta_Y \circ F \circ (\delta_X^{\mathrm{cyl}})^{-1}.$$

This shows that our final approach to define computability of operations is closely related to the first attempt above. It is easy to see that this modified definition has the property that equivalent representations induce the same kind of (strong) computability of operations. Moreover, a function $f :\subseteq X \to Y$, considered as an operation, is (δ_X, δ_Y)–computable, if and only if it is (δ_X, δ_Y)–computable as a function 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. Nevertheless, our notion of computability of operations is a suitable generalization of the notion of computability of functions. If not stated differently, then "strong computability" will always mean strong computability in the sense of operations.[1]

Now we will define representations of structures which will be used in order to define effective structures.

Definition 3.1.15 (Representations of structures) Let S be a structure with universe $X = X_1 \times ... \times X_n$ 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*.

Now we can also extend the notion of computability to structures.

Definition 3.1.16 (Computability in structures) Let S be a structure with universe $X = X_1 \times ... \times X_n$ and representation $\delta = [\delta_1, ..., \delta_n]$. 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* w.r.t. δ, if it is (strongly) (δ_Y, δ_Z)–computable.

If, for instance, $Y = (X_1 \times X_2)^{\mathbb{N}}$, then the corresponding representation is $\delta_Y = [\delta_1, \delta_2]^\infty$.

3.1.5 Computable operations are recursive

In this section we will prove that over recursive structures each strongly computable operation is also recursive. In order to define recursive structures we will use the notion of a recursive retraction.

Definition 3.1.17 (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. $f \circ f^- = \mathrm{id}_Y$.

[1] For other related notions of computability of operations cf. [BH94, Bra95] and the Appendix.

If f is a recursive retraction, then the restriction of f to range(f^-) is a surjective function. If f is a surjective and recursive function and the inverse operation f^{-1} of f is recursive, then f is a recursive retraction.

A structure will be called strongly recursive, if it admits a representation which is a recursive retraction. Hence, a recursive structure is a structure with a representation which can be "synthesized" as well as "analyzed" within the structure.

Definition 3.1.18 (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 *recursive via* or *strongly recursive via* δ, respectively.

By the following proposition $\delta = [\delta_1, ..., \delta_n]$ is a recursive retraction over a natural structure S, if $\delta_1, ..., \delta_n$ are.

Proposition 3.1.19 (Closure properties of recursive representations) *If the representations $\delta_i :\subseteq \mathbb{N}^{\mathbb{N}} \to X_i$, $i = 1, ..., n$, $\delta :\subseteq \mathbb{N}^{\mathbb{N}} \to X$ are recursive retractions over a natural structure S, then so are $[\delta_1, ..., \delta_n]$ and δ^∞.*

Proof. We just consider the case $n = 2$, the general case can be deduced by induction. Our pairing functions $\pi : \mathbb{N}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, $\pi' : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ as well as the projections of their inverses and the pairing function $\pi'' : (\mathbb{N}^{\mathbb{N}})^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ are recursive over $\mathbf{N}$.

Let δ_1, δ_2 be recursive retractions over S with recursive right inverse operations δ_1^-, δ_2^-, respectively. Then $[\delta_1, \delta_2] = (\delta_1 \times \delta_2) \circ \pi^{-1} = (\delta_1 \circ \pi_1, \delta_2 \circ \pi_2)$ is recursive over S and $[\delta_1, \delta_2]^- := \pi \circ (\delta_1^- \times \delta_2^-)$ is a recursive right inverse of $[\delta_1, \delta_2]$ over S. Thus, $[\delta_1, \delta_2]$ is a recursive retraction over S.

Let δ be a recursive retraction over S with recursive right inverse operation δ^-. Then $f := (\mathrm{id}_{\mathbb{N}^{\mathbb{N}}})_* \circ (\mathrm{id}_{\mathbb{N}^{\mathbb{N}}} \times \pi') : \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ is recursive over $\mathbf{N}$. We obtain $f(p, n, k) = p\langle n, k\rangle$. Hence, $\delta^\infty = \delta^{\mathbb{N}} \circ \pi''^{-1} = [\delta \circ [f]]$ is recursive over S and $(\delta^\infty)^- := \pi'' \circ (\delta^-)^{\mathbb{N}}$ is a recursive right inverse of δ^∞ over S. Thus, δ^∞ is a recursive retraction over S. □

Vice versa, it is easy to see that $\delta_1, ..., \delta_n$ are recursive retractions over a natural structure 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 3.1.20 *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

$$\begin{cases} f(k,0) & := & k \\ f(k,n+1) & := & \Omega_{\mathbb{N}}() \end{cases}$$

Then f is recursive over $\mathbf{N}$ and so is $\delta_{\mathbb{N}}^{-1} = [f]$. □

Since the structure of natural numbers is recursive it makes sense to apply the notion of recursiveness to natural structures at all. The following theorem will be the main result of this section.

Theorem 3.1.21 (Computable 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 w.r.t.* δ *is also recursive over* S.

Proof. Let $X = X_1 \times ... \times X_n$ be the universe of $S = (S_1, ..., S_n)$, 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 Proposition 3.1.19 that δ_X, as well as δ_Y are recursive retractions. From Proposition 3.1.22 below it follows that f is recursive over S. □

We finish the proof of our theorem with the following proposition.

Proposition 3.1.22 *Let* S *be a natural structure, let* $\delta_X :\subseteq \mathbb{N}^{\mathbb{N}} \to X$, $\delta_Y :\subseteq \mathbb{N}^{\mathbb{N}} \to Y$ *be representations, and let* $\delta_X^- : X \rightrightarrows \mathbb{N}^{\mathbb{N}}$ *be a right inverse of* δ_X, *such that* δ_Y *and* δ_X^- *are recursive over* S. *If* $f :\subseteq X \rightrightarrows Y$ *is strongly* (δ_X, δ_Y)*–computable via a computable function* $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, *then*

$$f = \delta_Y \circ F \circ \langle \delta_X^- \times \Omega_{\mathbb{N}^{\mathbb{N}}} \rangle$$

and hence f *is recursive over* S.

Proof. Let f be strongly (δ_X, δ_Y)–computable via a computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ and let $x \in X$. Then $\emptyset \neq \delta_X^-(x) \subseteq \delta_X^{-1}\{x\}$. If $x \in \mathrm{dom}(f)$ then

$$\begin{aligned} \langle \delta_X^- \times \Omega_{\mathbb{N}^{\mathbb{N}}} \rangle(x) &= \{\langle p, q \rangle : p \in \delta_X^-(x), q \in \mathbb{N}^{\mathbb{N}}\} \\ &= \bigcup_{p \in \delta_X^-(x)} \langle p, \mathbb{N}^{\mathbb{N}} \rangle \\ &\subseteq \mathrm{dom}(\delta_Y F), \end{aligned}$$

i.e. $x \in \mathrm{dom}(\delta_Y \circ F \circ \langle \delta_X^- \times \Omega_{\mathbb{N}^\mathbb{N}} \rangle)$ and

$$\begin{aligned}\delta_Y \circ F \circ \langle \delta_X^- \times \Omega_{\mathbb{N}^\mathbb{N}} \rangle(x) &= \bigcup_{p \in \delta_X^-(x)} \{\delta_Y F \langle p, q \rangle : q \in \mathbb{N}^\mathbb{N}\} \\ &= \bigcup_{p \in \delta_X^-(x)} f \delta_X(p) \\ &= f(x).\end{aligned}$$

If $x \notin \mathrm{dom}(f)$ then $\langle \delta_X^- \times \Omega_{\mathbb{N}^\mathbb{N}} \rangle(x) = \bigcup_{p \in \delta_X^-(x)} \langle p, \mathbb{N}^\mathbb{N} \rangle \not\subseteq \mathrm{dom}(F)$ and consequently $x \notin \mathrm{dom}(\delta_Y \circ F \circ \langle \delta_X^- \times \Omega_{\mathbb{N}^\mathbb{N}} \rangle)$. This proves $f = \delta_Y \circ F \circ \langle \delta_X^- \times \Omega_{\mathbb{N}^\mathbb{N}} \rangle$. Since F is computable, it is recursive over $\mathbf{N}$. Finally, $\langle \delta_X^- \times \Omega_{\mathbb{N}^\mathbb{N}} \rangle$ is recursive over S, since the tupling function $\mathbb{N}^\mathbb{N} \times \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$, 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 3.1.23 (Computable 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 w.r.t. δ also admits a recursive extension over S.*

3.1.6 Recursive operations are computable

In this section we will prove that recursive operations are computable w.r.t. effective representations.

Definition 3.1.24 (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 w.r.t. 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 with universe $X = X_1 \times ... \times X_n$, if $X_i = \mathbb{N}$ implies $\delta_i \equiv \delta_\mathbb{N}$. Now we can formulate the main result of this section.

Theorem 3.1.25 (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 w.r.t. δ.*

Proof. The proof can be obtained by structural induction. The initial operations of S are (strongly) computable w.r.t. δ by assumption. We have to show that the recursion operators projection, juxtaposition, product, composition, iteration, inversion, evaluation, transposition, exponentiation and sequentialization transfer operations which are (strongly) computable w.r.t. δ to operations which are (strongly) computable w.r.t. δ. This is done by the following proposition which finishes the proof. $\square$

The proof of the following closure properties of computable operations is based on corresponding closure properties of the computable functions (operators) $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$.

Proposition 3.1.26 (Closure properties of computable operations) *Let (U, δ_U), (V, δ_V), (X, δ_X), (Y, δ_Y), (Z, δ_Z) be represented spaces.*

(1) If $f :\subseteq X \rightrightarrows Y \times Z$ is $(\delta_X, [\delta_Y, \delta_Z])$–computable, then $f_1 :\subseteq X \rightrightarrows Y$ is (δ_X, δ_Y)–computable, and $f_2 :\subseteq X \rightrightarrows Z$ is (δ_X, δ_Z)–computable.

(2) If $f :\subseteq X \rightrightarrows Y$, $g :\subseteq X \rightrightarrows Z$ are (δ_X, δ_Y)–, (δ_X, δ_Z)–computable, respectively, then $(f, g) :\subseteq X \rightrightarrows Y \times Z$ is $(\delta_X, [\delta_Y, \delta_Z])$–computable.

(3) If $f :\subseteq X \rightrightarrows Y$, $g :\subseteq U \rightrightarrows V$ are (δ_X, δ_Y)–, (δ_U, δ_V)–computable, respectively, then $f \times g :\subseteq X \times U \rightrightarrows Y \times V$ is $([\delta_X, \delta_U], [\delta_Y, \delta_V])$–computable.

(4) If $f :\subseteq X \rightrightarrows Y$, $g :\subseteq Y \rightrightarrows Z$ are (δ_X, δ_Y)–, (δ_Y, δ_Z)–computable, respectively, then $g \circ f :\subseteq X \rightrightarrows Z$ is (δ_X, δ_Z)–computable.

(5) If $f :\subseteq X \rightrightarrows X$ is (δ_X, δ_X)–computable, then $f^* :\subseteq X \times \mathbb{N} \rightrightarrows X$ is $([\delta_X, \delta_{\mathbb{N}}], \delta_X)$–computable.

(6) If $f :\subseteq X \times \mathbb{N} \rightrightarrows Y \times \mathbb{N}$ is $([\delta_X, \delta_{\mathbb{N}}], [\delta_Y, \delta_{\mathbb{N}}])$–computable, then $f^{\leftrightarrow} :\subseteq X \times \mathbb{N} \rightrightarrows Y \times \mathbb{N}$ is $([\delta_X, \delta_{\mathbb{N}}], [\delta_Y, \delta_{\mathbb{N}}])$–computable.

(7) If $f :\subseteq X \rightrightarrows Y^{\mathbb{N}}$ is $(\delta_X, \delta_Y^{\infty})$–computable, then $f_* :\subseteq X \times \mathbb{N} \rightrightarrows Y$ is $([\delta_X, \delta_{\mathbb{N}}], \delta_Y)$–computable.

(8) If $f :\subseteq X \times \mathbb{N} \rightrightarrows Y$ is $([\delta_X, \delta_{\mathbb{N}}], \delta_Y)$–computable, then $[f] :\subseteq X \rightrightarrows Y^{\mathbb{N}}$ is $(\delta_X, \delta_Y^{\infty})$–computable.

(9) If $f :\subseteq X \rightrightarrows Y$ is (δ_X, δ_Y)–computable, then $f^{\mathbb{N}} :\subseteq X^{\mathbb{N}} \rightrightarrows Y^{\mathbb{N}}$ is $(\delta_X^{\infty}, \delta_Y^{\infty})$–computable.

(10) If $f :\subseteq X \rightrightarrows \mathbb{N}$ is $(\delta_X, \delta_{\mathbb{N}})$–computable, then $f^{\triangle} :\subseteq X \rightrightarrows \mathbb{N}^{\mathbb{N}}$ is $(\delta_X, \delta_{\mathbb{N}}^{\infty})$–computable.

Corresponding properties hold for strong computability.

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 we claim that $f_1 :\subseteq X \rightrightarrows Y$ is (strongly) (δ_X, δ_Y)–computable via G. Let $p \in \mathrm{dom}(\delta_X)$ and $x := \delta_X(p)$. We note $\mathrm{dom}(f_1) = \mathrm{dom}(f)$ and $\mathrm{dom}(G) = \mathrm{dom}(F)$. Then $x \in \mathrm{dom}(f_1)$ implies $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}(\delta_Y G)$ and if f is even strongly computable via F, then $x \notin \mathrm{dom}(f_1)$ implies $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \not\subseteq \mathrm{dom}(G)$. Moreover, if $x \in \mathrm{dom}(f_1)$, then

$$\begin{aligned} \{\delta_Y G\langle p, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\} &= \{\delta_Y \pi_1 F\langle p, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\} \\ &= \{\mathrm{pr}_1[\delta_Y, \delta_Z] F\langle p, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\} \\ &= \mathrm{pr}_1 f \delta_X(p) \\ &= f_1(x). \end{aligned}$$

Thus f_1 is (strongly) (δ_X, δ_Y)–computable via G. The proof for f_2 proceeds analogously.

(2) Let $f :\subseteq X \rightrightarrows Y$, $g :\subseteq X \rightrightarrows Z$ be (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 we claim that $(f, g) :\subseteq X \rightrightarrows Y \times Z$ is $(\delta_X, [\delta_Y, \delta_Z])$–computable via H. For the proof, let $p \in \mathrm{dom}(\delta_X)$ and $x := \delta_X(p)$. If $x \in \mathrm{dom}(f, g)$ then $x \in \mathrm{dom}(f) \cap \mathrm{dom}(g)$. Consequently, $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}(\delta_Y F) \cap \mathrm{dom}(\delta_Z G)$. Thus, $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}([\delta_Y, \delta_Z] H)$ and

$$\begin{aligned} & \{[\delta_Y, \delta_Z] H\langle p, \langle q, r\rangle\rangle : \langle q, r\rangle \in \mathbb{N}^{\mathbb{N}}\} \\ = \; & \{(\delta_Y F\langle p, q\rangle, \delta_Z G\langle p, r\rangle) : \langle q, r\rangle \in \mathbb{N}^{\mathbb{N}}\} \\ = \; & \{\delta_Y F\langle p, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\} \times \{\delta_Z G\langle p, r\rangle : r \in \mathbb{N}^{\mathbb{N}}\} \\ = \; & f\delta_X(p) \times g\delta_X(p) \\ = \; & (f, g)(x). \end{aligned}$$

Thus, (f, g) is $(\delta_X, [\delta_Y, \delta_Z])$–computable via H.

Now let f, g be even strongly (δ_X, δ_Y)–, (δ_X, δ_Z)–computable via F, G, respectively and let $x \notin \mathrm{dom}(f, g)$. Then $x \notin \mathrm{dom}(f)$ or $x \notin \mathrm{dom}(g)$. In the first case $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \not\subseteq \mathrm{dom}(F)$ and in the second case $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \not\subseteq \mathrm{dom}(G)$. In both cases $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \not\subseteq \mathrm{dom}(H)$. Hence, (f, g) is even strongly $(\delta_X, [\delta_Y, \delta_Z])$–computable via H.

(3) Let $f :\subseteq X \rightrightarrows Y$, $g :\subseteq U \rightrightarrows V$ be (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 we claim that $f \times g :\subseteq X \times U \rightrightarrows Y \times V$ is $([\delta_X, \delta_U], [\delta_Y, \delta_V])$–computable via H. For the proof, let $\langle p, p'\rangle \in \mathrm{dom}([\delta_X, \delta_U])$ and $x := \delta_X(p)$, $u := \delta_U(p')$. If $(x, u) \in \mathrm{dom}(f \times g)$ then $x \in \mathrm{dom}(f)$ and $u \in \mathrm{dom}(g)$. Consequently, $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}(\delta_Y F)$ and $\langle p', \mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}(\delta_V G)$. Thus $\langle\langle p, p'\rangle, \mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}([\delta_Y, \delta_V]H)$ and

$$\begin{aligned}
& \{[\delta_Y, \delta_V]H\langle\langle p, p'\rangle, \langle q, r\rangle : \langle q, r\rangle \in \mathbb{N}^{\mathbb{N}}\} \\
= \ & \{(\delta_Y F\langle p, q\rangle, \delta_V G\langle p', r\rangle) : \langle q, r\rangle \in \mathbb{N}^{\mathbb{N}}\} \\
= \ & \{\delta_Y F\langle p, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\} \times \{\delta_V G\langle p', r\rangle : r \in \mathbb{N}^{\mathbb{N}}\} \\
= \ & f\delta_X(p) \times g\delta_U(p') \\
= \ & (f \times g)(x, u).
\end{aligned}$$

Thus $f \times g$ is $([\delta_X, \delta_U], [\delta_Y, \delta_V])$–computable via H.

Now let $f \times g$ be even strongly (δ_X, δ_Y)–, (δ_U, δ_V)–computable via F, G, respectively and let $(x, u) \notin \mathrm{dom}(f \times g)$. Then $x \notin \mathrm{dom}(f)$ or $u \notin \mathrm{dom}(g)$. In the first case $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \not\subseteq \mathrm{dom}(F)$ and in the second case $\langle p', \mathbb{N}^{\mathbb{N}}\rangle \not\subseteq \mathrm{dom}(G)$. In both cases $\langle\langle p, p'\rangle, \mathbb{N}^{\mathbb{N}}\rangle \not\subseteq \mathrm{dom}(H)$. Hence, $(f \times g)$ is even 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 (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 we claim that $g \circ f :\subseteq X \rightrightarrows Z$ is (δ_X, δ_Z)–computable via H. For the proof, let $p \in \mathrm{dom}((g \circ f)\delta_X)$, $x := \delta_X(p)$. Then $x \in \mathrm{dom}(f)$ and $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}(\delta_Y F)$ and $f(x) \subseteq \mathrm{dom}(g)$ such that $\langle F\langle p, \mathbb{N}^{\mathbb{N}}\rangle, \mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}(\delta_Z G)$. Consequently, $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}(\delta_Z H)$ and

$$(g \circ f)\delta_X(p) \quad = \quad g(f\delta_X(p))$$

$$
\begin{aligned}
&= g\{\delta_Y F\langle p,q\rangle : q\in\mathbb{N}^{\mathbb{N}}\}\\
&= \{\delta_Z G\langle F\langle p,q\rangle, r\rangle : q,r\in\mathbb{N}^{\mathbb{N}}\}\\
&= \{\delta_Z H\langle p,\langle q,r\rangle\rangle : \langle q,r\rangle\in\mathbb{N}^{\mathbb{N}}\}
\end{aligned}
$$

Thus, $g\circ f$ is (δ_X,δ_Z)–computable via H.

Now, let f,g be even strongly (δ_X,δ_Y)–, (δ_Y,δ_Z)–computable via F,G, respectively and let $p\in\mathrm{dom}(\delta_X)\setminus\mathrm{dom}((g\circ f)\delta_X)$, $x:=\delta_X(p)$. Then either $x\notin\mathrm{dom}(f)$ or $f(x)\not\subseteq\mathrm{dom}(g)$. In the first case, $\langle p,\mathbb{N}^{\mathbb{N}}\rangle\not\subseteq\mathrm{dom}(F)$. In the second case there is some $q\in\mathbb{N}^{\mathbb{N}}$ such that $F\langle p,q\rangle\notin\mathrm{dom}(g\delta_Y)$ and hence $\langle F\langle p,q\rangle,\mathbb{N}^{\mathbb{N}}\rangle\not\subseteq\mathrm{dom}(G)$. In both cases $\langle p,\mathbb{N}^{\mathbb{N}}\rangle\not\subseteq\mathrm{dom}(H)$. Thus $g\circ f$ 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 $G:\subseteq\mathbb{N}^{\mathbb{N}}\to\mathbb{N}^{\mathbb{N}}$ inductively 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 we claim that $f^*:\subseteq X\times\mathbb{N}\to X$ is (strongly) $([\delta_X,\delta_{\mathbb{N}}],\delta_X)$–computable via G. By an easy induction one can prove

$$
G\langle\langle p,p'\rangle,\langle q_0,q_1,...\rangle\rangle \;=\; G\langle\langle p,p'\rangle,\langle q_0,...,q_{n-1},\hat{0},\hat{0},...\rangle\rangle \qquad (3.1)
$$

for all $p,p',q_i\in\mathbb{N}^{\mathbb{N}}, i\in\mathbb{N}$ and $n=p'(0)=\delta_{\mathbb{N}}(p)$. We will prove by induction on n

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

and $\langle\langle p,p'\rangle,\mathbb{N}^{\mathbb{N}}\rangle\subseteq\mathrm{dom}(\delta_X G)$ for all $\langle p,p'\rangle\in\mathrm{dom}(f^*[\delta_X,\delta_{\mathbb{N}}])$ with $x=\delta_X(p)$ and $n=p'(0)=\delta_{\mathbb{N}}(p')$.

(In case that f is strongly (δ_X,δ_X)–computable via F, we, additionally, show $\langle\langle p,p'\rangle,\mathbb{N}^{\mathbb{N}}\rangle\not\subseteq\mathrm{dom}(G)$ for all $\langle p,p'\rangle\in\mathrm{dom}[\delta_X,\delta_{\mathbb{N}}]\setminus\mathrm{dom}(f^*[\delta_X,\delta_{\mathbb{N}}])$ with $n=p'(0)=\delta_{\mathbb{N}}(p')$.)
Now, let $\langle p,p'\rangle\in\mathrm{dom}(f^*[\delta_X,\delta_{\mathbb{N}}])$ and $x:=\delta_X(p)$, $n:=p'(0)=\delta_X(p')$.
$\underline{n=0}$ Then $(x,0)\in\mathrm{dom}(f^*)$, $\langle\langle p,p'\rangle,\mathbb{N}^{\mathbb{N}}\rangle\subseteq\mathrm{dom}(\delta_X G)$ and

$$
\{\delta_X G\langle\langle p,p'\rangle,q\rangle : q\in\mathbb{N}^{\mathbb{N}}\}=\{\delta_X(p)\}=\{x\}=f^*(x,0).
$$

$\underline{n\to n+1}$ Let $(x,n+1)\in\mathrm{dom}(f^*)$, then $f^*(x,n)\subseteq\mathrm{dom}(f)$. By induction hypothesis $\langle\langle p,p''\rangle,\mathbb{N}^{\mathbb{N}}\rangle\subseteq\mathrm{dom}(\delta_X G)$ for all $p''\in\mathbb{N}^{\mathbb{N}}$ with $\delta_{\mathbb{N}}(p'')=p''(0)=n$ and

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

Especially, $\langle G\langle\langle p,\hat{n}\rangle,\mathbb{N}^{\mathbb{N}}\rangle,\mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}(\delta_X F)$ and $\langle\langle p,\hat{n}\rangle,\mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}(\delta_X G)$ and

$$\begin{array}{rl} & \{\delta_X G\langle\langle p,p'\rangle,q\rangle : q\in\mathbb{N}^{\mathbb{N}}\} \\ = & \{\delta_X G\langle\langle p,p'\rangle,\langle q_0,...,q_n,\hat{0},\hat{0},...\rangle\rangle : q_0,...,q_n\in\mathbb{N}^{\mathbb{N}}\} \text{ by Equation 3.1} \\ = & \{\delta_X F\langle G\langle\langle p,\hat{n}\rangle,\langle q_0,...,q_n,\hat{0},\hat{0},...\rangle\rangle,q_n\rangle : q_0,...,q_n\in\mathbb{N}^{\mathbb{N}}\} \\ = & \{\delta_X F\langle G\langle\langle p,\hat{n}\rangle,q\rangle,q'\rangle : q,q'\in\mathbb{N}^{\mathbb{N}}\} \text{ by Equation 3.1} \\ = & f\{\delta_X G\langle\langle p,\hat{n}\rangle,q\rangle : q\in\mathbb{N}^{\mathbb{N}}\} \\ = & f\circ f^*(x,n) \\ = & f^*(x,n+1). \end{array}$$

(In case that f is strongly (δ_X,δ_X)–computable via F and $(x,n+1)\notin \mathrm{dom}(f^*)$, then either $f^*(x,n)\not\subseteq \mathrm{dom}(f)$ or $(x,n)\notin\mathrm{dom}(f^*)$. In the first case there is some $q\in\mathbb{N}^{\mathbb{N}}$ such that $\delta_X G\langle\langle p,\hat{n}\rangle,q\rangle\notin\mathrm{dom}(f)$, consequently $\langle G\langle\langle p,\hat{n}\rangle,q\rangle,\mathbb{N}^{\mathbb{N}}\rangle\not\subseteq\mathrm{dom}(F)$. In the second case by induction hypothesis $\langle\langle p,\hat{n}\rangle,\mathbb{N}^{\mathbb{N}}\rangle\not\subseteq\mathrm{dom}(G)$. In both cases $\langle\langle p,p'\rangle,\mathbb{N}^{\mathbb{N}}\rangle\not\subseteq\mathrm{dom}(G)$ follows.)

(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 $F:\subseteq\mathbb{N}^{\mathbb{N}}\to\mathbb{N}^{\mathbb{N}}$, i.e.

$$f[\delta_X,\delta_{\mathbb{N}}](p)=\{[\delta_Y,\delta_{\mathbb{N}}]F\langle p,q\rangle : q\in\mathbb{N}^{\mathbb{N}}\}$$

and $\langle p,\mathbb{N}^{\mathbb{N}}\rangle\subseteq\mathrm{dom}([\delta_Y,\delta_{\mathbb{N}}]F)$ for all $p\in\mathrm{dom}(f[\delta_X,\delta_{\mathbb{N}}])$. 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

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

for all p,n,m, is computable. Define $G:\subseteq\mathbb{N}^{\mathbb{N}}\to\mathbb{N}^{\mathbb{N}}$ by

$$G\langle\langle p,p'\rangle,mq\rangle:=\langle\pi_1F\langle\langle p,\hat{k}\rangle,wq\rangle,\hat{k}\rangle$$

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. We claim that $f^{\leftrightarrow}$ is computable via G, i.e.

$$f^{\leftrightarrow}[\delta_X, \delta_{\mathbb{N}}](p) = \{[\delta_Y, \delta_{\mathbb{N}}]G\langle p, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\} \qquad (*)$$

and $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}([\delta_Y, \delta_{\mathbb{N}}]G)$ for all $p \in \mathrm{dom}(f^{\leftrightarrow}[\delta_X, \delta_{\mathbb{N}}])$. For the proof, let $\langle p, p'\rangle \in \mathrm{dom}(f^{\leftrightarrow}[\delta_X, \delta_{\mathbb{N}}])$, $x := \delta_X(p)$, $n := p'(0) = \delta_{\mathbb{N}}(p')$.

First we prove $\langle\langle p, p'\rangle, \mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}([\delta_Y, \delta_{\mathbb{N}}]G)$. Since $\langle p, p'\rangle \in \mathrm{dom}(f^{\leftrightarrow}[\delta_X, \delta_{\mathbb{N}}])$, it follows that there is a $k \in \mathbb{N}$ such that $n \in f_2(x, k)$ and $\{x\} \times \mathbb{N} \subseteq \mathrm{dom}(f)$. Thus $\langle\langle p, \mathbb{N}^{\mathbb{N}}\rangle, \mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}([\delta_Y, \delta_{\mathbb{N}}]F)$ and there are infinitely many $i \in \mathbb{N}$ such that $(p, n, k, i) \in P$ since F is continuous and φ is monotone and approximates F. Consequently, $h(p, n, m)$ is defined for all m. Altogether, this proves $\langle\langle p, p'\rangle, \mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}([\delta_Y, \delta_{\mathbb{N}}]G)$. Now we proceed to prove $(*)$.

"$\subseteq$" Let $(y, k) \in f^{\leftrightarrow}(x, n)$. Then $(y, n) \in f(x, k)$. By assumption there is a $q' \in \mathbb{N}^{\mathbb{N}}$ such that $[\delta_Y, \delta_{\mathbb{N}}]F\langle\langle p, \hat{k}\rangle, q'\rangle = (y, n)$. Then there is an $i \in \mathbb{N}$ such that $(p, n, k, i) \in P$ and $\nu^*(i) \sqsubseteq \langle\langle p, \hat{k}\rangle, q'\rangle$. Consequently, there is an $m \in \mathbb{N}$ such that $h(p, n, m) = \langle k, i\rangle$ and there is a $q \in \mathbb{N}^{\mathbb{N}}$ such that $\pi_2'\nu^*(i)q = q'$. Hence,

$$[\delta_Y, \delta_{\mathbb{N}}]G\langle\langle p, p'\rangle, mq\rangle = [\delta_Y, \delta_{\mathbb{N}}]\langle \pi_1 F\langle\langle p, \hat{k}\rangle, q'\rangle, \hat{k}\rangle = (y, k)$$

"$\supseteq$" Now let $q \in \mathbb{N}^{\mathbb{N}}$, $m \in \mathbb{N}$, and

$$(y, k) := [\delta_Y, \delta_{\mathbb{N}}]G\langle\langle p, p'\rangle, mq\rangle,$$

let $i := \pi_2 h(p, n, m)$, and $w := \pi_2'\nu^*(i)$. Then $y = \delta_Y \pi_1 F\langle\langle p, \hat{k}\rangle, wq\rangle$ and $k = \pi_1 h(p, n, m)$. Hence, $(p, n, k, i) \in P$. Thus

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

Since φ approximates F, $\varphi\nu^*(i) \sqsubseteq F\langle\langle p, \hat{k}\rangle, wq\rangle$ follows. Consequently, $\delta_{\mathbb{N}}\pi_2 F\langle\langle p, \hat{k}\rangle, wq\rangle = n$ and

$$(y, n) = [\delta_Y, \delta_{\mathbb{N}}]F\langle\langle p, \hat{k}\rangle, wq\rangle \in f(x, k),$$

i.e. $(y, k) \in f^{\leftrightarrow}(x, n)$.

Finally, let f be even strongly $([\delta_X, \delta_{\mathbb{N}}], [\delta_Y, \delta_{\mathbb{N}}])$–computable via F. We will show that $f^{\leftrightarrow}$ is strongly $([\delta_X, \delta_{\mathbb{N}}], [\delta_Y, \delta_{\mathbb{N}}])$–computable via G', where $G' :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is a restriction of G, defined by

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

for all $p, p', q, q' \in \mathbb{N}^\mathbb{N}$. Then G' is computable.

Since $\langle p, p'\rangle \in \mathrm{dom}(f^{\leftrightarrow}[\delta_X, \delta_\mathbb{N}])$ implies $\langle\langle p, \mathbb{N}^\mathbb{N}\rangle, \mathbb{N}^\mathbb{N}\rangle \subseteq \mathrm{dom}([\delta_Y, \delta_\mathbb{N}]F)$, it follows that $\langle p, \mathbb{N}^\mathbb{N}\rangle \subseteq \mathrm{dom}([\delta_Y, \delta_\mathbb{N}]G')$. Thus, since G' is a restriction of G, $f^{\leftrightarrow}$ is $([\delta_X, \delta_\mathbb{N}], [\delta_Y, \delta_\mathbb{N}])$–computable via G' too.

Now let $\langle p, p'\rangle \in \mathrm{dom}([\delta_X, \delta_\mathbb{N}]) \setminus \mathrm{dom}(f^{\leftrightarrow}[\delta_X, \delta_\mathbb{N}])$, $x := \delta_X(p)$, $n := p'(0) = \delta_\mathbb{N}(p')$. Then $(x, k) \in \mathrm{dom}(f)$ and $n \notin f_2(x, k)$ for all k or there is a k such that $(x, k) \notin \mathrm{dom}(f)$. In the first case there is no (k, i) such that $(p, n, k, i) \in P$. Hence $(p, n, m) \notin \mathrm{dom}(h)$ for all $m \in \mathbb{N}$ and $\langle\langle p, p'\rangle, q\rangle \notin \mathrm{dom}(G) \supseteq \mathrm{dom}(G')$ for all q. In the second case there is some $q \in \mathbb{N}^\mathbb{N}$ such that $\langle\langle p, \hat{k}\rangle, q\rangle \notin \mathrm{dom}(F)$. Consequently, $\langle\langle p, p'\rangle, \langle \hat{k}, q\rangle\rangle \notin \mathrm{dom}(G')$. Altogether, $\langle\langle p, p'\rangle, \mathbb{N}^\mathbb{N}\rangle \not\subseteq \mathrm{dom}(G')$.

(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 $G :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ 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 $\eta : \mathbb{N}^\mathbb{N} \times \mathbb{N} \to \mathbb{N}^\mathbb{N}$, is the computable evaluation function defined by $\eta(\langle p_0, p_1, ...\rangle, n) := p_n$. Then G is computable and we claim that $f_* :\subseteq X \times \mathbb{N} \rightrightarrows Y$ is (strongly) $([\delta_X, \delta_\mathbb{N}], \delta_Y)$–computable via G. Let $\langle p, p'\rangle \in \mathrm{dom}[\delta_X, \delta_\mathbb{N}]$ and $x := \delta_X(p)$, $n := p'(0) = \delta_\mathbb{N}(p)$. If $(x, n) \in \mathrm{dom}(f_*)$, then $x \in \mathrm{dom}(f)$, $\langle p, \mathbb{N}^\mathbb{N}\rangle \subseteq \mathrm{dom}(\delta_Y^\infty F)$ and $\langle\langle p, p'\rangle, \mathbb{N}^\mathbb{N}\rangle \subseteq \mathrm{dom}(\delta_Y G)$ and

$$\begin{aligned}
\{\delta_Y G\langle\langle p, p'\rangle, q\rangle : q \in \mathbb{N}^\mathbb{N}\} &= \{\delta_Y \eta(F\langle p, q\rangle, n) : q \in \mathbb{N}^\mathbb{N}\} \\
&= \{\delta_Y^\infty(F\langle p, q\rangle)(n) : q \in \mathbb{N}^\mathbb{N}\} \\
&= f_*(x, n).
\end{aligned}$$

Thus, f_* is $([\delta_X, \delta_\mathbb{N}], \delta_Y)$–computable via G.

If f is even strongly computable via F, then $(x, n) \notin \mathrm{dom}(f_*)$ implies $x \notin \mathrm{dom}(f)$, $\langle p, \mathbb{N}^\mathbb{N}\rangle \not\subseteq \mathrm{dom}(F)$ and thus $\langle\langle p, p'\rangle, \mathbb{N}^\mathbb{N}\rangle \not\subseteq \mathrm{dom}(G)$. Hence, f_* is even strongly $([\delta_X, \delta_\mathbb{N}], \delta_Y)$–computable via G. Thus f_* is strongly $([\delta_X, \delta_\mathbb{N}], \delta_Y)$–computable via G.

(8) Let $f :\subseteq X \times \mathbb{N} \rightrightarrows Y$ be $([\delta_X, \delta_\mathbb{N}], \delta_Y)$–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 we claim that $[f] :\subseteq X \rightrightarrows Y^\mathbb{N}$ is $(\delta_X, \delta_Y^\infty)$–computable via G. For the proof, let $p \in$

$\mathrm{dom}(\delta_X)$, $x := \delta_X(p)$. If $x \in \mathrm{dom}[f]$ then $\{x\} \times \mathbb{N} \subseteq \mathrm{dom}(f)$ and $\langle\langle p, \mathbb{N}^\mathbb{N}\rangle, \mathbb{N}^\mathbb{N}\rangle \subseteq \mathrm{dom}(\delta_Y F)$, and thus $\langle p, \mathbb{N}^\mathbb{N}\rangle \subseteq \mathrm{dom}(\delta_Y^\infty G)$ and

$$\begin{aligned} & \{\delta_Y^\infty G\langle p, \langle q_0, q_1, ...\rangle\rangle : \langle q_0, q_1, ...\rangle \in \mathbb{N}^\mathbb{N}\} \\ = \ & \{(\delta_Y F\langle\langle p, \hat{n}\rangle, q_n\rangle)_{n\in\mathbb{N}} : q_0, q_1, ... \in \mathbb{N}^\mathbb{N}\} \\ = \ & \{(y_n)_{n\in\mathbb{N}} : (\forall n) y_n \in f(x, n)\} \\ = \ & [f](x). \end{aligned}$$

Thus, $[f]$ is $(\delta_X, \delta_Y^\infty)$–computable via G.

Now let f be even strongly $([\delta_X, \delta_\mathbb{N}], \delta_Y)$–computable via F and let $x \notin \mathrm{dom}[f]$. Then $\{x\} \times \mathbb{N} \not\subseteq \mathrm{dom}(f)$, thus $\langle\langle p, \mathbb{N}^\mathbb{N}\rangle, \mathbb{N}^\mathbb{N}\rangle \not\subseteq \mathrm{dom}(F)$ and $\langle p, \mathbb{N}^\mathbb{N}\rangle \not\subseteq \mathrm{dom}(G)$. Hence, $[f]$ is even strongly $(\delta_X, \delta_Y^\infty)$–computable via G.

(9) Let $f :\subseteq X \rightrightarrows Y$ be (δ_X, δ_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

$$G\langle\langle p_0, p_1, ...\rangle, \langle q_0, q_1, ...\rangle\rangle := \langle F\langle p_0, q_0\rangle, F\langle p_1, q_1\rangle, ...\rangle$$

for all $p_i, q_i \in \mathbb{N}^\mathbb{N}$, $i \in \mathbb{N}$. Then G is computable and we claim that $f^\mathbb{N} :\subseteq X^\mathbb{N} \rightrightarrows Y^\mathbb{N}$ is $(\delta_X^\infty, \delta_Y^\infty)$–computable via G. For the proof, let $\langle p_0, p_1, ...\rangle \in \mathrm{dom}(\delta_X^\infty)$, $x := (x_n)_{n\in\mathbb{N}} := \delta_X^\infty(p)$. If $x \in \mathrm{dom}(f^\mathbb{N})$ then $x_n \in \mathrm{dom}(f)$ for all $n \in \mathbb{N}$ and $\langle p_n, \mathbb{N}^\mathbb{N}\rangle \subseteq \mathrm{dom}(\delta_Y F)$, and thus $\langle\langle p_0, p_1, ...\rangle, \mathbb{N}^\mathbb{N}\rangle \subseteq \mathrm{dom}(\delta_Y^\infty G)$ and

$$\begin{aligned} & \{\delta_Y^\infty G\langle\langle p_0, p_1, ...\rangle, \langle q_0, q_1, ...\rangle\rangle : \langle q_0, q_1, ...\rangle \in \mathbb{N}^\mathbb{N}\} \\ = \ & \{(\delta_Y F\langle p_n, q_n\rangle)_{n\in\mathbb{N}} : q_0, q_1, ... \in \mathbb{N}^\mathbb{N}\} \\ = \ & \{(y_n)_{n\in\mathbb{N}} : (\forall n) y_n \in f(x, n)\} \\ = \ & f^\mathbb{N}(x). \end{aligned}$$

Thus, $f^\mathbb{N}$ is $(\delta_X^\infty, \delta_Y^\infty)$–computable via G.

Now let f be even strongly (δ_X, δ_Y)–computable via F and let $x \notin \mathrm{dom}(f^\mathbb{N})$. Then there is some $n \in \mathbb{N}$ such that $x_n \notin \mathrm{dom}(f)$, thus $\langle p_n, \mathbb{N}^\mathbb{N}\rangle \not\subseteq \mathrm{dom}(F)$ and $\langle\langle p_0, p_1, ...\rangle, \mathbb{N}^\mathbb{N}\rangle \not\subseteq \mathrm{dom}(G)$. Hence, $f^\mathbb{N}$ is even 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 $r_n := \nu^*(n)\hat{0}$. 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

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

for all $p, q \in \mathbb{N}^{\mathbb{N}}$ and $n, k \in \mathbb{N}$, where H is the function which exists due to the following Lemma 3.1.27. 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.

Now let f be even strongly $(\delta_X, \delta_{\mathbb{N}})$–computable via F and let $p \in \mathrm{dom}(\delta_X) \setminus \mathrm{dom}(f^{\triangle}\delta_X)$. Since $\mathrm{dom}(f^{\triangle}) = \mathrm{dom}(f)$ it follows $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \not\subseteq \mathrm{dom}(F) \supseteq \mathrm{dom}(G)$. Hence, $f^{\triangle}$ is even strongly $(\delta_X, \delta_{\mathbb{N}}^{\infty})$–computable via G.

□

Now we formulate and proof a lemma which finishes the proof of (10) and which roughly states that there exists a computable function which can compute each enumeration of a set from a given one with the help of an "oracle".

Lemma 3.1.27 *There exists a 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 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. 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 $H\langle p, q \rangle = r$.

