Available online at www.sciencedirect.com
SCIENCE DIRECT®

Theoretical Computer Science 322 (2004) 85–136

Theoretical Computer Science

www.elsevier.com/locate/tcs

Recursion and topology on $2^{\leqslant\omega}$ for possibly infinite computations

Verónica Becher[a], Serge Grigorieff[b,*]

[a]Departamento de Computación, FCEyN, Universidad de Buenos Aires, Argentina
[b]LIAFA, Université Paris 7, France

Abstract

In the context of possibly infinite computations yielding finite or infinite (binary) outputs, the space $2^{\leqslant\omega} = 2^* \cup 2^\omega$ appears to be one of the most fundamental spaces in Computer Science. Though underconsidered, next to 2^ω, this space can be viewed (Section 3.5.2) as the simplest compact space native to computer science.

In this paper we study some of its properties involving topology and computability.

Though $2^{\leqslant\omega}$ can be considered as a computable metric space in the sense of computable analysis, a direct and self-contained study, based on its peculiar properties related to words, is much illuminating.

It is well known that computability for maps $2^\omega \to 2^\omega$ reduces to continuity with recursive modulus of continuity. With $2^{\leqslant\omega}$, things get less simple. Maps $2^\omega \to 2^{\leqslant\omega}$ or $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ induced by input/output behaviours of Turing machines on finite or infinite words—which we call semicomputable maps—are not necessarily continuous but merely lower semicontinuous with respect to the prefix partial ordering on $2^{\leqslant\omega}$. Continuity asks for a stronger notion of computability.

We prove for (semi)continuous and (semi)computable maps $F:\mathscr{I}\to\mathscr{O}$ with $\mathscr{I},\mathscr{O}\in\{2^\omega,2^{\leqslant\omega}\}$ a detailed representation theorem (Theorem 81) via functions $f:2^*\to 2^*$ following two approaches: bottom-up from f to F and top-down from F to f.

1. Introduction

1.1. Mixing finite and infinite sequences

Infinite computations on a Turing machine yielding infinite outputs are considered in recursive analysis in order to get the notion of computable map $2^\omega \to 2^\omega$. However,

[*] Corresponding author. UFR d'Informatique, Universite Paris 7, 2 Place Jussieu, Paris Cedex 05 F-75251, France.

E-mail addresses: vbecher@dc.uba.ar (V. Becher), seg@liafa.jussieu.fr (S. Grigorieff).

0304-3975/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2004.03.026

as observed by Turing in his fundamental paper of 1936 [43], in general an infinite computation may yield either a finite or an infinite output.

This leads to consider the space $2^{\leqslant\omega} = 2^* \cup 2^\omega$ of all finite and infinite binary sequences and maps with range in $2^{\leqslant\omega}$. In the context of possibly infinite computations, this space is indeed a fundamental space in Computer Science.

There is a natural zero-dimensional (i.e. with a topological basis of closed open sets) compact topology on $2^{\leqslant\omega}$ which induces the expected discrete and Cantor topologies on the subspaces 2^* and 2^ω (cf. Section 3.1). Next to the Cantor space, $2^{\leqslant\omega}$ is one of the simplest examples of compact zero-dimensional space: its characterization via Pierce duality [30] (cf. Section 3.5.2), involves a 4 elements topological boolean algebra, whereas 2^ω is associated to the trivial 2 elements boolean algebra. As higher-order spaces built on discrete spaces, such zero-dimensional compact spaces can be seen as *native* to computer science. Which gives $2^{\leqslant\omega}$ a prominent role next to 2^ω.

Up to our knowledge, this topological space was not explicitly considered up to the papers by Boasson and Nivat [4] and Head [18,19], in which it is studied in view of the representation of "adherences" in 2^ω of regular languages of finite words (i.e. languages recognizable by finite automata). Staiger [40,41] and Perrin and Pin [29], also consider this space in the same perspective of formal language theory. In relation with Wadge games, Duparc [10] also uses that space, viewed as a subspace of the Baire space $\mathbb{N}^\mathbb{N}$.

Redziejowski [31] introduced another topology on $2^{\leqslant\omega}$ to restrict convergent sequences to monotonous ones (cf. Section 3.6.2).

In the perspective of higher-order recursion, Weihrauch [47, pp. 328–329] considered on $2^{\leqslant\omega}$ a variant of the compact topology which is non-Hausdorff but merely $\mathscr{T}_0$ (cf. Section 3.6.1).

As is well-known, the $X \mapsto X^\delta$ operation

$$X^\delta = \{\alpha \in 2^\omega : \{n : \alpha \upharpoonright n \in X\} \text{ is infinite}\}$$

maps subsets of 2^* onto the family of G_δ subsets of the Cantor space 2^ω.

Building on that fact, Staiger [41] proposed not to try to construct a topology on $2^{\leqslant\omega}$ but to consider, along with the Cantor topology on 2^ω, the topology on 2^* such that $X \subseteq 2^*$ is open (resp. closed) if and only if so is X^δ in 2^ω. The associated Borel hierarchy on 2^* collapses: G_δ sets are open and F_σ sets are closed.

In this paper, we shall mainly stick to the natural zero-dimensional compact topology on $2^{\leqslant\omega}$.

1.2. Topology and computability of subsets of $2^{\leqslant\omega}$

If for finite objects (integers, words, etc.) the notion of computable set is "context-insensitive", this is no more the case with infinite objects. Considering computability theory over the Cantor space 2^ω or over $\{0,1,2\}^\omega$ or over the Baire space ω^ω changes the computability status of some sets. For instance, 2^ω is trivially computable as a subset of 2^ω but not as a subset of $\{0,1,2\}^\omega$ nor ω^ω: one cannot check in finite time if an infinite word does contain a letter different from $0,1$. This is merely a Π^0_1 subset.

In fact, higher order "context-insensitivity" only starts at level Π^0_1, a fact related to the topological background of higher-order computability.

This is to say that computability over $2^{\leqslant\omega}$ *does not* reduce to computability over 2^* and over 2^ω.

As is the case with the Cantor space 2^ω, subsets of $2^{\leqslant\omega}$ which are open and closed are very simple and constitute the natural class of computable subsets of $2^{\leqslant\omega}$ (cf. Sections 3.4 and 4.1).

This class, which is here defined in a direct way, coincides with that obtained from the general theory of representations of "computable" metric spaces, cf. Kreitz and Weihrauch [50] and Weihrauch [48].

Contrary to a priori expectation, not every recursive subset of 2^* is the trace on 2^* of a computable subset of $2^{\leqslant\omega}$. Worse, such traces form a very special subfamily of rational (i.e. regular) sets of words, which we call checkable sets (cf. Section 4.5).

As mentioned above, this is only from level Π^0_1 of the effective Borel hierarchy that reasonable expectation turns true (cf. Sections 4.2–4.4).

1.3. (Semi)computability of maps in $2^{\leqslant\omega}$

With a fixed Turing machine, one can consider various types of finite or possibly infinite computations, depending on whether infinite inputs and/or infinite outputs are allowed. This has been thoroughly investigated in Wagner [44], Wagner and Staiger [45] Staiger [39,42] and Engelfriet and Hoogeboom [11].

In Section 5, we review types of possibly infinite computations adapted to $2^{\leqslant\omega}$ and introduce notions of semicomputability and computability for maps $\mathscr{I} \to \mathscr{O}$ where $\mathscr{I}$ and $\mathscr{O}$ are 2^ω or $2^{\leqslant\omega}$. We also determine the syntactical complexity of such maps.

As recalled above, computable maps $2^\omega \to 2^\omega$ are exactly input/output behaviours of Turing machines on infinite inputs which have infinite outputs. Also, computable maps are continuous and every continuous map $2^\omega \to 2^\omega$ can be obtained as the extension to 2^ω of some monotone increasing function $2^* \to 2^*$ (cf. Theorem 80). As concerns the space $2^{\leqslant\omega}$, things get less simple. Maps $2^\omega \to 2^{\leqslant\omega}$ or $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ which correspond to input/output behaviours of Turing machines on finite or infinite inputs—which we call semicomputable maps—are not necessarily continuous but merely lower semicontinuous with respect to the prefix partial ordering on $2^{\leqslant\omega}$. To get continuity, one has to consider the stronger notion of computable map which asks that the machine halts in case its output is finite. Section 6 is devoted to a study of lower semicontinuity.

We prove for (semi)continuous and (semi)computable maps a detailed representation theorem (Theorem 81) via functions $2^* \to 2^*$ satisfying pertinent conditions. Prior to the proof of this theorem, we have to introduce (cf. Section 7) some consequent material about top-down and bottom-up representation of maps $\mathscr{I} \to \mathscr{O}$ ($\mathscr{I}, \mathscr{O} \in \{2^\omega, 2^{\leqslant\omega}\}$) via maps $2^* \to 2^*$. This material is also interesting on its own (cf. Sections 7.5 and 7.6).

Whereas every map (resp. semicomputable map) $2^* \to 2^*$ is trivially the trace of a semicontinuous (resp. semicomputable) map $2^{\leqslant\omega} \to 2^{\leqslant\omega}$, traces of continuous and computable maps $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ form very special classes which can be viewed as the checkable versions of continuity and computability, cf. Section 9.

1.4. Relation with computable analysis and type two theory of effectivity

As witnessed by the title of Turing's foundational paper [43] "On computable numbers…", computability on higher-order structures like $\mathbb{R}$ has been considered since the origin of computability theory. A considerable lot of work has been done on what is called Computable Analysis or Type Two theory of Effectivity, cf. the CCA Net-Bibliography available on the web [7], maintained by Vasco Brattka.

We already noticed that computability of higher-order objects (such as sets of infinite words) does depend on the context in which computability is considered.

Another fundamental problem related to computability on $\mathbb{R}$ or usual topological spaces is that it does depend on the chosen representation of elements. For instance, as is well-known, a real can be represented via a Cauchy sequence of rationals, via its left or right Dedekind cut, via its binary expansion or its Farey sequence. It turns out that all these approaches lead to the same notion of computable real. However, they lead to different notions of sequences of reals [27]. The "best" representations are that with Cauchy sequences and that with Avizienis binary expansions using digits $-1, 0, 1$.

Going to functions over reals or more complex objects like functionals asks for much care. Among the main authors who dealt first with these problems, let us cite Grzegorczyk [15–17] and Lacombe [22]. Extensions to general "computable structures" have been proposed by Lacombe [23], Ershov [12], Weihrauch [50,48], Kreitz and Weihrauch [46]. Cf. Weihrauch's books [47,49].

Computability of solutions of partial differential equations has been investigated. Up to distributions, Zhong and Weihrauch [52].

A Kleene like (i.e. closure of a family of basic functions by some operators) development of computability over reals has been introduced by Brattka [6].

Higher-order complexity theory has also been developed, cf. Kreitz and Weihrauch [51].

With 2^ω and $2^{\leqslant\omega}$ the representation problem *just vanishes*: a finite or infinite word is obviously to be represented by itself. This allows for a much easier and entirely self-contained access to (semi)computable maps as done in Section 5.2.

1.5. Notations

We denote $\mathbb{N}$ the set of natural numbers, and we work with the binary alphabet $\{0, 1\}$. As usual, a string is a finite sequence of elements of $\{0, 1\}$,

2^* is the set of all strings and λ is the empty string.
2^ω is the set of all infinite sequences of $\{0,1\}$, i.e. the Cantor space,
$2^{\leqslant\omega} = 2^* \cup 2^\omega$ is the set of all finite or infinite sequences of $\{0,1\}$.

For $a \in 2^*$, $|a|$ denotes the length of a.

If $a \in 2^*$ and $\alpha \in 2^\omega$ we denote $a \upharpoonright n$ the prefix of a with length $\min(n,|a|)$ and $\alpha \upharpoonright n$ the length n prefix of the infinite sequence α.

If $f : 2^* \to 2^*$ is a partial function then, as usual, we write $f(p)\!\downarrow$ when the function is defined, and $f(p)\!\uparrow$ otherwise. To deal with program inputs we consider a recursive bijection $\langle .,. \rangle : 2^* \times 2^* \to 2^*$, and we use the convention

$$f(p, s_1, s_2, \ldots, s_n) = f(\langle p, \langle s_1, \ldots \langle s_{n-1}, s_n \rangle \ldots \rangle \rangle).$$

2. Around sets of words

2.1. Prefix free sets of words

We recall some classical material around the prefix ordering on words.

We write $a \preccurlyeq b$ if a is a prefix of b, and $a \prec b$ if a is a proper prefix of b. We assume the recursive bijection *string*: $\mathbb{N} \to 2^*$ such that *string*(i) is the ith string in the length-lexicographic order over 2^*. Observe that $(i,k) \mapsto string^{-1}(string(i) \upharpoonright k)$ and $i \mapsto |string(i)|$ are recursive.

Definition 1. (1) $X \subseteq 2^*$ is *prefix-free* if and only if no proper extension of an element of the set belongs to the set, i.e.

$$\forall a, b \in 2^* \quad (a \in X \text{ and } b \neq \lambda \Rightarrow ab \notin X).$$

For example, the set $\{\lambda\}$ is prefix-free and so is $\{0^n 1 : n \geqslant 1\}$.

(2) $\min(X)$ denotes the prefix-free set consisting of all minimal elements of X with respect to the prefix-ordering $\preccurlyeq$.

If $X \subseteq 2^*$ then $X2^\omega$ denotes the open subset of 2^ω whose elements have an initial segment in X. For example, is $s \in 2^*$ is a particular string then $s2^\omega$ is the set of all sequences starting with s. The following proposition is straightforward (point 2 is to be compared with Proposition 3).

Proposition 2. *Let $X, Y, Z \subseteq 2^*$.*

1. $\min(X)$ *is the unique prefix-free set Y such that $X2^* = Y2^*$.*
2. $X2^* = 2^*$ *if and only if $\lambda \in X$. Hence, $\{\lambda\}$ is the unique prefix-free set X satisfying the previous equation.*
3. $X2^* \subseteq Y2^*$ *if and only if $X \subseteq Y2^*$.*
4. *Let X, Y be prefix-free. Then $X2^\omega \subseteq Y2^\omega$ if and only if $\exists Z$ (Z is finite $\wedge X2^* \subseteq Z \cup Y2^*$).*

Proof. Points 1–3 are straightforward.

