Mathematical Structures in Computer Science (2020), **30**, pp. 173–233
doi:10.1017/S096012952000002X

CAMBRIDGE
UNIVERSITY PRESS

ARTICLE

Computable analysis with applications to dynamic systems

Pieter Collins*

Department of Data Science and Knowledge Engineering, Maastricht University, Maastricht, The Netherlands
*Corresponding author. Email: pieter.collins@maastrichtuniversity.nl

(Received 22 December 2018; revised 4 December 2019; accepted 7 January 2020)

Abstract

Numerical computation is traditionally performed using floating-point arithmetic and truncated forms of infinite series, a methodology which allows for efficient computation at the cost of some accuracy. For most applications, these errors are entirely acceptable and the numerical results are considered trustworthy, but for some operations, we may want to have guarantees that the numerical results are correct, or explicit bounds on the errors. To obtain rigorous calculations, floating-point arithmetic is usually replaced by interval arithmetic and truncation errors are explicitly contained in the result. We may then ask the question of which mathematical operations can be implemented in a way in which the exact result can be approximated to arbitrary known accuracy by a numerical algorithm. This is the subject of *computable analysis* and forms a theoretical underpinning of rigorous numerical computation. The aim of this article is to provide a straightforward introduction to this subject that is powerful enough to answer questions arising in dynamic system theory.

Keywords: Computable analysis; Turing machine; dynamic system

1. Introduction

In this paper, we develop a theory of computation for continuous mathematics. The aim of this paper is to give an exposition which is explicitly based on Turing machine model of computation, is powerful enough to study real computational problems arising in practice (taken from the areas of dynamical systems and control theory), yet is as straightforward and direct as possible, and uses terminology which is natural in classical topology and analysis.

The main idea is to consider which mathematical operations are *computable*, by which we mean that the result can be computed by a program running on a digital computer. Since we are dealing with objects from continuous mathematics, we are typically dealing with uncountable sets of objects (such as the real numbers), so we cannot specify an arbitrary object with a finite amount of data. However, the objects we consider form topological or metric spaces and can be *approximated* arbitrarily accurately with a finite amount of data. Hence, we describe an object by an infinite *stream* of data, but in such a way that useful information can be obtained from a finite amount of data. The inherent use of approximations means that there is a very close link between topology and representations; indeed any representation of a set of objects induces a natural topology on that set.

An operation is *computable* if a description of the result can be computed from a description of the arguments. At any stage of computation, only a finite amount of memory can be used, but most computations need an unbounded amount of internal memory to compute the complete output. The input of a computation can in principle be an arbitrary valid sequence of characters; we do not require that there be an effective procedure to determine whether a string is valid. In practice, inputs will typically be taken from a countable set of *computable* sequences, that is, those which can be computed by a Turing machine, or from a subset which can be described symbolically, for example, the rational numbers. One of the fundamental results is that only continuous functions can be computable; however, which operations are continuous depends on the topology, since this affects the amount of information which is present in the input or required in the output.

The exact details as to *how* objects of a set are described is immaterial in determining which operations are possible, as long as a description in one representation can be effectively converted to a description in another representation. We therefore look at equivalence classes of representations as defining a *type* of object and consider the computable operations between types. In general, there are inequivalent representations of a given topological space, but typically in practice, one of these is canonical. For example, there is a unique equivalence class of representations of the real numbers $\mathbb{R}$ for which arithmetic is computable, strict comparison is verifiable, and limits of strongly-convergent Cauchy sequences are computable. Hence, there is a canonical type $\mathcal{R}$ of the real numbers. From this type, we can canonically build up other types, such as Euclidean space $\mathcal{R}^n$, continuous functions $\mathcal{C}(\mathcal{R}^n; \mathcal{R}^m)$, and open sets $\mathcal{O}(\mathcal{R}^n)$.

A canonical way of describing an element of a countably based topological space is to list the basic open sets containing it. In this way, the basic open sets become the fundamental objects describing the space, rather than the points. This observation indicates strong links with *locale theory*, which can be seen as a kind of "pointless topology". In our exposition, we work with countable sub-bases, which gives an equivalent theory, but which allows some spaces, notably function spaces, to be treated more conveniently.

The theory presented is a model of *intuitionistic type theory* (Martin-Löf 1984). This means that it is always possible to construct finite products and function types, with the corresponding natural computable operations. However, not all types and computable operations can be constructed from a finite collection of base types; arbitrary subtypes are allowable, and sometimes it is necessary to return to first principles to show that an operation is computable or that a constructive definition is well-defined and matches the classical definition. For example, to prove computability of the solution of an ordinary differential equation, it is necessary to appeal to results of classical analysis to prove that a construction based on Euler time steps has the classical properties of a solution.

The ideas in this theory can be traced back to the intuitionistic logic of Brouwer (see van Stigt 1990) and the constructive mathematics of Markov (see Kushner 1999) and Bishop (Bishop and Bridges 1985), Mazur's approach based on recursive function theory (Mazur 1963) (see also Grzegorczyk 1957) and Eršov's theory of numberings (Ershov 1973, 1975, 1977). Although these theories deal with constructive and recursive mathematics rather than explicitly with machine-based computation, the idea of constructive existence is clearly linked.

The first link with computability was via the theory of Scott domains (Gierz et al. 2003), which were initially developed to give a semantics for programming languages, but which were soon recognized as a possible foundation for real analysis. There is a considerable body of work applying domain theory to various bodies of real analysis (e.g. Edalat 2009; Edalat and Sünderhauf 1999). However, the foundations of domain theory are based on lattice theory, and the notation and terminology are still heavily based on these foundations, rather than on the natural language for topology and analysis. Further, domain theory is usually not directly presented in terms of Turing computation, and an extra level of theory is still needed to give an explicit relationship with computation (though intuitively it is clear for experts how to proceed).

A seminal work providing a simplified theory of computable analysis based on type-2 effectivity, which is explicitly based on Turing computation and using natural language, was given by Weihrauch (2000). In this theory, *representations* are used to give a computational meaning to objects from continuous mathematics. Unfortunately, the elegance of the framework tends to get lost in a plethora of subtly different representations for different classes of objects, and in the necessity to always explicitly specify the representation used. From this point of view, our use of types represents an important notational simplification, which we hope also improves the readability and accessibility of the theory. Further, in our approach, the results of Weihrauch (2000), which are mostly restricted to Euclidean spaces, extend naturally to spaces which are not Hausdorff or locally compact.

An exposition of computable analysis focusing on complexity theory of real functions was given by Ko, based on the notion of an *oracle Turing machine*. In computer science, an oracle is a machine which can provide the answer to an unsolvable problem. The use of the word "oracle" here may be slightly confusing; it does not mean that computations are performed with a fictional computing device, but that the computations must be able to handle inputs which need not be generated by a computational process (e.g. in the form of a decimal expansion of an uncomputable real number). The resulting framework is equivalent to that of Weihrauch (for which the input tapes may contain uncomputable sequences).

Analog computing devices based around differential equation models are natural to consider when working with real numbers. Their computational power can be shown to be essentially equivalent to that of Turing machines (Bournez et al. 2006, 2013). However, Turing computation is some sense more practical, since it can be applied to discrete mathematical structures and those without a direct physical interpretation. Further, analog computation is affected by hardware component imperfections and by noise in the system, whereas digital computation is extremely robust. We shall see that digital Turing computation can rigorously and effectively solve differential equations, so it can simulate analog systems.

This article is organized as follows. In Section 2, we give an overview of Turing computability theory for discrete computations on words over some alphabet Σ and show how this can be extended to computations over sequences. We then give a formal definition of *naming systems*, by which elements of some arbitrary set can be related to objects with some "computational meaning". Section 3 is the heart of the paper. Here, we give a complete exposition of computable analysis for elementary classes of objects, including points, sets, and functions. In Section 4, we relate the material of Section 3 to concepts from classical topology and locale theory, with a view to developing the most natural versions for use in applied mathematics. We give a characterization of topological spaces which have a representation which adequately preserves the topological structure, describe the Scott topology on open sets, explain the concept of a sober space, and give an overview of the theory of core-compact and locally compact spaces. Finally, in Section 5, we give some applications to dynamical systems and control theory.

We emphasize that article is intended primarily as an "expository" paper (describing a theory) rather than as a "survey" paper (describing existing results) or a "research" paper (describing new results). It provides a self-contained exposition of the main theory of computable analysis needed for the problems in dynamical systems we tackle, and we provide complete and simplified proofs of the main results.

There have been a number of excellent Ph.D. theses in the area of computable analysis, especially those of Brattka (1998), Bauer (2000) and Schröder (2002a) and Battenfeld (2008). In particular, it was Schröder who first classified the topological spaces which can be given a representation capturing the topology. Indeed, Schröder's classification extends to weak limit spaces, a generalization of topological spaces which may have some applications in probability theory. Other important articles giving an exposition of a large part of the theory include those of Escardo (2004), Blanck (2000), Taylor (2008), and the tutorial of Brattka et al. (2008). Books specifically relating to computability in analysis include those of Pour-El and Ian Richards (1989), Ko (1991) and Weihrauch (2000). Other interesting books which contain deeper material in logic, domain theory and topos theory include those of Vickers (1989), Johnstone (2002a,b), Gierz et al. (2003), Clementino et al. (2004). For an introduction to type theory, see Martin-Löf (1984). Books relating to rigorous computation include Jaulin et al. (2001), Hansen and William Walster (2004), Aberth (2007), and Moore et al. (2009).

To avoid unnecessary complications, we have restricted ourselves to computability theory for topological spaces, and not for the more general class of weak limit spaces, since in almost all applications we use topological spaces. However, we have given most of the development of the theory for general topological spaces and have only restricted to Hausdorff and local-compact spaces where necessary. This is important, since types of open/closed/compact subsets of a space are not Hausdorff, and non-locally compact spaces quickly arise as function spaces and need to be covered to discuss solution sets of dynamic systems.

We have not used the language of Scott domains, though we have given an explicit exposition of the Scott topology on the open sets, as this is the topology induced by the canonical representation. We have given an introduction to point-free topology, since this is the natural way to obtain a representation for a countably based topological space, but have not used the language of locale theory. We have introduced the notion of a sober space, since this is required to give a link between a type-theoretic construction of compact sets by the subset predicate and the classical notion of a compact point-set.

2. Turing Computability

In this section, we give an outline of the theory of computability based on the standard notion of *Turing machine*, an abstract digital computing device with unlimited memory. The Turing model of computation is the standard accepted model of digital computation. There are many variants of the basic Turing machine and generalizations to computational models which are closer to the architecture of modern digital computers, but they all yield the same computable functions. When the model was introduced by Turing in (1937), it was envisaged that a human was performing the calculations, but the theory is completely appropriate for mechanical devices. The only assumption is that the machine has enough memory available to complete the task. The *Church–Turing thesis* asserts that any algorithmic procedure for performing a calculation is given by a Turing-computable function. It seems reasonable that the thesis even holds for *reliable* analog computing devices, due to external noise and constraints on space and energy. See e.g. Beggs and Tucker (2009), Ziegler (2009) for more detailed discussions of analog computation.

2.1 Turing machines

The standard model of computation is that of the *Turing* machine. This is a model of a process for performing computations in which only a fixed finite amount of information can be used at any stage, but for which an arbitrary amount of storage is available. In the standard model, computations are performed on one or more infinite *tapes* whose cells contain symbols from an alphabet Γ. For a digital computer, one might expect the alphabet to be $\{0, 1\}$, but for expository purposes we can take a more expressive alphabet, such as the ASCII or Unicode character sets. It is also useful to distinguish a *tape alphabet* Γ from the *input/output* alphabet Σ.

We first give a description of a multiple-tape Turing machine as a dynamical system; later, we will see what it means for the machine to compute a function.

Definition 2.1 (Turing machine). *A k-tape Turing machine is described by a tuple* (Γ, Q, τ)*, where* Γ *and* Q *are finite sets, and* $\tau : Q \times \Gamma^k \to Q \times \Gamma^k \times \{-1, 0, +1\}^k$ *describes the* transition function.

The action of a Turing machine is as follows. The state of the machine is given by $(q, h_1, \ldots, h_k, \vec{s}_1, \ldots, \vec{s}_k)$*, where* $q \in Q$*, each* $h_i \in \mathbb{N}$ *and each* $\vec{s}_i \in \Gamma^\omega$*. The next state* $(q', h'_1, \ldots, h'_k, \vec{s}\,'_1, \ldots, \vec{s}\,'_k)$ *is determined by taking*

$$\big(q', (s'_{1,h_1}, \ldots, s'_{k,h_k}), (\delta_1, \ldots, \delta_n)\big) = \tau\big(q, (s_{1,h_1}, \ldots, s_{k,h_k})\big), \tag{1}$$

and setting

$$h'_i = \max\,(h_i + \delta_i, 0) \text{ for } i = 1, \ldots, k. \tag{2}$$

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Informally, the value of $q \in Q$ is the register state *of the machine and is a kind of "program counter". The h_i represents the position of the "tape head" for the ith tape. For each computational step starting at register state q, the machine scans the values $s_{1,h_1}, \ldots, s_{k,h_k}$ at the tape head, replaces them with new values $s'_{1,h_1}, \ldots, s'_{k,h_k}$, shifts the tape head left or right depending on the value of $\delta_1, \ldots, \delta_k$, and updates the register state to q'.*

Remark 2.2. In the definition given here, we "shift" the tapes heads left and right to scan new symbols. Sometimes, the symbols $\{\mathtt{L}, \mathtt{N}, \mathtt{R}\}$ are used instead of $\{-1, 0, +1\}$ for the δ_i. If $h_i = 0$ and $\delta_i = -1$, then the tape head does not move; an alternative would be for this to be an error. An equivalent model sometimes used in the literature is to have shiftable tapes rather than movable tape heads. This model yields the same computable functions in Definition 2.3.

Now suppose Γ is an alphabet containing a special blank symbol $\sqcup$, and $\Sigma \subset \Gamma \setminus \{\sqcup\}$. Define an encoding $\iota : \Sigma^* \to \Gamma^\omega$ by taking $(\iota(w))_j = w_j$ for $j = 0, \ldots, |w| - 1$, and $(\iota(w))_j = \sqcup$ otherwise. Since $\sqcup \notin \Sigma$, the encoding ι is injective, so w can be unambiguously recovered from $\vec{s} = \iota(w)$.

When using multiple tapes, we can separately consider *input*, *output*, and *work* tapes. The work tape and output tape start off completely blank. An input tape can only be read from and contains the initial input, the output tape can only be written to on blank spaces, but not subsequently altered, and contains the final output. Formally, the i^{th} tape is *unidirectional* if for any transition, we always have $\delta_i \in \{0, +1\}$, is *read-only* if we always have $s'_{i,h_i} = s_{i,h_i}$ (so $\vec{s}_i$ is constant), and is *write-only-once* if $s'_{i,h_i} = s_{i,h_i}$ or $s_{i,h_i} = \sqcup$. A unidirectional read-only tape is an *input tape*. A unidirectional write-only-once tape such that $\delta = +1$ implies $s' \neq \sqcup$ is an *output tape*.

We now show how Turing machines can be used to compute partial word functions $(\Sigma^*)^m \rightharpoonup (\Sigma^*)^n$. We need to consider partial functions since not all words should be considered valid inputs; for example, the string `2/3/4` is not a valid description of a rational number. To distinguish "ordinary" computation on words from computation on sequences (which will be introduced in Section 2.2), we call this *type-one* Turing computation.

Definition 2.3 (Type-1 Turing computation). *Let (Γ, Q, τ) be a k-tape Turing machine where Γ contains a special* blank *character $\sqcup$, and $\Sigma \subset \Gamma \setminus \{\sqcup\}$. Let $q_0, q_f \in Q$ be the* initial *and* final *states, respectively. Let $m, n \in \mathbb{N}$ define the number of* input *and* output *tapes, respectively, such that $m + n \leq k$.*

Then, $M = (\Sigma, m, n, \Gamma, Q, q_0, q_f, \tau)$ defines a partial function $(\Sigma^*)^m \to (\Sigma^*)^n$ as follows: For input $(w_1, \ldots, w_m) \in (\Sigma^*)^m$, the initial state is given by $q = q_0$, $\vec{s}_i = \iota(w_i)$ for $i = 1, \ldots, m$ and $\vec{s}_i = \sqcup^{\mathbb{Z}}$ otherwise. The computation proceeds as given by Definition 2.1 until the register state q is equal to q_f, at which point the machine halts. The computation is valid *if the machine halts, and in the halting state, there exist $v_i \in \Sigma^*$ such that $\vec{s}_{m+i} = \iota(v_i)$ for $i = 1, \ldots, n$. The function is defined on all inputs $(w_1, \ldots, w_m)$ for which the computation is valid, and the result of the computation is the tuple $(v_1, \ldots, v_n)$.*

A partial function $(\Sigma^*)^m \rightharpoonup (\Sigma^*)^n$ is computable *if it is the function computed by some Turing machine.*

An example of the input and output of a Turing machine computing the product of two integers is given in Figure 1.

Remark 2.4. A theory of computable word functions can be developed without separate "input", "output" and "work" tapes. Indeed, the computability theory can be developed for single-tape machines which replace the initial contents of the tape (the input) by the output. We use the definition above for compatibility with the definition of type-two computability in Section 2.2.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

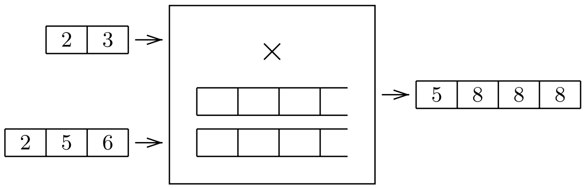

Figure 1. A Turing machine computing the product of two integers in decimal form.

Remark 2.5. The computability theory could equally well be presented using partial recursive functions on $\mathbb{N}$ instead of Turing computable functions on Σ^*. However, for a meaningful *complexity* theory, we need to use a finite alphabet Σ.

Remark 2.6. In order to show that a function is computable, we in principle need to explicitly construct a Turing machine which computes the function. In general, this is a tedious exercise in Turing machine programming. In this article, in the few cases where this is necessary, we shall merely give a sketch of how the function could be computed, without explicitly describing a Turing machine.

The most important properties of Turing computability are summarized in the following theorem.

Theorem 2.7.

(a) *If* $\xi : (\Sigma^*)^l \rightharpoonup (\Sigma^*)^m$ *and* $\eta : (\Sigma^*)^m \rightharpoonup (\Sigma^*)^n$ *are computable, then* $\zeta = \eta \circ \xi : (\Sigma^*)^l \rightharpoonup (\Sigma^*)^n$ *is computable, where we take* $\mathrm{dom}\,(\eta \circ \xi) = \xi^{-1}(\,\mathrm{dom}\,(\eta))$.
(b) *There is a computable* tupling *function* $\tau : \Sigma^* \times \Sigma^* \to \Sigma^*$ *and computable* projections $\pi_1, \pi_2 : \Sigma^* \rightharpoonup \Sigma^*$ *such that* $\pi_i(\tau(w_1, w_2)) = w_i$ *for* $i = 1, 2$.
(c) *There is a* universal *Turing machine* $\mathcal{U}$ *computing a function* $\varepsilon : \Sigma^* \times \Sigma^* \rightharpoonup \Sigma^*$ *such that for every Turing machine* $\mathcal{M}$ *computing a function* $\phi : \Sigma^* \rightharpoonup \Sigma^*$*, there is a word* $a \in \Sigma^*$ *such that* $\varepsilon(a, w) = \phi(w)$ *for all* $w \in \Sigma^*$.
(d) *There is a computable function* $\sigma : \Sigma^* \times \Sigma^* \rightharpoonup \Sigma^*$ *such that* $\epsilon(\sigma(a, v), w) = \epsilon(a, \tau_2(v, w))$ *for all* $a, v, w \in \Sigma^*$.

Part (a) shows that the composition of computable functions is computable. Roughly speaking, the computation can be carried out by running a machine computing ξ until it halts and then running the computation of η. Part (b) shows that the contents of two input tapes can be combined into one, and the contents recovered later. A (slightly) simpler case is that of computing a function $\tau : \Sigma^* \times \Sigma^* \to \hat{\Sigma}^*$, where $\hat{\Sigma}$ contains a symbol '$\texttt{,}$' not present in Σ, and $\tau(w_1, w_2) = w_1\,\texttt{,}\,w_2$. To avoid introducing a new symbol, we can take $\tau(w_1, w_2) = \texttt{1}^{|w_1|}\texttt{0}w_1w_2$ to recover the break between w_1 and w_2. Part (c) asserts the existence of a *universal* Turing machine, that is, a machine with two inputs, one of which is a "program" for the computation of another machine. For example, assuming $Q \subset \Sigma$ and $\texttt{L}, \texttt{N}, \texttt{R} \in \Sigma$, we can encode a transition $(q, s) \to (q', s', \delta)$ by the string $q\,\texttt{,}\,s\,\texttt{,}\,q'\,\texttt{,}\,s'\,\texttt{,}\,\delta$. Part (d) is Kleene's s-m-n theorem (Kleene 1936),[1] which shows that given a computable function of two variables $\xi(v, w)$ with program a, we can compute the program of the function taking w to $\xi(v, w)$ from a and v.

For more details on type-one Turing computation, see Sipser (1996).

2.2 *Type-two effectivity*

When dealing with computation on objects from continuous mathematics, we shall need functions on sequences $\Sigma^\omega \rightharpoonup \Sigma^\omega$ or, more generally, $(\Sigma^\omega)^m \rightharpoonup (\Sigma^\omega)^n$. In order to compute such a function, a machine needs to run forever, reading symbols from the *input* tape(s) and writing to

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

the *output* tape, with access to finitely many internal *work* tapes of unbounded length. Symbols from the output tape may not be overwritten once they have been produced. A computational run is *valid* if the machine writes infinitely many symbols to the output tape. A formal definition is given below. See Weihrauch (2000, Chapter 2) for a comprehensive treatment.

Definition 2.8 (Type-two computation). *Let (Γ, Q, τ) be a k-tape Turing machine where Γ contains a special* blank *character* $\sqcup$, *and* $\Sigma \subset \Gamma \setminus \{\sqcup\}$. *Let* $q_0, q_f \in Q$ *be the* initial *and* final *states, respectively. Let* $m, n \in \mathbb{N}$ *such that* $m + n \leq k$ *define the number of* input *and* output *tapes, respectively. We require that the input tapes* $\vec{s}_1, \ldots, \vec{s}_m$ *and output tapes* $\vec{s}_{m+1}, \ldots, \vec{s}_{m+n}$ *are unidirectional, in which the input tapes are read-only and the output tapes are write-only. Define an encoding* $\iota : \Sigma^\omega \to \Gamma^{\mathbb{Z}}$ *by taking* $(\iota(p))_j = p_j$ *for* $j \geq 0$ *and* $(\iota(w))_j = \sqcup$ *for* $j < 0$.

Then, M defines a partial function $(\Sigma^\omega)^m \to (\Sigma^\omega)^n$ as follows: For input $(p_1, \ldots, p_m) \in (\Sigma^\omega)^m$, the initial state is given by $q = q_0$, $\vec{s}_i = \iota(p_i)$ for $i = 1, \ldots, m$, and $\vec{s}_i = \sqcup^{\mathbb{Z}}$ otherwise. The computation proceeds as given by Definition 2.1. The computation is valid *if the machine does not halt and also writes infinitely many symbols on each output tape. The result of the computation is $(r_1, \ldots, r_n)$, where each r_i is defined by $(r_i)_j = s_{m+i,j}$ all j.*

Note that the question of whether a type-two Turing machine with no inputs produces a valid (i.e. infinite) output is undecidable.

Definition 2.9 (Machine computability). *A partial function* $\eta : (\Sigma^\omega)^m \rightharpoonup (\Sigma^\omega)^n$ *is* (machine) computable *if it can be computed by a type-two Turing machine.*

A sequence $p \in \Sigma^\omega$ *is* (machine) computable *if there is a type-two Turing machine with no inputs which outputs p.*

Since an output tape of one machine may be used as the input of another, and the resulting combination can be realized by a single machine, we have the following result:

Proposition 2.10 (Composition of machine-computable functions). *Let* $\eta : (\Sigma^\omega)^l \rightharpoonup (\Sigma^\omega)^m$ *and* $\zeta : (\Sigma^\omega)^m \rightharpoonup (\Sigma^\omega)^n$ *be computable. Then* $\zeta \circ \eta : (\Sigma^\omega)^l \rightharpoonup (\Sigma^\omega)^n$ *is computable with* $\mathrm{dom}\,(\zeta \circ \eta) = \eta^{-1}(\,\mathrm{dom}\,(\zeta))$.

Note that the definition ensures that if $\eta(p)$ is an invalid input to ζ, then p is not in the domain of $\zeta \circ \eta$.

Just as in the type-one case, we can combine the data on multiple input tapes into a single output tape and later recover the original data.

Proposition 2.11 (Tupling). *For any* $n \in \mathbb{N}$, *there is a machine-computable function* $\tau_n : (\Sigma^\omega)^n \to \Sigma^\omega$, *and machine-computable functions* $\pi_{n,i} : \Sigma^\omega \rightharpoonup \Sigma^\omega$ *such that for any* $i \leq n$, $\pi_{n,i}(\tau_n(p_1, \ldots, p_n)) = p_i$.

The natural tupling function is given by $(\tau_n(p_1, \ldots, p_n))_{in+j} = (p_j)_i$ for $i \in \mathbb{N}$ and $j = 1, \ldots, n$, and the projections by $(\pi_{n,i}(p))_j = p_{in+j}$. It is clear that these functions can be computed by a type-two Turing machine.

In the sequel, we will drop the subscript n when this is clear from the context.

As well as the finitary tupling functions $(\Sigma^*)^n \to \Sigma^*$ and $(\Sigma^\omega)^n \to \Sigma^\omega$, we can also define mixed and infinite tupling functions. The infinite tupling function $\tau_\infty : (\Sigma^\omega)^\omega \to \Sigma^\omega$ can be given by

$$\tau_\infty : (p_0, p_1, \ldots) = q \text{ if } q_{g(i,j)} = (p_i)_j, \text{ where } g(i,j) = (i+j)(i+j+1)/2 + j \text{ for } i, j \in \mathbb{N}. \tag{3}$$

It is clear that g is a computable bijection $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$. If $\widehat{\Sigma} \supset \Sigma$ and contains a special *separator* character ‘,’ not in Σ, then tupling $(\Sigma^*)^\omega \to \widehat{\Sigma}^\omega$ can be performed by taking $\tau_\infty(w_0, w_1, \ldots) = w_0\,,w_1\,,w_2\,,\cdots$. To tuple $(\Sigma^*)^\omega \to \Sigma^\omega$, we can choose an “escape” character ‘\’ as well as a separator ‘,’ and replace any occurrence of ‘\’ in w_i with the string “\\” and ‘,’ with “\,”.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

We shall also want to construct sequences by tupling infinitely many words from a subset W of Σ^*. We say W is *prefix-free* if v is not a prefix of w for all $v, w \in W$. For a prefix-free subset of W, if $v_1, \ldots, v_m, w_1, \ldots, w_n$ are words in W with $v_1 v_2 \cdots v_m = w_1 w_2 \cdots w_n$, then $m = n$ and $v_i = w_i$ for all i. In other words, tupling $W^\omega \to \Sigma^\omega$ can be accomplished by concatenation.

Notation 2.12. *We henceforth write $\langle \cdot, \cdots, \cdot \rangle$ for any tupling function, and subscript $\langle \cdot \rangle_i$ for the projection onto the* ith *element. We write $w \lhd p$ if $p = \langle w_0, w_1, \ldots \rangle$ and $w = w_i$ for some $i \in \mathbb{N}$.*

We now consider some topological aspects of computable functions.

Definition 2.13 (Topology on Σ^ω). *Define a topology on Σ^ω by taking the basic open sets to be the* cylinder sets

$$C_w = \{p \in \Sigma^\omega \mid p_i = w_i \text{ for } i = 0, \ldots, |w| - 1\} \tag{4}$$

where $w \in \Sigma^*$ is a word of length $|w|$.

The following theorem shows that machine-computable functions are continuous relative to the topology defined by the cylinder sets.

Theorem 2.14 (Machine computability implies continuity). *Any machine-computable function $\eta : (\Sigma^\omega)^m \rightharpoonup (\Sigma^\omega)^n$ is continuous.*

Sketch of proof. For simplicity, consider $\eta : \Sigma^\omega \rightharpoonup \Sigma^\omega$. Consider $p \in \mathrm{dom}\,(\eta)$. For all $n \in \mathbb{N}$, there exists $m \in \mathbb{N}$ such that n symbols of $\eta(p)$ have been written to the output tape after m computational steps, and hence after at most m digits of p have been read. Then for any $q \in \mathrm{dom}\,(\eta)$ such that $q|_{[0,m)} = p|_{[0,m)}$, the output after m computational steps is the same as that of p, so $\eta(q)|_{[0,n)} = \eta(p)|_{[0,n)}$. Note that setting $v = p|_{[0,m)}$ and $w = \eta(p)|_{[0,n)}$, we have shown that $C_v \subset \eta^{-1}(C_w)$. Hence η is continuous at p. □

This result provides the basis for the main results on *uncomputability*; it suffices to prove discontinuity.

Recall that a subset of a topological space is a G_δ*-set* if it is a countable intersection of open sets; in particular, any open set is a G_δ-set. In general, even for a locally compact Hausdorff space, not every closed set is a G_δ-set; a counterexample is the Tychonoff plank (Steen and Seebach 1978), but any closed subset of Σ^ω is a G_δ set.

Proposition 2.15 (Natural extension). *Any continuous function $\eta : (\Sigma^\omega)^m \rightharpoonup (\Sigma^\omega)^n$ extends naturally to a continuous function on a G_δ-set.*

Sketch of proof. For simplicity, consider $\eta : \Sigma^\omega \rightharpoonup \Sigma^\omega$. Define a partial function $\tilde{\eta} : \Sigma^* \rightharpoonup \Sigma^*$ by taking $\mathrm{dom}\,(\tilde{\eta}) = \{v \in \Sigma^* \mid \mathrm{dom}\,(\eta) \cap C_v \neq \emptyset\}$, and $w = \tilde{\eta}(v)$ of maximal length such that $\eta(C_v \cap \mathrm{dom}\,(\eta)) \subset C_w$. Note that if v_1 is a prefix of v_2, then $\tilde{\eta}(v_1)$ is a prefix of $\tilde{\eta}(v_2)$. Then for $p \in \Sigma^\omega$, if $C_v \cap \mathrm{dom}\,(\eta) \neq \emptyset$ for every prefix v of p, and $|\tilde{\eta}(v)| \to \infty$ as $|v| \to \infty$, then we can define $\hat{\eta}(p) = q$ where $\{q\} = \bigcap \{C_{\tilde{\eta}}(p|_{[0,k)}) \mid k \in \mathbb{N}\}$. Clearly, $\hat{\eta}$ is an extension of η, and $\mathrm{dom}\,(\hat{\eta})$ is a G_δ-set. □

The set of continuous partial functions $\eta : (\Sigma^\omega)^m \rightharpoonup (\Sigma^\omega)^n$ with G_δ-domain has continuum cardinality. This means that the continuous partial functions $(\Sigma^\omega)^m \rightharpoonup (\Sigma^\omega)^n$ with G_δ-domain can also be represented by sequences. This is a similar closure property for continuous functions provided by the universal Turing machine result for computable functions; computable word functions can be represented by words and continuous stream functions can be represented by streams.

We can now present the main result of this section, the universal Turing machine (UTM) theorem, and Kleene's s-m-n theorem, which is a type-two version of Theorem 2.7(c), (d). Note the interplay between the *finite* description of Turing machines by words, and the *infinite* description of continuous functions by sequences.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Theorem 2.16 (Universal Turing machines/s-m-n). *There exist machine-computable functions $\varepsilon : \Sigma^\omega \times \Sigma^\omega \rightharpoonup \Sigma^\omega$, $\mu : \Sigma^* \to \Sigma^\omega$ and $\sigma : \Sigma^\omega \times \Sigma^\omega \rightharpoonup \Sigma^\omega$ with the following properties:*

(a) For any continuous function $\eta : \Sigma^\omega \rightharpoonup \Sigma^\omega$ with G_δ-domain, there exists $a \in \Sigma^\omega$ such that $\varepsilon(a, \cdot) = \eta$.
(b) If η is computable, there exists $c \in \Sigma^*$ such that $\varepsilon(a, \cdot) = \eta$ where $a = \mu(c)$; in particular, a is computable.
(c) For all $a, p, q \in \Sigma^\omega$, $\varepsilon(\sigma(a, p), q) = \varepsilon(a, \tau_2(p, q))$, where τ_2 is the tupling function.

The function ε is the *evaluation* function. The first argument a is an encoding of the continuous partial function $\eta : \Sigma^\omega \rightharpoonup \Sigma^\omega$ as an element of Σ^ω. Further, if the partial function η is computable, then it has a computable encoding, and this encoding can be generated by a finite program. Note that there may be several different a corresponding to the same function f, some of which may be uncomputable even if f is computable. The function σ is a type-two analog of the function s in the s-m-n theorem. In particular, given a computable function η of two variables, there is a computable sequence a such that $\eta(p, q) = \varepsilon(a, \tau_2(p, q)) = \varepsilon(\sigma(a, p), q)$.

Sketch of Proof. Given a continuous function $\eta : \Sigma^\omega \rightharpoonup \Sigma^\omega$, we can define the set of all pairs $(v, w) \in \Sigma^* \times \Sigma^*$ such that $\eta(p)|_{|w|} = w$ whenever $p_{|v|} = v$. (In other words, any input with prefix v results in an output with prefix w.) By tupling in $(\Sigma^* \times \Sigma^*)^\omega$, we can list *all* such pairs as a sequence in Σ^ω and reconstruct the function η from this list. If η is computable, then the list can be constructed from a description of the Turing machine. □

Remark 2.17. An important consequence of this theorem is that it is possible to simulate the evaluation of a function as part of another computation. Indeed, it is even possible to simulate countably-many such computations in parallel.

2.3 Computation on words and sequences

We now have two theories of computation, type-one computability, which deals with finite computations on words Σ^*, and type-two computation, which deals with infinite computations on sequences Σ^ω. In many situations, we wish to combine these two types of computation; we may, for example, wish to compute a function with mixed arguments, for example, $\Sigma^* \times \Sigma^\omega \to \Sigma^\omega$, compute an infinite sequence from a finite input $\Sigma^* \to \Sigma^\omega$, or convert an infinite computation to a finite one.

At first sight, it seems straightforward to place type-one computation in the framework of type-two computation. Both words in Σ^* and sequences in Σ^ω are naturally expressed as sequences over the alphabet $\widehat{\Sigma} = \Sigma \sqcup \{␣\}$. Where the result of a computation on an output tape is a word $w \in \Sigma^*$, the computation continues forever, writing $w_0 \cdots w_{n-1}$ on the output tape, optionally followed by an arbitrary number of '␣'. Unfortunately, this approach suffers from the subtle drawback that it is impossible to know when a word has been completed, or whether a ␣ is merely an as-yet unwritten space.

The simplest way of correcting this defect is to introduce a special "carriage return symbol" '↵' to signify the end of a word. Hence, a word $w = w_0 w_1 \cdots w_{n-1}$ is encoded on an input tape as $w_0 w_1 \cdots w_{n-1}$↵␣␣ $\cdots$; note that all the symbols after the '↵' are taken to be blanks. Once the '↵' symbol has been encountered, the result for that tape is known. In principle, the computation could be halted once the result on all output tapes is known.