□

We close this section with a corollary which transfers the previous proposition to the additional operators which have been defined in the previous chapter.

Corollary 3.1.28 *Let S be a natural structure which is (strongly) effective via a natural representation δ. Then the operators substitution, primitive recursion, simultaneous recursion, definition-by-cases, course-of-value recursion, total μ-recursion, total minimization, section and union do preserve (strong) computability w.r.t. δ.*

3.2 Perfect structures and sets over structures

In the previous part of this chapter we have compared computable and recursive operations over structures. Now we want to investigate special structures and sets over structures. We start with the investigation of the class of *perfect structures* over which recursive and (strongly) computable operations coincide.

3.2.1 Perfect structures

In previous sections 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 3.2.1 (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 $\delta \equiv \delta'$.*

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, the structure S is *effectively categorical*, a notion that has been introduced in a slightly different setting by Hertling ([Her99b, Her99a]). 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, S 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. In our setting we have fixed effectivity on the natural numbers.

Now we will prove a proposition which implies the Stability Theorem.

Proposition 3.2.2 *Let S be a natural structure with a representation δ and a natural representation δ'. Let δ_+ be an extension of δ and let δ^- be a right inverse operation of δ. Then $\delta_- := (\delta^-)^{-1}$ is a representation of S too and the following holds:*

(1) $\delta \leq \delta_+$,

(2) if δ_+ is recursive over S and S is effective via δ', then $\delta_+ \leq \delta'$.

(3) if δ^- is recursive over S and S is effective via δ', then $\delta' \leq \delta_-$.

(4) $\delta_- \leq \delta$.

Thus, if δ_+, δ^- are recursive over S and S is effective via δ', then we obtain $\delta \equiv \delta_+ \equiv \delta' \equiv \delta_-$.

Proof. W.l.o.g. 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 $\delta_1'^\infty \equiv \delta_\mathbb{N}^\infty \equiv \mathrm{id}_{\mathbb{N}^\mathbb{N}}$.

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

(2) Let $\delta_+ :\subseteq \mathbb{N}^\mathbb{N} \to X$ be recursive over S and let S be effective via δ'. By Theorem 3.1.25 it follows that δ_+ is computable w.r.t. δ', i.e. δ_+ is $(\delta_1'^\infty, \delta')$–computable. Thus, δ_+ is $(\mathrm{id}_{\mathbb{N}^\mathbb{N}}, \delta')$–computable. But this means $\delta_+ \leq \delta'$.

(3) Now let $\delta^- : X \rightrightarrows \mathbb{N}^\mathbb{N}$ be recursive over S and let S be effective via δ'. Again we can deduce by Theorem 3.1.25 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. $\delta' \le \delta_-$.

(4) Since δ^- is a right inverse operation of δ, it follows that δ_- is a restriction of δ and thus $\delta_- \le \delta$.

□

The Stability Theorem suggests the following definitions.

Definition 3.2.3 (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 $\delta_\mathbb{N}$. Each represented space (X, δ) with a computable point $c \in X$ gives rise to a natural structure $S = (\mathbf{N}, (X, \mathrm{id}_X, \delta))$ which is effective via $[\delta_\mathbb{N}, \delta]$. Although δ is also recursive over S, it is not automatically clear that there is a recursive right inverse δ^- of δ over S. On the other hand, the structure $S' = (\mathbf{N}, (X, \mathrm{id}_X, \delta, \delta^{-1}))$ obviously is strongly recursive via δ but it is not clear whether the inverse operation $\delta^{-1} : X \rightrightarrows \mathbb{N}^\mathbb{N}$ is $(\delta, \mathrm{id}_{\mathbb{N}^\mathbb{N}})$–computable. Later on, we will see that a special class of represented topological spaces gives rise to perfect structures.

In other words, 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 3.2.4 (Computable operations over perfect structures) An operation f over a perfect structure S is called (*strongly*) *computable over* S, if it is (strongly) computable w.r.t. a standard representation δ of S.

Now we can formulate a corollary of Theorem 3.1.21 and 3.1.25 which characterizes recursive operations over strongly perfect structures.

Theorem 3.2.5 (Operations over strongly perfect structures) *An operation over a strongly perfect structure is strongly computable, if and only if it is recursive.*

Moreover, we can deduce by Corollary 3.1.23 and Theorem 3.1.25 a characterization of computable operations over perfect structures.

Theorem 3.2.6 (Operations over perfect structures) *An operation over a perfect structure is computable, if and only if it admits a recursive extension.*

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

Corollary 3.2.7 (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*. To make this more precise we introduce some definitions. If $S = (S_1, ..., S_k)$, $S' = (S_1, ..., S_n)$ are structures with $n \geq k$, then we will say that S is a *prefix structure* of S' and we will write $S \sqsubseteq S'$. Vice versa, we will say that S' is an *extension* of S.

Now consider a natural structure S with an extension S'. In general, the additional prestructures 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 3.2.8 (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. W.l.o.g. we assume that $S = (S_1, ..., S_k)$ and $S' = (S_1, ..., S_{k+1})$. The general case can be deduced by induction. Let $\delta = [\delta_1, ..., \delta_k]$ and $\delta' = [\delta'_1, ..., \delta'_{k+1}]$ be standard representations of S, S', respectively. Then S' is effective via δ' and hence S is effective via $\delta'' := [\delta'_1, ..., \delta'_k]$. On the other hand, S is effective via δ and since S is perfect we can conclude $\delta \equiv \delta''$ by the Stability Theorem 3.2.1. If f is recursive over S, then f obviously is recursive over S' too. If f is an operation which is recursive over S', then f is strongly computable over S' by Theorem 3.2.5 since S' is strongly perfect. Thus, f is strongly computable w.r.t. δ' and thus w.r.t. δ''. Consequently, f is strongly computable w.r.t. δ too and thus it is recursive over S by Theorem 3.2.5 since S is strongly perfect. $\square$

If we revisit the proof and replace Theorem 3.2.5 by Theorem 3.2.6 then we obtain the following version of the Conservation Theorem for perfect structures.

Corollary 3.2.9 *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'.*

Now, we will consider the union of (pre)structures. If $S = (S_1, ..., S_n)$ and $R = (R_1, ..., R_k)$ are many-sorted prestructures, then the *union* $S \oplus R$ of S and R is defined to be the many sorted prestructure $(S_1, ..., S_n, R_1, ..., R_k)$, where double occurrences of prestructures are eliminated from the left to the right. More formally,

$$S \oplus (R_1) := \begin{cases} (S_1, ..., S_n) & \text{if} \quad (\exists i = 1, ..., n) S_i = R_1 \\ (S_1, ..., S_n, R_1) & \text{else} \end{cases}$$

and

$$S \oplus R := (...((S \oplus (R_1)) \oplus (R_2))...) \oplus (R_k).$$

For instance, $(A, B, C) \oplus (D, A, E) = (A, B, C, D, E)$. Furthermore, we will say that $S = (S_1, ..., S_n)$ is a *(strongly) perfect prefix structure*, if all $S_i' := (S_1, ..., S_i)$, $i = 1, ..., n$ are perfect structures.

Theorem 3.2.10 (Union Theorem) *If S is a (strongly) perfect structure and R a (strongly) perfect prefix structure, then $S \oplus R$ is a (strongly) perfect structure.*

Proof. We prove the strong version. Let $S = (S_1, ..., S_n)$ be a strongly perfect structure with standard representation δ and let $R = (R_1, ..., R_k)$ be a strongly perfect prefix structure. By induction one can show that there are representations $\delta_1, ..., \delta_k$ such that $[\delta_1, ..., \delta_i]$ is a standard representation of the strongly perfect prefix structure $(R_1, ..., R_i)$ for $i = 1, ..., k$. By induction on $i \le k$ we will prove that $T_i := S \oplus (R_1, ..., R_i)$ is a strongly perfect structure with standard representation δ' such that the component δ_j' of δ' which represents the universe of R_j is equivalent to δ_j for $j = 1, ..., i$.
$\underline{i = 1.}$ Since (R_1) is a natural structure, we can conclude $R_1 = \mathbf{N}$ and since S is a natural structure we obtain that $T_1 = S \oplus (R_1) = S$ is strongly perfect. Hence $\delta' := \delta$ is a standard representation of T_1 and the corresponding component of δ' which represents $\mathbb{N}$ is equivalent to $\delta_1 \equiv \delta_{\mathbb{N}}$.

$\underline{i \to i+1.}$ Now, let us assume that T_i is strongly perfect for some $i = 1, ..., k-1$ with standard representation δ'. Let δ'_j be the component of δ' which represents the universe of R_j for $j = 1, ..., i$. By induction hypothesis we can assume $\delta'_j \equiv \delta_j$. If R_{i+1} is a prestructure of T_i, then $T_{i+1} = T_i$ and nothing is to be proved. Hence, let us assume that R_{i+1} is not a prestructure of T_i. We show that $T_{i+1} = T_i \oplus (R_{i+1})$ is strongly perfect with standard representation $\delta'' := [\delta', \delta_{i+1}]$. Since δ' is a recursive retraction over T_i and δ_{i+1} is a recursive retraction over $(R_1, ..., R_{i+1})$, we can conclude that δ', δ_{i+1} are recursive retractions over T_{i+1} and hence the same holds for δ''. Furthermore, the initial operations of T_i are strongly computable w.r.t. δ' and the initial operations of R_{i+1} are strongly computable w.r.t. $[\delta_1, ..., \delta_{i+1}]$. Since $\delta'_j \equiv \delta_j$ for $j = 1, ..., i$ we can conclude that the initial operations of R_{i+1} are strongly computable w.r.t. δ'' too. Thus, the initial operations of $T_{i+1} = T_i \oplus (R_{i+1})$ are strongly computable w.r.t. δ'' and altogether, T_{i+1} is strongly perfect with standard representation δ''. $\square$

We want to close this section with an important selection property of sets over $\mathbf{N}$. Implicitly, this property has already been used in the proof of Proposition 3.2.2.

Theorem 3.2.11 (Selection Theorem) *Let S be a perfect structure, $F :\subseteq \mathbb{N}^{\mathbb{N}} \rightrightarrows X$ a recursive operation over S. Then there is a recursive function $f :\subseteq \mathbb{N}^{\mathbb{N}} \to X$ over S, such that* $\mathrm{dom}(f) = \mathrm{dom}(F)$ *and $f(p) \in F(p)$ for all $p \in \mathrm{dom}(F)$. In this situation f is called a* recursive selector *of F. An analogous property holds for* $\mathbb{N}$ *instead of* $\mathbb{N}^{\mathbb{N}}$.

Proof. Let S be a perfect structure with standard representation δ and let δ_X be the corresponding induced representation of X. If $F :\subseteq \mathbb{N}^{\mathbb{N}} \rightrightarrows X$ is recursive over S, then it is $(\mathrm{id}_{\mathbb{N}^{\mathbb{N}}}, \delta_X)$–computable by Theorem 3.1.25 since $\delta_{\mathbb{N}}^{\infty} \equiv \mathrm{id}_{\mathbb{N}^{\mathbb{N}}}$. Then there is a computable function $G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that $F(p) = \{\delta_X G\langle p, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\}$ and $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}(\delta_X G)$ for all $p \in \mathrm{dom}(F)$. Define $g :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by $g\langle p, q\rangle := G\langle p, p\rangle$ and $f' :\subseteq \mathbb{N}^{\mathbb{N}} \to X$ by $f'(p) := \delta_X G\langle p, p\rangle$ for all $p, q \in \mathbb{N}^{\mathbb{N}}$. Then

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

and $\langle p, \mathbb{N}^{\mathbb{N}}\rangle \subseteq \mathrm{dom}(\delta_X g)$ for all $p \in \mathrm{dom}(f')$, i.e. f' is $(\mathrm{id}_{\mathbb{N}}^{\mathbb{N}}, \delta_X)$–computable via g and $f'(p) = \delta_X G\langle p, p\rangle \in F(p)$ for all $p \in \mathrm{dom}(F)$. Hence f' admits a recursive extension f'' over S by Corollary 3.1.23 and $f := f''|_{\mathrm{dom}(F)}$ is a recursive function over S with $f(p) \in F(p)$ for all $p \in \mathrm{dom}(F)$. $\square$

This theorem shows that many-valued operations can be replaced by functions over the natural numbers $\mathbf{N}$ in a reasonable way. Unfortunately, this result cannot be transferred to arbitrary higher-dimensional spaces Y instead of $\mathbb{N}^{\mathbb{N}}$. If such a selection property would hold in general, then there were no need for many-valued operations at all.

We close this section with a short analysis of our proofs and with a revision of the operators of our high-level language. For the proof that all computable functions $f :\subseteq \mathbb{N} \to \mathbb{N}$ are recursive over $\mathbf{N}$, i.e. Theorem 3.1.6, we have (implicitly) used the operators projection, juxtaposition, product, composition, iteration and inversion. For the proof that all computable functions $f :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ are recursive over $\mathbf{N}$, i.e. Theorem 3.1.9, we have, additionally, used evaluation, transposition and exponentiation. But neither for these theorems nor for the other results of this section we have used the sequentialization operator. Indeed, all results on perfect structures remain true, if we remove sequentialization from our high-level language. However, the class of perfect structures could be smaller in this case and, actually, this is the reason why we have included sequentialization (although, it evolves its full power not until it is applied to asymmetric spaces, which are out of scope of the present investigation).

3.2.2 Reducibility of structures

For a comparison of structures it will be useful to have a notion of reducibility and a notion of equivalence for structures.

Definition 3.2.12 (Reducibility of structures) Let S, S' be structures with common universe.

(1) S is *reducible* to S', for short $S \leq S'$, if all initial operations of S have recursive extensions over S'.

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

Analogously, S is *strongly reducible* to S', for short $S \leq_s S'$, if all initial operations of S are recursive over S'. If, additionally, S' is strongly reducible to S, then S is called *strongly equivalent* to S', for short $S \equiv_s S'$.

The following proposition formulates a criterion which characterizes equivalence of structures in case that one of the structures is perfect. This result will be useful to prove equivalence in applications.

Proposition 3.2.13 (Equivalence of structures) *Let S, S' be structures with common universe and let S be perfect. Then $S \equiv S'$, if and only if all initial operations of S have recursive extensions over S' and all initial operations of S' are computable over S. An analogous property holds for strong reducibility.*

The proof is obvious by definition of reducibility and since computable operations over recursive structures have recursive extensions over these structures by Corollary 3.1.23. The strong version follows by Theorem 3.1.21. Our next result about equivalent structures states that (strong) perfectness is a property which is invariant under strong equivalence.

Theorem 3.2.14 (Invariance theorem) *If S, S' are (strongly) equivalent natural structures, then S is (strongly) perfect, if and only if S' is (strongly) perfect.*

The proof follows immediately from the following proposition which states that in case $S \leq S'$ effectiveness is inherit from the right to the left while recursiveness is inherit from the left to the right.

Proposition 3.2.15 *Let S, S' be natural structures with common universe.*

(1) If S is (strongly) reducible to S' and S' is (strongly) effective via a natural representation δ', then S is (strongly) effective via δ' too.

(2) If S is (strongly) reducible to S' and S is (strongly) recursive via a representation δ, then S' is (strongly) recursive via δ too.

Here (1) follows immediately from Theorem 3.1.25 and (2) follows by an easy inductive proof. The following theorem is a further implication of the previous proposition.

Theorem 3.2.16 (Uniqueness Theorem) *If S, S' are (strongly) perfect structures with common universe such that S is (strongly) reducible to S', then S is (strongly) equivalent to S'.*

Proof. Let S, S' be (strongly) perfect with (strong) standard representations δ, δ', respectively. If S is (strongly) reducible to S', then S' is (strongly) recursive via δ by the previous Proposition. Hence, $\delta \equiv \delta'$ follows by the Stability Theorem 3.2.1. Since S' is (strongly) effective via δ', it is (strongly) effective via δ too. Hence, all initial operations of S' are (strongly) computable over S. Since S is (strongly) perfect, it follows by Proposition 3.2.13 that S is (strongly) equivalent to S'. □

We immediately obtain the following corollary.

Corollary 3.2.17 *Two perfect structures S, S' with common universe are either equivalent, i.e. $S \equiv S'$ holds, or incomparable, i.e. $S \not\leq S'$ and $S' \not\leq S$ holds.*

One can prove a corresponding strong version. The final result of this section shows that two strongly perfect structures are equivalent, if and only if their standard representations are equivalent.

Theorem 3.2.18 (Representation Theorem) *Let S, S' be strongly perfect structures with common universe and standard representations δ, δ', respectively. Then $S \equiv_s S'$, if and only if $\delta \equiv \delta'$. The same holds for perfect structures with $\equiv$ instead of $\equiv_s$.*

Proof. The "if" direction follows since $\delta \equiv \delta'$ implies that all initial operations of S are strongly computable w.r.t. δ'. Consequently, by Theorem 3.1.21 it follows that all initial operations of S are recursive over S', i.e. $S \leq_s S'$. The other reduction $S' \leq_s S$ follows by symmetry and the "only if" direction follows by the following proposition. □

Proposition 3.2.19 *Let S, S' be structures with common universe, let δ be a representation of S which is recursive over S and let S' be effective via a natural representation δ'. If $S \leq S'$, then $\delta \leq \delta'$.*

Proof. If $S \leq S'$ and S' is effective via δ' then by Proposition 3.2.15 S is effective via δ' too. By Proposition 3.2.2 $\delta \leq \delta'$ follows. □

Especially, for two perfect structures S, S' with common universe and standard representations δ, δ', respectively, $\delta \equiv \delta'$ implies $S \equiv S'$ and, on the other hand, $S \leq S'$ implies $\delta \leq \delta'$. However, this result does not state that $\delta \leq \delta'$ implies $S \leq S'$ and indeed there are incomparable perfect structures S, S' such that $\delta \leq \delta'$ holds. But we will not prove this here.

We close this section with two further definitions. Sometimes, we do not want to compare whole structures but two prestructures of a single structure. For this purpose we will use recursive isomorphisms.

Definition 3.2.20 (Recursive isomorphism) A function $f : X \to Y$ over a structure S is called a *recursive isomorphism* over S, if it is bijective and f, as well as $f^{-1} : Y \to X$ are recursive over S. In this situation X is called *recursively isomorphic* to Y over S.

Similarly, we define recursive embeddings.

Definition 3.2.21 (Recursive embeddings) A function $f : X \to Y$ over a structure S is called a *recursive embedding* over S, if it is injective and f, as well as $f^{-1} :\subseteq Y \to X$ are recursive over S.

3.2.3 Domains over structures

In this section we want to discuss domains of recursive operations. Our aim is not to classify these domains but to make them available for restrictions of operations.

If $f :\subseteq X \rightrightarrows Y$ is an operation and $A \subseteq X, B \subseteq Y$, then the *restriction in the domain* $f|_A :\subseteq X \rightrightarrows Y$ is defined by $f|_A(x) := f(x)$ for all $x \in \mathrm{dom}(f|_A) := \mathrm{dom}(f) \cap A$ and the *restriction in the range* $f|^B :\subseteq X \rightrightarrows Y$ is defined by $f|^B(x) := f(x)$ for all $x \in \mathrm{dom}(f|^B) := \{x \in X : f(x) \subseteq B\}$.

Occasionally, we will also use the *restriction in the source* $f\rfloor_A : A \rightrightarrows Y$, which is defined by $f\rfloor_A(x) := f(x)$ for all $x \in A$ and the *restriction in the target* $f\rceil^B :\subseteq X \rightrightarrows B$, which is defined by $f\rceil^B(x) := f(x)$ for all $x \in \mathrm{dom}(f\rceil^B) := \mathrm{dom}(f|^B)$.

Since our structures have only finitely many initial operations, the class of recursive operations over a fixed structure is countable. If the universe of the structure is at least countable, then its power set is uncountable. Thus, a restriction of a recursive operation in the domain or range need not to be recursive in general. With the next definition we single out a countable class of sets which is admissible for restriction.

Definition 3.2.22 (Domains) Let S be a natural structure, X a set over S and $A \subseteq X$. Then A is called a *domain over* S, if the restriction $\mathrm{id}_X|_A$ is recursive over S.

By definition each domain over a structure is the domain of a recursive operation. Since $(\mathrm{id}_X, f)_1 = \mathrm{id}_X|_{\mathrm{dom}(f)}$ for $f :\subseteq X \rightrightarrows Y$, the inverse implication is also true, i.e. domains over a structure are exactly those sets which occur as domains of recursive operations.

Proposition 3.2.23 (Domains) *Let S be a natural structure, X a set over S and $A \subseteq X$. Then A is a domain over S, if and only if there is a recursive operation $f :\subseteq X \rightrightarrows Y$ over S such that* $\mathrm{dom}(f) = A$.

Moreover, since $(f, \mathrm{id}_X|_A)_1 = f|_A$ and $\mathrm{id}_X|_A \circ g = g|^A$ for operations $f :\subseteq X \rightrightarrows Y, g :\subseteq Z \rightrightarrows X$ and $A \subseteq X$, domains are admissible for restriction, or more precisely:

Theorem 3.2.24 (Restriction Theorem) *Let S be a natural structure and $f :\subseteq X \rightrightarrows Y, g :\subseteq Z \rightrightarrows X$ recursive operations over S. If $A \subseteq X$ is a domain over S, then $f|_A :\subseteq X \rightrightarrows Y$ and $g|^A :\subseteq Z \rightrightarrows X$ are also recursive over S.*

Implicitly, we have already used this result in the proof of the Selection Theorem 3.2.11. If we have a structure S and a set X over S then it might be useful to consider domains $A \subseteq X$ as prestructures. This can be done by virtue of the canonical embedding $\iota : A \to X$ and its inverse $\iota^{-1} :\subseteq X \to A$, as the following theorem states.

Theorem 3.2.25 (Subspace theorem) *Let S be a (strongly) perfect structure, X a set over S with a recursive point $c \in X$, let $A \subseteq X$ be a domain over S with $c \in A$, and let $\iota : A \to X, x \mapsto x$ be the canonical embedding. Then the structure $S \oplus R$ with $R := (A, \iota, \iota^{-1})$ is also a (strongly) perfect structure and ι is a recursive embedding over $S \oplus R$.*

Proof. If δ is a standard representation of S and δ_X is the corresponding representation of X, then $\delta_A := \iota^{-1} \circ \delta_X$ is a representation of A, such that $[\delta, \delta_A]$ is a standard representation of $S \oplus R$. If δ_X^- is a right inverse of δ_X, then $\delta_A^- := \delta_X^- \circ \iota$ is a suitable right inverse of δ_A. Here, ι^{-1} is strongly (δ_X, δ_A)–computable since A is a domain over S. $\square$

3.2.4 Recursive sets over structures

In this section we want to study some canonical definitions for sets over structures. Especially, we will choose one of these definitions as standard way to embed predicates into structures.

Definition 3.2.26 (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 is 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 is 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 A^c is initially semi-recursive over S.

(4) A is *decidable* over S, if there is 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. Figure 3.2 illustrates the situation.

Now we will give some examples for the classes of sets that we have defined. Obviously, the empty set $\emptyset$ considered as a subset of a set X over a natural

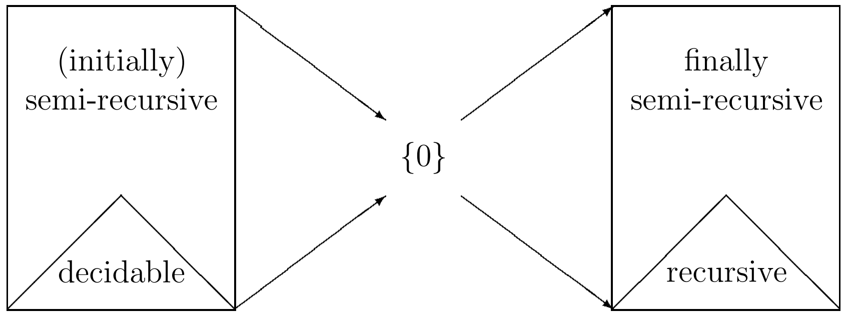

Figure 3.2: Effective sets over structures

structure S is always recursive and decidable. Here, $\emptyset$ is finally semi-recursive since there is an operation $f :\subseteq \mathbb{N} \rightrightarrows X$ with $f(0) =\uparrow$ and thus we obtain $f\{0\} = \emptyset$. Furthermore, the set X considered as a subset of itself always is 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 $\Omega_A : \{()\} \rightrightarrows X, () \mapsto A$, the omnipotent operation of A, is recursive over S. Especially, the following holds:

Proposition 3.2.27 *Let S be a natural structure, X a set over S and $x \in X$. Then x is recursive, if and only if $\{x\}$ is finally semi-recursive.*

Furthermore, over natural structures S a subset A is decidable, if and only if its *characteristic function*

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

is recursive over A. We introduce the *semi-characteristic operation* to characterize semi-recursive sets A in a corresponding way.

Definition 3.2.28 (Semi-characteristic operation) For each set X and each subset A 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.

Up to now we have investigated (pre)structures $(X, f_1, ..., f_n)$ with operations $f_1, ..., f_n$. We have defined a standard way to consider constants as operations and we have assumed that there always will be a way to consider predicates as operations too. From now on we will fix the following standard way to consider predicates as operations. If S is a structure with prestructure $(X, f_1, ..., f_n, A)$ such that A is a subset of X, then we will identify this prestructure with the prestructure $(X, f_1, ..., f_n, c_A)$, where $c_A : X \rightrightarrows \mathbb{N}$ is the semi-characteristic operation of A. In other words: predicates are represented by their semi-characteristic operations.

The question appears, how domains are related to the sets introduced in this section. Since $(f, c_A)_0 = f|_{c_A^{-1}\{0\}} = f|_A$ all semi-recursive sets are domains. On the other hand, there exist recursive functions $g :\subseteq \mathbb{N} \to \mathbb{N}$ with $\mathrm{dom}(g) = \mathbb{N} \setminus \{0\}$ and $\mathrm{dom}(g \circ c_A) = (c_A^{-1}\{0\})^c = A^c$, i.e. complements of semi-recursive sets are also domains. Some conclusions are formulated in the following proposition.

Proposition 3.2.29 (Recursive sets as domains) *Over natural structures all semi-recursive, recursive, and decidable subsets, as well as the complements of all semi-recursive sets are domains.*

In general, there is no converse statement of this type. In contrast to classical recursion theory, the domains of recursive operations can be much more complicated than semi-recursive sets.

In the following we will sometimes consider sequences of sets too. Therefore, we introduce the following notions.

Definition 3.2.30 (Recursively given sequences) Let S be a structure, X a set over S and let $(A_n)_{n\in\mathbb{N}}$ be a sequence of subsets of X.

(1) $(A_n)_{n\in\mathbb{N}}$ is a *recursively given sequence of (initially) semi-recursive sets* over S, if there is a recursive operation $f : X \times \mathbb{N} \rightrightarrows \mathbb{N}$ over S, such that $f(x, n) = c_{A_n}(x)$ for all x, n,

(2) $(A_n)_{n\in\mathbb{N}}$ is a *recursively given sequence of finally semi-recursive sets* over S, if the sets A_n are non-empty and the operation $f : \mathbb{N} \rightrightarrows X$, defined by $f(n) := A_n$ for all $n \in \mathbb{N}$ is recursive over S.

Again, we will identify a prestructure $(X, f_1, ..., f_k, (A_n)_{n\in\mathbb{N}})$ with a sequence of sets $(A_n)_{n\in\mathbb{N}}$ with the prestructure $(X, f_1, ..., f_k, f)$, where f is the operation given in (1) of the previous definition. In other words, sequences of predicates are considered as recursively given sequences of semi-recursive sets. Now we will formulate some closure properties of the defined classes of sets.

Proposition 3.2.31 (Closure properties of sets) *Let S be a natural structure, X, Y sets over S and $A, B, A_n \subseteq X$ for all $n \in \mathbb{N}$ and $C \subseteq Y$.*

(1) If A, B are semi-recursive, or decidable over S, then so is $A \cap B$.

(2) If A, B are semi-recursive, or finally semi-recursive, or recursive, or decidable over S, then so is $A \cup B$.

(3) If $(A_n)_{n\in\mathbb{N}}$ is a recursively given sequence of semi-recursive or of non-empty finally semi-recursive sets over S, then $\bigcup_{n=0}^{\infty} A_n$ has the corresponding property.

(4) If A is decidable over S, then so is A^c.

(5) If A, C are semi-recursive, or finally semi-recursive, or recursive, or decidable over S, then so is $A \times C \subseteq X \times Y$.

(6) If A is semi-recursive, or decidable over S, then $A \times X \times X...$ is a subset of $X^{\mathbb{N}}$ with the corresponding property.

(7) If $(A_n)_{n\in\mathbb{N}}$ is a recursively given sequence of finally semi-recursive sets over S, then $\times_{n=0}^{\infty} A_n$ is a finally semi-recursive subset of $X^{\mathbb{N}}$ over S.

Proof. If A, B, C are decidable, then $\mathrm{cf}_A, \mathrm{cf}_B, \mathrm{cf}_C$ are recursive and so are $\mathrm{cf}_{A\cup B} = \min(\mathrm{cf}_A, \mathrm{cf}_B)$ and $\mathrm{cf}_{A\cap B} = \max(\mathrm{cf}_A, \mathrm{cf}_B)$. Hence $A \cup B$ and $A \cap B$ are decidable. Moreover, A^c is decidable, since $\mathrm{cf}_{A^c} = 1 \dot{-} \mathrm{cf}_A$, $A \times C$ is decidable since $\mathrm{cf}_{A\times C} = \max(\mathrm{cf}_A \times \mathrm{cf}_C)$ and $A \times X \times X \times ...$ is decidable since $\mathrm{cf}_{A\times X\times X\times ...}((x_n)_{n\in\mathbb{N}}) = \mathrm{cf}_A(x_0)$.

If A, B, C be semi-recursive, then c_A, c_B, c_C are recursive and so are $c_{A\cup B} = \min(c_A, c_B)$ and $c_{A\cap B} = \max(c_A, c_B)$. Hence $A\cup B$ and $A\cap B$ are semi-recursive. Moreover, $A \times C$ is semi-recursive since $c_{A\times C} = \max(c_A \times c_C)$, $A \times X \times X \times ...$ is semi-recursive since $c_{A\times X\times X\times ...}((x_n)_{n\in\mathbb{N}}) = c_A(x_0)$.

Let A, B, C are finally semi-recursive. W.l.o.g. we can assume that A, B, C are non-empty since otherwise nothing is to be proved. Now $A \cup B$ is finally semi-recursive, since $\Omega_{A\cup B} = \Omega_A \cup \Omega_B$ and $A \times C$ is finally semi-recursive, since $\Omega_{A\times C} = \Omega_A \times \Omega_C$.

If A, B, C are recursive, then A, B, C are finally semi-recursive and A^c, B^c, C^c are semi-recursive. Hence $A \cup B$ is finally semi-recursive, $(A \cup B)^c = A^c \cap B^c$ is semi-recursive and thus $A \cup B$ is recursive. Moreover, $A \times C$ is finally semi-recursive and $(A \times C)^c = (A^c \times Y) \cup (X \times C^c)$ is semi-recursive and thus $A \times C$ is recursive.

If $(A_n)_{n\in\mathbb{N}}$ is a recursively given sequence of semi-recursive sets, then there is an operation $f : X \times \mathbb{N} \rightrightarrows \mathbb{N}$ such that $f(x, n) = c_{A_n}(x)$ for all $n \in \mathbb{N}, x \in X$. Since $\bigcup f$ is recursive and $\bigcup f(x) = \bigcup_{n=0}^{\infty} f(x, n) = \bigcup_{n=0}^{\infty} c_{A_n}(x) = c_{\cup_{n=0}^{\infty} A_n}(x)$ for all $x \in X$, we can conclude that $\bigcup_{n=0}^{\infty} A_n$ is semi-recursive.

If $(A_n)_{n\in\mathbb{N}}$ is a recursively given sequence of non-empty finally semi-recursive sets, then the operation $f : \mathbb{N} \rightrightarrows X$ with $f(n) := A_n$ is recursive over S. Since

$\Omega_{\bigcup_{n=0}^{\infty} A_n} = \bigcup f$, we can conclude that $\bigcup_{n=0}^{\infty} A_n$ is finally semi-recursive. Since $\Omega_{\times_{n=0}^{\infty} A_n} = [f]$, it follows that $\times_{n=0}^{\infty} A_n$ is finally semi-recursive. $\square$

Especially, we can conclude that all finite sets of recursive points are finally semi-recursive. As a direct consequence of the previous proposition we obtain the following weak version of the Projection Theorem.

Theorem 3.2.32 (Projection Theorem) *If S is a natural structure, X a set over S and $A \subseteq X \times \mathbb{N}$ is semi-recursive over S, then*

$$\mathrm{pr}_1(A) = \{x : (\exists n)(x, n) \in A\} \subseteq X$$

is semi-recursive over S too.

Proof. Let $A_n := \{x \in X : (x, n) \in A\}$. Then $c_{A_n}(x) = c_A(x, n)$ and $(A_n)_{n\in\mathbb{N}}$ is a recursively given sequence of semi-recursive sets over S. Thus, $\mathrm{pr}_1(A) = \bigcup_{n=0}^{\infty} A_n$ is semi-recursive over S by the previous proposition. $\square$

Another closure property illustrates how semi-recursive sets can be used as "predicates". The following proposition gives a generalization of the "definition-by-cases" operator.

Proposition 3.2.33 (Definition by cases) *Let S be a natural structure, let $f, g :\subseteq X \rightrightarrows Y$ be recursive operations over S and let $A, B \subseteq X$ be semi-recursive over S. Then $h :\subseteq X \rightrightarrows Y$, defined by*

$$h(x) := \begin{cases} f(x) & \text{if } x \in A \setminus B \\ g(x) & \text{if } x \in B \setminus A \\ f(x) \cup g(x) & \text{if } x \in A \cap B \end{cases}$$

for all $x \in X$, is recursive over S.

Proof. Let $s : X \to Y$ be a constant recursive function over S with value $y := s(x)$ for all $x \in X$. Let $t_A := 1 \dot{-} c_A$, $t_B := 1 \dot{-} c_B$, $h_f := ((s \times 1)|_{t_A}(f \times 0))$, $h_g := ((s \times 1)|_{t_B}(g \times 0)) :\subseteq X \rightrightarrows Y \times \mathbb{N}$. By the Proposition on the definition-by-cases operator 2.5.8 h_f, h_g are recursive over S. We obtain

$$h_f(x) = \begin{cases} \{(y, 1)\} \cup (f(x) \times \{0\}) & \text{if } x \in A \cap \mathrm{dom}(f) \\ \{(y, 1)\} & \text{if } x \notin A \end{cases}$$

and analogously for h_g with g, B instead of f, A, respectively. Thus

$$(h_f \cup h_g)(x) = \begin{cases} \{(y,1)\} \;\cup\; (f(x) \times \{0\}) & \text{if } x \in (A \setminus B) \cap \mathrm{dom}(f) \\ \{(y,1)\} \;\cup\; (g(x) \times \{0\}) & \text{if } x \in (B \setminus A) \cap \mathrm{dom}(g) \\ \{(y,1)\} \;\cup\; (f(x) \times \{0\}) & \text{if } x \in A \cap B \cap \mathrm{dom}(f) \cap \mathrm{dom}(g) \\ \qquad\qquad \cup\; (g(x) \times \{0\}) & \\ \{(y,1)\} & \text{if } x \notin A \cup B \end{cases}$$

and hence $h = (h_f \cup h_g)_0$. $\square$

As an easy consequence we get the following proposition.

Proposition 3.2.34 (Decidable subsets) *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.*

Some further closure properties involve sets and operations. Essentially, the proposition states that initially semi-recursive sets are preserved by preimages and finally semi-recursive sets are preserved by images of recursive operations.

Proposition 3.2.35 (Closure properties of sets and operations) *Let S be a natural structure, $f :\subseteq X \rightrightarrows Y$ a recursive operation over S, $A, A_n \subseteq X$ and $B, B_n \subseteq Y$ for all $n \in \mathbb{N}$. Then the following holds over S:*

(1) If B is semi-recursive, then $f^{-1}(B)$ is semi-recursive.

(2) If $A \subseteq \mathrm{dom}(f)$ is finally semi-recursive, then $f(A)$ is finally semi-recursive.

(3) If $\mathrm{dom}(f)$ is finally semi-recursive, then $\mathrm{range}(f)$ and $\mathrm{graph}(f)$ are finally semi-recursive.

(4) If $(B_n)_{n\in\mathbb{N}}$ is a recursively given sequence of semi-recursive sets, then $(f^{-1}(B_n))_{n\in\mathbb{N}}$ has the same property.

(5) If $(A_n)_{n\in\mathbb{N}}$ is a recursively given sequence of non-empty finally semi-recursive sets with $A_n \subseteq \mathrm{dom}(f)$ for all $n \in \mathbb{N}$, then $(f(A_n))_{n\in\mathbb{N}}$ is a recursively given sequence of non-empty finally semi-recursive sets too.

Proof. The whole proof follows from the formulas $c_{f^{-1}(B)} = c_B \circ f$, $\Omega_{f(A)} = f \circ \Omega_A$, $\Omega_{\mathrm{range}(f)} = f \circ \Omega_{\mathrm{dom}(f)}$ and $\Omega_{\mathrm{graph}(f)} = (\mathrm{id}_X, f) \circ \Omega_{\mathrm{dom}(f)}$. $\square$

Especially, we can conclude that the image $f(x)$ of a recursive point $x \in \mathrm{dom}(f)$ under a recursive operations f is a finally semi-recursive set and that the image $\mathrm{range}(f)$ of a total recursive function $f : \mathbb{N} \to X$ is finally semi-recursive. In other words: each recursive sequence enumerates a finally semi-recursive set.

3.2.5 Complete structures

We have already used the fact that the omnipotent operation $\Omega_X : \{()\} \rightrightarrows X$ is recursive for some sets X over structures. We will call such sets *complete*.

Definition 3.2.36 (Complete sets) A set X over a structure S is called *complete*, if Ω_X is recursive over S.

Obviously, a set X over a structure is complete, if and only if it is finally semi-recursive as a subset of itself over S. Now, we will call a structure S *complete*, if its universe is complete over S.

Definition 3.2.37 (Complete structure) A structure S is called a *complete structure*, if its universe X is complete over S.

Since $\Omega_{\mathbb{N}} = \bigcup \mathrm{id}_{\mathbb{N}}$, the structure of natural numbers $\mathbf{N}$ is complete. Complete structures have several nice properties. First we mention that each set over a complete natural structure is complete over S.

Proposition 3.2.38 (Sets over complete structures) *Each set Y over a complete natural structure S is complete (and thus finally semi-recursive) over S.*

Proof. The proof follows by structural induction from the following formulas. If $X = X_1 \times ... \times X_n$, then $\Omega_{X_i} = (\Omega_X)_i$. Moreover, $\Omega_{Y\times Z} = \Omega_Y \times \Omega_Z$ and $\Omega_{Y^{\mathbb{N}}} = [(\Omega_Y \times \mathrm{id}_{\mathbb{N}})_1]$. □

The next interesting property is that over complete natural structures semi-recursiveness implies final semi-recursiveness.

Proposition 3.2.39 (Semi-recursiveness over complete structures) *Over complete natural structures each semi-recursive set is also finally semi-recursive and each decidable subset is recursive.*

The proof 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 in case of completeness semi-recursive sets are graphs of recursive operations.

Theorem 3.2.40 (Uniformization Theorem) *Let X, Y be sets over a natural structure S and let $A \subseteq X \times Y$. If Y is complete 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, $\mathrm{graph}(f) = A$. $\square$

In the following we will occasionally apply this theorem implicitly by saying that an operation f is constructed by *uniformization*. We end this section with a criterion which guarantees completeness of structures.

Proposition 3.2.41 (Complete recursive structures) *If δ is a recursive representation of a natural structure S, and* $\mathrm{dom}(\delta)$ *is finally semi-recursive over S, then S is complete.*

Proof. If $A := \mathrm{dom}(\delta)$ is finally semi-recursive, then $X = \delta(A)$ is finally semi-recursive over S too and S is complete. $\square$

3.2.6 Structures with equality

An important predicate which we want to study in this section is equality. If X is a set over a structure S, then the *equality on* X is the predicate $=_X$, that is the set $\{(x,x) : x = x\} \subseteq X \times X$.

The first result shows that effective natural structures with semi-recursive equality are countable. Especially, there are no uncountable structures with decidable equality (cf. [Wei]).