Point 4 $\Rightarrow$. If $X2^*\setminus Y2^*$ were infinite then, by König's lemma, it would contain an infinite branch, hence there would be $\alpha \in 2^\omega$ with infinitely many segments in $X2^*$ but not in $Y2^*$. Therefore α would be in $X2^\omega \setminus Y2^\omega$, a contradiction. Hence $X2^*\setminus Y2^*$ is finite, which means $X2^* \subseteq Z \cup Y2^*$ for some finite Z.

Point 4 $\Leftarrow$. Observe that for n greater than the longest string in Z, we must have $X2^n \subset Y2^*$. Whence $X2^\omega \subseteq Y2^\omega$. □

A prefix-free set $X \subset 2^*$ is maximal if $X \cup \{a\}$ is no more prefix-free whenever $a \notin X$.

If $X \subset 2^*$ is prefix-free and every sequence $\alpha \in 2^\omega$ has an initial segment in X then X is maximal. The converse is not true: $\{1\}^*0$ is maximal prefix-free but contains no prefix of the sequence 1^ω. In fact, a simple application of König's Lemma proves that finiteness is required.

Proposition 3. *Let $X \subseteq 2^*$ and $u \in 2^*$.*
1. *$X2^\omega = 2^\omega$ (i.e. every infinite sequence has a prefix in X) if and only if X contains a finite maximal prefix-free set Z.*
2. *If X is prefix-free then $X2^\omega = 2^\omega$ if and only if X is finite and maximal prefix-free.*
3. *$s2^\omega \subseteq X2^\omega$ if and only if*
   - *either s extends some element of X,*
   - *or there exists some finite maximal prefix-free set Z such that $sZ \subseteq X$.*

Proof. The $\Leftarrow$ direction of Point 1 is easy. For the $\Rightarrow$ direction, suppose $X \subseteq 2^*$ contains no finite maximal prefix-free and define inductively an infinite sequence α such that for all $n \in \mathbb{N}$ the set

$$X^{(n)} = \{u \in 2^*:\ \alpha \restriction n\ u \in X\}$$

contains no finite maximal prefix-free set. Equality $X2^\omega = 2^\omega$ insures $\alpha \in X2^\omega$, hence there is an n such that $\alpha \restriction n \in X$. Whence, $\lambda \in X^{(n)}$ and the singleton set $\{\lambda\}$ is a finite maximal prefix-free subset of $X^{(n)}$. A contradiction.

Point 2 is a straightforward corollary of Point 1.

As for point 3, let $Y = \{u \in 2^*:\ su \in X\}$. If $s \notin X2^*$ then

$$\begin{aligned} s2^\omega \subseteq X2^\omega &\Leftrightarrow Y2^\omega = 2^\omega \\ &\Leftrightarrow \exists Z\ (Z \subseteq Y \text{ and } Z \text{ is finite maximal prefix-free}) \\ &\Leftrightarrow \exists Z\ (sZ \subseteq X \text{ and } Z \text{ is finite maximal prefix-free}). \qquad \square \end{aligned}$$

Definition 4. If $X \subseteq 2^*$ we let

$$\widehat{X} = \{s \in 2^*:\ \exists Z \text{ finite maximal prefix free s.t. } sZ \subseteq X\}.$$

Proposition 5. (1) *$X \subseteq \widehat{X} = \widehat{\widehat{X}}$. For all $x \in \widehat{X}$ there exists $n \in \mathbb{N}$ such that $x2^n2^* \subseteq X2^*$. In particular, $\widehat{X}2^\omega = X2^\omega$.*

(2) $\widehat{X}2^* = \widehat{X} \cup X2^* = \{s \in 2^*:\ s2^\omega \subseteq X2^\omega\}$.

In particular,

(i) $X2^\omega \subseteq Y2^\omega \Leftrightarrow \widehat{X}2^* \subseteq \widehat{Y}2^*$.

(ii) *If* $X2^* = X$ *then* $\widehat{X} = \{s \in 2^* : s2^\omega \subseteq X2^\omega\}$.

(iii) *If* $X2^* = X$ *and* $Y2^* = Y$ *then* $X2^\omega \subseteq Y2^\omega \Leftrightarrow \widehat{X} \subseteq \widehat{Y}$.

(3) *If* Y *is prefix-free then* $\widehat{X} = \widehat{Y}$ *if and only if* $Y = \bigcup_{s \in \min(\widehat{X})} sZ_s$ *where the* Z_s*'s are finite maximal prefix-free sets.*

In particular, if $X \neq \emptyset$ *then there are infinitely many prefix-free sets* $Y \subset 2^*$ *such that* $X2^\omega = Y2^\omega$.

(4) *The following conditions are equivalent*:

– *there exists a finite prefix-free set* Y *such that* $X2^\omega = Y2^\omega$,

– $\min(\widehat{X})$ *is finite.*

– *every prefix-free set* Y *such that* $X2^\omega = Y2^\omega$ *is finite.*

Proof. Point 1 is straightforward. Points 2 and 3 are easy consequences of Proposition 3. Point 4 is a corollary of point 3. □

2.2. Checkable sets of words

In this section, we introduce a particular notion of regular set of words which proves useful to characterize traces over 2^* of computable subsets of $2^{\leqslant\omega}$ (cf. Proposition 36) and of continuous maps on $2^{\leqslant\omega}$ (cf. Section 9.1).

Definition 6 (Checkable sets). (1) $Z \subseteq 2^*$ is *checkable* if $Z = X \cup Y2^*$ for some finite sets $X, Y \subset 2^*$.

(2) Let $\vec{\mathbb{X}}$ be any finite product of spaces $\mathbb{N}$ and/or 2^*. A set $Z \subset \vec{\mathbb{X}} \times 2^*$ is *checkable* (resp. *recursively checkable*) relative to its last component if there exist sets (resp. recursive sets) $X, Y \subset \vec{\mathbb{X}} \times 2^*$ such that for all $\vec{x} \in \vec{\mathbb{X}}$ the slices $X_{\vec{x}}, Y_{\vec{x}}$ are finite and $Z_{\vec{x}} = X_{\vec{x}} \cup Y_{\vec{x}}2^*$.

(3) Z is *simply checkable* (resp. *simply recursively checkable*) if it is checkable (resp. recursively checkable) and closed by extension relative to its last component, i.e. the X set in Point 1 is empty or the $X_{\vec{x}}$'s sets in Point 2 are all empty.

Definition 7. For $Z \subseteq 2^*$ we set $core(Z) = \{s \in 2^* : s2^* \subseteq Z\}$.

Proposition 8. *Let* $X, Y, Z \subseteq 2^*$.

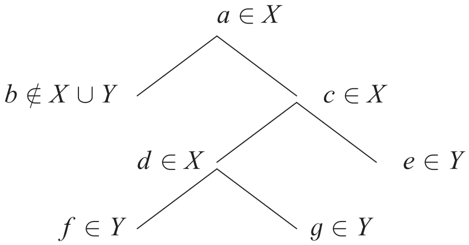
1. $core(Z)2^* = core(Z) \subseteq Z$.
2. *If* $X \subseteq 2^*$ *is finite*, $Y \subseteq 2^*$ *is prefix-free and* $X \cap Y2^* = \emptyset$ *then* $s \in core(X \cup Y2^*)$ *if and only if there exists* Z *finite maximal prefix free such that* $sZ \subseteq Y2^*$ *and* $\forall t\ \forall u\ ((s \preccurlyeq t \prec u \wedge u \in sZ) \Rightarrow t \in X)$.

Proof. Point 1 is straightforward. We prove point 2.

$\Rightarrow$. Suppose $s \in core(X \cup Y2^*)$. Then $s2^* \subseteq X \cup Y2^*$. Since X and Y are disjoint, $\forall q\ (q \in s2^* \backslash X) \Leftrightarrow (q \in Y2^* \cap s2^*)$. Since X is finite, X contains just finitely many extensions of s, and all the other extensions are in $Y2^*$. Then there exists a finite maximal prefix free set Z such that $sZ \subseteq Y2^*$ and $\forall t\ \forall u ((s \preccurlyeq t \prec u \wedge u \in sZ) \Rightarrow t \in X)$.

$\Leftarrow$. Assume Z is finite maximal prefix free set such that $sZ \subseteq Y2^*$ and $\forall t\ \forall u((s \preccurlyeq t \prec u \wedge u \in sZ) \Rightarrow t \in X)$. Then, $\forall t \in s2^* \setminus sZ2^* \quad t \in X$. Thus, $s2^* \subseteq X \cup Y2^*$, hence $s \in core(X \cup Y2^*)$. $\square$

Example. In the next picture, we have $c \in core(X \cup Y2^*)$ for the following reasons:
– $Z = \{e, f, g\}$ is a finite maximal prefix free of $c2^*$ included in Y,
– X contains all points of the "interval" $[c, Z[$ (namely c, d),
– the father a of c cannot be in $core(X \cup Y2^*)$ since $b \notin X \cup Y$.



Proposition 9. (1) *$Z \subseteq 2^*$ is checkable if and only if* $\min(\mathrm{core}(Z))$ *and $Z \setminus core(Z)$ are finite.*

(2) *Every checkable set can be written $Z = X \cup Y2^*$ where*
– X, Y are finite and $X \cap Y2^* = \emptyset$,
– Y is prefix-free and $Y2^* = core(Z)$.

Such a presentation is unique and is called the best presentation of Z.

Proof. (1) $\Leftarrow$ Use equality $Z = (Z \setminus core(Z)) \cup \min(core(Z))$ (which holds since $core(Z) = \min(core(Z))2^*$).

$\Rightarrow$ If $Z = X \cup Y2^*$ with X, Y finite and $X \cap Y = \emptyset$ then $\min(core(Z)) \subseteq X \cup Y$ is necessarily finite.

Since $Y \subseteq core(Z)$ we see that $Z \setminus core(Z) \subseteq Z \setminus Y2^* \subseteq X$ is also finite.

(2) Point 1 shows that every checkable set can be so written. We prove that such a presentation is unique. Proposition 2 insures that if $T = T2^*$ then the sole prefix-free set Y such that $Y2^* = T$ is $Y = \min(T)$. Taking $T = core(Z)$, this shows that Y is uniquely determined. Now, condition $X \cap Y2^* = \emptyset$ implies $X = Z \setminus Y2^*$. $\square$

2.3. Recursively enumerable sets of words

The following result is the basis for a normal form of Σ^0_1 subsets of the spaces 2^ω and $2^{\leqslant\omega}$ (Proposition 28).

Proposition 10. (1) *If $X \subseteq 2^*$ is recursively enumerable* (*r.e.*) *then there exists a recursive prefix-free set $Y \subset 2^*$ such that $X2^\omega = Y2^\omega$* (*in general, for such a Y one cannot take* $\min(X)$ *nor* $\min(\widehat{X})$ *which may even be non-r.e.*).

Moreover, one can suppose $Y \subseteq X$ and one can recursively go from an r.e. code for X to r.e. codes for Y and $2^* \setminus Y$.

(2) *If $X,Y \subseteq 2^*$ are r.e. then there exist an r.e. set $Z \subseteq 2^*$ and a recursive prefix-free set $T \subset 2^*$ such that $X \cup Y2^{\leqslant\omega} = Z \cup T2^{\leqslant\omega}$.*

Moreover, one can suppose $T \subseteq Z$ and one can recursively go from r.e. codes for X,Y to r.e. codes for Z,T and $2^*\backslash T$.

(3) *Points 1 and 2 hold in uniform versions. We state that for point 2.*

If $X,Y \subseteq \mathbb{N}^k \times 2^*$ are r.e. then there exist an r.e. set $Z \subseteq \mathbb{N}^k \times 2^*$ and a recursive set $T \subset \mathbb{N}^k \times 2^*$ such that *$T_{\vec{n}}$ is prefix-free and $X_{\vec{n}} \cup Y_{\vec{n}}2^{\leqslant\omega} = Z_{\vec{n}} \cup T_{\vec{n}}2^{\leqslant\omega}$ for every $\vec{n} \in \mathbb{N}^k$.*

Moreover, one can suppose $T \subseteq Z$ and one can recursively go from r.e. codes for X,Y to r.e. codes for Z,T and $(\mathbb{N}^k \times 2^*)\backslash T$.

Proof. (1) Let f be a partial recursive function with domain X. Let X_t be the set of strings with length $\leqslant t$ on which f is defined and converges in at most t computation steps. Set $n_t = t + \max_{v \in X_t} |v|$ (the sum with t is in order that n_t tends to $+\infty$) and

$$Y_t = \{u \in 2^* : \ |u| = n_t \wedge \exists v \in X_t \ v \preceq u \wedge \forall i < t \ \forall w \in X_i \ \neg(w \preccurlyeq u)\},$$

$$Y = \bigcup_{t \in \mathbb{N}} Y_t.$$

An easy induction shows that $X_t 2^\omega = (\bigcup_{i \leqslant t} Y_i)2^\omega$ for all t, whence $X2^\omega = Y2^\omega$. Also, the Y_t's are finite and prefix-free and their elements are pairwise incomparable, so that Y is also prefix-free.

Moreover, $Y \subseteq X$ and Y is recursive since a string of length k is in Y if and only if it is in Y_t for some $t \leqslant k$.

Finally, the passage from X to Y is clearly effective.

(2) To get point 2, use point 1, let $T \subseteq Y$ be recursive prefix-free such that $T2^\omega = Y2^\omega$ and set $Z = X \cup Y2^*$ (or $Z = X \cup (Y\backslash T)2^*$).

(3) Point 3 is an easy extension of points 1 and 2. □

3. The topological space $2^{\leqslant\omega}$

In this section we recall classical material from Head [18,19] and Pierce [30].

First, we extend to $2^{\leqslant\omega}$ the prefix partial order on 2^*.

Definition 11. For $\xi,\eta \in 2^{\leqslant\omega}$, we let $\xi \preccurlyeq \eta$ if and only if one of the following three situations occurs:
1. $\xi = \eta$.
2. $\xi,\eta \in 2^*$ and $\xi \preccurlyeq \eta$.
3. $\xi \in 2^*$, $\eta \in 2^\omega$ and $\eta \upharpoonright |\xi| = \xi$.