Definition 2.18 (Mixed Turing computation). *Computation on words Σ^* and sequences Σ^ω can be combined in a type-two Turing computation with alphabet Γ containing special symbols ␣, ↵ $\notin \Sigma$. An element p of Σ^ω is encoded on a tape by $\vec{s} \in \Gamma^{\mathbb{Z}} = \iota(p)$ with $s_i = p_i$ for $i \geq 0$ and $s_i =$ ␣*

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

otherwise. An element w of Σ^* is encoded on a tape by $\vec{s} = \iota(p)$ with $s_i = w_i$ for $0 \leq i < |w|$, $s_{|w|} = \lrcorner$, and $s_i = \sqcup$ otherwise.

The computation proceeds as for a standard type-two computation. A computation is valid *if it runs forever, writing either $\iota(p)$ for some $p \in \Sigma^\omega$ or $\iota(w)$ for some $w \in \Sigma^*$. Since every output tape starts completely blank, it is not necessary to write infinitely many symbols to an output tape for which the result is an element of Σ^*; instead the computation of that element is finished once the '$\lrcorner$' symbol has been written. If all outputs are elements of Σ^*, the computation may halt once all output tapes have had a '$\lrcorner$' written on them.*

Note that if we wish to restrict to purely using the binary alphabet, we can encode words $w \in \{0, 1\}^*$ as sequences in $\{0, 1\}^\omega$ by describing the length of w in some way. A simple method is to encode the word $w = w_0 \cdots w_{n-1}$ as $\hat{w} = 1w_0 1w_1 1 \cdots 1w_{n-1}00 \cdots$. In other words, each element of w is preceded by '1', and the word is followed by an infinite sequence of '0'. Alternatively, we can encode w as $1^{|w|}0w00\cdots$.

When performing type-two computations in practice, we cannot in general wait for the infinite amount of time needed to obtain the complete answer. Instead, we may wish to terminate the computation after a fixed number of output digits. This means that a type-two machine-computable function $\eta : \Sigma^\omega \rightharpoonup \Sigma^\omega$ would be replaced by a function $\bar{\eta} : \Sigma^* \times \Sigma^\omega \rightharpoonup \Sigma^*$, where $\bar{\eta}(v, p) = \eta(p)|_{|v|}$. Additionally, the infinite input p is itself likely to be generated from a finite input $w \in \Sigma^*$ by some function $p = \xi(w)$, for example, p is a decimal expansion of a rational defined by the word w. We then obtain a purely finite computation of a function $\zeta : \Sigma^* \times \Sigma^* \to \Sigma^*$ defined by $\zeta(v, w) = \bar{\eta}(v, \xi(w))$.

Given that we can reduce type-two computation to type-one computation, one might wonder what is the point of studying type-two computation in its own right. The main reason is that elements of Σ^ω can be used to describe objects from mathematical analysis, such as real numbers or continuous functions, completely, whereas restricting to type-one computation either forces one to always work with approximations or to only consider computable elements. In the former case, we are often required to work with messy ϵ-δ style considerations, while the latter is unnatural from the point of view of analysis. Even if we are ultimately only interested in finite computation, the use of sequences to represent intermediate results yields a much more elegant and simple theory than one based purely on finite computation.

2.4 Computability induced by representations

We have seen how computability theory can be developed for partial functions $(\Sigma^\omega)^m \rightharpoonup (\Sigma^\omega)^n$. However, we are really more interested in computations on more general mathematical objects, such as $\mathbb{N}$ or $\mathbb{R}$, and so we need a way to relate computation over Σ^ω to that over more general spaces.

The basic idea is that each element x of a set of interest X should be described by one or more sequences $p \in \Sigma^\omega$. We note that the description of an element need not be unique; for example, in the decimal representation, $0.999\cdots$ $1.000\cdots$ denote the same real number. Further, not every sequence needed correspond to an element of X; as an example, the strings $1000\cdots$ and $1.0.00\cdots$ are both invalid as decimal representations of a real number.

Definition 2.19 (Representations). *Let Σ be a fixed alphabet and X be a set. Then, a* representation *of X is a partial surjective function $\delta : \Sigma^\omega \rightharpoonup X$. For $x \in X$, an element $p \in \Sigma^\omega$ such that $\delta(p) = x$ is called a δ-name of x.*

Remark 2.20. In the definition of representation, there is no restriction placed on the domain. This is because there are spaces for which the domain is necessarily complicated. Ideally, we would like the domain to be an open set, so we can tell after a finite amount of data that a sequence is a valid input, or a closed set, so we can tell that a sequence is invalid. Frequently, there will be invalid

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

sequences, but most words can be continued to both valid and invalid inputs. Where possible, it would be advantageous to have a decision procedure for whether a given word can be extended to a valid name.

Definition 2.21 (Equivalence). *A representation δ_1 of X is said to* reduce *to δ_2, denoted $\delta_1 \leq \delta_2$ if there is a machine computable function $\eta : \Sigma^\omega \rightharpoonup \Sigma^\omega$ such that* $\mathrm{dom}\,(\delta_2 \circ \eta) \supset \mathrm{dom}\,(\delta_1)$ *and* $\delta_1 = \delta_2 \circ \eta|_{\mathrm{dom}\,(\delta_1)}$. *In other words, given any δ_1-name p of x, $\eta(p)$ is a δ_2-name of x. Representations δ_1 and δ_2 are* equivalent *if $\delta_1 \leq \delta_2$ and $\delta_2 \leq \delta_1$.*

Remark 2.22. A representation δ_1 reduces to δ_2 if a δ_1-name contains *as much or more information* about the object then a δ_2-name. The use of $\delta_1 \leq \delta_2$ to denote "δ_1 reduces to δ_2" is standard.

Since elements of finite and denumerable sets can be described by a finite amount of information, and these are often important in their own right, we define a similar notion for naming systems in terms of Σ^*.

Definition 2.23 (Notations). *Let Σ be a fixed alphabet and X be a countable set. Then a* notation *of X is a partial surjective function $\nu : \Sigma^* \rightharpoonup X$.*

Representations and notations are both *naming systems* since they relate general sets to words and sequences over some alphabet.

Remark 2.24 (Realizations). More generally, suppose we have a set Γ for which we already have some kind of "computational structure" defined. This could be Σ^ω, with computations described by type-two Turing machines, or $\mathbb{N}$ with computations described by partial recursive functions. Then, we can induce a computational structure on another set X using the computational structure on Γ. The traditional way of doing this is by a *realization relation* $\Vdash$ on $\Gamma \times X$. A realization is *sound* if for all $x_1, x_2 \in X$, if $\gamma \Vdash x_1$ and $\gamma \Vdash x_2$, then $x_1 = x_2$, and is *complete* if $\forall x \in X,\ \exists \gamma \in \Gamma,\ \gamma \Vdash x$.

A sound realization $\Vdash$ induces a partial function $\rho : \Gamma \rightharpoonup X$ by $\rho(\gamma) = x \iff \gamma \Vdash x$, and this function ρ is surjective if $\Vdash$ is complete. An element $\gamma \in \Gamma$ such that $\rho(\gamma) = x$ is called a *ρ-name* of c. Hence, a representation of X can be seen as sound and complete realization of X in Σ^ω. It is trivial that if $\delta : \Sigma^\omega \rightharpoonup \Gamma$ is a representation of Γ, and $\rho : \Gamma \rightharpoonup X$ is a sound and complete realization of X, then $\rho \circ \delta$ is a sound and complete realization of X.

Given a function $f : X \to Y$, a partial function $\eta : \Sigma^\omega \rightharpoonup \Sigma^\omega$ is a valid description of f if it translates any sequence p denoting x into a sequence q denoting $y = f(x)$. The function f is then *computable* if η is computable.

Definition 2.25 (Computability induced by representations). *A function $f : X_1 \times \cdots \times X_k \to X_0$ is* $(\delta_1, \ldots, \delta_k; \delta_0)$-computable *if there is a machine computable function $\eta : \Sigma^\omega \times \cdots \times \Sigma^\omega \rightharpoonup \Sigma^\omega$ such that*

$$f(\delta_1(y_1), \ldots, \delta_k(y_k)) = \delta_0(\eta(y_1, \ldots, y_k))$$

whenever the left-hand side is defined. We say that η is a realizer *for f.*

Note that it does not make sense to say whether a *representation* itself is computable. This is because representations are used to *induce* a computability structure on another set. However, most sets encountered in practice have a canonical equivalence class of "natural" representations, and we can, of course, consider the computability of a representation with respect to a natural representation.

Remark 2.26 (Multi-functions). As noted in Luckhardt (1977), not all computationally important operations can be realized by a computable function. A typical example is making a decision based on the value of a real number; we shall see in Section 3.2 that we must allow an overlap region on which both true and false values are allowed.

A *multifunction* $F : X \rightrightarrows Y$ is a function $X \to \mathcal{P}(Y)$, where $\mathcal{P}(Y)$ is the power-set of Y. An *intensional selection* (in the literature also known as a *realization*) of F is a machine-computable function $\eta : \Sigma^\omega \to \Sigma^\omega$ such that $\mathrm{dom}\,(\eta) \supset \mathrm{dom}\,(\delta_X)$ and $\delta_Y(\eta(p)) \in F(\delta_X(p))$ for all $p \in \mathrm{dom}\,(\delta_X)$. Here, the word *intensional* means that the image of x chosen depends on the description (name) of x, and not just on x itself.

Note that an *extensional* selection of F would be a function $f : X \to Y$ such that $f(x) \in F(x)$ for all x.

In this paper, we will mostly consider multivalued functions as representing nondeterministic dynamic systems. For these applications, we are interested in properties of the function $X \to \mathcal{P}Y$ itself, and not in its selections.

3. Computable Analysis

We now present the basic constructions of number, function, and set types we require for the foundations of continuous mathematics. These types and their computable operations are listed in Appendix A.1.

3.1 Review of classical notions

We begin by recalling basic concepts from topology and analysis. We denote the power-set of a set X, that is, the set of subsets of X by $\mathcal{P}(X)$.

Recall that a topological space $(X, \mathcal{T})$ consists of a set X and a set $\mathcal{T}$ of *open* subsets of X satisfying the axioms: $\emptyset, X \in \mathcal{T}$, if $U_1, U_2 \in \mathcal{T}$, then $U_1 \cap U_2 \in \mathcal{T}$, and if $\mathcal{U} \subset \mathcal{T}$, then $\bigcup_{U \in \mathcal{U}} U \in \mathcal{T}$. The open sets thus support finite intersections and arbitrary unions. A set is *closed* if its complement is open.

If the topology is clear from the context, we shall not mention it explicitly and simply say "X is a topological space".

A *base* $\mathcal{B}$ for a topological space is a set of subsets such that any open set is a union of sets in $\mathcal{B}$. In other words, $\mathcal{B} \subset \mathcal{P}(X)$ is such that for any $U \in \mathcal{T}$, there exists $\mathcal{U} \subset \mathcal{B}$ such that $U = \bigcup \mathcal{U}$. A topological space is *countably based* or *second-countable* if it has a countable base.

If $(X, \mathcal{T}_X)$ and $(Y, \mathcal{T}_Y)$ are topological spaces, then $f : X \to Y$ is *continuous* (with respect to $\mathcal{T}_X, \mathcal{T}_Y$) if for all $V \in \mathcal{T}_Y, f^{-1}(V) \in \mathcal{T}_X$.

A function $q : X \to Y$ is a *quotient* map if it is surjective, and whenever $q^{-1}(U)$ is open, then U is open. If $(X, \mathcal{T})$ is a topological space, Y is a set, and $f : X \to Y$, then the topology *coinduced* by f on Y consists of all sets V such that $f^{-1}(V)$ is open in X. If f is surjective, then it is a quotient map with respect to the coinduced topology on Y.

A topological space X is *Kolmogorov* or T_0 if for every pair of distinct points $x_1 \neq x_2$, either there exists open U_1 such that $x_1 \in U_1$ and $x_2 \notin U_1$ or open U_2 such that $x_2 \in U_2$ and $x_1 \notin U_2$. A space is *Hausdorff* or T_2 if for every pair of distinct points $x_1 \neq x_2$, there exist open U_1, U_2 such that $x_1, \in U_1, x_2 \in U_2$, and $U_1 \cap U_2 = \emptyset$. A space is *discrete* if every singleton set $\{x\}$ is open.

A subset C of X is *compact* if every open cover has a finite subcover, and *countably compact* if every countable open cover has a finite subcover. A subset S of X is *dense* if for all open U, there exists $s \in S$ such that $s \in U$. A set is *separable* if it has a countable dense subset.

We say a space X is *locally compact* if for all $x \in X$ and open $U \ni x$, there exist open V and compact K such that $x \in V \subset K \subset U$. Note that it is not always the case that a compact space is locally compact. A weaker definition which is often used in the literature is obtained by omitting

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

the condition on U. For Hausdorff spaces, the two definitions are equivalent and all compact spaces are locally compact.

A subset S of a topological space (X, τ) is *sequentially open* if whenever $\vec{x}$ is a convergent sequence with limit $x_\infty \in S$, there exists $N \in \mathbb{N}$ such that $x_n \in S$ for all $n \geq N$. Clearly, any open set is sequentially open. A topological space (X, τ) is *sequential* if any sequentially open set is open. It is straightforward to show that any countably based space is a sequential space.

A set A is *sequentially closed* if whenever $\vec{x}$ is a convergent sequence for which every $x_n \in A$, then the limit $x_\infty \in A$, or equivalently, if its complement is sequentially open. A set C is *sequentially compact* if every sequence has a convergent subsequence. We shall show in Section 4.1 that if C is sequentially compact, then it is countably compact, and if X is a sequential space, then any countably compact set is sequentially compact.

A function $f : X \to Y$ is *sequentially continuous* if for any convergent sequence $x_n \to x_\infty$, we have $\lim_{n\to\infty} f(x_n) = f(x_\infty)$. Any continuous function is continuous, and if X is a sequential space, then any sequentially continuous function is continuous.

A *metric space* (X, d) consists of a set X and a function $d : X \times X \to \mathbb{R}^+$ satisfying *reflexivity* $d(x, y) = 0 \iff x = y$, *symmetry* $d(x, y) = d(y, x)$ and the *triangle inequality* $d(x, z) \leq d(x, y) + d(y, z)$ for all $x, y, z \in X$. The *open ball* about x of radius ϵ is the set $B(x, \epsilon) = \{y \in X \mid d(x, y) < \epsilon\}$. The topology $\mathcal{T}$ of (X, d) is generated by the base consisting of all open balls for $x \in X$ and $\epsilon \in (0, \infty)$.

A sequence $\vec{x}$ in a metric space converges to x_∞ if $\forall\epsilon > 0$, $\exists N$, $\forall n \geq N$, $d(x_n, x_\infty) < \epsilon$. The sequence $\vec{x}$ is a *Cauchy* sequence if $\forall\epsilon > 0$, $\exists N$, $\forall m, n \geq N$, $d(x_m, x_n) < \epsilon$. Any convergent sequence is Cauchy, and a metric space is *complete* if every Cauchy sequence converges.

If (X, d_X) and (Y, d_Y) are metric spaces, then a function $f : X \to Y$ is uniformly continuous over a set U if there exists a *modulus of continuity* $\delta : \mathbb{Q}^+ \to \mathbb{Q}^+$ such that for all $x_1, x_2 \in U$, $d_X(x_1, x_2) < \delta(\epsilon) \implies d_Y(f(x_1), f(x_2)) < \epsilon$.

3.2 Representations of topological spaces

We now consider representations of topological spaces. We first show that a representation of a set X induces a natural topology on X.

Definition 3.1 (Topology induced by a representation). *Let X be a set and δ be a representation of X. Then, the topology τ induced by δ is the* final topology *of δ, namely $U \in \tau \iff \delta^{-1}(U)$ is open in* dom (δ).

By definition, the representation δ becomes a continuous quotient map. If X is already equipped with a topology τ, then we will require the representation δ to be a continuous quotient map.

Definition 3.2 (Continuous quotient representation). *Let (X, τ) be a topological space, and δ be a representation of the set X. We say δ is a* continuous *representation if it is a continuous map, that is, whenever U is open, then $\delta^{-1}(U) = W \cap$* dom (δ) *for some open W. We say δ is a* quotient *representation if it is a quotient map, that is, whenever $\delta^{-1}(U) = W \cap$* dom (δ) *for some open W, then U is open.*

However, not all continuous quotient maps are "good" representations, as the following example shows.

Example 3.3. *Consider the binary representation of $\mathbb{R}$. More precisely,*

$$\delta(\pm a_n a_{n-1} \cdots a_0 \,.\, a_{-1} a_{-2} \cdots) = x \iff x = \pm \textstyle\sum_{k=-\infty}^{n} a_k 2^k,$$

where each $a_k \in \{0, 1\}$. It can be shown (see Section 3.4) that δ is a partial surjective quotient map $\Sigma^\omega \rightharpoonup \mathbb{R}$. Let $x_n = 1 + (-1/2)^n/3$ for $n \in \mathbb{N}$, and $x_\infty = 1$, so the function $f : \mathbb{N} \cup \{\infty\} \to \mathbb{R}$, $f(n) = x_n$ is continuous. Then, we have names $p_\infty = \texttt{+1.000}\cdots$ and $p'_\infty = \texttt{+0.111}\cdots$, whereas for x_{2n}, we have names $p_{2n} = +1.0^{2n}(01)^\omega$ and for x_{2n+1} we have a unique name $p_{2n+1} = \texttt{+0.1}^{2n+1}(\texttt{10})^\omega$.

Hence even though $x_n \to 1$ as $n \to \infty$, we cannot choose binary expansions for x_n which converge. In terms of representations, any function $h : \mathbb{N} \cup \{\infty\} \to \Sigma^\omega$ satisfying $\delta(h(n)) = f(n)$ (there are only two) has a discontinuity at ∞, so f does not continuously lift through the representation δ.

In order to prevent pathological situations as described above, we impose an *admissibility* condition on representation in addition to the quotient condition.

Definition 3.4 (Admissible representation). *A representation δ of a topological space X is* admissible *if it is continuous, and whenever $\phi : \Sigma^\omega \rightharpoonup X$ is a continuous partial function, there exists a continuous partial function $\eta : \Sigma^\omega \rightharpoonup \Sigma^\omega$ such that $\phi = \delta \circ \eta$.*

A representation δ of a set X is admissible *if δ is an admissible representation with respect to its final topology.*

The condition that a representation of a topological space be continuous captures the notion that δ must contain at least the information given by the topology of the space. The quotient and admissibility conditions mean that δ does not contain "too much" information; neither implies the other. Note that by the definition of admissibility, an admissible quotient representation is continuous.

The following result is derived from Schröder (2002b). It gives the basic properties implied by the quotient and admissibility conditions on continuous representations.

Proposition 3.5. *Let (X, τ_X) and (Y, τ_Y) be topological spaces, and δ_X, δ_Y be continuous representations of X and Y.*

(a) Suppose η is a continuous realizer for a function $f : X \to Y$ and δ_X is a quotient representation. Then, f is continuous.
(b) Suppose $f : X \to Y$ is continuous, and δ_Y is an admissible representation. Then, f has a continuous realizer.

Proof.

(a) Take $\phi = \delta_Y \circ \eta$, which is continuous. We have $f \circ \delta_X = \delta_Y \circ \eta|_{\mathrm{dom}\,(\delta_X)}$, so $f \circ \delta_X$ is continuous. Since δ_X is a quotient map, f is continuous.
(b) Take $\phi = f \circ \delta_X$. Then since δ_Y is admissible, there exists a continuous η such that $\delta_Y \circ \eta = f \circ \delta_X$. □

The following result is a direct consequence of Proposition 3.5.

Corollary 3.6 (Discontinuity implies uncomputability). *Let (X_i, τ_i) be topological spaces, and δ_i be admissible quotient representations of X_i for $i = 0, \ldots, k$. Then if $f : X_1 \times \cdots \times X_k \to X_0$ is $(\delta_1, \ldots, \delta_k; \delta_0)$-computable, then f is $(\tau_1, \ldots, \tau_k; \tau_0)$-continuous.*

In other words, only continuous functions can be computable, yielding a very strong link between topology and computability. The main use of this result is to show certain operations are *uncomputable*, since it is sufficient to show that the operation is discontinuous. The converse is not true, as there are continuous functions that are not computable, but in practice, most "naturally-defined" continuous functions are computable.

A common critique of computable analysis is that "only" continuous functions can be handled, whereas there are plenty of important functions that are discontinuous. Perhaps, the most important discontinuous function is the Heaviside function

$$H(t) = \begin{cases} 0 \text{ if } t < 0; \\ 1 \text{ if } t \geq 0. \end{cases}$$

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

By Corollary 3.6, H is an uncomputable function. This negative computability result should be interpreted as follows: it is impossible to evaluate H arbitrarily accurately at every point if only approximations to the argument are known. The computation fails at $t = 0$; for example, it is impossible to determine $H(3x - 1)$ given finitely many digits of the decimal expansion of $x = \frac{1}{3} = 0.33333\cdots$. However, there are weaker function-space topologies with admissible quotient representations in which H has a computable name which is sufficient to perform useful computations. For example, interpreting H in the space L^1 of integrable functions (see Section 3.8) gives a valid name from which we can compute $\int_a^b f(x)H(x)\,dx$ for any continuous f. This is sufficient for most practical applications, in which the Heaviside function occurs on the right-hand side of a differential equation, and only its integrals are involved.

Alternatively, by refining the topology on the argument space (corresponding to more information) or coarsening the topology on the result space (corresponding to less information) again allows for a valid name. In this way, the computability theory yields important information on the properties and valid uses of the function.

We now turn to the question of which topological spaces have an admissible quotient representation. An obvious condition is that the space should have at most continuum cardinality $2^{\aleph_0}$, the cardinality of Σ^ω. A second condition is that the points in the space should be distinguishable based on the open sets containing them, so should at least satisfy the Kolmogorov's T_0 separation axiom (see Section 3.1). Henceforth, all topological spaces considered will be Kolmogorov spaces with at most continuum cardinality unless explicitly declared otherwise.

One natural class of spaces which have at most continuum cardinality is the *countably based Kolmogorov spaces*, since every point is described by the basic open sets containing it. In Section 3.7, we give an explicit construction of an admissible quotient representation for a countably based Kolmogorov space. However, it turns out that any topological quotient of a countably based space also has a quotient representation which is admissible and that the class of *quotients of countably based* (qcb) Kolmogorov spaces are exactly the spaces with an admissible quotient representation. We prove this result in Section 4.2.

In general, a quotient of a countably based space need not be countably based, but we will show that it has a countable *pseudobase* in the following sense:

Definition 3.7 (Pseudobase). *A collection ρ of subsets of a topological space (X, τ) is a* pseudobase *if for any $x \in X$ and $U \in \tau$ with $U \ni x$, there exists $P \in \rho$ such that $x \in P$ and $P \subset U$.*

It is clear that if (X, τ) is a Kolmogorov space with a countable pseudobase, then X has at most continuum cardinality. Further, any pseudobase is a base if, and only if, it consists only of open sets.

The admissibility condition is strongly tied to the concept of sequential space. The space Σ^ω is an example of a sequential space. The property of being sequential is preserved by taking subspaces and quotient spaces, so any space with a quotient representation is itself a sequential space. Conversely, for a sequential space to have a quotient representation, we also need to bound the cardinality in some way. In Section 4.2, we show that any sequential space with a countable *sequential pseudobase* has an admissible quotient representation.

Definition 3.8 (Sequential quotient). *A representation $\delta : \Sigma^\omega \rightharpoonup X$ of a topological space X is a* sequential quotient *if whenever $x_n \to x_\infty$ is a convergent sequence in X, there exists a convergent sequence $\xi_n \to \xi_\infty$ in Σ^ω such that $\delta(\xi_n) = x_n$ for all $n \in \mathbb{N} \cup \{\infty\}$.*

Proposition 3.9. *If δ is an admissible quotient representation, then it is a sequential quotient.*

Proof. If δ is admissible, and $x_n \to x_\infty$ is a convergent sequence in X, then define $\hat{x} : \{0, 1\}^\omega \to X$ by $\hat{x}(0^n 1 \cdots) = x_n$ and $\hat{x}(0^\infty) = x_\infty$. Then, $\hat{x}$ is continuous, so there exists $\eta : \Sigma^\omega \to \Sigma^\omega$ such that

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

$\hat{x} = \delta \circ \eta$. Define $\xi_n = \eta(0^n 10^\infty)$ and $\xi_\infty = \eta(0^\infty)$. Then $\xi_n \in \{0, 1\}^\omega$ and $\delta(\xi_n) = \hat{x}(0^n 10^\infty) = x_n$, $\delta(\xi_\infty) = x_\infty$. Hence, admissibility implies sequential admissibility. □

However, being a sequential quotient is weaker than admissibility, as shown by Schröder (2002a, Example 2.3.9). The binary representation of Example 3.3 is a quotient but not admissible.

3.3 Computable type theory

We now develop a computable type theory based on the admissible quotient representations. The most important goals are to give a collection of initial concrete types and then to build up new types from existing types. The resulting types form a *Cartesian-closed category*, with objects which are computable types, and morphisms which are continuous functions (some of which are computable).

Definition 3.10 (Computable type). *A* computable type *is a pair* $(X, [\delta])$ *where* X *is a set and* $[\delta]$ *is an equivalence class of admissible quotient representations of* X.

We henceforth denote computable types by script $\mathscr{X}$ or calligraphic $\mathcal{X}$ letters, and sets by italic X or blackboard-bold $\mathbb{X}$ letters. Note that we always consider different sets as having different types, even if the sets are bijective and the representations respect this bijection.

Note that there are two natural categories whose objects are computable types. In the first, all *continuous* functions are allowed as morphisms, while in the second, only *computable* functions are allowed. In general, the morphisms involved in universal constructions in a category will all be computable, but the conditions apply to all continuous functions.

Definition 3.11 (Category of computable types). *The category of* computable types *(with continuous morphisms) is the category whose objects are computable types* $\mathcal{X} = (X, [\delta])$ *where* δ *is an admissible representation of* X*, and morphisms* $\mathcal{X} \to \mathcal{Y}$ *are functions* $f : X \to Y$ *which are continuous with respect to the induced topologies on* X *and* Y*. A morphism is* computable *if it corresponds to a computable function. The category of* computable types with computable morphisms *is the subcategory with the same objects, but whose morphisms are the computable functions.*

If $f : \mathcal{X}_1 \times \cdots \mathcal{X}_n \to \mathcal{Y}$ is a computable function, we say that $y = f(x_1, \ldots, x_n)$ is *computable from* $x_1, \ldots, x_n$ *as* an element of type $\mathcal{Y}$.

Recall that a category is *Cartesian closed* if it satisfies the following properties:

(a) There is a *terminal* object $\mathscr{I}$, such that for any object $\mathcal{W}$, there is a unique morphism $\mathcal{W} \to \mathscr{I}$.
(b) For any objects $\mathcal{X}_1, \mathcal{X}_2$, there is a *product* object $\mathcal{X}_1 \times \mathcal{X}_2$, with *projection* morphisms $p_i : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathcal{X}_i$ such that whenever $f_i : \mathcal{W} \to \mathcal{X}_i$, there exists a unique morphism $f : \mathcal{W} \to \mathcal{X}_1 \times \mathcal{X}_2$ such that $f_i = p_i \circ f$ for $i = 1, 2$.
(c) For any objects $\mathcal{X}, \mathcal{Y}$, there is an *exponential* object $\mathcal{Y}^{\mathcal{X}}$, with an *evaluation* morphism $e : \mathcal{Y}^{\mathcal{X}} \times \mathcal{X} \to \mathcal{Y}$, such that whenever $f : \mathcal{W} \times \mathcal{X} \to \mathcal{Y}$, there is a unique morphism $\hat{f} : \mathcal{W} \to \mathcal{Y}^{\mathcal{X}}$ such that $e \circ (\hat{f} \times \mathrm{id}_X) = f$. i.e. $e(\hat{f}(w), x) = f(w, x)$.

An *element* of an object $\mathcal{X}$ is defined to a morphism $\mathscr{I} \to \mathcal{X}$. This identification is useful in that it allows us to work purely within the category itself, but still recover elements of the underlying set of $\mathcal{X}$.

The following equivalences hold in any Cartesian-closed category.

Proposition 3.12.

(a) For any object $\mathcal{X}$, the products of the terminal type $\mathscr{I}$ and $\mathcal{X}$ are isomorphic to $\mathcal{X}$.
(b) The elements of $\mathcal{Y}^{\mathcal{X}}$ are in bijective correspondence with the morphisms $\mathcal{X} \to \mathcal{Y}$.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Proof.

(a) Take $\mathcal{W} = \mathcal{I}$. Since an element of any type is associated with a morphism $\mathcal{I} \to \mathcal{X}$, the elements of $\mathcal{I} \times \mathcal{X}$ are associated with pairs (i, x) where i is the unique element of $\mathcal{I}$ and $x \in \mathcal{X}$.

(b) $f : \mathcal{X} \to \mathcal{Y}$, we can construct $\hat{f} : \mathcal{I} \to \mathcal{Y}^{\mathcal{X}}$ as the unique morphism satisfying $e \circ (\hat{f} \times \mathrm{id}_{\mathcal{X}}) = f \circ p$ where $p : \mathcal{I} \times \mathcal{X} \to \mathcal{X}$, and conversely. □

The next theorem gives some closure properties of the category of computable types. These properties imply that the category of computable types is Cartesian-closed. This is an important notion in intuitionistic/constructive type theory, since the lambda-calculus can be developed in any Cartesian-closed category.

Theorem 3.13. *The category of computable types is Cartesian closed. Further:*

(a) The morphisms $\mathcal{W} \to \mathcal{I}$ are computable.

(b) The projections $p_i : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathcal{X}_i$ are computable, and the function $f : \mathcal{W} \to \mathcal{X}_1 \times \mathcal{X}_2$ satisfying $f_i = p_i \circ f$ for $i = 1, 2$ is computable from f_1, f_2.

(c) The evaluation *function $e : \mathcal{Y}^{\mathcal{X}} \times \mathcal{X} \to \mathcal{Y}$ is computable, and the function $f, \hat{f}$ are each computable from the other.*

Proof.

(a) Let $\mathcal{I}$ be a one-point-set, with representation δ with $\mathrm{dom}\,(\delta) = \Sigma^\omega$. (Actually, any representation whose domain contains a machine-computable element of Σ^ω will do.)

(b) Let δ_i be a representation of X_i, $i = 1, 2$. We say that q is a name of $(x_1, x_2) \in X_1 \times X_2$ if $\pi_{2,i}(q) \in \mathrm{dom}\,(\delta_i)$ with $\delta_i(\pi_{2,i}(q)) = x_i$ for $i = 1, 2$. We have $p_i(x_1, x_2) = p_i(\delta(q)) = \delta_i(\pi_{2,i}(q))$ for $i = 1, 2$ so the projections are computable.

Suppose $\mathcal{P}$ and $\mathcal{P}'$ are two types satisfying the properties. Then taking $f_i = p_i'$, there exists computable $f : \mathcal{P}' \to \mathcal{P}$ such that $x_i = p_i'(x_1, x_2) = p_i(f(x_1, x_2))$ for all x_1, x_2, so id is computable. The same holds reversing the roles of $\mathcal{P}$ and $\mathcal{P}'$, so any two product types are equivalent.

(c) Let δ_X, δ_Y be admissible quotient representations of X and Y. We aim to define a representation $\delta_{X\to Y}$ on $\mathcal{C}(X; Y)$. Let $f : X \to Y$ be continuous. Then, there exists continuous $\eta : \Sigma^\omega \rightharpoonup \Sigma^\omega$ with G_δ-domain such that $f(\delta_X(q)) = \delta_Y(\eta(q))$ for all $q \in \mathrm{dom}\,(\delta_X)$. By Theorem 2.16, there exists $a \in \Sigma^\omega$ such that $\eta(\,\cdot\,) = \varepsilon(a, \cdot)$. We therefore take $\delta_{X\to Y}(a) = f$ if, and only if, $f(\delta_X(q)) = \delta_Y(\varepsilon(a, q))$ for all $q \in \mathrm{dom}\,(\delta_X)$.

The evaluation function e is computable, since if $f = \delta_{X\to Y}(a)$ and $x = \delta_X(q)$, we have $e(f, x) = f(x) = f(\delta_X(q)) = \delta_Y(\varepsilon(a, q))$. Further, if $f : \mathcal{W} \times \mathcal{X} \to \mathcal{Y}$ is computable, then there exists computable $\xi : \Sigma^\omega \times \Sigma^\omega \rightharpoonup \Sigma^\omega$ such that $f(\delta_W(p), \delta_X(q)) = \delta_Y(\xi(p, q))$. Take $\eta : \Sigma^\omega \rightharpoonup \Sigma^\omega$ such that $\eta(\tau_2(p, q)) = \xi(p, q)$, noting that η can be taken to be computable. Then, there exists computable a such that $\varepsilon(a, p) = \eta(p)$. Then for any $x \in X$, $\delta_{X\to Y}(\sigma(a, p))(x) = \delta_{X\to Y}(\sigma(a, p))(\delta_X(q)) = \delta_Y(\varepsilon(\sigma(a, p), q)) = \delta_Y(\varepsilon(a, \tau_2(p, q))) = \delta_Y(\eta(\tau_2(p, q))) = \delta_Y(\xi(p, q)) = f(\delta_W(p), \delta_X(q)) = f(w, x) = \hat{f}(w)(x)$, so $\delta_{X\to Y}(\sigma(a, p)) = \hat{f}(w)$. Hence, $\hat{f}$ is computable.

We henceforth identify $w \in \mathcal{Y}^{\mathcal{X}}$ with the (necessarily continuous) function $f(x) = e(w, x)$.

To show that the representation is unique up to equivalence, suppose $\mathcal{C}$ and $\mathcal{C}'$ are types of the continuous functions satisfying the required properties, with respective evaluation functions e and e'. Since evaluation $e : \mathcal{C} \times \mathcal{X} \to \mathcal{Y}$ is computable, the function $i : \mathcal{C} \to \mathcal{C}'$ satisfying $e'(i(w), x) = e(f, x)$ is computable. Then $e'(i(f), x) = f(x)$, so $i(f)$ is equal to f, that is, the identity $\mathcal{C}$ and $\mathcal{C}'$ are computable. Similarly, the identity i' from $\mathcal{C}'$ to $\mathcal{C}$ is computable. □

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Viewed in terms of computable types, any singleton space has a representation yielding the type of the terminal object. (Indeed, any representation whose domain contains a computable element of Σ^ω will do.) There is a natural bijection between elements x of a space X and continuous functions from the singleton space $\mathcal{I}$ to X. An element of $\mathcal{X}$ is computable if the corresponding function $\mathcal{I} \to \mathcal{X}$ is computable. It is easy to see that this is equivalent to x having a computable name.

Remark 3.14. The topology on $X_1 \times X_2$ induced by the representation $\delta_{X_1 \times X_2}$ is *not* necessarily the product topology. For the topology of $\mathcal{X}_1 \times \mathcal{X}_2$ must be sequential, but the product topology need not to be. In fact, the topology on $\mathcal{X}_1 \times \mathcal{X}_2$ is the *sequentialization* of the product topology. An elementary example is the product $\mathbb{Q} \times (\mathbb{Q}/\mathbb{Z})$ (Franklin 1965).

We can additionally define countable products in the category of computable types.