Proposition 3.2.42 (Effective structures with equality) *Let S be a natural structure with universe X which is effective via a natural representation δ. If the equality $=_X$ is semi-recursive over S, then X is countable.*

Proof. Let $=_X$ be semi-recursive over S. Then there is a total recursive operation $f : X \times X \rightrightarrows \mathbb{N}$ over S such that $=_X$ is equal to $f^{-1}\{0\}$. Since S is effective via δ by Theorem 3.1.25 there is a computable $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that $f[\delta,\delta](p) = \{\delta_{\mathbb{N}} F\langle p, r\rangle : r \in \mathbb{N}^{\mathbb{N}}\}$ for all $p \in \mathrm{dom}[\delta,\delta]$, i.e.

$$\delta(p) = \delta(q) \iff (\exists r \in \mathbb{N}^{\mathbb{N}})\delta_{\mathbb{N}} F\langle\langle p,q\rangle, r\rangle = 0$$

for all $p,q \in \mathrm{dom}(\delta)$. Let $\varphi : \mathbb{N}^* \to \mathbb{N}^*$ be a total computable and monotone function which approximates F. Define $g :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ by

$$g(p) := \mu k[(\exists r \in \mathbb{N}^{\mathbb{N}})\nu^*(k) \sqsubseteq \langle\langle p,p\rangle, r\rangle \text{ and } 0 \sqsubseteq \varphi\nu^*(k)]$$

for all $p \in \mathbb{N}^{\mathbb{N}}$. Then $\delta(p) = \delta(\nu^*(g(p))\mathbb{N}^{\mathbb{N}})$ for all $p \in \mathrm{dom}(\delta)$. Thus, $X = \mathrm{range}(\delta) = \{\delta(\nu^*(n)\mathbb{N}^{\mathbb{N}}) : n \in \mathrm{range}(g)\}$ is countable. $\square$

The next observation shows that finally semi-recursive subsets are also semi-recursive over structures with equality.

Proposition 3.2.43 (Structures with equality) *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 3.2.39 we get the following corollary.

Corollary 3.2.44 (Complete structures with equality) *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.*

We can apply the previous observation to prove the following graph theorem for structures with equality.

Proposition 3.2.45 (Graph of operations 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 over S.*

Proof. Let f be recursive and $\mathrm{dom}(f)$ semi-recursive. We already know by Proposition 3.2.35 and Corollary 3.2.44 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 3.2.40. Moreover, $c_{\mathrm{dom}(f)} = c_{\mathrm{graph}(f)} \circ (\mathrm{id}_X \times \Omega_Y)$ for all x, i.e. $\mathrm{dom}(f)$ is semi-recursive. □

Another property of structures with equality that we want to mention is the following inversion property which is an immediate corollary of the graph proposition. We just have to use the fact that the canonical twist function $X \times Y \to Y \times X, (x, y) \mapsto (y, x)$ is recursive over natural structures.

Corollary 3.2.46 (Inversion of operations 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. If f is recursive and* $\mathrm{dom}(f)$ *is semi-recursive over S, then the inverse operation $f^{-1} :\subseteq Y \rightrightarrows X$ is recursive and* $\mathrm{dom}(f^{-1}) = \mathrm{range}(f)$ *is semi-recursive over S.*

Finally, we will formulate a proposition which shows that the effective subsets of the natural numbers are the well-known classical ones. Here, we will use the classical notions of *recursive enumerability* and *classical decidability* (cf. [Odi89]).

Proposition 3.2.47 (Subsets of natural numbers) *Let $A \subseteq \mathbb{N}$.*

(1) A is semi-recursive over $\mathbf{N}$, if and only if A is finally semi-recursive over $\mathbf{N}$, if and only if A is recursively enumerable.

(2) A is recursive over $\mathbf{N}$, if and only if A is decidable over $\mathbf{N}$, if and only if A is classically decidable/recursive.

Proof. By Theorem 3.2.5 a total function $f : \mathbb{N} \to \mathbb{N}$ is recursive over $\mathbf{N}$, if and only if it is computable w.r.t. to $\delta_{\mathbb{N}}$. And since f is total this holds, if and only if f is classically computable. Since a set A is classically decidable, if and only if its characteristic function is classically computable, (2) follows.

Now let A be recursively enumerable. Since nothing has to be proved if A is empty we assume that A is non-empty. Then there is a classically recursive total function $f : \mathbb{N} \to \mathbb{N}$ such that $A = \text{range}(f)$. Since f is recursive over $\mathbf{N}$, it follows that $A = \text{range}(f)$ is finally semi-recursive and thus semi-recursive, since $=_{\mathbb{N}}$ is semi-recursive over $\mathbf{N}$. On the other hand, let A be semi-recursive. Especially, A is finally semi-recursive, since $\mathbf{N}$ is complete and $c_A : \mathbb{N} \rightrightarrows \mathbb{N}$ is recursive over $\mathbf{N}$. By Theorem 3.2.5 c_A is $(\delta_{\mathbb{N}}, \delta_{\mathbb{N}})$–computable, i.e. there is a computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that

$$c_A(n) = \{F\langle \hat{n}, q\rangle(0) : q \in \mathbb{N}^{\mathbb{N}}\}$$

Thus there is a computable, total and monotone function $\varphi : \mathbb{N}^* \to \mathbb{N}^*$ which approximates F. Hence

$$A = c_A^{-1}\{0\} = \{n : (\exists i \in \mathbb{N})\pi_1\nu^*(i) \sqsubseteq \hat{n} \text{ and } 0 \sqsubseteq \varphi\nu^*(i)\}$$

is recursively enumerable by the Projection Theorem. □

3.2.7 Structures with inequality

As we have seen, equality is only available for those effective structures which are countable. We will discuss some countable structures in the next section. For uncountable structures we can still hope that inequality is semi-recursive and indeed we will see later that this holds in many cases. 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 structures semi-recursiveness of the inequality implies recursiveness of equality, as the following proposition states.

Proposition 3.2.48 (Complete structures with inequality) *Let S be a natural structure and X a complete set over S. Then the equality $=_X$ is finally semi-recursive over S. Thus, the equality $=_X$ is recursive over S, if and only if the inequality $\neq_X$ is semi-recursive over S.*

Proof. Since $=_X$ is equal to $\mathrm{range}((\mathrm{id}_X, \mathrm{id}_X) \circ \Omega_X)$, it follows that $=_X$ is finally semi-recursive if X is complete over S. $\square$

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 3.2.49 *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. By Proposition 3.2.27 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 $y \in X$. $\square$

Last not least we will see that we can also strengthen the graph closure property for functions over perfect structures with inequality.

Proposition 3.2.50 (Graph of functions 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.*

Proof. By Proposition 3.2.35 we already know that $\mathrm{graph}(f)$ is finally semi-recursive if $\mathrm{dom}(f)$ is finally semi-recursive. Therefore, it suffices to prove that $\mathrm{graph}(f)^c$ is semi-recursive. If $\mathrm{dom}(f)$ is recursive over S, then $\mathrm{dom}(f)^c$ is semi-recursive over S. Since $\neq_Y$ is semi-recursive over S, we obtain that $c_{\neq_Y} \circ (f \times \mathrm{id}_Y) :\subseteq X \times Y \rightrightarrows \mathbb{N}$, as well as $c_{\mathrm{dom}(f)^c} : X \rightrightarrows \mathbb{N}$ are recursive over S and since S is perfect they are computable via computable functions $F, F' :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, respectively. Hence, there are computable, total and monotone functions $\varphi, \varphi' : \mathbb{N}^* \to \mathbb{N}^*$ which approximate F, F', respectively. Define $G : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by

$$G\langle\langle p, p'\rangle, q\rangle := \begin{cases} \hat{0} & \text{if } ((\pi_1\nu^*q(0) \sqsubseteq \langle p, p'\rangle \text{ and } 0 \sqsubseteq \varphi\nu^*q(0)) \\ & \text{or } (\pi_1\nu^*q(0) \sqsubseteq p \text{ and } 0 \sqsubseteq \varphi'\nu^*q(0))) \text{ and } q(1) = 0 \\ \hat{1} & \text{else} \end{cases}$$

for all $p, p', q \in \mathbb{N}^{\mathbb{N}}$. Then G is computable and $c_{\mathrm{graph}(f)^c} : X \times Y \rightrightarrows \mathbb{N}$ is computable via G, since

$$(x, y) \in \mathrm{graph}(f)^c \iff x \in \mathrm{dom}(f)^c \text{ or } f(x) \neq y.$$

Since S is perfect, it follows that $\mathrm{graph}(f)^c$ is semi-recursive over S. $\square$

3.2.8 Countable structures

In this section we want briefly discuss countable structures which are commonly used in analysis. We have already introduced the structure of natural numbers $\mathbf{N} = (\mathbb{N}, 0, n, n{+}1)$. Now we define the *structure of integers* $\mathbf{Z}$ and the *structure of rationals* $\mathbf{Q}$ by

$$\boxed{\begin{array}{lcl} \mathbf{Z} & := & \mathbf{N} \oplus (\mathbb{Z}, 0, 1, x+y, -x, x=y) \\ \mathbf{Q} & := & \mathbf{N} \oplus (\mathbb{Q}, 0, 1, x+y, -x, x \cdot y, 1/x, x=y). \end{array}}$$

Here $0, 1$ are constants of the corresponding set, $x+y, -x, x \cdot y, 1/x$ denote the usual arithmetic operations defined on the corresponding set and $x = y$ denotes the equality of the corresponding set. Of course, inversion is considered with its natural domain $\{q \in \mathbb{Q} : q \neq 0\}$. The substructure of natural numbers is needed since the semi-characteristic operation of equality yields a natural number as result. The following theorem is the main theorem of this section.

Theorem 3.2.51 (Discrete structures) *The structures* $\mathbf{N}, \mathbf{Z}, \mathbf{Q}$ *are complete and strongly perfect structures and the equalities* $=_\mathbb{N}$, $=_\mathbb{Z}$, $=_\mathbb{Q}$ *are decidable over the corresponding structures.*

For the proof we will define some further technical notions. In classical theory of effectivity countable sets X are usually treated via *numberings*, that are surjective functions $\nu :\subseteq \mathbb{N} \to X$. Some standard numberings of $\mathbb{Z}$, $\mathbb{Q}$, are defined by $\nu_\mathbb{Z} : \mathbb{N} \to \mathbb{Z}, \langle n, k \rangle \mapsto n - k$ and $\nu_\mathbb{Q} : \mathbb{N} \to \mathbb{Q}, \langle n, k, m \rangle \mapsto \frac{n-k}{m+1}$, respectively.

One can define computability of functions (operations resp.) and effectivity of countable structures w.r.t. numberings correspondingly as we have defined these notions w.r.t. representations. We will not expose these definitions here. Instead of that we will consider numberings as representations. More precisely, with each numbering $\nu :\subseteq \mathbb{N} \to X$ we associate a representation $\delta_\nu :\subseteq \mathbb{N}^\mathbb{N} \to X$, defined by $\delta_\nu(p) := \nu p(0)$ for each $p \in \mathbb{N}^\mathbb{N}$. Obviously, δ is total, if and only if ν is. It is easy to see that a function is computable w.r.t. numberings, if and only if it is computable w.r.t. the corresponding representations (but in general the same does not hold for strong computability). Moreover, if ν is recursive, then so is $\delta_\nu = \nu \circ (\mathrm{id}_{\mathbb{N}^\mathbb{N}})_* \circ (\mathrm{id}_\mathbb{N}^\mathbb{N} \times 0)$ and if ν has a recursive right inverse ν^-, then $\delta^- := [\nu^- \times \mathrm{id}_\mathbb{N}]$ is a recursive right inverse of δ_ν. The following proposition states that in presence of equality recursive numberings with semi-recursive domains always have recursive right inverses.

Proposition 3.2.52 (Inverse numberings) *Let S be a natural structure, let $\nu :\subseteq \mathbb{N} \to X$ be a recursive numbering over S and let* $\mathrm{dom}(\nu)$ *and $=_X$ be semi-recursive over S. Then there is a right inverse $\nu^- : X \to \mathbb{N}$ of ν, which is recursive over S and $=_X$ is even decidable over S.*

Proof. Since $\mathrm{dom}(\nu)$, $=_X$ are semi-recursive over S, so is

$$A := \{(x,n) : n \in \mathrm{dom}(\nu) \text{ and } \nu(n) = x\} \subseteq X \times \mathbb{N}.$$

By the Uniformization Theorem 3.2.40 there is a recursive operation $f : X \rightrightarrows \mathbb{N}$ over S such that $\mathrm{graph}(f) = A$ and by the Selection Theorem 3.2.11 there is a recursive selection $\nu^- : X \to \mathbb{N}$ of f. We obtain

$$x = y \iff \nu^-(x) = \nu^-(y)$$

for all $x, y \in X$. Since $=_{\mathbb{N}}$ is decidable, $=_X$ is decidable over S too. $\square$

One can verify that in the situation of the proposition there is even a bijective numbering of X which is a recursive retraction over S, defined by $\nu' := (\nu^-)^{-1} :\subseteq \mathbb{N} \to X$. Obviously, ν' is a bijective restriction of ν and $\mathrm{dom}(\nu')$ is semi-recursive over S. As a corollary of the previous proposition and the Stability Theorem 3.2.1 we obtain the following stability theorem for countable structures, which is closely related to Mal'cev's Stability Theorem for finitely generated algebras [Mal71, SHT95].

Corollary 3.2.53 (Countable Stability Theorem) *Let $S = \mathbf{N} \oplus R$ be a structure with $R = (X, f_1, ..., f_n, =_X)$, let $\nu :\subseteq \mathbb{N} \to X$ be a recursive numbering over S and let $\mathrm{dom}(\nu)$ be semi-recursive over S. If S is (strongly) effective via a natural representation δ, then S is (strongly) perfect and $\delta \equiv [\delta_{\mathbb{N}}, \delta_\nu]$. Moreover, $=_X$ is decidable over S.*

Now for the proof of Theorem 3.2.51 we only have to investigate the numberings $\nu_{\mathbb{Z}}$ and $\nu_{\mathbb{Q}}$. Since $\nu_{\mathbb{Z}}$ and $\nu_{\mathbb{Q}}$ are total, it is obvious that $\mathrm{dom}(\nu_{\mathbb{Z}})$ and $\mathrm{dom}(\nu_{\mathbb{Q}})$ are semi-recursive (even decidable). Moreover, it is easy to verify that all initial operations of the structure $\mathbf{Z}$, $\mathbf{Q}$, are strongly computable w.r.t. $\delta_{\nu_{\mathbb{Z}}}$, $\delta_{\nu_{\mathbb{Q}}}$, respectively. It remains to prove that $\nu_{\mathbb{Z}}$, $\nu_{\mathbb{Q}}$ are recursive over $\mathbf{Z}$, $\mathbf{Q}$, respectively, which is an easy exercise.

Chapter 4

Recursive Operations over Topological Structures

In this chapter we want to specialize the investigation to recursion over topological structures; that are structures which have a topological space as universe and continuous operations as initial operations. Obviously, such structures are of great importance in analysis. We will concentrate the investigation to separable metric structures, which, on the one hand, have several pleasant properties and which, on the other hand, include a large class of topological spaces. Among the pleasant properties we will find that metric spaces which fulfill certain effectivity conditions give rise to perfect structures.

The chapter will be organized as follows: first we investigate general properties of recursive functions and sets over topological structures. Essentially, we will see that the recursion operators preserve continuity. Then the main part of this chapter follows, which is dedicated to metric structures. Within several subsections we will investigate structures for the real numbers, for compact sets, continuous functions and closed sets. We will emphasize those structures which arise from the investigation of recursive operations and sets but the whole theory can easily be applied to structures which are used in functional analysis and other parts of mathematics. We close this chapter with a section which discusses order-free recursion on the real numbers.

4.1 Recursive operations over topological structures

First we will define 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 4.1.1 (Topological structure) A structure S with universe $X = X_1 \times ... \times X_n$ is called a *topological structure*, if $X_1, ..., X_n$ are equipped with topologies and all initial operations are continuous w.r.t. 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.

We have used product topologies in the definition. If Y, Z are topological spaces, then the *product topology* on $Y \times Z$ is generated by the base

$$\{U \times V : U \subseteq X \text{ open}, V \subseteq Y \text{ open }\}$$

and the *product topology* on $Y^{\mathbb{N}}$ is generated by the base

$$\{\times_{i=0}^{\infty} U_i : U_i \subseteq Y \text{ open and } U_i = Y \text{ for almost all } i \in \mathbb{N}\}.$$

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 with metric space as universe.

Of course, one notion which occurs in the definition of topological structures is still undefined: continuity of operations. We will define this notion as a direct generalization of the corresponding notion for functions.

Definition 4.1.2 (Continuity of operations) Let X, Y be topological spaces. An operation $f :\subseteq X \rightrightarrows Y$ is called *continuous*, if for each open set $V \subseteq Y$ the preimage $f^{-1}(V)$ is open in $\mathrm{dom}(f)$, i.e. there is an open set $U \subseteq X$ such that $f^{-1}(V) = U \cap \mathrm{dom}(f)$.

Obviously, a function $f :\subseteq X \to Y$ considered as operation is continuous, if and only if it is continuous as a function in the usual sense. An operation $f :\subseteq X \rightrightarrows Y$ is continuous, if and only if it is *lower semi-continuous* as a set-valued function in the sense of analysis (cf. [Bee93]). It should be noticed that preimages of continuous operations do not admit the same closure properties as preimages of continuous functions. For instance, they do neither preserve intersections nor complements. Hence, the class of operations such that the preimages of closed sets are closed (called *upper semi-continuous*) is different from the class of (lower semi-)continuous operations in general. Since we do not need upper semi-continuity of operations we well always say “continuous” instead of "lower semi-continuous". Continuous operations have already been investigated in theory of effectivity [BH94, Bra96].

One nice closure property of preimages of operations is that they preserve unions: $f^{-1}(\bigcup_{i=0}^{\infty} U_i) = \bigcup_{i=0}^{\infty} f^{-1}(U_i)$. Thus an operation $f :\subseteq X \rightrightarrows Y$ is continuous, if and only if $f^{-1}(U)$ is open in $\mathrm{dom}(f)$ for all elements U of a base of the topological space Y. We will use this fact for the proof of the following theorem.

Theorem 4.1.3 (Continuity of recursive operations) *All recursive operations over natural topological structures are continuous.*

Initial operations of topological structures are continuous by definition. Hence the proof can be performed by structural induction with the help of the following proposition.

Proposition 4.1.4 *All recursion operators preserve continuity. More precisely, let $\mathbb{N}$ be equipped with the discrete topology, let X, Y, Z, X', Y' be topological spaces and consider induced product topologies on product and sequence spaces.*

(1) If $f :\subseteq X \rightrightarrows Y \times Z$ is a continuous operation, then so are $f_1 :\subseteq X \rightrightarrows Y$ and $f_2 :\subseteq X \rightrightarrows Z$.

(2) If $f :\subseteq X \rightrightarrows Y$ and $g :\subseteq X \rightrightarrows Z$ are continuous operations, then so is $(f, g) :\subseteq X \rightrightarrows Y \times Z$.

(3) If $f :\subseteq X \rightrightarrows Y$ and $g :\subseteq X' \rightrightarrows Y'$ are continuous operations, then so is $f \times g :\subseteq X \times X' \rightrightarrows Y \times Y'$.

(4) If $f :\subseteq X \rightrightarrows Y$ and $g :\subseteq Y \rightrightarrows Z$ are continuous operations, then so is $g \circ f :\subseteq X \rightrightarrows Z$.

(5) If $f :\subseteq X \rightrightarrows X$ is continuous, then so is $f^* :\subseteq X \times \mathbb{N} \rightrightarrows X$.

(6) If $f :\subseteq X \times \mathbb{N} \rightrightarrows Y \times \mathbb{N}$ is continuous, then so is $f^{\leftrightarrow} :\subseteq X \times \mathbb{N} \rightrightarrows Y \times \mathbb{N}$.

(7) If $f :\subseteq X \rightrightarrows Y^{\mathbb{N}}$ is an continuous, then so is $f_* :\subseteq X \times \mathbb{N} \rightrightarrows Y$.

(8) If $f :\subseteq X \times \mathbb{N} \rightrightarrows Y$ is continuous, then so is $[f] :\subseteq X \rightrightarrows Y^{\mathbb{N}}$.

(9) If $f :\subseteq X \rightrightarrows Y$ is continuous, then so is $f^{\mathbb{N}} :\subseteq X^{\mathbb{N}} \rightrightarrows Y^{\mathbb{N}}$.

(10) If $f :\subseteq X \rightrightarrows \mathbb{N}$ is continuous, then so is $f^{\triangle} :\subseteq X \rightrightarrows \mathbb{N}^{\mathbb{N}}$.

Proof. We will use the fact that an operation is continuous if the preimages of base elements are open in the domain of the operation.

(1) Let f be continuous and $U \subseteq Y$ be open. Then

$$f_1^{-1}(U) = f^{-1}(U \times Z)$$

is open in $\mathrm{dom}(f_1) = \mathrm{dom}(f)$. Hence f_1 is continuous.

(2) Let f, g be continuous and let $U \subseteq Y, V \subseteq Z$ be open. Then

$$(f,g)^{-1}(U \times V) = f^{-1}(U) \cap g^{-1}(V)$$

is open in $\mathrm{dom}(f,g) = \mathrm{dom}(f) \cap \mathrm{dom}(g)$. Hence (f,g) is continuous.

(3) Let f, g be continuous and let $U \subseteq Y, V \subseteq Z$ be open. Then

$$(f \times g)^{-1}(U \times V) = f^{-1}(U) \times g^{-1}(V)$$

is open in $\mathrm{dom}(f \times g) = \mathrm{dom}(f) \times \mathrm{dom}(g)$. Hence $f \times g$ is continuous.

(4) Let f, g be continuous, and $V \subseteq Z$ open. Then there is an open $U \subseteq Y$ such that $g^{-1}(V) = U \cap \mathrm{dom}(g)$. Then

$$(g \circ f)^{-1}(V) = f^{-1}g^{-1}(V) = f^{-1}(U \cap \mathrm{dom}(g))$$

Consequently,

$$(g \circ f)^{-1}(V) \cap \mathrm{dom}(g \circ f) = f^{-1}(U) \cap \mathrm{dom}(g \circ f)$$

is open in $\mathrm{dom}(g \circ f) = \{x : f(x) \subseteq \mathrm{dom}(g)\} \subseteq \mathrm{dom}(f)$. Hence $g \circ f$ is continuous.

(5) Let f be continuous. By (4) and induction one can show that f^n is continuous for all $n \in \mathbb{N}$. Let $U \subseteq X$ be open. Then

$$(f^*)^{-1}(U) = \bigcup_{n=0}^{\infty}((f^n)^{-1}(U) \times \{n\})$$

is open in $\mathrm{dom}(f^*) = \bigcup_{n=0}^{\infty}(\mathrm{dom}(f^n) \times \{n\})$. Hence f^* is continuous.

(6) Let f be continuous and let $V \subseteq Y$ be open and $k \in \mathbb{N}$. Then

$$\begin{aligned} & (f^{\leftrightarrow})^{-1}(V \times \{k\}) \\ = \ & \bigcup_{n=0}^{\infty} \left((f^{\leftrightarrow})^{-1}(V \times \{k\}) \cap (X \times \{n\})\right) \\ = \ & \mathrm{dom}(f^{\leftrightarrow}) \cap \bigcup_{n=0}^{\infty} \left(\mathrm{pr}_1 \left(f^{-1}(V \times \{n\}) \cap (X \times \{k\})\right) \times \{n\}\right) \end{aligned}$$

is open in $\mathrm{dom}(f^{\leftrightarrow})$ since $\mathrm{pr}_1(\mathrm{dom}(f^{\leftrightarrow})) \subseteq \mathrm{dom}(f)$. Hence $f^{\leftrightarrow}$ is continuous.

(7) Let f be open and let $V \subseteq Y$ be open. Then

$$f_*^{-1}(V) = \bigcup_{n=0}^{\infty} \left(f^{-1}(Y^n \times V \times Y \times Y \times ...) \times \{n\}\right)$$

is open in $\mathrm{dom}(f_*) = \mathrm{dom}(f) \times \mathbb{N}$. Hence f_* is continuous.

(8) Let f be continuous and let $V_0, ..., V_n \subseteq Y$ be open. Then

$$\begin{aligned} & [f]^{-1}(V_0 \times ... \times V_n \times Y \times Y \times ...) \\ = \;& \mathrm{pr}_1 \left(\bigcup_{i=0}^{n} \left(f^{-1}(V_i) \cap (X \times \{i\})\right) \right) \cap \mathrm{dom}([f]) \end{aligned}$$

is open in $\mathrm{dom}[f]$ since $\mathrm{dom}[f] \times \mathbb{N} \subseteq \mathrm{dom}(f)$. Hence $[f]$ is continuous.

(9) Let f be continuous and let $V_0, ..., V_n \subseteq Y$ be open. Then

$$\begin{aligned} & (f^{\mathbb{N}})^{-1}(V_0 \times ... \times V_n \times Y \times Y \times ...) \\ = \;& f^{-1}(V_0) \times ... \times f^{-1}(V_n) \times \mathrm{dom}(f) \times \mathrm{dom}(f) \times ... \end{aligned}$$

is open in $\mathrm{dom}(f^{\mathbb{N}}) = (\mathrm{dom}(f))^{\mathbb{N}}$. Hence $f^{\mathbb{N}}$ is continuous.

(10) Let f be continuous and let $w = y_0...y_n \in \mathbb{N}^*$. Then

$$(f^{\triangle})^{-1}(w\mathbb{N}^{\mathbb{N}}) = f^{-1}\{y_0\} \cap ... \cap f^{-1}\{y_n\}$$

is open in $\mathrm{dom}(f^{\triangle}) = \mathrm{dom}(f)$. Hence $f^{\triangle}$ is continuous.

□

4.2 Recursive sets over topological structures

In this section we want to investigate 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 continuous, if and only if A is open. Thus we can immediately deduce some facts about (semi-)recursive sets over topological structures.

Corollary 4.2.1 (Recursive sets over topological structures) *Over natural topological structures semi-recursive subsets are open, recursive subsets are closed and decidable subsets are open and closed.*

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

Corollary 4.2.2 (Topological structures with (in)equality) *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 by Proposition 3.2.42 that X is even a countable discrete space. Moreover we can conclude that there are no non-trivial decidable subsets over connected topological spaces since decidable subsets are both: open and closed.

Corollary 4.2.3 (Decidable subsets over connected spaces) *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 $A = X$.*

Over topological structures we have the possibility to strengthen the notion of recursiveness. Since the closure of the complement of a recursive set need not to be recursive and this operation can be iterated once, we get the following stronger versions. Here, we will use the notation A° for the *interior*, $\overline{A}$ for the *closure* and ∂A for the *border* of a set A.

Definition 4.2.4 (Birecursive and double birecursive subsets) Let S be a natural topological structure, X a topological space over S and $A \subseteq X$ a subset. Then A is called *birecursive over S*, if A and $\overline{A^c}$ are recursive over S. Moreover, A is called *double birecursive over S*, if A and $\overline{A^c}$ are birecursive over S.

It is a surprising fact of topology that the operations "closure" and "complement" allow to construct at most 14 different sets, starting from an arbitrary subsets of a topological space (cf. [Kur66]). As a consequence $\overline{\overline{\overline{A^c}}^c}^c = \overline{A^c}$ holds for each closed set A and for each double birecursive A the set $\overline{A^c}$ is double birecursive too.

The class of decidable sets over a connected topological structure is trivial. But over disconnected structures, such as Baire's space $\mathbb{N}^\mathbb{N}$, it makes sense to ask how decidable and recursive sets are related. The following proposition answers the question.

Proposition 4.2.5 (Decidable and recursive sets) *Let S be a natural complete topological structure, X a topological space over S and $A \subseteq X$. Then A is decidable over S, if and only if A is birecursive over S and $\partial A = \emptyset$.*

Proof. Let A be decidable over S. Then A, A^c are both open and closed and thus $\partial A = \emptyset$ and $\overline{A^c} = A^c$. Since A is decidable, it follows that A, A^c are semi-recursive over S and thus recursive over S by Proposition 3.2.39, since S is complete. Altogether, A is birecursive over S.

Now let A be birecursive and $\partial A = \emptyset$. Again $\overline{A^c} = A^c$ and A^c, A are recursive. Especially, A, A^c are semi-recursive over S and thus by Proposition 3.2.34 A is decidable over S. □

4.3 Perfect topological structures

In this section we want to investigate some consequences of the previous results w.r.t. perfect topological structures. A strongly perfect structure, especially, is strongly recursive, i.e. it admits a representation which is a recursive retraction. Consequently, the representation is continuous and admits a continuous right inverse operation. But a surjective function $f :\subseteq X \to Y$ is continuous and admits a continuous right inverse operation $f^- : Y \rightrightarrows X$, if and only if it is continuous and admits a surjective and open restriction $f' :\subseteq X \to Y$. Thus, representations which are recursive retractions over topological structures are topologically *admissible* in the sense of theory of effectivity (cf. [Wei87]). Following this idea we can prove a characterization of topological structures with continuous representations which admit a recursive right inverse. For the proof we will use a further definition: if X is a T_0-space with countable base $\{B_n : n \in \mathbb{N}\}$, then the representation $\delta :\subseteq \mathbb{N}^\mathbb{N} \to X$, defined by

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

is called *standard representation* of X w.r.t. $(B_n)_{n\in\mathbb{N}}$ (cf. [Wei87]).

Proposition 4.3.1 (Recursive topological structures) *Let S be a natural T_0-structure with universe X. Then there is a continuous representation δ of X with a recursive right inverse over S, if and only if there is a recursively given sequence of semi-recursive sets $(B_n)_{n\in\mathbb{N}}$ over S such that $\{B_n : n \in \mathbb{N}\}$ is a base of X.*

Proof. Let δ be a representation of X with a recursive right inverse $\delta^- : X \rightrightarrows \mathbb{N}^\mathbb{N}$ over S. Then $\{(\delta^-)^{-1}(w\mathbb{N}^\mathbb{N}) : w \in \mathbb{N}^*\}$ is a base of X since δ, δ^- are continuous. Now let $w_n := \nu^*(n)$ for all $n \in \mathbb{N}$, then $(w_n\mathbb{N}^\mathbb{N})_{n\in\mathbb{N}}$ is a recursively given sequence of semi-recursive sets over $\mathbf{N}$, since $f :\mathbb{N}^{\mathbb{N}} \times \mathbb{N} \rightrightarrows \mathbb{N}$, defined by

$$f(p,n) := \begin{cases} \{0,1\} & \text{if} \quad w_n \sqsubseteq p \\ \{1\} & \text{else} \end{cases}$$

is recursive over $\mathbf{N}$ and $f(p,n) = c_{w_n\mathbb{N}^{\mathbb{N}}}(p)$ for all $n \in \mathbb{N}$, $p \in \mathbb{N}^{\mathbb{N}}$. Since δ^- is recursive over S we can conclude that $(B_n)_{n\in\mathbb{N}}$ with $B_n := (\delta^-)^{-1}(w_n\mathbb{N}^{\mathbb{N}})$ is a recursively given sequence of semi-recursive sets over S and $\{B_n : n \in \mathbb{N}\}$ is a base of X.

Now let $(B_n)_{n\in\mathbb{N}}$ be a recursively given sequence of semi-recursive sets over S such that $\{B_n : n \in \mathbb{N}\}$ is a base of X. Then there is a recursive operation $f : X \times \mathbb{N} \rightrightarrows \mathbb{N}$ over S such that $f(x,n) = c_{B_n}(x)$ for all $x \in X$, $n \in \mathbb{N}$. Let $\delta :\subseteq \mathbb{N}^{\mathbb{N}} \to X$ be the standard representation of X w.r.t. $(B_n)_{n\in\mathbb{N}}$. Then δ is continuous. The operation $g : X \rightrightarrows \mathbb{N}$, defined by $g(x) = f^{\leftrightarrow}(x,0)$, is recursive over S and

$$\begin{aligned} g^{\triangle}(x) &= \{p \in \mathbb{N}^{\mathbb{N}} : \mathrm{range}(p) = g(x)\} \\ &= \{p \in \mathbb{N}^{\mathbb{N}} : \mathrm{range}(p) = \{n : 0 \in f(x,n)\}\} \\ &= \{p \in \mathbb{N}^{\mathbb{N}} : \mathrm{range}(p) = \{n : x \in B_n\}\} \\ &= \delta^{-1}\{x\} \end{aligned}$$

for all $x \in X$. Thus $g^{\triangle}$ is a recursive right inverse of δ over S. $\square$

The T_0-property has only been used for the "if"-direction of the proof and thus we get the following corollary.

Corollary 4.3.2 (Countability of recursive topological structures) *If S is a recursive topological structure with universe X, then the topology of X has a countable base.*

Since we are mainly interested in perfect topological structures we have to restrict ourselves to topological spaces with countable bases. We obtain the following standard structure for such spaces.

Theorem 4.3.3 (Perfect topological structures) *If X is a T_0-space with countable base $\{B_n : n \in \mathbb{N}\}$, standard representation δ, and if there is some computable $p \in \mathrm{dom}(\delta)$, then $\mathbf{N} \oplus (X, \mathrm{id}_X, \delta, (B_n)_{n\in\mathbb{N}})$ is a perfect topological structure.*

Proof. By the proof of the previous proposition δ admits a recursive right inverse over $S := \mathbf{N} \oplus (X, \mathrm{id}_X, \delta, (B_n)_{n\in\mathbb{N}})$. Thus S is recursive via $[\delta_{\mathbb{N}}, \delta]$. Moreover, S is a natural topological structure, since δ is continuous and the sets B_n are open. We have to prove that S is effective via $[\delta_{\mathbb{N}}, \delta]$. Obviously, it suffices to show that $f : X \times \mathbb{N} \rightrightarrows \mathbb{N}$ with $f(x,n) = c_{B_n}(x)$ is $([\delta, \delta_{\mathbb{N}}], \delta_{\mathbb{N}})$–computable. We define $F : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by

$$F\langle\langle p, r\rangle, q\rangle := \begin{cases} \hat{0} & \text{if} \quad pq(0) = r(0) \text{ and } q(1) = 0 \\ \hat{1} & \text{else} \end{cases}$$

for all $p, q, r \in \mathbb{N}^{\mathbb{N}}$. Then F is computable, $1 \in \{\delta_{\mathbb{N}} F\langle\langle p, \hat{n}\rangle, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\}$ and

$$\begin{aligned} 0 \in \{\delta_{\mathbb{N}} F\langle\langle p, \hat{n}\rangle, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\} \quad &\iff \quad (\exists k) p(k) = n \\ &\iff \quad \delta(p) \in B_n \\ &\iff \quad 0 \in c_{B_n}(\delta(p)) \end{aligned}$$

for all $p \in \mathrm{dom}(\delta)$, $n \in \mathbb{N}$. Hence, f is $([\delta, \delta_{\mathbb{N}}], \delta_{\mathbb{N}})$–computable via F. $\square$

It should be emphasized that for the proofs of the results of this section we have used the sequentialization operator.

4.4 Metric structures

In this section we will specialize the investigation to separable metric structures which are very fruitful since they canonically offer functions which can be taken as initial operations: the metric and the limit. In case the metric fulfills a natural effectivity condition, the metric space will be called recursive and we will see that recursive metric spaces give rise to canonical perfect structures. Recursive metric spaces have already been investigated by several authors in recursive analysis: Lacombe has investigated recursive complete separable metric spaces [Lac59] from a classical point of view, in the russian school of constructive analysis Ceitin [Cei62], Šanin [Šan68], and Kushner [Kus84] have investigated metric spaces. Similarly, Moschovakis [Mos64a, Mos64b] has investigated recursive metric spaces restricted to computable points. In theory of effectivity Weihrauch [Wei93] studied computable metric spaces, in the domain representation approach Blanck investigated them [Bla97b, Bla97a, Bla99] and the Pour-El and Richards approach to computable analysis has been extended to metric spaces by Mori, Tsujii, Yasugi, Yoshiki and Washihara [MTY97, YMT99, WY96]. All the definitions used for recursive/computable metric spaces are quite similar to the definition we will present. We start with an investigation of a special case, with the structure of the real numbers.

4.4.1 The structure of real numbers

In this section we want to investigate the structure of the real numbers

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

As usual, 0 and 1 denote the corresponding real constants, $+, -, \cdot, /$ the usual arithmetic operations on the real numbers with their natural domains (the identity can be constructed since $\mathrm{id}_{\mathbb{R}}(x) = -(-x)$). Following our convention, $<$ stands for the semi-characteristic operation of the usual order relation:

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

Last not least, Lim denotes the *limit operator* $\mathrm{Lim} :\subseteq \mathbb{R}^n \to \mathbb{R}, (x_n)_{n \in \mathbb{N}} \mapsto \lim_{n \to \infty} x_n$ restricted to rapidly converging sequences:

$$\mathrm{dom}(\mathrm{Lim}) := \{(x_n)_{n \in \mathbb{N}} : (\forall n > k)|x_n - x_k| \leq 2^{-k}\}.$$

The restriction to rapidly converging sequences is a well-known technique in computable analysis and can be motivate by the fact that the non-restricted limit operator is discontinuous while the restricted one is continuous. If we equip $\mathbb{R}$ with the usual Euclidean topology, induced by the metric $d_{\mathbb{R}} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}, (x, y) \mapsto |x - y|$, then $\mathbf{R}$ obviously is a topological structure. As a first result we will show that some elementary functions are recursive over $\mathbf{R}$.

Proposition 4.4.1 *The following functions with their natural domains are recursive over* $\mathbf{R}$*:*

(1) $\alpha_{\mathbb{R}} : \mathbb{N} \to \mathbb{R}, \langle n, k, m \rangle \mapsto \dfrac{n - k}{m + 1}$,

(2) $s :\subseteq \mathbb{R} \to \mathbb{R}, x \mapsto \sqrt{x}$,

(3) $|\ | : \mathbb{R} \to \mathbb{R}, x \mapsto |x|$,

(4) $d_{\mathbb{R}} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}, (x, y) \mapsto |x - y|$,

(5) $\max : \mathbb{R} \times \mathbb{R} \to \mathbb{R}, (x, y) \mapsto \max\{x, y\}$,

(6) $\sigma : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}, (x_n)_{n \in \mathbb{N}} \mapsto \displaystyle\sum_{i=0}^{\infty} 2^{-i-1} \frac{x_i}{1 + |x_i|}$.

More precisely, $\mathrm{dom}(s) = \{x : x \geq 0\}$.

Proof.

(1) It is easy to prove that $\alpha_{\mathbb{R}}$ is recursive over $\mathbf{R}$.

(2) We will use the Heron algorithm to prove that the square root function s is recursive over $\mathbf{R}$. The Heron iteration function $h :\subseteq \mathbb{R} \times \mathbb{N} \to \mathbb{R}$, defined by

$$\begin{cases} h(x,0) & := \ 1 \\ h(x,n+1) & := \ \frac{1}{2}\left(h(x,n) + \frac{x}{h(x,n)}\right) \end{cases}$$

for all $x \in \mathbb{R}, n \in \mathbb{N}$ is recursive over $\mathbf{R}$. By a well-known a posteriori error estimation

$$0 \leq h(x,n+2) - \sqrt{x} \leq h(x,n+1) - h(x,n+2)$$

for all $x \geq 0$, $n \in \mathbb{N}$. Especially, $\lim_{n\to\infty} h(x,n) = \sqrt{x}$ for all $x \geq 0$. Now

$$A := \{(x,k,n) : h(x,n+1) - h(x,n+2) < 2^{-k-1}\} \subseteq \mathbb{R} \times \mathbb{N} \times \mathbb{N}$$

is semi-recursive over $\mathbf{R}$ and thus by the Uniformization Theorem 3.2.40 there is a recursive operation $f : \mathbb{R} \times \mathbb{N} \rightrightarrows \mathbb{N}$ over $\mathbf{R}$ with $\mathrm{graph}(f) = A$. Then $g : \mathbb{R} \times \mathbb{N} \rightrightarrows \mathbb{N}$, defined by $g(x,k) := h(x, f(x,k) + 2)$ is recursive over $\mathbf{R}$ and so is $s = \mathrm{Lim} \circ [g]|_{\mathrm{dom}(s)}$ since $\mathrm{dom}(s) = [0,\infty)$ is a domain over the structure $\mathbf{R}$.

(3) The absolute value function $|\ |$ is recursive over $\mathbf{R}$, since $|x| = \sqrt{x^2}$.

(4) The metric $d_{\mathbb{R}}$ is recursive over $\mathbf{R}$, since $d_{\mathbb{R}}(x,y) = |x-y|$.

(5) The maximum function max is recursive over $\mathbb{R}$, since $\max(x,y) = \frac{1}{2}(x + y + |x-y|)$.

(6) Since the evaluation $\mathrm{ev}_{\mathbb{R}^{\mathbb{N}}} := (\mathrm{id}_{\mathbb{R}}^{\mathbb{N}})_*$ is recursive over $\mathbf{R}$, it follows that the partial sum function $s : \mathbb{R}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{R}$, defined by

$$s((x_n)_{n\in\mathbb{N}}, k) := \sum_{i=0}^{k} 2^{-i-1} \frac{x_i}{1 + |x_i|}$$

is recursive over $\mathbf{R}$ too. Thus $\sigma = \mathrm{Lim} \circ [s]$ is recursive over $\mathbf{R}$.

□

Moreover, it is easy to see, that $\neq_{\mathbb{R}}$ is semi-recursive over $\mathbf{R}$, since $c_{\neq_{\mathbb{R}}}(x,y) = c_<(x,y) \cdot c_<(y,x)$. Our goal is it to prove the following essential property of the structure of the real numbers. For this we will use the *Cauchy representation* $\delta_{\mathbb{R}} := \mathrm{Lim} \circ \alpha_{\mathbb{R}}^{\mathbb{N}}$ of the real numbers.

Theorem 4.4.2 (The structure of the real numbers) *The structure* $\mathbf{R}$ *is a strongly perfect complete topological structure and the inequality* $\neq_{\mathbb{R}}$ *is recursive over* $\mathbf{R}$*. The Cauchy representation yields a standard representation of this structure.*

More precisely, $[\delta_{\mathbb{N}}, \delta_{\mathbb{R}}]$ is a standard representation of $\mathbf{R}$. As a direct consequence of this theorem 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 [Grz55, Lac55]).