3.1. *The compact zero-dimensional topology on $2^{\leqslant\omega}$*

We consider on 2^ω the usual compact Cantor topology generated by the countable family of basic open (and closed) sets $s2^\omega$ where s varies over 2^*. This topology can also be defined by the distance

$$d(\xi,\eta) = \mathtt{IF}(\xi = \eta)\ \mathtt{THEN}\ 0\ \mathtt{ELSE}\ 2^{-|\xi \frown \eta|},$$

where $\xi \frown \eta$ denotes the longest common prefix to ξ, η.

The natural compact topology on $2^{\leqslant\omega}$ [4,18,19,40,41], see also Chap. 3 of Perrin and Pin [29] is very similar to the Cantor topology on 2^ω. The next definition and proposition stress this similarity.

Definition 12. We consider on $2^{\leqslant\omega}$ the topology generated by the basic open singleton sets $\{s\}$ and the sets $s2^{\leqslant\omega} = \{\xi \in 2^{\leqslant\omega}:\ s \preccurlyeq \xi\}$, where s varies over 2^*.

Thus, a sequence $(\xi_i)_{i\in\mathbb{N}}$, $\xi_i \in 2^{\leqslant\omega}$ converges to $\eta \in 2^{\leqslant\omega}$ if and only if

$$\forall n\ \exists m\ \forall p > m\ \ \xi_p \upharpoonright n = \eta \upharpoonright n.$$

(Recall that if $\xi \in 2^*$ then $\xi \upharpoonright n$ is the prefix of ξ with length $\min(n, |\xi|)$).

Proposition 13. (1) *With the above topology*, $2^{\leqslant\omega}$ *is a compact space. The basic open sets are also closed, so that* $2^{\leqslant\omega}$ *is zero-dimensional.*

(2) *The topology on* $2^{\leqslant\omega}$ *can also be defined by the metrics*

$$d(\xi,\eta) = \mathtt{IF}(\xi = \eta)\ \mathtt{THEN}\ 0\ \mathtt{ELSE}\ 2^{-|\xi \frown \eta|},$$

where $\xi\frown\eta$ *denotes the longest common prefix to* ξ,η (*cf. Definition* 11).

(3) *The induced topology on the subspace* 2^* *is the discrete topology and that on the subspace* 2^ω *is the compact Cantor topology.*

Proof. For sake of completeness, we recall a proof of this proposition.

(1) The fact that the basic open sets are closed is straightforward. We show that $2^{\leqslant\omega}$ is compact, i.e. that any covering by basic open sets contains a finite subcovering. Suppose

$$\left(\bigcup_{i\in I} u_i 2^{\leqslant\omega}\right) \cup \left(\bigcup_{j\in J} \{v_j\}\right) = 2^{\leqslant\omega}.$$

The trace of this covering on 2^ω induces a covering on 2^ω:

$$\bigcup_{i\in I} u_i 2^\omega = 2^\omega.$$

Since 2^ω is compact, there is a finite subset $\{i_1,\ldots,i_n\} \subseteq I$ such that $\{u_{i_1},\ldots,u_{i_n}\}2^\omega = 2^\omega$. Hence (Proposition 3) $\{u_{i_1},\ldots,u_{i_n}\}$ contains a maximal prefix-free set. In particular, every word is comparable to some u_i for the prefix ordering. Now, $\{u_{i_1},\ldots,u_{i_n}\}2^{\leqslant\omega}$ contains all infinite sequences and all finite extensions of $u_{i_1},\ldots,u_{i_n}$. So that the only remaining words are the proper prefixes of $\{u_{i_1},\ldots,u_{i_n}\}$. Such prefixes are finitely many:

– some are among the v_j's, say $v_{j_1},\ldots,v_{j_p}$,

– some belong to some $u_k 2^{\leqslant\omega}$'s, say to $u_{k_1}2^{\leqslant\omega},\ldots,u_{k_q}2^{\leqslant\omega}$.

Hence,

$$u_{i_1}2^{\leqslant\omega},\ldots,u_{i_n}2^{\leqslant\omega},v_{j_1},\ldots,v_{j_p},u_{k_1}2^{\leqslant\omega},\ldots,u_{k_q}2^{\leqslant\omega}$$

constitute a finite subcovering.

(2) Obvious.

(3) The induced topology on 2^* is clearly the discrete one. To see that the induced topology on 2^ω is the expected one, observe that

– the $\{s\}\cap 2^\omega$'s are empty,

– the $s2^{\leqslant\omega}\cap 2^\omega$'s are exactly the basic open sets of the Cantor topology. □

Remark 14. (1) 2^* is open and dense in the topological space $2^{\leqslant\omega}$, hence not closed. So that 2^ω is closed and not open in $2^{\leqslant\omega}$.

(2) The family of basic open sets coincides with that of open balls and also with that of closed balls: if $\alpha\in 2^\omega$ then

$$\{\xi\in 2^{\leqslant\omega}:\ d(\alpha,\xi)<r\}=(\alpha\restriction n)2^{\leqslant\omega}\quad\text{where } n \text{ s.t. } 2^{-n}<r\leqslant 2^{-n+1},$$
$$\{\xi\in 2^{\leqslant\omega}:\ d(\alpha,\xi)\leqslant r\}=(\alpha\restriction n)2^{\leqslant\omega}\quad\text{where } n \text{ s.t. } 2^{-n}\leqslant r<2^{-n+1}.$$

Similar characterizations hold for balls centered in some $p\in 2^*$ with some distorsion due to the fact that $d(p,\xi)$ is either 0 (in case $\xi=p$) or $\geqslant 2^{-|p|}$.

3.2. Embeddings between 2^ω and $2^{\leqslant\omega}$

As is well known, every compact zero-dimensional space is homeomorphic to a closed subset of the Cantor space. Let us explicit such an embedding for $2^{\leqslant\omega}$.

Proposition 15. *Let $\varphi: 2^*\to 2^*$ be the morphism which adds a* 1 *right to each letter of the alphabet, i.e.* $\varphi(0)=01$, $\varphi(1)=11$.

Let $\phi: 2^{\leqslant\omega}\to 2^\omega$ be defined as follows:

– $\phi(s)=\varphi(s)0^\omega$ *for all* $s\in 2^*$,

– $\phi(\alpha)=\lim_{n\to\infty}\varphi(\alpha\restriction n)$ *for all* $\alpha\in 2^\omega$.

Then ϕ is a homeomorphism defined on $2^{\leqslant\omega}$ with range the closed subset $\{01,11\}^\omega\cup\{01,11\}^{<\omega}\,0^\omega$ *of* 2^ω.

Moreover, the graph and the range of ϕ are Π^0_1 in 2^ω.

Of course, 2^ω is homeomorphically embedded in $2^{\leqslant\omega}$ by mere inclusion.

3.3. Open sets, closed sets

Definition 16. If $\mathscr{X}\subseteq 2^{\leqslant\omega}$ we let

$$\partial^{<\omega}(\mathscr{X})=\{p\in 2^*:\ p\,2^*\subseteq\mathscr{X}\},$$

$$\partial^{\omega}(\mathscr{X})=\{p\in 2^*:\ p\,2^\omega\subseteq\mathscr{X}\},$$

$$\partial^{\leqslant\omega}(\mathscr{X})=\{p\in 2^*:\ p\,2^{\leqslant\omega}\subseteq\mathscr{X}\}=\partial^{<\omega}(\mathscr{X})\cap\partial^{\omega}(\mathscr{X}).$$

Proposition 17. *Let $\mathscr{X} \subseteq 2^{\leqslant\omega}$. Then $\partial^{<\omega}(\mathscr{X})$, $\partial^{\omega}(\mathscr{X})$ and $\partial^{\leqslant\omega}(\mathscr{X})$ are closed by extension, i.e. satisfy the equation $Z = Z2^*$.*

Using Proposition 5, point 2, we see that the $^\wedge$ operator on subsets of 2^* introduced in Definition 4 is simply related to the above ∂^{ω} operator on subsets of $2^{\leqslant\omega}$.

Proposition 18. (1) *If $X \subseteq 2^*$ then $\partial^{\omega}(X2^{\omega}) = \widehat{X}2^*$.*

(2) *If $\mathscr{X} \subseteq 2^{\leqslant\omega}$ then $\partial^{\omega}(\mathscr{X}) = \widehat{\partial^{\omega}(\mathscr{X})}$.*

The well-known characterization of open sets in the Cantor space extends to the space $2^{\leqslant\omega}$ via a straightforward application of Propositions 2 and 5. We state both characterizations in parallel.

Proposition 19. (1) *Let $\mathscr{X} \subseteq 2^{\omega}$. The following conditions are equivalent:*

(i) *$\mathscr{X}$ is an open subset of the Cantor space.*
(ii) *$\mathscr{X} = X2^{\omega}$ for some $X \subseteq 2^*$.*
(iii) *$\mathscr{X} = Y2^{\omega}$ for some prefix-free set $Y \subseteq 2^*$.*
(iv) *$\mathscr{X} = \partial^{\omega}(\mathscr{X})2^{\omega} = \min(\partial^{\omega}(\mathscr{X}))2^{\omega}$.*

(2) *Let $\mathscr{X} \subseteq 2^{\leqslant\omega}$. The following conditions are equivalent:*

(i) *$\mathscr{X}$ is open in the topological space $2^{\leqslant\omega}$.*
(ii) *$\mathscr{X} = X \cup Y2^{\leqslant\omega}$ for some $X, Y \subseteq 2^*$.*
(iii) *$\mathscr{X} = Z \cup T2^{\leqslant\omega}$ for some $Z, T \subseteq 2^*$ where T is prefix-free.*
(iv) *$\mathscr{X} = (\mathscr{X} \cap 2^*) \cup \partial^{\leqslant\omega}(\mathscr{X})2^{\leqslant\omega} = (\mathscr{X} \cap 2^*) \cup \min(\partial^{\leqslant\omega}(\mathscr{X}))2^{\leqslant\omega}$.*

(3) *Moreover, one can recursively go from X to Y (resp. from X, Y to Z, T) in the equivalences of point* 1 *(resp.* 2*).*

Proof. Point 2. Observe that $X2^{\leqslant\omega} = \min(X)2^{\leqslant\omega}$ and $\min(X)$ is prefix-free. Also, $\min(X)$ can be recursively obtained from X. □

From the above characterization of open sets, going from X to $T = \{s\colon\ s \text{ has no prefix in } X\}$, we get characterizations of closed sets.

Proposition 20. (1) *For $\mathscr{X} \subseteq 2^{\omega}$, the following conditions are equivalent:*

(i) *$\mathscr{X}$ is closed in the Cantor space.*
(ii) *There exists a tree $T \subseteq 2^*$ (i.e. a set of words closed by prefix) such that $\mathscr{X}$ is the set of infinite branches of T, i.e.*

$$\mathscr{X} = \{\alpha \in 2^{\omega}\colon\ \forall n\ \alpha \upharpoonright n \in T\}.$$

One can also suppose T to be a pruned, i.e. every $s \in T$ has arbitrarily long extensions in T.

(2) *For $\mathscr{X} \subseteq 2^{\leqslant\omega}$, the following conditions are equivalent:*

(i) *$\mathscr{X}$ is closed in the topological space $2^{\leqslant\omega}$.*
(ii) *There exist $U \subseteq T \subseteq 2^*$ such that T is a tree (i.e. is closed by prefix), though not necessarily pruned, and*

$$\mathscr{X} = U \cup \{\alpha \in 2^{\omega}\colon\ \forall n \alpha \upharpoonright n \in T\}.$$

3.4. Clopen sets

The well-known characterization of clopen sets in the Cantor space also extends to the space $2^{\leqslant\omega}$. We again state both characterizations in parallel.

Proposition 21. (1) *Let $\mathscr{X}\subseteq 2^{\omega}$. The following conditions are equivalent*:

(i) *$\mathscr{X}$ is clopen (open and closed) in the Cantor space.*

(ii) *$\mathscr{X}=X2^{\omega}$ for some finite set $X\subset 2^*$.*

(iii) *$\mathscr{X}=Y2^{\omega}$ for some finite prefix-free set $Y\subseteq 2^*$.*

(iv) *$\mathscr{X}$ is open in 2^{ω} and all prefix-free sets Y such that $\mathscr{X}=Y2^{\omega}$ are finite.*

(v) *$\mathscr{X}$ is open in 2^{ω} and $\min(\partial^{\omega}(\mathscr{X}))$ is finite.*

(2) *Let $\mathscr{X}\subseteq 2^{\leqslant\omega}$. The following conditions are equivalent*:

(i) *$\mathscr{X}$ is clopen (open and closed) in $2^{\leqslant\omega}$.*

(ii) *$\mathscr{X}=X\cup Y2^{\leqslant\omega}$ for some finite sets $X,Y\subseteq 2^*$.*

(iii) *$\mathscr{X}=Z\cup T2^{\leqslant\omega}$ for some finite sets $Z,T\subseteq 2^*$ with T prefix-free.*

(iv) *$\mathscr{X}$ is open in $2^{\leqslant\omega}$ and whenever $\mathscr{X}=Z\cup T2^{\leqslant\omega}$ with $Z,T\subseteq 2^*$ and T prefix-free then T and $Z\backslash T2^*$ are finite.*

(v) *$\mathscr{X}$ is open in $2^{\leqslant\omega}$ and the following sets are finite*:

$$(\mathscr{X}\cap 2^*)\backslash(\partial^{\leqslant\omega}(\mathscr{X})2^* \textit{ and } \min(\partial^{\leqslant\omega}(\mathscr{X})).$$

Proof. Point 2 (i)⇒(iii). Suppose $Z,T,U,V\subseteq 2^*$ and $\mathscr{X}=Z\cup T2^{\leqslant\omega}$ and $\overline{\mathscr{X}}=U\cup V2^{\leqslant\omega}$. We then have $\mathscr{X}=Z\cup\min(T)2^{\leqslant\omega}$ and $\overline{\mathscr{X}}=U\cup\min(V)2^{\leqslant\omega}$ and $\min(T)$, $\min(V)$ are prefix-free. Therefore $\min(T)\cup\min(V)$ is prefix-free and $(\min(T)\cup\min(V))2^{\omega}=2^{\omega}$, whence (Proposition 3) $\min(T)\cup\min(V)$ is finite maximal prefix-free. In particular, $\min(T)$ is finite.