Definition 3.15. *Let $(\mathcal{X}_n)_{n\in\mathbb{N}}$ be computable types. Then, $\prod_{n=0}^{\infty} \mathcal{X}_n$ is a computable type with representation δ given by taking*

$$\delta(\tau(p_0, p_1, \ldots)) = (x_0, x_1, \ldots) \iff \delta_i(p_i) = x_i \tag{5}$$

where τ is the infinite tupling function given by (3).

There are similar initial constructions in the category of computable types. The initial type $\mathcal{E}$ is the type of the empty set, and the sum type $\mathcal{X}_1 + \mathcal{X}_2$ is the type of the disjoint union.

Theorem 3.16.

(a) There is a computable initial *type $\mathcal{E}$ which is unique up to computable isomorphism such that for any computable type $\mathcal{X}$, there is a unique function $\mathcal{E} \to \mathcal{X}$, and this function is computable.*
(b) If $\mathcal{X}_1$ and $\mathcal{X}_2$ are computable types, then there is a unique computable sum *type $\mathcal{X}_1 + \mathcal{X}_2$ together with computable embeddings $j_i : \mathcal{X}_i \to \mathcal{X}_1 + \mathcal{X}_2$ such that for all continuous/computable functions $f_i : \mathcal{X}_i \to \mathcal{W}$, there exists a continuous/computable function $f : \mathcal{X}_1 + \mathcal{X}_2 \to \mathcal{W}$ such that $f_i = f \circ j_i$ for $i = 1, 2$.*

The proof is similar to that of Theorem 3.13, and it was omitted.

3.4 Fundamental types

We now describe three fundamental logical and numerical types, and some derived product types, which form the cornerstone of computable analysis. Where possible, we give representations where the domain has a particularly nice form.

3.4.1 The natural numbers

We begin with the natural numbers $\mathbb{N}$ and define the corresponding type $\mathcal{N}$. A simple representation using the alphabet $\{0, 1, \lrcorner\}$ is given by the binary expansion using the '$\lrcorner$' symbol to act as a terminator:

$$\delta(p_k p_{k-1} \cdots p_1 p_0 \lrcorner \cdots) = \sum_{i=0}^{k} 2^i p_i.$$

The decimal representation uses the alphabet $\{0, 1, 2, \ldots, 9, \lrcorner\}$ and is defined by

$$\delta(p_k p_{k-1} \cdots p_1 p_0 \lrcorner \cdots) = \sum_{i=0}^{k} 10^i p_i.$$

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

In both cases, any symbol may appear after the terminator, though equivalent representations are obtained by, for example, requiring all symbols after the '↵' to be fixed (e.g. '↵').

A simple representation of $\mathbb{N}$ with the alphabet $\Sigma = \{0, 1\}$ is given by a function δ with domain $\{0, 1\}^{\omega} \setminus \{1^{\omega}\}$ by

$$\delta(1^n 0 \cdots) = n.$$

Alternatively, we can restrict dom (δ) to $\{1^n 0^{\omega} \mid n \in \mathbb{N}\}$. The above representation is very inefficient, since the number n requires $n + 1$ bits to determine. In order to encode the binary representation using only symbols $\{0, 1\}$, we need to know how to specify the end of the input. A simple way of doing this is to prefix the binary name with a unary name giving the number of binary digits; or better, the number of "machine words":

$$\delta(1^m 0 p_{km-1} p_{km-2} \cdots p_1 p_0 \cdots) = \textstyle\sum_{i=0}^{km-1} 2^i p_i.$$

or the logarithm of the number of digits:

$$\delta(1^m 0 p_{2^m-1} p_{2^m-2} \cdots p_1 p_0 \cdots) = \textstyle\sum_{i=0}^{2^m-1} 2^i p_i.$$

For example, the number 425 has a binary expansion of 110101001 with 9 digits, so the names of 425 with $m = 4$ are $11110\,0000000110101001 \cdots$, where the elements indicated by $\cdots$ are arbitrary.

Under any of the above representations, it is easy to show that addition and multiplication are computable and that comparisons are decidable predicates.

Note that there is no continuous representation of the natural numbers with domain Σ^{ω}, since Σ^{ω} is compact but $\mathbb{N}$ is not.

The integers can be constructed by introducing a symbol $-$, or as the quotient of $\mathcal{N} \times \mathcal{N}$ under the function $z(m, n) = m - n$. The rationals can be constructed by introducing a symbol $/$, or as the quotient of $\mathcal{Z} \times (\mathcal{N} \setminus \{0\})$ under the function $q(m, n) = m/n$.

The space $\mathbb{N}^{\infty} = \mathbb{N} \cup \{\infty\}$ is particularly important, since any convergent sequence in a topological space X can be viewed as a continuous function $s : \mathbb{N}^{\infty} \to X$. A representation of $\mathbb{N}^{\infty}$ as a total function is $\delta(1^n 0 \cdots) = n$; $\delta(1^{\omega}) = \infty$.

3.4.2 Logical types

We shall later see that almost all problems in computable analysis are undecidable. In particular, equality on Σ^{ω} is undecidable, since we can never tell in finite time whether two sequences are equal. For this reason, it is most natural to use a three-valued logical type $\mathcal{K}$ with values $\{\top, \bot, \uparrow\}$ denoting, *true*, *false*, and *indeterminate*. These values represent the results of predicates which are, respectively, *verifiable* (*provable*), *falsifiable* (*disprovable*), and *undecidable*.

The type $\mathcal{K}$ can be given a representation δ which is a total function on $\{0, 1\}^{\omega}$. We define

$$\delta(0^* 10 \cdots) = \bot; \quad \delta(0^* 11 \cdots) = \top; \quad \delta(0^{\omega}) = \uparrow,$$

where any symbol may appear in the trailing $\cdots$. We interpret the representation as follows. A leading 0 indicates that at the given stage of computation, there is insufficient information to determine whether the value should be true or false. The first 1 indicates that a decision has been made, and the next digit is 0 for false and 1 for true. If the result is undecidable, then the output is an infinite sequence of zeroes.

The induced topology has basic open sets $\{\top\}$ and $\{\bot\}$, so the open sets are $\big\{\{\,\}, \{\top\}, \{\bot\}, \{\top, \bot\}, \{\top, \bot, \uparrow\}\big\}$. In particular, the set $\{\uparrow\}$ is closed but not open, so although the topology is finite, it is not the discrete topology. Indeed, it is not even Hausdorff, since the only open set containing $\uparrow$ is $\{\top, \bot, \uparrow\}$.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Subtypes of $\mathcal{K}$ include the Boolean type $\mathcal{B} = \{\top, \bot\}$ and the Sierpinskian type $\mathcal{S} = \{\top, \uparrow\}$. The Sierpinskian type is closed under finite conjunctions $\wedge : \mathcal{S} \times \mathcal{S} \to \mathcal{S}$ and *countable* disjunctions $\bigvee : \mathcal{S}^\omega \to \mathbb{S}$, with realizer $(\Sigma^\omega)^\omega \to \Sigma$ given by $\eta_\vee(p_0, p_1, \ldots)_i = 1 \iff \exists j \leq i, (p_j)_i = 1$.

The following theorem gives a generally applicable result on quantifiers:

Proposition 3.17.

(a) Given $\pi : \mathcal{N} \to \mathcal{S}$, then the existential quantifier $\exists n,\ \pi(n)$ is computable.
(b) Given $\pi : \mathcal{C} \to \mathcal{S}$, then the universal quantifier $\forall p,\ \pi(p)$ is computable.

Sketch of proof.

(a) Simulate parallel computation of every $\pi(n)$ on a single machine, which returns true in finite time if any one is true.
(b) By Brouwer's *fan theorem* (see e.g. Troelstra and van Dalen 1988), if $\pi(p)$ is true for all $p : \mathcal{C}$, then there is a uniform bound m on the number of bits of p which need to be read in order to prove $\pi(p)$, so the returns true in finite time. □

3.4.3 The real numbers

Perhaps the most important computable type, and our first concrete example of an uncountable type, is the type of the real numbers. Similar considerations hold for the decimal representation.

A class of representations which *are* admissible are known as *extended digit representations*. The simplest is the *binary signed-digit representation* with alphabet $\Sigma = \{0, 1, \bar{1}, .\}$ The representation is defined by

$$\delta(a_n a_{n-1} \cdots a_0 \,.\, a_{-1} a_{-2} \cdots) = \sum_{i=-\infty}^{n} a_i\, 2^i, \tag{6}$$

where the symbol '$\bar{1}$' is read as -1.

It is easy to show that arithmetic is computable with respect to the binary signed-digit representation and that strict inequality $<$ is semidecidable. We also need to account for the topological/metric structure of the real line. The most straightforward way of doing this is by looking at limits of convergent sequences. Since a finite part of a general convergent sequence gives no information about the limit, we need to restrict to *effectively convergent sequences* for which the rate of convergence is known.

Definition 3.18 (Effective limit). *A limit in a metric space is* effective *if there exists a computable sequence of rationals ϵ_n such that $\epsilon_n \searrow 0$ and $d(x_m, x_n) < \epsilon_{\min(m,n)}$ for all m, n.*

Without loss of generality, by proceeding to an appropriate subsequence, we can always take $\epsilon_n = 2^{-n}$ or $\epsilon_n = 1/n$.

The following theorem is due to Hertling (1999, Theorem 3.5). It asserts the existence of a canonical real number type.

Theorem 3.19 (Real number type). *There is a unique computable type $\mathcal{R}$ with underlying set $\mathbb{R}$ such that the constant 1 is a computable number, arithmetical operations $+, -, \times, \div$ are computable, strict comparison $<$ is semidecidable, and effective limits* lim *are computable.*

Sketch of proof. Suppose δ_1 and δ_2 are representations of $\mathbb{R}$ satisfying the required properties. Given a δ_1-name p_1 of $x \in \mathbb{R}$, we need to compute a δ_2-name of p.

Since the constant one and addition are computable, given a positive integer n, we can find a δ_1- or δ_2-name of n by computing $1 + 1 + \cdots + 1$. Since subtraction is computable, we can find a name of any integer m as $m = n_1 - n_2$. Since division is computable, we can find a name of any rational number q as $q = m/n$.

Given a δ_1-name p of $x \in \mathbb{R}$, since we can construct any rational number q and verify $q < x$ and $x < q$, we enumerate all rational numbers $l < x$ and all rational numbers $u > x$. We can therefore construct a sequence of rational numbers q_i such that $|q_i - q_j| < 2^{-\min(i,j)}$ and $\lim_{i\to\infty} q_i = x$. Since the q_i are rational, we can find a δ_2-name of each q_i. We can find a δ_2-name of x since effective limits are δ_2-computable. $\square$

3.5 Set and function types

We now define types for the main classes of sets and functions arising in topology. The constructions in this section are general; in Section 3.7 we will consider countably based spaces, following Weihrauch (2000, Chapter 5), Weihrauch and Grubba (2009). Many of the results in this section can be found in Pauly (2016). Extensions to general descriptive set theory (of Borel sets) can be found in Pauly and de Brecht (2015).

We have already seen in Theorem 3.13(c) that the exponential type is suitable as a type of continuous functions:

Definition 3.20. *Let $\mathcal{X}$, $\mathcal{Y}$ be computable types. The type of continuous functions $\mathcal{C}(\mathcal{X};\mathcal{Y})$ is the exponential type $\mathcal{Y}^{\mathcal{X}}$.*

We now develop the theory of subsets of a topological type. We begin by introducing some convenient notation.

Notation 3.21. *Write $A \between U$ to denote the classical predicate $U \cap A \neq \emptyset$. If U is open, we say that* A overlaps U. *Write $U_n \nearrow U_\infty$ if $U_{n+1} \supset U_n$ for all n and $\bigcup_{n=0}^{\infty} U_n = U_\infty$.*

From classical topology, we have the property that a set U is open if, and only if, its characteristic function is a continuous map from X to $\mathbb{S}$:

Proposition 3.22 (Open sets). *Let X be a topological space. Then there is a canonical bijection between $\mathcal{O}(X)$ and continuous functions $X \to \mathbb{S}$ given by $\chi(x) = \top \iff x \in U$ for $U \in \mathcal{O}(X)$ and $\chi : X \to \mathbb{S}$.*

We use this as a basis for the definitions of types of subsets of a computable type.

Definition 3.23 (Open and closed set types).

(a) *The type of open subsets of $\mathcal{X}$, denoted $\mathcal{O}(\mathcal{X})$, is defined to be the exponential object $\mathcal{S}^{\mathcal{X}}$. The interpretation of continuous $\chi : X \to \mathbb{S}$ as a point-set $U \in \mathcal{O}(X)$ is given by $U = \chi^{-1}(\{\top\})$.*
(b) *The type of closed subsets of $\mathcal{X}$, denoted $\mathcal{A}(\mathcal{X})$, is identified with $\mathcal{S}^{\mathcal{X}}$. The interpretation of continuous $\overline{\chi} : X \to \mathbb{S}$ as a point-set $A \in \mathcal{A}(X)$ is given by $A = \overline{\chi}^{-1}(\{\uparrow\})$.*

We now turn to types of *compact* sets, and the dual type of *separable* or *overt* sets, which are defined in terms of subset and overlap relations. Note that the overlap and subset relations can be defined in terms of existential and universal quantifiers as

$$S \between U \iff \exists x \in S,\ x \in U$$

$$S \subset U \iff \forall x \in S,\ x \in U.$$

Further, for any set S, the overlap relation and subset relations satisfy:

$$S \between (U_1 \cup U_2) \iff (S \between U_1) \vee (S \between U_2); \tag{7}$$

$$S \subset (U_1 \cap U_2) \iff (S \subset U_1) \wedge (S \subset U_2). \tag{8}$$

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Given a fixed set S, these relations define functions $\mathcal{O}(X) \to \mathbb{S}$, or equivalently, a set of open sets $\mathcal{O}(X)$. It can be shown that the functions defined by $\between$ are always continuous, but the functions defined by $\subset$ are continuous if, and only if, S is compact The observations above motivate the definition of set types as functions defined by the overlap and subset relations.

Definition 3.24 (Overt and compact set types).

(a) The type of overt[2] *subsets of $\mathcal{X}$, denoted $\mathcal{V}(\mathcal{X})$, is defined to be the subtype of $\mathcal{S}^{\mathcal{O}(\mathcal{X})}$ defined by functions $b : \mathcal{O}(\mathcal{X}) \to \mathcal{S}$ satisfying $b(U \cup V) = b(U) \vee b(V)$ and $b(\emptyset) = \uparrow$. Given any set B, we can define such a function b by $b(U) = \top \iff B \between U$. The interpretation of the function b as a point-set B is given by $B = \{x \in X \mid \forall\, U \in \mathcal{O}(X),\ (x \in U \implies b(U) = \top)\}$.*

(b) The type of compact *subsets of $\mathcal{X}$, denoted $\mathcal{K}(\mathcal{X})$, is defined to be the subtype of $\mathcal{S}^{\mathcal{O}(\mathcal{X})}$ defined by functions $c : \mathcal{O}(\mathcal{X}) \to \mathcal{S}$ satisfying $c(U \cap V) = c(U) \wedge c(V)$ and $c(X) = \top$. Given any compact set C, we can define such a function c by $c(U) = \top \iff C \subset U$. The interpretation of the function c as a point-set C is given by $C = \{x \in X \mid \forall\, U \in \mathcal{O}(X),\ (c(U) = \top \implies x \in U)\}$.*

Since $S \between U \iff \mathrm{cl}(S) \between U$, the function $b : \mathcal{O}(\mathcal{X}) \to \mathcal{S}$ defines a set only up to its closure; the point-set construction of B always yields a closed set. If X is a Hausdorff space, then the function $c : \mathcal{O}(\mathcal{X}) \to \mathcal{S}$ defines a compact set uniquely. However, if X is not Hausdorff, then the point-set is defined only up to its *saturation*:

Definition 3.25 (Saturation). *Let (X, τ) be a topological space and $S \subset X$. Then the* saturation *of S in (X, τ), denoted* sat (S), *is*

$$\mathrm{sat}\,(S) = \bigcap \{U \in \tau \mid S \subset U\}. \tag{9}$$

We say S is saturated *if $S = \mathrm{sat}\,(S)$.*

For example, if $X = \mathbb{R}$ with topology $\tau_< = \{(-\infty, a) \mid a \in \mathbb{R}\}$, then any set S with $\sup\,(S) \in S$ is compact, and the saturated compact sets have the form $(-\infty, s]$ for $s \in \mathbb{R}$.

Remark 3.26. The interpretation of $c : \mathcal{O}(\mathcal{X}) \to \mathcal{S}$ as a point-set may fail to be compact. For if $X = (\mathbb{Q}, \tau_<)$ where $\tau_< = \{(-\infty, a) \cap \mathbb{Q} \mid a \in \mathbb{R}\}$ and $r \notin \mathbb{Q}$, then the function $c : (-\infty, a) \mapsto \top \iff a > r$ is continuous and satisfies (8), but the corresponding point-set C is $(-\infty, r] \cap \mathbb{Q} = (-\infty, r) \cap \mathbb{Q}$ which is not compact. Additionally, the identity $c(U) = \top \iff C \subset U$ fails for the set $U = (-\infty, r)$. In order that the interpretation of c as a point-set yields a compact set C, we require the space X to be *sober*. We shall return to this point in Section 4.4.

We now give some computability results which are valid for any topological type.

Theorem 3.27. *The following operators are computable:*

(a) Complement $\mathcal{O} \leftrightarrow \mathcal{A}$.
(b) Finite intersection $\mathcal{O} \times \mathcal{O} \to \mathcal{O}$.
(c) Countable union $\mathcal{O}^{\mathcal{N}} \to \mathcal{O}$.
(d) Finite union $\mathcal{A} \times \mathcal{A} \to \mathcal{A}$.
(e) Countable intersection $\mathcal{A}^{\mathcal{N}} \to \mathcal{A}$.
(f) (Closed) intersection $\mathcal{V} \times \mathcal{O} \to \mathcal{V}$.
(g) (Saturated) intersection $\mathcal{K} \times \mathcal{A} \to \mathcal{K}$.
(h) (Closed) countable union $\mathcal{V}^{\mathcal{N}} \to \mathcal{V}$.
(i) Finite union $\mathcal{K} \times \mathcal{K} \to \mathcal{K}$.
(j) Singleton $\mathcal{X} \to \mathcal{V}$ and $\mathcal{X} \to \mathcal{K}$.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Proof. The basic proof technique in all cases is the same. Since our set types are special function types, we need to give method for evaluating the function on its arguments in terms of operations which are already known to be computable. For overt/compact sets, we use the general principle that in order to prove that a function $\hat{f} : \mathcal{W} \to \mathcal{Y}^{\mathcal{X}}$ is computable, it suffices to show that the function $f : \mathcal{W} \times \mathcal{X} \to \mathcal{Y}$ satisfying $f(w, x) = \hat{f}(w)(x)$ is computable.

(a) $x \in U \iff x \notin X \setminus U; x \notin A \iff x \in X \setminus A$.
(b) $x \in U_1 \cap U_2 \iff (x \in U_1) \wedge (x \in U_2)$.
(c) $x \in \bigcup_{n=1}^{\infty} U_n \iff \bigvee_{n=1}^{\infty} (x \in U_n)$.
(d) $x \notin A_1 \cup A_2 \iff (x \notin A_1) \wedge (x \notin A_2)$.
(e) $x \notin \bigcap_{n=1}^{\infty} A_n \iff \bigwedge_{n=1}^{\infty} (x \notin A_n)$.
(f) $\mathrm{cl}(B \cap U) \between V \iff B \cap U \between V \iff B \between U \cap V$.
(g) $\mathrm{sat}\,(C \cap A) \subset U \iff C \cap A \subset U \iff C \subset U \cup (X \setminus A)$.
(h) $\mathrm{cl}(\bigcup_{n=1}^{\infty} B_n) \between U \iff \bigcup_{n=1}^{\infty} B_n \between U \iff \bigvee_{n=1}^{\infty} (B \between U_n)$.
(i) $C_1 \cup C_2 \subset U \iff (C_1 \subset U) \wedge (C_2 \subset U)$.
(j) $x \in U \iff \{x\} \between U \iff \{x\} \subset U$.

In more detail, to prove (b), we need to show that we can compute $U_1 \cap U_2$ given (names of) U_1 and U_2. Since computing an open set means verifying its elements, we need to verify $x \in U_1 \cap U_2$ whenever this is true. Since $x \in U_1 \cap U_2 \iff x \in U_1 \wedge x \in U_2$, it suffices to compute this formula. Given U_1 and U_2, we can compute $p_1 = x \in U_1$ and $p_2 = x \in U_2$ in $\mathcal{S}$ by definition of the open set type. Then, we can compute $p_1 \wedge p_2$ since conjunction is computable on $\mathcal{S}$.

Similarly, in order to prove (f), we need to compute $\mathrm{cl}\, B \cap U$ given (names of) overt B and open U, which means verifying intersections of $\mathrm{cl}\, B \cap U$ with arbitrary open sets V. We show $\mathrm{cl}(B \cap U) \between V \iff B \between U \cap V$, which suffices since $U \cap V$ is computable from U, V, and nonempty intersection of B with $U \cap V$ is verifiable by the definition of the overt set type. □

The set types are all closed under finite products:

Theorem 3.28. *Let $\mathcal{X}_1, \mathcal{X}_2$ be computable types. Then the Cartesian product is computable:*

(a) $\mathcal{O}(\mathcal{X}_1) \times \mathcal{O}(\mathcal{X}_2) \to \mathcal{O}(\mathcal{X}_1 \times \mathcal{X}_2)$;
(b) $\mathcal{A}(\mathcal{X}_1) \times \mathcal{A}(\mathcal{X}_2) \to \mathcal{A}(\mathcal{X}_1 \times \mathcal{X}_2)$.
(c) $\mathcal{V}(\mathcal{X}_1) \times \mathcal{V}(\mathcal{X}_2) \to \mathcal{V}(\mathcal{X}_1 \times \mathcal{X}_2)$.
(d) $\mathcal{K}(\mathcal{X}_1) \times \mathcal{K}(\mathcal{X}_2) \to \mathcal{K}(\mathcal{X}_1 \times \mathcal{X}_2)$.

Proof.

(a) $(x_1, x_2) \in U_1 \times U_2 \iff x_1 \in U_1 \wedge x_2 \in U_2$.
(b) $(x_1, x_2) \notin A_1 \times A_2 \iff x_1 \notin A_1 \vee x_2 \notin A_2$.
(c) $V_1 \times V_2 \between U \iff \exists x_1 \in V_1, (V_2 \between \{x_2 \mid (x_1, x_2) \in U\})$.
(d) $C_1 \times C_2 \subset U \iff \forall x_1 \in C_1, (C_2 \subset \{x_2 \mid (x_1, x_2) \in U\})$. □

We would also like to prove that countable products of the set types are computable. For compact sets, this is an effective version of the Tychonoff theorem. While it is true that countable products of overt and compact sets depend continuously on their arguments, we were unable to prove computability.

Conjecture 3.29. *Let $\mathcal{X}_n$, $n \in \mathbb{N}$ be computable types.*

(a) Countable product $(S_0, S_1, S_2, \ldots) \to \prod_{n=0}^{\infty} S_n$ *of nonempty sets is computable* $\prod_{n=0}^{\infty} \mathcal{V}(\mathcal{X}_n) \rightharpoonup \mathcal{V}(\prod_{n=0}^{\infty} \mathcal{X}_n)$.
(b) Countable product is computable $\prod_{n=0}^{\infty} \mathcal{K}(\mathcal{X}_n) \to \mathcal{K}(\prod_{n=0}^{\infty} \mathcal{X}_n)$.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

We shall see in Theorem 3.47 that this holds under a stronger notion of overtness and compactness. We shall also see in Theorem 3.66 that countable products of overt and compact sets are computable in countably based spaces.

We have seen that the exponential $\mathcal{Y}^{\mathcal{X}}$ (alternatively denoted $\mathcal{C}(\mathcal{X};\mathcal{Y})$) is a canonical type for the continuous functions $X \to Y$ and that evaluation $e : \mathcal{C}(\mathcal{X};\mathcal{Y}) \times \mathcal{X} \to \mathcal{Y}$ is computable. We can extend computability to preimages and images of sets.

Theorem 3.30 (Computable preimage/image). *The following operators are computable:*

(a) Preimage $(f, S) \mapsto f^{-1}(S)$ is computable $\mathcal{C}(\mathcal{X};\mathcal{Y}) \times \mathcal{O}(\mathcal{Y}) \to \mathcal{O}(\mathcal{X})$.
(b) Preimage is computable $\mathcal{C}(\mathcal{X};\mathcal{Y}) \times \mathcal{A}(\mathcal{Y}) \to \mathcal{A}(\mathcal{X})$.
(c) Closed-image $(f, S) \mapsto \mathrm{cl}(f(S))$ is computable $\mathcal{C}(\mathcal{X};\mathcal{Y}) \times \mathcal{V}(\mathcal{X}) \to \mathcal{V}(\mathcal{Y})$.
(d) Saturated-image $(f, S) \mapsto \mathrm{sat}\,(f(S))$ is computable $\mathcal{C}(\mathcal{X};\mathcal{Y}) \times \mathcal{K}(\mathcal{X}) \to \mathcal{K}(\mathcal{Y})$.

Proof.

(a) $x \in f^{-1}(U) \iff f(x) \in U$.
(b) $x \notin f^{-1}(A) \iff f(x) \notin A$.
(c) $\mathrm{cl}(f(A)) \between V \iff f(A) \between V \iff A \between f^{-1}(V)$.
(d) $\mathrm{sat}\,(f(C)) \between V \iff f(C) \subset V \iff C \subset f^{-1}(V)$. □

Remark 3.31. As we have seen, when computing natural operators (such as intersection) yielding an overt or compact set, it may be that the result of an operation is not, respectively, closed or saturated. In these cases, we henceforth implicitly take the closed or saturated version, which is the canonical representative of the type. This does not affect further operations, for example, $\mathrm{cl}(f(B)) = f(\mathrm{cl}(B))$ for continuous f.

We now turn to multivalued functions $F : X \rightrightarrows Y$, which can be thought of as set-valued functions $F : X \to \mathcal{P}(Y)$. A multivalued function is open- (respectively closed- or compact-) valued if $F(x)$ is open (respectively closed or compact) for all x. The *image* of a set S is defined by $F(S) = \bigcup\{F(x) \mid x \in S\} = \{y \in Y \mid \exists x \in S,\ y \in F(x)\}$. The *composition* of $F : X \rightrightarrows Y$ and $G : Y \rightrightarrows Z$ is defined by $G \circ F(x) = G(F(x)) = \{z \in Z \mid \exists y \in Y,\ \ y \in F(x) \wedge z \in F(y)\}$. The *weak preimage* F^{-1} of $F : X \rightrightarrows Y$ is defined by $F^{-1}(B) = \{x \in X \mid F(x) \cap B \neq \emptyset\}$ and is a multivalued function $F^{-1} : Y \rightrightarrows X$. The *strong preimage* $F^{\Leftarrow}$ of F is defined by $F^{\Leftarrow}(B) = \{x \in X \mid F(x) \subset B\}$. A multivalued function $F : X \rightrightarrows Y$ is *upper-semicontinuous* if $F^{-1}(A)$ is closed whenever A is closed; equivalently if $F^{\Leftarrow}(U)$ is open whenever U is open. F is *lower-semicontinuous* if $F^{-1}(U)$ is open whenever U is open; equivalently if $F^{\Leftarrow}(A)$ is closed whenever A is closed. The *graph* of a multivalued map $F : X \rightrightarrows Y$ is the set graph $F = \{(x, y) \in X \times Y \mid y \in F(x)\}$.

Theorem 3.32 (Action of multivalued functions).

(a) Multivalued preimage $(F, S) \mapsto F^{-1}(S)$ is computable $\mathcal{C}(\mathcal{X};\mathcal{V}(\mathcal{Y})) \times \mathcal{O}(\mathcal{Y}) \to \mathcal{O}(\mathcal{X})$.
(b) Multivalued preimage is computable $\mathcal{C}(\mathcal{X};\mathcal{K}(\mathcal{Y})) \times \mathcal{A}(\mathcal{Y}) \to \mathcal{A}(\mathcal{X})$.
(c) Multivalued image $(F, S) \mapsto F(S)$ is computable $\mathcal{C}(\mathcal{X};\mathcal{V}(\mathcal{Y})) \times \mathcal{V}(\mathcal{X}) \to \mathcal{V}(\mathcal{Y})$.
(d) Multivalued image is computable $\mathcal{C}(\mathcal{X};\mathcal{K}(\mathcal{Y})) \times \mathcal{K}(\mathcal{X}) \to \mathcal{K}(\mathcal{Y})$.

Proof.

(a) Given $F : \mathcal{C}(\mathcal{X};\mathcal{V}(\mathcal{Y}))$ and $V : \mathcal{O}(\mathcal{Y})$, to show $F^{-1}(V)$ is computable in $\mathcal{O}(\mathcal{X})$, we need to verify $x \in F^{-1}(V)$ for $x : \mathcal{X}$. This is effective since $x \in F^{-1}(V) \iff F(x) \between V$.
(b) Given $F : \mathcal{C}(\mathcal{X};\mathcal{K}(\mathcal{Y}))$ and $B : \mathcal{A}(\mathcal{Y})$, we have $x \notin F^{-1}(B) \iff F(x) \cap B = \emptyset \iff F(x) \subset (Y \setminus B)$, so $F^{-1}(B)$ is computable in $\mathcal{A}(\mathcal{X})$.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

(c) Given $F : \mathcal{C}(\mathcal{X}; \mathcal{V}(\mathcal{Y}))$ and $A : \mathcal{V}(\mathcal{X})$, to show $F(A)$ is computable in $\mathcal{V}(\mathcal{Y})$, we need to verify $F(A) \between V$ for $V : \mathcal{O}(\mathcal{Y})$. This is effective since $F(A) \between V \iff A \between F^{-1}(V)$, and $F^{-1}(V)$ is computable in $\mathcal{O}(\mathcal{X})$ by (a) above.

(d) Given $F : \mathcal{C}(\mathcal{X}; \mathcal{K}(\mathcal{Y}))$ and $C : \mathcal{K}(\mathcal{X})$, we have $F(C) \subset V \iff C \subset F^{\Leftarrow}(V) \iff C \subset X \setminus F^{-1}(Y \setminus V)$, which is effective since $Y \setminus V : \mathcal{A}(\mathcal{Y})$, $F^{-1}(Y \setminus V) : \mathcal{A}(\mathcal{X})$ and $X \setminus F^{-1}(Y \setminus V) : \mathcal{O}(\mathcal{X})$. □

Considering the proof of Theorem 3.32, we see that computation of $F(S)$ can be performed directly using F^{-1}. Further, since $F(x) = F(\{x\})$, the action of F on points can be recovered from its action on sets. We therefore obtain equivalence of representations of lower/upper semicontinuous functions.

Corollary 3.33. *The following types are equivalent:*

(a) $F : \mathcal{X} \to \mathcal{V}(\mathcal{Y})$, $F^{-1} : \mathcal{O}(\mathcal{Y}) \to \mathcal{O}(\mathcal{X})$, *and* $F : \mathcal{V}(\mathcal{X}) \to \mathcal{V}(\mathcal{Y})$ *as representations of lower-semicontinuous closed-valued functions* F.

(b) $F : \mathcal{X} \to \mathcal{K}(\mathcal{Y})$, $F^{-1} : \mathcal{A}(\mathcal{Y}) \to \mathcal{A}(\mathcal{X})$, *and* $F : \mathcal{K}(\mathcal{X}) \to \mathcal{K}(\mathcal{Y})$ *as representations of upper-semicontinuous compact-valued functions* F.

Classically, the is a bijection between multivalued functions $X \rightrightarrows Y$ and their graphs as subsets of $X \times Y$. However, this only holds computably for open- and closed-valued functions.

Proposition 3.34 (Graphs of multivalued functions). *Under the graph operator:*

(a) The types $\mathcal{C}(\mathcal{X}; \mathcal{O}(\mathcal{Y}))$ *and* $\mathcal{O}(\mathcal{X} \times \mathcal{Y})$ *are equivalent.*

(b) The types $\mathcal{C}(\mathcal{X}; \mathcal{A}(\mathcal{Y}))$ *and* $\mathcal{A}(\mathcal{X} \times \mathcal{Y})$ *are equivalent.*

Proof.

(a) $y \in F(x) \iff (x, y) \in \mathrm{graph}\,(F)$.

(b) $y \notin F(x) \iff (x, y) \notin \mathrm{graph}\,(F)$. □

Example 3.35.

(a) The types $\mathcal{C}(\mathcal{X}; \mathcal{V}(\mathcal{Y}))$ *and* $\mathcal{V}(\mathcal{X} \times \mathcal{Y})$ *are in general not equivalent. Consider* $\mathcal{X} = \mathcal{Y} = \mathcal{R}$ *and the functions* F_n *with* $\mathrm{graph}(F_n) = \{(2^{-n}, 0)\}$ *and* $\mathrm{graph}(F_\infty) = \{(0, 0)\}$. *Then* $\mathrm{graph}(F_n) \to \mathrm{graph}(F_\infty)$ *in* $\mathcal{V}(\mathcal{R} \times \mathcal{R})$, *but since* $F_n(0) = \emptyset$ *for* $n \in \mathbb{N}$ *and* $F_\infty(0) = \{0\}$, $F_n \not\to F_\infty$ *in* $\mathcal{C}(\mathcal{X}; \mathcal{V}(\mathcal{Y}))$.

(b) The types $\mathcal{C}(\mathcal{X}; \mathcal{K}(\mathcal{Y}))$ *and* $\mathcal{K}(\mathcal{X} \times \mathcal{Y})$ *are in general also not equivalent. For if* $F_n(n) = \{0\}$ *and* $F_n(x) = \emptyset$ *otherwise (including* $F_\infty(x) = \emptyset$*), then* $F_n \to F_\infty$ *in* $\mathcal{C}(\mathcal{X}; \mathcal{K}(\mathcal{X}))$, *but* $\mathrm{graph}(F_n) \not\to \mathrm{graph}(F_\infty)$ *in* $\mathcal{K}(\mathcal{R} \times \mathcal{R})$ *since* $\mathrm{graph}(F_\infty) \subset \emptyset$ *but* $\mathrm{graph}(F_n) \not\subset \emptyset$ *for all* $n \in \mathbb{N}$.

The above observation has important ramifications for the notion of causality in dynamic systems. With additional *effectivity properties* of Definition 3.38, we can derive reductions between $\mathcal{C}(\mathcal{X}; \mathcal{V}(\mathcal{Y}))$ and $\mathcal{V}(\mathcal{X} \times \mathcal{Y})$, and between $\mathcal{K}(\mathcal{X} \times \mathcal{Y})$ and $\mathcal{C}(\mathcal{X}; \mathcal{K}(\mathcal{Y}))$, given as Proposition 3.44.

Define the multivalued *restriction map* R_S for a set S by $R_S(x) = \{x\}$ if $x \in S$ and $R_S(x) = \emptyset$ otherwise. The following useful result shows that open/closed sets have overt/compact restriction maps.