Corollary 4.4.3 *The recursive functions* $f : \mathbb{R}^n \to \mathbb{R}$ *over* $\mathbf{R}$ *are exactly the classically computable real functions.*

Now it remains to prove Theorem 4.4.2. The definition of $\delta_{\mathbb{R}}$ already shows that $\delta_{\mathbb{R}}$ is recursive over $\mathbf{R}$. It is well-known and easy to prove that the arithmetic operation $+, -, \cdot, /$ are computable w.r.t. the Cauchy representation (cf. Wei87,Wei95). We will postpone the rest of the proof until the next two sections where we will give it in the more general context of metric spaces. Here we will only show that $c_<$ is computable w.r.t. the Cauchy representation.

Proposition 4.4.4 *The semi-characteristic operation* $c_< : \mathbb{R} \times \mathbb{R} \rightrightarrows \mathbb{N}$ *of the order* $<$ *is* $([\delta_{\mathbb{R}}, \delta_{\mathbb{R}}], \delta_{\mathbb{N}})$*–computable.*

Proof. Define $F : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by

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

for all $p, p', q \in \mathbb{N}^{\mathbb{N}}$. Then F is computable since comparisons w.r.t. $\alpha_{\mathbb{R}}$ are decidable. Then $1 \in \{\delta_{\mathbb{N}} F\langle\langle p, p'\rangle, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\}$ and

$$\begin{aligned} & 0 \in \{\delta_{\mathbb{N}} F\langle\langle p, p'\rangle, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\} \\ \iff & (\exists k) \alpha_{\mathbb{R}} p(k) < \alpha_{\mathbb{R}} p'(k) - 2^{-k+1} \\ \iff & \delta_{\mathbb{R}}(p) < \delta_{\mathbb{R}}(p') \\ \iff & 0 \in c_<(\delta_{\mathbb{R}}(p), \delta_{\mathbb{R}}(p')), \end{aligned}$$

i.e. $c_<$ is $([\delta_{\mathbb{R}}, \delta_{\mathbb{R}}], \delta_{\mathbb{N}})$–computable via F. $\square$

At the end of this section we shortly mention the structure of complex numbers $\mathbb{C}$. From the topological point of view as well as from the computational point of view there is no substantial difference between $\mathbb{R}^2$ and the complex numbers $\mathbb{C}$. Consequently, we can consider the structure $\mathbf{C} := \mathbf{R} \oplus (\mathbb{C}, \mathrm{in}, \mathrm{Re}, \mathrm{Im})$, where $\mathrm{in} : \mathbb{R}^2 \to \mathbb{C}, (x, y) \mapsto x + iy$ denotes the canonical bijection, and $\mathrm{Re}, \mathrm{Im} : \mathbb{C} \to \mathbb{R}$ the usual projections on the real/imaginary part of a complex number.

4.4.2 Recursive metric spaces

In this section we will study the general case of recursive metric spaces. Since we have seen that each recursive topological structure necessarily has a countable base, we only have to treat the case of separable metric spaces. By $B(x, \varepsilon) := \{y \in X : d(x, y) < \varepsilon\}$ we will denote the *open balls* of a metric space (X, d). These balls form a base of the topology induced by d and $d : X \times X \to \mathbb{R}$ is always a continuous mapping w.r.t. to this topology on X and the Euclidean topology on $\mathbb{R}$. Analogously, the metric will be used to "transfer" recursiveness from $\mathbb{R}$ to X. We will say that a sequence $\alpha : \mathbb{N} \to X$ is *dense* in X, if $\mathrm{range}(\alpha)$ is dense in X.

Definition 4.4.5 (Recursive metric space) We will call a triple (X, d, α) a *recursive 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 recursive over $\mathbf{R}$.

If (X, d, α) fulfills (1) and (2), then it is called a *separable metric space*.

For all following proofs about recursive metric spaces it would suffice if α is partial and $\mathrm{dom}(\alpha)$ is semi-recursive over $\mathbb{N}$. But we will not need this general case in the following. It is easy to see that $(\mathbb{R}, d_{\mathbb{R}}, \alpha_{\mathbb{R}})$ with $d_{\mathbb{R}}, \alpha_{\mathbb{R}}$ as in Proposition 4.4.1 is a recursive metric space.

With each separable metric space (X, d, α) we associate the *standard structure*

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

where $\mathbf{R}$ denotes the structure of the real numbers, id is the identity on X and $\mathrm{Lim} :\subseteq X^n \to X, (x_n)_{n\in\mathbb{N}} \mapsto \lim_{n\to\infty} x_n$ is the *limit operator* of (X, d), restricted to rapidly converging sequences:

$$\mathrm{dom}(\mathrm{Lim}) := \{(x_n)_{n\in\mathbb{N}} : (\forall n > k) d(x_n, x_k) \leq 2^{-k} \text{ and } (x_n)_{n\in\mathbb{N}} \text{ converges } \}.$$

The only difference to the limit operation on the real numbers is that not each rapidly converging Cauchy sequence has to have a limit in X, since the space (X, d) is not necessarily complete. Sometimes we will also write Lim_X to express that we consider the limit of space X. By abuse of notation we write X for the set, as well as for the separable metric space (X, d, α), provided α, d are fixed or out of consideration.

It is straightforward to prove that the recursive sequences $s : \mathbb{N} \to X$ over the standard structure of a recursive metric space X form a sequential computability structure of a metric space in the sense of Mori, Tsujii, Yoshiki, and Yasugi [MTY97, YMT99].

If we consider the standard structure of the recursive metric space $(\mathbb{R}, d_{\mathbb{R}}, \alpha_{\mathbb{R}})$, then we obtain two "copies" of the real numbers. It is easy to see that the second prestructure with the set of real numbers is redundant but we will not prove this here.

It is easy to see that each recursive metric space inherit semi-recursiveness of the inequality $\neq_X$ from $\mathbb{R}$. Since $d(x, y) = 0 \iff x = y$ we obtain $c_{\neq_X}(x, y) = c_<(0, d(x, y))$ for all x, y. Now we can formulate the main result about recursive metric spaces. We will use the *Cauchy representation* $\delta := \mathrm{Lim} \circ \alpha^{\mathbb{N}}$ of a recursive metric space (X, d, α).

Theorem 4.4.6 (Recursive metric spaces) *If X is a recursive metric space, then its standard structure $\mathbf{X}$ is a perfect topological structure and the inequality $\neq_X$ is semi-recursive over $\mathbf{X}$. The Cauchy representation yields a standard representation of this structure.*

More precisely, if δ is the Cauchy representation of X, then $[\delta_{\mathbb{N}}, \delta_{\mathbb{R}}, \delta]$ is a standard representation of $\mathbf{X}$. We have to show that δ is a recursive retraction over $\mathbf{X}$ and that $\mathbf{X}$ is effective via $[\delta_{\mathbb{N}}, \delta_{\mathbb{R}}, \delta]$. The first property is provided by the following proposition.

Proposition 4.4.7 (Cauchy sequence representation) *The Cauchy sequence representation δ of a separable metric space X is a recursive retraction over its standard structure $\mathbf{X}$.*

Proof. Since Lim, α are recursive over $\mathbf{X}$, δ is recursive over $\mathbf{X}$ too. Thus, $A := \{(x, k, n) \in X \times \mathbb{N} \times \mathbb{N} : d(\alpha(n), x) < 2^{-k-1}\}$ is semi-recursive over $\mathbf{X}$ and by uniformization there is a recursive operation $f : X \times \mathbb{N} \rightrightarrows \mathbb{N}$ such that $\mathrm{graph}(f) = A$. Here f is total since $\mathrm{range}(\alpha)$ is dense in X. Hence, $\delta^- := [f] : X \rightrightarrows \mathbb{N}^{\mathbb{N}}$ is recursive over $\mathbf{X}$. Moreover, δ^- is a right inverse of δ, since $p \in \delta^-(x)$ implies $d(\alpha p(k), x) < 2^{-k-1}$, i.e. $(\forall k)(\forall n > k) d(\alpha p(k), \alpha p(n)) < 2^{-k}$ and $\delta(p) = x$. $\square$

Now we have to prove that the standard structure of a recursive metric space is effective via its Cauchy representation. We will prove more than this and give characterizations of recursive metric spaces and their Cauchy representations (cf. [Wei93]).

Lemma 4.4.8 (Characterization of recursive metric spaces) *Let (X, d, α) be a separable metric space. Then (X, d, α) is a recursive metric space, if and only if the sets*

$$D_< := \{(i, j, k) \in \mathbb{N}^3 : d(\alpha(i), \alpha(j)) < \alpha_{\mathbb{R}}(k)\},$$

$$D_> := \{(i, j, k) \in \mathbb{N}^3 : d(\alpha(i), \alpha(j)) > \alpha_{\mathbb{R}}(k)\},$$

are semi-recursive over $\mathbf{N}$.

Proof. Let (X, d, α) be a recursive metric space. Then $d \circ (\alpha \times \alpha)$ is recursive over $\mathbf{R}$. Since

$$c_{D_<}(i, j, k) = c_<(d(\alpha(i), \alpha(j)), \alpha_{\mathbb{R}}(k))$$

for all $i, j, k \in \mathbb{N}$, it follows that $D_<$ is semi-recursive over $\mathbf{R}$. Analogously, one can show that $D_>$ is semi-recursive over $\mathbf{R}$. Since $\mathbf{R}$ is perfect by Theorem 4.4.2 we can conclude by the Conservation Theorem 3.2.8 that $D_<, D_>$ are recursive over $\mathbf{N}$.

Now let $D_<, D_>$ be semi-recursive over $\mathbf{N}$. Then the set

$$A := \{(i, j, k, n) : \alpha_{\mathbb{R}}(n) < d(\alpha(i), \alpha(j)) < \alpha_{\mathbb{R}}(n) + 2^{-k-1}\}$$

is also semi-recursive over $\mathbf{N}$. By uniformization and selection there is a recursive function $f : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ over $\mathbf{N}$ such that $\mathrm{graph}(f) \subseteq A$. Then $d \circ (\alpha \times \alpha) = \mathrm{Lim}_{\mathbb{R}} \circ \alpha_{\mathbb{R}}^{\mathbb{N}} \circ [f]$ is recursive over $\mathbf{R}$ (where $\mathrm{Lim}_{\mathbb{R}} :\subseteq \mathbb{R}^{\mathbb{N}} \to \mathbb{R}$ is the limit operator of $\mathbb{R}$) and (X, d, α) is a recursive metric space. $\square$

The previous lemma already uses the unproved Theorem 4.4.2. For a proper "bootstrap proof" we have to mention that the lemma can be proved directly in case of the real numbers, since the Euclidean distance of two rational numbers is a rational numbers again which can be easily computed (the sets $D_<$ and $D_>$ are even decidable in this case).

The following characterization of the Cauchy representation illustrates that the limit operation Lim can be used to "synthesize" and the metric d can be used to "analyze" a metric space.

Proposition 4.4.9 (Characterization of the Cauchy representation) *Let (X, d, α) be a recursive metric space with Cauchy representation δ and let δ' be a further representation of X. Then $\alpha : \mathbb{N} \to X$ is $(\delta_{\mathbb{N}}, \delta)$–computable and*

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

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

Proof. Define $F : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by $F\langle p, q\rangle(k) := p(0)$. Then F is computable and $\{\delta F\langle p, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\} = \{\mathrm{Lim} \circ \alpha^{\mathbb{N}} \circ F\langle p, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\} = \alpha p(0) = \alpha\delta_{\mathbb{N}}(p)$ for all $p \in \mathbb{N}^{\mathbb{N}}$. Thus, α is $(\delta_{\mathbb{N}}, \delta)$–computable via F.

(1) Let $\delta' \leq \delta$. It suffices to prove that d is $([\delta, \delta], \delta_{\mathbb{R}})$–computable. By Lemma 4.4.8 we can see that the set

$$A := \{(p, p', k, n) : \alpha_{\mathbb{R}}(n) < d(\alpha p(k+3), \alpha p'(k+3)) < \alpha_{\mathbb{R}}(n) + 2^{-k-2}\}$$

is semi-recursive over $\mathbf{N}$. By uniformization and selection we can show that there is a recursive function $f : \mathbb{N}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{N}$ over $\mathbf{N}$ such that $\mathrm{graph}(f) \subseteq A$. Then $F : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, defined by $F\langle\langle p, p'\rangle, q\rangle(k) := f(p, p', k)$ for all $p, p', q \in \mathbb{N}^{\mathbb{N}}$, $k \in \mathbb{N}$ is recursive over $\mathbf{N}$ and

$$F\langle\langle p, p'\rangle, q\rangle(k) = n \Longrightarrow (p, p', k, n) \in A$$

for all $p, p', q \in \mathbb{N}^{\mathbb{N}}, k, n \in \mathbb{N}$. Now let $p, p' \in \mathrm{dom}(\delta)$ and $x := \delta(p)$, $y := \delta(p')$ and $n_k := F\langle\langle p, p'\rangle, q\rangle(k)$ for all $k \in \mathbb{N}$. Then $|d(x, y) - \alpha_{\mathbb{R}}(n_k)| \leq 2^{-k-1}$ and thus

$$\delta_{\mathbb{R}} F\langle\langle p, p'\rangle, q\rangle = \lim_{k\to\infty} \alpha_{\mathbb{R}}(n_k) = d(x, y) = d \circ [\delta, \delta]\langle p, p'\rangle.$$

Thus, d is $([\delta, \delta], \delta_{\mathbb{R}})$–computable via F.

Now let d be $([\delta', \delta], \delta_{\mathbb{R}})$–computable. Since α is $(\delta_{\mathbb{N}}, \delta)$–computable, there is a computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ 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 $\delta'(p) = \mathrm{Lim} \circ \alpha^{\mathbb{N}} \circ F(p) = \delta F(p)$ for all $p \in \mathrm{dom}(\delta')$, i.e. $\delta' \leq \delta$.

(2) Let $\delta \leq \delta'$. It suffices to prove that Lim is (δ^∞, δ)–computable.

Define $F : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by $F\langle\langle p_0, p_1, ...\rangle, q\rangle(k) := p_{k+2}(k+2)$. Then F is computable. Let $\langle p_0, p_1, ...\rangle \in \mathrm{dom}(\delta^\infty)$, i.e. $(\forall n > k) d(\delta(p_n), \delta(p_k)) \leq 2^{-k}$ and $(\delta(p_n))_{n\in\mathbb{N}}$ converges in (X, d). Especially, $p_n \in \mathrm{dom}(\delta)$ for all $n \in \mathbb{N}$ and $(\forall i > j) d(\alpha p_n(i), \alpha p_n(j)) \leq 2^{-j}$. Thus $(\forall n > k) d(\alpha p_{n+2}(n+2), \alpha p_{k+2}(k+2)) \leq 2^{-k}$ follows by triangle inequality. We obtain

$$\begin{aligned}
\{\delta F\langle\langle p_0, p_1, ...\rangle, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\} &= \lim_{k\to\infty} \alpha p_{k+2}(k+2) \\
&= \lim_{n\to\infty} \lim_{k\to\infty} \alpha p_n(k) \\
&= \lim_{n\to\infty} \delta(p_n) \\
&= \mathrm{Lim} \circ \delta^\infty\langle p_0, p_1, ...\rangle
\end{aligned}$$

for all $p \in \mathbb{N}^{\mathbb{N}}$. Consequently, Lim is (δ^∞, δ)–computable via F.

Now let Lim be (δ^∞, δ')–computable. Since α is $(\delta_{\mathbb{N}}, \delta)$–computable and $\delta = \mathrm{Lim} \circ \alpha^{\mathbb{N}}$, it follows that δ is $(\delta_{\mathbb{N}}^\infty, \delta')$–computable. Since $\delta_{\mathbb{N}}^\infty \equiv \mathrm{id}_{\mathbb{N}^{\mathbb{N}}}$, this implies $\delta \leq \delta'$.

□

This characterization of the Cauchy sequence representation δ especially shows that the standard structure of a recursive metric space is effective via $[\delta_{\mathbb{N}}, \delta_{\mathbb{R}}, \delta]$. Here the identity is obviously computable (w.r.t. each representation). Thus Theorem 4.4.6 is proved.

4.4.3 Complete recursive metric spaces

Now we want to strengthen our main theorem on recursive metric spaces 4.4.6 for the case of complete recursive metric spaces.

Theorem 4.4.10 (Complete recursive metric spaces) *If X is a complete recursive metric space, then the standard structure $\mathbf{X}$ is a complete and strongly perfect topological structure and equality $=_X$ is recursive over $\mathbf{X}$.*

We start to prove the following proposition which states that the limit operator of a complete separable metric space has a useful total extension.

Proposition 4.4.11 (Limit operator) *If (X, d, α) is a complete recursive metric space, then there is a total recursive operation $L : X^{\mathbb{N}} \rightrightarrows X$ over $\mathbf{X}$ such that $L|_{\mathrm{dom(Lim)}} = \mathrm{Lim}$ and $y \in L(x_n)_{n\in\mathbb{N}}$ implies $y = \lim_{n\to\infty} x_n$ or there is a $k \in \mathbb{N}$ such that $y = x_k$ for all $(x_n)_{n\in\mathbb{N}} \in X^{\mathbb{N}}$.*

Proof. Since X is complete we obtain

$$\mathrm{dom(Lim)} = \{(x_n)_{n\in\mathbb{N}} : (\forall n > k)\ d(x_n, x_k) \leq 2^{-k}\}.$$

The sets

$$\begin{aligned} A_0 &:= \{((x_n)_{n\in\mathbb{N}}, m) \in \mathbb{N}^{\mathbb{N}} \times \mathbb{N} : (\forall n, k)\ k < n \leq m \Longrightarrow d(x_n, x_k) < 2^{-k+1}\} \\ A_1 &:= \{((x_n)_{n\in\mathbb{N}}, m) \in \mathbb{N}^{\mathbb{N}} \times \mathbb{N} : (\exists n, k)\ k < n \leq m \text{ and } d(x_n, x_k) > 2^{-k}\} \end{aligned}$$

are semi-recursive over $\mathbf{X}$ and thus $g : X^{\mathbb{N}} \times \mathbb{N} \rightrightarrows \mathbb{N}$, defined by

$$g((x_n)_{n\in\mathbb{N}}, m) := \begin{cases} \{0\} & \text{if} \quad ((x_n)_{n\in\mathbb{N}}, m) \in A_0 \setminus A_1 \\ \{1\} & \text{if} \quad ((x_n)_{n\in\mathbb{N}}, m) \in A_1 \setminus A_0 \\ \{0, 1\} & \text{else} \end{cases}$$

is recursive over $\mathbf{X}$ by Proposition 3.2.33. The function $f : X^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \to X^{\mathbb{N}}$, defined by $f((x_n)_{n\in\mathbb{N}}, q)(0) := x_1$ and

$$f((x_n)_{n\in\mathbb{N}}, q)(n+1) := \begin{cases} x_{n+2} & \text{if} \quad (\forall i \le n+2)\ q(i) = 0 \\ f((x_n)_{n\in\mathbb{N}}, q)(n) & \text{else} \end{cases}$$

is recursive over $\mathbf{X}$ too and so is $L :\subseteq X^{\mathbb{N}} \rightrightarrows X$, defined by

$$L(x_n)_{n\in\mathbb{N}} := \mathrm{Lim}\ f((x_n)_{n\in\mathbb{N}}, [g](x_n)_{n\in\mathbb{N}}).$$

Let $x := (x_n)_{n\in\mathbb{N}} \in X^{\mathbb{N}}$ and define $x' \in X^{\mathbb{N}}$ by $x'_n = x_{n+1}$. If $x \in \mathrm{dom}(\mathrm{Lim})$, then $(x, m) \in A_0 \setminus A_1$ and thus $g(x, m) = 0$ for all $m \in \mathbb{N}$. Hence, $f(x, [g](x)) = x' \in \mathrm{dom}(\mathrm{Lim})$ and $L(x) = \mathrm{Lim}(x') = \mathrm{Lim}(x)$. This proves $L|_{\mathrm{dom(Lim)}} = \mathrm{Lim}$. Now let $x \notin \mathrm{dom}(\mathrm{Lim})$ and $y \in L(x)$. Then there is some $q \in [g](x)$ such that $\mathrm{Lim}\ f(x, q) = y$. If $l := \min q^{-1}\{1\}$ exists, then $f(x, q) = (x_1, x_2, ..., x_k, x_k, ...) \in \mathrm{dom}(\mathrm{Lim})$ with $k := \max\{1, l-1\}$ and $y = L(x) = \mathrm{Lim}\ f(x, q) = x_k$; otherwise $(x, m) \in A_0$ for all m and $y = L(x) = \mathrm{Lim}\ f(x, q) = \mathrm{Lim}(x') = \lim_{n\to\infty} x_n$. In any case, $x \in \mathrm{dom}(L)$, i.e. L is total. $\square$

As a corollary we obtain that there is a total recursive selector $\delta' : \mathbb{N}^{\mathbb{N}} \to X$ of $L \circ \alpha^{\mathbb{N}} : \mathbb{N}^{\mathbb{N}} \rightrightarrows X$ over $\mathbf{X}$ and δ' is an extension of the Cauchy representation δ. A recursive right inverse δ^- of δ over $\mathbf{X}$ is a recursive right inverse of δ' too and $\delta' \equiv \delta$ by the Stability Theorem 3.2.1.

Corollary 4.4.12 (Cauchy representation of Polish spaces) *Let (X, d, α) be a complete recursive metric space. Then the Cauchy representation δ of X can be extended to an equivalent total recursive retraction $\delta' : \mathbb{N}^{\mathbb{N}} \to X$ over $\mathbf{X}$.*

By Proposition 3.2.41 we obtain completeness of $\mathbf{X}$. Now we still have to show that in case of completeness the standard structure of a recursive metric space is strongly effective via the Cauchy representation δ. Since id, d, α are total we only have to show that Lim is strongly computable w.r.t. δ.

Lemma 4.4.13 *If (X, d, α) is a complete recursive metric space with Cauchy representation δ, then* Lim *is strongly (δ^∞, δ)–computable.*

Proof. By the previous proposition it suffices to prove that Lim is $(\delta'^\infty, \delta')$–computable for the total recursive and equivalent extension $\delta' : \mathbb{N}^{\mathbb{N}} \to X$ of δ. By Proposition 4.4.9 we already know that Lim is (δ^∞, δ)–computable. Thus it suffices to prove that

$$\begin{aligned} A &:= \mathrm{dom}(\mathrm{Lim} \circ \delta'^\infty) \\ &= \{\langle p_0, p_1, ...\rangle : (\forall i > j) d(\delta'(p_i), \delta'(p_j)) \le 2^{-j}\} \end{aligned}$$

is a domain over $\mathbf{N}$. By the Projection Theorem 3.2.32 it follows that A^c is semi-recursive over $\mathbf{X}$ and hence over $\mathbf{N}$. Consequently, A is a domain over $\mathbf{N}$ by 3.2.29. $\square$

Completeness of the structure and semi-recursiveness of the inequality implies recursiveness of the equality by Proposition 3.2.48. This finishes the proof of Theorem 4.4.10.

Now we want to show that the results of this section and the previous one also prove Theorem 4.4.2 and that the proof is not cyclic. First we mention that $\alpha_{\mathbb{R}}, d_{\mathbb{R}}$ are recursive over $\mathbf{R}$ by Proposition 4.4.1. Thus the proofs in this section can also be read for the real numbers. The only point where Theorem 4.4.2 had been used in this section was the proof of Lemma 4.4.8. But as already mentioned this lemma can be proved directly for the real numbers.

4.4.4 Constructions on recursive metric spaces

In this section we want to consider some canonical constructions on recursive metric spaces and their stability properties. If we endow the natural numbers $\mathbb{N}$ with the discrete metric d and the identity as a dense sequence, then we obtain a separable metric space. We want to show that this space is a recursive metric space which is recursively isomorphic to the "original" natural numbers. Therefore we write $\mathbb{N}'$ for our "second copy" of the natural numbers.

Proposition 4.4.14 (Stability of natural numbers) *The space* $(\mathbb{N}', d, \mathrm{id}')$ *with the discrete metric* $d : \mathbb{N}' \times \mathbb{N}' \to \mathbb{R}$ *and the identity* $\mathrm{id}' : \mathbb{N} \to \mathbb{N}'$ *is a recursive metric space and* id' *is a recursive isomorphism over* $\mathbf{N}'$.

Proof. Since

$$d(n,k) = \begin{cases} 0 & \text{if} \quad n = k \\ 1 & \text{else} \end{cases}$$

for all $n, k \in \mathbb{N}'$ and equality $=_{\mathbb{N}}$ is decidable over $\mathbf{N}$ we obtain that $d \circ (\mathrm{id}' \times \mathrm{id}') : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$ is recursive over $\mathbf{R}$. Thus, $(\mathbb{N}', d, \mathrm{id}')$ is a recursive metric space. Now $\mathrm{id}' : \mathbb{N} \to \mathbb{N}'$ is recursive over $\mathbf{N}'$ by definition and we still have to show that id'^{-1} is recursive over $\mathbf{N}'$. Let δ be the Cauchy representation of $\mathbb{N}'$ and δ^- a recursive right inverse of δ over $\mathbf{N}'$. Then $f : \mathbb{N}' \times \mathbb{N} \rightrightarrows \mathbb{N}$, defined by $f(n,k) := (\delta^-)_*(n,k)$, is recursive over $\mathbf{N}'$. By definition of δ it is easy to see that $id'^{-1}(n) = f(n,1)$ for all $n \in \mathbb{N}'$. Thus id'^{-1} is recursive over $\mathbf{N}'$. $\square$

Now we want to prove that recursive metric spaces are invariant w.r.t. the canonical constructions of metric product and sequence spaces over structures.

Definition 4.4.15 (Metric product and sequence spaces) Let (X, d_X, α_X) and (Y, d_Y, α_Y) be separable metric spaces.

(1) The *metric product space* $(X \times Y, d_{X\times Y}, \alpha_{X\times Y})$ is defined by

  (a) $d_{X\times Y} : (X \times Y)^2 \to \mathbb{R}, ((x, y), (x', y')) \mapsto \max\{d_X(x, x'), d_Y(y, y')\}$,

  (b) $\alpha_{X\times Y} : \mathbb{N} \to X \times Y, \langle n, k\rangle \mapsto (\alpha_X(n), \alpha_Y(k))$.

(2) The *metric sequence space* $(X^{\mathbb{N}}, d_{X^{\mathbb{N}}}, \alpha_{X^{\mathbb{N}}})$ with

  (a) $d_{X^{\mathbb{N}}} : X^{\mathbb{N}} \times X^{\mathbb{N}} \to \mathbb{R}, ((x_n)_{n\in\mathbb{N}}, (y_n)_{n\in\mathbb{N}}) \mapsto \displaystyle\sum_{i=0}^{\infty} 2^{-i-1} \frac{d_X(x_i, y_i)}{1 + d_X(x_i, y_i)}$,

  (b) $\alpha_{X^{\mathbb{N}}} : \mathbb{N} \to X^{\mathbb{N}}, \alpha_{X^{\mathbb{N}}}\langle\langle n_0, ..., n_k\rangle, k\rangle(i) := \begin{cases} \alpha_X(n_i) & \text{if} \quad i \le k \\ \alpha_X(0) & \text{else} \end{cases}$.

It is easy to show that $(X \times Y, d_{X\times Y}, \alpha_{X\times Y})$ and $(X^{\mathbb{N}}, d_{X^{\mathbb{N}}}, \alpha_{X^{\mathbb{N}}})$ are separable metric spaces if X, Y are so. Here we will show that the same holds for recursive metric spaces. In the following we will sometimes write $X \times Y$ for the set as well as for the metric product space, and $X^{\mathbb{N}}$ for the set of sequences as well as for the metric sequence space.

Proposition 4.4.16 (Metric product and sequence spaces) *If X, Y are recursive metric spaces, then $X \times Y$ and $X^{\mathbb{N}}$ are so too.*

Proof. If (X, d_X, α_X) and (Y, d_Y, α_Y) are recursive metric spaces, then $d_X \circ (\alpha_X \times \alpha_X)$ and $d_Y \circ (\alpha_Y \times \alpha_Y)$ are recursive over $\mathbf{R}$. Thus $d_{X\times Y} \circ (\alpha_{X\times Y} \times \alpha_{X\times Y})$ and $d_{X^{\mathbb{N}}} \circ (\alpha_{X^{\mathbb{N}}} \times \alpha_{X^{\mathbb{N}}})$ are recursive over $\mathbf{R}$ too, since $\max : \mathbb{R} \times \mathbb{R} \to \mathbb{R}, (x, y) \mapsto \max\{x, y\}$ and $\sigma : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}, (x_n)_{n\in\mathbb{N}} \mapsto \sum_{i=0}^{\infty} 2^{-i-1} \frac{x_i}{1+|x_i|}$ are recursive over $\mathbf{R}$ by Proposition 4.4.1. $\square$

Now we have the following situation: whenever we have recursive metric spaces X, Y, then $X \times Y$ is a set over the structure $\mathbf{X} \oplus \mathbf{Y}$ and on the other hand the recursive metric product space $Z := X \times Y$ gives rise to another structure $\mathbf{Z}$. We will prove that Z and $X \times Y$ are recursively isomorphic over $\mathbf{X} \oplus \mathbf{Y} \oplus \mathbf{Z}$. A corresponding statement holds for sequence spaces.

Proposition 4.4.17 (Stability of product and sequence spaces) *Let X and Y be separable metric spaces, let $Z := X \times Y$ be the metric product space and $Z' := X^{\mathbb{N}}$ the metric sequence space. Then $X \times Y$ is recursively isomorphic to Z over $\mathbf{X} \oplus \mathbf{Y} \oplus \mathbf{Z}$ and $X^{\mathbb{N}}$ is recursively isomorphic to Z' over $\mathbf{X} \oplus \mathbf{Z}'$.*

Proof. Since X, Y, Z, Z' are separable metric spaces, the corresponding Cauchy representations $\delta_X, \delta_Y, \delta_Z, \delta_{Z'}$ are recursive and there are recursive right inverses δ_Z^-, $[\delta_X, \delta_Y]^-$, $\delta_{Z'}^-$, and $(\delta_X^\infty)^-$. Now one verifies $[\delta_X, \delta_Y] \equiv \delta_Z$ and $\delta_X^\infty \equiv \delta_{Z'}$. Thus $\mathrm{id} : Z \to X \times Y$ is a recursive isomorphism over $\mathbf{X} \oplus \mathbf{Y} \oplus \mathbf{Z}$ and $\mathrm{id} : Z' \to X^{\mathbb{N}}$ is a recursive isomorphism over $\mathbf{X} \oplus \mathbf{Z}'$. $\square$

Now we want to discuss recursive metric subspaces. First, consider a recursive metric space (X, d, α) and a sequence α' which is dense in X and recursive over $\mathbf{X}$. Then (X, d, α') is a recursive metric space too. Now the question arises whether the standard structure $\mathbf{X}'$ of this recursive metric space is equivalent to $\mathbf{X}$. By definition we have $\mathbf{X}' \leq_s \mathbf{X}$. By Theorem 4.4.6 $\mathbf{X}$, $\mathbf{X}'$ are perfect and thus $\mathbf{X}' \equiv \mathbf{X}$ follows by Theorem 3.2.16. Hence we have proved the following stability result for recursive metric spaces.

Proposition 4.4.18 (Stability w.r.t. dense subsets) *Let (X, d, α) be a recursive metric space and let $\alpha' : \mathbb{N} \to X$ be a sequence which is dense in X and recursive over $\mathbf{X}$. Then (X, d, α') is a recursive metric space with standard structure $\mathbf{X}' \equiv \mathbf{X}$. If (X, d) is complete, then even $\mathbf{X}' \equiv_s \mathbf{X}$ holds.*

Here, the strong equivalence follows by Theorem 4.4.10. This stability result justifies the following definition of a recursive metric subspace.

Definition 4.4.19 (Recursive metric subspaces) Let (X, d, α) be a recursive metric space. If $X' \subseteq X$ is a domain over $\mathbf{X}$ which is *recursively separable* over $\mathbf{X}$, i.e. there is a recursive sequence $\alpha' : \mathbb{N} \to X$ over $\mathbf{X}$ which is dense in X', then $(X', d\rfloor_{X' \times X'}, \alpha'\rceil^{X'})$ is called a *recursive metric subspace* of (X, d, α).

Straightforwardly, we obtain the following result on recursive metric subspaces (cf. Theorem 3.2.25).

Proposition 4.4.20 (Recursive metric subspaces) *Let (X, d, α) be a recursive metric space with a recursive metric subspace $(X', d\rfloor_{X' \times X'}, \alpha'\rceil^{X'})$. Then $\mathrm{in} : X' \to X$ is a recursive embedding over $\mathbf{X} \oplus \mathbf{X}'$.*

We want to close this section with a discussion of equivalence of metrics. Two metrics are called *equivalent*, if they induce the same topology. We want to formulate a sufficient criterion which guarantees that two equivalent metrics induce equivalent structures. Later this will allow us to replace certain metrics by equivalent ones.

Definition 4.4.21 (Recursively related metrics) Let (X, d, α) be a recursive metric space. A metric $d' : X \times X \to \mathbb{R}$ is called *recursively related* to

(X, d, α), if d' is recursive over $\mathbf{X}$ and there is a recursive function $f : \mathbb{N} \to \mathbb{N}$ over $\mathbf{N}$ such that

$$d'(x, y) < 2^{-f(n)} \Longrightarrow d(x, y) < 2^{-n}$$

for all $x, y \in X$, $n \in \mathbb{N}$.

If d' is recursively related to (X, d, α), then we will sometimes say for short that d' is recursively related to d. It should be noticed that if d' is recursively related to d, then d' is equivalent to d and if d is complete, then so is d'. If, additionally, d is recursively related to d', then d and d' are even uniformly equivalent. Now we can formulate our stability condition for the replacement of metrics.

Proposition 4.4.22 (Stability w.r.t. metrics) *Let (X, d, α) be a recursive metric space and let $d' : X \times X \to \mathbb{R}$ be another metric which is recursively related to (X, d, α). Then (X', d', α) with $X' := X$ is a recursive metric space with standard structure $\mathbf{X}' \equiv \mathbf{X}$. If (X, d) is complete, then even $\mathbf{X}' \equiv_s \mathbf{X}$ follows.*

Proof. Since d' is recursive over $\mathbf{X}$, it follows that $d' \circ (\alpha \times \alpha) : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$ is recursive over $\mathbf{X}$, and thus over $\mathbf{R}$, since $\mathbf{X}$ is perfect. Hence (X', d', α) is a recursive metric space with corresponding perfect standard structure $\mathbf{X}' = \mathbf{R} \oplus (X, \alpha, \mathrm{id}, d', \mathrm{Lim}')$. If we can prove that Lim' has a recursive extension over $\mathbf{X}$, then $\mathbf{X}' \leq \mathbf{X}$ and thus $\mathbf{X}' \equiv \mathbf{X}$ by Theorem 3.2.16. By assumption there is a recursive function $f : \mathbb{N} \to \mathbb{N}$ over $\mathbf{N}$ such that $d'(x, y) < 2^{-f(n)} \Longrightarrow d(x, y) < 2^{-n}$ for all $x, y \in X$, $n \in \mathbb{N}$. W.l.o.g. we can assume that f is strictly monotone, i.e. $f(n) > f(k)$ for each $n > k$. Now, the function $F : X^{\mathbb{N}} \to X^{\mathbb{N}}$, defined by $F(x_n)_{n\in\mathbb{N}} := (x_{f(n)+1})_{n\in\mathbb{N}}$ is recursive over $\mathbf{X}$. We claim that $\mathrm{Lim} \circ F$ is a recursive extension of Lim' over $\mathbf{X}$. Let $(x_n)_{n\in\mathbb{N}}$ be a sequence in $\mathrm{dom}(\mathrm{Lim}')$. Then we obtain $d'(x_{f(n)+1}, x_{f(k)+1}) \leq 2^{-f(k)-1}$ for all $n > k$, since $f(n) > f(k)$. Thus, $d(x_{f(n)+1}, x_{f(k)+1}) < 2^{-k}$ for all $n > k$ and $F(x_n)_{n\in\mathbb{N}} \in \mathrm{dom}(\mathrm{Lim})$. Moreover, $\mathrm{Lim} \circ F(x_n)_{n\in\mathbb{N}} = \lim_{n\to\infty} x_{f(n)+1} = \lim_{n\to\infty} x_n = \mathrm{Lim}'(x_n)_{n\in\mathbb{N}}$.

If, additionally, (X, d) is complete, then we have to prove $\mathbf{X}' \leq_s \mathbf{X}$ in order to conclude $\mathbf{X}' \equiv_s \mathbf{X}$. It suffices to prove that $\mathrm{dom}(\mathrm{Lim}')$ is a domain over $\mathbf{X}$. Since (X, d) is complete, (X', d') is complete too and $\mathrm{dom}(\mathrm{Lim}') = \{(x_n)_{n\in\mathbb{N}} : (\forall n > k) d'(x_n, x_k) \leq 2^{-k}\}$. Thus, by the Projection Theorem 3.2.32 it follows that $\mathrm{dom}(\mathrm{Lim})^c$ is semi-recursive over $\mathbf{X}$ and hence $\mathrm{dom}(\mathrm{Lim}')$ is a domain over $\mathbf{X}$ by Proposition 3.2.29. $\square$

Thus, w.r.t. to recursion, we can w.l.o.g. replace a metric by a recursively related one. Especially, we obtain that each recursive metric space (X, d, α) admits a recursively related metric d' which is bounded by 1 and induces an equivalent perfect structure.

Proposition 4.4.23 (Bounded recursive metric spaces) *If (X, d) is a recursive metric space, then by*

$$d'(x,y) := \frac{d(x,y)}{1+d(x,y)}, \quad d''(x,y) := \min\{1, d(x,y)\}$$

two metrics $d', d'' : X \times X \to \mathbb{R}$ are defined, which are bounded by 1 and which are recursively related to (X, d).

Proof. Obviously, d', d'' are recursive over $\mathbf{X}$ and bounded by 1. It is easy to see that d', d'' are metrics. Furthermore, there is a recursive function $f : \mathbb{N} \to \mathbb{N}$ over $\mathbf{N}$ such that $2^{-f(n)} \leq 2^{-n}/(1+2^{-n})$ for all $n \in \mathbb{N}$. Thus, $d'(x,y) < 2^{-f(n)}$ implies

$$\frac{d(x,y)}{1+d(x,y)} = d'(x,y) < 2^{-f(n)} \leq \frac{2^{-n}}{1+2^{-n}}$$

and thus $d(x,y) < 2^{-n}$ for all $x, y \in X$ and $n \in \mathbb{N}$. Hence, d' is recursively related to d. Moreover, $d''(x,y) < 2^{-n}$ implies $d(x,y) = d''(x,y) < 2^{-n}$ such that d'' is recursively related to d too. $\square$

4.4.5 Recursive functions over metric structures

In this section we will give some useful characterizations of recursive functions over recursive metric spaces. Especially, we will show that each recursive function is effectively continuous w.r.t. the corresponding metrics. Therefore, we define the modulus of continuity.

Definition 4.4.24 (Modulus of continuity) Let (X, d), (Y, d') be metric spaces and let $f :\subseteq X \to Y$ be a function. Then $M :\subseteq X \rightrightarrows \mathbb{N}^{\mathbb{N}}$ is called a *modulus of continuity* of f, if $\mathrm{dom}(f) \subseteq \mathrm{dom}(M)$ and

$$d(x,y) < 2^{-m(n)} \Longrightarrow d'(f(x), f(y)) < 2^{-n}$$

for all $x, y \in \mathrm{dom}(f)$, $m \in M(x)$, and $n \in \mathbb{N}$. A function $m : \mathbb{N} \to \mathbb{N}$ is called a *uniform modulus of continuity* of f, if the constant function $M : X \to \mathbb{N}^{\mathbb{N}}, x \mapsto m$ is a modulus of continuity of f.

First we will prove that over recursive metric spaces each recursive function has a recursive modulus of continuity.