Also, since $\min(T)\cup\min(V)$ is finite maximal prefix-free, the set $2^*\backslash(\min(T)\cup\min(V))2^*$ is the set of strict prefixes of strings in $\min(T)\cup\min(V)$, hence it is finite. In particular, $Z\backslash(\min(T)\cup\min(V))2^*$ is finite. Since Z is disjoint from $\min(V)2^*$, we see that $Z\backslash\min(T)2^*=Z\backslash(\min(T)\cup\min(V))2^*$ is finite.

(iii)⇒(v). Equality (iii) implies $\min(\partial^{\leqslant\omega}(\mathscr{X}))\subseteq \mathit{Prefix}(T)$. Since T is finite so is $\mathit{Prefix}(T)$ and $\min(\partial^{\leqslant\omega}(\mathscr{X}))$. Also, $T\subseteq\partial^{\leqslant\omega}(\mathscr{X})$ so that

$$(\mathscr{X}\cap 2^*)\backslash\partial^{\leqslant\omega}(\mathscr{X})\subseteq(\mathscr{X}\cap 2^*)\backslash T\subseteq Z.$$

Since Z is finite so is $(\mathscr{X}\cap 2^*)\backslash\partial^{\leqslant\omega}(\mathscr{X})$.

(v)⇒(iv). Suppose $\mathscr{X}=Z\cup T2^{\leqslant\omega}$ with T prefix-free. We then have

$$\mathscr{X}=Z\cup T2^{\leqslant\omega}=(\mathscr{X}\cap 2^*)\backslash\partial^{\leqslant\omega}(\mathscr{X})\cup\min(\partial^{\leqslant\omega}(\mathscr{X}))2^{\leqslant\omega}.$$

Traces on 2^{ω} give the equality $T2^{\omega}=\min(\partial^{\leqslant\omega}(\mathscr{X}))2^{\omega}$. Since $\min(\partial^{\leqslant\omega}(\mathscr{X}))$ is finite so is T (Proposition 5, point 4).

Equality $\mathscr{X}=Z\cup T2^{\leqslant\omega}$ implies $T\subseteq\partial^{\leqslant\omega}(\mathscr{X})$. Hence

$$Z\backslash T2^*\subseteq Z\backslash\partial^{\leqslant\omega}(\mathscr{X})2^*\subseteq(\mathscr{X}\cap 2^*)\backslash\partial^{\leqslant\omega}(\mathscr{X})2^*$$

and since the rightmost set is finite, so is the leftmost.

(iv) ⇒ (ii) is trivial.

(ii) ⇒ (i). Let $n = \max_{s \in X \cup Y} |s|$ and set

$$Y' = \bigcup_{y \in Y} y\{0,1\}^{n-|y|} \quad \text{and} \quad X' = X \cup \bigcup_{y \in Y} y\{0,1\}^{<n-|y|}.$$

Clearly,
- $X \cup Y2^{\leqslant\omega} = X' \cup Y'2^{\leqslant\omega}$,
- all strings in X' have length less than n,
- all strings in Y' have length exactly n,
- X', Y' are still finite.

Now,

$$2^{\leqslant\omega} \setminus (X' \cup Y'2^{\leqslant\omega}) = (\{0,1\}^{<n} \setminus X') \cup (\{0,1\}^n \setminus Y')2^{\leqslant\omega}$$

is also open. □

For the construction of the Arithmetical Hierarchy, one needs open subsets of topological products $\mathbb{N}^k \times 2^\omega$ and $\mathbb{N}^k \times 2^{\leqslant\omega}$ with the discrete topology on $\mathbb{N}$. The next proposition gives the obvious reduction.

Proposition 22. *Let E be any topological space. Consider on $\mathbb{N}^k \times E$ the product topology* (*relative to the discrete topology on* $\mathbb{N}$). *A subset $\mathscr{X} \subseteq \mathbb{N}^k \times E$ is open* (*resp. closed, resp. clopen*) *if and only if for all $\vec{n} \in \mathbb{N}^k$ the slice $\mathscr{X}_{\vec{n}} = \{e \in E\colon (\vec{n}, e) \in \mathscr{X}\}$ is open* (*resp. closed, resp. clopen*) *in E.*

3.5. Stone and Pierce dualities

3.5.1. Stone algebras of 2^ω and $2^{\leqslant\omega}$

Stone duality associates to any zero-dimensional topological (i.e. with a basis of clopen sets) compact space $\mathscr{S}$ the boolean algebra of its clopen subsets.

In particular, conditions 1(ii) and 2(ii) in Proposition 21 show that the Stone algebras of 2^ω and $2^{\leqslant\omega}$ are countable.

Clearly, the Stone algebra of 2^ω has no atom. It is known that there is only one countable atomless boolean algebra (up to isomorphism).

On the opposite, the Stone algebra of $2^{\leqslant\omega}$ has countably many atoms: the singleton clopen sets $\{s\}$ for $s \in 2^*$. It is known that there are uncountably many non-isomorphic countable boolean algebras with countably many atoms. In fact, the Cantor–Bendixson derivative process over $2^{\leqslant\omega}$ stops after exactly 1 step, but it is easy to design compact subsets of $2^{\leqslant\omega}$ for which this process is indexed by arbitrary countable ordinals.

Remark 23. These Stone algebras have simple presentations as inductive limits of directed sequences of finite boolean algebras.

1. Denote P_n the boolean algebra of all sets of words of length n and $\mu_n : P_n \to P_{n+1}$ the homomorphism $X \mapsto X0 \cup X1$. From Proposition 21, it is easy to see that the Stone algebra of 2^ω is isomorphic to the inductive limit of the directed sequence $(P_n, \mu_n)_{n \in \mathbb{N}}$.

2. Denote Q_n the boolean algebra of all pairs of disjoint sets of words of length n and $v_n : Q_n \to Q_{n+1}$ the homomorphism $(X, Y) \mapsto (X \cup Y, Y0 \cup Y1)$. From Proposition 21, it is easy to see that the Stone algebra of $2^{\leqslant\omega}$ is isomorphic to the inductive limit of the directed sequence $(Q_n, v_n)_{n\in\mathbb{N}}$.

3.5.2. Pierce duality

A variant of Stone duality, which is much adapted to the present context, has been introduced by Pierce [30].

On any topological space $\mathscr{S}$, one can consider the operations of adherence (topological closure) $\mathscr{X} \mapsto \overline{\mathscr{X}}$ and derivation (which deletes the isolated points) $\mathscr{X} \mapsto \partial(\mathscr{X})$. Any boolean algebra of subsets of $\mathscr{S}$ which is closed by these two operations is called a topological boolean algebra.

The Pierce algebra $Pierce(\mathscr{S})$ of a zero-dimensional compact space $\mathscr{S}$ is defined as the smallest topological Boolean algebra of subsets of $\mathscr{S}$, enriched with the function $X \mapsto card(X)$ where $card(X)$ is the cardinality of X (which necessarily lies in $\mathbb{N} \cup \{\aleph_0, 2^{\aleph_0}\}$).

Pierce [30, Corollary 4.4, p. 12] proves that if $Pierce(\mathscr{S}_1), Pierce(\mathscr{S}_2)$ are finite then they are isomorphic if and only if $\mathscr{S}_1, \mathscr{S}_2$ are homeomorphic.

Clearly, in the present context,

- The Pierce algebra of the Cantor space has 2 elements: $\emptyset$, 2^ω.
- The Pierce algebra of the space $2^{\leqslant\omega}$ has 4 elements: $\emptyset$, $2^{\leqslant\omega}$, 2^ω, 2^* (the last two elements being obtained as $\partial(2^{\leqslant\omega})$ and its complement set).

As a simple application of Pierce duality, let us mention the fact that $2^{\leqslant\omega}$ is homeomorphic to $2^{\leqslant\omega} \times 2^{\leqslant\omega}$ (a result not as easy to obtain directly as its analog with the Cantor space).

Remark 24. As is well-known, Stone duality gives a correspondence between continuous maps $\mathscr{S}_1 \to \mathscr{S}_2$ and homomorphisms $Stone(\mathscr{S}_2) \to Stone(\mathscr{S}_1)$.

Such a correspondence is no more possible with Pierce duality since it involves finite algebras (so that there are only finitely many homomorphisms). Pierce duality deals with the *existence* of homeomorphisms $\mathscr{S}_1 \to \mathscr{S}_2$ and isomorphisms $Pierce(\mathscr{S}_2) \to Pierce(\mathscr{S}_1)$.

3.6. Other topologies on $2^{\leqslant\omega}$

3.6.1. $\mathscr{T}_0$ topologies on 2^* and $2^{\leqslant\omega}$

We shall also refer in Section 6.2 and 6.3 to another compact (but not Hausdorff) topology on $2^{\leqslant\omega}$ which has as basic open sets the sole sets $s2^{\leqslant\omega}$ for $s \in 2^*$ (cf. Weihrauch [47], 1987, pp. 228–229). We shall call this topology the weak topology on $2^{\leqslant\omega}$.

We shall also consider the analogous weak topology on 2^* which has as basic open sets the sets $s2^*$ for $s \in 2^*$. The trace on 2^ω of the weak topology of $2^{\leqslant\omega}$ is the Cantor topology. Also, for every $\alpha \in 2^\omega$, exactly the same sequences of elements of $2^{\leqslant\omega}$ converge towards α for the compact topology and for the weak topology.

As concerns words, the trace on 2^* of the weak topology of $2^{\leqslant\omega}$ is the weak topology on 2^*. Let us observe that if $s \in 2^*$ then s has a smallest weak neighbourhood in $2^{\leqslant\omega}$ (resp. 2^*) which is not $\{s\}$ but $s2^{\leqslant\omega}$ (resp. $s2^*$). Hence, $s2^{\leqslant\omega}$ (resp. $s2^*$) is the weak adherence of $\{s\}$ in $2^{\leqslant\omega}$ (resp. 2^*).

Thus, the weak topologies on $2^{\leqslant\omega}$ and 2^* are non-Hausdorff. Nevertheless, they are $\mathscr{T}_0$ (in the sense of Kolmogorov, cf. [21] p. 51 or [5] p. 135, Exercise 2 for Section 1): for every pair of different points there exists an open set which contains one of the points and does not contain the other one.

3.6.2. Redziejowski topology on $2^{\leqslant\omega}$

Redziejowski [31] introduced another topology on $2^{\leqslant\omega}$ such that a sequence of words $(s_i)_{i\in\mathbb{N}}$ converges to $\alpha \in 2^\omega$ if and only if for i large enough the s_i's are prefixes of α and the length of s_i tends to $+\infty$ with i. Thus, a sequence like $(0^i1)_{i\in\mathbb{N}}$ does not converge towards 0^ω (contrary to the case with the compact topology). The basic open sets of Redziejowski's topology are

$$R_{\alpha,n} = \{\alpha\} \cup \{\alpha \upharpoonright p\colon\ p \geqslant n\},$$

where α varies over 2^ω and n over $\mathbb{N}$. This topology is clearly stronger than the one of Definition 12 since $s2^{\leqslant\omega} = \bigcup_{\alpha \geqslant_{q} s} R_{\alpha,|s|}$. Thus, 2^* is strongly open and 2^ω is strongly closed.

Though Hausdorff, Redziejowski's topology is not metrizable. It induces the discrete topology on both subspaces 2^* and 2^ω since

- $\{s\} = R_{s0^\omega,|s|} \cap R_{s1^\omega,|s|}$ is open,
- $\{\alpha\} = R_{\alpha,n} \cap 2^\omega$ is relatively open in the subspace 2^ω.

As a consequence, the associated Borel hierarchy collapses since every set $\mathscr{X} \subseteq 2^{\leqslant\omega}$ can be written $\mathscr{X} = (\mathscr{X} \cap 2^*) \cup (\mathscr{X} \cap 2^\omega)$ which is the union of an open set and a set relatively open in a closed set (i.e. the intersection of an open set with a closed set), hence is F_σ and G_δ.

4. Computability over subsets of 2^ω and $2^{\leqslant\omega}$

4.1. Computable subsets of 2^ω and $2^{\leqslant\omega}$

This subsection can be considered as relevant from the general theory of representations of "computable" metric spaces, cf. Kreitz and Weihrauch [50] and Weihrauch [48]. However, we prefer to give a self-contained equivalent direct approach for $2^{\leqslant\omega}$ by elaborating from the classical computability theory for 2^ω.

The notion of computable (or recursive) subset of $2^{\leqslant\omega}$ is defined in the same way as that of computable subset of 2^ω (cf. any text book, e.g. [32]) and has an analogous characterization as clopen subsets.

Nevertheless, it is important to notice that computability over 2^ω *is not* induced by that over $2^{\leqslant\omega}$ (nor by that over the Baire space ω^ω). In fact, 2^ω is not a computable subset of $2^{\leqslant\omega}$.

Again, we state both definitions and characterizations in parallel and (in view of the construction of the Arithmetical Hierarchy) let them involve some extra integer arguments.

Definition 25. (1) A set $\mathcal{X} \subseteq \mathbb{N}^k \times 2^\omega$ is computable if there exists some Turing machine which, given any input $(\vec{n}, \alpha) \in \mathbb{N}^k \times 2^\omega$, halts in finite time and accepts (resp. rejects) if $(\vec{n}, \alpha) \in \mathcal{X}$ (resp. $(\vec{n}, \alpha) \notin \mathcal{X}$).

(2) Idem for $\mathcal{X} \subseteq \mathbb{N}^k \times 2^{\leqslant\omega}$.

Proposition 26. (1) *Let* $\mathcal{X} \subseteq 2^{\leqslant\omega}$. *The following conditions are equivalent*:

(i) $\mathcal{X}$ *is computable as a subset of* $2^{\leqslant\omega}$.

(ii) $\mathcal{X}$ *is clopen in the compact space* $2^{\leqslant\omega}$.

(iii) $\mathcal{X} = X \cup Y2^{\leqslant\omega}$ *for some finite* $X, Y \subset 2^*$.

(2) *Let* $\mathcal{X} \subseteq \mathbb{N}^k \times 2^{\leqslant\omega}$. *The following conditions are equivalent*:

(i) $\mathcal{X}$ *is computable as a subset of* $\mathbb{N}^k \times 2^{\leqslant\omega}$.