Theorem 3.36 (Properties of restriction maps). *The restriction map* $S \mapsto R_S$ *is* $\mathcal{O}(\mathcal{X}) \to \mathcal{C}(\mathcal{X}; \mathcal{V}(\mathcal{X}))$*-computable and* $\mathcal{A}(\mathcal{X}) \to \mathcal{C}(\mathcal{X}; \mathcal{K}(\mathcal{X}))$*-computable.*

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Proof. If U is open, then $R_U(x) \between V \iff x \in U \cap V$. If A is closed, then $R_A(x) \subset V \iff x \in V \cup (X \setminus A)$. $\square$

We can also define types of partial functions $f : X \rightharpoonup Y$, which are defined on $\mathrm{dom}\,(f) \subset X$. Note that a partial function f gives rise to a multivalued function F defined by

$$F(x) = \begin{cases} \{f(x)\} \text{ if } x \in \mathrm{dom}\,(f); \\ \emptyset \text{ if } x \notin \mathrm{dom}\,(f). \end{cases} \tag{10}$$

This allows partial function types to be defined as subtypes of multivalued function types:

Definition 3.37 (Partial function types).

(a) The type of partial functions with open domain, $\mathcal{C}_{\mathcal{O}}(\mathcal{X};\mathcal{Y})$, is the subtype of $\mathcal{C}(\mathcal{X};\mathcal{V}(\mathcal{Y}))$ under (10).
(b) The type of partial functions with closed domain, $\mathcal{C}_{\mathcal{A}}(\mathcal{X};\mathcal{Y})$, is the subtype of $\mathcal{C}(\mathcal{X};\mathcal{K}(\mathcal{Y}))$ under (10).

In each case, $\mathrm{dom}\,(f)$ is computable from f and $f(x)$ can be computed for all x in the domain; the difference being that for $f : \mathcal{C}_{\mathcal{O}}(\mathcal{X};\mathcal{Y})$, we can prove f is defined for all x in its domain.

3.6 Effective topological properties

We now give "effective" versions of some classical properties of topological spaces and deduce some of their properties.

Definition 3.38 (Effectivity properties). *Let $\mathcal{X}$ be a topological type. Then:*

(a) $\mathcal{X}$ is effectively discrete *if the equality relation is verifiable, that is, $= : \mathcal{X} \times \mathcal{X} \to \mathcal{S}$ is computable.*
(b) $\mathcal{X}$ is effectively Hausdorff *if the inequality relation is verifiable, that is, $\neq : \mathcal{X} \times \mathcal{X} \to \mathcal{S}$ is computable.*
(c) $\mathcal{X}$ is effectively overt *if X is a computable element of $\mathcal{V}(\mathcal{X})$.*
(d) $\mathcal{X}$ is effectively compact *if X is a computable element of $\mathcal{K}(\mathcal{X})$.*

Remark 3.39. If $X \times X$ with the product topology is a sequential space (in particular, if X is locally compact), and X is effectively Hausdorff, then it is Hausdorff. However, if $X \times X$ is not a sequential space, it is possible that $\mathcal{X}$ is effectively Hausdorff but not Hausdorff. For if $\{(x, y) \mid x \neq y\}$ is sequentially open but not open, and hence contain points $x \neq y$ such that for any open $U \ni x$ and $V \ni y$, $U \cap V \neq \emptyset$.

Remark 3.40. Note that for open $U \subset X$, we have $X \between U \iff U \neq \emptyset$. Hence saying X is effectively overt is equivalent to saying nonemptiness of open sets in X is verifiable. Similarly, $X \subset U \iff U = X$, so saying X is effectively compact is equivalent to saying entireness of open sets is verifiable.

The following result gives a link between effectivity properties of the topological type and conversions between subsets of that type.

Theorem 3.41. *Let $\mathcal{X}$ be a topological type. Then:*

(a) The interior function $\mathcal{V}(\mathcal{X}) \to \mathcal{O}(\mathcal{X})$ is computable if $\mathcal{X}$ is effectively discrete.
(b) The closure function $\mathcal{K}(\mathcal{X}) \to \mathcal{A}(\mathcal{X})$ is computable if $\mathcal{X}$ is effectively Hausdorff.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

(c) The closure function $\mathcal{O}(\mathcal{X}) \to \mathcal{V}(\mathcal{X})$ is computable if, and only if $\mathcal{X}$ is effectively overt.
(d) The saturation function $\mathcal{A}(\mathcal{X}) \to \mathcal{K}(\mathcal{X})$ is computable if, and only if $\mathcal{X}$ is effectively compact.

Proof.

(a) If $\mathcal{X}$ is effectively discrete, then the function $x \mapsto \{x\}$ is computable $\mathcal{X} \to \mathcal{O}(\mathcal{X})$, since $y \in \{x\} \iff x = y$. Then for any $W \in \mathcal{V}(\mathcal{X})$, we have $x \in W \iff W \between \{x\}$.
(b) If $\mathcal{X}$ is effectively Hausdorff, then $x \mapsto \{x\}$ is computable $\mathcal{X} \to \mathcal{A}(\mathcal{X})$, since $y \notin \{x\} \iff x \neq y$. Then $x \notin C \iff C \subset (X \setminus \{x\})$.
(c) Suppose X is computable in $\mathcal{V}(\mathcal{X})$, and let $U \in \mathcal{O}(\mathcal{X})$. Then, $\mathrm{cl}(U) = \mathrm{cl}(U \cap X)$ is computable in $\mathcal{V}(\mathcal{X})$ by Theorem 3.27(f). Conversely, if the closure function $\mathcal{O}(\mathcal{X}) \to \mathcal{V}(\mathcal{X})$ is computable, then since X is computable in $\mathcal{O}(\mathcal{X})$, it is also computable in $\mathcal{V}(\mathcal{X})$.
(d) Suppose X is computable in $\mathcal{K}(\mathcal{X})$, and let $A \in \mathcal{A}(\mathcal{X})$. Then $A = A \cap X$ is computable in $\mathcal{K}(\mathcal{X})$ by Theorem 3.27(g). Conversely, if the saturation function $\mathcal{A}(\mathcal{X}) \to \mathcal{K}(\mathcal{X})$ is computable, then since X is computable in $\mathcal{A}(\mathcal{X})$, it is also computable in $\mathcal{K}(\mathcal{X})$. □

Computability of interior of an overt set is in general weaker than effective discreteness, and computability of the closure of a compact set is in general weaker than effective Hausdorffness:[3]

Example 3.42. *Consider the Sierpinski type $\mathcal{S} = \{\top, \uparrow\}$. The open sets are $\{\}, \{\top\}, \{\top, \uparrow\}$. The overt sets are given by the closed sets $\{\}, \{\uparrow\}, \{\top, \uparrow\}$ with interiors $\{\}^\circ = \{\uparrow\}^\circ = \{\}$ and $\{\top, \uparrow\}^\circ = \{\top, \uparrow\}$. Hence, the interior function satisfies $S^\circ(\top) = S^\circ(\uparrow) = b_S(\{\top\})$, so is computable given the $\mathcal{V}(\mathcal{S})$-name of S. However, $\mathcal{S}$ is not a discrete space.*

Similarly, the closed sets are $\{\}, \{\uparrow\}, \{\top, \uparrow\}$, and the compact sets are given by saturated sets $\{\}, \{\top\}, \{\top, \uparrow\}$ with closures $\overline{\{\}} = \{\}$ and $\overline{\{\top\}} = \overline{\{\top, \uparrow\}} = \{\top, \uparrow\}$. Hence the closure function is computable, though $\mathcal{S}$ is not a Hausdorff space.

The following result shows that projection maps are computable on open and closed subsets given suitable effectivity properties.

Proposition 3.43.

(a) Suppose $\mathcal{Y}$ is effectively overt. Then the set projection operator π_X defined by $\pi_X(S) = \{x \in X \mid \exists y \in Y,\ (x, y) \in S\}$ is computable $\mathcal{O}(\mathcal{X} \times \mathcal{Y}) \to \mathcal{O}(\mathcal{X})$.
(b) Suppose $\mathcal{Y}$ is effectively compact. Then the set projection operator π_X is computable $\mathcal{A}(\mathcal{X} \times \mathcal{Y}) \to \mathcal{A}(\mathcal{X})$.

Proof.

(a) $x \in \pi_X(U) \iff (\{x\} \times Y) \cap U \neq \emptyset$.
(b) $x \notin \pi_X(A) \iff (\{x\} \times Y) \cap A = \emptyset$. □

Proposition 3.44.

(a) The graph operator is computable $\mathcal{C}(\mathcal{X}; \mathcal{V}(\mathcal{Y})) \to \mathcal{V}(\mathcal{X} \times \mathcal{Y})$ if $\mathcal{X}$ is effectively overt, and its inverse is computable $\mathcal{V}(\mathcal{X} \times \mathcal{Y}) \to \mathcal{C}(\mathcal{X}; \mathcal{V}(\mathcal{Y}))$ if $\mathcal{X}$ is effectively discrete.
(b) The graph operator is computable $\mathcal{C}(\mathcal{X}; \mathcal{K}(\mathcal{Y})) \to \mathcal{K}(\mathcal{X} \times \mathcal{Y})$ if $\mathcal{X}$ is effectively compact, and its inverse is computable $\mathcal{K}(\mathcal{X} \times \mathcal{Y}) \to \mathcal{C}(\mathcal{X}; \mathcal{K}(\mathcal{Y}))$ if $\mathcal{X}$ is effectively Hausdoff.

Proof.

(a) If $\mathcal{X}$ is effectively overt, then we can compute $\text{graph}\,(F) \between W \iff \exists x,\ F(x) \between \{y \mid (x, y) \in W\}$. Conversely, $F(x) \between V \iff \text{graph}\,(F) \between \{x\} \times V$.
(b) If $\mathcal{X}$ is effectively compact, then we can compute $\text{graph}\,(F) \subset W \iff \forall x,\ F(x) \subset \{y \mid (x, y) \in W\}$. Conversely, $F(x) \subset V \iff \text{graph}\,(F) \subset (X \times Y) \setminus (\{x\} \times (Y \setminus V))$. □

Note that since the effective Hausdorff property and the effective overtness property are relatively common for the main types used in analysis, the conversions $\mathcal{O}(\mathcal{X}) \to \mathcal{V}(\mathcal{X})$ and $\mathcal{K}(\mathcal{X}) \to \mathcal{A}(\mathcal{X})$ are usually computable, whereas the conversions $\mathcal{V}(\mathcal{X}) \to \mathcal{O}(\mathcal{X})$ and $\mathcal{A}(\mathcal{X}) \to \mathcal{K}(\mathcal{X})$ are usually not. This means that a description of an open set in $\mathcal{O}(\mathcal{X})$ provides more information than a description of its closure in $\mathcal{V}(\mathcal{X})$, and a description of a compact set in $\mathcal{K}(\mathcal{X})$ provides more information than its description in $\mathcal{A}(\mathcal{X})$.

However, we would really like to prove that the countable product of effectively compact spaces is effectively compact. This is known to be true for countably based spaces (Escardó 2004) (see our proof in Theorem 3.66(b)), but unknown in general.

Definition 3.45. *Let $\mathcal{X}$ be a topological type. Then:*

(a) $\mathcal{X}$ is effectively separable *if there is a computable function $\mathcal{N} \to \mathcal{X}$ with dense range.*
(b) $\mathcal{X}$ is effectively coverable *if there is a computable surjective function $\mathcal{C} \to \mathcal{X}$, where $\mathcal{C}$ is the Cantor space $\{0, 1\}^{\omega}$.*

The following proposition provides the relationship between overtness and separability, similarly for compactness and coverabilty.

Proposition 3.46. *Let $\mathcal{X}$ be a topological type. Then:*

(a) $\mathcal{X}$ is effectively overt if it is effectively separable.[4]
(b) $\mathcal{X}$ is effectively compact if it is effectively coverable.[5]

Proof.

(a) Let $\xi : \mathcal{N} \to \mathcal{X}$ be a computable function encoding a dense sequence. Let U be a nonempty open set, with $\chi_U : \mathcal{X} \to \mathcal{S}$ its characteristic function. We need to show that $U \neq \emptyset$ is verifiable, for which it suffices to prove $x \in U$ for some x. Since ξ has dense range, there exists n such that $\xi(n) \in U$. The result follows since we can simulate computing $\chi_U \circ \xi$ over all $n \in \mathbb{N}$, which is guaranteed to terminate once some n for which $\xi(n) \in U$ is found. The result follows by Proposition 3.17(a).
(b) Let $\xi : \mathcal{C} \twoheadrightarrow \mathcal{X}$ be a computable surjective function. Let U be an open set. We need to show that the property $X \subset U$ is verifiable, which reduces to showing $\chi_U \circ \xi$ is always $\top$ by Proposition 3.17(b) □

We now prove that the effective separability and coverability properties extend to countable products. Note that Theorem 3.47(b) is an effective version of Tychonoff's theorem that the countable product of compact spaces is compact, but under the stronger hypothesis that the spaces are effectively coverable.

Theorem 3.47. *Let $\mathcal{X}_n$, $n \in \mathbb{N}$ be topological types.*

(a) If each $\mathcal{X}_n$ is effectively separable, then $\prod_{n=0}^{\infty} \mathcal{X}_n$ is effectively separable.
(b) If each $\mathcal{X}_n$ is effectively coverable, then $\prod_{n=0}^{\infty} \mathcal{X}_n$ is effectively coverable.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Proof.

(a) Suppose $\xi_n : \mathcal{N} \to \mathcal{X}_n$ are computable functions with countable dense range. Let $h : \mathbb{N}^* \to \mathbb{N}$ be a computable bijective function. Define $\eta : \mathcal{N} \to \prod_{n=0}^{\infty} \mathcal{X}_n$ by $\eta(h(m_0, \ldots, m_j)) = (\xi_0(m_0), \xi_1(m_1), \ldots, \xi_j(m_j), \xi_{j+1}(0), \xi_{j+2}(0), \ldots)$. Then, η has dense range and is computable by construction.

(b) Since each $\mathcal{X}_n$ is effectively coverable, there exist computable functions $\xi_n : \mathcal{C} \to \mathcal{X}_n$ for which the saturation of the range is X_n. Define $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ by $g(i,j) = (i+j)(i+j+1)/2 + j$, and note that g is a bijection. Define $\tau : \mathcal{C}^{\omega} \to \mathcal{C}$ by $\tau(p_0, p_1, \ldots)_k = p_{i,j}$ where $g(i,j) = k$, and note that τ is also a bijection. Define $\eta : \mathcal{C} \to \prod_{n=0}^{\infty} X_n$ by $(\eta(q))_i = \xi_i(p_i)$ for $q = \tau(p_0, p_1, \ldots)$. Then, $\eta : \mathcal{C} \to \prod_{n=0}^{\infty} X_n$ is surjective since each ξ_i is surjective and is computable by construction. □

We now consider effective version of the local-compactness property. Recall from Section 3.1 that a topological space is *locally compact* if every point has arbitrarily-small compact neighborhoods.

Notation 3.48. *We write $U \Subset V$ if every open cover of V has a finite subcover of U. We define $\Uparrow U = \{V \in \mathcal{O}(X) \mid U \Subset V\}$.*

Note that the notation $A \Subset B$ (or $A \subset\subset B$) is often used in classical topology to denote that the closure of A is a compact subset of the interior of B and is analogous to the way-below relation $\ll$ of Scott domain theory (Gierz et al. 2003, Definition 1.1). $\twoheadrightarrow$ is also used in domain theory.

Definition 3.49 (Effectively locally compact). *A computable type $\mathcal{X}$ is* effectively locally compact *if there is a computable function $\mathcal{N} \to \mathcal{O}(\mathcal{X}) \times \mathcal{K}(\mathcal{X})$, $n \mapsto (V_n, K_n)$ such that for any n, $V_n \subset K_n$, and for any $W \in \mathcal{O}(X)$ and $x \in W$, there exists n such that $x \in V_n$ and $K_n \subset W$.*

We say that $\mathcal{X}$ is strongly effectively locally compact *if we can take $K_n = \bigcap\{W \in \mathcal{O}(X) \mid V_n \Subset W\}$ for all n; if X is a Hausdorff space, this is equivalent to $K_n = \mathrm{cl}(V_n)$.*

Remark 3.50. Using the definition given, it is immediate that X is a countably based space, since any U is given by $U = \bigcup\{V_n \mid K_n \subset U\}$, and is effectively overt, since $U \neq \emptyset \iff \exists n,\ U \supset K_n$. It would be interesting to find a weaker notion of effective local compactness which does not imply these properties.

Frequently, rather than find a neighborhood of a point, we need a neighborhood of a compact set. The next result shows that this can always be done.

Proposition 3.51. *Suppose $\mathcal{X}$ is effectively locally compact. Then, there is a computable function $\mathcal{N} \to \mathcal{O}(\mathcal{X}) \times \mathcal{K}(\mathcal{X})$, $n \mapsto (W_n, L_n)$ with $W_n \subset L_n$ such that for any compact C and open U with $C \subset U$, there exists n with $C \subset W_n$ and $L_n \subset U$.*

Proof. Consider the pairs $(\bigcup_{n \in N} V_n, \bigcup_{n \in N} K_n)$ for N a finite subset of $\mathbb{N}$. Let $C \subset U$ be compact. Then, $\{V_n \mid K_n \subset U\}$ is an open cover of C, so has a finite subcover $\{V_n \mid n \in N\}$. By construction $\{K_n \mid n \in N\} \subset U$. □

3.7 Standard representations of topological spaces

In this section, we show a general purpose way of building representations of topological and metric spaces. These constructions can be used to give equivalent representations for the spaces constructed as category-theoretic products and exponentials, or to give representations of new spaces which extend the type theory. We also give concrete representations of set and function types, and conditions under which they are equivalent to the general category-theoretic constructions. Early work on computability in spaces with computable basis was given by Hauck (1980, 1981). The material of this section is based on the definitions of Weihrauch and Grubba (2009).

Recall that if (X, τ) is a topological space, then a collection $\beta \subset \tau$ is a *base* if any open set is a union of elements of β, and $\sigma \subset \tau$ is a *sub-base* if the set of finite intersections of elements of σ forms a base.

Definition 3.52 (Effective topological spaces; standard representation). *A* sub-effective topological space *is a tuple (X, τ, σ, ν) where (X, τ) is a second-countable Kolmogorov (T_0) space, σ is a countable sub-base for τ, and ν is a notation for σ .*

The standard representation *δ of (X, τ, σ, ν) is the function $\delta : \Sigma^\omega \rightharpoonup X$ such that*

$$\delta(p) = x \iff \{\nu(w) \mid w \lhd p\} = \{J \in \sigma \mid x \in J\}. \tag{11}$$

An effective topological space *(X, τ, β, ν) is a sub-effective topological space where β is a base for τ and $\emptyset \in \beta$.*

In other words, a name of x in the standard representation encodes a list of *all* $J \in \sigma$ such that $x \in J$.

Note that given a sub-effective topological space (X, τ, σ, ν), we can canonically construct an equivalent effective topological space (X, τ, β, ρ) by taking β to be the set of finite intersections of elements of σ, and $\rho(\langle p_1, \ldots, p_n\rangle) = \nu(p_1) \cap \cdots \cap \nu(p_n)$.

The following result shows that the standard representation of a sub-effective topological space is an admissible quotient representation. The proof is similar to that of the corresponding parts of Theorem 4.5, but is considerably simpler, so we include it in full.

Theorem 3.53. *Let (X, τ, σ, ν) be a sub-effective topological space. Then, the standard representation δ of (X, τ, σ, ν) is an admissible quotient representation.*

Proof. *δ is continuous:* Let $U \subset X$ be open and $x = \delta(p) \in U$. Since σ is a sub-base, there exist $J_1, \ldots, J_k \in \sigma$ such that $x \in \bigcap_{i=1}^k J_i \subset U$. Since p is a name of x, we have $w_i \lhd p$ where $\nu(w_i) = J_i$ for $i = 1, \ldots, k$. Hence, there is a prefix v of p such that $w_i \lhd v$ for all $i = 1, \ldots, k$. Hence, $\delta(q) \in \bigcap_{i=1}^k J_i \subset U$ for any $q \in v\Sigma^\omega$.

δ is a quotient map: Suppose $W \subset X$ and $\delta^{-1}(W)$ is open. Let $x \in W$ and $p \in \delta^{-1}(x)$. Then, there is a prefix v of p such that $q \in \delta^{-1}(W)$ whenever v is a prefix of q. From the information contained in v, we can only deduce that there are open sets $J_1, \ldots, J_k \in \sigma$ such that $x \in J_i$ for all $i = 1, \ldots, k$. Hence, $\bigcap_{i=1}^k J_i$ is an open neighborhood of x which is a subset of W.

δ is admissible: Suppose $\phi : \Sigma^\omega \rightharpoonup X$ is a continuous partial function. Define a function $\tilde{\eta} : \Sigma^\omega \rightharpoonup \Sigma^\omega$ such that if v_1 is a prefix of v_2, then $\tilde{\eta}(v_1)$ is a prefix of $\tilde{\eta}(v_2)$, and that $w \lhd \eta(v)$ if, and only if, $\phi(p) \in \nu(w)$ whenever $p \in v\Sigma^\omega$. Define $\eta(p) = \lim_{n\to\infty} \tilde{\eta}(p|_n)$. By padding $\tilde{\eta}(v)$ with names of the empty set if necessary, we can ensure that $\eta : \Sigma^\omega \rightharpoonup \Sigma^\omega$. Then by definition, $w \lhd \eta(p) \iff \phi(p) \in \nu(W)$, so $\delta(\eta(p)) = \phi(p)$ for all $p \in \mathrm{dom}\,(\phi)$. □

We would like to define concrete representations of subsets of an effective topological space. Since any open set is a union of basic open sets, we can define a representation of open sets by taking a name of U to be an encoding of a list of basic open sets J whose union is U.

Definition 3.54 (Standard representations of open and closed sets). *Let (X, τ, β, ν) be an effective topological space.*

(a) *The standard representation $\theta_<$ of open sets $\mathcal{O}(X)$ is given by*

$$\theta_<(p) = U \iff \bigcup\{\nu(w) \mid w \lhd p\} = U.$$

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

(b) *The standard upper representation $\psi_>$ of closed sets $\mathcal{A}(X)$ is given by*

$$\psi_>(p) = A \iff \bigcup\{\nu(w) \mid w \lhd p\} = X \setminus A.$$

Note that a $\theta_<$-name of $\emptyset$ is $\langle w_0, w_0, \ldots\rangle$ where $\nu(w_0) = \emptyset$.

The following result shows that it is always possible to verify $x \in U$ given a $\theta_<$-name of U.

Lemma 3.55. *Let (X, τ, β, ν) be an effective topological space. Then inclusion $\in : X \times \mathcal{O}(X) \to \mathbb{S}$ is computable.*

Proof. If $x \in U$ and $U = \bigcup_{i=0}^{\infty} J_i$, then $x \in U \iff \exists m \in \mathbb{N},\ x \in J_m$. □

Without additional assumptions, it is not possible to compute intersections, in general.

Definition 3.56 (Computable intersection property). *An effective topological space (X, τ, β, ν) has the* computable intersection property *if there is a recursively-enumerable subset $\mathcal{I}$ of* dom $(\nu) \times$ dom $(\nu) \times$ dom (ν) *such that for all $w_1, w_2 \in$* dom (ν),

$$\nu(v_1) \cap \nu(v_2) = \bigcup\{\nu(w) \mid (v_1, v_2, w) \in \mathcal{I}\}.$$

For convenience, we may sometimes say that there is a recursively-enumerable subset $\mathcal{I}$ of $\beta \times \beta \times \beta$ such that $I_1 \cap I_2 = \bigcup\{J \in \beta \mid (I_1, I_2, J) \in \mathcal{I}\}$.

Proposition 3.57. *Let (X, τ, β, ν) be an effective topological space. Then intersection $\mathcal{O}(X) \times \mathcal{O}(X) \to \mathcal{O}(X)$ is $(\theta_<, \theta_<; \theta_<)$-computable if, and only if, (X, τ, β, ν) has the computable intersection property.*

Proof. If (X, τ, β, ν) has the computable intersection property, then we have $(\bigcup_{i=0}^{\infty} \nu(u_i)) \cap (\bigcup_{j=0}^{\infty} \nu(v_j)) = \bigcup_{i,j=0}^{\infty} (\nu(u_i) \cap \nu(v_j)) = \bigcup_{i,j=0}^{\infty} \bigcup\{\nu(w) \mid (u_i, v_j, w) \in \mathcal{I}\}$, so intersection is $(\theta_<, \theta_<; \theta_<)$-computable. Conversely, if intersection is $(\theta_<, \theta_<; \theta_<)$-computable, then $\nu(u) \cap \nu(v) = (\bigcup_{i=0}^{\infty} \nu(u)) \cap (\bigcup_{j=0}^{\infty} \nu(v)) = \bigcup_{k=0}^{\infty} \nu(w_k)$, so (X, τ, β, ν) has the effective intersection property. □

We now show that the computable intersection property is precisely what is needed that $(\mathcal{O}(X), \theta_<)$ has the type of $\mathcal{O}(\mathcal{X})$.

Theorem 3.58. *Let (X, τ, β, ν) be an effective topological space, and $\mathcal{X} = (X, [\delta])$ where δ is the standard representation of X. Then $(\mathcal{O}(X), \theta_<) \equiv (\mathcal{O}(X), \delta_{\mathcal{X}\to\mathbb{S}})$ (i.e. $\theta_<$ is computably equivalent to $\delta_{\mathcal{X}\to\mathbb{S}}$) if, and only if, (X, τ, β, ν) has the computable intersection property.*

Proof. If $\theta_<$ is equivalent to $\delta_{\mathcal{X}\to\mathbb{S}}$, then since intersection is $(\delta_{\mathcal{X}\to\mathbb{S}}, \delta_{\mathcal{X}\to\mathbb{S}}; \delta_{\mathcal{X}\to\mathbb{S}})$-computable by Theorem 3.27(b), it is also $(\theta_<, \theta_<; \theta_<)$-computable, so (X, τ, β, ν) has the computable intersection property by Proposition 3.57.

Conversely, suppose (X, τ, β, ν) has the computable intersection property, and $p : X \to \mathbb{S}$ is continuous. We need to show that we can compute a $\theta_<$-name of $U = p^{-1}(\top)$ given a name of p.

Recall that the valid names of $\top$ in $\mathbb{S}$ have the form `00···011···`, whereas the only valid name of $\perp$ is 0^ω. By Remark 2.17, it is possible to simulate computations of p, terminating whenever a '1' has been an output.

Let $w_1, \ldots, w_k \in$ dom (ν), and simulate a computation of p on input $\langle w_1, w_2, \ldots, w_k, \ldots\rangle$. If this computation outputs `1` after reading at most the words $w_1, \ldots, w_k$, then we can deduce that $\bigcap_{i=1}^{k} \nu(w_i) \subset U = p^{-1}(\top)$. For either the $\langle w_1, \ldots, w_k\rangle$ is a prefix of some valid name of some x, in which case the computation is valid, and $x \in U$ whenever $x \in \bigcap_{i=1}^{k} \nu(w_i)$, or the computation is invalid, in which case $\bigcap_{i=1}^{k} \nu(w_i) = \emptyset$.

Now since any $x \in p^{-1}(\top)$ has a δ-name, and the computation π of $p(x)$ is completed in a finite time, there exists $(w_1, \ldots, w_k)$ such that $q = \langle w_1, \ldots, w_k, \ldots\rangle$ is a δ-name of x, and computation of p on input q outputs `1` after reading at most the words $w_1, \ldots, w_k$, proving that $x \in \bigcap_{i=1}^{k} J_i \subset U$ with $J_i = \nu(w_i)$. Hence, $p^{-1}(\top)$ is equal to $\bigcup\{\bigcap_{i=1}^{k} \nu(w_i) \mid \pi$ outputs 1 after reading at most $w_1, \ldots, w_k\}$.

The result follows since we can write $\bigcap_{i=1}^{k} \nu(v_i) = \bigcup_{j=0}^{\infty} \nu(w_j)$ for some w_j computable from $v_1, \ldots, v_k$ by recursively applying the computable intersection property. $\square$

Since the type $(\mathcal{O}(X), [\theta_<])$ embeds computably into $\mathcal{X} \to \mathcal{S}$ by Lemma 3.55, without the computable intersection property, the reverse embedding may not hold. This indicates that the representation $\theta_<$ is in general too strong; it requires too much information about an open set to compute the intersection. Hence, the computable intersection property is what is needed to ensure that the explicit $\theta_<$ representation yields the same computability theory as our type-theoretic constructions. Following Weihrauch and Grubba (2009) (but differently from Weihrauch 2000, Definition 3.2.1), we make the following definition:

Definition 3.59 (Computable topological space). *An effective topological space* (X, τ, β, ν) *is a* computable topological space *if* dom (ν) *is recursively-enumerable and* (X, τ, β, ν) *has the computable intersection property.*

We now give concrete standard representations of overt and compact subsets of computable topological spaces. For the representation of open sets, a name p is a list of words $w \in \mathrm{dom}\,(\nu)$ encoding a list of basic open sets $J \in \beta$.

Definition 3.60 (Standard overt and compact set representations). *Let* (X, τ, β, ν) *be an effective topological space.*

(a) The standard lower representation $\psi_<$ *of the closed subsets of* X *is given by*

$$\psi_<(p) = A \iff \{\nu(w) \mid w \lhd p\} = \{J \in \beta \mid A \between J\}.$$

The corresponding type is the overt set type.

(b) The standard representation $\kappa_>$ *of compact sets* $\mathcal{K}(X)$ *is given by*

$$\kappa_>(p) = C \iff \{(\nu(w_1), \ldots, \nu(w_k)) \mid \langle w_1, \ldots, w_k \rangle \lhd p\} \\ = \{(J_1, \ldots, J_k) \in \beta^* \mid C \subset J_1 \cup \cdots \cup J_k\}.$$

We can also define a standard representation for the space of continuous functions.

Definition 3.61 (Standard representation of continuous functions). *Let* $(X, \tau_X, \beta_X, \nu_X)$ *be an effective topological space and* $(Y, \tau_Y, \sigma_Y, \nu_Y)$ *a sub-effective topological space. The standard representation* $\gamma_{X\to Y}$ *of* $\mathcal{C}(X; Y)$ *is given by*

$$\gamma_{X\to Y}(p) = f \iff \forall w_Y \in \mathrm{dom}\,(\nu_Y),\ f^{-1}(\nu(w_Y)) = \bigcup\{\nu_X(w_X) \mid \langle w_X, w_Y \rangle \lhd p\}.$$

Standard representations for spaces of multivalued functions can be defined analogously.

Intuitively, a name of f in the standard representation $\gamma_{X\to Y}$ of $\mathcal{C}(X; Y)$ encodes a list $\mathcal{F}$ of pairs $(I, J) \in \beta_X \times \sigma_Y$ such that $f^{-1}(J) = \bigcup\{I \mid (I, J) \in \mathcal{F}\}$.

The standard representations of spaces of overt and compact subsets and of continuous functions are equivalent to those in Section 3.5 for computable topological spaces:

Proposition 3.62. *Let* (X, τ, β, ν) *be a computable topological space. Then*

(a) The representation $\psi_<$ *of the closed subsets of* X *is equivalent to the representation of the overt set type given in Definition 3.24(a).*
(b) The representation $\kappa_>$ *of the compact subsets of* X *is equivalent to the representation of the compact set type given in Definition 3.24(b).*
(c) The representation $\gamma_{X\to Y}$ *of the continuous function* $X \to Y$ *is equivalent to the representation of the function type* $\mathcal{Y}^{\mathcal{X}}$ *given in Definition 3.20.*

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Even for effective topological spaces without the computable intersection property, evaluation of continuous functions $\varepsilon(f,x)=f(x)$ is $(\gamma_{X\to Y},\delta_X;\delta_Y)$-computable, intersection of overt and open sets is verifiable, as is subset of a compact set in an open set.

A useful technique for computing a value $x\in X$ is to compute the singleton $\{x\}$ as a compact (or overt) set. In a computable topological space, x can be computed from $\{x\}$, giving a converse to Theorem 3.27(j):

Proposition 3.63. *If (X,τ,β,ν) and $\{x\}$ are $\psi_<$-computable (i.e. as an element of $\mathcal{V}(\mathcal{X})$) or $\kappa_>$-computable (i.e. as an element of $\mathcal{K}(\mathcal{X})$), then it is computable in $\mathcal{X}$.*

Proof. Computing $\{x\}$ means verifying $x\in U$ for any open set U, including the basic sets β. □

In order to relate open and overt sets, or compact and closed sets, we need some extra effectivity properties (see Brattka and Presser 2003) of the notation ν of the basic open sets $J\in\beta$.

Definition 3.64. *Let (X,τ,β,ν) be an effective topological space. Then (X,τ,β,ν) has the:*

(a) effective overlap property *if $\{w_1,w_2\in\Sigma^*\times\Sigma^*\mid \nu(w_1)\cap\nu(w_2)\neq\emptyset\}$ is recursively-enumerable.*
(b) effective disjointness property *if there is a recursively-enumerable set $\mathcal{D}\subset\mathrm{dom}\,(\nu)\times\mathrm{dom}\,(\nu)$ such that $\bigcup\{\nu(w_1)\times\nu(w_2)\mid(w_1,w_2)\in\mathcal{D}\}=\{(x,y)\in X\times X\mid x\neq y\}$.*

The following theorem relates the effectivity properties for the standard representations with

Theorem 3.65. *Let (X,τ,β,ν) be a computable topological space. Then*

(a) the effective overlap property is equivalent to every open set being effectively overt.
(b) the effective disjointness property is equivalent to every saturated compact set being effectively closed.

Proof.

(a) If (X,τ,β,ν) has the effective intersection property, then if $U=\bigcup_{i=0}^{\infty}J_i$ is open and $L\in\beta$, then $U\between L\iff\exists i\in\mathbb{N},\ J_i\cap L\neq\emptyset$, so U is computable as an overt set since we can verify intersection with elements of β. Conversely, for arbitrary $J,L\in\beta$, we can write $J=\bigcup_{n=0}^{\infty}J$ as a countable union of basic open sets and verify $J\cap L\neq\emptyset$ since $J\between L\iff(\bigcup_{i=0}^{\infty}J)\between L$.
(b) The effective disjointness property implies that $\{(x_1,x_2)\in X\times X\mid x_1\neq x_2\}$ is computable as an open set, so inequality is verifiable. Hence, X is effectively Hausdoff, so every saturated compact set is effectively closed. Conversely, suppose every saturated compact set is effectively closed, and $x_1\neq x_2$. Then, there exists $I\in\beta$ containing only one of x_1 or x_2, say $I\ni x_1$. Then, $\mathrm{sat}\,(\{x_1\})$ is a saturated compact set and a subset of U. Since $\mathrm{sat}\,(\{x_1\})$ is computable as a closed set, $x_2\notin\mathrm{sat}\,(\{x_1\})$ is verifiable, so $x_1\neq x_2$ is verifiable. □

The definition of the effective disjointness predicate used here is different from that of Brattka and Presser (2003). We return to this point in Remark 3.73.

An effective version of Tychonoff's theorem holds for computable topological spaces.

Theorem 3.66. *Suppose every $\mathcal{X}_n$ is a computable countably based space.*

(a) The countable-product operator on nonempty *sets is computable $\prod_{n=0}^{\infty}\mathcal{V}(\mathcal{X}_n)\rightharpoonup\mathcal{V}(\prod_{n=0}^{\infty}\mathcal{X}_n)$.*

(b) The countable-product operator is computable $\prod_{n=0}^{\infty}\mathcal{K}(\mathcal{X}_n)\to\mathcal{K}(\prod_{n=0}^{\infty}\mathcal{X}_n)$.