Lemma 4.4.25 (Modulus of continuity) *Let X, Y be recursive metric spaces. Then each recursive function $f : X \to Y$ over $\mathbf{X} \oplus \mathbf{Y}$ admits a modulus of continuity $M : X \rightrightarrows \mathbb{N}^{\mathbb{N}}$ which is recursive over $\mathbf{X}$.*

Proof. Let (X, d, α), (Y, d') be recursive metric spaces with Cauchy representations δ_X, δ_Y, respectively, and let δ_X^- be a recursive right inverse of δ_X. If f is recursive over $\mathbf{X} \oplus \mathbf{Y}$, then f is (δ_X, δ_Y)–computable by a computable function $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ by Theorem 3.1.25. Let $F' :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ be defined by $F'(p) := F\langle p, p\rangle$ and let $\varphi : \mathbb{N}^* \to \mathbb{N}^*$ be a computable function which approximates F'. Let the shift $\vec{p}$ be defined by $\vec{p}(n) := p(n+2)$ for each $p \in \mathbb{N}^\mathbb{N}$ and $n \in \mathbb{N}$. Then

$$A := \{(p, n, k) : (\exists w \in \mathbb{N}^*) w \sqsubseteq \vec{p}, \lg \varphi(w) \geq n+3, \lg(w) = k-1\} \subseteq \mathbb{N}^\mathbb{N} \times \mathbb{N} \times \mathbb{N}$$

is semi-recursive over $\mathbf{N}$. Thus, there is a recursive operation $S :\subseteq \mathbb{N}^\mathbb{N} \times \mathbb{N} \rightrightarrows \mathbb{N}$ such that $\mathrm{graph}(S) = A$ and we can select a recursive function $s :\subseteq \mathbb{N}^\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that $\mathrm{dom}(s) = \mathrm{dom}(S)$ and $\mathrm{graph}(s) \subseteq \mathrm{graph}(S)$. Hence, $M : X \to \mathbb{N}^\mathbb{N}$ with $M := [s] \circ \delta_X^-$ is recursive over $\mathbf{X}$. We prove that M is a modulus of continuity of f. Let $x \in X$ and $m \in M(x)$. Then there is a $p \in \delta_X^-(x)$ such that $m = [s](p)$. Let $n \in \mathbb{N}$ and $y \in X$ such that $d(x, y) < 2^{-m(n)}$. Then $(p, n, m(n)) \in A$ and there is a $w \in \mathbb{N}^*$ such that $w \sqsubseteq \vec{p}$, $\lg \varphi(w) \geq n + 3$ and $\lg(w) = m(n) - 1$. Let $a := \alpha(w(m(n) - 2))$ and $b := \alpha(\varphi(w)(n + 2))$. Then $d(x, a) = d(\delta_X(p), \alpha p(m(n))) \leq 2^{-m(n)}$, $d(y, a) < 2^{-m(n)+1}$ and there is a $q \in \delta_X^{-1}\{y\}$ such that $w \sqsubseteq q$. Thus, $d(f(x), f(y)) \leq d(f(x), b) + d(b, f(y)) \leq 2^{-n-2} + 2^{-n-2} < 2^{-n}$. Hence, M is a modulus of continuity of f. $\square$

Now we can prove the following main characterization of recursive functions over metric spaces. We will confine ourself to total functions. The definition of recursive metric subspaces in the previous section allows to relativize it to certain partial functions.

Proposition 4.4.26 (Characterization of recursive functions) *Let (X, d, α), Y be separable metric spaces and let S be an extension of $\mathbf{X} \oplus \mathbf{Y}$. Let $f : X \to Y$ be a function with a modulus of continuity $M : X \rightrightarrows \mathbb{N}^\mathbb{N}$ which is recursive over S. Then the following is equivalent over S:*

(1) f is recursive,

(2) $(x_n)_{n\in\mathbb{N}} \in X^\mathbb{N}$ recursive $\Longrightarrow$ $(f(x_n))_{n\in\mathbb{N}}$ recursive,

(3) $f \circ \alpha$ is recursive.

Proof. Let f be recursive over S and let $(x_n)_{n\in\mathbb{N}}$ be a recursive sequence over S. Then $(f(x_n))_{n\in\mathbb{N}}$ is recursive over S since the composition of recursive functions is recursive again. Moreover, this implies that $f \circ \alpha$ is recursive over S since α is a recursive sequence over $\mathbf{X}$.

Now let $f \circ \alpha$ be recursive over S. W.l.o.g. we assume that all $m \in \mathrm{range}(M)$ are monotone. By Proposition 4.4.7 the Cauchy representation δ_X is a recursive retraction over $\mathbf{X}$, hence it admits a recursive right inverse δ^- over $\mathbf{X}$. Since the sequence composition $s : \mathbb{N}^\mathbb{N} \times \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$, defined by $s(p,q)(n) := pq(n+1)$, is recursive over $\mathbf{N}$,

$$f = \mathrm{Lim}_Y \circ (f \circ \alpha)^\mathbb{N} \circ s \circ (\delta_X^-, M)$$

is recursive over S. □

We will say that a function $f :\subseteq X \to Y$ is *Lipschitz continuous* w.r.t. metric spaces (X,d) and (Y,d'), if there is a constant $L \in \mathbb{R}$ such that $d'(f(x),f(y)) \leq Ld(x,y)$ for all $x,y \in \mathrm{dom}(f)$. Obviously, each Lipschitz continuous function has a recursive modulus of continuity. Thus, the previous equivalence of (1), (2), and (3) especially holds for Lipschitz continuous functions.

We will close this section with an inversion property which has several useful applications. The inverse f^{-1} of an injective recursive function f is not necessarily recursive, nor is it continuous in general. But if the inverse of a recursive injective function admits a recursive modulus of continuity, then it is recursive itself.

Proposition 4.4.27 (Inversion Theorem) *Let X, Y be separable metric spaces and let S be an extension of $\mathbf{X} \oplus \mathbf{Y}$. If $f : X \to Y$ is an injective function which is recursive over S and $M :\subseteq Y \rightrightarrows \mathbb{N}^\mathbb{N}$ is a recursive modulus of continuity of $f^{-1} :\subseteq Y \to X$ over S, then f^{-1} is recursive over S.*

Proof. Let $(X,d,\alpha),(Y,d',\alpha')$ be separable metric spaces. Since $f,d',\alpha,c_<$ are recursive over S, the set

$$A := \{(y,k,n) : d'(f\alpha(n),y) < 2^{-k}\} \subseteq Y \times \mathbb{N} \times \mathbb{N}$$

is semi-recursive over S. By uniformization there is a recursive operation $u : Y \times \mathbb{N} \rightrightarrows \mathbb{N}$ such that $A = \mathrm{graph}(u)$. Let $s : \mathbb{N}^\mathbb{N} \times \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ be the sequence composition, defined by $s(p,q)(k) := pq(k+1)$, which is recursive over S. W.l.o.g. we can assume $m(k) \geq k$ for all $m \in \mathrm{range}(M)$ and $k \in \mathbb{N}$. The function

$$F := \mathrm{Lim}_X \circ \alpha^\mathbb{N} \circ s \circ ([u], M)$$

is recursive over S. We claim $F = f^{-1}$.

Therefore, let $y \in \mathrm{dom}(f^{-1})$, $m \in M(y)$, $n \in u(y, m(k+1))$. Then $d'(f\alpha(n),y) < 2^{-m(k+1)}$. Since f^{-1} is injective and M is a modulus of continuity of f^{-1}, we obtain $d(\alpha(n), f^{-1}(y)) < 2^{-k-1}$, i.e. $F(y) = f^{-1}(y)$.

Now let $y \in \mathrm{dom}(F)$. Then there is an $m \in M(y)$ and since $m(k) \geq k$ for all $k \in \mathbb{N}$ there is some $n \in u(y, m(k+1))$ for each $k \in \mathbb{N}$ such that $d'(f\alpha(n), y) < 2^{-k}$ and $d(\alpha(n), F(y)) \leq 2^{-k}$. Since f is continuous it follows $fF(y) = y$, i.e. $y \in \mathrm{dom}(f^{-1})$. Thus $\mathrm{dom}(F) = \mathrm{dom}(f^{-1})$ and altogether $F = f^{-1}$. $\square$

4.4.6 Recursive sets over metric structures

In this section we will prove some characterizations of (finally) semi-recursive sets over metric structures.

Proposition 4.4.28 (Semi-recursive sets) *Let (X, d, α) be a recursive metric space and $A \subseteq X$. Then the following is equivalent:*

(1) A is semi-recursive over $\mathbf{X}$*,*

(2) $c_A : X \rightrightarrows \mathbb{N}$ is recursive over $\mathbf{X}$*,*

(3) there is a recursive function $f : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$ over $\mathbf{N}$ *such that*

$$A = \bigcup_{(n,k) \in \mathrm{range}(f)} B(\alpha(n), 2^{-k}),$$

(4) there is a recursive function $f : X \to \mathbb{R}$ over $\mathbf{X}$ *such that $A^c = f^{-1}\{0\}$.*

Proof. "(1) $\iff$ (2)" It is easy to see that (1) is equivalent to (2).

"(2) $\Longrightarrow$ (3)" Let δ_X be the Cauchy representation of X. Then c_A is $(\delta_X, \delta_{\mathbb{N}})$–computable via some computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ and there is a computable $\varphi : \mathbb{N}^* \to \mathbb{N}^*$ which approximates F. The set

$$W := \{w = a_0 \ldots a_m \in \mathbb{N}^* : (\forall k)(\forall n) k \leq n \leq m \Longrightarrow d(\alpha(a_k), \alpha(a_n)) < 2^{-k-2}\}$$

is recursively enumerable since (X, d, α) is a recursive metric space and with

$$V := \{p \in \mathbb{N}^{\mathbb{N}} : (\forall n > k) d(\alpha p(n), \alpha p(k)) < 2^{-k-2}\}$$

we obtain $\delta_X(V) = X$. We define

$$B := \{(n,k) \in \mathbb{N}^2 \;:\; (\exists m) w := \pi_1 \nu^*(m) \in W, \lg(w) = k, w(k-1) = n \text{ and } 0 \sqsubseteq \varphi\nu^*(m)\},$$

where $\pi_1 : \mathbb{N}^* \to \mathbb{N}^*$ denotes a computable projection such that $w \sqsubseteq \langle p, q\rangle$ is equivalent to $\pi_1(w) \sqsubseteq p$ for all $p, q \in \mathbb{N}^{\mathbb{N}}$ and $w \in \mathbb{N}^*$. Then B is recursively enumerable too and there is a recursive function $f : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$ over $\mathbf{N}$ with range$(f) = B$. We claim $A = \bigcup_{(n,k)\in B} B(\alpha(n), 2^{-k})$.

"$\subseteq$" Let $x \in A$. Then there is some $p \in V \cap \delta_X^{-1}\{x\}$ and some $q \in \mathbb{N}^{\mathbb{N}}$ such that $\delta_{\mathbb{N}} F\langle p, q\rangle = 0$. Thus there is some $m \in \mathbb{N}$ such that $\nu^*(m) \sqsubseteq \langle p, q\rangle$, $w := \pi_1\nu^*(m) \in W$ and $0 \sqsubseteq \varphi\nu^*(m)$. Let $k := \lg(w)$ and $n := w(k-1)$. Then $(n, k) \in B$ and $d(\alpha(n), x) \leq 2^{-k-1}$, i.e. $x \in B(\alpha(n), 2^{-k})$.

"$\supseteq$" Let $(n, k) \in B$ and $x \in B(\alpha(n), 2^{-k})$. Then there is some $m \in \mathbb{N}$ such that $w := \pi_1\nu^*(m) \in W$, $\lg(w) = k$, $w(k-1) = n$ and $0 \sqsubseteq \varphi\nu^*(m)$. Since $w \in W$ and $d(\alpha(w(k-1)), x) < 2^{-k}$, we can conclude that there is some $p \in \delta_X^{-1}\{x\}$ with $w \sqsubseteq p$ and there is some $q \in \mathbb{N}^{\mathbb{N}}$ such that $\nu^*(m) \sqsubseteq \langle p, q\rangle$ and $\delta_{\mathbb{N}} F\langle p, q\rangle = 0$. Thus, $x = \delta_X(p) \in A$.

"(3) $\Longrightarrow$ (4)" Let $g : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$ be a recursive function over $\mathbf{N}$ such that $A = \bigcup_{(n,k)\in\mathrm{range}(g)} B(\alpha(n), 2^{-k})$. Define $f : X \to \mathbb{R}$ by

$$f(x) := \sum_{i=0}^{\infty} 2^{-i-1} \frac{\max\{0, 2^{-g_2(i)} - d(x, \alpha g_1(i))\}}{1 + 2^{-g_2(i)}}$$

for all $x \in X$. Then f is recursive over $\mathbf{X}$ and

$$f(x) > 0 \iff (\exists i) d(x, \alpha g_1(i)) < 2^{-g_2(i)} \iff x \in A,$$

i.e. $f^{-1}\{0\} = A^c$.

"(4) $\Longrightarrow$ (2)" Let $f : X \to \mathbb{R}$ be recursive over $\mathbf{X}$ and $A^c = f^{-1}\{0\}$. Then $x \in A \iff |f(x)| > 0$ for all $x \in X$, thus $c_A(x) = c_<(0, |f(x)|)$ and c_A is recursive over $\mathbf{X}$. □

An analogous proposition can be formulated for recursively given sequences of semi-recursive sets.

Proposition 4.4.29 (Sequences of semi-recursive sets) *Let (X, d, α) be a recursive metric space and let $(A_n)_{n\in\mathbb{N}}$ be a sequence of subsets of X. Then the following is equivalent:*

(1) $(A_n)_{n\in\mathbb{N}}$ is a recursively given sequence of semi-recursive sets over $\mathbf{X}$,

(2) there is a recursive operation $f : X \times \mathbb{N} \rightrightarrows \mathbb{N}$ over $\mathbf{X}$ such that $f(x, i) = c_{A_i}(x)$ for all $x \in X$, $i \in \mathbb{N}$,

(3) there is a recursive function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N} \times \mathbb{N}$ over $\mathbf{N}$ such that

$$A_i = \bigcup\{B(\alpha(n), 2^{-k}) : (\exists j) f(j, i) = (n, k)\},$$

for all $i \in \mathbb{N}$,

(4) there is a recursive function $f : X \times \mathbb{N} \to \mathbb{R}$ over $\mathbf{X}$ such that $A_i = \{x \in X : f(x,i) \neq 0\}$ for all $i \in \mathbb{N}$.

We omit the proof which is a straightforward generalization of the previous one. Now we prove a similar result for closed finally semi-recursive sets over complete metric structures.

Proposition 4.4.30 (Finally semi-recursive sets) *Let (X,d,α) be a complete recursive metric space and let $A \subseteq X$ be non-empty and closed. Then the following is equivalent:*

(1) A is finally semi-recursive over $\mathbf{X}$,

(2) Ω_A is recursive over $\mathbf{X}$,

(3) there is a recursive function $f : \mathbb{N} \to X$ over $\mathbf{X}$ such that $A = \overline{\mathrm{range}(f)}$.

Proof. "(1) $\iff$ (2)" It is easy to see that (1) is equivalent to (2).

"(2) $\Longrightarrow$ (3)" Let δ_X be the Cauchy representation of X. Then Ω_A is $(\delta_{\{()\}}, \delta_{\mathbb{N}})$–computable via some computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$. Then $f : \mathbb{N} \to X$, defined by $f(n) := \delta_X F\langle \hat{n}, \nu^*(n)\hat{0}\rangle$, is recursive over $\mathbf{X}$ and $\mathrm{range}(f) = \{\delta_X F\langle \hat{n}, \nu^*(n)\hat{0}\rangle : n \in \mathbb{N}\}$ is dense in

$$A = \Omega_A() = \{\delta_X F\langle p, q\rangle : q \in \mathbb{N}^{\mathbb{N}}\}$$

for each $p \in \mathbb{N}^{\mathbb{N}}$, since δ_X, F are continuous. Since A is closed we obtain $A = \overline{\mathrm{range}(f)}$.

"(3) $\Longrightarrow$ (1)" Let $f : \mathbb{N} \to X$ be recursive over $\mathbf{X}$ and $A = \overline{\mathrm{range}(f)}$. Let $L : X^{\mathbb{N}} \rightrightarrows X$ be the total recursive extension of the limit operator which exists by Proposition 4.4.11. Then $g : \mathbb{N}^{\mathbb{N}} \to X$, defined by $g := L \circ f^{\mathbb{N}}$, is recursive over $\mathbf{X}$ and $\mathrm{range}(g) = \overline{\mathrm{range}(f)} = A$. Since $\mathbb{N}^{\mathbb{N}}$ is finally semi-recursive over $\mathbf{N}$, it follows that A is finally semi-recursive over $\mathbf{X}$ by Proposition 3.2.35. $\square$

We could generalize this result straightforwardly to recursively given sequences of finally semi-recursive sets but since we will not use such a generalization in the following we will not explore it here. From the results of this section we can conclude that the (finally) semi-recursive subsets of the real numbers are the well-known r.e. and co-r.e. sets which have been studied by several authors (cf. [Lac58, KF82, Ko91, MN82, NH85, BB85, Ge93, GN94, Zho98, MTY97, Bra99, BW99]).

Corollary 4.4.31 *A set $A \subseteq \mathbb{R}^n$ is semi-recursive over $\mathbf{R}$, if and only if it is r.e. open, a closed set $A \subseteq \mathbb{R}^n$ is finally semi-recursive over $\mathbf{R}$, if and only if it is r.e. closed, and it is recursive over $\mathbf{R}$, if and only if it is a recursive closed set.*

4.4.7 The structure of compact subsets

In this section we will consider the set $\mathcal{K}(X)$ of non-empty compact[1] subsets of a topological space X. In case that (X,d) is a metric space, $\mathcal{K}(X)$ is metrizable and the *Hausdorff metric*

$$d_{\mathcal{K}} : \mathcal{K}(X) \times \mathcal{K}(X) \to \mathbb{R}, (A,B) \mapsto \max\left\{\sup_{a\in A}\inf_{b\in B} d(a,b), \sup_{b\in B}\inf_{a\in A} d(a,b)\right\}$$

induces the *Vietoris topology* on $\mathcal{K}(X)$. From now on we will assume that $\mathcal{K}(X)$ is endowed with this topology. If (X,d) is a (complete) metric space, then $(\mathcal{K}(X), d_{\mathcal{K}})$ is so too, and if Q is a dense subset of X, then the set $\mathcal{F}(Q)$ of non-empty finite subsets of Q is dense in $\mathcal{K}(X)$. With each sequence $\alpha : \mathbb{N} \to X$ with range Q we associate a canonical sequence $\alpha_{\mathcal{K}} : \mathbb{N} \to \mathcal{K}(X)$ with range $\mathcal{F}(Q)$, defined by

$$\alpha_{\mathcal{K}}\langle\langle n_0, ..., n_k\rangle, k\rangle := \{\alpha(n_0), ..., \alpha(n_k)\}$$

for all $n_0, ..., n_k, k \in \mathbb{N}$. Thus, if (X, d, α) is a separable metric space, then $(\mathcal{K}(X), d_{\mathcal{K}}, \alpha_{\mathcal{K}})$ is so too. We will prove now that the same holds for recursive metric spaces.

Proposition 4.4.32 (Space of compact subsets) *If (X, d, α) is a recursive metric space, then $(\mathcal{K}(X), d_{\mathcal{K}}, \alpha_{\mathcal{K}})$ is so too.*

Proof. We have to show that $f := d_{\mathcal{K}} \circ (\alpha_{\mathcal{K}} \times \alpha_{\mathcal{K}}) : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$ is recursive over $\mathbf{R}$. For finite subsets $A, B \subseteq Q := \text{range}(\alpha)$ we obtain

$$d_{\mathcal{K}}(A,B) = \max\left\{\max_{a\in A}\min_{b\in B} d(a,b), \max_{b\in B}\min_{a\in A} d(a,b)\right\}.$$

Since α, d are recursive over $\mathbf{X}$, max is recursive over $\mathbf{R}$ by Proposition 4.4.1, min analogously, it is easy to show that f is recursive over $\mathbf{X}$ too. Since $\mathbf{X}$ is perfect and f is total, it follows that f is recursive over $\mathbf{R}$ by the Conservation Theorem 3.2.8. $\square$

In case that (X,d) is a metric space we will consider the following structure for the space of non-empty compact subsets:

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

[1]The notation $\mathcal{K}(X)$ is motivated by the german translation "kompakt" of "compact"

where $\{x\}$ denotes the canonical injection $\mathrm{in} : X \to \mathcal{K}(X), x \mapsto \{x\}$, A denotes the identity on $\mathcal{K}(X)$, and $A \cup B$ denotes the union operation $\cup : \mathcal{K}(X) \times \mathcal{K}(X) \to \mathcal{K}(X), (A, B) \mapsto A \cup B$, and $d_{\mathcal{K}}$ the Hausdorff metric. As defined in the previous section Lim denotes the standard limit operation of the metric space $(\mathcal{K}(X), d_{\mathcal{K}})$. Our main result about this structure will be the following theorem.

Theorem 4.4.33 (Structure of compact subsets) *If X is a (complete) recursive metric space, then $\mathcal{K}(\mathbf{X})$ is a (complete and strongly) perfect topological structure and the inequality is semi-recursive (and equality is recursive) over this structure.*

The proof follows from Theorem 4.4.6, if we show that our notation $\mathcal{K}(\mathbf{X})$ for the structure above is not ambiguous, that is $\mathcal{K}(\mathbf{X})$ is strongly equivalent to the standard structure of the separable metric space $(\mathcal{K}(X), d_{\mathcal{K}}, \alpha_{\mathcal{K}})$.

Proposition 4.4.34 (Standard structure of compact subsets) *If (X, d, α) is a separable metric space, then $\mathcal{K}(\mathbf{X}) \equiv_s \mathbf{X} \oplus (\mathcal{K}(X), \alpha_{\mathcal{K}}, \mathrm{id}, d_{\mathcal{K}}, \mathrm{Lim})$.*

Proof. Let $\mathcal{K}(\mathbf{X})' := \mathbf{X} \oplus (\mathcal{K}(X), \alpha_{\mathcal{K}}, \mathrm{id}, d_{\mathcal{K}}, \mathrm{Lim})$. We have to show that $\alpha_{\mathcal{K}}$ is recursive over $\mathcal{K}(\mathbf{X})$ and that $\mathrm{in}, \cup$ are recursive over $\mathcal{K}(\mathbf{X})'$. The first follows directly, since

$$\alpha_{\mathcal{K}}\langle\langle n_0, ..., n_k\rangle, k\rangle = \bigcup_{i=0}^{k} \mathrm{in} \circ \alpha(n_i)$$

and $\mathrm{in}, \alpha, \cup$ are recursive over $\mathcal{K}(\mathbf{X})$. The latter follows since it is easy to see that there are recursive functions f, g over $\mathbf{N}$, such that $\mathrm{in} \circ \alpha = \alpha_{\mathcal{K}} \circ f$ and $\cup \circ (\alpha_{\mathcal{K}} \times \alpha_{\mathcal{K}}) = (\alpha_{\mathcal{K}} \times \alpha_{\mathcal{K}}) \circ g$. Thus $\mathrm{in} \circ \alpha, \cup \circ (\alpha_{\mathcal{K}} \times \alpha_{\mathcal{K}})$ are recursive over $\mathcal{K}(\mathbf{X})'$. Moreover, $d_{\mathcal{K}}(\{x\}, \{y\}) = d(x, y)$ for all $x, y \in X$ and $d_{\mathcal{K}}(A \cup B, A' \cup B') \leq \max\{d_{\mathcal{K}}(A, A'), d_{\mathcal{K}}(B, B')\}$ for all $A, A', B, B' \in \mathcal{K}(X)$, thus $\mathrm{in}, \cup$ are Lipschitz continuous (w.r.t. the product metric on $\mathcal{K}(X) \times \mathcal{K}(X)$) and hence they are recursive over $\mathcal{K}(\mathbf{X})'$ by Proposition 4.4.26. $\square$

Now we want to give some examples of recursive functions over the structure of compact subsets. The first example shows that the supremum and infimum functions on recursive subsets of the real numbers are recursive.

Proposition 4.4.35 (Supremum and infimum) *The infimum and supremum functions* $\inf, \sup : \mathcal{K}(\mathbb{R}) \to \mathbb{R}$ *are recursive over* $\mathcal{K}(\mathbf{R})$.

Proof. We will only prove that sup is recursive, the case inf can be proved analogously. For each finite set $A = \{a_1, ..., a_n\} \subseteq \mathbb{Q}$ we have $\sup(A) =$

$\max\{a_1, ..., a_n\}$. Since max is recursive over $\mathbf{R}$ by Proposition 4.4.1, it is easy to see that $\sup \circ \alpha_{\mathcal{K}} : \mathbb{N} \to \mathbb{R}$ is recursive over $\mathcal{K}(\mathbf{R})$. By Proposition 4.4.26 it suffices to prove that sup is Lipschitz continuous, but this follows from $|\sup(A) - \sup(B)| \leq d_{\mathcal{K}}(A, B)$. $\square$

The next proposition shows that the product operation on compact subsets is a recursive isomorphism.

Proposition 4.4.36 (Product) *Let X, Y be separable metric spaces. The function $\times : \mathcal{K}(X) \times \mathcal{K}(Y) \to \mathcal{K}(X \times Y), (A, B) \mapsto A \times B$ is a recursive isomorphism over $\mathcal{K}(\mathbf{X}) \oplus \mathcal{K}(\mathbf{Y}) \oplus \mathcal{K}(\mathbf{X} \times \mathbf{Y})$.*

Proof. Let (X, d, α), (Y, d', α') be separable metric spaces, let $(X \times Y, d'', \alpha'')$ be the corresponding metric product space, and let $(\mathcal{K}(X), d_{\mathcal{K}}, \alpha_{\mathcal{K}})$, $(\mathcal{K}(Y), d'_{\mathcal{K}}, \alpha'_{\mathcal{K}})$, $(\mathcal{K}(X \times Y), d''_{\mathcal{K}}, \alpha''_{\mathcal{K}})$ be the associated spaces of compact subsets. For finite sets $A = \{a_1, ..., a_m\} \subseteq \text{range}(\alpha)$ and $B = \{b_1, ..., b_l\} \subseteq \text{range}(\alpha')$ we obtain $A \times B = \{(a_i, b_j) : i = 1, ..., m, j = 1, ..., l\}$. Thus, it is easy to show that there is a recursive function $s : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ over $\mathbf{N}$ such that $\alpha''_{\mathcal{K}} s(n, k) = \alpha_{\mathcal{K}}(n) \times \alpha'_{\mathcal{K}}(k)$. Furthermore, $\times$ as well as its inverse are Lipschitz continuous, since $d''_{\mathcal{K}}(A \times B, A' \times B') = \max\{d_{\mathcal{K}}(A, A'), d'_{\mathcal{K}}(B, B')\}$. By Proposition 4.4.26 and 4.4.27 it follows that $\times$ is a recursive isomorphism over $\mathcal{K}(\mathbf{X}) \oplus \mathcal{K}(\mathbf{Y}) \oplus \mathcal{K}(\mathbf{X} \times \mathbf{Y})$. $\square$

The next lemma shows that the set $\{\{x\} : x \in X\} \subseteq \mathcal{K}(X)$ of single-valued subsets of X is recursively isomorphic to X.

Lemma 4.4.37 (Single-valued subsets) *If X is a separable metric space, then* $\text{in} : X \to \mathcal{K}(X)$ *is a recursive embedding over $\mathcal{K}(\mathbf{X})$.*

Proof. Let (X, d) be a separable metric space. First we note that $\text{in} : X \to \mathcal{K}(X)$ is recursive over $\mathcal{K}(\mathbf{X})$ by definition. Moreover, $d_{\mathcal{K}}(\{x\}, \{y\}) = d(x, y)$ for all $x, y \in X$, i.e. in is an injective isometry and especially, in^{-1} is Lipschitz continuous. By the Inversion Theorem 4.4.27 it follows that in^{-1} is recursive over $\mathcal{K}(\mathbf{X})$. $\square$

4.4.8 Lifting of functions

In this section we want to show that one can lift recursive functions to the hyperspace of compact subsets, i.e. with each function $f : X \to Y$ we associate a *lifted function* $\hat{f} : \mathcal{K}(X) \to \mathcal{K}(Y), A \mapsto f(A)$. First we want to show that $\hat{f}$ has a recursive modulus of continuity $\hat{M} : \mathcal{K}(X) \rightrightarrows \mathbb{N}^{\mathbb{N}}$, if f is recursive and

X, Y are recursive metric spaces. Additionally, our construction guarantees that each $m \in \hat{M}(A)$ is a uniform modulus of continuity of $f|_A$ for each $A \in \mathcal{K}(X)$.

Lemma 4.4.38 (Uniform modulus of continuity) *Let X, Y be recursive metric spaces. Then for each recursive function $f : X \to Y$ over $\mathbf{X} \oplus \mathbf{Y}$ there is a modulus of continuity $\hat{M} : \mathcal{K}(X) \rightrightarrows \mathbb{N}^{\mathbb{N}}$ of $\hat{f} : \mathcal{K}(X) \to \mathcal{K}(Y), A \mapsto f(A)$ which is recursive over $\mathcal{K}(\mathbf{X})$. Additionally, each $m \in \hat{M}(A)$ is a uniform modulus of continuity of $f|_A$ for each $A \in \mathcal{K}(X)$.*

Proof. Let (X, d, α), (Y, d') be recursive metric spaces with Cauchy representations δ_X, δ_Y, respectively, and let $\delta_\mathcal{K}$ be the corresponding Cauchy representation of $(\mathcal{K}(X), d_\mathcal{K}, \alpha_\mathcal{K})$ with recursive right inverse $\delta_\mathcal{K}^-$. If f is recursive over $\mathbf{X} \oplus \mathbf{Y}$, then f is (δ_X, δ_Y)–computable by a computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ by Theorem 3.1.25. Let $F' :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ be defined by $F'(p) := F\langle p, p\rangle$ and let $\varphi : \mathbb{N}^* \to \mathbb{N}^*$ be a computable function which approximates F'. We define a function $W :\subseteq \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \to \mathcal{F}(\mathbb{N}^*)$ by

$$\begin{cases} W(p, 0) & := & \{n_{0,0}, ..., n_{0,k_0}\} \\ W(p, i+1) & := & \{w_0, ..., w_{k_{i+1}}\} \end{cases}$$

for all $p \in \mathrm{dom}(\delta_\mathcal{K})$, $i \in \mathbb{N}$, where $\langle\langle n_{i,0}, ..., n_{i,k_i}\rangle, k_i\rangle := p(i+2)$ for all $i \in \mathbb{N}$ and the words $w_0, ..., w_{k_{i+1}}$ are selected effectively such that for all $i \in \mathbb{N}$ and $j = 0, ..., k_{i+1}$ there exists $w = a_0...a_i \in W(p, i)$ with

$$w_j = wn_{i+1,j} \text{ and } (\forall m = 0, ..., i) d(\alpha(a_m), \alpha(n_{i+1,j})) < 2^{-m-1},$$

which is possible by Lemma 4.4.8 and since $d_\mathcal{K}(\alpha_\mathcal{K} p(i+2), \alpha_\mathcal{K} p(i+3)) \leq 2^{-i-2}$ for all $i \in \mathbb{N}$. Thus

$$A := \{(p, n, k) : k \geq 1 \text{ and } \min \lg \varphi(W(p, k-1)) \geq n+3\} \subseteq \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \times \mathbb{N}$$

is semi-recursive over $\mathbf{N}$. Hence there is a recursive operation $S :\subseteq \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \rightrightarrows \mathbb{N}$ such that $\mathrm{graph}(S) = A$ and we can select a recursive function $s :\subseteq \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{N}$ such that $\mathrm{dom}(s) = \mathrm{dom}(S)$ and $\mathrm{graph}(s) \subseteq \mathrm{graph}(S)$. Hence $\hat{M} :\subseteq \mathcal{K}(X) \rightrightarrows \mathbb{N}^{\mathbb{N}}$ with $\hat{M} := [s] \circ \delta_\mathcal{K}^-$ is recursive over $\mathcal{K}(\mathbf{X})$. We claim that $\hat{M}$ is a modulus of continuity of $\hat{f}$. First we have to show that $\hat{M}$ is total. Therefore, it suffices to prove that for each $p \in \mathrm{dom}(\delta_\mathcal{K})$ and $n \in \mathbb{N}$ there exists some $k \in \mathbb{N}$ such that $(p, n, k) \in A$. Assume that the latter does not hold, then by König's Lemma there is a $q \in \mathbb{N}^{\mathbb{N}}$ and an $n \in \mathbb{N}$ such that

$$(\forall k \in \mathbb{N})(\exists w \in W(p, k)) w \sqsubseteq q \text{ and } \lg \varphi(w) < n+3,$$

i.e. $q \notin \mathrm{dom}(F')$. But on the other hand, by construction of W and completeness of $A := \delta_{\mathcal{K}}(p)$ we have $q \in \delta_X^{-1}(A)$. Contradiction! This proves that $\hat{M}$ is total. Moreover, for each $A, B \in \mathcal{K}(X)$ and $m \in \hat{M}(A)$ we obtain

$$d_{\mathcal{K}}(A, B) < 2^{-m(n)} \Longrightarrow d'_{\mathcal{K}}(f(A), f(B)) < 2^{-n}$$

for all $n \in \mathbb{N}$ and $d(x, y) < 2^{-m(n)} \Longrightarrow d'(f(x), f(y)) < 2^{-n}$ for all $x, y \in A$ and $n \in \mathbb{N}$. Thus $\hat{M}$ is a modulus of continuity of $\hat{f}$ and m is a uniform modulus of continuity of $f|_A$. □

The next proposition shows that a function is recursive, if and only if the corresponding lifted function is recursive.

Proposition 4.4.39 (Lifting of functions) *Let X, Y be separable metric spaces and let S be an extension of $\mathcal{K}(\mathbf{X}) \oplus \mathcal{K}(\mathbf{Y})$. Let $f : X \to Y$ be a function and $\hat{M} : \mathcal{K}(X) \rightrightarrows \mathbb{N}^{\mathbb{N}}$ a modulus of continuity of $\hat{f} : \mathcal{K}(X) \to \mathcal{K}(Y), A \mapsto f(A)$, such that $\hat{M}$ is recursive over S. Then f is recursive over S, if and only if $\hat{f}$ is recursive over S.*

Proof. Let (X, d, α), Y be separable metric spaces. Let f be recursive over S. By Proposition 4.4.26 it suffices to prove that $\hat{f} \circ \alpha_{\mathcal{K}} : \mathbb{N} \to \mathcal{K}(Y)$ is recursive over S. This follows from

$$\hat{f} \circ \alpha_{\mathcal{K}} \langle \langle n_0, ..., n_k \rangle, k \rangle = \hat{f} \circ \bigcup_{i=0}^{k} \mathrm{in} \circ \alpha(n_i) = \bigcup_{i=0}^{k} \mathrm{in} \circ f \circ \alpha(n_i)$$

since $f \circ \alpha$ is recursive over S. Thus, $\hat{f}$ is recursive over S.

Now let $\hat{f} : \mathcal{K}(X) \to \mathcal{K}(Y)$ be recursive over S. It follows that $\hat{f} \circ \alpha_{\mathcal{K}}$ is recursive over S. Since $f \circ \alpha(n) = \mathrm{in}^{-1} \circ \hat{f} \circ \alpha_{\mathcal{K}} \langle n, 0 \rangle$, it follows that $f \circ \alpha$ is recursive over S by Lemma 4.4.37. Moreover, $M : X \rightrightarrows \mathbb{N}^{\mathbb{N}}$, defined by $M := \hat{M} \circ \mathrm{in}$ is recursive over S and a modulus of continuity of f. By Proposition 4.4.26 it follows that f is recursive over S. □

With Lemma 4.4.38 we immediately get the following corollary.

Corollary 4.4.40 (Lifting of functions) *Let X, Y be recursive metric spaces and let $f : X \to Y$ be a function. Then the following is equivalent:*

(1) $f : X \to Y$ is recursive over $\mathbf{X} \oplus \mathbf{Y}$,

(2) $\hat{f} : \mathcal{K}(X) \to \mathcal{K}(Y), A \mapsto f(A)$ is recursive over $\mathcal{K}(\mathbf{X}) \oplus \mathcal{K}(\mathbf{Y})$.

4.4.9 Recursively locally compact metric spaces

In this section we want to introduce recursively locally compact metric spaces. Classically, a Hausdorff space is called *locally compact*, if each point has a compact neighbourhood. One can prove that locally compact separable metric spaces are K_σ-spaces, i.e. they can be represented as a countable union of compact subsets. More than this, they admit an exhausting sequence of compact subsets. We will take this characterization for the definition of recursively locally compact separable metric spaces.

Definition 4.4.41 (Recursively locally compact metric spaces) A separable metric space X is called *recursively locally compact*, if there is a sequence $(K_i)_{i\in\mathbb{N}}$ of compact subsets $K_i \in \mathcal{K}(X)$ such that:

(1) $X = \bigcup_{i=0}^{\infty} K_i$ and $K_i \subseteq K_{i+1}^\circ$ for all $i \in \mathbb{N}$,

(2) $(K_i)_{i\in\mathbb{N}}$ is a recursive sequence over $\mathcal{K}(\mathbf{X})$,

(3) $(K_i^\circ)_{i\in\mathbb{N}}$ is a recursively given sequence of semi-recursive sets over $\mathbf{X}$.

In this situation $(K_i)_{i\in\mathbb{N}}$ is called a *recursive exhausting sequence* of X. If at least (1) holds, then X is called *locally compact*.

As an example we prove that the Euclidean space is recursively locally compact.

Proposition 4.4.42 *The recursive metric space $\mathbb{R}^n$ is recursively locally compact with recursive exhausting sequence $([-i,i]^n)_{i\in\mathbb{N}}$.*

Proof. Define a function $s : \mathbb{N} \times \mathbb{N} \to \mathcal{K}(\mathbb{R}^n)$ by

$$s(i,k) := \left\{-i, \frac{-ik}{k+1}, ..., \frac{-1}{k+1}, 0, \frac{1}{k+1}, ..., \frac{ik}{k+1}, i\right\}^n$$

for all $i, k \in \mathbb{N}$. Then $K_i := \lim_{k\to\infty} s(i, 2^k) = [-i,i]^n$ and $(K_i)_{i\in\mathbb{N}}$ is a recursive sequence over $\mathcal{K}(\mathbf{R^n})$. Moreover, $f : \mathbb{R}^n \times \mathbb{N} \rightrightarrows \mathbb{N}$, defined by

$$f((x_1, ..., x_n), i) := \max\{c_<(-i, x_1), ..., c_<(-i, x_n), c_<(x_1, i), ..., c_<(x_n, i)\}$$

is recursive over $\mathbf{R}$ and $f(x,i) = c_{(-i,i)^n}(x) = c_{K_i^\circ}(x)$ for all $x \in \mathbb{R}, i \in \mathbb{N}$. Thus, $(K_i^\circ)_{i\in\mathbb{N}}$ is a recursively given sequence of semi-recursive sets over $\mathbf{R}$. Altogether, $(K_i)_{i\in\mathbb{N}}$ is a recursive exhausting sequence of $\mathbb{R}^n$ over $\mathcal{K}(\mathbf{R^n})$. $\square$

An important property of recursively locally compact spaces spaces is that they not only provide compact neighbourhoods of points, but given a point we can find such a neighbourhood. This is made precise in the following proposition.

Proposition 4.4.43 (Modulus of exhausting) *If X is a locally compact separable metric space with exhausting sequence $(K_i)_{i\in\mathbb{N}}$ such that $(K_i^\circ)_{i\in\mathbb{N}}$ is a recursively given sequence of semi-recursive sets over $\mathbf{X}$, then there is a recursive operation $m : X \rightrightarrows \mathbb{N}$ over $\mathbf{X}$ such that $n \in m(x)$ implies $x \in K_n^\circ$ for all $x \in X$.*

Proof. By assumption there exists a recursive function $f : X \times \mathbb{N} \rightrightarrows \mathbb{N}$ over $\mathbf{X}$ such that $f(x,n) = c_{K_n^\circ}(x)$ for all $x \in X, n \in \mathbb{N}$. Then $m : X \rightrightarrows \mathbb{N}$, defined by $m(x) := f^{\leftrightarrow}(x,0)$ for all $x \in X$ is recursive over $\mathbf{X}$ and we obtain

$$n \in m(x) \Longrightarrow 0 \in f(x,n) \Longrightarrow x \in K_n^\circ$$

for all $x \in X$. □

A recursive exhausting sequence $(K_i)_{i\in\mathbb{N}}$ fulfills two different kind of effectivity conditions. On the one hand, it is a recursive sequence over $\mathcal{K}(\mathbf{X})$, on the other hand, the interiors form a recursively given sequence of semi-recursive sets over $\mathbf{X}$. Sometimes, it will be helpful to know that the complements K_i^c form a recursively given sequence of semi-recursive sets too.

Lemma 4.4.44 *Let X be a recursive metric space. If $(K_i)_{i\in\mathbb{N}}$ is a recursive sequence over $\mathcal{K}(\mathbf{X})$, then $(K_i^c)_{i\in\mathbb{N}}$ is a recursively given sequence of semi-recursive sets over $\mathbf{X}$.*

Proof. Define $f : X \times \mathbb{N} \to \mathbb{R}$ by

$$f(x,n) := d_{\mathcal{K}}(K_n, \{x\})$$

for all $x \in X, n \in \mathbb{N}$. Then f is recursive over $\mathcal{K}(\mathbf{X})$, and thus over $\mathbf{X}$ and

$$K_n^c = \{x \in X : d_{\mathcal{K}}(K_n, \{x\}) > 0\} = \{x \in X : f(x,n) \neq 0\}$$

for all $n \in \mathbb{N}$. Thus $(K_n^c)_{n\in\mathbb{N}}$ is a recursively given sequence of semi-recursive sets over $\mathbf{X}$ by Proposition 4.4.29. □

4.4.10 Nice metric spaces

For the construction of hyperspaces of locally compact metric spaces (X,d) we will need special metrics with nice additional properties. In a certain sense these metrics have to be subordinated to the exhausting sequence $(K_i)_{i\in\mathbb{N}}$ of the space.[2]

[2] For the idea of nice metrics cf. [Bee93].

Definition 4.4.45 (Nice metric spaces) A locally compact separable metric space (X, d) with exhausting sequence $(K_i)_{i\in\mathbb{N}}$ is called *nice* w.r.t. $(K_i)_{i\in\mathbb{N}}$, if d is bounded by 1 and if $x \in K_i$, $d(x, y) < 1$ implies $y \in K_{i+1}^{\circ}$ for all $x, y \in X$ and $i \in \mathbb{N}$.