(ii) *There exist recursive sets* $X, Y \subset \mathbb{N}^k \times 2^*$ *such that for all* $\vec{n} \in \mathbb{N}^k$:

- *the slices* $X_{\vec{n}}, Y_{\vec{n}}$ *are finite*,
- $\mathcal{X}_{\vec{n}} = X_{\vec{n}} \cup Y_{\vec{n}} 2^{\leqslant\omega}$

(*in other words*, $\mathcal{X}$ *is clopen and there is a recursive representation of the slices* $\mathcal{X}_{\vec{n}}$ *as unions* $X_{\vec{n}} \cup Y_{\vec{n}} 2^{\leqslant\omega}$).

(3) With obvious changes (forget $X, X_{\vec{n}}$, only keep $Y, Y_{\vec{n}}$ and replace $2^{\leqslant\omega}$ by 2^ω), the same equivalences hold for the Cantor case, i.e. for $\mathcal{X} \subseteq 2^\omega$ or $\mathcal{X} \subseteq \mathbb{N}^k \times 2^\omega$.

Proof. (1) We prove only (i) ⇒ (ii). The proof is an easy adaptation of the classical one for the Cantor case. Consider a Turing machine which computes the computable subset $\mathcal{X}$ of $2^{\leqslant\omega}$. It is clear that the machine reads only a finite prefix of its input before it halts. A simple application of König's lemma gives a uniform bound: there exists $M \in \mathbb{N}$ such that, for every input $\xi \in 2^{\leqslant\omega}$, the machine reads at most M letters of ξ before halting.

In fact, if there were no such bound M then there would exist a sequence $(\xi_m)_{m\in\mathbb{N}}$ of elements of $2^{\leqslant\omega}$ such that $|\xi_m| \geqslant m$ and the machine does read the first m letters of ξ_m before halting. In particular, in case ξ_m is a finite word, the machine halts before reading the end-marker attached to ξ_m as a finite input. König's lemma insures that there is a strictly monotonous prefix increasing subsequence $(\xi_{k_m})_{m\in\mathbb{N}}$. Let $\xi \in 2^\omega$ be the limit of this subsequence. It is clear that on input ξ, the machine will read ξ entirely, hence will not halt. A contradiction.

Now, given this uniform bound M, we see that

$$\mathcal{X} = (\mathcal{X} \cap \{0,1\}^{<M}) \cup (\mathcal{X} \cap \{0,1\}^M) 2^{\leqslant\omega}$$

which shows that $\mathcal{X}$ is clopen.

(2) We prove only (i) ⇒ (ii). Fix $\vec{n} \in \mathbb{N}^k$. Dovetailing over all computations on input $(\vec{n}, \xi)$ where ξ varies in $2^{\leqslant\omega}$, one can get a uniform bound $M_{\vec{n}}$ for the number of letters of ξ read before the machine halts.

The function $\vec{n} \mapsto M_{\vec{n}}$ is clearly recursive and leads to the recursive family $(X_{\vec{n}}, Y_{\vec{n}})_{\vec{n}\in\mathbb{N}^k}$. □

Remark 27. In particular, 2^* and 2^ω are not computable subsets of $2^{\leqslant\omega}$ (cf. Corollary 30). This corresponds to the fact that a machine cannot decide in finite time whether its input is infinite.

4.2. The Arithmetical Hierarchy over $2^{\leqslant\omega}$

Starting from computable subsets of $\mathbb{N}^k \times 2^{\leqslant\omega}$, the Arithmetical Hierarchy over $2^{\leqslant\omega}$ is obtained in the usual way as an effectivization of the finite levels of the Borel hierarchy.

An easy application of point 2 of Proposition 10 allows to effectivize the normal form of open sets stated in conditions (i)–(iii) of Proposition 19. This normal form then propagates through successive finite levels of the Borel hierarchy.

Proposition 28. *Let "T is rpf" mean that*
– T is recursive prefix-free when $T \subseteq 2^*$,
– T is recursive and all slices $T_{\vec{n}}$'s (for $\vec{n} \in \mathbb{N}^k$) are prefix-free in case $T \subseteq \mathbb{N}^k \times 2^*$.

$\widetilde{\mathcal{X}}$ denotes the complement of $\mathcal{X}$ in $2^{\leqslant\omega}$.

The correspondence between effective Borel sets and the Arithmetical Hierarchy over $2^{\leqslant\omega}$ is as follows:

$$\begin{aligned}
(Open)\colon\ \mathcal{X} \text{ is } \Sigma^0_1 &\equiv \mathcal{X} = X \cup Y2^{\leqslant\omega} \text{ with } X, Y \text{ r.e.}\\
&\equiv \mathcal{X} = Z \cup T2^{\leqslant\omega} \text{ with } Z \text{ r.e. and } T \text{ rpf},\\
&\quad \text{where } X, Y, Z, T \subseteq 2^*.
\end{aligned}$$

$$\begin{aligned}
(Closed)\colon\ \mathcal{X} \text{ is } \Pi^0_1 &\equiv \mathcal{X} = \widetilde{X \cup Y2^{\leqslant\omega}} \text{ with } X, Y \text{ r.e.}\\
&\equiv \mathcal{X} = \widetilde{Z \cup T2^{\leqslant\omega}} \text{ with } Z \text{ r.e., } T \text{ rpf},\\
&\quad \text{where } X, Y, Z, T \subseteq 2^*.
\end{aligned}$$

$$\begin{aligned}
(F_\sigma)\colon\ \mathcal{X} \text{ is } \Sigma^0_2 &\equiv \mathcal{X} = \bigcup_{i\in\mathbb{N}} \widetilde{X_i \cup Y_i2^{\leqslant\omega}} \text{ with } X, Y \text{ r.e.}\\
&\equiv \mathcal{X} = \bigcup_{i\in\mathbb{N}} \widetilde{Z_i \cup T_i2^{\leqslant\omega}} \text{ with } Z \text{ r.e., } T \text{ rpf},\\
&\quad \text{where } X, Y, Z, T \subseteq \mathbb{N} \times 2^*.
\end{aligned}$$

$$\begin{aligned}
(G_\delta)\colon\ \mathcal{X} \text{ is } \Pi^0_2 &\equiv \mathcal{X} = \bigcap_{i\in\mathbb{N}} (X_i \cup Y_i2^{\leqslant\omega}) \text{ with } X, Y \text{ r.e.}\\
&\equiv \mathcal{X} = \bigcap_{i\in\mathbb{N}} (Z_i \cup T_i2^{\leqslant\omega}) \text{ with } Z \text{ r.e., } T \text{ rpf},\\
&\quad \text{where } X, Y, Z, T \subseteq \mathbb{N} \times 2^*.
\end{aligned}$$

$$(G_{\delta\sigma}):\ \mathscr{X} \textit{ is } \Sigma^0_3 \equiv \mathscr{X} = \bigcup_{i\in\mathbb{N}} \bigcap_{j\in\mathbb{N}} X_{i,j} \cup Y_{i,j}2^{\leqslant\omega} \textit{ with } X, Y \textit{ r.e.}$$

$$\equiv \mathscr{X} = \bigcup_{i\in\mathbb{N}} \bigcap_{j\in\mathbb{N}} Z_{i,j} \cup T_{i,j}2^{\leqslant\omega} \textit{ with } Z \textit{ r.e.}, \ T \textit{ rpf},$$

$$\textit{where } X, Y, Z, T \subseteq \mathbb{N}^2 \times 2^*.$$

$$(F_{\sigma\delta}):\ \mathscr{X} \textit{ is } \Pi^0_3 \equiv \mathscr{X} = \bigcap_{i\in\mathbb{N}} \widetilde{\bigcup_{j\in\mathbb{N}} (X_{i,j} \cup Y_{i,j}2^{\leqslant\omega})} \textit{ with } X, Y \textit{ r.e.}$$

$$\equiv \mathscr{X} = \bigcap_{i\in\mathbb{N}} \widetilde{\bigcup_{j\in\mathbb{N}} (Z_{i,j} \cup T_{i,j}2^{\leqslant\omega})} \textit{ with } Z \textit{ r.e.}, \ T \textit{ rpf},$$

$$\textit{where } X, Y, Z, T \subseteq \mathbb{N}^2 \times 2^*.$$

...

Moreover, one can recursively go from X, Y to Z, T in the above equivalences.

The usual characterization of recursive sets as Δ^0_1 sets also holds.

Proposition 29. *Computable subsets of $2^{\leqslant\omega}$ are exactly the Δ^0_1 subsets (i.e. sets which are both Σ^0_1 and Π^0_1 in $2^{\leqslant\omega}$).*

Moreover, there is a recursive process to go from (codes of) r.e. sets of words X, Y, X', Y' to finite sets of words Z, T such that if $X \cup Y2^{\leqslant\omega}$ and $X' \cup Y'2^{\leqslant\omega}$ are complementary subsets of $2^{\leqslant\omega}$ then $X \cup Y2^{\leqslant\omega} = Z \cup T2^{\leqslant\omega}$.

Proof. Let $\mathscr{X} = X \cup Y2^{\leqslant\omega}$ and $\overline{\mathscr{X}} = X' \cup Y'2^{\leqslant\omega}$ where X, X', Y, Y' are r.e. Using the last assertion of Proposition 28, one can recursively in X, Y, X', Y' get r.e. sets U, U', and recursive prefix free sets V, V' such that $\mathscr{X} = U \cup V2^{\leqslant\omega}$ and $\widetilde{\mathscr{X}} = U' \cup V'2^{\leqslant\omega}$. Then $V \cup V'$ is also prefix-free and (considering traces on 2^ω), we have $(V \cup V')2^\omega = 2^\omega$. Therefore (Proposition 3) $V \cup V'$ is finite maximal prefix-free. V, V' can be obtained from Y, Y' by the process described in the proof of Proposition 10, as the (finite) limit of V_t, V'_t (where $t \in \mathbb{N}$). The property of maximal prefix-freeness allows to stop this process as soon as $V_t \cup V'_t$ becomes maximal prefix-free since necessarily we then have $V = V_t$ and $V' = V'_t$.

Let n be the maximum length of words in the finite prefix-free set $V \cup V'$. Let T be $\{0,1\}^n \cap V2^*$ and let Z be $(\{0,1\}^{<n} \cap V2^*) \cup (U \cap \{0,1\}^{<n})$. Define T', Z' similarly. Since $T, T' \subseteq \{0,1\}^n$ and $T \cup T'$ is maximal prefix-free, then T, T' constitute a partition of $\{0,1\}^n$. $Z, Z' \subseteq \{0,1\}^{<n}$, so they are finite, and are recursively obtainable from U, U', T, T'. Therefore, we have

$\mathscr{X} = Z \cup T2^{\leqslant\omega}$ and $\widetilde{\mathscr{X}} = Z' \cup T'2^{\leqslant\omega}$, where Z, Z', T, T' are finite. □

Corollary 30. *2^* and 2^ω are respectively Σ^0_1 and Π^0_1 in $2^{\leqslant\omega}$ (but none is computable in $2^{\leqslant\omega}$).*

4.3. Logical characterization of the Arithmetical Hierarchy

We now explicit a logical characterization of this Arithmetical Hierarchy. Recall *string*: $\mathbb{N} \to 2^*$ denotes a recursive bijection such that all usual associated functions (length, restriction, etc.) are recursive (cf. Section 2.1).

Proposition 31. *If $n \geqslant 1$ then $\mathscr{X} \subseteq (2^{\leqslant\omega})^l \times \mathbb{N}^k$ is Σ^0_n (resp. Π^0_n) if and only if it can be expressed via some formula $\Phi(\vec{\xi}, \vec{i})$ which is obtained via some Σ^0_n (resp. Π^0_n) prefix of quantifications over $\mathbb{N}$ from a boolean combination of atomic arithmetical formulas (involving integers) and atomic formulas of the form*

$$string(i) = \xi, \quad string(i) \preccurlyeq \xi$$

involving variables i varying over $\mathbb{N}$ and ξ varying over $2^{\leqslant\omega}$.

Remark 32. In Proposition 31 one can also take $\xi \upharpoonright n = string(i)$ as the sole atomic relation (besides the purely arithmetical ones). In fact,

$$string(i) \preccurlyeq \xi \Leftrightarrow \xi \upharpoonright |string(i)| = string(i),$$
$$n > |string(i)| \Rightarrow (\xi = string(i) \Leftrightarrow \xi \upharpoonright n = string(i)).$$

4.4. Traces of the Arithmetical Hierarchy over $2^{\leqslant\omega}$

The Arithmetical Hierarchy over $2^{\leqslant\omega}$ is related to those over 2^* and 2^ω.

Proposition 33. *Denote $\Sigma^0_n(\mathscr{S})$ and $\Pi^0_n(\mathscr{S})$ the Σ^0_n and Π^0_n classes relative to the space $\mathscr{S}$ (which is to be 2^*, 2^ω or $2^{\leqslant\omega}$).*
1. *If $n \geqslant 2$ then*

$$\mathscr{X} \subseteq 2^{\leqslant\omega} \text{ is } \Sigma^0_n(2^{\leqslant\omega}) \Leftrightarrow \mathscr{X} \cap 2^* \text{ is } \Sigma^0_n(2^*) \wedge \mathscr{X} \cap 2^\omega \text{ is } \Sigma^0_n(2^\omega),$$
$$\mathscr{X} \subseteq 2^{\leqslant\omega} \text{ is } \Pi^0_n(2^{\leqslant\omega}) \Leftrightarrow \mathscr{X} \cap 2^* \text{ is } \Pi^0_n(2^*) \wedge \mathscr{X} \cap 2^\omega \text{ is } \Pi^0_n(2^\omega).$$

2.

$$\mathscr{X} \subseteq 2^{\leqslant\omega} \text{ is } \Sigma^0_1(2^{\leqslant\omega}) \Rightarrow \mathscr{X} \cap 2^* \text{ is } \Sigma^0_1(2^*) \wedge \mathscr{X} \cap 2^\omega \text{ is } \Sigma^0_1(2^\omega),$$
$$\mathscr{X} \subseteq 2^{\leqslant\omega} \text{ is } \Pi^0_1(2^{\leqslant\omega}) \Rightarrow \mathscr{X} \cap 2^* \text{ is } \Pi^0_1(2^*) \wedge \mathscr{X} \cap 2^\omega \text{ is } \Pi^0_1(2^\omega).$$

3.

$$\mathscr{X} \subseteq 2^* \Rightarrow (\mathscr{X} \text{ is } \Sigma^0_1(2^*) \Leftrightarrow \mathscr{X} \text{ is } \Sigma^0_1(2^{\leqslant\omega})),$$
$$\mathscr{X} \subseteq 2^\omega \Rightarrow (\mathscr{X} \text{ is } \Pi^0_1(2^\omega) \Leftrightarrow \mathscr{X} \text{ is } \Pi^0_1(2^{\leqslant\omega})).$$

4. *If $\mathscr{X}$ is $\Sigma^0_1(2^{\leqslant\omega})$ then $\partial^\omega(\mathscr{X})$ and $\partial^{\leqslant\omega}(\mathscr{X})$ are $\Sigma^0_1(2^*)$.*
Conversely, if $\mathscr{X}$ is open in $2^{\leqslant\omega}$ and $\mathscr{X} \cap 2^*$ and $\partial^{\leqslant\omega}(\mathscr{X})$ are $\Sigma^0_1(2^*)$ then $\mathscr{X}$ is $\Sigma^0_1(2^{\leqslant\omega})$.