Proof. Use base β_n of $\mathcal{X}_n$. A base β_∞ for the topology of $\prod_{n=0}^{\infty}X_n$ is given by sets $J=I_0\times I_1\times\cdots\times I_{n-1}\times X_n\times X_{n+1}\times\cdots$ where $I_n\in\beta_n$.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

(a) Given nonempty overt sets $V_n : \mathcal{V}(\mathcal{X})$, since a general open set U in $\prod_{n=0}^{\infty} X_n$ is represented as a countable union of $J \in \beta_\infty$, it suffices to show intersection of $\prod_{n=0}^{\infty} V_n$ with some J can be verified. Since each V_k is nonempty, $V_k \between X_K$, so $\prod_{k=0}^{\infty} V_k \between I_0 \times I_1 \times \cdots \times I_{n-1} \times X_n \times X_{n+1} \times \cdots \iff \prod_{k=0}^{n-1} V_k \between \prod_{k=0}^{n-1} I_k$. The result follows from Theorem 3.28(d), since $\prod_{n=0}^{m-1} \mathcal{X}_n$ is effectively overt.

(b) By the classical Tychonoff theorem, it suffices to show that given a *finite* basic open cover of $\prod_{n=0}^{\infty} C_n$, we can prove that this is indeed a cover. Given the presentation $J_i = \prod_{j=1}^{m_i-1} I_{i,j} \times \prod_{j=m_i}^{\infty} X_j$, by taking $M = \max_i m_i$, we need only to consider the open cover $\bigcup_{i=0}^{N-1} (\prod_{j=1}^{m_i-1} I_{i,j} \times \prod_{j=m_i}^{M-1} X_j)$ of $\prod_{n=0}^{M-1} C_n$. The result follows from Theorem 3.28, since $\prod_{n=0}^{M-1} C_n$ is computable in $\mathcal{K}(\prod_{n=0}^{M-1} \mathcal{X}_n)$. □

Note that if the overt sets were allowed to be empty, computing $\prod_{n=0}^{\infty} V_n$ would enable verifying $\prod_{n=0}^{\infty} V_n \between \prod_{n=0}^{\infty} X_n$, which is equivalent to $\bigwedge_{n=0}^{\infty} V_n \neq \emptyset$, a countable conjunction.

An alternative representation of open sets of an effective topological space (X, τ, β, ν) is the *Scott representation*, based on the Scott topology (Gierz et al. 1980). For locally compact spaces, it turns out that this is equivalent (following Definition 2.21) to the standard representation.

Definition 3.67. *Let (X, τ, β, ν) be an effective topological space. The Scott representation $\theta'_<$ of open sets $\mathcal{O}(X)$ is given by*

$$\theta'_<(p) = U \iff \{\nu(w) \mid w \lhd p\} = \{I \in \beta \mid I \Subset U\}.$$

Remark 3.68. If (X, τ) is a Hausdorff space, then $U \Subset V$ for open U, V is equivalent to $\overline{U}$ a compact subset of U.

In order to relate the representations θ and θ', we need an extra *effective covering property*.

Definition 3.69 (Effective covering property). *Let (X, τ, β, ν) be an effective topological space. Then, (X, τ, β, ν) has the* effective covering property *if $\{v, w_1, \ldots, w_k \in \Sigma^* \times \Sigma^* \times \cdots \times \Sigma^* \mid \nu(v) \Subset \nu(w_1) \cup \cdots \cup \nu(w_k)\}$ is recursively-enumerable.*

The following theorem shows that the effective covering property is a necessary and sufficient condition for equivalence of the representations $\theta_<$ and $\theta'_<$.

Theorem 3.70. *Let (X, τ, β, μ) be an effective topological space which is locally compact. Then, the representations $\theta_<$ and $\theta'_<$ are equivalent if, and only if, (X, τ, β, μ) has the effective covering property.*

Proof. Suppose (X, τ, β, μ) has the effective covering property. For any locally compact space (X, τ) and any $U \in \mathcal{O}(X)$, we have $U = \bigcup\{I \in \beta \mid I \Subset U\}$. Hence, any $\theta'_<$-name is also a $\theta_<$-name. If $U = \bigcup_{i=0}^{\infty} J_i$ and $I \Subset U$, there exists k such that $I \subset \bigcup_{i=1}^{k} J_i$, so we can prove that $I \Subset U$. Hence, any $\theta_<$-name is also a $\theta'_<$-name

Conversely, if $\theta_<$ and $\theta'_<$ are equivalent, then we can compute a $\theta'_<$-name of $\bigcup_{i=1}^{k} J_i$, which amounts to enumerating $\{w \in \mathrm{dom}(\nu) \mid \nu(w) \Subset \bigcup_{i=1}^{k} J_i\}$. Hence, (X, τ, β, μ) has the effective covering property. □

We now give standard representations of spaces of functions and multifunctions in an effectively locally compact space, based on the *Isbell topology* (Isbell 1975).

Definition 3.71. *If $(X, \tau_X, \beta_X, \nu_X)$ is a computable topological space and $(Y, \tau_Y, \sigma_Y, \nu_Y)$ be a sub-effective topological space, then*

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

1'. *The* Isbell representation γ' *of* $\mathcal{C}(X; Y)$ *is given by*

$$\gamma'(p) = f \iff \{(\nu_X(v), \nu_Y(w)) \mid \langle v, w\rangle \lhd p\} = \{(I, J) \in \beta_X \times \beta_Y \mid I \Subset f^{-1}(J)\}.$$

In other words, a γ'*-name of* f *encodes a list* $\mathcal{F}'$ *of* all *pairs* $(I, J) \in \beta_X \times \sigma_Y$ *such that* $I \Subset f^{-1}(J)$.

2'. *The Isbell representation* $\mu'_<$ *of* $\mathcal{C}(X; \mathcal{V}(Y))$ *is given by*

$$\mu'_<(p) = F \iff \{(\nu_X(v), \nu_Y(w)) \mid \langle v, w\rangle \lhd p\} = \{(I, J) \in \beta_X \times \beta_Y \mid I \Subset F^{-1}(J)\}.$$

3'. *The Isbell representation* $\mu'_>$ *of* $\mathcal{C}(X; \mathcal{K}(Y))$ *is given by*

$$\begin{aligned}\mu'_<(p) = F \iff \{(\nu_X(v), \nu_Y(w_1), \ldots, \nu_Y(w_k)) \mid \langle v, w_1, \ldots, w_k\rangle \lhd p\}\\ = \{(I, J_1, \ldots, J_k) \in \beta_X \times \beta_Y^* \mid I \Subset F^{\Leftarrow}(J_1 \cup \cdots \cup J_k)\}.\end{aligned}$$

The following result is a direct corollary of Theorem 3.70.

Corollary 3.72. *Let* $(X, \tau_X, \beta_X, \nu_X)$ *be an effectively locally compact space, and* $(Y, \tau_Y, \sigma_Y, \nu_Y)$ *be a sub-effective topological space. Then, the standard representation* γ *of* $\mathcal{C}(X; Y)$ *is equivalent to the Isbell representation* γ'.

Remark 3.73. If (X, τ) is a locally compact Hausdorff space, then we can choose a basis β such that $\bar{I}$ is compact for all $I \in \beta$. Then, the effective covering property can be written as $\{I, J_1, \ldots, J_k \mid \bar{I} \subset \bigcup_{i=1}^k J_i\}$ is recursively-enumerable. Further, the effective disjointness property that can be written as $\{I_1, I_2 \in \beta \times \beta \mid \bar{I}_1 \times \bar{I}_2 = \emptyset\}$ is recursively-enumerable. This recovers the definitions of Brattka and Presser (2003).

3.8 Metric spaces

We now turn to the metric spaces.

Definition 3.74 (Computable metric space). *A computable metric space is a tuple* (X, d, ξ, α), *where* (X, d) *is a metric space,* $\xi : \Sigma^* \rightharpoonup X$ *encodes a countable dense subset of* X, *and* $\alpha : \Sigma^* \times \Sigma^* \times \mathbb{N} \rightharpoonup \mathbb{Q}$ *is such that* $|d(\xi(w_1), \xi(w_2)) - \alpha(w_1, w_2, n)| < 2^{-n}$.

The standard representation of a computable metric space is the Cauchy representation, *that is,*

$$\delta(\langle w_1, w_2, \ldots\rangle) = x \iff \lim_{n\to\infty} \xi(w_n) = x \text{ and } \alpha(w_m, w_n, \min(m, n) + 1) < 2^{-(\min(m,n)+1)}.$$

Theorem 3.75. *The standard representation* δ *of a computable metric space* (X, d, ξ, α) *is an admissible quotient representation, and the metric* d *is a computable function* $\mathcal{X} \times \mathcal{X} \to \mathcal{R}$.

Proof. The proof that δ is an admissible quotient representation is similar to that of Theorem 3.53 and is omitted. To show that the metric is computable, let $\delta(\langle v_1, v_2, \ldots\rangle) = x$ and $\delta(\langle w_1, w_2, \ldots\rangle) = y$. Observe that $d(\xi(v_n), x) \le 2^{-n}$ and $d(\xi(w_n), y) \le 2^{-n}$ If $\delta(\langle v_1, v_2, \ldots\rangle) = x$ we have

$$\begin{aligned}|d(x, y) - \alpha(v_{n+2}, w_{n+2}, n + 1)| &\le d(x, \xi(v_{n+2})) + d(\xi(w_{n+2}), y)\\ &\quad + \left|d(\xi(v_{n+2}), \xi(w_{n+2})) - \alpha(v_{n+2}, w_{n+2}, n + 1)\right|\\ &\le 2^{-(n+2)} + 2^{-(n+2)} + 2^{-(n+1)} \le 2^{-n}.\end{aligned}$$

Hence $\alpha(v_{n+2}, w_{n+2}, n + 1)$ is a fast-convergent Cauchy sequence with limit $d(x, y)$. □

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

We can therefore construct a complete computable metric space from a countable metric space by defining a representation on equivalence classes of fast Cauchy sequences. Note that (X, d) is a computable metric space if it is effectively separable and d is computable.

Let (X, d_X) and (Y, d_Y) be computable complete metric spaces, and Q a countable dense subset of X. An important way of defining a function $X \to Y$ is to define it on Q and extend over X by continuity; if a continuous extension exists, then it is unique.

Following O'Connor and Spitters (2010) if f is computable, and uniformly continuous on Q with a computable global modulus of continuity δ, then f extends to a computable function over X. However, many important functions, such as $x \mapsto x^2$, are only *locally* uniformly continuous. In order to computably extend such functions, we need to weaken the requirement of global uniform continuity. However, it does not suffice that f is locally uniformly continuous on Q, as the following example shows:

Example 3.76. *Take $Q = \mathbb{Q}$, $X = \mathbb{R}$ and cover $U_0 = \{q \in \mathbb{Q} \mid q < 0 \vee q^2 < 2\}$ and $U_1 = \{q \in \mathbb{Q} \mid q > 0 \wedge q^2 > 0\}$, and the function $f(x) = 0$ for $x \in U_0$ and $f(x) = 1$ for $x \in U_1$. Then, f is locally uniformly continuous on $\mathbb{Q}$, as it is constant over U_0 and over U_1, but does not extend continuously over $\mathbb{R}$ at $\sqrt{2}$.*

Definition 3.77 (Completion locally uniform continuous). *Let (X, d_X) and (Y, d_Y) be complete metric spaces, and Q a countable dense subset of X. We say $f : Q \to Y$ is* completion locally uniformly continuous *if there exists a countable open cover $\{U_0, U_1, \ldots\}$ of X and moduli of continuity $\delta_n : \mathbb{Q}^+ \to \mathbb{Q}^+$ such that for all n, f is δ_n-continuous on $Q \cap U_n$.*

Proposition 3.78 (Extension of functions). *Suppose $f : Q \to Y$ is computable and is completion locally uniformly continuous over computable open sets U_n with computable moduli of continuity δ_n. Then, f extends to a unique continuous function over X, and this extension is computable.*

Proof. Given $f : \mathcal{Q} \to \mathcal{Y}$, the sets $U_n : \mathcal{O}(\mathcal{X})$ and the moduli of continuity $\delta_n : \mathcal{Q}^+ \to \mathcal{Q}^+$, we can compute f at any $x \in X$ to arbitrary precision $\epsilon \in \mathbb{Q}^+$ by finding some $U_n \ni x$, choosing $q \in Q \cap U_n$ such that $d(q, x) < \delta_n(\epsilon)$, and approximating $f(x) \approx f(q)$ with $d(f(q), f(x)) < \epsilon$. □

An important class of computable metric spaces are the L^p-spaces. In order to study these, we first need to define integrals. A simple approach is to define the integral by extension of locally constant functions and relying on monotonicity:

Definition 3.79. *The* integral *is the linear function $(\mathbb{R} \to \mathbb{R}) \times \mathbb{R}^2 \to \mathbb{R}$ written $(f, a, b) \mapsto \int_a^b f(x)dx$ satisfying:*

(a) $f \mapsto \int_a^b f(x)dx$ is linear: $\int_a^b [\alpha_1 f_1 + \alpha_2 f_2](x)dx = \alpha_1 \int_a^b f_1(x)dx + \alpha_2 \int_a^b f_2(x)dx$.
(b) If f is constant c on $[a, b]$, then $\int_a^b f(x)\,dx = c(b - a)$.
(c) If $f(x) \le g(x)$ on $[a, b]$ and $a \le b$, then $\int_a^b f(x)\,dx \le \int_a^b g(x)dx$.
(d) $\int_a^b f(x)dx = \int_a^c f(x)dx + \int_c^b f(x)dx$.

It is straightforward to show that the integral is well-defined for continuous functions and is computable:

Theorem 3.80. *The integral operator $(\mathbb{R} \to \mathbb{R}) \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ taking $(f, a, b) \mapsto \int_a^b f(x)\,dx$ is computable.*

Sketch of proof. Subdivide the interval $[a, b]$ into $n = 2^m$ equal pieces with boundaries $t_0, t_1, \ldots, t_n$. Since $[t_k, t_{k+1}]$ is compact, $f([t_k, t_{k+1}]) = [\underline{y}_k, \overline{y}_k]$ is computable as a compact subset of $\mathbb{R}$. Hence, $S_n = \sum_{k=0}^{n-1} (t_{k+1} - t_k)[\underline{y}_k, \overline{y}_k]$ is computable as a compact subset of $\mathbb{R}$. The result

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

follows since $\{\int_a^b f(x)dx\} = \bigcap_{m=0}^{\infty} S_{2^m}$ (using classical continuity arguments), so is computable as a compact set, and $\{\int_a^b f(x)dx\} \subset U \iff \int_a^b f(x)dx \in U$ for open U. □

The result extends in a natural way to Euclidean space, and also to generally measure spaces; see Collins (2014) for details.

Definition 3.81 (Integrable functions). *We start with a notion of* integral *of functions* $I : (\mathcal{X} \to \mathcal{R}) \rightharpoonup \mathcal{R}$ *which is a computable linear functional defined on functions of compact support, and satisfying* $f \geq 0 \implies I(f) \geq 0$. *The* L^p *space is defined as the completion of the functions of compact support under the metric* $d_p(f_1, f_2) = (I(|f_1 - f_2|^p))^{1/p}$.

Since elements of L^p need not be continuous, evaluation on L^p is not a computable operator. However, we can show that the integral extends computably over L^1, since the integral is Lipschitz:

$$|I(f_1) - I(f_2)| = |I(f_1 - f_2)| \leq I(|f_1 - f_2|) = d_1(f_1, f_2).$$

We can also show standard properties of L^p spaces. For example, if $f : L^1$ and g is uniformly bounded with (known) bound B, then

$$\begin{aligned} |I(f \times g) - I(f_n \times g)| = |I((f - f_n) \times g)| &\leq I(|(f - f_n) \times g|) \leq I(|f - f_n| \times B) \\ &\leq B \times I(|f - f_n|) \leq 2^{-n} B. \end{aligned}$$

4. Classical Topology

In this section, we relate the computability theory developed in Section 3 to concepts from classical topology. The material in this section is mostly quite technical and is not needed to actually use the type theory. However, it is useful to relate the type theory to classical mathematics. We answer the question of which spaces have an admissible quotient representation and so are amenable to the theory. We consider the various set types and give a condition (*sobriety*) under which these types are equivalent to their classical point-set definition. We also consider local-compactness properties, including the result that locally compact spaces (which are suitable state spaces of dynamical systems) have a locally compact open set type (which may be useful in control synthesis).

4.1 Sequential spaces

Since all spaces with an admissible quotient representation are sequential spaces, we begin with a brief overview of sequential spaces, and the relation between sequential spaces and topological spaces.

Lemma 4.1. *The space* Σ^ω *is a countably based sequential space.*

Proof. A countable base for Σ^ω is given by the *cylinder sets* $C_w = \{p \in \Sigma^\omega \mid \forall i < |w|,\ p_i = w_i\}$ where $w \in \Sigma^*$ and $|w|$ denotes the length of w.

Suppose U is sequentially open and $p \in U$, but $p \notin \mathrm{int}\,(U)$. Then for all n, the set $p|_n \Sigma^\omega \setminus U = \{q \notin U \mid p|_n = q|_n\}$ is nonempty. Therefore, there exists a sequence p_n such that $p_n|_n = p|_n$ and $p_n \notin U$, contradicting U being sequentially open. Hence, $p \in \mathrm{int}\,(U)$, and since $p \in U$ is arbitrary, U is open. □

Recall that a subset C of X is *compact* if any open cover of C has a finite subcover, *countably compact* if any countable open cover of C has a finite subcover, and *sequentially compact* if any sequence in C has a convergent subsequence.

The following result shows that for representable spaces, these three notions of compactness coincide. Hence, one may use whichever definition of compactness is most appropriate. The equivalence of sequential- and countable-compactness for sequential spaces was proved in Engelking (1989, Theorem 3.10.31).

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Theorem 4.2. *Let X be a topological space. Then any compact subset is countably compact, as is any sequentially compact subset. If X is a sequential space, then any countably compact set is sequentially compact. If X has a countable pseudobase, then any countably compact set is compact.*

Proof. It is immediate from the definition that any compact subset is countably compact. If C is sequentially compact, suppose $\mathcal{U} = \{U_0, U_1, \ldots\}$ is a countable collection of open sets with no finite subset covering C. Then, there exists a sequence of points x_n such that $x_n \in C \setminus \bigcup_{k=0}^{n} U_k$. Since C is sequentially compact, the sequence $\vec{x}$ has a convergent subsequence $x_{k(n)}$ with limit $x_\infty \in C$. By taking $y_n = x_{k_m}$ with $k_m \geq m$, we can construct a convergent sequence $y_n \to y_\infty$ with $y_n \in C \setminus \bigcup_{k=0}^{n} U_k$ and limit $y_\infty \in C$. Then, $y_\infty \notin \bigcup_{n=0}^{\infty} U_n$, so $\mathcal{U}$ is not an open cover of C. Taking the contrapositive, we see that any countable open cover has a finite subcover.

Let X be a sequential space, and suppose C is a countably compact subset of X. Let $\vec{x}$ be a sequence of mutually distinct points in C, define $S_n = \{x_n, x_{n+1}, \ldots\}$, $A_n = \mathrm{cl}(S_n)$ and $A_\infty = \bigcap_{n=0}^{\infty} A_n$, the set of limit points of $\vec{x}$ Let $U_n = X \setminus A_n$. Then, $\{U_n \mid n \in \mathbb{N}\}$ is a countable collection of open increasing open sets such that no U_n contains C. Since C is countably compact, $U_\infty = \bigcup_{n=0}^{\infty} U_n$ is not a cover of C, so $A_\infty \cap C \neq \emptyset$. Take $x_\infty \in A_\infty \cap C$. Then, $x_\infty \in A_n$ for all n, but $x_\infty \notin S_m$ for some m, so $S_m \neq A_m = \mathrm{cl}(S_m)$. Since S_m is not closed and X is a sequential space, S_m contains a sequence $\vec{y}$ converging to a point y_∞ not in S_m. Taking a subsequence of $\vec{y}$ yields a convergent subsequence of $\vec{x}$ with limit y_∞. So some subsequence of $\vec{x}$ has limit in C. This shows that C is sequentially compact.

To prove that if X has a countable pseudobase, then any countably compact set is compact, we use the following result:

Lemma 4.3. *Suppose X has a countable pseudobase. Then for any $\mathcal{U} \subset \mathcal{O}(X)$, there exists countable $\mathcal{V} \subset \mathcal{U}$ such that $\bigcup \mathcal{V} = \bigcup \mathcal{U}$.*

For every $x \in \bigcup \mathcal{U}$, choose $U \ni x$. Since $\mathcal{B}$ is a pseudobase, there exists $B \in \mathcal{B}$ such that $x \in B \subset U$. By the axiom of choice, there is a countable subset $\mathcal{C}$ of $\mathcal{B}$ such that $\bigcup \mathcal{C} = \bigcup U$ and for all $B \in \mathcal{C}$, there exists $U \in \mathcal{U}$ such that $B \subset U$. By the axiom of countable choice, there is a countable subset $\mathcal{V}$ of $\mathcal{U}$ such that for all $B \in \mathcal{C}$, there exists $V \in \mathcal{V}$ with $B \subset V$. By construction $\bigcup \mathcal{U} = \bigcup \mathcal{C} \subset \bigcup \mathcal{V}$, so $\bigcup \mathcal{V} = \bigcup \mathcal{U}$.

Thus if C is countably compact and $\mathcal{U}$ is a possibly uncountable open cover of C, there is a countable subcover $\mathcal{V} \subset \mathcal{U}$, hence a finite cover. □

4.2 Spaces with an admissible quotient representation

We now prove one of the most important results of computable type theory, namely a characterization of topological spaces with an admissible quotient representation. We will need the following definition:

Definition 4.4 (Sequential pseudobase). *A collection ρ of subsets of a topological space X is called a* sequential pseudobase *if for any convergent sequence $x_n \to x_\infty$ and any open set $U \ni x_\infty$, there exists $V \in \rho$ such that $V \subset U$ and $x_n \in V$ for all sufficiently large n.*

The following result combines Proposition 3.1.15 and Theorem 3 of Schröder (2002a). Due to its fundamental importance, and the fact that in the original work, the results are scattered through a sequence of lemmas, we give a simplified and self-contained proof.

Theorem 4.5 (Schröder 2002a, 2007). *Let X be a Kolmogorov topological space. Then the following are equivalent:*

(a) X has an admissible quotient representation.
(b) X is a quotient of a countably based space.
(c) X is a sequential space with a countable sequential pseudobase.

Proof.

($1 \Longrightarrow 2$) Σ^ω is a countably based space, so any subspace R is also countably based. Let $\delta : \Sigma^\omega \rightharpoonup X$ be a quotient representation, and take $R = \mathrm{dom}\,(\delta)$, so $\delta : R \to X$ is a total quotient map. Thus, X is a quotient of the countably based space R.

($2 \Longrightarrow 3$) Suppose X is a topological space, β is a countable base for X, and $q : X \to Y$ is a quotient map.

Y is a sequential space: Suppose $V \subset Y$ is sequentially open. We need to show that V is open. Since q is a quotient map, V is open if, and only if, $q^{-1}(V)$ is open. Let $x_\infty \in X$ but $x_\infty \notin \mathrm{int}\,(q^{-1}(V))$. Let $\{I_0, I_1, \ldots\} = \{I \in \beta \mid x \in I\}$. Since for any $n \in \mathbb{N}$, $\bigcap_{j=0}^{n} J_i$ is open and $x_\infty \notin \mathrm{int}\,(q^{-1}(V))$, the set $\bigcap_{j=0}^{n} J_i \setminus q^{-1}(V)$ is nonempty. Hence, we can choose a sequence x_n with $x_n \in \bigcap_{j=0}^{n} J_i \setminus q^{-1}(V)$. Since β is a base for X, for any open $U \ni x$, there exists n such that $\bigcap_{j=0}^{n} J_i \subset U$. Hence, $x_n \to x_\infty$. Then, $f(x_n) \to x_\infty$, so $f(x_n)$ is a convergent sequence in $Y \setminus V$. Since V is sequentially open, $x_\infty = \lim_{n\to\infty} f(x_n) \notin V$, so $x_\infty \notin f^{-1}(V)$. Hence $\mathrm{int}\,(q^{-1}(V)) = q^{-1}(V)$, so $q^{-1}(V)$ is open.

Y has a countable pseudobase: Let β_X be a countable base of X. Define

$$\mathcal{B}_Y = \{\mathrm{sat}\,(q(I_1 \cup \ldots \cup I_k)) \mid (I_1, \ldots, I_k) \in \beta_X^*\}. \tag{12}$$

We claim that $\mathcal{B}$ is a countable pseudobase of Y.

We first prove the following: Suppose K is a compact subset of Y, V open and $K \subset V$. Let U_n be a sequence of increasing open subsets of X such that $\bigcup_{n=0}^{\infty} U_n = q^{-1}(V)$. Then, there exists m such that $\mathrm{sat}\,(q(U_m)) \supset K$. To this end, suppose that for all n, $\mathrm{sat}\,(q(U_m)) \not\supset K$. Then, there exists a sequence of open sets $(W_n)_{n\in\mathbb{N}}$ such that for all n, $q(U_n) \subset W_n$ but $W_n \not\supset K$. Then, $U_n \subset q^{-1}(W_n)$, which implies $\bigcap_{n\geq m} q^{-1}(W_n)$. Since q is a quotient map and $q^{-1}(\bigcap_{n\geq m} W_n) = \bigcap_{n\geq m} q^{-1}(W_n)$, we see that $V_m = \bigcap_{n\geq m} W_n$ is open for any m. Since $q^{-1}(V_m) \supset U_m$, we have $\bigcup_{m=0}^{\infty} q^{-1}(V_m) = q^{-1}(V)$, from which $\bigcup_{m=0}^{\infty} V_m = V$. Since K is compact, there exists m such that $K \subset V_m$, which means $K \subset \bigcap_{n=m}^{\infty} W_n$ and hence $K \subset W_m$, a contradiction. Hence, there exists m such that $\mathrm{sat}\,(q(U_m)) \supset K$.

To complete the proof, let $y_n \to y_\infty$ with $y_\infty \in V$. Then, there exists m such that $y_n \in V$ for all $n \geq m$. Take $K = \{y_m, y_{m+1}, \ldots, y_\infty\}$. Let $(I_0, I_1, \ldots)$ be a list of all $I \in \beta$ such that $I \subset q^{-1}(V)$, and let $U_j = \bigcup_{i=0}^{j} I_j$. Hence, there exists k such that $y_n \in \mathrm{sat}\,(q(I_0 \cup \cdots \cup I_k))$ for all $n \geq m$.

($3 \Longrightarrow 1$) Let X be a sequential topological space, and $\mathcal{B} = \{B_0, B_1, \ldots\}$ be a countable pseudobase with prefix-free notation $\nu : \Sigma^* \twoheadrightarrow \mathcal{B}$. Define a function $\delta : \Sigma^\omega \rightharpoonup X$ by

$$\delta(p) = x \iff \forall w \lhd p,\ x \in \nu(w) \text{ and } \forall \text{ open } U \ni x, \exists w \lhd p,\ \nu(w) \subset U. \tag{13}$$

Here, by $w \lhd p$, we mean that $p = w_0 w_1 w_2 \cdots$ with $w_i \in \mathrm{dom}\,(\nu)$ for all i and $w = w_i$ for some i. We claim that δ is an admissible quotient representation.

δ is single-valued: If $x \neq y$, there exists open U containing exactly one of x, y. Without loss of generality, suppose $x \in U$ and $y \notin U$. Then for any name p of x, there exists $w \lhd p$ such that $x \in \nu(w) \subset U$, but for any name q of y and any w such that $\nu(w) \subset U$, we have $y \notin \nu(w)$, so $w \not\lhd q$.

δ is surjective: For any $x \in X$ and open $U \ni x$, there exists $B \in \mathcal{B}$ such that $x \in B \subset U$.

δ is continuous: Let $U \in \mathcal{O}(X)$ and $p \in \delta^{-1}(U)$ with $x = \delta(p)$. Then, there exists $B = \nu(w) \in \mathcal{B}$ such that $x \in B \subset U$, and then $w \lhd p$. There is an open neighborhood W of p in Σ^ω such that $w \lhd q$ for all $q \in W$, so for any $q \in W \cap \mathrm{dom}\,(\delta)$, we have $\delta(q) \in B \subset U$. Hence, $p \in W \subset \delta^{-1}(U)$, so δ is continuous.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

δ is a quotient: Suppose that $f : X \to Y$ and $\phi = f \circ \delta$ is continuous. Let V be open in Y and $U = f^{-1}(V)$ with $x_\infty \in U$. Let p_∞ be a δ-name of x_∞ so $y_\infty = \phi(p_\infty) \in V$. Let p_n be a sequence in dom (δ) such that $p_n \to p_\infty$, and let $x_n = \delta(p_n)$. Since δ is continuous and X is a sequential space, $x_n \to x_\infty$ as $n \to \infty$. Since ϕ is continuous, there exists N such that $\phi(p_n) \in V$ for all $n \geq N$. Then for $n \geq N$, we have $f(x_n) = f(\delta(p_n)) = \phi(p_n) \in V$, so $x_n \in f^{-1}(V)$. Hence, $f^{-1}(V)$ is sequentially open, so is open.

δ is admissible: Suppose $\phi : \Sigma^\omega \rightharpoonup X$ is a continuous partial function. Define a function $\tilde{\eta} : \Sigma^\omega \rightharpoonup \Sigma^\omega$ such that if v_1 is a prefix of v_2, then $\tilde{\eta}(v_1)$ is a prefix of $\tilde{\eta}(v_2)$, and that $w \lhd \eta(v)$ if, and only if, $\phi(p) \in \nu(w)$ whenever $p \in v\Sigma^\omega$ and $|w| \leq |v|$, the latter condition being to ensure that given $v \in \Sigma^*$, we do not obtain infinitely many w such that $\phi(v\Sigma^\omega) \subset \nu(w)$. Define $\eta(p) = \lim_{n\to\infty} \tilde{\eta}(p|_n)$. By padding $\tilde{\eta}(v)$ with names of the empty set if necessary, we can ensure that $\eta : \Sigma^\omega \rightharpoonup \Sigma^\omega$. Clearly η is continuous.

It remains to show that for all $p \in \text{dom}\,(\phi)$, $\eta(p)$ is a δ-name of $\phi(p)$. First, if $w \lhd \eta(p)$, then $\phi(p) \in \nu(w)$. It remains to show that if $\phi(p) \in U$, there exists $w \lhd \eta(p)$ such that $x \in \nu(w) \subset U$. Let $v_n = (p_0, p_1, \ldots, p_n)$. If $\phi(p) \in U$, then $p \in \phi^{-1}(U)$ so there exists n such that $\phi(v_n\Sigma^\omega \cap \text{dom}\,(\phi)) \subset U$. Let $\mathcal{B}_{x,U} = \{B \in \mathcal{B} \mid x \in B \subset U$. Suppose for all $B_m \in \mathcal{B}_{x,U}$, there is a sequence $(q_{m,n})_{n\in\mathbb{N}} \in \text{dom}\,(\phi)$ such that $(q_{m,n})_i = p_i$ for $i \leq n$ and $q_{m,n} \notin B_m$ for all B. Then, there is a sequence $(r_k)_{k\in\mathbb{N}}$ such that $r_k \in \text{dom}\,(\phi)$, $(r_k)_i = p_i$ for all $i \leq k$, and for each $B \in \mathcal{B}_{x,U}$, $r_k \notin B$ infinitely often. Then, $r_k \to p$ as $k \to \infty$, so $\phi(r_k) \to x$ but there is no $B \in \mathcal{B}_{x,U}$ such that eventually $\phi(r_k) \in B$, contradicting the fact that $\mathcal{B}$ is a sequential pseudobase. Hence, there exists $B = \nu(w) \in \mathcal{B}_{x,U}$ and $n \in \mathbb{N}$ such that $\phi(v_n\Sigma^\omega) \subset B$. Then, we have $w \lhd p$ as required. □

Remark 4.6. It *not* true that convergent sequences lift under topological quotients. For example, the disjoint union of $[-1, 0]$ and $[0, +1]$ quotients onto $[-1, +1]$, but any sequence in $[-1, +1]$ with infinitely many positive and negative elements does not lift to the base space.

The construction of the pseudobase for the topological quotient is Schröder (2002a, Lemma 3.1.12).

Remark 4.7. It is not true in general that if $q : Y \to X$ is a quotient map and δ_Y is an admissible quotient representation of Y, then $q \circ \delta_Y$ is an admissible quotient representation of X. See Dahlgren (2007).

Remark 4.8. There are non-topological *limit spaces* which have an admissible representation.

Remark 4.9. To the best of our knowledge, it is an open question as to whether every separable sequential space satisfies the equivalent conditions of Theorem 4.5.

4.3 The Scott topology on open sets

We now give a more detailed study of the open set type introduced by Definition 3.23.

We first study the induced topology on $\mathcal{O}(X)$. Recall that $U_n \nearrow U_\infty$ means $U_{n+1} \supset U_n$ for all n and $\bigcup_{n=0}^\infty U_n = U_\infty$.

Definition 4.10 (Scott topology). *Let X be a topological space. A collection $\mathcal{U}$ of open subsets of X is* Scott open *if:*

1. *$U \in \mathcal{U}$, $V \in \mathcal{O}(X)$, and $U \subset V$ imply $V \in \mathcal{U}$, and*
2. *whenever $\mathcal{V} \subset \mathcal{O}(X)$ and $\bigcup \mathcal{V} \in \mathcal{U}$, there is a finite subset $\{V_1, \ldots, V_n\}$ of $\mathcal{V}$ such that $\bigcup_{k=1}^n V_k \in \mathcal{U}$.*

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

*A collection $\mathcal{U}$ of open sets is ω-*Scott open *if 1. above and*

2'. *whenever $\mathcal{V} \subset \mathcal{O}(X)$ is countable and $\bigcup \mathcal{V} \in \mathcal{U}$, there is a finite subset $\{V_1, \ldots, V_n\}$ of $\mathcal{V}$ such that $\bigcup_{k=1}^{n} V_k \in \mathcal{U}$, or equivalently*
2''. *whenever $U_n \nearrow U_\infty$ with $U_\infty \in \mathcal{U}$, then there exists n such that $U_n \in \mathcal{U}$.*

It is straightforward to show that the collection of Scott open sets and ω-Scott open sets are topologies on $\mathcal{O}(X)$, called the Scott topology *and ω-*Scott topology, *respectively.*

In general, the ω-Scott topology is finer than the Scott topology. However, the following result shows that the ω-Scott topology coincides with the usual Scott topology if X has a countable pseudobase; in particular, if X is a quotient of a countably based space.

Proposition 4.11. *Let X be a topological space with a countable pseudobase. Then, a set is ω-Scott open if, and only if, it is Scott open.*

The proof is immediate from Theorem 4.2. we henceforth use 2. and 2'. interchangeably for defining Scott open sets when working in a space with a countable pseudobase, in particular, in a quotient of a countably based space.

The following properties of the ω-Scott topology are elementary.

Lemma 4.12. *Let X be a topological space.*

(a) If $U_n \nearrow U_\infty$ with $U_\infty \supset V$, then $U_n \to V$ in the ω-Scott topology.
(b) If C is countably compact, then $\{U \mid C \subset U\}$ is open in the ω-Scott topology.
(c) The set of open sets with the ω-Scott topology is a sequential space.

Proof.