In other words $\bigcup_{x\in K_i} B(x, 1) \subseteq K_{i+1}^{\circ}$ holds for nice metric spaces. It is easy to see that nice metric spaces are complete. If (X, d) is a nice metric space, then we will sometimes say for short that d is nice. The following example shows that there are nice recursively locally compact metric spaces.

Example 4.4.46 *The Euclidean space $\mathbb{R}^n$ with maximum metric d and exhausting sequence $([-i, i]^n)_{i\in\mathbb{N}}$ admits the nice metric d', defined by $d'(x, y) := \min\{1, d(x, y)\}$.*

The proof follows immediately, since $|x| \le i$ and $|x - y| < 1$ implies $|y| < i + 1$. Our following main result about nice metrics states that each recursively locally compact recursive metric space admits a recursively related nice metric.

Proposition 4.4.47 (Nice metric spaces) *Let (X, d) be a recursively locally compact recursive metric space with recursive exhausting sequence $(K_i)_{i\in\mathbb{N}}$. Then there exists a nice metric $d' : X \times X \to \mathbb{R}$ w.r.t. $(K_i)_{i\in\mathbb{N}}$ which is recursively related to d.*

Proof. By Proposition 4.4.23 and since recursive relatedness is a transitive relation we can assume w.l.o.g. that d is bounded by 1. By Lemma 4.4.44 the sequences $(K_{i+1}^{\circ})_{i\in\mathbb{N}}$ and $(K_i^c)_{i\in\mathbb{N}}$ are recursively given sequences of semi-recursive sets over $\mathbf{X}$. By Proposition 4.4.29 there are recursive functions $f, g : X \times \mathbb{N} \to \mathbb{R}$ over $\mathbf{X}$ such that $\overline{K_{i+1}^c} = \{x \in X : f(x, i) = 0\}$ and $K_i = \{x \in X : g(x, i) = 0\}$. We define a function $h : X \times \mathbb{N} \to \mathbb{R}$ by $h(x, i) := |f(x, i)|/(|f(x, i)| + |g(x, i)|)$ for all $x \in X$ and $i \in \mathbb{N}$. Then h is recursive over $\mathbf{X}$ and h is total since $\overline{K_{i+1}^c} \cap K_i = \emptyset$ for all $i \in \mathbb{N}$. We obtain $\overline{K_{i+1}^c} = \{x \in X : h(x, i) = 0\}$ and $K_i = \{x \in X : h(x, i) = 1\}$ for all $i \in \mathbb{N}$. Now, let $d' : X \times X \to \mathbb{R}$ be defined by

$$d'(x, y) = \min\{1, d(x, y) + \sum_{i=0}^{\infty} |h(x, i) - h(y, i)|\}$$

for all $x, y \in X$. By Proposition 4.4.43 there exists a recursive modulus of exhausting $m : X \rightrightarrows \mathbb{N}$ of $(K_i)_{i\in\mathbb{N}}$ over $\mathbf{X}$. Since $j \in \max(m(x), m(y))$ implies $h(x, i) - h(y, i) = 0$ for all $i \ge j$ we obtain $\sum_{i=0}^{\infty} |h(x, i) - h(y, i)| = \sum_{i=0}^{\max(m(x),m(y))} |h(x, i) - h(y, i)|$ and thus d' is recursive over $\mathbf{X}$. Moreover, $d(x, y) \le d'(x, y)$ for all $x, y \in X$ since d is bounded by 1 and thus d'

is recursively related to d. Now let $x, y \in X$ with $d'(x,y) < 1$. Then $|h(x,i) - h(y,i)| < 1$ for all $i \in \mathbb{N}$. Consider some i such that $x \in K_i$, then $h(x,i) = 1$ and hence $h(y,i) > 0$. But this implies $y \in K_{i+1}^{\circ}$. □

We have constructed a recursively related nice metric d' for a given recursive metric space (X,d) with exhausting sequence $(K_i)_{i\in\mathbb{N}}$. Let $X' := X$ denote the metric space endowed with the metric d'. Although we know that $\mathbf{X}' \equiv \mathbf{X}$ by Proposition 4.4.22, up to now we do not know whether $\mathcal{K}(\mathbf{X}') \equiv \mathcal{K}(\mathbf{X})$ and thus whether $(K_i)_{i\in\mathbb{N}}$ is a recursive sequence over $\mathcal{K}(\mathbf{X}')$. The following proposition answers this question in the affirmative.

Proposition 4.4.48 (Stability of the structure of compact subsets) *Let (X,d,α) be a recursive metric space with a recursively related metric d'. Then the corresponding metric $d'_{\mathcal{K}}$ is recursively related to $(\mathcal{K}(X), d_{\mathcal{K}}, \alpha_{\mathcal{K}})$.*

Proof. Since d' is recursively related to (X,d,α) it is recursive over $\mathbf{X}$ and we obtain that the lifted metric $\widehat{d'} : \mathcal{K}(X \times X) \to \mathcal{K}(\mathbb{R})$ is recursive over $\mathcal{K}(\mathbf{R}) \oplus \mathcal{K}(\mathbf{X})$ by Corollary 4.4.40. Together with Proposition 4.4.35 and 4.4.36 we can conclude that

$$d'_{\mathcal{K}} : \mathcal{K}(X) \times \mathcal{K}(X) \to \mathbb{R}, (A,B) \mapsto \max\left\{\sup_{a\in A}\inf_{b\in B} d'(a,b), \sup_{b\in B}\inf_{a\in A} d'(a,b)\right\}$$

is recursive over $\mathcal{K}(\mathbf{X})$. Since d' is recursively related to d there exists a recursive function $f : \mathbb{N} \to \mathbb{N}$ such that $d'(x,y) < 2^{-f(n)}$ implies $d(x,y) < 2^{-n}$ for all $x, y \in X$ and $n \in \mathbb{N}$. Let $A \in \mathcal{K}(X)$, $x \in X$, $n \in \mathbb{N}$ and $d'_A(x) < 2^{-f(n)}$. Then there is some $a \in A$ such that $d'(a,x) = d'_A(x) < 2^{-f(n)}$ and thus $d_A(x) \leq d(a,x) < 2^{-n}$. Now let $B \in \mathcal{K}(X)$ and $\sup_{b\in B} d'_A(b) < 2^{-f(n)}$. Then for each $x \in B$ we obtain $d'_A(x) \leq \sup_{b\in B} d'_A(b) < 2^{-f(n)}$ and thus $d_A(x) < 2^{-n}$. Consequently, $\sup_{b\in B} d_A(b) \leq 2^{-n}$. Altogether, $d'_{\mathcal{K}}(A,B) < 2^{-f(n+1)}$ implies $d_{\mathcal{K}}(A,B) \leq 2^{-n-1} < 2^{-n}$ and $d'_{\mathcal{K}}$ is recursively related to $(\mathcal{K}(X), d_{\mathcal{K}}, \alpha_{\mathcal{K}})$. □

With regard to recursiveness the results of this section allow us to assume w.l.o.g. that a recursively locally compact recursive metric space is nice.

4.4.11 Recursive Banach spaces

Once we have established a definition of a recursive metric space, we can easily define recursiveness of spaces with a richer structure, like Banach spaces, Banach algebras, Hilbert spaces, and so on. We will just demand that the induced separable metric space is recursive and the canonical operations become recursive. Following this line we will define recursive Banach spaces. For each

Banach space $(X, +, \cdot, || \, ||)$ there is an induced metric $d : X \times X \to \mathbb{R}$, defined by $d(x, y) := ||x - y||$. Moreover, to each sequence $e : \mathbb{N} \to X$ such that the linear span of range(e) is dense in X, there is a canonical sequence $\alpha_e : \mathbb{N} \to X$ which is dense in X, defined by

$$\alpha_e\langle\langle i_0, j_0\rangle, ..., \langle i_n, j_n\rangle, n\rangle := \sum_{k=0}^{n} \alpha_{\mathbb{R}}(i_k)e(j_k).$$

Here we assume that X is a Banach space over $\mathbb{R}$, the case of the complex numbers $\mathbb{C}$ can be treated correspondingly.

Definition 4.4.49 (Recursive Banach space) We will call a triple $(X, +, \cdot, || \, ||, e)$ a *recursive Banach space*, if

(1) $(X, +, \cdot, || \, ||)$ is a Banach space with addition $+ : X \times X \to X$, multiplication $\cdot : \mathbb{R} \times X \to X$, and norm $|| \, || : X \to \mathbb{R}$,

(2) $e : \mathbb{N} \to X$ is a sequence such that the linear span of range(e) is dense in X,

(3) the induced separable metric space (X, d, α_e) is recursive and the operations $+$ and $\cdot$ are recursive over the corresponding standard structure $\mathbf{X}$.

If $(X, +, \cdot, || \, ||, e)$ fulfills (1) and (2), then it is called a *separable Banach space*.

It is easy to see that (3) implies that $0 \in X$ is a recursive point over $\mathbf{X}$ and $- : X \times X \to X$, $|| \, || : X \to \mathbb{R}$ are recursive over $\mathbf{X}$, since e is recursive over $\mathbf{X}$. By abuse of notation we will sometimes write X for the set X as well as for the recursive Banach space $(X, +, \cdot, || \, ||, e)$. We will use the following *standard structure*

$$\boxed{\mathbf{X} := \mathbf{R} \oplus (X, e, \mathrm{id}, +, \cdot, || \, ||, \mathrm{Lim})}$$

for a recursive Banach space $(X, +, \cdot, || \, ||, e)$. By definition it follows immediately $\mathbf{X} \equiv_s \mathbf{R} \oplus (X, \alpha_e, \mathrm{id}, d, \mathrm{Lim})$ for the induced metric space (X, d, α_e).

Since a linear function is continuous, if and only if it is Lipschitz continuous, and if and only if it is bounded, it is easy to prove the following characterization of recursive bounded linear functions.

Corollary 4.4.50 (Characterization of recursive linear functions) *Let $(X, +, \cdot, || \, ||, e)$ and Y be separable Banach spaces and let $f : X \to Y$ be a linear and bounded function. Then the following is equivalent over (each extension of) $\mathbf{X} \oplus \mathbf{Y}$:*

(1) f *is recursive,*

(2) $(x_n)_{n\in\mathbb{N}} \in X^{\mathbb{N}}$ *recursive* $\Longrightarrow (f(x_n))_{n\in\mathbb{N}}$ *recursive,*

(3) $f \circ e$ *is recursive.*

4.4.12 The space of continuous functions

In this section we will discuss the set $\mathcal{C}(X,Y)$ of continuous functions $f : X \to Y$. We will write $\mathcal{C}(X)$ for the special case of the continuous functions $f : X \to \mathbb{R}$. Our goal is to endow these sets with a recursive metric space structure. In case that $(Y, ||\ ||)$ is a normed space and X is compact, by $||f||_X := \sup_{x\in X} ||f(x)||$ we can define the well-known *supremum norm* on $\mathcal{C}(X,Y)$. In case that X is a locally compact metric space with exhausting sequence $(K_i)_{i\in\mathbb{N}}$, the Fréchet combination

$$d_{\mathcal{C}}(f,g) := \sum_{i=0}^{\infty} 2^{-i-1} \frac{||f-g||_{K_i}}{1+||f-g||_{K_i}}$$

yields a metric on $\mathcal{C}(X,Y)$. The topology induced by this metric is the *compact-open topology* and from now on we assume that $\mathcal{C}(X,Y)$ is endowed with this topology. Sometimes we will write $d_{\mathcal{C}(X,Y)}$ or $d_{\mathcal{C}(X)}$ instead of $d_{\mathcal{C}}$ to indicate the target space Y, $\mathbb{R}$, respectively. Since $\mathbb{R}, Y$ are complete spaces, the spaces $(\mathcal{C}(X), d_{\mathcal{C}(X)})$ and $(\mathcal{C}(X,Y), d_{\mathcal{C}(X,Y)})$ are complete too. Now we want to construct dense sequences in these spaces. The following classical Stone-Weierstraß Approximation Theorem (cf. [Bou66]) guarantees the existence of dense subsets in the spaces of continuous functions. We will say that a set $\mathcal{F} \subseteq \mathcal{C}(X)$ *separates points*, if for all $x, y \in X$ with $x \neq y$ there is an $f \in \mathcal{F}$ with $f(x) \neq f(y)$. The set $\mathcal{P}$ of polynomials in functions of $\mathcal{F}$ with coefficients in $\mathbb{Q}$ is defined as the set of polynomials $\bigcup_{i=0}^{\infty} \mathbb{Q}[x_1, ..., x_i]$ in finite variables, where we substitute functions $f \in \mathcal{F}$ for $x_1, ..., x_i$.

Theorem 4.4.51 (Stone-Weierstraß Approximation Theorem) *Let X be a locally compact Hausdorff space, let $\mathcal{F} \subseteq \mathcal{C}(X)$ separate points, and let Y be a separable normed space over $\mathbb{R}$.*

(1) The set $\mathcal{P}$ of polynomials in functions of $\mathcal{F}$ with coefficients from $\mathbb{Q}$ is dense in $\mathcal{C}(X)$.

(2) The set $\mathcal{L}$ of all linear combinations of functions from $\mathcal{P}$ with coefficients in a dense subset of Y is dense in $\mathcal{C}(X,Y)$.

The proof can easily be deduced from Bourbaki [Bou66]. In case that (X,d) is a separable metric space with a dense subset $D \subseteq X$, we obtain a canonical set $\mathcal{F} := \{d_q : q \in D\} \subseteq \mathcal{C}(X)$, which separates points. Here $d_x : X \to \mathbb{R}, y \mapsto d(x,y)$ is the *distance function* of $x \in X$. For each separable metric space (X,d,α) we define a dense sequence $\alpha_{\mathcal{C}(X)} : \mathbb{N} \to \mathcal{C}(X)$ by

$$\alpha_{\mathcal{C}(X)}\langle\langle l_{01},...,l_{0j_0},k_0,j_0\rangle,...,\langle l_{n1},...,l_{nj_n},k_n,j_n\rangle,n\rangle := \sum_{i=0}^{n} \alpha_{\mathbb{R}}(k_i)\prod_{j=1}^{j_i} d_{\alpha(l_{ij})}.$$

By definition we assume $\prod_{j=1}^{0} f = \hat{1} : X \to \mathbb{R}, x \mapsto 1$. If, additionally, $(Y,||\ ||,e)$ is a separable Banach space, then we define a dense sequence by $\alpha_{\mathcal{C}(X,Y)} : \mathbb{N} \to \mathcal{C}(X,Y)$ by

$$\alpha_{\mathcal{C}(X,Y)}\langle\langle k_0,l_0\rangle,...,\langle k_n,l_n\rangle,n\rangle := \sum_{i=0}^{n} \alpha_e(k_i)\alpha_{\mathcal{C}(X)}(l_i).$$

If (X,d,α) is a locally compact separable metric space, then by the Stone-Weierstraß Approximation Theorem $(\mathcal{C}(X),d_{\mathcal{C}},\alpha_{\mathcal{C}(X)})$ is a complete separable metric space. If, additionally, $(Y,||\ ||,e)$ is a separable Banach space, then $(\mathcal{C}(X,Y),d_{\mathcal{C}},\alpha_{\mathcal{C}(X,Y)})$ is a complete separable metric space. We will prove now that a corresponding recursive statement holds. Therefore, we will use the following lemma which states that evaluation of the dense sequences is recursive.

Lemma 4.4.52 (Evaluation of dense sequences) *If (X,d,α) is a separable metric space, then $(\alpha_{\mathcal{C}(X)})_* : \mathbb{N} \times X \to \mathbb{R}, (n,x) \mapsto \alpha_{\mathcal{C}(X)}(n)(x)$ is recursive over $\mathbf{X}$. If, additionally, $(Y,||\ ||,e)$ is a separable Banach space, then $(\alpha_{\mathcal{C}(X,Y)})_* : \mathbb{N} \times X \to Y, (n,x) \mapsto \alpha_{\mathcal{C}(X,Y)}(n)(x)$ is recursive over $\mathbf{X} \oplus \mathbf{Y}$.*

Proof. Let (X,d,α) be a separable metric space. For $m = \langle\langle l_{01},...,l_{0j_0},k_0,j_0\rangle,$ $...,\langle l_{n1},...,l_{nj_n},k_n,j_n\rangle,n\rangle$ we obtain

$$(\alpha_{\mathcal{C}(X)})_*(m,x) = \sum_{i=0}^{n} \alpha_{\mathbb{R}}(k_i)\prod_{j=1}^{j_i} d(\alpha(l_{ij}),x).$$

Thus $(\alpha_{\mathcal{C}(X)})_*$ is recursive over $\mathbf{X}$, since $\alpha, d, \alpha_{\mathbb{R}}, +, \cdot$ are so. Let, additionally, $(Y,||\ ||,e)$ be a separable Banach space. For $m = \langle\langle k_0,l_0\rangle,...,\langle k_n,l_n\rangle,n\rangle$ we obtain

$$(\alpha_{\mathcal{C}(X,Y)})_*(m,x) = \sum_{i=0}^{n} \alpha_e(k_i)(\alpha_{\mathcal{C}(X)})_*(l_i,x).$$

Thus $(\alpha_{\mathcal{C}(X,Y)})_*$ is recursive over $\mathbf{X} \oplus \mathbf{Y}$, since $\alpha_e, (\alpha_{\mathcal{C}(X)})_*, +$ are so. $\square$

Now we are ready to prove the following proposition.

Proposition 4.4.53 (Space of continuous functions) *If (X, d, α) is a recursively locally compact recursive metric space, then $(\mathcal{C}(X), d_{\mathcal{C}}, \alpha_{\mathcal{C}(X)})$ is a recursive metric space. If, additionally, $(Y, || \, ||, e)$ is a recursive Banach space, then $(\mathcal{C}(X,Y), d_{\mathcal{C}}, \alpha_{\mathcal{C}(X,Y)})$ is a recursive metric space.*

Proof. Let (X, d, α) be a locally compact recursive metric space with recursive exhausting sequence $(K_i)_{i\in\mathbb{N}}$. It is easy to see that there is a recursive function $s : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that $\alpha_{\mathcal{C}(X)}(n) - \alpha_{\mathcal{C}(X)}(k) = \alpha_{\mathcal{C}(X)} s(n,k)$. With $\sigma : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}, (x_n)_{n\in\mathbb{N}} \mapsto \sum_{i=0}^{\infty} 2^{-i-1} \frac{x_i}{1+|x_i|}$ we obtain

$$\begin{aligned} d_{\mathcal{C}(X)} \circ (\alpha_{\mathcal{C}(X)} \times \alpha_{\mathcal{C}(X)})(n,k) &= \sum_{i=0}^{\infty} 2^{-i-1} \frac{|\alpha_{\mathcal{C}(X)}(n) - \alpha_{\mathcal{C}(X)}(k)|_{K_i}}{1 + |\alpha_{\mathcal{C}(X)}(n) - \alpha_{\mathcal{C}(X)}(k)|_{K_i}} \\ &= \sum_{i=0}^{\infty} 2^{-i-1} \frac{\sup_{x\in K_i} |\alpha_{\mathcal{C}(X)} s(n,k)(x)|}{1 + \sup_{x\in K_i} |\alpha_{\mathcal{C}(X)} s(n,k)(x)|} \\ &= \sigma\left(\sup\left|(\alpha_{\mathcal{C}(X)})_* \left(\{s(n,k)\} \times K_i\right)\right|\right)_{i\in\mathbb{N}} \end{aligned}$$

for all $n, k \in \mathbb{N}$. Thus $d_{\mathcal{C}(X)} \circ (\alpha_{\mathcal{C}(X)} \times \alpha_{\mathcal{C}(X)}) : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$ is recursive over $\mathbf{R}$ by Proposition 4.4.36 and Corollary 4.4.40, since $\sup : \mathcal{K}(\mathbb{R}) \to \mathbb{R}$ is recursive over $\mathcal{K}(\mathbf{R})$ by Proposition 4.4.35, and $| \, |, \sigma$ are recursive over $\mathbf{R}$ by Proposition 4.4.1.

Now let, additionally, $(Y, || \, ||, e)$ be a recursive Banach space. It is easy to see that there is a recursive function $t : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that $\alpha_{\mathcal{C}(X,Y)}(n) - \alpha_{\mathcal{C}(X,Y)}(k) = \alpha_{\mathcal{C}(X,Y)} t(n,k)$. We obtain

$$\begin{aligned} & d_{\mathcal{C}(X,Y)} \circ (\alpha_{\mathcal{C}(X,Y)} \times \alpha_{\mathcal{C}(X,Y)})(n,k) \\ = & \sum_{i=0}^{\infty} 2^{-i-1} \frac{||\alpha_{\mathcal{C}(X,Y)}(n) - \alpha_{\mathcal{C}(X,Y)}(k)||_{K_i}}{1 + ||\alpha_{\mathcal{C}(X,Y)}(n) - \alpha_{\mathcal{C}(X,Y)}(k)||_{K_i}} \\ = & \sum_{i=0}^{\infty} 2^{-i-1} \frac{\sup_{x\in K_i} ||\alpha_{\mathcal{C}(X,Y)} t(n,k)(x)||}{1 + \sup_{x\in K_i} ||\alpha_{\mathcal{C}(X,Y)} t(n,k)(x)||} \\ = & \sigma\left(\sup ||(\alpha_{\mathcal{C}(X,Y)})_* \left(\{t(n,k)\} \times K_i\right)||\right)_{i\in\mathbb{N}} \end{aligned}$$

for all $n, k \in \mathbb{N}$. Thus $d_{\mathcal{C}(X,Y)} \circ (\alpha_{\mathcal{C}(X,Y)} \times \alpha_{\mathcal{C}(X,Y)}) : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$ is recursive over $\mathbf{R}$ by Proposition 4.4.36 and 4.4.39, since $\sup : \mathcal{K}(\mathbb{R}) \to \mathbb{R}$ is recursive over $\mathcal{K}(\mathbf{R})$ by Proposition 4.4.35, $|| \, ||$ is recursive over $\mathbf{Y}$ by definition, and σ is recursive over $\mathbf{R}$ by Proposition 4.4.1. $\square$

It should be noticed that for the proof we have not used the property that $(K_i^{\circ})_{i\in\mathbb{N}}$ is a recursively given sequence of semi-recursive sets. Provisionally, we define

$$\begin{aligned} \mathcal{C}(\mathbf{X}) &:= \mathbf{X} \oplus (\mathcal{C}(X), \alpha_{\mathcal{C}(X)}, \mathrm{id}, d_{\mathcal{C}(X)}, \mathrm{Lim}) \\ \mathcal{C}(\mathbf{X}, \mathbf{Y}) &:= \mathcal{C}(\mathbf{X}) \oplus \mathbf{Y} \oplus (\mathcal{C}(X,Y), \alpha_{\mathcal{C}(X,Y)}, \mathrm{id}, d_{\mathcal{C}(X,Y)}, \mathrm{Lim}) \end{aligned}$$

as standard metric space structures. Later on, we will replace them by equivalent structures. Now we can deduce the following main theorem on the structure of continuous functions which is a direct consequence of Theorem 4.4.6 and the previous proposition.

Theorem 4.4.54 (Structure of continuous functions) *If X is a recursively locally compact recursive metric space and Y is a recursive Banach space, then $\mathcal{C}(\mathbf{X})$ and $\mathcal{C}(\mathbf{X},\mathbf{Y})$ are complete and strongly perfect topological structures. The equality over these structures is recursive.*

Since it is easy to see that $\mathcal{C}(X)$ and $\mathcal{C}(X,\mathbb{R})$ are recursively isomorphic, we will not discuss the case $\mathcal{C}(X)$ separately for the following.

4.4.13 Evaluation of continuous functions

In this section we want to discuss the extension of the evaluation operator. For functions $F :\subseteq Z \to Y^{\mathbb{N}}$ we have defined the evaluation $F_* :\subseteq Z\times\mathbb{N} \to Y$. Now we want to extend this operator to the case of functions $F : Z \to \mathcal{C}(X,Y)$. With each such function we associate its *evaluation* $F_* : Z \times X \to Y$. In the previous section we have already seen that the evaluation of the dense sequence $\alpha_{\mathcal{C}(X,Y)}$ is recursive. The main result of this section roughly states that a function is recursive, if and only if its evaluation is.

Theorem 4.4.55 (Evaluation Theorem) *Let X be a recursively locally compact separable metric space, let Y be a separable Banach space, let Z be a separable metric space, and let S be an extension of $\mathbf{Z}\oplus\mathcal{C}(\mathbf{X},\mathbf{Y})$. Then for each function $F : Z \to \mathcal{C}(X,Y)$ the following is equivalent over S:*

(1) $F : Z \to \mathcal{C}(X,Y)$ is recursive,

(2) $F_* : Z \times X \to Y$ is recursive.

Proof. Let $(K_i)_{i\in\mathbb{N}}$ be a recursive exhausting sequence of X. Let $\delta_{\mathcal{C}}$ be the Cauchy representation of $(\mathcal{C}(X,Y), d_{\mathcal{C}}, \alpha_{\mathcal{C}(X,Y)})$ with recursive right inverse $\delta_{\mathcal{C}}^-$.

"(1)$\Longrightarrow$(2)" By Proposition 4.4.43 there exists a recursive modulus of exhausting $m : X \rightrightarrows \mathbb{N}$ of X w.r.t. $(K_i)_{i\in\mathbb{N}}$. Let $g : Z \times X \times \mathbb{N} \to Y$ be defined by

$$g(z,x,k) := (\alpha_{\mathcal{C}(X,Y)})_*((\delta_{\mathcal{C}}^-)_*(F(z), m(x)+k+3), x).$$

Then g is recursive over S and we claim $F_* = \mathrm{Lim}\circ[g]$ and hence F_* is recursive over S. Therefore, let $z \in Z$, $x \in X$, $f := F(z)$, $n \in m(x)$, and let

$(f_k)_{k\in\mathbb{N}} \in \alpha^{\mathbb{N}}_{\mathcal{C}(X,Y)} \circ \delta^-_{\mathcal{C}}(f)$. Then $x \in K_n$ and $d_{\mathcal{C}}(f, f_k) \le 2^{-k}$ for all $k \in \mathbb{N}$. We obtain

$$\begin{aligned} 2^{-n-1}\frac{||f(x) - f_{k+n+3}(x)||}{1 + ||f(x) - f_{k+n+3}(x)||} &\le 2^{-n-1}\frac{||f - f_{k+n+3}||_{K_n}}{1 + ||f - f_{k+n+3}||_{K_n}} \\ &\le \sum_{i=0}^{\infty} 2^{-i-1}\frac{||f - f_{k+n+3}||_{K_i}}{1 + ||f - f_{k+n+3}||_{K_i}} \\ &\le 2^{-k-n-3}, \end{aligned}$$

thus

$$||f(x) - f_{k+n+3}(x)|| \le \frac{2^{-k-2}}{1 - 2^{-k-2}} < 2^{-k-1}.$$

Hence, $\mathrm{Lim}(f_{k+n+3}(x))_{k\in\mathbb{N}} = f(x)$, which proves the claim.

"(2)$\Longrightarrow$(1)" Let $g : Z \times \mathbb{N} \times X \to Y$ be defined by

$$g(z, n, x) := F_*(z, x) - (\alpha_{\mathcal{C}(X,Y)})_*(n, x).$$

By Lemma 4.4.52 g is recursive over S. Define $h : Z \times \mathbb{N} \to Y$ by

$$\begin{aligned} h(z, n) &:= d_{\mathcal{C}(X,Y)}(F(z), \alpha_{\mathcal{C}(X,Y)}(n)) \\ &= \sum_{i=0}^{\infty} 2^{-i-1}\frac{\sup_{x\in K_i} ||F_*(z, x) - (\alpha_{\mathcal{C}(X,Y)})_*(n, x)||}{1 + \sup_{x\in K_i} ||F_*(z, x) - (\alpha_{\mathcal{C}(X,Y)})_*(n, x)||} \\ &= \sigma\left(\sup ||g\left(\{z\} \times \{n\} \times K_i\right)||\right)_{i\in\mathbb{N}}. \end{aligned}$$

Then h is recursive over S, by Proposition 4.4.36 and 4.4.39, since $\sup : \mathcal{K}(\mathbb{R}) \to \mathbb{R}$ is recursive over $\mathcal{K}(\mathbf{R})$ by Proposition 4.4.35, $||\ ||$ is recursive over $\mathbf{Y}$ by definition, and σ is recursive over $\mathbf{R}$ by Proposition 4.4.1. Thus,

$$A := \{(z, n, k) : h(z, n) < 2^{-k-1}\} \subseteq Z \times \mathbb{N} \times \mathbb{N}$$

is semi-recursive over S and by uniformization there is a recursive operation $f : Z \times \mathbb{N} \rightrightarrows \mathbb{N}$ over S such that $\mathrm{graph}(f) = A$. Hence, $F = \mathrm{Lim}_{\mathcal{C}(X,Y)} \circ \alpha^{\mathbb{N}}_{\mathcal{C}(X,Y)} \circ [f]$ is recursive over S. $\square$

The Evaluation Theorem has many interesting consequences. It can be compared to the smn- and utm-Theorem in classical recursion theory (cf. [Rog67, Odi89, Wei87]). We can immediately derive a series of important corollaries. If we define the *evaluation function* $\mathrm{ev}_{\mathcal{C}(X,Y)} : \mathcal{C}(X, Y) \times X \to Y, (f, x) \mapsto f(x)$, then we can conclude that $\mathrm{ev}_{\mathcal{C}(X,Y)}$ is recursive since $\mathrm{ev}_{\mathcal{C}(X,Y)} = (\mathrm{id}_{\mathcal{C}(X,Y)})_*$.

Corollary 4.4.56 (Evaluation function) *Let X be a recursively locally compact separable metric space and let Y be a separable Banach space. Then $\mathrm{ev}_{\mathcal{C}(X,Y)} : \mathcal{C}(X,Y) \times X \to Y, (f,x) \mapsto f(x)$ is recursive over $\mathcal{C}(\mathbf{X},\mathbf{Y})$.*

Another direct consequence shows that the recursive points in $\mathcal{C}(X,Y)$ are exactly the recursive functions.

Corollary 4.4.57 (Recursive functions as points) *Let X be a recursively locally compact separable metric space and let Y be a separable Banach space. Then over $\mathcal{C}(\mathbf{X},\mathbf{Y})$ the following holds: $f \in \mathcal{C}(X,Y)$ is a recursive point, if and only if $f : X \to Y$ is a recursive function.*

The next corollary states that composition is a recursive operation on continuous functions.

Corollary 4.4.58 (Composition) *Let X be a separable metric space, Y, Z separable Banach spaces, and let X, Y be recursively locally compact. Then the composition $\circ : \mathcal{C}(X,Y) \times \mathcal{C}(Y,Z) \to \mathcal{C}(X,Z), (f,g) \mapsto g \circ f$ is recursive over $\mathcal{C}(\mathbf{X},\mathbf{Y}) \oplus \mathcal{C}(\mathbf{Y},\mathbf{Z}) \oplus \mathcal{C}(\mathbf{X},\mathbf{Z})$.*

The proof follows from $g \circ f(x) = \mathrm{ev}_{\mathcal{C}(Y,Z)}(g, \mathrm{ev}_{\mathcal{C}(X,Y)}(f,x))$. With each continuous function $f : X \to Y$ we can associate its *dual function* $f^\star : \mathcal{C}(Y) \to \mathcal{C}(X), g \mapsto g \circ f$. The following corollary states that the dual of a recursive function is recursive too.

Corollary 4.4.59 (Dual) *Let X, Y be recursively locally compact separable metric spaces. If $f : X \to Y$ is recursive over $\mathbf{X} \oplus \mathbf{Y}$, then the dual function $f^\star : \mathcal{C}(Y) \to \mathcal{C}(X), g \mapsto g \circ f$ is recursive over $\mathcal{C}(\mathbf{X}) \oplus \mathcal{C}(\mathbf{Y})$.*

The proof follows from $(f^\star)_*(g,x) = \mathrm{ev}_{\mathcal{C}(Y)}(g, f(x))$. In the same way as we have extended the Evaluation from sequences to continuous functions we can extend exponentiation. With each function $f : X \to Y$ and each locally compact separable metric space Z we can associate the exponentiation $f^Z : \mathcal{C}(Z,X) \to \mathcal{C}(Z,Y)$, defined by $f^Z(g)(z) := f \circ g(z)$. The following corollary states that a function is recursive, if and only if its exponentiation is.

Corollary 4.4.60 (Exponentiation) *Let Z be a recursively locally compact separable metric space, let X, Y be separable Banach spaces and let S be an extension of $\mathcal{C}(\mathbf{Z},\mathbf{X}) \oplus \mathcal{C}(\mathbf{Z},\mathbf{Y})$. Then for each function $f : X \to Y$ the following is equivalent over S:*

(1) $f : X \to Y$ *is recursive,*

(2) $f^Z : \mathcal{C}(Z,X) \to \mathcal{C}(Z,Y), g \mapsto f \circ g(z)$ *is recursive.*

The proof follows from $(f^Z)_*(g,z) = f \circ \mathrm{ev}_{\mathcal{C}(Z,X)}(g,z)$ and $f(x) = (f^Z \circ c)_*(x,z')$, where $c : X \to \mathcal{C}(Z,X)$ is the recursive function, defined by $c_*(x,z) := x$ and $z' \in Z$ is some recursive constant.

4.4.14 The structure of continuous functions

In this section we will replace the standard structures $\mathcal{C}(\mathbf{X})$ and $\mathcal{C}(\mathbf{X},\mathbf{Y})$ by equivalent ones. Therefore, the quite technical dense sequences will be expressed in terms of the algebraic structure of these spaces. First we prove that some elementary functions are recursive over the space of continuous functions since their pointwise counterparts are recursive over the underlying spaces.

Lemma 4.4.61 (Elementary functions) *If X is a recursively locally compact metric space, then the following functions are recursive over $\mathcal{C}(\mathbf{X})$:*

(1) $\mathbb{R} \times \mathcal{C}(X) \to \mathcal{C}(X), (y, f) \mapsto y \cdot f,$

(2) $\mathcal{C}(X) \times \mathcal{C}(X) \to \mathcal{C}(X), (f, g) \mapsto f + g,$

(3) $\mathcal{C}(X) \times \mathcal{C}(X) \to \mathcal{C}(X), (f, g) \mapsto f \cdot g,$

If, additionally, Y is a separable Banach space, then the following functions are recursive over $\mathcal{C}(\mathbf{X}) \oplus \mathcal{C}(\mathbf{X},\mathbf{Y})$:

(1) $Y \times \mathcal{C}(X) \to \mathcal{C}(X, Y), (y, f) \mapsto y \cdot f,$

(2) $\mathcal{C}(X,Y) \times \mathcal{C}(X,Y) \to \mathcal{C}(X,Y), (f, g) \mapsto f + g,$

(3) $\mathcal{C}(X,Y) \times \mathcal{C}(X,Y) \to \mathcal{C}(X,Y), (f, g) \mapsto f \cdot g.$

The lemma can easily be proved by the Evaluation Theorem 4.4.55. Now we will define our canonical structures for the space of continuous functions. In case that (X, d) is a locally compact separable metric space we will use the following structure:

$$\boxed{\mathcal{C}(\mathbf{X}) := \mathbf{X} \oplus (\mathcal{C}(X), 1, d_x, f, y \cdot f, f + g, f \cdot g, d_{\mathcal{C}}, \mathrm{Lim}).}$$

More precisely, 1 denotes the constant function $\{()\} \to \mathcal{C}(X), () \mapsto \hat{1}$ with $\hat{1} : X \to \mathbb{R}, x \mapsto 1$, d_x denotes the function $X \to \mathcal{C}(X), x \mapsto d_x$, where d_x denotes the distance function of $x \in X$, f denotes the identity $\mathcal{C}(X) \to \mathcal{C}(X)$, $y \cdot f$ the scalar multiplication $\mathbb{R} \times \mathcal{C}(X) \to \mathcal{C}(X), (y, f) \mapsto y \cdot f$, $f + g$ denotes the addition $\mathcal{C}(X) \times \mathcal{C}(X) \to \mathcal{C}(X)$, analogously $f \cdot g$ denotes the multiplication.

If, additionally, $(Y, ||\ ||)$ is a separable Banach space, we will use the following structure:

$$\boxed{\mathcal{C}(\mathbf{X},\mathbf{Y}) := \mathcal{C}(\mathbf{X}) \oplus \mathbf{Y} \oplus (\mathcal{C}(X,Y), f, y \cdot f, f + g, d_{\mathcal{C}}, \mathrm{Lim}).}$$

Here, $y \cdot f$ denotes the function $Y \times \mathcal{C}(X) \to \mathcal{C}(X,Y), (y,f) \mapsto y \cdot f$. In both structures $d_{\mathcal{C}}$ denotes the corresponding metric defined above and Lim the corresponding limit operator. Now we still have to prove that our notation is not ambiguous, that is our new structure of the continuous functions is strongly equivalent to the standard metric space structure of the continuous functions, which we have used before.

Proposition 4.4.62 (Standard structure of continuous functions) *If (X, d, α) is a recursively locally compact separable metric space, then*

$$\mathcal{C}(\mathbf{X}) \equiv_s \mathbf{X} \oplus (\mathcal{C}(X), \alpha_{\mathcal{C}(X)}, \mathrm{id}, d_{\mathcal{C}(X)}, \mathrm{Lim}).$$

If, additionally, $(Y, ||\ ||, e)$ is a separable Banach space, then

$$\mathcal{C}(\mathbf{X}, \mathbf{Y}) \equiv_s \mathcal{C}(\mathbf{X}) \oplus \mathbf{Y} \oplus (\mathcal{C}(X,Y), \alpha_{\mathcal{C}(X,Y)}, \mathrm{id}, d_{\mathcal{C}(X,Y)}, \mathrm{Lim}).$$

Proof. Let $\mathcal{C}(\mathbf{X})' := \mathbf{X} \oplus (\mathcal{C}(X), \alpha_{\mathcal{C}(X)}, \mathrm{id}, d_{\mathcal{C}}, \mathrm{Lim})$ and $\mathcal{C}(\mathbf{X},\mathbf{Y})' := \mathcal{C}(\mathbf{X}) \oplus \mathbf{Y} \oplus (\mathcal{C}(X,Y), \alpha_{\mathcal{C}(X,Y)}, \mathrm{id}, d_{\mathcal{C}}, \mathrm{Lim})$. From the definition it follows that $\alpha_{\mathcal{C}(X)}$, $\alpha_{\mathcal{C}(X,Y)}$ are recursive over $\mathcal{C}(\mathbf{X})', \mathcal{C}(\mathbf{X},\mathbf{Y})'$, respectively. For the other direction we can apply Lemma 4.4.61, which shows that $y \cdot f$, $f + g$, (and $f \cdot g$) are recursive over $\mathcal{C}(\mathbf{X},\mathbf{Y})'$ ($\mathcal{C}(\mathbf{X})'$, respectively). It remains to prove that $1 : \{()\} \to \mathcal{C}(X)$ and $X \to \mathcal{C}(X), x \mapsto d_x$ are recursive over $\mathcal{C}(\mathbf{X})'$. It is easy to see that there is an $m \in \mathbb{N}$ and there is a recursive functions r over $\mathbf{N}$ such that $\alpha_{\mathcal{C}(X)}(m) = \hat{1}$, $d_{\alpha(n)} = \alpha_{\mathcal{C}(X)} r(n)$ for all $n \in \mathbb{N}$. This proves that 1 is recursive over $\mathcal{C}(X)'$. Since $d_{\mathcal{C}}(d_x, d_y) \leq \sup_{z \in \mathbb{R}} |d_x(z) - d_y(z)| \leq d(x,y)$, it follows that $x \mapsto d_x$ is Lipschitz continuous and hence recursive over $\mathcal{C}(\mathbf{X})'$ by Proposition 4.4.26. $\square$

At the end of this section we just want to mention the important special case $\mathcal{C}[0,1]$. In this case we can use the following structure:

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

4.4.15 Recursive partition of unity

In this section we want to show that for recursively locally compact recursive metric spaces X there exists a partition of unity which is subordinated to the exhausting sequence of the space. First we have to introduce some further notions. A covering $(A_i)_{i \in \mathbb{N}}$ of X is called *locally finite covering*, if for each $x \in X$ there is a neighbourhood U of x such that $A_i \cap U \neq \emptyset$ for at most finitely many $i \in \mathbb{N}$. For each function $f : X \to \mathbb{R}$ we will call

$$\mathrm{supp}(f) := \overline{\{x \in X : f(x) \neq 0\}}$$

the *support* of f. Now we can define recursive partitions of unity.

Definition 4.4.63 (Recursive partition of unity) Let X be a recursive metric space. A sequence $(\pi_i)_{i\in\mathbb{N}}$ of non-negative functions $\pi_i : X \to \mathbb{R}$ is called a *recursive partition of unity* of X, if

(1) $(\pi_i)_{i\in\mathbb{N}}$ is a recursive sequence over $\mathcal{C}(\mathbf{X})$,

(2) $(\mathrm{supp}(\pi_i))_{i\in\mathbb{N}}$ is a locally finite covering of X,

(3) $\sum_{i=0}^{\infty} \pi_i(x) = 1$ for all $x \in X$.