Proof. (1) $\Rightarrow$. Observe that 2^* and 2^ω are respectively $\Sigma^0_1(2^{\leqslant\omega})$ and $\Pi^0_1(2^{\leqslant\omega})$ hence $\Delta^0_n(2^{\leqslant\omega})$.

(2) Suppose $\mathscr{X}$ is Σ^0_1 in $2^{\leqslant\omega}$ and let $\mathscr{X}=X\cup Y2^{\leqslant\omega}$ where X,Y are r.e. subsets of 2^*. Then $\mathscr{X}\cap 2^*=X\cup Y2^*$ and $\mathscr{X}\cap 2^\omega=Y2^\omega$. Since X,Y are r.e., these sets are which are respectively $\Sigma^0_1(2^*)$ and $\Sigma^0_1(2^\omega)$.

Going to the complement, we get the case $\mathscr{X}$ is $\Pi^0_1(2^{\leqslant\omega})$.

(3) *Case* $\mathscr{X}\subseteq 2^*$. Then both conditions $\mathscr{X}$ is $\Sigma^0_1(2^*)$ and $\mathscr{X}$ is $\Sigma^0_1(2^{\leqslant\omega})$ express that $\mathscr{X}$ is an r.e. set of words, hence they are equivalent.

Case $\mathscr{X}\subseteq 2^\omega$. If $\mathscr{X}$ is $\Pi^0_1(2^\omega)$ then $2^\omega\backslash\mathscr{X}=Y2^\omega$ where $Y\subseteq 2^*$ is r.e. Thus, $2^{\leqslant\omega}\backslash\mathscr{X}=2^*\cup Y2^\omega$ is therefore $\Sigma^0_1(2^{\leqslant\omega})$ so that $\mathscr{X}$ is $\Pi^0_1(2^{\leqslant\omega})$.

If $\mathscr{X}$ is $\Pi^0_1(2^{\leqslant\omega})$ then $2^{\leqslant\omega}\backslash\mathscr{X}=X\cup Y2^{\leqslant\omega}$ where $X,Y\subseteq 2^*$ are r.e. Thus, $2^\omega\backslash\mathscr{X}=Y2^\omega$ is $\Sigma^0_1(2^\omega)$ so that $\mathscr{X}$ is $\Pi^0_1(2^\omega)$.

(4) Using Proposition 3, we see that $\partial^\omega(\mathscr{X}),\partial^{\leqslant\omega}(\mathscr{X})$ are Σ^0_1 as follows:

$$\begin{aligned} p\in\partial^\omega(\mathscr{X}) &\Leftrightarrow p2^\omega\subseteq Y2^\omega\\ &\Leftrightarrow p2^\omega=(p2^*\cap Y)2^\omega\\ &\Leftrightarrow 2^\omega=\{q: pq\in Y\}2^\omega\\ &\Leftrightarrow \exists Z\ (Z \text{ is finite maximal prefix-free} \wedge \forall z\in Z\ \ pz\in Y), \end{aligned}$$

$$\begin{aligned} p\in\partial^{\leqslant\omega}(\mathscr{X}) &\Leftrightarrow (p2^\omega\subseteq Y2^\omega)\wedge(p2^*\subseteq(X\cup Y2^*))\\ &\Leftrightarrow \exists Z\ (Z \text{ is finite maximal prefix-free}\\ &\qquad \wedge\ \forall z\in Z\ \ pz\in Y\wedge\forall z\in Z\ \forall u\prec pz\ \ (p\preccurlyeq u\Rightarrow u\in X)). \end{aligned}$$

Conversely, since $\mathscr{X}$ is open, we have $\mathscr{X}=(\mathscr{X}\cap 2^*)\cup\partial^{\leqslant\omega}(\mathscr{X})(2^{\leqslant\omega})$. □

Remark 34. Corollary 30 shows that points 2 and 3 of the above result cannot be improved. Though its traces on the spaces 2^* and 2^ω are computable in the sense of these respective spaces, the set 2^* (resp. 2^ω) is not closed (resp. not open) nor computable nor Π^0_1 (resp. not Σ^0_1) as a subset of the space $2^{\leqslant\omega}$. It is solely open and Σ^0_1 (resp. closed and Π^0_1).

Remark 35. In general, the syntactical complexity of $\partial^{<\omega}(X)$ involves an extra $\forall$ quantifier:

(i) Let X be the set of strings 0^i1u such that M_i does not converge on any input of length $\leqslant|u|$ in $\leqslant|u|$ steps. Then X is recursive but $\partial^{<\omega}(X)$ is strict Π^0_1 since

$$Dom(M_i)=\emptyset \quad\Leftrightarrow\quad 0^i1\in\partial^{<\omega}(X).$$

(ii) Let $X=\{0^i1u\text{: } Dom(M_i) \text{ has at least } |u| \text{ elements}\}$. Then X is r.e. but $\partial^{<\omega}(X)$ is strict Π^0_2 since

$$Dom(M_i)\text{ is infinite} \quad\Leftrightarrow\quad 0^i1\in\partial^{<\omega}(X).$$

This last example can easily be extended to get $X\subseteq 2^*$ which is Σ^0_n and such that $\partial^{<\omega}(X)$ is Π^0_{n+1} and not Σ^0_n.

4.5. Checkable sets as clopen traces

The characterization of clopen subsets of $2^{\leqslant\omega}$ (cf. 21) motivates the following proposition. As can be expected, checkable (resp. recursively checkable) sets are exactly the traces of clopen (resp. computable) sets.

Proposition 36 (Checkable sets as clopen traces). (1) $Z \subseteq 2^*$ *is checkable if and only if* $Z = \mathscr{Z} \cap 2^*$ *for some clopen* (*hence computable*) $\mathscr{Z} \subseteq 2^{\leqslant\omega}$.

(2) *Let* $\vec{\mathbb{X}}$ *be any finite product of spaces* $\mathbb{N}$ *and/or* 2^*. *A set* $Z \subset \vec{\mathbb{X}} \times 2^*$ *is checkable* (*resp. recursively checkable*) *relative to its last component if and only if* $Z = \mathscr{Z} \cap (\vec{\mathbb{X}} \times 2^*)$ *for some clopen* (*resp. computable*) $\mathscr{Z} \subseteq \vec{\mathbb{X}} \times 2^{\leqslant\omega}$.

Proof. Let X, Y be such that $Z_{\vec{x}} = X_{\vec{x}} \cup Y_{\vec{x}} 2^*$ for all $\vec{x} \in \vec{\mathbb{X}}$ and define $\mathscr{Z}$ so that $\mathscr{Z}_{\vec{x}} = X_{\vec{x}} \cup Y_{\vec{x}} 2^{\leqslant\omega}$. □

5. (Semi)computability with possibly infinite computations

We concentrate now on the maps associated to the input/output behaviour of Turing machines performing possibly infinite computations. We limit ourselves to notions which will prove to be effective versions of continuity and lower semicontinuity, cf. Section 6.

More related material can be found in Wagner [44], Wagner and Staiger [45], Staiger [39,42] and Engelfriet and Hoogeboom [11].

5.1. Possibly infinite computations and architectural decisions

A possibly infinite computation on a Turing machine is either a halting or a non-halting computation. The output may be finite or infinite, and the input actually read by the machine may also be finite or infinite. In full generality, this leads to consider 2^* or 2^ω or $2^{\leqslant\omega}$ as the set of inputs, and $2^{\leqslant\omega}$ as the set of outputs. Hence to represent the machine behaviour as maps

$$2^* \to 2^{\leqslant\omega}, \quad 2^\omega \to 2^{\leqslant\omega} \quad \text{or} \quad 2^{\leqslant\omega} \to 2^{\leqslant\omega}.$$

As is well known, in the case of halting computations different architectures of Turing machines are irrelevant in terms of computability. Turing machines, under any architecture whatsoever, compute exactly all partial recursive functions. However, *architectures do matter for non-halting computations*.

5.1.1. Monotone Turing machines

Architectural decisions on the moving abilities of the output head and the possibility of overwriting the output do affect the class of functions that become computable via possibly infinite computations.

In all this paper, *we shall consider solely monotone Turing machines*. This was indeed Turing's original assumption [43], insuring that in the limit of time the output of a non-halting computation always converges, either to a finite or an infinite sequence. A concept also reconsidered by Levin [24] and Schnorr [33,34], see [25, p. 276].

Definition 37. A Turing machine is *monotone* if its output tape is one-way and write-only (hence no erasing nor overwriting is possible).

Thus, the sequence of symbols written on the output tape increases monotonically with respect to the prefix ordering as the number of computation steps grows.

Remark 38. An infinite sequence $\beta \in 2^{\omega}$ is the output of a machine of this type with input $\alpha \in 2^{\omega}$ if and only if β is a sequence recursive in α (i.e. β is the characteristic function of a set that is recursive in α).

Thus, we shall consider Turing machines with the following architecture:

- A pre-given finite transition table determines the computation. The computation may either lead to a halting state or may go on forever.
- The input and output tapes are infinite to the right and their heads *move only rightwards*.
- The input (resp. output) tape can only be read (resp. written) by the machine.
- The work tape is infinite in both directions and its head moves in both directions.
- Work tapes can be read, written and erased.
- A computation starts with the heads of the input and output tapes in their respective leftmost cells and the work tape being all blank.
- In order to properly deal with the case of an empty input, we suppose that the input tape contains a first dummy cell which receives no symbol and which is scanned by the head when the computation starts.

Remark 39. Let us cite two alternative choices as concerns outputs.

(1) *Increasing overwriting*: The output head moves in both directions, but overwriting must be increasing in the lexicographic order. For the output alphabet $\{0,1\}$ this means that it is only possible that to overwrite 0s with 1s (that was a compulsory condition in the time of punched cards). This condition also insures that in the limit of time the output of a non-halting computation converges, either to finite or an infinite sequence. An infinite sequence $\beta \in 2^{\omega}$ is the output of a machine of this type with input α if and only if β is a strongly computably enumerable sequence in α (i.e. β is the characteristic function of some set recursively enumerable in α).

(2) *Arbitrary moves and overwriting* (cf. Wagner [44] and Freund and Staiger [14]): The output head moves arbitrarily and it has complete freedom to overwrite or erase symbols. In the limit of time the output of a non-halting computation may not converge. If it converges, it results in either a finite or an infinite sequence. Shoenfield's limit lemma (cf. [35,36] or [28, p. 373]) insures that an infinite sequence $\beta \in 2^{\omega}$ is the output of a machine of this type with input α if and only if β is $\Delta^0_2(\alpha)$ (i.e. β is the characteristic function of a set that is $\Delta^0_2(\alpha)$).

5.1.2. Oracles

Turing machines can be equipped with an oracle A, adding to the previous architecture an oracle tape which is infinite to the right and can only be read by the machine. The ith square of this oracle tape contains 1 if $string(i) \in A$, and 0 otherwise. All the material in the following sections go through mutatis mutandis when oracles are considered.

5.1.3. Input delimitation

Another architectural decision is how to delimit the input. In this section, *we will assume the usual assumption on Turing machines which is to use blank symbols to delimit a finite input*.

5.1.4. When does a computation converge?

As we consider possibly infinite computations with monotone Turing machines, there is always a limit output so that there is no reason to discard any computation. Thus, such machines compute total maps $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ or (if we restrict the inputs to finite words or to infinite sequences) total maps $2^* \to 2^{\leqslant\omega}$ or total maps $2^\omega \to 2^{\leqslant\omega}$.

Remark 40. One also naturally gets total maps in case the output head is allowed to move in both directions but overwriting is constrained to be increasing relative to some ordering on the alphabet (architectural choice 1 in Remark 39).

However, if the output head is allowed to move and overwrite with no constraint (architectural choice 2 in Remark 39) then computations may suffer of an infinite fluctuation of their output. So that such a machine necessarily defines a partial function.

As said in Section 5.1.1, we shall not consider these architectures in this paper. In Section 10 we review different sources of divergence. The study of the partial maps corresponding to these computations are the subject of the forthcoming paper [2].

5.2. Computable and semicomputable maps into $2^{\leqslant\omega}$

The following natural definition will also be supported by Corollary 45 below.

Definition 41. Let $\mathscr{I}$ be among the sets 2^*, 2^ω and $2^{\leqslant\omega}$ ($\mathscr{I}$ stands for "input set") and let $F : \mathscr{I} \to 2^{\leqslant\omega}$ be a total map.

(1) F is *semicomputable* if it is the input/output behaviour of some monotone Turing machine with inputs in $\mathscr{I}$ and possibly infinite computations.

(2) F is *computable* if it is the input/output behaviour of some Turing machine with inputs in $\mathscr{I}$ and possibly infinite computations which halts in case the output is finite.

For instance, the map $input \mapsto run$ is semicomputable for any Turing machine M.

Remark 42. (1) It is easy to check that in the above definition of computable map, one can require that the machine halts exactly when the output is finite and completely written (delay the output of any letter until the next output comes or until the machine halts).

(2) It is clear that total computable maps $2^* \to 2^*$ are exactly recursive ones. However, as concerns semicomputability, infinite computations really add. For instance, let $\varphi : \mathbb{N}^2 \to \mathbb{N}$ be a universal partial recursive function and define $F : 2^* \to \{\lambda, 0\}$ as follows:

$$F(0^n) = \lambda,$$

$$F(0^n 1s) = \texttt{if } \varphi_n(n) \text{ is defined } \texttt{then } 0 \texttt{ else } \lambda,$$

where λ is the empty word. Then F is neither recursive nor computable in the sense of Definition 41, but it is semicomputable.