(a) Immediate from the definition of the ω-Scott topology.
(b) Suppose $U_n \nearrow U_\infty$ with $C \subset U_\infty$. Then, $\{U_n \mid n \in \mathbb{N}\}$ is a countable open cover of C so has a finite subcover $\{U_1, \ldots, U_n\}$. Then, $C \subset U_n$.
(c) Suppose $\mathcal{W} \subset \mathcal{O}(X)$, and $\mathcal{W}$ is sequentially open in the ω-Scott topology. Then, if $U \in \mathcal{W}$ and $V \supset U$, then since $V \to U$ in the ω-Scott topology, we have $\mathcal{W} \ni V$. Further, if $U_n \nearrow U_\infty$ with $U_\infty \in \mathcal{W}$, then $U_n \to U_\infty$ in the ω-Scott topology, so there exists N such that $U_n \in \mathcal{W}$ for $n \geq N$ since $\mathcal{W}$ is sequentially open. Hence the ω-Scott topology is sequential. □

The following theorem shows that the Scott topology (and by Proposition 4.11 the ω-Scott topology) is an explicit description of the topology on $\mathcal{O}(X)$ induced by the representation of the topological type $\mathbb{S}^{\mathcal{X}}$.

Theorem 4.13. *Let $\mathcal{X} = (X, [\delta])$ be a topological type. A subset of $\mathcal{O}(X)$ is open in the topology induced by the representation $\delta_{\mathcal{X} \to \mathbb{S}}$ if, and only if, it is Scott open.*

Proof. It suffices to show that the Scott topology is the coarsest topology such that the inclusion map $X \times \mathcal{O}(X) \to \mathbb{S}$, $(x, U) \mapsto \top \iff x \in U$ is sequentially continuous.

Suppose $x_n \to x_\infty$ in X, and $U_n \to U_\infty$ in the Scott topology with $x_\infty \in U_\infty$. Since $x_\infty \in U_\infty$, there exists N such that $x_n \in U_\infty$ for all $n \geq N$. Consider $\mathcal{V} = \{V \in \mathcal{O}(X) \mid \{x_N, x_{N+1}, \ldots, x_\infty\} \subset V\}$. The set $\mathcal{V}$ is Scott open, since $\{x_n, x_{n+1}, \ldots, x_\infty\}$ is compact. Hence, there exists M such that $U_m \in \mathcal{V}$ for all $m \geq M$. Then, for $k \geq \max(M, N)$, we have $\{x_k, \ldots, x_\infty\} \subset U_k$, so $x_k \in U_k$. Thus, inclusion is sequentially continuous using the Scott topology on $\mathcal{O}(X)$.

Suppose $\mathcal{T}$ is a sequential topology on $\mathcal{O}(X)$ such that inclusion $X \times \mathcal{O}(X) \to \mathbb{S}$ is sequentially continuous. Suppose $\mathcal{U}$ is Scott-open but not $\mathcal{T}$-open. Since $\mathcal{T}$ is a sequential topology, $\mathcal{U}$ is not $\mathcal{T}$-sequentially-open. Hence, there exists a sequence $U_n \to_{\mathcal{T}} U_\infty$ with $U_\infty \in \mathcal{U}$ but $U_n \notin \mathcal{U}$

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

for any $n \in \mathbb{N}$. Since $\mathcal{U}$ is Scott-open, we have $\bigcup_{n=0}^{\infty} U_n \not\supset U_\infty$. Let $x \in U_\infty \setminus \bigcup_{n=0}^{\infty} U_n$. Then, $x \in U_\infty$ but $x \notin U_n$ for any n, contradicting inclusion being sequentially continuous with respect to $\mathcal{T}$. Hence, the Scott topology is coarser than any other topology making inclusion sequentially continuous. □

For any quotient of a countably based space X, we henceforth assume the Scott topology on $\mathcal{O}(X)$, so $\mathcal{O}(\mathcal{O}(X))$ consists of Scott-open subsets of $\mathcal{O}(X)$.

4.4 Sober spaces

We now consider the closed and compact subsets of X, and their relationships with the overt and compact set types $\mathcal{V}(\mathcal{X})$ and $\mathcal{K}(\mathcal{X})$ given by Definition 3.24 as subtypes of $\mathcal{O}(\mathcal{O}(\mathcal{X}))$, and the consequences for $\mathcal{V}(\mathcal{X})$ and $\mathcal{K}(\mathcal{X})$. This is important when studying systems, since we need to know when the classical notion of (compact) set actually corresponds to the effective notion of Definition 3.24.

In particular, we are interested in spaces X for which any element of $\mathcal{O}(\mathcal{O}(\mathcal{X}))$ which satisfies both (7) and (8) arises from a singleton set. In other words, if $Q : \mathcal{O}(\mathcal{X}) \to \mathcal{S}$ satisfies

$$Q(\emptyset) = \uparrow, \quad Q(U_1 \cup U_2) \iff Q(U_1) \vee Q(U_2),$$

$$Q(X) = \top \text{ and } Q(U_1 \cap U_2) \iff Q(U_1) \wedge Q(U_2).$$

then there exists $x \in X$ such that $Q(U) \iff x \in U$.

Definition 4.14 (Filters and cofilters). *A subset $\mathcal{D}$ of $\mathcal{P}(X)$ is* directed *if $S \in \mathcal{D}$ and $T \supset S$ implies $T \in \mathcal{D}$. A directed subset $\mathcal{F}$ of $\mathcal{P}(X)$ is a* filter *if $S_1 \in \mathcal{F} \wedge S_2 \in \mathcal{F} \implies S_1 \cap S_2 \in \mathcal{F}$. A directed subset $\mathcal{F}$ of $\mathcal{P}(X)$ is a* cofilter *if $\emptyset \notin \mathcal{F}$ and $S_1 \cup S_2 \in \mathcal{F} \implies S_1 \in \mathcal{F} \vee S_2 \in \mathcal{F}$. A directed subset of $\mathcal{P}(X)$ is an* ultrafilter *if it is both a filter and a cofilter. A (co)filter which is a Scott-open subset of $\mathcal{O}(X)$ is a* Scott-open (co)filter.

Remark 4.15. The condition $\emptyset \notin \mathcal{F}$ in the definition of a cofilter could have been omitted, but since for any set $S \subset X$, we have $S \not\between \emptyset$, such a cofilter could never arise as the collection of sets intersecting some set S.

If S is any set, then $\{U \in \mathcal{O}(X) \mid S \subset U\}$ is a filter, and $\{U \in \mathcal{O}(X) \mid S \between U\}$ is a cofilter. If $x \in X$ is a point, then we have $\{x\} \subset U \iff \{x\} \between U \iff x \in U$, so the set of neighborhoods of x is an ultrafilter and is easily shown to be open in the Scott topology. Conversely, if X is a T_0 space, and S is a set such that for all open sets U, $S \subset U \iff S \between U$, then S is a singleton. Hence, the set of Scott-open ultrafilters are in some sense "point-like". However, as the following example shows, not all Scott-open ultrafilters are the set of neighborhoods of a point.

Example 4.16. *Let $X = (\mathbb{Q}, \tau_<)$, where $\tau_< = \{(-\infty, a) \cap \mathbb{Q} \mid a \in \mathbb{R}\}$. In other words, X is the restriction of the reals with the topology of lower convergence to the subspace of the rationals. Take $r \notin \mathbb{Q}$, and let $\mathcal{U} = \{(-\infty, a) \mid a > r\}$. Then, $\mathcal{U}$ is Scott-open, since if $U_n = (-\infty, a_n)$ with $U_n \nearrow U_\infty$ with $U_\infty = (-\infty, a_\infty) \in \mathcal{U}$, then $a_n \nearrow a_\infty > r$, so $a_n > r$ for some n. Further, it is clear that $\mathcal{U}$ is an ultrafilter. However, there is no $q \in \mathbb{Q}$ such that $\mathcal{U} = \{(-\infty, a) \mid a > q\}$, so $\mathcal{U}$ is not the set of neighborhoods of a point.*

The topology $\tau_<$ on $\mathbb{Q}$ is equivalent to the topology $\tau_<$ on $\mathbb{R}$, in the sense that there is a bijection between open sets preserving unions and intersections. This shows that a set of points of a topological space cannot be recovered from the lattice of open sets. However, if we view the Scott-open ultrafilters as providing a canonical set of points, we do obtain a unique space. We call a space *sober* if every Scott-open ultrafilter is the closure of a (unique) point. Thus, a sober space "has enough points" in the sense that any "point-like" collection of open sets corresponds to a real point.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Definition 4.17 (Sober space). *A topological space X is* sober *if any Scott-open ultrafilter of X is the set of neighborhoods of a point.*

In the literature (see Gierz et al. 1980; Hofmann and Mislove 1981), an alternative definition of *sober* space is sometimes used. A closed set A is *irreducible* if whenever $A = A_1 \cup A_2$ where A_1, A_2 are closed sets, either $A_1 = A$ or $A_2 = A$. A topological space X is *(classically) sober* if every non-empty irreducible closed subset is the closure of a point.

We now show that the definition of sober space used here coincides with the classical definition.

Theorem 4.18. *Let X be a topological space. Then the following are equivalent:*

(a) Any Scott-open ultrafilter of X is the set of neighborhoods of a point.
(b) Any irreducible closed subset of X is the closure of a point.

Proof. Let A be an irreducible closed set, and $\mathcal{U} = \{U \in \mathcal{O}(X) \mid A \between U\}$. If A is disjoint from $U_1 \cap U_2$, then $A = A \setminus (U_1 \cap U_2) = (A \setminus U_1) \cup (A \setminus U_2)$, so either $A \setminus U_1 = A$ or $A \setminus U_2 = A$ since A is irreducible, and hence A is disjoint from either U_1 or U_2. Hence if $A \between U_1$ and $A \between U_2$, then $A \between (U_1 \cap U_2)$. Hence, $\mathcal{U}$ is an ultrafilter. Further, if $A \between \bigcup \mathcal{V}$ with $\bigcup \mathcal{V} \in \mathcal{U}$, there exists $x \in A$ such that $x \in \bigcup \mathcal{V}$, and then $x \in V$ for some $V \in \mathcal{V}$. Hence, $\mathcal{U}$ is Scott-open. Therefore, if any open ultrafilter is the neighborhood filter of a point, we have $\mathcal{U} = \{U \in \mathcal{O}(X) \mid x \in U\}$. Then, $\mathrm{cl}\{x\} \between U \iff \{x\} \between U \iff x \in U \iff U \in \mathcal{U} \iff A \between U$, so $\mathrm{cl}\{x\} = A$.

Let $\mathcal{U}$ be a Scott-open ultrafilter, and let $V = \bigcup\{U \mid U \notin \mathcal{U}\}$. Suppose $V \in \mathcal{U}$. Then, since $\mathcal{U}$ is Scott open, we have $V_1 \cup \cdots \cup V_k \in \mathcal{U}$ with $V_i \notin \mathcal{U}$ for all i. Taking k minimal, we have $V_1 \cup \cdots \cup V_{k-1} \notin \mathcal{U}$ and $V_k \notin \mathcal{U}$, contradicting $\mathcal{U}$ being an cofilter. Hence, $V \notin \mathcal{U}$. Suppose $V = V_1 \cap V_2$ with $V_1, V_2 \neq V$. Then, each V_i is a strict superset of V, so $V_1, V_2 \in \mathcal{U}$ by definition of V. But, then $V = V_1 \cap V_2 \in \mathcal{U}$ since $\mathcal{U}$ is a filter, again a contradiction. Hence, $A = X \setminus V$ is an irreducible closed set. Therefore, if every irreducible closed set is the closure of a point, we have $X \setminus \bigcup\{U \mid U \notin \mathcal{U}\} = \mathrm{cl}\{x\}$. Then, $U \in \mathcal{U} \iff U \not\subset V \iff \mathrm{cl}\{x\} \between U \iff x \in U$. $\square$

Remark 4.19. A space is *supersober* if the set of limit points of each ultrafilter on X is either empty or a singleton closure. Theorem 4.18 does not show that any supersober space is sober, since it refers only to Scott-open ultrafilters, whereas the definition of a supersober space refers to arbitrary ultrafilters.

We now give two results which relate the classical closed and compact subsets of X with $\mathcal{V}(X)$ and $\mathcal{K}(X)$ considered as subsets of $\mathcal{O}(\mathcal{O}(X))$.

Theorem 4.20. *Let X be a T_0 topological space. Then there is a bijection between closed subsets of X and Scott-open cofilters in $\mathcal{O}(X)$.*

Proof. Given $A \in \mathcal{A}(X)$, define $\mathcal{F} = \{U \in \mathcal{O}(X) \mid A \between U\}$. Clearly $\mathcal{F}$ is a Scott-open cofilter. Conversely, given a Scott-open cofilter $\mathcal{F}$, define $A = \{x \in X \mid x \in U \in \mathcal{O}(X) \implies U \in \mathcal{F}\}$. If $x \notin A$, there exists $U \in \mathcal{O}(X)$ such that $U \notin \mathcal{F}$. But, then $A \cap U = \emptyset$. Hence, A is closed. Finally, if $A \in \mathcal{A}(X)$ and $\mathcal{F} = \{U \in \mathcal{O}(X) \mid A \between U\}$, define $B = \{x \in X \mid x \in U \in \mathcal{O}(X) \implies U \in \mathcal{F}\}$. If $x \in A$, then for any $U \in \mathcal{O}(X)$ with $x \in U$, we have $x \in A \cap U$, so $A \between U$ and $U \in \mathcal{F}$, hence $x \in B$. If $x \notin A$, then taking $V = X \setminus A$ we have $x \in V$ but $x \notin \mathcal{F}$, so $x \notin B$. $\square$

Recall that the *saturation* of a set S is $\mathrm{sat}\,(S) = \bigcap\{U \in \mathcal{O}(X) \mid S \subset U\}$, and a set S is *saturated* if $S = \mathrm{sat}\,(S)$. The next result is due to Hofmann and Mislove (1981), which establishes an isomorphism between compact saturated sets and open filters. We give a direct proof due to Keimel and Paseka (1994).

Theorem 4.21 (Hofmann-Mislove). *Let X be a sober space. Then, there is a bijection between saturated compact subsets of X and Scott-open filters in $\mathcal{O}(X)$.*

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Proof. Let C be a compact subset of X and $\mathcal{U} = \{U \in \mathcal{O}(X) \mid C \subset U\}$. Then, $\mathcal{U}$ is open since C is compact, and is clearly a filter.

Let $\mathcal{U}$ be an open filter and $C = \bigcap \mathcal{U}$. Suppose $C \subset V$ but $V \notin \mathcal{U}$. Then by Zorn's lemma, there is an open set W containing V which is maximal among all open sets not in $\mathcal{U}$. If $W = W_1 \cap W_2$ with $W_1, W_2 \neq W$, then we would have $W_1, W_2 \in \mathcal{U}$ and so $W_1 \cap W_2 \in \mathcal{W}$, a contradiction. Hence, $X \setminus W$ is irreducible, so $X \setminus W = \mathrm{cl}\{x\}$ for some $x \in X$. Any open set not containing x is therefore a subset of W. Hence, $x \in U$ for all $U \in \mathcal{U}$ so $x \in C$, but $x \notin W$ and $C \subset W$, a contradiction. Thus, if $C \subset U$ then $U \in \mathcal{U}$.

Now let $\mathcal{V}$ be an open cover of C, so $C \subset \bigcup \mathcal{V}$. Then $\bigcup \mathcal{V} \in \mathcal{U}$, and since $\mathcal{U}$ is open, there exists $V_1, \ldots, V_k \in \mathcal{V}$ such that $\bigcup_{i=1}^{k} V_i \in \mathcal{U}$, so $C \subset \bigcup_{i=1}^{k} V_i$. Thus, C is compact. □

4.5 Core compact spaces

We now consider the property of effective local compactness, which is needed for many system-theoretic properties. Classically, a weaker version of local compactness is that of *core compactness*. We first show (Theorem 4.27) that any sober core-compact space is locally compact. This justifies restricting to local compactness in the computability theory. We then prove (Theorem 4.28) that any core-compact quotient of a countably based space is countably based. In view of Theorem 3.66, this shows that countable products of compact and locally compact representable spaces have the effective Tychonoff property, which will be important when working with trajectories of dynamic systems. Finally, we show that if X is core-compact, then its open sets are locally compact. This allows us, in principle, to work with systems over subsets of a space.

Much of the material in this sections is based on the work of Escardo and others (Escardó and Heckmann 2002; Escardó et al. 2004), which is in turn based on Hofmann and Lawson (1978), Gierz et al. (1980).

For core-compact spaces, we have an alternative representation of open sets; instead of denoting an open set as a countable union of basic open sets, we denote it by sets which are "compactly contained" in U. Recall from Notation 3.48 that $U \Subset V$ if every open cover of V has a finite subcover of U, and $\Uparrow U = \{V \in \mathcal{O}(X) \mid U \Subset V\}$. Note that if X has a countable pseudobase, then $U \Subset V$ if any monotone sequence $V_n \nearrow V$ has $V_n \supset U$ for some n. We first give some elementary properties of the relation $\Subset$.

Lemma 4.22. *Let X be a topological space and $U, V, W \in \mathcal{O}(X)$. Then:*

(a) $U \Subset V \implies U \subset V$.
(b) $U \subset U' \Subset V' \subset V \implies U \Subset V$.
(c) $U \Subset W \wedge V \Subset W \implies U \cup V \Subset W$

Note that it is not true in general that $U \Subset V$ and $U \Subset W$ imply $U \Subset V \cap W$ (though we shall see that this does hold in a sober core-compact space).

Definition 4.23 (Core compact). *A topological space X is* core-compact *(Hofmann and Lawson 1978) if, for every open set V and every $x \in V$, there exists an open set U such that $x \in U$ and $U \Subset V$.*

If X is core-compact, then for any $W \in \mathcal{O}(X)$, we have $W = \bigcup\{V \in \mathcal{O}(X) \mid V \Subset W\}$.

The following result is Escardó and Heckmann (2002, Lemma 5.2):

Lemma 4.24. *Let X be a core-compact space.*

(a) For any $U \in \mathcal{O}(X)$, the set $\Uparrow U = \{V \in \mathcal{O}(X) \mid U \Subset V\}$ is Scott open.
(b) If $\mathcal{Q} \subset \mathcal{O}(X)$ is Scott open and $V \in \mathcal{Q}$ then $U \Subset V$ for some $U \in \mathcal{Q}$.
(c) The sets $\Uparrow U$ for $U \in \mathcal{O}(X)$ form a base of the Scott topology of $\mathcal{O}(X)$.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Proof.

(a) If $V \in \Uparrow U$ and $V \subset W$, then $U \Subset V \subset W$ so $U \Subset W$ and hence $W \in \Uparrow U$. If $W \in \Uparrow U$ then there exists $V \in \mathcal{O}(X)$ with $U \Subset V \Subset W$, so $V \in \Uparrow U$. Hence, every open cover of a member W of $\Uparrow U$ has a finite subcover of a member V of $\Uparrow U$.
(b) The open set V is the union of the open sets $U \Subset V$, and such open sets are closed under the formation of finite unions.
(c) An immediate consequence of (2) and (3). □

The following lemma shows that the relation $U \Subset W$ can be interpolated.

Lemma 4.25 (Interpolation lemma). *Suppose X is core-compact and $U \Subset W$. Then, there exists $V \in \mathcal{O}(X)$ with $U \Subset V \Subset W$.*

Proof. The open set W is the union of the open sets $V \Subset W$, and each open set $V \Subset W$ is the union of the open sets $T \Subset V$. Hence, W is the union of the collection $\mathcal{T} = \{T \in \mathcal{O}(X) \mid \exists V \in \mathcal{O}(X),\ T \Subset V \Subset X\}$. For $T_1, T_2 \in \mathcal{T}$, we have $T_1 \cup T_2 \Subset V_1 \cup V_2 \Subset W$, so $\mathcal{T}$ is closed under finite union. Hence, there exists $T \in \mathcal{T}$ with $U \subset T$. Taking the corresponding V, we have $U \subset T \Subset V \Subset W$ and hence $U \Subset V \Subset W$. □

The following result shows that we can find a Scott-open filter $\mathcal{V}$ interpolating $U \Subset W$.

Lemma 4.26. *Suppose X is a core-compact space and $U \Subset W$. Then, there exists a Scott open filter $\mathcal{V}$ such that $U \subset \bigcap \mathcal{V}$ and $W \in \mathcal{V}$.*

Proof. Recursively construct a sequence of open sets V_n such that $V_0 = V$ and for all n, $U \Subset V_{n+1} \Subset V_n$. Define $\mathcal{V} = \{W \in \mathcal{O}(X) \mid \exists n,\ \ V_n \subset W\}$. Then if $W \in \mathcal{V}$ and $W' \supset W$, we have $V_n \subset W \subset W'$ for some n, so $W' \in \mathcal{V}$. Further, if $W_1, W_2 \in \mathcal{V}$, then there exists n_1, n_2 such that $V_{n_1} \subset W_1$ and $V_{n_2} \subset W_2$. Let $n = \max(n_1, n_2)$, so $V_n \Subset V_{n_1} \subset W_1$ and $V_n \Subset V_{n_2} \subset W_2$, so $V_n \subset W_1 \cap W_2$, so $W_1 \cap W_2 \in K$. Finally, if $\mathcal{W} \subset \mathcal{O}(X)$ such that $\bigcup \mathcal{W} \in \mathcal{V}$, then there exists n such that $\bigcup \mathcal{W} \supset V_n$, so $V_{n+1} \Subset \bigcup \mathcal{W}$ and hence there is a finite subset $\{W_1, \ldots, W_m\}$ of $\mathcal{W}$ such that $\bigcup_{j=1}^{m} W_j \supset V_{n+1}$. Hence, $\mathcal{V}$ is a Scott-open filter. □

The following result shows that core-compactness generalizes the classical notion of local compactness, and that for sober space, local compactness is equivalent to core-compactness. The proof that any sober core-compact space is locally compact is due to Hofmann and Lawson (1978).

Theorem 4.27. *Any locally compact space X is core-compact, and any sober core-compact space is locally compact.*

Proof. If X is locally compact, then for any open set V and any $x \in V$, there exists open U and compact K such that $x \in U$, $U \subset K$ and $K \subset V$. Then any open cover of V has a finite subcover of K and hence of U, so $U \Subset V$.

Conversely, suppose X is a sober core-compact space, that W is open and $x \in W$. Since X is core-compact, there exists an open set U such that $x \in U$ and $U \Subset V$. By Lemma 4.26, there exists a Scott open filter $\mathcal{V}$ such that $U \subset \bigcap \mathcal{V}$ and $W \in \mathcal{V}$. By Theorem 4.21, the set $K = \bigcap \mathcal{V}$ is compact. Hence, $x \in U \subset K \subset W$. □

The following result is Escardó et al. (2004, Corollary 6.11). We give a direct proof.

Theorem 4.28. *If a core-compact space is a quotient of a countably based space, then it is itself countably based.*

Proof. It suffices to show the result for a sober core-compact space X, which is therefore locally compact. By Theorem 4.5, the space X has an admissible quotient representation $\delta : R \to X$ where $R \subset \Sigma^{\omega}$. Take β to be a countable base for R. Let $x \in X$ and $U \in \mathcal{O}(X)$ be such that $x \in U$. Since X is

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

locally compact, there exists $V \in \mathcal{O}(X)$ and $K \in \mathcal{K}(X)$ such that $x \in V \subset K \subset U$. Let $(I_0, I_1, \ldots)$ be a list of sets with each $I_j \in \beta$ such that $\delta^{-1}(U) = \bigcup_{j=0}^{\infty} I_j$. Suppose $\delta(\bigcup_{j=0}^{k-1} I_j)$ is not a cover of K for any k. Then, there is a sequence $(x_k)_{k\in\mathbb{N}}$ with $x_k \in K \setminus \delta(\bigcup_{j=0}^{k-1} I_j)$ for all k. Since K is compact and X is a sequential space, K is sequentially compact, so (x_k) has a convergent subsequence. Without loss of generality, we can assume that (x_k) is itself convergent and $\lim_{k\to\infty} x_k = x_\infty$.

Since δ is admissible, there is a convergent sequence $(r_k)_{k\in\mathbb{N}}$ with limit r_∞ such that $\delta(r_k) = x_k$ for all $k \in 0, 1, \ldots, \infty$. Since $r_\infty \in I_n$ for some n, there is a finite subset $I_{m_1}, \ldots, I_{m_k}$ of $(I_0, I_1, \ldots)$ such that $V \subset K \subset \bigcup_{j=1}^{k} I_{m_j} \subset U$. This contradicts the assumption that $\delta(\bigcup_{j=0}^{k-1} I_j)$ is not a cover of K.

We have shown that if $x \in X$ and $U \in \mathcal{O}(X)$, then there is a finite subset $\{J_1, \ldots, J_k\}$ of β such that $x \in \operatorname{int}(\delta(\bigcup_{i=1}^{k} J_i)) \subset U$. Hence, $\{\operatorname{int}(\delta(\bigcup_{i=1}^{k} J_i)) \mid J_1, \ldots, J_k \in \beta^*\}$ is a countable base for X. □

Note that the proof given relies crucially on the admissibility of the admissible quotient representation δ, which may be different from the original quotient map q.

The following theorem can be derived from Escardó and Heckmann (2002, Theorem 5.3). Its main significance is to show that core-compactness of X implies local-compactness of $\mathcal{O}(X)$.

Theorem 4.29. *Let X be core-compact. Then, $\mathcal{O}(X)$ with the Scott topology is locally compact.*

Proof. Let $U \in \mathcal{O}(X)$ and $\mathcal{W} \subset \mathcal{O}(X)$ be Scott open with $U \in \mathcal{W}$. By Lemma 4.24, there exists $W \in \mathcal{O}(X)$ such that $U \in \Uparrow W$ and $\Uparrow W \subset \mathcal{W}$. Since $U \in \Uparrow W$ means $W \Subset U$, and X is core-compact, by Lemma 4.25 there exists $V \in \mathcal{O}(X)$ such that $W \Subset V \Subset U$. Define $\uparrow V = \{S \in \mathcal{O}(X) \mid V \subset S\}$, and note that $\uparrow V$ is compact in the Scott topology, since any open cover has a singleton subcover. By Lemma 4.24, the set $\Uparrow V$ is open in the Scott topology. Then, $U \in \Uparrow V \subset \uparrow V \Subset \mathcal{W}$ as required. □

5. Applications to Dynamic Systems

We now use the computable type theory developed in Section 3 to give some results on computable properties of dynamic systems. When considering solutions of nondeterministic systems, we are often interested in function spaces with set-valued types. In this section, we now give some applications of the computability type theory to problems in control and systems theory. We focus on three problems, namely, the evolution of hybrid systems, computation of reachable and viable sets, and control synthesis. Note that using the computable types developed earlier, many of the results are almost trivial to prove.

The computations in this paper were performed with the ARIADNE tool for rigorous numerics and analysis of dynamic systems (Ariadne 2018), commit number 33a1b8dd.

5.1 System behavior

The set of *trajectories* or *solutions* of a dynamic system is the space of continuous functions $\xi : T \to X$, where T is the *time domain*, and X is the *state space*. For a discrete-time system, $T = \mathbb{N}$; for a continuous-time system, $T = \mathbb{R}^+$. Throughout this section, we will assume that X is effectively Hausdorff and effectively overt (Definition 3.38). We require the *property of state*, that if ξ and η are solutions with $\xi(s) = \eta(s)$, then there is a solution ζ with $\zeta(t) = \xi(t)$ for $t \leq s$, and $\zeta(t) = \eta(t)$ for $t \geq s$. For an autonomous system, we also require *time-invariance*, that if ξ is a solution and $s \in T$, then the function defined by $\eta(t) = \xi(t + s)$ is also a solution.

We write trajs (S, x) for the set of trajectories of a system S with initial condition x. For a deterministic system, there is only one trajectory through a given initial state. The solution operator may be represented either as a function $\phi : X \to \mathcal{C}(T; X)$ taking an initial point x to the trajectory

$\xi : T \to X$ with $\xi(0) = x$, by the function $\tilde{\phi} : X \times T \to X$ given by $\tilde{\phi}(x)[t]$. By the exponentiation property, the types $\mathcal{X} \times \mathcal{T} \to \mathcal{X}$ and $\mathcal{X} \to \mathcal{C}(\mathcal{T};\mathcal{X})$ are equivalent.

In a nondeterministic system, there may be many different trajectories with the same initial state. In a Markov stochastic system, the behavior can be described by a function $X \times T \to Pr(X)$, where $Pr(X)$ is the set of probability measures on X. Since we have not considered computable measure theory in this article, we will not consider stochastic systems further.

In general, we do not merely wish to compute the evolution only for a system described by computable data, but for all systems within a class, even if the system data are uncomputable. We therefore express the computability results in terms of both the system description and the initial state.

The simplest class of system to consider is that of a deterministic discrete-time system defined by a continuous function $f : X \to X$ with the update law given by $x_{n+1} = f(x_n)$. The trajectory of f starting at a point x is the function $\xi : \mathbb{N} \to X$ given by $\xi(0) = x$ and $\xi(n+1) = f(\xi(n))$. The evolution function is given by $\phi(x)[0] = x$ and $\phi(x)[n+1] = f(\phi(x)[n])$.

Proposition 5.1. *The evolution of a discrete-time system defined by the update rule $x' = f(x)$ is computable as a function* $\mathrm{traj} : \mathcal{C}(\mathcal{X};\mathcal{X}) \times \mathcal{X} \to \mathcal{C}(\mathbb{N};\mathcal{X})$ *(equivalently as a function $\mathcal{C}(\mathcal{X};\mathcal{X}) \to \mathcal{C}(\mathcal{X};\mathcal{C}(\mathbb{N};\mathcal{X}))$ or $\mathcal{C}(\mathcal{X};\mathcal{X}) \times \mathcal{X} \times \mathbb{N} \to \mathcal{X}$).*

Proof. Immediate, since given f, x, and n, the value $\xi(n)$ for $\xi = \mathrm{traj}\,(f, x)$ can be computed by recursion. □

Remark 5.2. By saying "the evolution is computable as a function $\mathcal{C}(\mathcal{X};\mathcal{X}) \times \mathcal{X} \to \mathcal{C}(\mathbb{N};\mathcal{X})$", we mean that for a system defined by an object f of type $\mathcal{C}(\mathcal{X};\mathcal{X})$, the trajectory starting at x_0 in $\mathcal{X}$ can be computed as a function $\xi : \mathcal{C}(\mathbb{N};\mathcal{X})$ given a name of $f : \mathcal{C}(\mathcal{X};\mathcal{X})$ and $x_0 : \mathcal{X}$.

Example 5.3. *The* logistic map *with parameter μ is defined by $f_\mu(x) = \mu x(1-x)$. It is well-known that for a positive-measure set of values of μ in the range* $[3.56, 4.0]$, *f_μ has a* strange (chaotic) attractor *and that the long-term behavior of the system is unpredictable in practice. However, Proposition 5.1 shows that the evolution is computable given μ and x_0!*

The answer to this conundrum is that very high accuracy must be used to compute the behavior accurately over a long-time interval. As an example, take $\mu = 3.75$ and compute the orbit of $x_0 = 0.5$ using double-precision interval arithmetic. Then after 88 *iterates, the result stabilizes on the interval* $x_{88} \in [0.2197265625{:}0.9375]$, *which provides no useful information about the value of x_{88}. However, using* 106 *bits of precision allows the computation* $x_{88} \in 0.264510325218310[779:867]$, *with an error of the double-precision approximation.*

The system has a fixed-point $p = 1 - 1/\mu = 0.7\dot{3}$. The orbit starting at the closest double-precision approximation x_0 to p (a distance of $\approx 5 \times 10^{-17}$ away, reaches $0.23[2:3]$ *after* 66 *iterates, a distance of* ≈ 0.5 *away. This illustrates the phenomenon of* sensitive dependence on initial conditions, *in which nearby orbits diverge from each other at an average growth factor of $\lambda \approx 1.75$ per unit time.*

5.2 Nondeterministic systems

Nondeterministic systems frequently arise in control and systems theory as models of systems with control or disturbance inputs. A system with state space X and input space U is described by a function $f : X \times U \to X$ with $x_{n+1} = f(x_n, u_n)$.

If the inputs u_n are under the control of the user, we are interested in determining whether there *exists* a trajectory with some given property. Hence, we should compute *overt* sets of trajectories, since given a verifiable predicate p on trajectories, that is, $p : \mathcal{C}(\mathcal{T};\mathcal{X}) \to \mathcal{S}$, we can verify existence of a trajectory in a set B satisfying p if $B : \mathcal{V}(\mathcal{C}(\mathcal{T};\mathcal{X}))$ is effectively overt. If instead the inputs u_n are disturbances from the external environment, then we are interested in properties which hold for

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

all possible trajectories. Hence, we should try to obtain a *compact* set of trajectories B, for which we can verify $\forall \xi \in B : \ p(\xi)$. Note that there is no distinction between which solutions are likely or unlikely; merely between what is possible and impossible.

In both cases, we can define a multivalued map $F : X \rightrightarrows X$ describing the evolution by $F(x) = f(x, U)$ and obtain an update law $x_{n+1} \in F(x_n)$. If U is overt, then $F(x)$ is computable from f and U as an overt set, and if U is compact, then $F(x)$ is computable from x and U as a compact set. Hence, it suffices to consider systems defined by multimaps $F : X \to \mathcal{V}(X)$ and $F : X \to \mathcal{K}(X)$

There are many different ways of describing the solution space of a system:

Definition 5.4 (Representations of solution spaces). *The* behavior *of a system is the set of all solutions,* $\bar{\Phi} \in \mathcal{P}(C(T; X))$. *The canonical* solution trajectory *operator is a function* $\hat{\Phi} : X \to \mathcal{P}(C(T; X))$ *such that* $\hat{\Phi}(x) = \{\xi : T \to X \mid \xi \in \bar{\Phi}$ *and* $\xi(0) = x\}$. *The finite* reachability *operator,* $\tilde{\Phi} : X \times T \to \mathcal{P}(X)$ *defined as* $\tilde{\Phi}(x, t) = \{\xi(t) \mid \xi \in \bar{\Phi} \wedge \xi(0) = x\}$.

The following result shows that the reachable sets can be recovered from the solution trajectory operator. Further, unless X is compact, the set of all solutions cannot be represented as an element of $\mathcal{K}(\mathcal{C}(\mathcal{N}; \mathcal{X}))$, whereas $\hat{\Phi} : \mathcal{X} \to \mathcal{V}(\mathcal{C}(\mathcal{N}; \mathcal{X}))$ cannot be recovered from $\bar{\Phi} \in \mathcal{V}(\mathcal{C}(\mathcal{N}; \mathcal{X}))$. This means that in order to study properties of the system, we should compute $\hat{\Phi}$ and not $\bar{\Phi}$ or $\tilde{\Phi}$.

Proposition 5.5.

(a) Given the solution trajectory operator $\hat{\Phi} : \mathcal{C}(\mathcal{X}; \mathcal{K}(\mathcal{C}(\mathcal{N}; \mathcal{X})))$, *we can compute the reachability operator* $\tilde{\Phi} : \mathcal{C}(\mathcal{X}; \mathcal{C}(\mathcal{N}; \mathcal{K}(\mathcal{X})))$, *and if* $\mathcal{X}$ *is effectively compact, then given* $\hat{\Phi}$, *we can compute the behavior* $\bar{\Phi} : \mathcal{K}(\mathcal{C}(\mathcal{N}; \mathcal{X}))$.