If $(U_i)_{i\in\mathbb{N}}$ is a covering of X, then $(\pi_i)_{i\in\mathbb{N}}$ is called *subordinated* to $(U_i)_{i\in\mathbb{N}}$, if $\mathrm{supp}(\pi_i) \subseteq U_i$ for all $i \in \mathbb{N}$.

Now we can formulate the main result of this section about partitions of unity. The proof is somewhat similar to the proof of Proposition 4.4.47.

Theorem 4.4.64 (Recursive partition of unity) *If X is a recursively locally compact recursive metric space with recursive exhausting sequence $(K_i)_{i\in\mathbb{N}}$, then there exists a recursive partition of unity $(\pi_i)_{i\in\mathbb{N}}$ of X which is subordinated to $(K_i^\circ)_{i\in\mathbb{N}}$.*

Proof. By Lemma 4.4.44 the sequences $(K_{i+1}^\circ)_{i\in\mathbb{N}}$ and $(K_i^c)_{i\in\mathbb{N}}$ are recursively given sequences of semi-recursive sets over $\mathbf{X}$. By Proposition 4.4.29 there are recursive functions $f, g : X \times \mathbb{N} \to \mathbb{R}$ over $\mathbf{X}$ such that $\overline{K_{i+1}^c} = \{x \in X : f(x,i) = 0\}$ and $K_i = \{x \in X : g(x,i) = 0\}$. We define functions $h_i : X \to \mathbb{R}$ by $h_i(x) := |f(x,i)|/(|f(x,i)| + |g(x,i)|)$ for all $x \in X$ and $i \in \mathbb{N}$. Then $(h_i)_{i\in\mathbb{N}}$ is a recursive sequence over $\mathcal{C}(\mathbf{X})$ by the Evaluation Theorem 4.4.55 and each h_i is total since $\overline{K_{i+1}^c} \cap K_i = \emptyset$ for all $i \in \mathbb{N}$. For technical reasons we define $h_{-1} := h_{-2} := 0$ and $K_{-1} := \emptyset$. Now let $\pi_i : X \to \mathbb{R}$ be defined by

$$\begin{cases} \pi_0 & := & 0 \\ \pi_{i+1} & := & h_{i-1} - h_{i-2} \end{cases}$$

for all $i \in \mathbb{N}$. Then $(\pi_i)_{i\in\mathbb{N}}$ is a recursive sequence over $\mathcal{C}(\mathbf{X})$ too. We obtain $h_i^{-1}\{0\} = \overline{K_{i+1}^c}$ and $h_i^{-1}\{1\} = K_i$ for all $i \in \mathbb{N}$ and hence

$$\mathrm{supp}(h_i) = \overline{\{x \in X : h_i(x) \neq 0\}} = \overline{\overline{K_{i+1}^c}^{\,c}} = \overline{K_{i+1}^\circ} \subseteq K_{i+1}$$

for all $i \in \mathbb{N}$. Especially, $\mathrm{supp}(h_i) \subseteq K_{i+1} = h_{i+1}^{-1}\{1\}$ implies that $\pi_i : X \to \mathbb{R}$ is non-negative and $\mathrm{supp}(\pi_i) \subseteq \mathrm{supp}(h_{i-2}) \subseteq K_{i-1} \subseteq K_i^\circ$, for all $i \in \mathbb{N}$. Hence, $(\pi_i)_{i\in\mathbb{N}}$ is subordinated to $(K_i^\circ)_{i\in\mathbb{N}}$.

Moreover, we obtain $K_i = h_i^{-1}\{1\} \subseteq \pi_{i+3}^{-1}\{0\}$ and hence $K_i^\circ \subseteq \mathrm{supp}(\pi_{i+3})^c$ for all $i \in \mathbb{N}$. Thus, $\mathrm{supp}(\pi_i) \subseteq K_i^\circ \subseteq K_{i+j}^\circ \subseteq \mathrm{supp}(\pi_{i+j+3})^c$ for all $i, j \in \mathbb{N}$. Hence, $(\mathrm{supp}(\pi_i))_{i\in\mathbb{N}}$ is locally finite.

Finally, let $x \in X = \bigcup_{i=0}^\infty K_i$ and let $m := \min\{i \in \mathbb{N} : x \in K_i\}$. Then $x \in K_i = h_i^{-1}\{1\}$ if and only if $i \geq m$. Especially, $x \in \mathrm{supp}(\pi_{m+2})$ and $(\mathrm{supp}(\pi_i))_{i\in\mathbb{N}}$ is a covering of X. Since $x \in \overline{K_{m-1}^c} = h_{m-2}^{-1}\{0\}$ and $\pi_i(x) = 0$ for $i < m+1$ and $i \geq m+3$ it follows that

$$\sum_{i=0}^\infty \pi_i(x) = \pi_{m+1}(x) + \pi_{m+2}(x) = h_m(x) - h_{m-2}(x) = 1$$

for all $x \in X$. Altogether, $(\pi_i)_{i\in\mathbb{N}}$ is a recursive partition of unity of X. □

Now we will apply the partition of unity to prove that a recursively locally compact space admits a *compactifier*. We will apply this result in section 4.5.

Proposition 4.4.65 (Compactifier) *If X is a recursively locally compact recursive metric space with recursive exhausting sequence $(K_i)_{i\in\mathbb{N}}$, then there exists a recursive function $m : X \to \mathbb{R}$ over $\mathbf{X}$ such that $0 < m(x) \leq 2^{-k}$ for all $x \in X$ and $k := \min\{i \in \mathbb{N} : x \in K_i\}$.*

Proof. By the previous Partition Theorem there exists a recursive partition of unity $(\pi_i)_{i\in\mathbb{N}}$ of X which is subordinated to $(K_i^\circ)_{i\in\mathbb{N}}$. We define $m : X \to \mathbb{R}$ by

$$m(x) := \sum_{i=0}^\infty 2^{-i-1}\pi_i(x)$$

for all $x \in X$. Then m is well-defined since $\pi_i(x) \leq 1$ for all $x \in X$, $m(x) > 0$ for all $x \in X$ since $\sum_{i=0}^\infty \pi_i(x) = 1$ and m is recursive over $\mathbf{X}$. Now, let $x \in X$, $k := \min\{i \in \mathbb{N} : x \in K_i\}$. Then $m(x) = \sum_{i=k}^\infty 2^{-i-1}\pi_i(x) \leq 2^{-k}$, since $(\pi_i)_{i\in\mathbb{N}}$ is subordinated to $(K_i^\circ)_{i\in\mathbb{N}}$ and $\pi_i(x) \leq 1$ for all $x \in X$. □

4.4.16 The structure of closed subsets

In this section we will consider the set $\mathcal{A}(X)$ of non-empty closed[3] subsets of a topological space X. In case that (X, d) is a nice locally compact separable metric space with exhausting sequence $(K_i)_{i\in\mathbb{N}}$, $\mathcal{A}(X)$ is metrizable and the following Fréchet combination defines a metric

$$d_{\mathcal{A}} : \mathcal{A}(X) \times \mathcal{A}(X) \to \mathbb{R}, (A, B) \mapsto \sum_{i=0}^\infty 2^{-i-1}|d_A - d_B|_{K_i}$$

[3]The notation $\mathcal{A}(X)$ is motivated by the german translation “abgeschlossen” of “closed”

which induces the *Fell topology* on $\mathcal{A}(X)$. Here $d_A : X \to \mathbb{R}, x \mapsto \inf_{a\in A} d(x,a)$ denotes the *distance function* of the set $A \subseteq X$. The Fell topology has as a subbase all sets

$$\begin{aligned} U^- &:= \{A \in \mathcal{A}(X) : A \cap U \neq \emptyset\},\ U \subseteq X \text{ open} \\ (K^c)^+ &:= \{A \in \mathcal{A}(X) : A \cap K = \emptyset\},\ K \subseteq X \text{ compact,} \end{aligned}$$

where U^- is called *hit set* of U and $(K^c)^+$ is called *miss set* of K^c (cf. [Bee93] for the Fell topology). From now on we will assume that $\mathcal{A}(X)$ is endowed with the Fell topology.

For the construction of the metric $d_{\mathcal{A}}$ it is important that d is nice because this property guarantees that the values of the distance function d_A in K_i do not depend on the elements of A which are not in K_{i+1} (provided that $A \cap K_{i+1} \neq \emptyset$). In other words, for nice metrics localness of sets corresponds to localness of distance functions.

If (X,d) is a nice locally compact metric space, then $(\mathcal{A}(X), d_{\mathcal{A}})$ is a complete metric space, and if Q is a dense subset of X, then the set $\mathcal{F}(Q)$ of non-empty finite subsets of Q is dense in $\mathcal{A}(X)$. Consequently, we will associate with each sequence $\alpha : \mathbb{N} \to X$ with range Q a canonical sequence $\alpha_{\mathcal{A}} : \mathbb{N} \to \mathcal{A}(X)$ with range $\mathcal{F}(Q)$, which is defined by $\alpha_{\mathcal{A}}(n) := \alpha_{\mathcal{K}}(n)$ (cf. Section 4.4.7). Thus, if (X,d,α) is a separable metric space, then $(\mathcal{A}(X), d_{\mathcal{A}}, \alpha_{\mathcal{A}})$ is so too. We will prove now that the same holds for recursive metric spaces. The proof mainly relies on the fact that $d_{\mathcal{A}}$ is closely related to $d_{\mathcal{C}}$ via the distance operation $\mathrm{dist} : \mathcal{A}(X) \to \mathcal{C}(X), A \mapsto d_A$. More precisely, we obtain

$$\tfrac{1}{2} d_{\mathcal{A}}(A,B) \leq d_{\mathcal{C}}(d_A, d_B) \leq d_{\mathcal{A}}(A,B) \tag{4.1}$$

for all $A, B \in \mathcal{A}(X)$, since d is nice and thus $d \leq 1$.

Proposition 4.4.66 (Space of closed subsets) *If (X,d,α) is a nice recursively locally compact recursive metric space, then $(\mathcal{A}(X), d_{\mathcal{A}}, \alpha_{\mathcal{A}})$ is a recursive metric space.*

Proof. Let $(K_i)_{i\in\mathbb{N}}$ be a recursive exhausting sequence of X. By Equation 4.1 it follows that $\mathrm{dist} : \mathcal{A}(X) \to \mathcal{C}(X), A \mapsto d_A$ is Lipschitz continuous. Since $x \mapsto d_x$ is recursive over $\mathcal{C}(\mathbf{X})$ and $d_{A\cup\{a\}} = \min \circ (d_A, d_a)$ for all $A \subseteq X$ and $a \in X$, and min is recursive over $\mathbf{R}$, it follows that $\mathrm{dist} \circ \alpha_{\mathcal{A}} : \mathbb{N} \to \mathcal{C}(X)$ is recursive over $\mathcal{C}(\mathbf{X})$. Analogously to the proof of Proposition 4.4.53 one can show that $d_{\mathcal{A}} \circ (\alpha_{\mathcal{A}} \times \alpha_{\mathcal{A}})$ is recursive over $\mathbb{R}$, since

$$d_{\mathcal{A}} \circ (\alpha_{\mathcal{A}} \times \alpha_{\mathcal{A}})(n,k) = \sum_{i=0}^{\infty} |\mathrm{dist} \circ \alpha_{\mathcal{A}}(n) - \mathrm{dist} \circ \alpha_{\mathcal{A}}(k)|_{K_i}$$

for all $n, k \in \mathbb{N}$. □

In case that (X, d) is a nice locally compact separable metric space we will consider the following structure for the space of non-empty closed subsets:

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

where $\{x\}$ denotes the canonical injection $\mathrm{in} : X \to \mathcal{A}(X), x \mapsto \{x\}$, A denotes the identity on $\mathcal{A}(X)$, and $A \cup B$ denotes the union operation $\cup : \mathcal{A}(X) \times \mathcal{A}(X) \to \mathcal{A}(X), (A, B) \mapsto A \cup B$. Moreover, Lim denotes the standard limit operation of the metric space $(\mathcal{A}(X), d_{\mathcal{A}})$. Our main result about this structure will be the following theorem.

Theorem 4.4.67 (Structure of closed subsets) *If X is a nice recursively locally compact recursive metric space, then $\mathcal{A}(\mathbf{X})$ is a complete and strongly perfect topological structure and equality is recursive over this structure.*

The proof follows from Theorem 4.4.6 if we show that our notation $\mathcal{A}(\mathbf{X})$ for the structure above is not ambiguous, that is $\mathcal{A}(\mathbf{X})$ is strongly equivalent to the standard structure of the separable metric space $(\mathcal{A}(X), d_{\mathcal{A}}, \alpha_{\mathcal{A}})$.

Proposition 4.4.68 (Standard structure of closed subsets) *If (X, d, α) is a nice recursively locally compact separable metric space, then $\mathcal{A}(\mathbf{X}) \equiv_s \mathbf{X} \oplus (\mathcal{A}(X), \alpha_{\mathcal{A}}, \mathrm{id}, d_{\mathcal{A}}, \mathrm{Lim})$.*

Proof. Let $\mathcal{A}(\mathbf{X})' := \mathbf{X} \oplus (\mathcal{A}(X), \alpha_{\mathcal{A}}, \mathrm{id}, d_{\mathcal{A}}, \mathrm{Lim})$. We have to show that $\alpha_{\mathcal{A}}$ is recursive over $\mathcal{A}(\mathbf{X})$ and that $\mathrm{in}, \cup$ are recursive over $\mathcal{A}(\mathbf{X})'$. The first follows directly, in the same way as we have shown that $\alpha_{\mathcal{K}}$ is recursive over $\mathcal{K}(\mathbf{X})$. The latter follows since it is easy to see that there are recursive functions f, g over $\mathbf{N}$, such that $\mathrm{in} \circ \alpha = \alpha_{\mathcal{A}} \circ f$ and $\cup \circ (\alpha_{\mathcal{A}} \times \alpha_{\mathcal{A}}) = (\alpha_{\mathcal{A}} \times \alpha_{\mathcal{A}}) \circ g$. Thus, $\mathrm{in} \circ \alpha, \cup \circ (\alpha_{\mathcal{A}} \times \alpha_{\mathcal{A}})$ are recursive over $\mathcal{A}(\mathbf{X})'$. Moreover, due to Equation 4.1 we obtain $1/2 \cdot d_{\mathcal{A}}(\{x\}, \{y\}) \leq d_{\mathcal{C}}(d_x, d_y) \leq d(x, y)$ for all $x, y \in X$ and $d_{\mathcal{A}}(A \cup B, A' \cup B') \leq \max\{d_{\mathcal{A}}(A, A'), d_{\mathcal{A}}(B, B')\}$ for all $A, A', B, B' \in \mathcal{A}(X)$, thus $\mathrm{in}, \cup$ are Lipschitz continuous (w.r.t. the product metric on $\mathcal{A}(X) \times \mathcal{A}(X)$) and hence they are recursive over $\mathcal{A}(\mathbf{X})'$ by Proposition 4.4.26. □

Implicitly, we have already shown that the set $\{d_A : A \in \mathcal{A}(X)\} \subseteq \mathcal{C}(X)$ of distance functions is recursively isomorphic to $\mathcal{A}(X)$.

Proposition 4.4.69 (Distance function) *If X is a nice recursively locally compact separable metric space, then* $\mathrm{dist} : \mathcal{A}(X) \to \mathcal{C}(X), A \mapsto d_A$ *is a recursive embedding over $\mathcal{A}(\mathbf{X}) \oplus \mathcal{C}(\mathbf{X})$.*

Proof. Due to Equation 4.1 we know that dist and dist^{-1} are Lipschitz continuous. In the proof of Proposition 4.4.66 we have already seen that $\text{dist} \circ \alpha_{\mathcal{A}} : \mathbb{N} \to \mathcal{C}(X)$ is recursive over $\mathcal{C}(\mathbf{X})$. Thus, by Proposition 4.4.26 dist is recursive and by the Inversion Theorem 4.4.27 it follows that dist^{-1} is recursive over $\mathcal{A}(\mathbf{X}) \oplus \mathcal{C}(\mathbf{X})$. □

The space of compact subsets $\mathcal{K}(X)$ is not recursively isomorphic to the corresponding subset of $\mathcal{A}(X)$ in general. Nevertheless, the following holds.

Proposition 4.4.70 (Compact and closed subsets) *If X is a nice recursively locally compact separable metric space, then $\text{id}' : \mathcal{K}(X) \to \mathcal{A}(X), A \mapsto A$ is recursive over $\mathcal{K}(\mathbf{X}) \oplus \mathcal{A}(\mathbf{X})$.*

Proof. Since $d_{\mathcal{A}}(A, B) \leq |d_A - d_B|_{A \cup B} = d_{\mathcal{K}}(A, B)$ for all $A, B \in \mathcal{K}(X)$, it follows that id' is Lipschitz continuous. Since $\text{id}' \circ \alpha_{\mathcal{K}} = \alpha_{\mathcal{A}}$ is recursive over $\mathcal{A}(\mathbf{X})$, we can conclude that id' is recursive over $\mathcal{K}(\mathbf{X}) \oplus \mathcal{A}(\mathbf{X})$ by Proposition 4.4.26. □

Although id' is recursive it is not a recursive embedding in general, as the following example shows: the function $\sup :\subseteq \mathcal{A}(\mathbb{R}) \to \mathbb{R}$ restricted to the compact subsets is not even continuous, if $\mathcal{A}(\mathbb{R})$ is endowed with the Fell topology but $\sup : \mathcal{K}(\mathbb{R}) \to \mathbb{R}$ is recursive over $\mathcal{K}(\mathbf{R})$ by Proposition 4.4.35.

4.5 Order-free recursion over the real numbers

In this section we want to investigate whether we need proper operations or whether functions suffice to generate all recursive functions $f : \mathbb{R}^n \to \mathbb{R}$. More precisely, we want to investigate whether all recursive functions $f : \mathbb{R}^n \to \mathbb{R}$ over $\mathbf{R}$ are recursive over

$$\boxed{\mathbf{R}^\star := \mathbf{N} \oplus (\mathbb{R}, 0, 1, x + y, -x, x \cdot y, 1/x, \text{Lim})}$$

too. Here $\mathbf{R}^\star$ differs from $\mathbf{R}$ since the test $<$ is missing and thus, all initial operations of $\mathbf{R}^\star$ are functions. We do not care about the question whether $\mathbf{R}^\star$ is perfect or not (and we cannot expect that it is perfect) but we will focus attention on the functions that are recursive over $\mathbf{R}^\star$. Surprisingly, we find that at least all total recursive functions $f : \mathbb{R}^n \to \mathbb{R}$ over $\mathbf{R}$ are recursive over $\mathbf{R}^\star$ too. On the first sight, one should expect the opposite since tests seem to be inevitable. For instance, the power series expansion of the exponential function $\exp(x) = \sum_{i=0}^{\infty} \frac{x^i}{i!}$ can be used to compute the exponential function; but in order to compute $\exp(x)$ up to precision 2^{-k} we first have to determine an index n up to which the sum has to be evaluated. Since the series does not converge uniformly, the value n depends on k and x and cannot be computed with a function. Nevertheless, there is a recursive operation $\mathbb{R} \times \mathbb{N} \rightrightarrows \mathbb{N}$ which computes such a value, but this operation is recursive over $\mathbf{R}$ and it will hardly be recursive over $\mathbf{R}^\star$. Thus, the proof that exp is recursive over $\mathbf{R}^\star$ has to follow a different line. It will be based on an idea of Shepherdson [She76] who observed that the effective version of the Weierstraß Approximation Theorem, independently proved by Pour-El and Caldwell [PEC75] and Hauck [Hau76], implies that functions $f : [0,1] \to \mathbb{R}$ have straight-line programs. First we will formulate a suitable version of the Weierstraß Approximation Theorem.

Proposition 4.5.1 (Order-free approximation) *Let $f : \mathbb{R}^n \to \mathbb{R}$, $I : \mathbb{R} \to \mathbb{R}$ be recursive over $\mathbf{R}$. Then there exists a sequence of functions $(f_n)_{n\in\mathbb{N}}$ in $\mathcal{C}(\mathbb{R}^n)$ such that $F : \mathbb{R}^n \times \mathbb{N} \to \mathbb{R}, (x,k) \mapsto f_k(x)$ is recursive over $\mathbf{R}^\star$ and $d_{\mathcal{C}}(If, If_k) < 2^{-k-1}$ for all $k \in \mathbb{N}$.*

Proof. It is easy to see that the enumeration of rational polynomials $\alpha : \mathbb{N} \to \mathcal{C}(\mathbb{R}^n)$, defined by

$$\alpha\langle\langle l_{01}, ..., l_{0n}, k_0\rangle, ..., \langle l_{m1}, ..., l_{mn}, k_m\rangle, m\rangle := \sum_{i=0}^{m} \alpha_{\mathbb{R}}(k_i) \prod_{j=1}^{n} x_j^{l_{ij}}$$

is dense in $\mathcal{C}(\mathbb{R}^n)$ and recursive over $\mathcal{C}(\mathbf{R}^n)$. Since f, I are recursive over $\mathbf{R}$, the set

$$A := \{(k,m) \in \mathbb{N} \times \mathbb{N} : d_{\mathcal{C}}(If, I\alpha(m)) < 2^{-k-1}\}$$

is semi-recursive over the perfect structure $\mathcal{C}(\mathbf{R}^n)$ and thus over $\mathbf{N}$. Hence, there is a recursive function $g : \mathbb{N} \to \mathbb{N}$ over $\mathbf{N}$ such that $\mathrm{graph}(g) \subseteq A$. Let $f_k := \alpha g(k)$ for all $k \in \mathbb{N}$. We obtain

$$F(x,k) = f_k(x) = \alpha g(k)(x) = \sum_{i=0}^{m} \alpha_{\mathbb{R}}(k_i) \prod_{j=1}^{n} x_j^{l_{ij}}$$

for all $x \in \mathbb{R}^n$, $k \in \mathbb{N}$ and $\langle\langle l_{01}, ..., l_{0n}, k_0\rangle, ..., \langle l_{m1}, ..., l_{mn}, k_m\rangle, m\rangle := g(k)$. Thus, F is recursive over $\mathbf{R}^\star$ since $\alpha_{\mathbb{R}}$ and the arithmetic operations are recursive over $\mathbf{R}^\star$. $\square$

Analogously, one can prove the following more specialized version of order-free approximation for compact subsets.

Corollary 4.5.2 (Order-free approximation on compact sets) *Let $f : \mathbb{R}^n \to \mathbb{R}$ be recursive over* $\mathbf{R}$ *and $K \in \mathcal{K}(\mathbb{R}^n)$ be recursive over $\mathcal{K}(\mathbf{R}^n)$. Then there exists a sequence of functions $(f_n)_{n\in\mathbb{N}}$ in $\mathcal{C}(\mathbb{R}^n)$ such that $F : \mathbb{R}^n \times \mathbb{N} \to \mathbb{R}, (x,k) \mapsto f_k(x)$ is recursive over $\mathbf{R}^\star$ and $||f - f_k||_K < 2^{-k-1}$ for all $k \in \mathbb{N}$ and thus there is a recursive function $f' :\subseteq \mathbb{R}^n \to \mathbb{R}$ over $\mathbf{R}^\star$ with $f'(x) = f(x)$ for all $x \in K$.*

The function f' can easily be obtained by $f' := \mathrm{Lim} \circ [F]$, First, we will use this specialized version of order-free approximation to prove directly that some elementary functions are recursive over $\mathbf{R}^\star$. The main idea of our proof will be to "compress" the graph of the function into a compact rectangle and to apply the approximation theorem to the compressed function (cf. Figure 4.1).

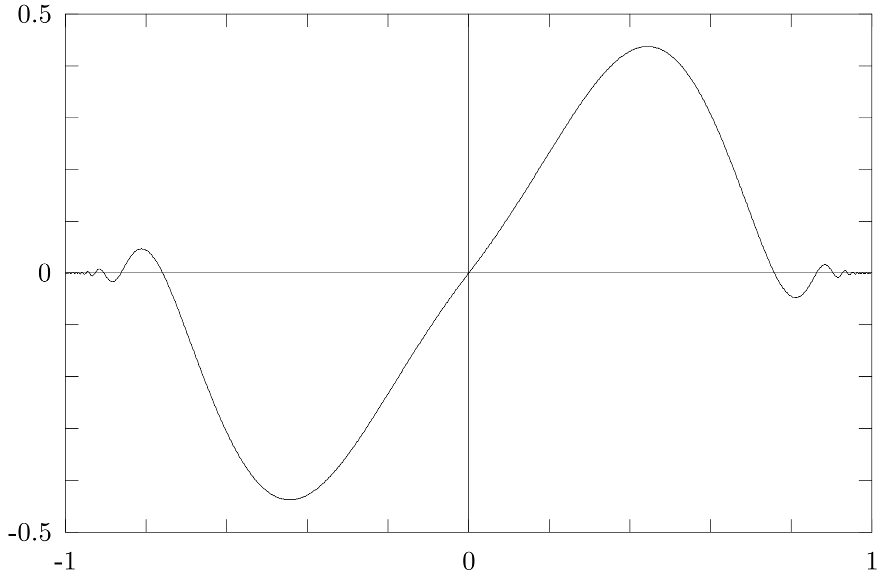

Figure 4.1: Transformed sine function $\frac{\sin S(x)}{(S(x))^2+1}$.

Proposition 4.5.3 (Elementary order-free recursive functions) *The following real-valued functions have recursive extensions over $\mathbf{R}^\star$:*

$$\sqrt{x}, |x|, \max, \exp, \ln, \sin, \cos, \tan\,.$$

Proof.

(1) The transformation $T :\subseteq \mathbb{R} \to \mathbb{R}, x \mapsto \frac{x}{1-x}$ is injective and T and its inverse $T^{-1} :\subseteq \mathbb{R} \to \mathbb{R}, x \mapsto \frac{x}{1+x}$ are rational functions and hence recursive over $\mathbf{R}^\star$. Furthermore, $T^{-1}\sqrt{T}$ has the total recursive extension $f : \mathbb{R} \to \mathbb{R}$ over $\mathbf{R}$, given by

$$f(x) := \begin{cases} 0 & \text{if} \quad x < 0 \\ T^{-1}\sqrt{T(x)} & \text{if} \quad 0 \le x < 1 \\ 1 & \text{if} \quad 1 \le x \end{cases} .$$

By the previous corollary there is a recursive function $f' :\subseteq \mathbb{R} \to \mathbb{R}$ over $\mathbf{R}^\star$ which is equal to f on $[0, 1]$. Hence, $Tf'T^{-1}$ is a recursive extension of $\sqrt{}$ over $\mathbf{R}^\star$. Thus, by $|x| = \sqrt{x^2}$, the absolute value function is recursive over $\mathbf{R}^\star$ too. The function $\max : \mathbb{R}^2 \to \mathbb{R}, (x, y) \mapsto \max\{x, y\}$ is recursive over $\mathbf{R}^\star$ since $\max(x, y) = \frac{1}{2}(x + y + |x - y|)$.

(2) The transformation $S :\subseteq \mathbb{R} \to \mathbb{R}, x \mapsto \frac{x}{1-|x|}$ is injective and S and its inverse $S^{-1} : \mathbb{R} \to \mathbb{R}, x \mapsto \frac{x}{1+|x|}$ are recursive over $\mathbf{R}^\star$ by (1). Furthermore, $T^{-1} \exp S$ has the total recursive extension $f : \mathbb{R} \to \mathbb{R}$, given by

$$f(x) := \begin{cases} 0 & \text{if} \quad x \le -1 \\ T^{-1} \exp S(x) & \text{if} \quad -1 < x < 1 \\ 1 & \text{if} \quad 1 \le x \end{cases}$$

and $S^{-1} \ln T$ has the total recursive extension $g : \mathbb{R} \to \mathbb{R}$, given by

$$g(x) := \begin{cases} -1 & \text{if} \quad x \le 0 \\ S^{-1} \ln T(x) & \text{if} \quad 0 < x < 1 \\ 1 & \text{if} \quad 1 \le x \end{cases} .$$

By the previous corollary there are recursive functions $f', g' :\subseteq \mathbb{R} \to \mathbb{R}$ over $\mathbf{R}^\star$ which coincide with f, g on $[-1, 1], [0, 1]$, respectively. Hence, $\exp = Tf'S^{-1}$ and $\ln = Sg'T^{-1}$ are recursive over $\mathbf{R}^\star$.

(3) The function $f :\subseteq \mathbb{R} \to \mathbb{R}, x \mapsto \frac{\sin S(x)}{(S(x))^2+1}$ has the total recursive extension $g : \mathbb{R} \to \mathbb{R}$ over $\mathbf{R}$, given by

$$g(x) := \begin{cases} f(x) & \text{if} \quad x \in (-1, 1) \\ 0 & \text{else} \end{cases} .$$

Figure 4.1 shows the transformed sine function. By the previous corollary there is a recursive function $g' :\subseteq \mathbb{R} \to \mathbb{R}$ over $\mathbf{R}^\star$ which coincides with g on $[-1, 1]$. Since $\sin(x) = gS^{-1}(x) \cdot (x^2 + 1), \cos(x) = \sin(x + \pi/2)$ and $\tan(x) = \frac{\sin(x)}{\cos(x)}$, the trigonometric functions sin, cos, tan are recursive over $\mathbf{R}^\star$.

□

Now we will apply the idea, which has been demonstrated in some special cases, to the general case of a total recursive function $f : \mathbb{R}^n \to \mathbb{R}$ over $\mathbf{R}$. For the compression we will use the compactifier provided by Proposition 4.4.65.

Theorem 4.5.4 (Order-free recursion) *If $f : \mathbb{R}^n \to \mathbb{R}$ is recursive over $\mathbf{R}$, then it is recursive over $\mathbf{R}^\star$ too.*

Proof. By Proposition 4.4.42 $\mathbb{R}^n$ is a recursively locally compact recursive metric space with recursive exhausting sequence $([-i,i]^n)_{i\in\mathbb{N}}$. By Proposition 4.4.65 there exists a recursive compactifier $m : \mathbb{R}^n \to \mathbb{R}$ over $\mathbf{R}$, i.e. $0 < m(x) \le 2^{-k}$ for all $x \in \mathbb{R}^n$ and $k := \min\{i \in \mathbb{N} : x \in [-i,i]^n\}$. First, we prove that m is recursive over $\mathbf{R}^\star$. The transformations $S :\subseteq \mathbb{R}^n \to \mathbb{R}^n, x \mapsto \frac{x}{1-||x||}$ and its inverse $S^{-1} : \mathbb{R}^n \to \mathbb{R}^n, x \mapsto \frac{x}{1+||x||}$ are recursive over $\mathbf{R}^\star$ by Proposition 4.5.3 and since $||(x_1,...,x_n)|| = \max\{|x_1|,...,|x_n|\}$. Moreover, m vanishes at infinity and thus mS has the total recursive extension $g : \mathbb{R}^n \to \mathbb{R}$ over $\mathbf{R}$, defined by

$$g(x) := \begin{cases} mS(x) & \text{if} \quad ||x|| < 1 \\ 0 & \text{else} \end{cases}$$

Thus, by Corollary 4.5.2 there is a recursive function $g' :\subseteq \mathbb{R}^n \to \mathbb{R}$ over $\mathbf{R}^\star$ which coincides with g on $K := \{x \in \mathbb{R}^n : ||x|| \le 1\}$. Thus, $m = g'S^{-1}$ is recursive over $\mathbf{R}^\star$.

As a special case of S^{-1} the transformation $I : \mathbb{R} \to \mathbb{R}, y \mapsto \frac{y}{1+||y||}$ and its inverse $I^{-1} :\subseteq \mathbb{R} \to \mathbb{R}, y \mapsto \frac{y}{1-||y||}$ are recursive over $\mathbf{R}^\star$. We apply Proposition 4.5.1 to f, I and thus there is a sequence $(f_n)_{n\in\mathbb{N}}$ in $\mathcal{C}(\mathbb{R}^n)$ such that $F : \mathbb{R}^n \times \mathbb{N} \to \mathbb{R}, (x,k) \mapsto f_k(x)$ is recursive over $\mathbf{R}^\star$ and $d_{\mathcal{C}}(If, If_k) < 2^{-k-1}$ for all $k \in \mathbb{N}$. Let $f' := If$ and $f'_k := If_k$ for all $k \in \mathbb{N}$. Since $||I(y)|| < 1$ for all $y \in \mathbb{R}$ we obtain

$$\frac{1}{3}||f'(x) - f'_k(x)|| \le \frac{||f'(x) - f'_k(x)||}{1 + ||f'(x)|| + ||f'_k(x)||} \le \frac{||f'(x) - f'_k(x)||}{1 + ||f'(x) - f'_k(x)||}$$

for all $x \in \mathbb{R}^n$, $k \in \mathbb{N}$ and thus

$$\begin{aligned} \frac{2^{-j-1}}{3}||f' - f'_k||_{[-j,j]^n} &\le \sum_{i=0}^{\infty} \frac{2^{-i-1}}{3}||f' - f'_k||_{[-i,i]^n} \\ &\le \sum_{i=0}^{\infty} 2^{-i-1} \frac{||f' - f'_k||_{[-i,i]^n}}{1 + ||f' - f'_k||_{[-i,i]^n}} \\ &= d_{\mathcal{C}}(If, If_k) < 2^{-k-1} \end{aligned}$$

for all $j, k \in \mathbb{N}$. Hence,

$$\frac{1}{6}m(x)||f'(x) - f'_k(x)|| \le \frac{2^{-\min\{i\in\mathbb{N}:x\in[-i,i]^n\}-1}}{3}||f'(x) - f'_k(x)|| < 2^{-k-1}$$

for all $x \in \mathbb{R}^n$ and $k \in \mathbb{N}$. Since $m(x) > 0$ for all $x \in \mathbb{R}^n$ we obtain

$$f'(x) = \frac{6}{m(x)} \cdot \lim_{k\to\infty} (\frac{m(x)}{6} f'_k(x))$$

for all $x \in \mathbb{R}^n$. Let $F' : \mathbb{R}^n \times \mathbb{N} \to \mathbb{R}$ be defined by $F'(x,k) := IF(x,k)$ for all $x \in \mathbb{R}^n$ and $k \in \mathbb{N}$. Then F' is recursive over $\mathbf{R}^\star$ and so are $f' = \frac{6}{m} \cdot \mathrm{Lim}[\frac{m}{6} \cdot F']$ and $f = I^{-1} \circ f'$. $\square$

In case of the real numbers one could use special compactifiers, constructed with the help of the exponential function. The proof presented here can be generalized to finite dimensional Banach spaces and more general classes of partial functions (cf. [Bra97]).

Chapter 5

Conclusion

We have introduced a high-level language which is suitable to describe effectivity in analysis. This high-level language is based on structures and the special class of perfect structures has been proved to be quite useful. Several perfect structures have been explicitly investigated and a general method has been proposed, which allows to construct perfect structures based on recursive metric spaces. Admittedly, numerous questions remain open and we want to mention some of them:

(1) Does the recursive language provide a suitable tool to investigate computability in analysis and other parts of mathematics?

(2) Is the presented high-level language non-redundant or can it be simplified?

(3) Are there any other abstract languages which can be based on perfect structures?

(4) Is there a feasible implementation of such a language on computers and do they provide a realistic framework for concrete programming?

(5) How could a formal theory of verification for such programs look like?

(6) How can the domains of recursive operations over topological structures be classified in topological hierarchies?

(7) Are there any other general methods to construct perfect structures, than by recursive metric spaces?

(8) Is it possible to express relative computability in analysis within the recursive language?

(9) Is there a general abstract model of complexity for metric spaces?

(10) Which properties does the category of recursive operations over perfect structures offer?

(11) Is our high-level language related to fuzzy set theory and logic and can it be applied to hybrid systems?

(12) Which is the relation of our high-level language to other languages based on domain theory?

Of course, this list could be continued. According to the last question, it has to be noted that domain theory has been used by Di Gianantonio [DG93, DG96], Sünderhauf [Sün94, Sün95], Escardo, Streicher [Esc96, Esc97, ES99c], Blanck [Bla97b, Bla97a, Bla99] and Edalat, Heckmann and Sünderhauf [EH98, ES99b, ES99a] to construct programming languages or models of computability in analysis. Some of these approaches are more concrete, since domains are used to represent spaces and others are more abstract, since high-level languages over domains are proposed (cf. also [SHT99]).

Moreover, it would be interesting (and it should be possible) to define a WHILE-programming language over (perfect) structures, which is equivalent to our recursive language. Such a language would be a step closer to real imperative programming languages. A related WHILE-programming language has been investigated by Tucker, Zucker and Stewart [TZ, TZ99, Ste99].

For the special case of real numbers, there exists a feasible real random access machine model, which has been developed in a joint project with Peter Hertling [BH98]. This model does not only offer the correct class of computable functions, but it is even polynomially realistic, compared to the time complexity theory of Ko and Weihrauch [Ko91, Wei97]. Thus, theoretically, there exists a realistic imperative programming language on the real numbers (cf. also [MHW97]). A prototype implementation of such a programming language has been introduced by Müller [Mül96].

However, it is still a long way from recursion over perfect structures to real programming languages over "perfect data structures".

Appendix: Computable Operations and Relations

The aim of this appendix is to compare the notion of computability of operations, as defined in Definition 3.1.14 with the notion of computability of relations, as defined by Weihrauch [Wei95, Wei, BH94]. We start with the definition of a computable relation $R \subseteq X \times Y$. For such relations we define $\mathrm{dom}(R) := \{x \in X : (\exists y \in Y)(x, y) \in R\}$.

Definition A.0.5 (Computable relation) Let (X, δ_X), (Y, δ_Y) be represented spaces. Then $R \subseteq X \times Y$ is called a (δ_X, δ_Y)*–computable relation*, if there is a computable function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that

$$(\delta_X(p), \delta_Y F(p)) \in R$$

for all $p \in \mathrm{dom}(\delta_X)$ such that $\delta_X(p) \in \mathrm{dom}(R)$. Furthermore, R is called a *strongly* (δ_X, δ_Y)*–computable relation*, if, additionally, $p \notin \mathrm{dom}(F)$ for all $p \in \mathrm{dom}(\delta_X)$ such that $\delta_X(p) \notin \mathrm{dom}(R)$.

Now we obtain the following result about computable relations and operations.

Proposition A.0.6 (Computable operations and relations) *Let* (X, δ_X)*,* (Y, δ_Y) *be represented spaces.*

(1) If $f :\subseteq X \rightrightarrows Y$ *is a* (δ_X, δ_Y)*–computable operation, then* $\mathrm{graph}(f) \subseteq X \times Y$ *is a* (δ_X, δ_Y)*–computable relation.*

Now let X*, additionally, be a* T_0*-space with countable base and let* δ_X *be a standard representation of* X*.*

(2) If $R \subseteq X \times Y$ *is a (strongly)* (δ_X, δ_Y)*–computable relation, then there is a (strongly)* (δ_X, δ_Y)*–computable operation* $f :\subseteq X \rightrightarrows Y$ *such that* $\mathrm{graph}(f) \subseteq R$ *and* $\mathrm{dom}(f) = \mathrm{dom}(R)$*.*

Proof.

(1) Let $f :\subseteq X \rightrightarrows Y$ be a (δ_X, δ_Y)–computable operation. Then there is some computable function $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ such that

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

and $\langle p, \mathbb{N}^\mathbb{N}\rangle \subseteq \mathrm{dom}(\delta_Y F)$ for all $p \in \mathrm{dom}(f\delta_X)$. Define $G :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ by $G(p) := F\langle p, p\rangle$. Then G is computable and $(\delta_X(p), \delta_Y G(p)) \in \mathrm{graph}(f)$ for all $p \in \mathrm{dom}(\delta_X)$ such that $\delta_X(p) \in \mathrm{dom}(f)$.

(2) Now let X be a T_0-space with countable base $\{B_n : n \in \mathbb{N}\}$ and let $\delta_X :\subseteq \mathbb{N}^\mathbb{N} \to X$ be the standard representation of X, defined by

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

Let $R \subseteq X \times Y$ be a (δ_X, δ_Y)–computable relation. Then there is a computable function $G :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ such that $(\delta_X(p), \delta_Y G(p)) \in R$ for all $p \in \mathrm{dom}(\delta_X)$ such that $\delta_X(p) \in \mathrm{dom}(R)$. Let $H : \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ be the computable function which exists by Lemma 3.1.27. Define $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ by $F\langle p, q\rangle := GH\langle p, q\rangle$ for all $p, q \in \mathbb{N}^\mathbb{N}$. Then F is computable. Let $f :\subseteq X \rightrightarrows Y$ be defined by $\mathrm{dom}(f) := \mathrm{dom}(R)$ and

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

for all $p \in \delta_X^{-1}(\mathrm{dom}(R))$. 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. f is well-defined and (δ_X, δ_Y)–computable. Furthermore,

$$(\delta_X(p), F\langle p, q\rangle) = (\delta_X(p), GH\langle p, q\rangle) \in R,$$

hence, $\mathrm{graph}(f) \subseteq R$.

Now let R be even strongly (δ_X, δ_Y)–computable via G, i.e., additionally, $p \notin \mathrm{dom}(G)$ for all $p \in \mathrm{dom}(\delta_X)$ with $\delta_X(p) \notin \mathrm{dom}(R)$. Then $\langle p, \mathbb{N}^\mathbb{N}\rangle \subseteq \mathrm{dom}(\delta_Y F)$ for all $p \in \mathrm{dom}(\delta_X)$ with $\delta_X(p) \in \mathrm{dom}(R)$ and $\langle p, \mathbb{N}^\mathbb{N}\rangle \not\subseteq \mathrm{dom}(F)$ for all $p \in \mathrm{dom}(\delta_X)$ with $\delta_X(p) \notin \mathrm{dom}(R)$. Thus, f is a strongly (δ_X, δ_Y)–computable operation.