The following result is trivial.

Proposition 43. *For total maps $F : \mathscr{I} \to 2^\omega$ semicomputability coincides with computability.*

In case $\mathscr{I} = 2^\omega$ or $\mathscr{I} = 2^{\leqslant\omega}$, the next proposition is an effectivized version of the below Proposition 54.

Proposition 44. *Let $F : \mathscr{I} \to 2^{\leqslant\omega}$ be a total map where $\mathscr{I}$ is among the sets 2^*, 2^ω and $2^{\leqslant\omega}$. The following conditions are equivalent*:

(i) *F is computable.*

(ii) *The relation $\{(s,\xi) \mid s \preccurlyeq F(\xi)\}$ (resp. $\prec$) is computable in $2^* \times \mathscr{I}$.*

(iii) *F is semicomputable and the relation $\{(s,\xi) \mid s = F(\xi)\}$ is computable in $2^* \times \mathscr{I}$.*

(iv) *F is semicomputable and the relation $\{(i,\xi) \mid i \leqslant |F(\xi)|\}$, (resp. $<$, resp. $=$) is computable in $\mathbb{N} \times \mathscr{I}$.*

Proof. (i) $\Rightarrow$ (ii)$_\preccurlyeq$: We can decide whether $s \preccurlyeq F(\xi)$ as follows: go on the computation on input ξ until the output gets incomparable or larger than s or M halts (with an output shorter than s). The hypothesis that M halts in case the output is finite insures that this process does stop.

(ii)$_\preccurlyeq \Rightarrow$ (ii)$_\prec$: Observe that $s \prec \eta \Leftrightarrow (s \preccurlyeq \eta \wedge \neg(s0 \preccurlyeq \eta \vee s1 \preccurlyeq \eta))$.

(ii)$_\prec \Rightarrow$ (iii): Consider the monotone Turing machine M which behaves as follows on input ξ:

```
$u := \lambda$
repeat
test $u0 \prec F(\xi)$ and $u1 \prec F(\xi)$
if the first test is positive then output 0 and set $u := u0$
if the second test is positive then output 1 and set $u := u1$
if both are negative then halt
until halt
```

It is easy to see that M semicomputes F. To see that $\{(s,\xi) \mid s = F(\xi)\}$ is computable, observe that $s = \eta \Leftrightarrow (s \prec \eta 0 \wedge \neg(s \prec \eta))$.

(iii) $\Rightarrow$ (iv)$_{\leqslant}$: Observe that $i \leqslant |\eta| \Leftrightarrow \bigwedge_{|u|<i} u \neq \eta$.

(iv)$_{\leqslant} \Rightarrow$ (iv)$_{<} \Rightarrow$ (iv)$_{=}$: Straightforward.

(iv)$_{=} \Rightarrow$ (i): Consider a monotone Turing machine which semicomputes F. Let M' be M modified so that
– at each step M' tests $|s| = |F(\xi)|$ where s is the current output,
– M' halts if and when the test is positive.
Then M' computes F since it halts whenever F has finite value. □

Using Proposition 26, we get

Corollary 45. (1) *A total map $F: 2^{\omega} \to 2^{\leqslant\omega}$ is computable if and only if there exists a recursive set $Y \subset 2^* \times 2^*$ such that, for every $s \in 2^*$,*
– the slice Y_s is finite (where $Y_s = \{u \in 2^* \mid (s,u) \in Y\}$),
– $F^{-1}(s2^{\leqslant\omega}) = Y_s 2^{\omega}$.

(2) *A total map $F: 2^{\leqslant\omega} \to 2^{\leqslant\omega}$ is computable if and only if there exist recursive sets X, $Y \subset (2^*)^2$ such that, for every $s \in 2^*$,*
– the slices X_s, Y_s are finite,
– $F^{-1}(s2^{\leqslant\omega}) = X_s \cup Y_s 2^{\leqslant\omega}$.

5.3. Syntactical complexity of (semi)computable maps

Proposition 46. *Let $F: \mathscr{I} \to 2^{\leqslant\omega}$ be a total map where $\mathscr{I}$ is 2^* or 2^{ω} or $2^{\leqslant\omega}$. The following table gives the syntactical complexity of the predicates*:

$$|F(\xi)| < +\infty, \quad |F(\xi)| \geqslant i, \quad |F(\xi)| = i, \quad s \preccurlyeq F(\xi), \quad s = F(\xi)$$

as relations included in $\mathscr{I}$, $\mathscr{I} \times \mathbb{N}$ and $\mathscr{I} \times 2^*$ (variables ξ, i, s respectively varying in $\mathscr{I}$, $\mathbb{N}$, 2^*), in case F is semicomputable or computable.

F	$F(\xi) \succcurlyeq s$	$\lvert F(\xi)\rvert \geqslant i$	$F(\xi) = s$	$\lvert F(\xi)\rvert = i$	$F(\xi) \in 2^*$
Total comp.	Recursive	Recursive	Recursive	Recursive	Σ^0_1
Total semicomp.	Σ^0_1	Σ^0_1	$\Sigma^0_1 \wedge \Pi^0_1$	$\Sigma^0_1 \wedge \Pi^0_1$	Σ^0_2

No result in the table can be improved.
In particular, the $\Sigma^0_1 \wedge \Pi^0_1$ complexity cannot be replaced by $\Sigma^0_1 \vee \Pi^0_1$.

Proof. (1) *Case F is total semicomputable*. Let F be the input/output behaviour of the Turing machine M and let $K^{\mathscr{I}}(\xi, t, s)$ be the usual Kleene predicate expressing that $s \in 2^*$ is the current output at time $t \in \mathbb{N}$ of the computation of M on input $\xi \in \mathscr{I}$. Using Definition 25, observe that the predicate $K^{\mathscr{I}}(\xi, t, s)$ is recursive in the sense of the space $\mathscr{I} \times \mathbb{N} \times 2^*$.

The following easy equivalences prove the assertions in the table:

$$F(\xi) \succcurlyeq s \Leftrightarrow \exists t \ K^{\mathscr{I}}(\xi,t,s),$$

$$|F(\xi)| \geqslant i \Leftrightarrow \exists s \ \exists t \ (|s| = i \wedge K^{\mathscr{I}}(\xi,t,s)),$$

$$F(\xi) < +\infty \Leftrightarrow \exists s \ \exists t \ \forall t' > t \ K^{\mathscr{I}}(\xi,t',s),$$

$$|F(\xi)| = i \Leftrightarrow (|F(\xi)| \geqslant i \wedge \neg(|F(\xi)| \geqslant i+1)),$$

$$F(\xi) = s \Leftrightarrow (s \preccurlyeq F(\xi) \wedge \neg(s0 \preccurlyeq F(\xi) \vee s1 \preccurlyeq F(\xi))).$$

(2) *Case F is total computable. Use Proposition* 44.

(3) *Optimality. Case* $F(\xi) \in 2^*$, $F(\xi) \succcurlyeq s$ *and* $|F(\xi)| \geqslant i$.

In case $\mathscr{I} = 2^*$, we reduce these problems to classical complete problems. In fact, consider a universal Turing machine M such that on input $u \in 2^*$:

(1) M dovetails all computations of the partial recursive function φ_u: $\mathbb{N} \to \mathbb{N}$ (with code u) on inputs $0, 1, 2, \ldots$.

(2) M outputs 1 each time (resp. the first time) it finds some new point in the domain of φ_u.

Clearly, the output of M on input u is finite (resp. is 1) if and only if φ_u has finite domain (resp. has non-empty domain), which is known to be a Σ^0_2 (resp. Σ^0_1) complete problem.

In case $\mathscr{I} = 2^\omega$, let F be the input/output behaviour of the machine M which, at step t:

- reads the tth letter of its input $\alpha \in 2^\omega$,
- outputs a 1 if this letter is 1 (else it outputs nothing).

It is clear that $F(\alpha)$ is a finite word (resp. $1 \preccurlyeq F(\alpha)$) if and only if α has finitely many 1's (resp. at least one 1). Since it is well known that the set of such α's is not G_δ (resp. not closed), it cannot be Π^0_2 (resp. Π^0_1).

For the case $\mathscr{I} = 2^{\leqslant\omega}$, we can use the same machine M on inputs in $2^{\leqslant\omega}$. Since 2^ω is a Π^0_1 subset of $2^{\leqslant\omega}$ and the trace on 2^ω of the relation $|F(\xi)| < +\infty$ (resp. $1 \preccurlyeq F(\xi)$) is not Π^0_2 (resp. Π^0_1) in 2^ω, this relation cannot be Π^0_2 (resp. Π^0_1) in $2^{\leqslant\omega}$ (cf. Proposition 33).

(4) *Optimality. Case* $F(\xi) = s$ *and* $|F(\xi)| = i$. Consider Kolmogorov complexity K: $\mathbb{N} \to \mathbb{N}$. As is well-known, K has a linear bound: $K(x) \leqslant x + c$ for some constant c.

Observe that the function $x \mapsto x + c - K(x)$ is semicomputable with respect to unary representation of integers:

- dovetail over all computations of a universal function on length increasing inputs in 2^*,
- if and when some computation on input $\mathtt{p}$ halts and outputs x (in unary) then increase the current output to $x - |\mathtt{p}|$.

The relation $y = K(x)$ is $\Sigma^0_1 \wedge \Pi^0_1$. It is proved in Ferbus and Grigorieff [13] that it is not $\Sigma^0_1 \vee \Pi^0_1$. Since $y = F(x) \Leftrightarrow x + c - y = K(x)$, we see that the graph of F is $\Sigma^0_1 \wedge \Pi^0_1$ and not $\Sigma^0_1 \vee \Pi^0_1$. □

6. Topological counterpart of (semi)computability

In this section we assume *total maps* $\mathscr{I} \to \mathscr{O}$, where $\mathscr{I}, \mathscr{O}$ vary in $2^*, 2^\omega, 2^{\leqslant\omega}$. The analysis of continuity and computability of partial maps will be treated in [2].

As is well known, computable maps $2^\omega \to 2^\omega$ are continuous. Indeed, for maps $2^\omega \to 2^\omega$, computability is the effectivization of continuity.

Whereas there is a unique notion of computability for maps with values in 2^ω (cf. Remark 42, point 1), when values in $2^{\leqslant\omega}$ are allowed, there are two notions: computability and semicomputability (cf. Definition 41). Their topological counterparts involve continuity and lower semicontinuity.

6.1. (Semi)computability and (lower semi)continuity

The classical notion of lower semicontinuity for real valued functions has an analog for functions with values in $2^{\leqslant\omega}$ with respect to the prefix ordering on this space. It happens that this notion is the topological counterpart of semicomputability, cf. Theorem 51 below.

Definition 47. Let $\mathscr{I}$ be 2^ω or $2^{\leqslant\omega}$. A total map $F : \mathscr{I} \to 2^{\leqslant\omega}$ is *lower semicontinuous* at $\xi \in \mathscr{I}$ if for all $n \in \mathbb{N}$ there exists a neighbourhood $\mathscr{V}$ of ξ such that

$$\forall \eta \in \mathscr{V} \quad F(\eta) \succcurlyeq F(\xi) \restriction n$$

(recall that if $F(\xi)$ is finite then $F(\xi) \restriction n$ is the prefix of $F(\xi)$ with length $\min(n, |F(\xi)|)$).

Example 48. (1) Let $F : 2^\omega \to 2^*$ be defined as follows:
- $F(0^\omega) = \lambda$, $F(0^i 1\alpha) = 0^i$ for every $i \in \mathbb{N}$ and $\alpha \in 2^\omega$. Then F is semicomputable and lower semicontinuous but not continuous at 0^ω.

(2) Let $F : 2^{\leqslant\omega} \to 2^{\leqslant\omega}$ be defined as follows:
- $F(s) = |s|$ for $s \in 2^*$, $F(\alpha) = \lambda$ for every $\alpha \in 2^\omega$.

Then F is everywhere lower semicontinuous and is semicomputable but is discontinuous at every point of 2^ω.

(3) Let $F : 2^{\leqslant\omega} \to 2^{\leqslant\omega}$ be defined as follows:
- $F(0^k) = 0^\omega$, $F(0^k 1\xi) = 1$, $F(0^\omega) = \lambda$.

Then F is everywhere lower semicontinuous and is semicomputable:
- Read the input tape until it finds a blank or a 1.
- If it finds a blank then output infinitely many 0's,
- If it finds a 1 then outputs 1 and halt.

Remark 49. (1) If $F : 2^{\leqslant\omega} \to 2^{\leqslant\omega}$ is continuous then $\lim_{n\to\infty} F(\alpha \restriction n) = F(\alpha)$ for all $\alpha \in 2^\omega$ (i.e., $\overline{F \restriction 2^*} = F \restriction 2^\omega$ with the notation in Definition 59). But this is not necessarily the case for lower semicontinuous maps $2^{\leqslant\omega} \to 2^{\leqslant\omega}$. For instance, consider the map of Example 48, point 3:

$$0^\omega = F(0^k) = \lim_{k\to\infty} F(0^k) \neq F(0^\omega).$$

(2) Being continuous on a compact space, continuous maps are in fact uniformly continuous. However, there is no proper notion of uniform lower semicontinuity. If for all n there is a uniform p such that

$$\forall \alpha, \quad \beta \in 2^{\omega} \beta \upharpoonright p = \alpha \upharpoonright p \Rightarrow F(\beta) \succcurlyeq F(\alpha) \upharpoonright n$$

then, exchanging the roles of α, β, we get

$$\forall \alpha, \quad \beta \in 2^{\omega} \beta \upharpoonright p = \alpha \upharpoonright p \;\Rightarrow\; F(\beta) \upharpoonright n = F(\alpha) \upharpoonright n$$

which is uniform continuity.

The next proposition insures that lower semicontinuity differs from continuity at the sole points having finite image. In particular, the above definition would not be meaningful for maps $2^{\omega} \to 2^{\omega}$ or $2^{\leqslant \omega} \to 2^{\omega}$.