If $\mathcal{X}$ *is effectively Hausdorff, then given the behavior* $\bar{\Phi}$, *we can compute the solution trajectory operator* $\hat{\Phi}$.

(b) Given the solution trajectory operator $\hat{\Phi} : \mathcal{C}(\mathcal{X}; \mathcal{V}(\mathcal{C}(\mathcal{N}; \mathcal{X})))$, *we can compute the reachability operator* $\tilde{\Phi} : \mathcal{C}(\mathcal{X}; \mathcal{C}(\mathcal{N}; \mathcal{V}(\mathcal{X})))$, *and If* $\mathcal{X}$ *is effectively overt, then given* $\hat{\Phi}$, *we can compute the behavior* $\bar{\Phi} : \mathcal{V}(\mathcal{C}(\mathcal{N}; \mathcal{X}))$.

Proof.

(a) Given $\hat{\Phi} : X \to \mathcal{K}(C(\mathbb{N}; X))$, define $\tilde{\Phi} : X \times \mathbb{N} \to \mathcal{K}(X)$ by $\tilde{\Phi}(x, n) = \{\xi(n) \mid \xi \in \hat{\Phi}(x)) = (\hat{\Phi}(x))(n)$, which is computable. Given $\hat{\Phi} : X \to \mathcal{K}(C(\mathbb{N}; X))$, define $\bar{\Phi} = \hat{\Phi}(X)$, which is computable since X is effectively compact.

Given $\bar{\Phi} \in \mathcal{K}(C(\mathbb{N}; X))$, define $\hat{\Phi} : X \to \mathcal{K}(C(\mathbb{N}; X))$ by $\hat{\Phi}(x) = \bar{\Phi} \cap \{\xi \in C(\mathbb{N}; X) \mid 0 \in \xi^{-1}(\{x\}\}$. The Hausdorff property ensures that $\{x\}$ is computable from x as a closed set.

(b) Given $\hat{\Phi} : X \to \mathcal{V}(C(\mathbb{N}; X))$, define $\bar{\Phi} = \hat{\Phi}(X)$, which is computable since X is assumed to be effectively overt, and hence a computable element of $\mathcal{V}(\mathcal{X})$. Define $\tilde{\Phi} : X \times \mathbb{N} \to \mathcal{V}(X)$ by $\tilde{\Phi}(x, n) = \{\xi(n) \mid \xi \in \hat{\Phi}(x)) = (\hat{\Phi}(x))(n)$. □

This means that in order to study properties of the system, we should compute $\hat{\Phi}$ and not $\bar{\Phi}$ or $\tilde{\Phi}$. We henceforth write $\Phi_F : X \to \mathcal{P}(C(\mathbb{N} \to X))$ for the solution trajectory operator of the discrete-time system defined by $F : X \to \mathcal{P}(X)$. Using Theorem 3.32 and Proposition 5.5, computability of the forward-time evolution of discrete-time nondeterministic systems is immediate:

Theorem 5.6. *The dynamics of a nondeterministic discrete-time system F is computable in the following cases:*

(a) If $F : \mathcal{X} \to \mathcal{V}(\mathcal{X})$, *then* $\Phi_F : \mathcal{X} \to \mathcal{V}(\mathcal{C}(\mathcal{N}, \mathcal{X}))$ *is computable from F.*
(b) If $F : \mathcal{X} \to \mathcal{K}(\mathcal{X})$, *then* $\Phi_F : \mathcal{X} \to \mathcal{K}(\mathcal{C}(\mathcal{N}, \mathcal{X}))$ *is computable from F.*

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

5.3 Differential systems

We now consider the computability of systems defined by differential equations or differential inclusions. For simplicity, we assume that X is a Euclidean space $\mathbb{R}^n$, though these results also extend to differential manifolds and locally compact Banach spaces. To prove the results of this section, we need to go back to first principles to solve the differential systems; in particular, we need to resort to the classical Arzela–Ascoli and Michael theorems to assert the existence of solutions.

It is well-known that for general differentiable systems, solutions may escape to infinity in finite time. We therefore consider trajectories which may be defined on an open subinterval $[0, t_\infty)$ of $\mathbb{R}^+$, as well as over $[0, \infty)$, so have the partial function type $\mathcal{C}_\mathcal{O}(\mathcal{R}^+; \mathcal{X})$ from Definition 3.37.

Recall that a function $f : X \to Y$ is *Lipschitz* over set U if there exists $L > 0$ such that for all $x_1, x_2 \in U$, $d(f(x_1), f(x_2)) \leq Ld(x_1, x_2)$, and *locally Lipschitz* if there is an open cover $\mathcal{U}$ of X such that f is Lipschitz over all $U \in \mathcal{U}$. We say that a function has (at most) *linear growth* if there exists a constant $C > 0$ such that $d(f(x), 0) \leq \mathcal{C}(1 + d(x, 0))$.

Theorem 5.7. *Let $f : X \to X$ be locally Lipschitz continuous with linear growth. Then, the solution operator $\hat{\phi}_f$ of $\dot{x} = f(x)$ is computable $\mathcal{C}(\mathcal{X}; \mathcal{X}) \times \mathcal{X} \to \mathcal{C}_\mathcal{O}(\mathcal{R}^+; \mathcal{X})$.*

There are many possible proofs; a simple proof can be found in Collins and Graça (2008). Here, we give two proofs; one based on Picard iteration (see Daniel and Moore 1970) and one based on Euler steps using boxes to bound the flow tube (Collins and Graça 2008). For simplicity, we restrict to the globally Lipschitz case. Note that we can weaken the locally Lipschitz condition to simply requiring uniqueness of solutions (Ruohonen 1996).

Sketch of proof based on Picard iteration. Let L be a Lipschitz constant for f. Then, $\|f(x)\| \leq K + L\|x\|$ where $K \geq \|f(0)\|$. We restrict to finding solutions on the interval $[0, h]$ with $h < 1/L$ starting at a given x_0 with $\|x_0\| \leq M$. Define the *Picard* operator by $\text{Pic}\,[\xi](t) = x_0 + \int_0^t f(\xi(\tau))d\tau$. Then, $\|\,\text{Pic}\,[\xi] - \text{Pic}\,[\eta]\| \leq \int_0^t |f(\xi(\tau)) - f(\eta(\tau))|d\tau \leq \int_0^t L|\xi(\tau) - \eta(\tau)|d\tau \leq Lh\|\xi - \eta\|$, so Pic is a contraction operator. Define $\xi_0(t) = x_0$, and $\xi_{n+1} = \text{Pic}\,[\xi_n]$. Then for all $m \leq n$, $\|\xi_n - \xi_m\| \leq \sum_{k=m}^{n-1} (Lh)^k \leq (Lh)^m/(1 - Lh)$, so ξ_n is a uniformly convergent sequence of functions. Since $\mathbb{R}$ is complete, $\xi_\infty(t) = \lim_{n\to\infty} \xi_n(t)$ exists pointwise and standard results of classical analysis show that ξ_∞ is continuous, is the uniform limit of the ξ_n, and satisfies the differential equation. □

Sketch of proof based on Euler scheme. If B, D, B' are coordinate-aligned boxes in $\mathbb{R}^n$. Write $\langle S\rangle$ for the convex hull of a set S. If $B + [0, h]\langle f(D)\rangle \subset D$ and $B + h\langle f(D)\rangle \subset B'$, then by then mean value theorem, we can show that the evolution ϕ of $\dot{x} = f(x)$ satisfies $\phi(x, t) \in D$ whenever $x \in B$ and $t \in [0, h]$, and $\phi(x, h) \in B'$ whenever $x \in B$. Note that here, the set $B + [0, h]\langle f(D)\rangle$ is explicitly taken as $\{x' \mid \exists x \in B,\ \ \tau \in [0, h]\ \text{ and } y \in \text{conv}\,(F(D)), x' = x + \tau y\}$. Suppose we can find times $0 = t_0 < t_1 < \cdots < t_m = T$, and boxes $B_0, B_1, \ldots, B_n$ and $D_0, D_1, \ldots, D_{n-1}$ such that setting $h_k = t_{k+1} - t_k$ we have $B_k + [0, h_k]\langle f(D_k)\rangle \subset D_k$ and $B_k + h_k\langle f(D_k)\rangle \subset B_{k+1}$ for $k = 0, 1, \ldots, n-1$. Then, we have shown that for $x_0 \in B_0$, the solution ξ with $\xi(0) = x_0$ has $\xi(T) \in B_n$. The result follows that an exhaustive enumeration over rational boxes finds boxed satisfying the condition with all B_k arbitrarily small. An estimate for the radius of B_k is given by $|B_{k+1}| \leq \delta + (1 + Lh)|B_k|$, where δ can be taken to be arbitrarily small, from which we find $|B_k| \leq \delta\big((1 + Lh)^{k+1} - 1\big)/Lh \leq \delta\big((1 + Lh) \exp\,(LT) - 1\big)/Lh$. □

Example 5.8 (The Van der Pol oscillator). *The* Van der Pol oscillator *is a nonlinear differential equation defined by*

$$\ddot{x} + \mu(x^2 - 1)\dot{x} + x = 0.$$

Introducing variable $y = \dot{x}$ gives the system of coupled first-order equations

$$\dot{x} = y; \qquad \dot{y} = \mu(1 - x^2)y - x.$$

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

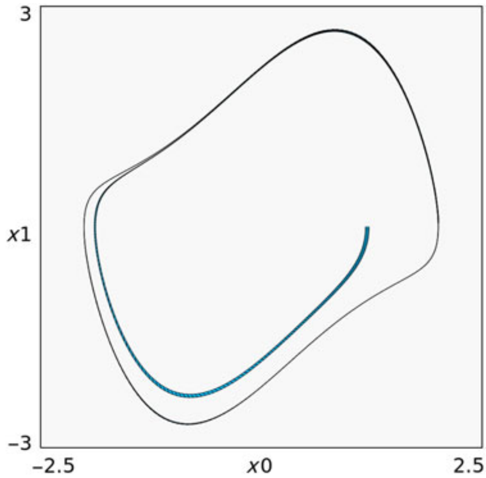

Figure 2. The limit cycle of the Van der Pol oscillator.

For $\mu = 1.0$, the system exhibits a limit cycle passing through x_m when $\dot{x} = 0$, for some $x_m \in [2.0, 2.01]$. A rigorous over-approximation to a trajectory converging to the limit-cycle is shown in Figure 2.

We now turn to nondeterministic differential systems as defined by differential inclusions $\dot{x} \in F(x)$. For an introduction to differential inclusions, see Aubin and Cellina (1984). Following the well-known solution concept of Filippov (1988), we may first need to compute the convex hull of the right-hand side. The continuous case was first proved in Puri (1996), but easily splits into the lower- and upper-semicontinuous cases. The lower-semicontinuity with the one-sided Lipschitz condition was proved in Gabor (2007). Full proofs can be found in Collins and Graça (2009).

We say that a function $f : \mathbb{R}^n \to \mathcal{P}(\mathbb{R}^n)$ is *one-sided Lipschitz* over a set U if for every $w, x \in U$, and $f_w \in F(w)$, there exists $f_z \in F(x)$ such that $(w - x) \cdot (f_w - f_x) \leq L\|w - x\|^2$ and that f is *one-sided locally Lipschitz* if there is an open cover $\mathcal{U}$ of X such that for all $U \in \mathcal{U}$, f is one-sided Lipschitz over U.

Theorem 5.9.

(a) Let F be one-sided locally Lipschitz lower-semicontinuous with closed convex values. Then the solution operator $(F, x) \mapsto \hat{\Phi}_F(x)$ of $\dot{x} \in F(x)$ is computable $\mathcal{C}(\mathcal{X}; \mathcal{V}(\mathcal{X})) \times \mathcal{X} \to \mathcal{V}(\mathcal{C}_{\mathcal{O}}(\mathcal{R}^+; \mathcal{X}))$.

(b) Let F be upper-semicontinuous with compact convex values. Then, the solution operator of $\dot{x} \in F(x)$ is computable $\mathcal{C}(\mathcal{X}; \mathcal{K}(\mathcal{X})) \times \mathcal{X} \to \mathcal{K}(\mathcal{C}_{\mathcal{O}}(\mathcal{R}^+; \mathcal{X}))$.

5.4 Evolution of hybrid systems

A *hybrid system* is a dynamic system in which the state undergoes continuous evolution (governed by differential equations/inclusions) interspersed with discrete jumps, which are instantaneous changes in the state occurring when the continuous evolution enters a *guard set* G. See Goebel and Teel (2006), Collins (2011) for more details.

It is well-known in the literature, for example, Asarin et al. (1995), Blondel and Tsitsiklis (2000) that basic problems of reachability and stability are undecidable for even simple classes of hybrid systems. However, the cause of undecidability for the reachability problem is *infinite-time* behavior, as we shall see in Section 5.5. In this section, we shall show that even the *finite-time* of a nonlinear hybrid system is uncomputable. However, under appropriate assumptions on the dynamics, we can compute compact sets of trajectories for compact-valued systems and compute backward-time evolution for overt-valued systems.

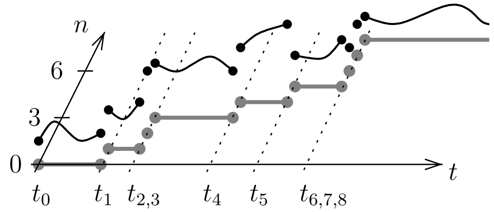

Figure 3. Hybrid trajectory over a hybrid time domain. Notice that there are two events at time $t_2 = t_3$, and three at time $t_6 = t_7 = t_8$.

Definition 5.10 (Hybrid system). *A* hybrid system *is defined by a tuple* $H = (X, F, G, R)$, *where*

- $X = \mathbb{R}^n$ *is the* state space.
- $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ *defines the* continuous dynamics $\dot{x} \in F_q(x)$ *with solution* flow Φ_F.
- $G \subset X$ *is the* guard set, *defined as* $G = \{x \in X \mid g(x) \geq 0\}$ *for the* guard function $g : X \to \mathbb{R}$.
- $R : X \rightrightarrows X$ *is the* reset *map.*

Remark 5.11. There are many alternative definitions of hybrid system, most of which also have a *discrete* state q as well as a *continuous* state x, yielding a state space $\bigsqcup_{q \in Q} \{q\} \times \mathbb{R}^{n_q}$. Since in this paper we are essentially interested in the interplay between the continuous dynamics and instantaneous jumps, our simpler class of systems suffices.

Definition 5.12 (Hybrid trajectory). *A* hybrid trajectory *in state space* X *is a function* $\xi : \mathcal{T} \to X$, *where* $\mathcal{T}$ hybrid time domain, *namely a subset of* $\mathbb{R}^+ \times \mathbb{Z}^+$ *of the form* $\bigcup_{n=0}^{N} [t_n, t_{n+1}] \times \{n\}$. *Here,* $N \in \mathbb{Z}^+ \cup \{\infty\}$ *is the* number of events, *and the final interval is* $[t_N, \infty)$ *if* N *is finite, whereas* $t_n \to \infty$ *if* $N = \infty$. *We let* $\xi_n : [t_n, t_{n+1}]$ *be such that* $\xi(t, n) = \xi_n(t)$, *and set* $t_k = \tau_k(\xi)$

A hybrid trajectory ξ *with domain* $\prod_{n=0}^{N} [t_n, t_{n+1}] \times \{n\}$ *is a solution of the hybrid system* (X, F, G, R) *if:*

- *for almost every* $t \in [t_n, t_{n+1}]$, ξ_n *is differentiable at* t *with* $\xi'(t) \in F(\xi(t))$.
- $\xi_n(t_{n+1}) \in G$ *for all* n, *but* $\xi_n(t) \notin G^\circ$ *for all* $t \in [t_n, t_{n+1})$.
- $\xi_{n+1}(t_{n+1}) \in R(\xi_n(t_{n+1}))$ *for all* n.

We write trajs (H, x) *for the set of solutions of* H *starting at a point* x, *and* trajs (H) *for the set of all solutions.*

The *reachable set* of a hybrid system H is defined to be $\{\xi(t, n) \mid \xi \in \text{trajs}\,(H) \wedge (t, n) \in \text{dom}\,(\xi)\}$. The set of states Ψ_1 reachable *after the first event* is defined to be $\{\xi_1(t_1) \mid \xi \in \text{trajs}\,(H)\}$.

Suppose $\xi(t)$ is a continuous trajectory with $g(\xi(0)) < 0$, and $g(\xi(t)) > 0$ for some $t > 0$. Then, clearly the trajectory ξ crosses the guard set at some time. We define the *hitting time* τ_h by $\tau_h(g, \xi) = \min\{t \in \mathbb{R} \mid g(\xi(t)) = 0\}$ and the *crossing time* τ_c as $\tau_c(g, \xi) = \inf\{t \in \mathbb{R} \mid g(\xi(t)) > 0\}$. Clearly $\tau_h(g, \xi) \leq \tau_c(g, \xi)$, but the two need not be equal, in general. If $\tau_h(g, \xi) = \tau_c(g, \xi)$, then we say that ξ crosses g *instantaneously* at $\tau = \tau_h(g, \xi)$. Otherwise, it may be the case that $\xi(t)$ slides along the guard set G between τ_h and τ_c, or touches G and re-enters D before later crossing G. We define the *touching time set* as $\tau(g, \xi) = \{t \in \mathbb{R}^+ \mid g(\xi(t)) = 0 \ \wedge \ \forall s \leq t, \ g(\xi(s)) \leq 0\}$.

An example of a hybrid trajectory is shown in Figure 3.

Lemma 5.13. *Let* $g : \mathcal{X} \to \mathcal{R}$. *Then set of continuous functions* $\xi : [0, \infty) \to \mathcal{X}$ *with* $g(\xi(0)) < 0$ *and* $g(\xi(t)) > 0$ *for some* $t > 0$ *is computable in* $\mathcal{O}(\mathcal{C}(\mathcal{R}^+; \mathcal{X}))$ *from* g.

Proof. For fixed t, $\{\xi : \mathcal{R} \to \mathcal{X} \mid g(\xi(t)) < 0\}$ is computable in $\mathcal{O}(\mathcal{C}(\mathcal{R}^+; \mathcal{X}))$, since $\xi \mapsto g(\xi(t))$ is computable with values in $\mathbb{R}$, and $x \mapsto (x < 0)$ is computable in $\mathcal{S}$. The result follows since $\exists t \in \mathbb{R},\ g(\xi(t)) > 0 \iff \exists t \in \mathbb{Q},\ g(\xi(t)) > 0$, so $\{\xi : \mathcal{R}^+ \to \mathcal{X} \mid \exists t \in \mathbb{R}^+\ g(\xi(t)) > 0\}$ is a countable union of computable open sets. $\square$

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Theorem 5.14. *The touching time set $\tau(g,\xi)$ is computable in $\mathcal{A}(\mathcal{R})$ from $g:\mathcal{C}(\mathcal{X},\mathcal{R})$ and $\xi:\mathcal{C}(\mathcal{R}^+,\mathcal{X})$. Further, if $g(\xi(t))>0$ for some $t>0$, then $\tau(g,\xi)$ can be computed in $\mathcal{K}(\mathcal{R})$.*

Proof. Define $\gamma_{\xi,g}(t)=g(\xi(t))$ and $\mu_{\xi,g}(t)=\sup\{g(\xi(s))\mid s\in[0,t]\}$ which is a computable function. Then, $\tau(g,\xi)=\gamma_{\xi,g}^{-1}(\{0\})\cap\mu_{\xi,g}^{-1}((-\infty,0])$ so is computable. If $g(\xi(t))>0$, then $\tau(g,\xi)=\tau(g,\xi)\cap[0,t]$, so is effectively compact. □

Theorem 5.15. *Consider a hybrid system (X,F,G,R) whose flow $\Phi_F:\mathcal{X}\to\mathcal{K}(\mathcal{C}(\mathcal{R}^+;\mathcal{X}))$ is a compact-valued multiflow, guard set $G=\{x\in X\mid g(x)\geq 0\}$ for continuous $g:\mathcal{X}\to\mathcal{R}$, and reset map $R:\mathcal{X}\to\mathcal{K}(\mathcal{X})$ is compact-valued. Let $X_0:\mathcal{K}(\mathcal{X})$ be a compact set of initial states. Then, the set of points $\Psi_1(X_0)$ reachable after the first event is computable in $\mathcal{K}(\mathcal{X})$ from Φ_F,g,R and X_0.*

Proof. The set of points reachable after the first event of a continuous solution ξ is $R(\xi(\tau(g,\xi))$, which is computable in $\mathcal{K}(\mathcal{X})$ from $\xi\in\mathcal{C}(\mathcal{R}^+;\mathcal{X})$. The set of trajectories with initial condition X_0 is $\Phi_F(X_0)$, so is computable in $\mathcal{K}(\mathcal{C}(\mathcal{R}^+;\mathcal{X}))$. Then, $\Psi(X_0)$ is the union of $R(\xi(\tau(g,\xi))$ for ξ in the compact set $\Phi_F(X_0)$, so is computable in $\mathcal{K}(\mathcal{X})$. □

Unfortunately, the set of points reachable after the first event is not computable as an overt set, since the touching time set is not computable as an overt set.

Lemma 5.16. *The touching time set $\tau(g,\xi)$ is not computable as an overt set from $g:\mathcal{X}\to\mathcal{R}$ and $\xi:\mathcal{R}^+\to\mathcal{X}$.*

Proof. Consider $X=\mathbb{R}$, $g(x)=x$ and $\xi_\epsilon(t)=\xi(t)+\epsilon$ where $\xi(t)=\max(t-2,0)+\min(t-1,0)$. Note that $\xi(t)=t-1$ for $0\leq t\leq 1$, $\xi(t)=0$ for $1\leq t\leq 2$, and $\xi(t)=t-2$ for $t\geq 2$. The touching time set of ξ_ϵ is $\{2-\epsilon\}$ for $\epsilon<0$, $[1,2]$ for $\epsilon=0$ and $\{1-\epsilon\}$ for $0<\epsilon<1$. □

It turns out that event detection is easier in the context of backward reachability

Theorem 5.17. *Consider a hybrid system $H=(X,F,G,R)$ where $\Phi_F:\mathcal{X}\to\mathcal{V}(\mathcal{C}(\mathcal{R}^+;\mathcal{X}))$ is an overt multiflow, $G=\{x\in X\mid g(x)\geq 0\}$ for $g:\mathcal{X}\to\mathcal{R}$ and $R:\mathcal{X}\to\mathcal{V}(\mathcal{X})$ is overt-valued. Let $V:\mathcal{O}(\mathcal{X})$ be an open set. Define $\Psi_1^{-1}(V)$ to be the set of initial points for which there is a solution for which the state is in V immediately after the first event. Then $\Psi_1^{-1}(V)$ is computable in $\mathcal{O}(\mathcal{X})$ from Φ_F,G,R and V.*

Proof. The trajectory ξ crosses G in $R^{-1}(V)$ if $\xi(\tau(g,\xi))\subset V$. Since $R:\mathcal{X}\to\mathcal{V}(\mathcal{X})$, $U=R^{-1}(V)$ is computable in $\mathcal{O}(X)$. Since $\tau(g,\xi)$ is compact and ξ is continuous, $W=\{\xi\mid\tau(g,\xi)\subset U\}$ is computable in $\mathcal{O}(\mathcal{C}(\mathcal{R}^+;\mathcal{X}))$. Since $\Phi:\mathcal{X}\to\mathcal{V}(\mathcal{C}(\mathcal{R}^+;\mathcal{X}))$, $\Phi^{-1}(W)$ is computable in $\mathcal{O}(\mathcal{X})$. We have $\Psi^{-1}(V)=\{x\mid\exists\xi\in\Phi_F(x)\text{ s.t. }\xi(\tau(g,\xi))\subset R^{-1}\}\}=\Phi_F^{-1}(\{\xi(\tau(g,\xi))\subset R^{-1}\})$, so $\Psi^{-1}(V)$ is computable in $\mathcal{O}(\mathcal{X})$. □

Example 5.18. *Consider a simple heating system, in which the temperature is to be controlled by means of a thermostat. The thermostat turns the heater on when the temperature goes below $15.0°\pm 0.2°$, and off when the temperature goes above $20.0°$. The temperature satisfies the differential equation $\dot{T}=P+K(T_e+T_a\cos(2\pi t)-T)$ where t is the time (in days), $K=1.0$ is the conductivity, $T_e=16.0$ is the average external temperature, $T_a=8.0$ is half the amplitude of the external temperature range, and P is the heater power, which is 4.0 when turned on, and 0 when turned off.*

The behavior of the system is shown in Figure 4. The heater is initially turned off, and the temperature decreases until it reaches the threshold for the heater to turn on. However, even if the heater is not turned on, the external temperature increases sufficiently that on the first day, the temperature never decreases below the threshold 14.8° below in which the heater must turn on. Hence, there are two qualitatively different evolutions, one in which the heater turns on (in red) and one in which it

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

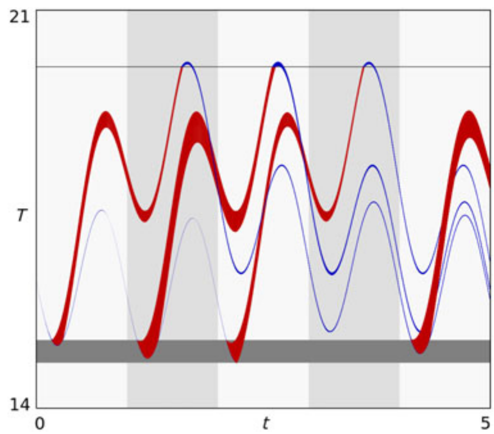

Figure 4. Behavior of the heating system. The blue flow tubes indicate where the heater is off, and the red where it is on.

remains off (in blue). By the third day, the heater must turn itself on, but if it was turned on during the first day, then it is turned off in the second day, though may then turn on again during the fifth day.

5.5 Reachable and viable sets

We now apply the results of Section 5.2 to prove computability of some infinite-time operators in discrete-time dynamical systems. Computability of reachable sets was considered in Collins (2005). Computability of the viability kernel was considered in Saint-Pierre (1994). Similar results for upper-semicontinuous hybrid systems have been obtained in Fränzle (1999), Aubin et al. (2002), Goebel and Teel (2006).

In this section, we will sometimes require that $\mathcal{X}$ is effectively locally compact, so there is a recursively-enumerable set $\mathcal{D}$ of pairs $(V_n, K_n) \in \mathcal{O}(\mathcal{X}) \times \mathcal{K}(\mathcal{X})$ such that $V_n \subset K_n$ for all n, and for any compact C and open U with $C \subset U$ there exists n such that $C \subset V_n$ and $K_n \subset U$.

We define the *reachable set* of a system $F : X \to \mathcal{P}(X)$ with initial state set X_0 as

$$\operatorname{reach}(F, X_0) = \{x \in X \mid \exists \text{ solution } \xi \text{ and } t \in T \text{ with } \xi(0) \in X_0 \text{ and } \xi(t) = x\}.$$

Theorem 5.19. *The reachable set operator* reach *is computable as a function* $\mathcal{C}(\mathcal{X}; \mathcal{V}(\mathcal{X})) \times \mathcal{V}(\mathcal{X}) \to \mathcal{V}(\mathcal{X})$*, but not as a function* $\mathcal{C}(\mathcal{X}; \mathcal{K}(\mathcal{X})) \times \mathcal{K}(\mathcal{X}) \to \mathcal{K}(\mathcal{X})$.

Proof. We can write $\operatorname{reach}(F, X_0) = \bigcup_{i=0}^{\infty} X_i$, where $X_{i+1} = X_i \cup F(X_i)$. Then $\operatorname{reach} : \mathcal{C}(X; \mathcal{V}(X)) \times \mathcal{V}(X) \to \mathcal{V}(X)$ is computable since all operations are computable. However, reach fails to be computable from $\mathcal{C}(\mathcal{X}; \mathcal{K}(\mathcal{X})) \times \mathcal{K}(\mathcal{X})$ to $\mathcal{K}(\mathcal{X})$ even if X is compact since it is easy to show that reach is not upper-semicontinuous in parameters, as in Example 5.20. □

Example 5.20. *Consider the system* $f : \mathbb{R} \to \mathbb{R}$ *defined by* $f_\epsilon(x) = \epsilon + x + x^2 - x^4$. *Then,* $\operatorname{reach}(f_0, \{-1/2\}) \subset [-1, 0]$, *but* $\operatorname{reach}(f_\epsilon, \{-1/2\}) \not\subset [-1, 1/2]$ *for any* $\epsilon > 0$.

Let (X, d) be a metric space. An ϵ-trajectory of $F : X \to \mathcal{P}(X)$ is a sequence $\vec{x} \in X^\omega$, such that for all $n \in \mathbb{N}$, there exists $y_{n+1} \in X$ such that $y_{n+1} \ni F(X_n)$ and $d(y_{n+1}, x_{n+1}) < \epsilon$. The *chain-reachable set* of F the set of all points reachable by ϵ-orbits for arbitrarily-small ϵ.

An equivalent definition of the chain-reachable set, which is valid in arbitrary locally compact spaces, is:

$$\operatorname{chainreach}(F, X_0) = \bigcap \{U \in \mathcal{O}(X) \mid \operatorname{cl}(U) \text{ is compact, and } X_0 \cup F(\operatorname{cl}(U)) \subset U\}.$$

Theorem 5.21. *Let $\mathcal{X}$ be an effectively locally compact type. If $F : \mathcal{C}(\mathcal{X}; \mathcal{K}(\mathcal{X}))$, $X_0 : \mathcal{K}(\mathcal{X})$ and* chainreach (F, X_0) *is bounded, then* chainreach (F, X_0) *is computable in $\mathcal{K}(\mathcal{X})$ from F and X_0. Further,* chainreach (F, X_0) *is the optimal $\mathcal{K}(\mathcal{X})$-computable over-approximation to* reach (F, X_0).

Proof. It is clear that chainreach $(F, X_0) = \bigcap\{V \mid (K, V) \in \mathcal{D} \text{ and } X_0 \cup F(K) \subset V\}$, proving computability. The proof of optimality involves considering perturbations and can be found in Collins (2007). □

The *viability kernel* of a multivalued map F and a set S is given by

$$\text{viab}\,(F, S) = \{x \in X \mid \exists \text{ solution } \xi \text{ s.t. } x = \xi(0) \text{ and } \forall t \in T,\ x(t) \in S\}.$$

Since we consider existence of solutions of the system, the nondeterminism is most naturally interpreted as a user input, rather than as noise. The viability kernel of S for F represents the set of initial points which can be controlled to remain in S for all times.

For compact-valued systems, the viability kernel of a compact set is computable:

Theorem 5.22. *The viability kernel operator* viab (F, S) *is computable as a function* $\mathcal{C}(\mathcal{X}, \mathcal{K}(\mathcal{X})) \times \mathcal{K}(\mathcal{X}) \to \mathcal{K}(\mathcal{X})$.

Proof. Write viab $(F, S) = \bigcap_{i=0}^{\infty} S_i$, where $S_0 = S$ and $S_{i+1} = S_i \cap F^{-1}(S_i)$. □

Given the interpretation of a compact-valued nondeterminism as uncontrollable noise, this result is not physically relevant. Unfortunately, the viability kernel of an overt-valued system is not computable as an open or overt set. However, we can define a *robust viability kernel*

$$\text{robviab}\,(F, S) = \bigcup\{C \in \mathcal{K}(X) \mid C \subset S \cap F^{-1}(\,\text{int}\,(C))\}.$$

Theorem 5.23. *Let $\mathcal{X}$ be an effectively locally compact type. The robust viability kernel operator* robviab (F, S) *is computable as a function* $\mathcal{C}(\mathcal{X}, \mathcal{V}(\mathcal{X})) \times \mathcal{O}(\mathcal{X}) \to \mathcal{O}(\mathcal{X})$.

Proof. Write robviab $(F, S) = \bigcup\{K \mid (K, V) \in \mathcal{D} \text{ and } K \subset S \cap F^{-1}(V)\}$. □

Example 5.24. *The Hénon map is the function $f(x, y) = (a - x^2 + by, x)$. For parameter values $a = 1.3$, $b = 0.3$, the system is chaotic. We computed the reachable set starting at* $(0.55, 1.0)$ *up to* 114 *iterates with* ARIADNE *using zonotope (affine) enclosures. An approximate analysis shows that f^{114} magnifies initial errors by a factor of approximately* $6\times10^{1}2$. *A lower approximation to the reachable set and an over-approximation to the chain-reachable set are shown in Figure 5. The approximation to the reachable set is defined by a collection of zonotopes, each of which is guaranteed to contain a point in the true reachable set.*

5.6 Control synthesis

A *noisy control system* with state space X, input space U and noise space V is a function $f : X \times U \times V \to X$. We assume that U is an overt space and V a compact space, and define $F_U : X \to \mathcal{P}(X \times U)$, $F_U(x) = \{(x, u) \mid u \in U\}$, and $F_V : X \times U \to \mathcal{P}(X)$ by $F_V(x, u) = \{f(x, u, v) \mid v \in V\}$.

The *controllable set* ctrl (f, T, S) of f with target set T and safe set S is the set of initial points for which there is a control law guiding the state into T in finite time, while remaining in S. It can be determined recursively by $T_0 = T \cap S$, $T_{i+1} = T_i \cup \{x \in X \mid \exists u \in U,\ \forall v \in V,\ f(x, u, v) \in T_i\} \cap S$ and ctrl $(f, T, S) = \bigcup_{i=0}^{\infty} T_i$.

The problem of *control synthesis* is to develop a *controller* for the system, which uses the system outputs to generate suitable inputs to solve the control problem. The methods given here generalize provide a general framework for control synthesis for discrete-time systems, similar to existing approaches for linear (Asarin et al. 2000) and nonlinear (Tomlin et al. 2000) hybrid systems.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

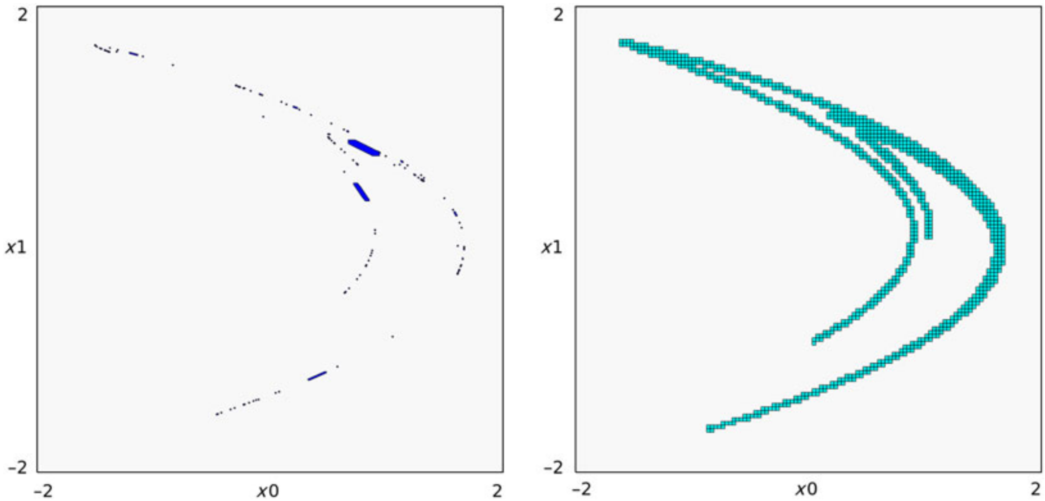

Figure 5. A lower approximation to the reachable set (left) and an over-approximation to the chain-reachable set (right).