$\square$

Tables of Perfect Structures

In the following we will list those perfect (pre)structures which have been investigated in the previous chapters with the precise definition of their initial operations. Additionally, the tables include the recursive points and the underlying topologies of the corresponding structures. We recall the fact that predicates (sets) are identified with their semi-characteristic operation (cf. Section 3.2.4).

$\mathbb{N}$	**Naturals**	$\{0, 1, 2, ...\}$, discrete topology
0	constant	0
n	identity	$\mathrm{id} : \mathbb{N} \to \mathbb{N}, n \mapsto n$
$n+1$	successor function	$s : \mathbb{N} \to \mathbb{N}, n \mapsto n+1$

$\mathbb{Z}$	**Integers**	integer numbers, discrete topology
0	constant	0
1	constant	1
$x + y$	addition	$\mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}, (x, y) \mapsto x + y$
$-x$	negation	$\mathbb{Z} \to \mathbb{Z}, x \mapsto -x$
$=$	equality test	$\{(x, y) \in \mathbb{Z} \times \mathbb{Z} : x = y\}$

$\mathbb{Q}$	**Rationals**	rational numbers, discrete topology
0	constant	0
1	constant	1
$x+y$	addition	$\mathbb{Q}\times\mathbb{Q}\to\mathbb{Q},(x,y)\mapsto x+y$
$-x$	negation	$\mathbb{Q}\to\mathbb{Q},x\mapsto -x$
$x\cdot y$	multiplication	$\mathbb{Q}\times\mathbb{Q}\to\mathbb{Q},(x,y)\mapsto x\cdot y$
$1/x$	inversion	$\subseteq\mathbb{Q}\to\mathbb{Q},x\mapsto 1/x$
$=$	equality test	$\{(x,y)\in\mathbb{Q}\times\mathbb{Q}:x=y\}$

$\mathbb{R}$	**Reals**	computable real numbers, Euclidean topology
0	constant	0
1	constant	1
$x+y$	addition	$\mathbb{R}\times\mathbb{R}\to\mathbb{R},(x,y)\mapsto x+y$
$-x$	negation	$\mathbb{R}\to\mathbb{R},x\mapsto -x$
$x\cdot y$	multiplication	$\mathbb{R}\times\mathbb{R}\to\mathbb{R},(x,y)\mapsto x\cdot y$
$1/x$	inversion	$\subseteq\mathbb{R}\to\mathbb{R},x\mapsto 1/x$
Lim	limit	$\mathrm{Lim}:\subseteq\mathbb{R}^{\mathbb{N}}\to\mathbb{R},(x_n)_{n\in\mathbb{N}}\mapsto\lim_{n\to\infty}x_n$
		$\mathrm{dom}(\mathrm{Lim}):=\{(x_n)_{n\in\mathbb{N}}:(\forall n>k)\ \lvert x_n-x_k\rvert\le 2^{-k}\}$
$x<y$	comparison	$\{(x,y)\in\mathbb{R}\times\mathbb{R}:x<y\}$

$\mathbb{C}$	**Complex numbers**	computable complex numbers, Gaußian topology
$x+iy$	bijection	$\mathbb{R}\times\mathbb{R}\to\mathbb{C},(x,y)\mapsto x+iy$
Re	real part	$\mathbb{C}\to\mathbb{R},x+iy\mapsto x$
Im	imaginary part	$\mathbb{C}\to\mathbb{R},x+iy\mapsto y$

$\mathcal{K}(X)$	**Compact subsets**	recursive compact sets, Vietoris topology
$\{x\}$	injection	$X \to \mathcal{K}(X), x \mapsto \{x\}$
A	identity	$\mathcal{K}(X) \to \mathcal{K}(X), A \mapsto A$
$A \cup B$	union	$\mathcal{K}(X) \times \mathcal{K}(X) \to \mathcal{K}(X), (A, B) \mapsto A \cup B$
$d_\mathcal{K}$	Hausdorff metric	$d_\mathcal{K} : \mathcal{K}(X) \times \mathcal{K}(X) \to \mathbb{R}$,
		$(A, B) \mapsto \max \left\{ \sup\limits_{a \in A} \inf\limits_{b \in B} d(a, b), \sup\limits_{b \in B} \inf\limits_{a \in A} d(a, b) \right\}$
Lim	limit	$\mathrm{Lim} :\subseteq \mathcal{K}(X)^\mathbb{N} \to \mathcal{K}(X), (A_n)_{n \in \mathbb{N}} \mapsto \lim\limits_{n \to \infty} A_n$
		$\mathrm{dom}(\mathrm{Lim}) := \{(A_n) : (\forall n > k)\ d_\mathcal{K}(A_n, A_k) \leq 2^{-k}\}$
Here, (X, d) is a recursive metric space.		

$\mathcal{A}(X)$	**Closed subsets**	recursive closed sets, Fell topology
$\{x\}$	injection	$X \to \mathcal{A}(X), x \mapsto \{x\}$
A	identity	$\mathcal{A}(X) \to \mathcal{A}(X), A \mapsto A$
$A \cup B$	union	$\mathcal{A}(X) \times \mathcal{A}(X) \to \mathcal{A}(X), (A, B) \mapsto A \cup B$
$d_\mathcal{A}$	metric	$d_\mathcal{A} : \mathcal{A}(X) \times \mathcal{A}(X) \to \mathbb{R}$,
		$(A, B) \mapsto \sum_{i=0}^{\infty} 2^{-i-1} \lvert d_A - d_B \rvert_{K_i}$
Lim	limit	$\mathrm{Lim} :\subseteq \mathcal{A}(X)^\mathbb{N} \to \mathcal{A}(X), (A_n)_{n \in \mathbb{N}} \mapsto \lim\limits_{n \to \infty} A_n$
		$\mathrm{dom}(\mathrm{Lim}) := \{(A_n) : (\forall n > k)\ d_\mathcal{A}(A_n, A_k) \leq 2^{-k}\}$
Here, (X, d) is a nice recursively locally compact recursive metric space with recursive exhausting sequence $(K_i)_{i \in \mathbb{N}}$.		

$\mathcal{C}(X)$	**Continuous functions**	computable functions, compact open topology
1	constant function	$\{()\} \to \mathcal{C}(X), () \mapsto \hat{1}$
		$\hat{1} : X \to \mathbb{R}, x \mapsto 1$
d_x	point distance	$X \to \mathcal{C}(X), x \mapsto d_x$
		$d_x : X \to \mathbb{R}, y \mapsto d(x, y)$
f	identity	$\mathcal{C}(X) \to \mathcal{C}(X), f \mapsto f$
$y \cdot f$	scalar product	$\mathbb{R} \times \mathcal{C}(X) \to \mathcal{C}(X), (y, f) \mapsto y \cdot f$
$f + g$	addition	$\mathcal{C}(X) \times \mathcal{C}(X) \to \mathcal{C}(X), (f, g) \mapsto f + g$
$f \cdot g$	multiplication	$\mathcal{C}(X) \times \mathcal{C}(X) \to \mathcal{C}(X), (f, g) \mapsto f \cdot g$
$d_{\mathcal{C}(X)}$	metric	$d_{\mathcal{C}(X)} : \mathcal{C}(X) \times \mathcal{C}(X) \to \mathbb{R}$,
		$(f, g) \mapsto \sum_{i=0}^{\infty} 2^{-i-1} \frac{\lvert f-g\rvert_{K_i}}{1+\lvert f-g\rvert_{K_i}}$
Lim	limit	$\mathrm{Lim} :\subseteq \mathcal{C}(X)^{\mathbb{N}} \to \mathcal{C}(X), (f_n)_{n\in\mathbb{N}} \mapsto \lim\limits_{n\to\infty} f_n$
		$\mathrm{dom}(\mathrm{Lim}) := \{(f_n) : (\forall n > k) d_{\mathcal{C}(X)}(f_n, f_k) \leq 2^{-k}\}$

Here, (X, d) is a recursively locally compact recursive metric space with recursive exhausting sequence $(K_i)_{i\in\mathbb{N}}$.

$\mathcal{C}(X, Y)$	**Continuous functions**	computable functions, compact open topology
f	identity	$\mathcal{C}(X, Y) \to \mathcal{C}(X, Y), f \mapsto f$
$y \cdot f$	scalar product	$Y \times \mathcal{C}(X) \to \mathcal{C}(X, Y), (y, f) \mapsto y \cdot f$
$f + g$	addition	$\mathcal{C}(X, Y) \times \mathcal{C}(X, Y) \to \mathcal{C}(X, Y), (f, g) \mapsto f + g$
$d_{\mathcal{C}}$	metric	$d_{\mathcal{C}} : \mathcal{C}(X, Y) \times \mathcal{C}(X, Y) \to \mathbb{R}$,
		$(f, g) \mapsto \sum_{i=0}^{\infty} 2^{-i-1} \frac{\lvert\lvert f-g\rvert\rvert_{K_i}}{1+\lvert\lvert f-g\rvert\rvert_{K_i}}$
Lim	limit	$\mathrm{Lim} :\subseteq \mathcal{C}(X, Y)^{\mathbb{N}} \to \mathcal{C}(X, Y), (f_n)_{n\in\mathbb{N}} \mapsto \lim\limits_{n\to\infty} f_n$
		$\mathrm{dom}(\mathrm{Lim}) := \{(f_n) : (\forall n > k) d_{\mathcal{C}}(f_n, f_k) \leq 2^{-k}\}$

Here, (X, d) is a recursively locally compact recursive metric space with recursive exhausting sequence $(K_i)_{i\in\mathbb{N}}$ and $(Y, \lvert\lvert\ \rvert\rvert)$ is a recursive Banach space.

$\mathcal{C}[0,1]$	**Continuous functions**	computable functions, topology of uniform convergence
1	constant function	$\{()\} \to \mathcal{C}[0,1], () \mapsto \hat{1}$
		$\hat{1} : [0,1] \to \mathbb{R}, x \mapsto 1$
f	identity	$\mathcal{C}[0,1] \to \mathcal{C}[0,1], f \mapsto f$
$y \cdot f$	scalar product	$\mathbb{R} \times \mathcal{C}[0,1] \to \mathcal{C}[0,1], (y,f) \mapsto y \cdot f$
$f + g$	addition	$\mathcal{C}[0,1] \times \mathcal{C}[0,1] \to \mathcal{C}[0,1], (f,g) \mapsto f + g$
$f \cdot g$	multiplication	$\mathcal{C}[0,1] \times \mathcal{C}[0,1] \to \mathcal{C}[0,1], (f,g) \mapsto f \cdot g$
$\|\|f\|\|$	norm	$\|\| \; \|\| : \mathcal{C}[0,1] \to \mathbb{R}, f \mapsto \sup_{x \in [0,1]} \|f(x)\|$
Lim	limit	$\mathrm{Lim} :\subseteq \mathcal{C}[0,1]^{\mathbb{N}} \to \mathcal{C}[0,1], (f_n)_{n \in \mathbb{N}} \mapsto \lim\limits_{n \to \infty} f_n$
		$\mathrm{dom}(\mathrm{Lim}) := \{(f_n)_{n \in \mathbb{N}} : (\forall n > k) \|\|f_n - f_k\|\| \leq 2^{-k}\}$

Symbols

$\emptyset$	Empty set
$\mathbb{N}$	Set of natural numbers $\{0, 1, 2, ...\}$
$\mathbb{Z}$	Set of integer numbers
$\mathbb{Q}$	Set of rational numbers
$\mathbb{R}$	Set of real numbers
$\mathbb{C}$	Set of complex numbers
$\mathbb{N}^{\mathbb{N}}$	Set of sequences of natural numbers
(x, y)	Open interval of real numbers
$(x, y]$	Left open interval of real numbers
$[x, y)$	Right open interval of real numbers
$[x, y]$	Closed interval of real numbers
id_X	Identity of set X
cf_X	Characteristic function of X
c_X	Semi-characteristic operation of X
Ω_X	Omnipotent operation of X
in	Injection which maps a point to a single-valued set
π	Real number $\pi = 3.141...$
∞	Infinity
$\dot{-}$	Arithmetical difference
S	Successor function
$\mathrm{pr}_i^{(n)}, \mathrm{pr}_i$	Projection of an n-tuple to the i-th component
$\langle n, k\rangle$	Cantor's pairing function for natural numbers n, k
$\langle ., .\rangle$	Pairing function
π_i	Projection of the i-th component of the inverse of a pairing function
max	Maximum
min	Minimum

sup	Supremum
inf	Infimum
$\lvert x\rvert$	Absolute value of real number x
$\sqrt{x}$	Square root of x
ln	Logarithm function
exp	Exponential function
sin	Sine function
cos	Cosine function
tan	Tangens function
$f :\subseteq X \to Y$	Partial function
$f : X \to Y$	Total function
$f :\subseteq X \rightrightarrows Y$	Partial operation (many-valued function)
$f : X \rightrightarrows Y$	Total operation (many-valued function)
f^{-1}	Inverse operation of f
$f(x) =\uparrow$	Operation f is undefined at x
$f(X)$	Image of set X under operation f
$f^{-1}(Y)$	Preimage of set Y under operation f
$f\lvert_X$	Restriction of operation f in the domain
$f\lvert^Y$	Restriction of operation f in the range
$f\rfloor_X$	Restriction of operation f in the source
$f\rceil^Y$	Restriction of operation f in the target
$\mathrm{dom}(f)$	Domain of operation f
$\mathrm{range}(f)$	Range of operation f
$\mathrm{graph}(f)$	Graph of operation f
$\mathrm{supp}(f)$	Support of function f
$x \in X$	x is element of X
$X \subseteq Y$	X is included in Y
X^0	Set of empty tuple $\{()\}$
X^c	Complement of set X
$X \cup Y$	Union of set X and Y
$X \cap Y$	Intersection of set X and Y
$X \setminus Y$	Difference of set X and Y
$X \times Y$	Product of set X and Y
2^X	Power set of set X
$X^{\mathbb{N}}$	Set of sequences $s : \mathbb{N} \to X$
Y^X	Set of functions $f : X \to Y$

$=_X$	Equality of X
$\neq_X$	Inequality of X
$\overline{A}$	Closure of set A
A°	Interior of set A
∂A	Border of set A
A^+	Miss set of A
A^-	Hit set of A
f_0	Section of operation f
f_i	Projection of operation f on the i-th component
(f, g)	Juxtaposition of operation f and g
$f \times g$	Product of operation f and g
$f \circ g$	Composition of operation f and g
f^*	Iteration of operation f
f_*	Evaluation of operation f
$[f]$	Transposition of operation f
$f^{\mathbb{N}}$	Exponentiation of operation f
$f^{\triangle}$	Sequentialization of operation f
$f^{\leftrightarrow}$	(Twisted) Inversion of operation f
μf	μ-recursion of function f
$f_{\min}$	Minimization of operation f
$f \cup g$	Union of operation f and g
$\bigcup f$	Union of operation f
$(f\|_t g)$	Definition by cases with cases f, g and test t
$\hat{f}$	Lifting of function f
f^Z	Exponentiation of function f
$f^\star$	Dual of function f
$\mathrm{ev}_{X^{\mathbb{N}}}$	Evaluation function of $X^{\mathbb{N}}$
$\mathrm{ev}_{\mathcal{C}(X,Y)}$	Evaluation function of $\mathcal{C}(X, Y)$
$\mathcal{K}(X)$	Set of non-empty compact subsets of X
$\mathcal{A}(X)$	Set of non-empty closed subsets of X
$\mathcal{C}(X)$	Set of continuous functions $f : X \to \mathbb{R}$
$\mathcal{C}(X, Y)$	Set of continuous functions $f : X \to Y$
$\|\| \ \|\|$	Norm
$\|\| \ \|\|_K$	Supremum of norm over set K

$d_{\mathbb{R}}$	Euclidean metric
$d_{\mathcal{K}}$	Hausdorff metric
$d_{\mathcal{A}}$	Metric for the Fell topology
$d_{\mathcal{C}}$	Metric for the compact open topology
α_X	Dense sequence in X
α_e	Dense sequence in a Banach space
Lim	Limit operator for rapidly converging Cauchy sequences
Lim_X	Limit operator of space X
dist	Distance operator
d_A	Distance function of set A
d_x	Distance function of point x
$B(x, \varepsilon)$	Open ball with center x and radius ε
$\mathbf{N}$	Structure of natural numbers
$\mathbf{Z}$	Structure of integer numbers
$\mathbf{Q}$	Structure of rational numbers
$\mathbf{R}$	Structure of real numbers
$\mathbf{R}^\star$	Order-free structure of real numbers
$\mathbf{C}$	Structure of complex numbers
$\mathcal{K}(\mathbf{X})$	Structure of the set of non-empty compact subsets
$\mathcal{A}(\mathbf{X})$	Structure of the set of non-empty closed subsets
$\mathcal{C}[\mathbf{0}, \mathbf{1}]$	Structure of continuous functions $f : [0,1] \to \mathbb{R}$
$\mathcal{C}(\mathbf{X})$	Structure of continuous functions $f : X \to \mathbb{R}$
$\mathcal{C}(\mathbf{X}, \mathbf{Y})$	Structure of continuous functions $f : X \to Y$
$(X, f_1, ..., f_n)$	(Pre)structure with universe X and initial operations $f_1, ..., f_n$
$S \sqsubseteq S'$	Extension of structures
$S \oplus S'$	Union of (pre)structures
$S \leq S'$	Reducibility of structures
$S \equiv S'$	Equivalence of structures
$S \leq_s S'$	Strong reducibility of structures
$S \equiv_s S'$	Strong equivalence of structures
$\mathbb{N}^*$	Set of finite words over $\mathbb{N}$
$\sqsubseteq$	Prefix relation for words and sequences
$\lg(w)$	Length of word w
ν^*	Standard numbering of $\mathbb{N}^*$
$\overline{w}$	Number of word w w.r.t. ν^*

wp	Concatenation of word w and sequence p
$w\mathbb{N}^{\mathbb{N}}$	Set of all sequences which extend the word w
$\langle p, \mathbb{N}^{\mathbb{N}} \rangle$	Set of all tuples $\langle p, q \rangle$ for fixed p
$\hat{n}$	Sequence $n00...$
$\delta_{\mathbb{N}}$	Representation of natural numbers
$\delta_{\mathbb{R}}$	Representation of real numbers
δ_X	Representation of set X
$[\delta, \delta']$	Product representation
δ^∞	Sequence representation
δ^{cyl}	Cylindrification of representation
$\delta \leq \delta'$	Reducibility of representations
$\delta \equiv \delta'$	Equivalence of representations

Bibliography

[Abe80] Oliver Aberth. *Computable Analysis.* McGraw-Hill, New York, 1980.

[BB85] Errett Bishop and Douglas S. Bridges. *Constructive Analysis*, volume 279 of *Grundlehren der mathematischen Wissenschaft.* Springer, Berlin, 1985.

[Bee93] Gerald Beer. *Topologies on Closed and Closed Convex Sets*, volume 268 of *Mathematics and Its Applications.* Kluwer Academic, Dordrecht, 1993.

[BH94] Vasco Brattka and Peter Hertling. Continuity and computability of relations. Informatik Berichte 164, FernUniversität Hagen, Hagen, September 1994.

[BH98] Vasco Brattka and Peter Hertling. Feasible real random access machines. *Journal of Complexity*, 14(4):490–526, 1998.

[Bla97a] Jens Blanck. *Computability on topological spaces by effective domain representations.* PhD thesis, Uppsala University, Department of Mathematics, Uppsala, Sweden, 1997.

[Bla97b] Jens Blanck. Domain representability of metric spaces. *Annals of Pure and Applied Logic*, 83:225–247, 1997.

[Bla99] Jens Blanck. Effective domain representations of $H(X)$, the space of compact subsets. *Theoretical Computer Science*, 219:19–48, 1999.

[BM37] S. Banach and S. Mazur. Sur les fonctions calculables. *Ann. Soc. Pol. de Math.*, 16:223, 1937.

[Bou66] Nicolaus Bourbaki. *General Topology, Part 2*, volume 3 of *Elements of Mathematics.* Addison-Wesley, Reading, 1966.

[Bra95] Vasco Brattka. Computable selection in analysis. In Ker-I Ko and Klaus Weihrauch, editors, *Computability and Complexity in Analysis*, volume 190 of *Informatik-Berichte*, pages 125–138. FernUniversität Hagen, September 1995. CCA Workshop, Hagen, August 19-20, 1995.

[Bra96] Vasco Brattka. Recursive characterization of computable real-valued functions and relations. *Theoretical Computer Science*, 162:45–77, 1996.

[Bra97] Vasco Brattka. Order-free recursion on the real numbers. *Mathematical Logic Quarterly*, 43:216–234, 1997.

[Bra99] Vasco Brattka. Computable invariance. *Theoretical Computer Science*, 210:3–20, 1999.

[BW99] Vasco Brattka and Klaus Weihrauch. Computability on subsets of Euclidean space I: Closed and compact subsets. *Theoretical Computer Science*, 219:65–93, 1999.

[Cei62] G.S. Ceitin. Algorithmic operators in constructive metric spaces. *Tr. Mat. Inst. Steklov*, 67:295–361, 1962. (in Russian, English trans. in AMS Trans. 64, 1967).

[Cei64] G.S. Ceitin. Mean value theorems in constructive analysis. *Tr. Mat. Inst. Steklov*, 72:537–543, 1964. (in Russian, English trans. in AMS Trans. 100, 1972).

[Dei84] Thomas Deil. Darstellungen und Berechenbarkeit reeller Zahlen. Informatik Berichte 51, FernUniversität Hagen, Hagen, December 1984. Dissertation.

[DG93] Pietro Di Gianantonio. *A Functional Approach to Computability on Real Numbers*. Dottorato di ricerca in informatica, Università di Pisa-Genova-Udine, Udine, June 1993.

[DG96] Pietro Di Gianantonio. Real number computation and domain theory. *Information and Computation*, 127:11–25, 1996.

[EH98] Abbas Edalat and Reinhold Heckmann. A computational model for metric spaces. *Theoretical Computer Science*, 193:53–73, 1998.

[Eng89] Ryszard Engelking. *General Topology*, volume 6 of *Sigma series in pure mathematics*. Heldermann, Berlin, 1989.

[ES99a] Abbas Edalat and Philipp Sünderhauf. Computable Banach spaces via domain theory. *Theoretical Computer Science*, 219:169–184, 1999.

[ES99b] Abbas Edalat and Philipp Sünderhauf. A domain-theoretic approach to computability on the real line. *Theoretical Computer Science*, 210:73–98, 1999.

[ES99c] Martin Hötzel Escardó and Thomas Streicher. Induction and recursion on the partial real line with applications to real PCF. *Theoretical Computer Science*, 210:121–157, 1999.

[Esc96] Martin Hötzel Escardó. PCF extended with real numbers. *Theoretical Computer Science*, 162:79–115, 1996.

[Esc97] Martin Hötzel Escardó. *PCF extended with real numbers: a domain-theoretic approach to higher-order exact real number computation.* PhD thesis, Imperial College, University of London, London, Great Britain, 1997.

[Fen80] Jens E. Fenstad. *General Recursion Theory.* Perspectives in Mathematical Logic. Springer, Berlin, 1980.

[Fri71] Harvey Friedman. Algorithmic procedures, generalized Turing algorithms, and elementary recursion theory. In R.O. Gandy and C.M.E. Yates, editors, *Logic Colloquium'69*, volume 61 of *Studies in Logic and the Foundations of Mathematics*, pages 361–389, Amsterdam, 1971. North-Holland. Proceedings of the Summer School and Colloquium in Mathematical Logic, Manchester, August 1969.

[Ge93] Xiaolin Ge. *Some Algorithms in Euclidean Space and Group Representations.* Thesis, University of Minnesota, Minneapolis, May 1993.

[GN94] Xiaolin Ge and Anil Nerode. On extreme points of convex compact Turing located sets. In Anil Nerode and Yu. V. Matiyasevich, editors, *Logical Foundations of Computer Science*, volume 813 of *Lecture Notes in Computer Science*, pages 114–128, Berlin, 1994. Springer. Third International Symposium, LFCS'94, St. Petersburg, Russia, July 11-14, 1994.

[Grz55] Andrzej Grzegorczyk. Computable functionals. *Fundamenta Mathematicae*, 42:168–202, 1955.

[Grz57] Andrzej Grzegorczyk. On the definitions of computable real continuous functions. *Fundamenta Mathematicae*, 44:61–71, 1957.

[GS90] C.A. Gunter and D.S. Scott. Semantic domains. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 633–674, Amsterdam, 1990. Elsevier. Formal Models and Semantics.

[Hau71] Jürgen Hauck. Zur Präzisierung des Begriffes berechenbare reelle Funktion. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 17:295–300, 1971.

[Hau73] Jürgen Hauck. Berechenbare reelle Funktionen. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 19:121–140, 1973.

[Hau76] Jürgen Hauck. Berechenbare reelle Funktionenfolgen. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 22:265–282, 1976.

[Hau78] Jürgen Hauck. Konstruktive Darstellungen reeller Zahlen und Folgen. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 24:365–374, 1978.

[Hau80] Jürgen Hauck. Stetigkeitseigenschaften berechenbarer reeller Funktionen. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 26:69–76, 1980.

[Hau81] Jürgen Hauck. Berechenbarkeit in topologischen Räumen mit rekursiver Basis. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 27:473–480, 1981.

[Her99a] Peter Hertling. A real number structure that is effectively categorical. *Mathematical Logic Quarterly*, 45(2):147–182, 1999.

[Her99b] Peter Hertling. Standard representations of real numbers. *Chaos, Solitons & Fractals*, 10(6):1087–1101, 1999.

[KF82] Ker-I Ko and H. Friedman. Computational complexity of real functions. *Theoretical Computer Science*, 20:323–352, 1982.

[Kla56] D. Klaua. Berechenbare Analysis. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 2:265–303, 1956.

[Kla61] D. Klaua. *Konstruktive Analysis*. Deutscher Verlag der Wissenschaften, Berlin, 1961.

[Kle36] Stephen Cole Kleene. General recursive functions of natural numbers. *Mathematische Annalen*, 112:727–742, 1936.

[Kle52] Stephen Cole Kleene. *Introduction to Metamathematics*, volume 1 of *Bibliotheca Mathematica*. North-Holland, Amsterdam, 1952.

[Ko91] Ker-I Ko. *Complexity Theory of Real Functions*. Progress in Theoretical Computer Science. Birkhäuser, Boston, 1991.

[Kur66] K. Kuratowski. *Topology, Volume 1*. Academic Press, New York, 1966.

[Kus84] Boris Abramovich Kushner. *Lectures on Constructive Mathematical Analysis*, volume 60 of *Translation of Mathematical Monographs*. American Mathematical Society, Providence, 1984.

[KW84] Christoph Kreitz and Klaus Weihrauch. A unified approach to constructive and recursive analysis. In M.M. Richter, E. Börger, W. Oberschelp, B. Schinzel, and W. Thomas, editors, *Computation and Proof Theory*, volume 1104 of *Lecture Notes in Mathematics*, pages 259–278, Berlin, 1984. Springer. Proceedings of the Logic Colloquium, Aachen, July 18-23, 1983, Part II.

[KW87] Christoph Kreitz and Klaus Weihrauch. Compactness in constructive analysis revisited. *Annals of Pure and Applied Logic*, 36:29–38, 1987.

[Lac55] Daniel Lacombe. Classes récursivement fermés et fonctions majorantes. *Comptes Rendus Académie des Sciences Paris*, 240:716–718, June 1955. Théorie des fonctions.

[Lac58] Daniel Lacombe. Les ensembles récursivement ouverts ou fermés, et leurs applications à l'Analyse récursive. *Comptes Rendus Académie des Sciences Paris*, 246:28–31, 1958. Logique.

[Lac59] Daniel Lacombe. Quelques procédés de définition en topologie récursive. In A. Heyting, editor, *Constructivity in mathematics*, pages 129–158, Amsterdam, 1959. North Holland. Colloquium at Amsterdam, 1957.

[Mal71] Anatolií Ivanovič Mal'cev. Constructive algebras. I. In *The Metamathematics of Algebraic Systems*, pages 148–214, Amsterdam, 1971. North Holland. Collected papers: 1936–1967.

[Maz63] S. Mazur. *Computable Analysis*, volume 33. Razprawy Matematyczne, Warsaw, 1963.

[McL92] Colin McLarty. *Elementary Categories, Elementary Toposes*, volume 21 of *Oxford Logic Guides*. Clarendon Press, Oxford, 1992.

[MHW97] Friedhelm Meyer auf der Heide and Jiří Wiedermann. Numerical RAM: A realistic machine model for scientific computing. Preprint, 1997.

[MN82] George Metakides and Anil Nerode. The introduction of non-recursive methods into mathematics. In A.S. Troelstra and D. van Dalen, editors, *The L.E.J. Brouwer Centenary Symposium*, volume 110 of *Studies in Logic and the foundations of mathematics*, pages 319–335, Amsterdam, 1982. North-Holland. Proceedings of the conference held in Noordwijkerhout, June 8-13, 1981.

[Mos57] A. Mostowski. On computable sequences. *Fundamenta Mathematicae*, 44:37–51, 1957.

[Mos64a] Yiannis Nicholas Moschovakis. Notation systems and recursive ordered fields. *Compositio Math.*, 17:40–71, 1964.

[Mos64b] Yiannis Nicholas Moschovakis. Recursive metric spaces. *Fundamenta Mathematicae*, 55:215–238, 1964.

[Mos69] Yiannis Nicholas Moschovakis. Abstract first order computability I, II. *Transactions of the American Mathematical Society*, 138:427–464,465–504, 1969.

[Mos71] Yiannis Nicholas Moschovakis. Axioms for computation theories - first draft. In R.O. Gandy and C.M.E. Yates, editors, *Logic Colloquium'69*, volume 61 of *Studies in Logic and the Foundations of Mathematics*, pages 199–255, Amsterdam, 1971. North-Holland. Proceedings of the Summer School and Colloquium in Mathematical Logic, Manchester, August 1969.

[MTY97] Takakazu Mori, Yoshiki Tsujii, and Mariko Yasugi. Computability structures on metric spaces. In Douglas S. Bridges, Cristian S. Calude, Jeremy Gibbons, Steve Reeves, and Ian H. Witten, editors, *Combinatorics, Complexity, and Logic*, Discrete Mathematics and Theoretical Computer Science, pages 351–362, Singapore, 1997. Springer. Proceedings of DMTCS'96.

[Mül96] Norbert Th. Müller. Towards a real Real RAM: a prototype using C++. In Ker-I Ko, Norbert Müller, and Klaus Weihrauch, editors, *Computability and Complexity in Analysis*, pages 59–66. Universität Trier, 1996. Second CCA Workshop, Trier, August 22-23, 1996.

[NH85] Anil Nerode and Wen-Qi Huang. Application of pure recursion theory in recursive analysis. *Acta Mathematica Sinica*, 28:625–636, 1985. (in Chinese).

[Odi89] Piergiorgio Odifreddi. *Classical Recursion Theory*, volume 129 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, 1989.

[PEC75] Marian Pour-El and J. Caldwell. On a simple definition of computable functions of a real variable. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 21:1–19, 1975.

[PER89] Marian B. Pour-El and J. Ian Richards. *Computability in Analysis and Physics*. Perspectives in Mathematical Logic. Springer, Berlin, 1989.

[Pét67] Rózsa Péter. *Recursive Functions*. Academic Press, New York, 3 edition, 1967.

[Pét81] Rózsa Péter. *Recursive Functions in Computer Theory*, volume 9 of *Computers and their applications*. Ellis Horwood Publisher, Chichester, 1981.

[Ric54] H. Rice. Recursive real numbers. *Proc. Amer. Math. Soc.*, 5:784–791, 1954.

[Rob50] Julia Robinson. General recursive functions. *Proceedings of the American Mathematical Society*, 1:703–718, 1950.

[Rob51] R.M. Robinson. Review of "Peter, R., Rekursive Funktionen". *The Journal of Symbolic Logic*, 16:280, 1951.

[Rob68] Julia Robinson. Recursive functions of one variable. *Proceedings of the American Mathematical Society*, 19:815–820, 1968.

[Rog67] Hartley Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, New York, 1967.

[Šan68] N.A. Šanin. *Constructive Real Numbers and Constructive Function Spaces*, volume 21 of *Translations of Mathematical Monographs.* American Mathematical Society, Providence, 1968.

[She75] J.C. Shepherdson. Computation over abstract structures: serial and parallel procedures and Friedman's effective definitional schemes. In H.E. Rose and J.C. Shepherdson, editors, *Logic Colloquium'73*, volume 80 of *Studies in Logic and the Foundations of Mathematics*, pages 445–513, Amsterdam, 1975. North-Holland.

[She76] J.C. Shepherdson. On the definition of computable functions of a real variable. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 22:391–402, 1976.

[She85] J.C. Shepherdson. Algorithmic procedures, generalized Turing algorithms, and elementary recursion theory. In L.A. Harrington, M.D. Morley, A. Sčědrov, and S.G. Simpson, editors, *Harvey Friedman's Research on the Foundations of Mathematics*, volume 117 of *Studies in Logic and the Foundations of Mathematics*, pages 285–307, Amsterdam, 1985. North-Holland.

[SHT95] Viggo Stoltenberg-Hansen and John V. Tucker. Effective algebras. In S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science, Volume 4*, pages 357–527, Oxford, 1995. Clarendon Press.

[SHT99] Viggo Stoltenberg-Hansen and John V. Tucker. Concrete models of computation for topological algebras. *Theoretical Computer Science*, 219:347–378, 1999.

[Ste99] Kristian Jon Stewart. *Concrete and Abstract Models of Computation over Metric Algebras.* PhD thesis, University of Wales Swansea, Singleton Park, Swansea, SA2 8PP, UK, 1999.

[Sün94] Philipp Sünderhauf. *Discrete Approximation of Spaces, A Uniform Approach to Topologically Structured Datatypes and their Function Spaces.* PhD thesis, Fachbereich Mathematik, Technische Hochschule Darmstadt, Darmstadt, Germany, 1994.

[Sün95] Philipp Sünderhauf. A faithful computational model of the real numbers. *Theoretical Computer Science*, 151:277–294, 1995.

[SVW96] Nicoletta Sabadini, Sebastiano Vigna, and Robert F.C. Walters. A note on recursive functions. *Mathematical Structures in Computer Science*, 6:127–139, 1996.

[Tur36] Alan M. Turing. On computable numbers, with an application to the "Entscheidungsproblem". *Proceedings of the London Mathematical Society*, 42(2):230–265, 1936.

[Tur37] Alan M. Turing. On computable numbers, with an application to the "Entscheidungsproblem". A correction. *Proceedings of the London Mathematical Society*, 43(2):544–546, 1937.

[TZ] J.V. Tucker and J.I. Zucker. Computable functions and semicomputable sets on many-sorted algebras. In S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science, Volume 5*, Oxford. Clarendon Press.

[TZ88] J.V. Tucker and J.I. Zucker. *Program Correctness over Abstract Data Types, with Error-State Semantics*, volume 6 of *CWI Monographs*. North-Holland, Amsterdam, 1988.

[TZ99] J.V. Tucker and J.I. Zucker. Computation by 'While' programs on topological partial algebras. *Theoretical Computer Science*, 219:379–420, 1999.

[Wei] Klaus Weihrauch. *An Introduction to Computable Analysis*. (Book, to appear 1999).

[Wei85] Klaus Weihrauch. Type 2 recursion theory. *Theoretical Computer Science*, 38:17–33, 1985.

[Wei87] Klaus Weihrauch. *Computability*, volume 9 of *EATCS Monographs on Theoretical Computer Science*. Springer, Berlin, 1987.

[Wei93] Klaus Weihrauch. Computability on computable metric spaces. *Theoretical Computer Science*, 113:191–210, 1993. Fundamental Study.

[Wei95] Klaus Weihrauch. A simple introduction to computable analysis. Informatik Berichte 171, FernUniversität Hagen, Hagen, July 1995. 2nd edition.

[Wei97] Klaus Weihrauch. A foundation for computable analysis. In Douglas S. Bridges, Cristian S. Calude, Jeremy Gibbons, Steve Reeves, and Ian H. Witten, editors, *Combinatorics, Complexity, and Logic*, Discrete Mathematics and Theoretical Computer Science, pages 66–89, Singapore, 1997. Springer. Proceedings of DMTCS'96.

[WY96] Masako Washihara and Mariko Yasugi. Computability and metrics in a Fréchet space. *Mathematica Japonica*, 43(3):431–443, 1996.

[YMT99] Mariko Yasugi, Takakazu Mori, and Yoshiki Tsujii. Effective properties of sets and functions in metric spaces with computability structure. *Theoretical Computer Science*, 219:467–486, 1999.

[Zho98] Ning Zhong. Recursively enumerable subsets of R^q in two computing models: Blum-Shub-Smale machine and Turing machine. *Theoretical Computer Science*, 197:79–94, 1998.

Index

(δ_X, δ_Y)–computable function, 39
(δ_X, δ_Y)–computable operation, 41
μ–recursion, 10
μ-recursion, 28

admissible, 85
approximates, 37
arithmetical difference, 28

Baire's space, 37
birecursive over, 84
border, 84

Cantor's pairing function, 36
Cauchy representation, 89, 92
characteristic function, 66
classical decidability, 73
classically computable, 36, 37
classically recursive functions, 9
closure, 84
compact-open topology, 117
compactifier, 126
complete, 71
complete structure, 71
composition, 14
computable, 38, 42
computable G_δ-set, 37
computable operator, 37
computable over, 57
computable relation, 137
concatenation, 37
conservative extension, 58
continuous, 80
course-of-value recursion, 27
cylindrification, 41

decidable, 65
definition-by-cases operator, 27
dense, 91
distance function, 118, 127
domain, 12
domain over, 64
double birecursive over, 84
dual function, 122

effective structure, 45
effective via, 45
effectively categorical, 56
equality on , 72
equivalent, 40, 61, 99
essentials, 23
evaluation, 14, 120
evaluation function, 121
exponentiation, 14
extension, 13, 58

Fell topology, 127
finally semi-recursive, 65
finite operators, 15
finite union, 32

general recursive operator, 36
graph, 12

Hausdorff metric, 107
hit set, 127

identity, 35, 36
image, 13
inequality on, 74
initial constants and functions, 9

initial operation, 34
initial operations, 34
initially semi-recursive, 65
injective in the last component, 18
injective inversion, 18
interior, 84
inverse operation, 13
inversion, 14
iteration, 14

juxtaposition, 13

length function, 37
lifted function, 109
limit operator, 88, 91
Lipschitz continuous, 103
locally compact, 112
locally finite covering, 124
lower semi-continuous, 80

many-sorted prestructure, 34
metric product space, 98
metric sequence space, 98
minimization, 29
miss set, 127
modulus of continuity, 101
monotone, 37

natural, 36
natural numbers, 9
natural representation, 45
natural setting, 23
nice, 114
numberings, 76

omnipotent operation, 38
open balls, 91
operation over, 34
operations, 12

perfect, 57
perfect prefix structure, 59
prefix, 37
prefix structure, 58
preimage, 13
prestructure, 34
prestructures of, 34
primitive recursion, 10, 24
primitive recursive, 10
product, 13
product representation, 40
projection, 13
projection function, 16

range, 12
recursion operators, 9, 13
recursive, 65
recursive Banach space, 116
recursive constant over a structure, 35
recursive embedding, 63
recursive enumerability, 73
recursive exhausting sequence, 112
recursive isomorphism, 63
recursive metric space, 91
recursive metric subspace, 99
recursive operations over a structure, 35
recursive partition of unity, 125
recursive retraction, 42
recursive selector, 60
recursive structure, 43
recursive via, 43
recursively categorical, 56
recursively given sequence of (initially) semi-recursive sets, 67
recursively given sequence of finally semi-recursive sets, 67
recursively isomorphic, 63
recursively locally compact, 112
recursively related, 99
recursively separable, 99
reducible, 40, 61
representation, 39, 42

represented space, 39
represented structure, 42
restriction in the domain, 64
restriction in the range, 64
restriction in the source, 64
restriction in the target, 64

section, 31
semi-characteristic operation, 66
semi-distributivity, 20
semi-recursive, 65
separable Banach space, 116
separable metric space, 91
separates points, 117
sequence representation, 40
sequentialization, 15
set of all sequences, 13
set of finite sequences, 37
sets over, 34
simultaneous recursion, 26
source, 12
standard numbering of finite sequences, 37
standard representation, 57, 85
standard structure, 91, 116
strongly (δ_X, δ_Y)–computable, 39
strongly (δ_X, δ_Y)–computable operation, 41
strongly equivalent, 61
strongly recursive structure, 43
strongly recursive via, 43
strongly reducible, 61
structure, 34, 35
structure of integers, 76
structure of natural numbers, 35
structure of rationals, 76
subordinated, 125
substitution, 10, 23
substructure, 35
successor function, 35
support, 125
supremum norm, 117

target, 12
topological space over S, 80
total μ-recursion, 28
total minimization, 30
transposition, 14
twisted inversion, 14

uniform modulus of continuity, 101
uniformization, 72
union, 32, 59
universe, 34
upper semi-continuous, 80

Vietoris topology, 107

words, 37

zero-ary constant function, 35