Theorem 5.25. *Let $\mathcal{U}$ be an effectively overt type, and $\mathcal{V}$ effectively compact. Then, the controllable set operator* $\mathrm{ctrl} : \mathcal{C}(\mathcal{X},\mathcal{U},\mathcal{V};\mathcal{X}) \times \mathcal{O}(\mathcal{X}) \times \mathcal{O}(\mathcal{X}) \to \mathcal{O}(\mathcal{X})$ *is computable.*

Proof. The multivalued functions $F_U : X \to \mathcal{V}(X \times U)$ and $F_V : (X \times U) \to \mathcal{K}(X)$ can be computed from f, U and V. Write $T_0 = T$ and $T_{i+1} = T_i \cup (F_U^{-1}(F_V^{\Leftarrow}(T_i)) \cap S)$. Then, the controllable set is given by $\bigcup_{i=0}^{\infty} T_i$. □

Classically, a *state feedback control law* is a function $g : X \to U$. There are systems which are controllable by a discontinuous state feedback, but not a continuous feedback, so we cannot hope to solve a general control problem by computing a continuous state feedback. One solution to this difficulty is to first compute a *supervisor*, which is a multivalued function $G : X \rightrightarrows U$ such that taking inputs $u_n \in G(x_n)$ always gives a solution to the control problem. If G is open-valued, we can then construct a deterministic feedback law by taking $g(x) \in G(x)$.

Theorem 5.26. *Let $\mathcal{X}$ be an effectively locally compact type, $\mathcal{U}$ be an effectively overt type, $\mathcal{V}$ be effectively compact, $f : \mathcal{X} \times \mathcal{U} \times \mathcal{V} \to \mathcal{X}$, $T, S : \mathcal{O}(\mathcal{X})$, and $X_0 : \mathcal{K}(\mathcal{X})$. If* $\mathrm{ctrl}\,(f, T, S) \supset X_0$*, then there is a supervisor $G : X \to O(U)$ which can be computed from f, T, S, X_0.*

Proof. From $T_{i+1} = T_i \cup F_U^{-1}(F_V^{\Leftarrow}(T_i))$ and $X_0 \subset T_n$, by local-compactness of $\mathcal{X}$ we can find open B_i, C_i such that $\overline{C}_0 \subset T$, $X_0 \subset \bigcup_{i=0}^{n} B_i$, and for all i, $\overline{C}_i \subset B_i$ and $B_{i+1} \subset F_U^{-1}(F_V^{\Leftarrow}(C_i)$. For $x \in B_i \setminus \bigcup_{j=0}^{i-1} \overline{C}_j$, define $G_i(x) = \{u \in U \mid (x, u) \in F_V^{\Leftarrow}(C_{i-1})\}$, and define $G(x) = \bigcup\{G_i(x) \mid x \in B_i \setminus \bigcup_{j=0}^{i-1} \overline{C}_j\}$. □

A noisy control system with *partial observations* with output space Y and measurement noise space W is defined by functions $f : X \times U \times V \to X$ and $h : X \times W \to Y$. We assume W is compact and define $H : X \to \mathcal{K}(Y)$ by $H(x) = \{h(x, w) \mid w \in W\}$. An *observer* for the system takes values $\hat{X} \in \mathcal{K}(X)$, with initialization $\hat{X}_0 = X_0 \cap H^{-1}(y_0)$ and update rule

$$\hat{X}_{n+1} = \hat{F}(\hat{X}_n, u_n, y_{n+1}) = F_V(\hat{X}_n, u_n) \cap H^{-1}(y_{n+1}). \tag{14}$$

In order to guarantee control to the target set within the safe set, we require $\hat{X}_n \subset T$ for some n, and $\hat{X}_i \subset S$ for all $i \leq n$. We can therefore define sets $T_i \subset \mathcal{K}(X)$ by $T_0 = \{C \in \mathcal{K}(S) \mid C \subset T\}$ and $T_{i+1} = \{C \in \mathcal{K}(S) \mid \exists u \in U,\ \forall y \in Y,\ \hat{F}(C, u, y) \in T_i\}$. Note that $\{C \in \mathcal{K}(X) \mid C \subset U\}$ is open in $\mathcal{K}(X)$

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

for U open in X. Then, T_{i+1} is the set of all state estimates for which we can choose an input such that the next state estimate is guaranteed to be in T_i. The problem is solvable if, and only if, there exists n such that $\bigcup_{i=1}^{n} T_i \supset \{C \in \mathcal{K}(X) \mid \exists y \in Y \text{ s.t. } C = X_0 \cap H^{-1}(y)\}$.

Define the controllable set operator ctrl by $X_0 \in \text{ctrl}\,(f, h, S, T)$ if X_0 is controllable into T inside S under the system (f, h).

Theorem 5.27. *Let $\mathcal{U}$ be an effectively overt type and $\mathcal{V}, \mathcal{W}$ be effectively compact. The controllable set operator* $\text{ctrl}\,(f, h, T, S)$ *for systems with partial observations is computable* $\text{ctrl} : \mathcal{C}(\mathcal{X}, \mathcal{U}, \mathcal{V}; \mathcal{X}) \times \mathcal{C}(\mathcal{X}, \mathcal{W}; \mathcal{Y}) \times \mathcal{O}(\mathcal{X}) \times \mathcal{O}(\mathcal{X}) \to \mathcal{O}(\mathcal{K}(\mathcal{X}))$.

Proof. Define a function $\hat{H} : \mathcal{K}(X) \times Y \to \mathcal{K}(X)$ by $\hat{H}(C, y) = C \cap H^{-1}(Y)$. Then $\hat{H}$ is computable. Further, either Y is compact or $\hat{H}(C, Y) = \hat{H}(C, H(C)) \cup \emptyset$, so $\hat{H}(C, Y) = \{\hat{H}(C, y) \mid y \in Y\} \in \mathcal{K}(\mathcal{K}(X))$. For $X_0 \in \mathcal{K}(X)$, define $\hat{X}_0 = \{C \in \mathcal{K}(X) \mid \exists y \in Y \text{ s.t. } C = X_0 \cap H^{-1}(y)\}$. Then, $\hat{X}_0 = \hat{H}(X_0)$ is computable in $\mathcal{K}(\mathcal{K}(\mathcal{X}))$.

The set $T_0 = \{C \in \mathcal{K}(S) \mid C \subset T\} = \{C \in \mathcal{K}(X) \mid C \subset S \cap T\}$ is computable in $\mathcal{O}(\mathcal{K}(X))$ directly from S and T. We now show that T_{i+1} is computable in $\mathcal{O}(\mathcal{K}(X))$. Let $V_i = \{C \in \mathcal{K}(X) \mid \forall y \in Y,\ C \cap H^{-1}(y) \in T_i\}$ and $U_{i+1} = \{(C, u) \in \mathcal{K}(X) \times U \mid F_V(C, u) \in V_i\}$, so $T_{i+1} = \{C \in \mathcal{K}(S) \mid \exists u \in U,\ (C, u) \in U_{i+1}\}$. Then $V_i = \{C \in \mathcal{K}(X) \mid \hat{H}(C, Y) \subset T_i\}$, and since $\hat{H}(C, Y) \in \mathcal{K}(\mathcal{K}(X))$, we have V_i computable in $\mathcal{O}(\mathcal{K}(\mathcal{X}))$. Then $U_{i+1} = F_V^{-1}(V_i)$, so is computable in $\mathcal{O}(\mathcal{K}(\mathcal{X}) \times \mathcal{U})$. Defining $\tilde{F}_U(C) = C \times U \in \mathcal{V}(\mathcal{K}(X) \times U)$, we obtain $\{C \in \mathcal{K}(X) \mid \exists u \in U,\ (C, u) \in U_{i+1}\} = \tilde{F}_U^{-1}(U_{i+1})$, so $T_{i+1} = \tilde{F}_U^{-1}(U_{i+1} \cap \{C \in \mathcal{K}(X) \mid C \subset S\}$ is computable in $\mathcal{O}(\mathcal{K}(\mathcal{X}))$. □

We can construct a controller solving the problem in a similar way to the case of state feedback. However, since the type of compact sets is not Hausdorff, we add an additional state k, and take the controller to be a partial functions $\mathcal{K}(\mathcal{X}) \times \mathbb{N} \rightharpoonup \mathcal{O}(\mathcal{U})$ defined on (C_k, k) when $C_k \in T_k$.

Theorem 5.28. *Let $\mathcal{X}$ be an effectively locally compact type, $\mathcal{U}$ be an effectively overt type, $\mathcal{V}, \mathcal{W}$ be effectively compact, $f : \mathcal{X} \times \mathcal{U} \times \mathcal{V} \to \mathcal{X}$, $h : \mathcal{X} \times \mathcal{W} \to \mathcal{X}$, $T, S : \mathcal{O}(\mathcal{X})$, and $X_0 : \mathcal{K}(\mathcal{X})$. If* $\text{ctrl}\,(f, h, T, S) \supset X_0$, *then there is an supervisor $G : \mathcal{K}(X) \times \mathbb{N} \rightharpoonup O(U)$ which can be computed from f, h, T, S, X_0.*

Sketch of proof. Since $X_0 \in \text{ctrl}\,(f, h, S, T)$, there exists N such that $X_0 \in T_N$, and since each T_n is open, we can effectively find such an N. We initialize the supervisor to $(\hat{X}_0, N)$ where $\hat{X}_0 = X_0 \cap H^{-1}(y)$. From controller state $(\hat{X}_i, j)$, we choose input $u_i \in G_j(\hat{X}_i)$, update the controller state to $(\hat{X}_{i+1}, j-1)$ where $\hat{X}_{i+1}$ is given by (14). It is immediate from construction that $\hat{X}_i \in T_{N-i}$, so $\hat{X}_N \in T$ as required. □

Remark 5.29. An alternative approach is to consider a fully-discrete controller, as in Collins (2008, Theorem 18). Here, the main difficulty is in moving from observers with states in $\mathcal{K}(\mathcal{X})$, which is not Hausdorff, to an element of a discrete Hausdorff space.

6. Conclusions

In this paper, we have given an exposition of a theory of rigorous computation for objects of continuous mathematics and shown that this theory is powerful enough to construct algorithms to solve a variety of problems of dynamic systems theory, including differential equations/inclusions, evolution of hybrid systems, reachability analysis, and control synthesis. By expressing the results within a simple type theory, we can give simple proofs which are similar in style to those of classical or constructive mathematics.

There are many possible directions for future work, including differentiable dynamical systems, partial differential equations, and stochastic systems. Differentiable dynamical systems are a rich field for which many powerful analytic results exists (Katok and Hasselblatt 1995). We do not expect any significant problems effectivizing much of this theory, though results of ergodic theory may be more challenging. Partial differential equations rely crucially on existence and uniqueness results on Sobolev spaces; again we do not expect significant problems. For stochastic systems, we need a theory of probability. This is more challenging, since classical probability is based on measure theory, which is not directly computable. We have shown that an approach based on *valuations* is sufficient to prove computability of solutions of stochastic differential equations (Collins 2014).

Finally, it is important to provide rigorous and efficient implementations of the operations proved computable. The tool ARIADNE (Bresolin et al. 2015) provides such functionality for reachability analysis of nonlinear hybrid systems.

Acknowledgements. This research was supported by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) Vidi grant 639.032.408.

Notes

1 The term "s-m-n" is standard and refers to letters occurring in the original statement of the theorem in Kleene (1936) rather than being an abbreviation.
2 The terminology *overt* is becoming standard in the literature. One might instead use the terminology *separable* since this is the classical property most closely related to this type.
3 The author wishes to thank the anonymous referee for pointing out the (rather obvious) Sierpinski space counterexample.
4 On (Pauly 2016, p. 170), it is claimed that this result is wrong. The counterexample given, namely $\{p \in [0,1]^\omega \mid p \text{ is not computable}\}$, is not effectively separable, since any countable dense computable sequence in $[0,1]^\omega$ has computable elements.
5 On (Pauly 2016, p. 166), it is claimed that this result is also wrong and that the one-point compactificatio of Baire space, $\mathbb{N}^\omega \cup \{\infty\}$, is a counterexample, as communicated by de Brecht. We suspect this is an example of an effectively compact space which is not effectively coverable and that the claim was due to misreading "if" for "if, and only if".

References

Aberth, O. (2007). *Introduction to Precise Numerical Methods*, 2nd edn., Amsterdam, Elsevier/Academic Press. With 1 CD-ROM (Windows).

Ariadne: A C++ library for formal verification of cyber-physical systems, using reachability analysis for nonlinear hybrid automata (Release 1.9, 2018). http://www.ariadne-cps.org/.

Asarin, E., Bournez, O., Dang, T., Maler, O. and Pnueli, A. (2000). Effective synthesis of switching controllers for linear systems. *Proceedings of the IEEE* **88** 1011–1025.

Asarin, E., Maler, O. and Pnueli, A. (1995). Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoretical Computer Science* **138** (1) 35–65.

Aubin, J.-P. and Cellina, A. (1984). *Differential Inclusions*, Grundlehren der Mathematischen Wissenschaften, vol. 264, Berlin, Springer-Verlag.

Aubin, J.-P., Lygeros, J., Quincampoix, M. and Sastry, S. (2002). Impulse differential inclusions: a viability approach to hybrid systems. *IEEE Transactions on Automatic Control* **47** (1) 2–20.

Battenfeld, I. (2008). *Topological Domain Theory*. Phd thesis, University of Edinburgh.

Battenfeld, I., Schröder, M. and Simpson, A. (2007). A convenient category of domains. In: Cardelli, L., Fiore, M. and Winskel, G. (eds.) *Computation, Meaning, and Logic: Articles dedicated to Gordon Plotkin*, Electronic Notes in Theoretical Computer Science, vol. 172, Amsterdam, Elsevier, 69–99.

Bauer, A. (2000). *The Realizability Approach to Computable Analysis and Topology*. Phd thesis, Carnegie Mellon University.

Beggs, E. J. and Tucker, J. V. (2009). Computations via Newtonian and relativistic kinematic systems. *Applied Mathematics and Computation* **215** (4) 1311–1322.

Bishop, E. and Bridges, D. (1985). *Constructive Analysis*, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], vol. 279, Berlin, Springer-Verlag.

Blanck, J. (2000). Domain representations of topological spaces. *Theoretical Computer Science* **247** (1–2) 229–255.

Blondel, V. D. and Tsitsiklis, J. N. (2000). The boundedness of all products of a pair of matrices is undecidable. *Systems & Control Letters* **41** (2) 135–140.

Bournez, O., Campagnolo, M. L., Graça, D. S. and E. Hainry. (2006). The general purpose analog computer and computable analysis are two equivalent paradigms of analog computation. In: Cai, J.-Y., Barry Cooper, S. and Li, A. (eds.) *Proceedings of the Third International Conference on Theory and Applications of Models of Computation (TAMC)*, Springer, 631–643.

Bournez, O., Graça, D. S. and Pouly, A. (2013). Turing machines can be efficiently simulated by the general purpose analog computer. In: Hubert Chan, T.-H., Lau, L. C. and Trevisan, L. (eds.) *Theory and Applications of Models of Computation*, Springer, 169–180.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Brattka, V. (1998). *Recursive and Computable Operations over Topological Structures*. Phd thesis, FernUniversität Hagen.

Brattka, V., Hertling, P. and Weihrauch, K. (2008). A tutorial on computable analysis. In: Barry Cooper, S., Löwe, B. and Sorbi, A. (eds.) *New Computational Paradigms: Changing Conceptions of What is Computable*, New York, NY, Springer, 425–491.

Brattka, V. and Presser, G. (2003). Computability on subsets of metric spaces. *Theoretical Computer Science* **305** 43–76.

Bresolin, D., Geretti, L., Villa, T. and Collins, P. (2015). An introduction to the verification of hybrid systems using ARIADNE. In: Jan van Schuppen, H. and Villa, T. (eds.) *Coordination Control of Distributed Systems*, Lecture Notes in Control and Information Sciences, vol. 456, Cham, Springer International, 339–346.

Clementino, M. M., Giuli, E. and Tholen, W. (2004). A functional approach to general topology. In: Pedicchio, M. C. and Tholen, W. (eds.) *Categorical Foundations*, Encyclopedia of Mathematics and its Applications, vol. 97, Cambridge, Cambridge University Press, 103–163.

Collins, P. (2005). Continuity and computability of reachable sets. *Theoretical Computer Science* **341** (1–3) 162–195.

Collins, P. (2007). Optimal semicomputable approximations to reachable and invariant sets. *Theory of Computing Systems* **41** (1) 33–48.

Collins, P. (2008). Computability of controllers for discrete-time semicontinuous systems. In: *Proceedings of the 18th International Symposium on the Mathematical Theory of Networks and Systems, Blacksburg, Virginia*.

Collins, P. (2011). Semantics and computability of the evolution of hybrid systems. *SIAM Journal on Control and Optimization* **49** (2) 890–925.

Collins, P. (2014). Computable stochastic processes. arXiv:1409.4667.

Collins, P. and Graça, D. (2008). Effective computability of solutions of ordinary differential equations — the thousand monkeys approach. In: *Proceedings of the 5th International Conference on Computability and Complexity in Analysis (CCA'08)*, Electronic Notes in Theoretical Computer Science, Amsterdam, The Netherlands, Elsevier, 53–64.

Collins, P. and Graça, D. S. (2009) Effective computability of solutions of differential inclusions: the ten thousand monkeys approach. *J.UCS* **15** (6) 1162–1185.

Dahlgren, F. (2007). Partial continuous functions and admissible domain representations. *Journal of Logic and Computation* **17** (6) 1063–1081.

Daniel, J. W. and Moore, R. E. (1970). *Computation and Theory in Ordinary Differential Equations*, San Francisco, W. H. Freeman & Co Ltd.

Edalat, A. (2009). A computable approach to measure and integration theory. *Information and Computation* **207** (5) 642–659.

Edalat, A. and Sünderhauf, P. (1999). A domain-theoretic approach to computability on the real line. *Theoretical Computer Science* **210** (1) 73–98.

Engelking, R. (1989). *General Topology*, Sigma Series in Pure Mathematics, vol. 6, Berlin, Heldermann.

Ershov, Ju. L. (1973). Theorie der numerierungen i. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* **19** (4) 289–388.

Ershov, Ju. L. (1975). Theorie der numerierungen ii. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* **21** (6) 473–584.

Ershov, Ju. L. (1977). Theorie der numerierungen iii. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* **23** (4) 289–371.

Escardó, M. (2004). Synthetic topology of data types and classical spaces. *Electronic Notes in Theoretical Computer Science* **87** 21–156.

Escardó, M. and Heckmann, R. (2002). Topologies on spaces of continuous functions. *Topology Proceedings* **26** (2) 545–564.

Escardó, M., Lawson, J. and Simpson, A. (2004). Comparing Cartesian closed categories of (core) compactly generated spaces. *Topology and its Applications* **143** (1–3) 105–145.

Filippov, A. F. (1988). *Differential Equations with Discontinuous Righthand Sides*, Mathematics and Its Applications, Dordrecht, Kluwer Academic Publishers.

Franklin, S. P. (1965). Spaces in which sequences suffice. *Fundamenta Mathematicae* **57** (1) 107–115.

Fränzle, M. (1999). Analysis of hybrid systems: an ounce of realism can save an infinity of states. In: Flum, J. and Rodriguez-Artalejo, M. (eds.) *Computer Science Logic*, Lecture Notes in Computer Science, vol. 1683, Berlin, Heidelberg, New York, Springer-Verlag.

Gabor, G. (2007). Continuous selection of the solution map for one-sided Lipschitz differential inclusions. *Nonlinear Analysis* **66** (5) 1185–1197.

Gierz, G., Hofmann, K. H., Keimel, K., Lawson, J. D., Mislove, M. W. and Scott, D. S. (1980). *A Compendium of Continuous Lattices*, Berlin, Springer-Verlag.

Gierz, G., Hofmann, K. H., Keimel, K., Lawson, J. D., Mislove, M. and Scott, D. S. (2003). *Continuous Lattices and Domains*, Encyclopedia of Mathematics and its Applications, vol. 93, Cambridge, Cambridge University Press.

Goebel, R. and Teel, A. R. (2006). Solutions to hybrid inclusions via set and graphical convergence with stability theory applications. *Automatica - A Journal of IFAC* **42** (4) 573–587.

Grzegorczyk, A. (1957). On the definitions of computable real continuous functions. *Fundamenta Mathematicae* **44** 44–61.

Hansen, E. and William Walster, G. (2004). *Global Optimization using Interval Analysis*, 2nd edn., Monographs and Textbooks in Pure and Applied Mathematics, vol. 264, New York, Marcel Dekker Inc.

https://doi.org/10.1017/S096012952000002X Published online by Cambridge University Press

Hauck, J. (1980). Konstruktive Darstellungen in topologischen Räumen mit rekursiver Basis. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* **26** 565–576.

Hauck, J. (1981). Berechenbarkeit in topologischen Räumen mit rekursiver Basis. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* **27** 473–480.

Hertling, P. (1999). A real number structure that is effectively categorical. *Mathematical Logic Quarterly* **45** (2) 147–182.

Hofmann, K. H. and Lawson, J. D. (1978). The spectral theory of distributive continuous lattices. *Transactions of the American Mathematical Society* **246** 285–310.

Hofmann, K. and Mislove, M. (1981). Local compactness and continuous lattices. In: *Continuous Lattices*, vol. 871, Lecture Notes in Mathematics, Berlin, Heidelberg, Springer, 209–248.

Isbell, J. R. (1975). Function spaces and adjoints. *Mathematica Scandinavica* **36** 317–339.

Jaulin, L., Kieffer, M., Didrit, O. and Walter, É. (2001). *Applied Interval Analysis*, London, Springer-Verlag London Ltd. With examples in parameter and state estimation, robust control and robotics, With 1 CD-ROM (UNIX, Sun Solaris).

Johnstone, P. T. (2002a). *Sketches of an Elephant: a Topos Theory Compendium. Volume 1*, Oxford Logic Guides, vol. 43. New York, The Clarendon Press, Oxford University Press.

Johnstone, P. T. (2002b). *Sketches of an Elephant: a Topos Theory Compendium. Volume 2*, Oxford Logic Guides, vol. 44, Oxford, The Clarendon Press, Oxford University Press.

Katok, A. and Hasselblatt, B. (1995). *Introduction to the Modern Theory of Dynamical Systems*, Encyclopedia of Mathematics and its Applications, vol. 54. Cambridge, Cambridge University Press.

Keimel, K. and Paseka, J. (1994). A direct proof of the Hofmann-Mislove theorem. *Proceedings of the American Mathematical Society* **120** (1) 301–303.

Kleene, S. C. (1936). General recursive functions of natural numbers. *Mathematische Annalen* **112** (1) 727–742.

Ko, K.-I. (1991). *Complexity Theory of Real Functions*, Progress in Theoretical Computer Science, Boston, MA, Birkhäuser Boston Inc.

Kushner, B. A. (1999). Markov's constructive analysis; a participant's view. *Theoretical Computer Science* **219** (1–2) 267–285. Computability and complexity in analysis (Castle Dagstuhl, 1997).

Luckhardt, H. (1977). A fundamental effect in computations on real numbers. *Theoretical Computer Science* **5** (3) 321–324.

Martin-Löf, P. (1984). *Intuitionistic Type Theory*, Napoli, Bibliopolis.

Mazur, S. (1963). *Computable Analysis*, Razprawy Matematyczne, vol. 33, Warszawa, Państwowe Wydawnictwo Naukowe.

Moore, R. E., Baker Kearfott, R. and Cloud, M. J. (2009). *Introduction to Interval Analysis*, Philadelphia, PA, Society for Industrial and Applied Mathematics (SIAM).

O'Connor, R. and Spitters, B. (2010). A computer-verified monadic functional implementation of the integral. *Theoretical Computer Science* **411** (37) 3386–3402.

Pauly, A. (2016). On the topological aspects of the theory of represented spaces. *Computability* **5** (2) 159–180.

Pauly, A. and de Brecht, M. (2015). Descriptive set theory in the category of represented spaces. In: *Proceedings of the 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, 438–449.

Pour-El, M. B. and Ian Richards, J. (1989). *Computability in Analysis and Physics*, Perspectives in Mathematical Logic, Berlin, Springer-Verlag.

Puri, A., Varaiya, P. and Borkar, V. (1996). Epsilon-approximation of differential inclusions. In: Alur, R., Henzinger, T. A. and Sontag, E. D. (eds.) *Hybrid Systems III*, *LNCS*, vol. 1066, Berlin, Springer, 362–376.

Ruohonen, K. (1996). An effective Cauchy-Peano existence theorem for unique solutions. *International Journal of Foundations of Computer Science* **7** (2) 151–160.

Saint-Pierre, P. (1994). Approximation of the viability kernel. *Applied Mathematics & Optimization* **29** (2) 187–209.

Schröder, M. (2002a). *Admissible Representations for Continuous Computations*. Phd thesis, Fachbereich Informatik, FernUniversität Hagen.

Schröder, M. (2002b). Extended admissibility. *Theoretical Computer Science* **284** (2) 519–538. Computability and complexity in analysis (Castle Dagstuhl, 1999).

Sipser, M. (1996). *Introduction to the Theory of Computation*, Boston, Course Technology, International Thomson.

Steen, L. A. and Seebach, J. A. (1978). *Counterexamples in Topology*, 2nd edn., New York, Springer-Verlag.

Taylor, P. (2008). A lambda calculus for real analysis. http://www.monad.me.uk/.

Tomlin, C., Lygeros, J. and Sastry, S. (2000). A game theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE* **88** (7), 949–970.

Turing, A. M. (1937) On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society (2)* **43** (1) 230–265.

Troelstra, A. S. and van Dalen, D. (1988). *Constructivism in Mathematics: An Introduction, Volume 1*, Studies in Logic and the Foundations of Mathematics, vol. 121, Amsterdam, North-Holland.

van Stigt, W. P. (1990). *Brouwer's Intuitionism*, Studies in the History and Philosophy of Mathematics, vol. 2, Amsterdam, North-Holland Publishing Co.

Vickers, S. (1989). *Topology via Logic*, Cambridge Tracts in Theoretical Computer Science, vol. 5, Cambridge, Cambridge University Press.

Weihrauch, K. (2000). *Computable Analysis: An Introduction*, Texts in Theoretical Computer Science. An EATCS Series, Berlin, Springer-Verlag.

Weihrauch, K. and Grubba, T. (2009). Elementary computable topology. *J.UCS* **15** (6) 1381–1422.
Ziegler, M. (2009). Physically-relativized Church-Turing Hypotheses: physical foundations of computing and complexity theory of computational physics. *Applied Mathematics and Computation* **215** (4) 1431–1447.

A. Appendix

A.1 Summary of computable types and operations

Predicates:

Element $\chi_S(x) \iff x \in S$.

Overlap $S \between U \iff \exists\, x \in S,\ x \in U \iff A \cap U \neq \emptyset$.

Subset $S \subset U \iff \forall\, x \in S,\ x \in U$.

General constructions:

Terminal $\mathcal{I}$.

Element $\mathcal{X} \equiv \mathcal{I} \to \mathcal{X}$.

Subtype $\{x \in \mathcal{X} \mid p(x)\}$.

Sum $\mathcal{X}_1 + \mathcal{X}_2$ with inclusions $i_{1,2} : \mathcal{X}_{1,2} \to \mathcal{X}_1 + \mathcal{X}_2$.

Product $\mathcal{X}_1 \times \mathcal{X}_2$ with projections $p_{1,2} : \mathcal{X}_1 \times \mathcal{X}_2 \to X_{1,2}$.

Exponential $\mathcal{Y}^{\mathcal{X}} \equiv (\mathcal{X} \to \mathcal{Y})$; $\mathcal{Y}^{\mathcal{A}\times\mathcal{X}} \equiv (\mathcal{Y}^{\mathcal{X}})^{\mathcal{A}}$ (or $\mathcal{A} \times \mathcal{X} \to \mathcal{Y} \equiv \mathcal{A} \to \mathcal{Y}^{\mathcal{X}}$) with evaluation $\varepsilon : \mathcal{Y}^{\mathcal{X}} \times \mathcal{X} \to \mathcal{Y}$.

Countable types:

Binary Words $\{0, 1\}^*$.

Boolean $\mathcal{B} := \{\top, \bot\}$.

Sierpinskian $\mathcal{S} := \{\top, \uparrow\}$.

Kleenean $\mathcal{K} := \{\top, \bot, \uparrow\}$.

Naturals $\mathcal{N} := \{0, 1, 2, \ldots\}$.

Integers $\mathcal{Z} := \{\ldots, -2, -1, 0, 1, 2, \ldots\}$.

Rationals $\mathcal{Q} := \{m/n \mid m \in \mathbb{Z},\ n \in \mathbb{N},\ \text{and } n \neq 0\}$.

Uncountable types:

Cantor space $\{0, 1\}^\omega$.

Baire space $\mathcal{N}^\omega$.

Real numbers $\mathcal{R} := \{\vec{x} \in \mathcal{Q}^{\mathcal{N}} \mid \mathrm{abs}(x_m - x_n) \leq 2^{-\min(m,n)},\quad x \sim y \iff \mathrm{abs}(x_n - y_n) \leq 2^{1-n}\}$.

Continuous functions $\mathcal{C}(\mathcal{X};\mathcal{Y}) := \mathcal{Y}^{\mathcal{X}}$.

Linear functions $\mathcal{L}(\mathcal{V};\mathcal{W}) := \{f \in \mathcal{C}(\mathcal{V};\mathcal{W}) \mid f(x + sy) = f(x) + sf(y)\}$.

Measures (weak) $\mathcal{M}(\mathcal{X}) := \{\mu \in \mathcal{L}(\mathcal{C}(\mathcal{X};\mathcal{R});\mathcal{R}) \mid f \geq 0 \implies \mu(f) \geq 0\}$.

Open sets $\mathcal{O}(\mathcal{X}) :\simeq \mathcal{S}^{\mathcal{X}}$; $\quad U \simeq \chi_U$.

Closed sets $\mathcal{A}(\mathcal{X}) :\simeq \mathcal{S}^{\mathcal{X}}$; $\quad A \simeq \chi_{X\setminus A}$.

Overt sets $\mathcal{V}(\mathcal{X}) := \{(A \between \cdot) \in \mathcal{S}^{\mathcal{O}(\mathcal{X})} \mid A \between (U \cup V) \iff A \between U \vee A \between V\}$.

Compact sets $\mathcal{K}(\mathcal{X}) := \{(C \subset \cdot) \in \mathcal{S}^{\mathcal{O}(\mathcal{X})} \mid C \subset (U \cap V) \iff C \subset U \wedge C \subset V\}$.

Flow $\{\phi \in \mathcal{C}(\mathcal{X}; \mathcal{C}(\mathcal{R}; \mathcal{X})) \equiv \mathcal{C}(\mathcal{X} \times \mathcal{R}; \mathcal{X}) \mid \phi(x, s+t) = \phi(\phi(x, s), t)\}$.

Logic:

Unprovability $\mathcal{B} \to \mathcal{S}, \perp \mapsto \uparrow$.

Countable disjunction $\mathcal{S}^{\mathcal{N}} \to \mathcal{S}, (a_n)_{n \in \mathbb{N}} \mapsto \bigvee_{n \in \mathbb{N}} a_n$.

Finite conjunction $\mathcal{S} \times \mathcal{S} \to \mathcal{S}, (a_1, a_2) \mapsto a_1 \wedge a_2$.

Arithmetic:

Unit $1 \in \mathcal{R}$.

Addition $(x, y) \mapsto x + y : \mathcal{R} \times \mathcal{R} \to \mathcal{R}$.

Subtraction $(x, y) \mapsto x - y : \mathcal{R} \times \mathcal{R} \to \mathcal{R}$.

Multiplication $(x, y) \mapsto x \times y : \mathcal{R} \times \mathcal{R} \to \mathcal{R}$.

Division $(x, y) \mapsto x \div y : \mathcal{R} \times (\mathcal{R} \setminus 0) \to \mathcal{R}$.

Comparison $x \mapsto \operatorname{sgn}(x) : \mathcal{R} \to \mathcal{K}$.

Limit $(x_n)_{n \in \mathbb{N}} \mapsto \lim_{n \to \infty} x_n : \mathcal{R}^{\mathcal{N}} \rightharpoonup \mathcal{R}$ if $\exists \epsilon : \mathbb{N} \to \mathbb{Q}^+$ such that $\lim_{n \to \infty} \epsilon_n = 0$ and $|x_m - x_n| \leq \epsilon_{\min(m,n)}$; in particular $\epsilon_n = 2^{-n}$.

Sets:

Complement $\mathcal{O} \leftrightarrow \mathcal{A}$.

Finite union $\mathcal{O} \times \mathcal{O} \to \mathcal{O}, \mathcal{A} \times \mathcal{A} \to \mathcal{A}, \mathcal{V} \times \mathcal{V} \to \mathcal{V}, \mathcal{K} \times \mathcal{K} \to \mathcal{K}$.

Countable union $\mathcal{O}^{\mathcal{N}} \to \mathcal{O}, \mathcal{V}^{\mathcal{N}} \to \mathcal{V}$.

Finite intersection $\mathcal{O} \times \mathcal{O} \to \mathcal{O}, \mathcal{A} \times \mathcal{A} \to \mathcal{A}, \mathcal{V} \times \mathcal{O} \to \mathcal{V}, \mathcal{K} \times \mathcal{A} \to \mathcal{K}$.

Countable intersection $\mathcal{A}^{\mathcal{N}} \to \mathcal{A}$. ($\mathcal{K}^{\mathcal{N}} \to \mathcal{K}$ can be derived if $\mathcal{X}$ is Hausdorff.)

Singleton $\mathcal{X} \to \mathcal{V}(\mathcal{X}), \mathcal{X} \to \mathcal{K}(\mathcal{X})$.

Closure $\mathcal{O}(\mathcal{X}) \to \mathcal{V}(\mathcal{X})$ if $\mathcal{X}$ is effectively overt.

Saturation $\mathcal{K}(\mathcal{X}) \to \mathcal{A}(\mathcal{X})$ if $\mathcal{X}$ is effectively Hausdorff.

Evaluation $\mathcal{C}(\mathcal{X}; \mathcal{Y}) \times \mathcal{X} \to \mathcal{Y}$.

Preimage $\mathcal{C}(\mathcal{X}; \mathcal{Y}) \times \mathcal{O}(\mathcal{Y}) \to \mathcal{O}(\mathcal{X}), \mathcal{C}(\mathcal{X}; \mathcal{Y}) \times \mathcal{A}(\mathcal{Y}) \to \mathcal{A}(\mathcal{X})$.

Image $\mathcal{C}(\mathcal{X}; \mathcal{Y}) \times \mathcal{V}(\mathcal{X}) \to \mathcal{V}(\mathcal{Y}), \mathcal{C}(\mathcal{X}; \mathcal{Y}) \times \mathcal{K}(\mathcal{X}) \to \mathcal{K}(\mathcal{Y})$.

Element $\mathcal{X} \times \mathcal{O}(\mathcal{X}) \to \mathcal{S}$.

Overlap $\mathcal{V}(\mathcal{X}) \times \mathcal{O}(\mathcal{X}) \to \mathcal{S}$.

Subset $\mathcal{K}(\mathcal{X}) \times \mathcal{O}(\mathcal{X}) \to \mathcal{S}$.

Cite this article: Collins P (2020). Computable analysis with applications to dynamic systems. *Mathematical Structures in Computer Science* **30**, 173–233. https://doi.org/10.1017/S096012952000002X