Proposition 50. (1) *Any total map $F: 2^{\leqslant \omega} \to 2^{\omega}$ or $F: 2^{\leqslant \omega} \to 2^{\leqslant \omega}$ is continuous at every point which lies in the subset 2^* of $2^{\leqslant \omega}$.*

(2) *Let $F: \mathscr{I} \to 2^{\leqslant \omega}$ be a total map where $\mathscr{I}$ is 2^{ω} or $2^{\leqslant \omega}$. If F is lower semicontinuous at ξ and $F(\xi) \in 2^{\omega}$ then F is continuous at ξ.*

Proof. (1) Obvious since any singleton word is open in $2^{\leqslant \omega}$.

(2) Observe that if $F(\xi)$ is infinite then the condition $F(\eta) \succcurlyeq F(\xi) \upharpoonright n$ is equivalent to $F(\eta) \upharpoonright n = F(\xi) \upharpoonright n$ which is the usual condition for continuity. □

Theorem 51. (1) *Every total semicomputable map $F: \mathscr{I} \to 2^{\leqslant \omega}$ (where $\mathscr{I}$ is 2^{ω} or $2^{\leqslant \omega}$) is lower semicontinuous.*

(2) *Every total computable map $F: \mathscr{I} \to \mathscr{O}$ (where $\mathscr{I}, \mathscr{O}$ are 2^{ω} or $2^{\leqslant \omega}$) is continuous.*

Proof. Case $\xi \in 2^*$. Then $\{\xi\}$ is a neighbourhood of ξ and F is continuous at ξ (cf. Proposition 50).

Case $\xi \in 2^{\omega}$. (1) Let p be the length of the input which has been read when the last letter of $F(\xi) \upharpoonright n$ is output. It is clear that $\forall \eta \succ (\xi \upharpoonright p)\ F(\eta) \succcurlyeq F(\xi) \upharpoonright n$.

(2) If $F(\xi) \in 2^{\omega}$ then $F(\eta) \succcurlyeq F(\xi) \upharpoonright n \Rightarrow F(\eta) \upharpoonright n = F(\xi) \upharpoonright n$, which yields continuity a ξ. If $F(\xi) \in 2^*$ then the machine halts at some step t and F is constant on $(\xi \upharpoonright t)\mathscr{I}$ hence continuous at ξ. □

Remark 52. If a map $2^{\leqslant \omega} \to 2^{\leqslant \omega}$ is semicontinuous (resp. semicomputable) and length preserving then its is continuous (resp. computable).

The following results delimitate the interaction between topology and computability.

Proposition 53. *There exists a total continuous map $F: 2^{\omega} \to 2^{\leqslant \omega}$ which is semicomputable but not computable.*

Proof. Define F as the following variation of Example 42:
– $F(0^\omega)=0^\omega$.
– $F(0^n1\alpha)=$ IF $\varphi_n(n)$ is defined THEN 0^ω ELSE 0^n.
It is clear that F is sequentially continuous hence continuous. It is easy to check that F is semicomputable:

Output 0 while the head reads 0.
After the first 1 has appeared there is no more output until the computation of $\varphi_n(n)$ halts.
If this happens then do not halt and output 0 forever.

However, F cannot be computable since $F(0^n1\alpha)$ is finite if and only if $\varphi_n(n)$ is undefined, which is an undecidable problem. □

Note 1. The missing hypothesis to get computability from continuity is the recursive enumerability of the family of basic open sets on which F is constant. Cf. Lemmas 75 and 77.

6.2. Lower semicontinuity and the weak topology

Lower semicontinuity can also be expressed as continuity with respect to the weak topology on the range space $2^{\leqslant\omega}$ (cf. Section 3.6.1). The next proposition sums up this characterization together with the related one for continuity.

Proposition 54. (1) *Let $F:\mathscr{I}\to\mathcal{O}$ be a total map where $\mathscr{I},\mathcal{O}$ are 2^ω or $2^{\leqslant\omega}$ and let $\alpha\in 2^\omega\cap\mathscr{I}$. The following conditions are equivalent*:
(i) *F is continuous on $\mathscr{I}$ (resp. at α) with respect to the compact topologies on the domain and range spaces.*
(ii) *For all $s\in 2^*$ the set $F^{-1}(s\mathcal{O})$ is clopen in $\mathscr{I}$ (resp. for all $n\in\mathbb{N}$ the set $F^{-1}((F(\alpha)\upharpoonright n)\mathcal{O})$ is clopen in $\mathscr{I}$).*

(2) *Let $F:\mathscr{I}\to 2^{\leqslant\omega}$ be a total map where $\mathscr{I}$ is 2^ω or $2^{\leqslant\omega}$ and let $\alpha\in 2^\omega\cap\mathscr{I}$. The following conditions are equivalent*:
(i) *F is lower semicontinuous on $\mathscr{I}$ (resp. at α) with respect to the compact topologies on the domain and range spaces.*
(ii) *F is continuous on $\mathscr{I}$ (resp. at α) with respect to the compact topology on the domain space and the weak topology on the range space $2^{\leqslant\omega}$.*
(iii) *For all $s\in 2^*$ the set $F^{-1}(s2^{\leqslant\omega})$ is open in $\mathscr{I}$ (resp. for all $n\in\mathbb{N}$ the set $F^{-1}((F(\alpha)\upharpoonright n)2^{\leqslant\omega})$ is open in $\mathscr{I}$).*

Proof. (1) (i)$\Rightarrow$(ii): Observe that the basic open set $s2^{\leqslant\omega}$ is clopen in $2^{\leqslant\omega}$. Since F is continuous, $F^{-1}(s\mathcal{O})$ is also clopen in $\mathscr{I}$.

(ii)$\Rightarrow$(i): Trivial if $\mathcal{O}=2^\omega$ since all basic open sets are of the form $s\mathcal{O}$. In case $\mathcal{O}=2^{\leqslant\omega}$, observe that the basic open set $\{s\}$ of $2^{\leqslant\omega}$ can be expressed as a boolean combination

$$\{s\}=s2^{\leqslant\omega}\backslash(s02^{\leqslant\omega}\cup s12^{\leqslant\omega}),$$

so that its inverse image is a boolean combination of clopen sets, hence is also clopen.

(2) (i)$\Leftrightarrow$(ii): Inequality $F(\beta) \succcurlyeq F(\alpha) \upharpoonright n$ asserts that $F(\beta)$ belongs to $(F(\alpha) \upharpoonright n)2^{\leqslant\omega}$. Since these sets are exactly the weak basic neighbourhoods of $F(\alpha)$, we see that lower semicontinuity exactly expresses that the inverse images of the weak neighbourhoods of $F(\alpha)$ contain neighbourhoods of α. Which is continuity with respect to the weak topology on the range space $2^{\leqslant\omega}$. □

Remark 55. In relation with point 2(iii) of Proposition 54, observe that if $F : \mathscr{I} \to 2^{\leqslant\omega}$ is lower semicontinuous then

$$F^{-1}(\{s\}) = F^{-1}(s2^{\leqslant\omega}) \backslash (F^{-1}(s02^{\leqslant\omega}) \cup F^{-1}(s12^{\leqslant\omega}))$$

is the difference of two open sets.

Finally, let us mention the case when F has values in 2^*.

Proposition 56. *Let $F : \mathscr{I} \to 2^{\leqslant\omega}$ be a total map where $\mathscr{I}$ is 2^ω or $2^{\leqslant\omega}$:*
1. *If F is continuous and $range(F) \subseteq 2^*$ then $range(F)$ is finite.*
2. *If F is lower semicontinuous and $range(F) \subseteq 2^*$ then $\min(range(F))$ is a finite prefix-free set.*

Proof. (1) Observe that the range of F is compact (as is $\mathscr{I}$) and 2^* is a discrete subspace of $2^{\leqslant\omega}$.

(2) We know that any $F^{-1}(s2^{\leqslant\omega})$ is weak open hence open for the compact topology on $\mathscr{I}$. Now, $range(F) \subseteq \bigcup_{s \in \min(range(F))} s2^{\leqslant\omega}$ so that the $F^{-1}(s2^{\leqslant\omega})$, $s \in \min(range(F))$ constitute a partition of $\mathscr{I}$. By compactness, such a partition is necessarily finite. Hence $\min(range(F))$ is finite. □

Remark 57. However, the range of a lower semicontinuous map may contain infinite prefix-free sets. For instance, let $F(0^n) = F(0^\omega) = \lambda$ and $F(0^n1\xi) = 0^n1$ for all $\xi \in 2^{\leqslant\omega}$.

6.3. Continuity and weak topology on both the domain and range spaces

Proposition 54 considers the weak topology on the range space. Endowing the domain space with the weak topology leads to a completely different picture.

Proposition 58. *We consider on 2^* and $2^{\leqslant\omega}$ the weak topologies, cf. Section 3.6.1.*
1. *$f : 2^* \to 2^{\leqslant\omega}$ is weak continuous if and only if it is monotone increasing.*
2. *$F : 2^{\leqslant\omega} \to 2^{\leqslant\omega}$ is weak continuous if and only if it is monotone increasing and continuous with respect to the compact topology on $2^{\leqslant\omega}$.*

Proof. 1 $\Rightarrow$. Weak continuity of f at $s \in 2^*$ expresses that the inverse image of any neighbourhood of $f(s)$ is a neighbourhood of s. Since s has a smallest weak neighbourhood, namely $s2^*$, this amounts to the inclusion of $f(s2^*)$ into any neighbourhood

of $f(s)$, i.e.
– if $f(s)$ is finite then $f(s2^*) \subseteq f(s)2^{\leqslant\omega}$,
– if $f(s)$ is infinite then $\forall n\ f(s2^*) \subseteq (f(s) \upharpoonright n)2^{\leqslant\omega}$, hence $f(s2^*) = \{f(s)\}$.
Thus, f is monotone increasing.

1 $\Leftarrow$ and 2 $\Leftarrow$ are easy.

2 $\Rightarrow$. Monotonicity is proved as in 1 $\Rightarrow$. We prove continuity for the compact topologies.

If $F(\xi) \in 2^\omega$ then it has the same neighbourhoods for the compact and weak topologies. Their inverse images by F are neighbourhoods of ξ for the weak topology hence also for the compact one.

If $F(\alpha) \in 2^*$ then $F(\alpha)2^{\leqslant\omega}$ is the smallest weak neighbourhood of $F(\alpha)$ and weak continuity insures that there exists p such that $(F((\alpha \upharpoonright p)2^{\leqslant\omega}) \subseteq F(\alpha)2^{\leqslant\omega}$. Since F is monotonous, this means that F is constant with value $F(\alpha)$ on $(\alpha \upharpoonright p)2^{\leqslant\omega}$. Whence F is also continuous at α with respect to the compact topologies. □

7. Tools for representation of maps

Let $\mathscr{I}, \mathscr{O}$ be 2^ω or $2^{\leqslant\omega}$. In this section and the two next ones, we look at diverse ways to represent maps $F: \mathscr{I} \to \mathscr{O}$ using maps $f: 2^* \to 2^*$ or maps $f: 2^* \to 2^{\leqslant\omega}$.

In the bottom-up approach, we go from f to F, whereas in the top-down approach we get f from F.

7.1. Bottom-up approach: the bar operator on maps $2^* \to 2^*$

Definition 59. Let $f: 2^* \to 2^*$ or $f: 2^* \to 2^{\leqslant\omega}$. We denote $\overline{f}: 2^\omega \to 2^{\leqslant\omega}$ the map such that, for all $\alpha \in 2^\omega$:

$$\overline{f}(\alpha) = \underline{\lim}_{n\to\infty} f(\alpha \upharpoonright n) = \sup(\{gcp(\{f(\alpha \upharpoonright p):\ p \geqslant n\}):\ n \in \mathbb{N}\}),$$

where the sup is relative to the prefix ordering on $2^{\leqslant\omega}$ (cf. Section 3) and $gcp(\mathscr{X})$ denotes the greatest common prefix of all elements of $\mathscr{X}$.

In case f is monotone increasing with respect to the prefix ordering then $\overline{f}(\alpha) = \sup_{n\in\mathbb{N}} f(\alpha \upharpoonright n)$.

The next definition formalizes some notions about maps $f: 2^* \to 2^{\leqslant\omega}$ which are related to continuity or computability of the associated $\overline{f}: 2^\omega \to 2^{\leqslant\omega}$.

Definition 60. Let $f: 2^* \to 2^{\leqslant\omega}$. We define

$Open_{<\omega}(f) = \{s \in 2^* \mid f$ *is constant on* $s2^*$ *with finite value*$\}$,
$Tree(f) = 2^* \setminus Open_{<\omega}(f)$.

Proposition 61. *Let* $f: 2^* \to 2^{\leqslant\omega}$. *Then* $Open_{<\omega}(f)$ *is closed by extensions while* $Tree(f)$ *is a tree, i.e. is prefix closed.*

Definition 62. Let f: $2^* \to 2^{\leqslant\omega}$ (resp. f: $2^* \to \mathbb{N}$) and let $T \subseteq 2^*$ be a tree, i.e. T is prefix closed.

We say that f is *totally unbounded* on T if $\overline{f}(\alpha) \in 2^\omega$ for every infinite branch α of T (i.e. $\forall n\ \alpha \upharpoonright n \in T$).

In case f is monotone increasing, this is equivalent to $\lim_{n\to\infty} f(\alpha \upharpoonright n) \in 2^\omega$ for every infinite branch α of T.

In case $T = 2^*$ (resp. $T = Tree(f)$), we simply say that f is totally unbounded (resp. totally unbounded on its tree).

Proposition 63. *Let* $f: 2^* \to 2^{\leqslant\omega}$ (*resp.* $f: 2^* \to \mathbb{N}$) *be monotone increasing and let* $T \subseteq 2^*$ *be an* infinite *tree. Then* f *is totally unbounded on* T *if and only if*

$$\lim_{n\to\infty} \left(\min_{s\in T,\ |s|=n} |f(s)| \right) = +\infty.$$

Proof. $\Leftarrow$ is trivial. For the $\Rightarrow$ direction, consider the tree $T_k = \{s \in T:\ |f(s)| \leqslant k\}$. Assuming f is totally unbounded on T, a direct application of König's Lemma and of the monotonicity of f shows that T_k is finite. Whence the desired conclusion. □

7.2. Bottom-up approach with maps $2^* \times \mathbb{N} \to 2^*$ monotone increasing with respect to $\mathbb{N}$

Definition 64. Let v: $2^* \times \mathbb{N} \to 2^*$ be monotone increasing in the second argument with respect to the prefix ordering. We let $\overline{v}: 2^* \to 2^{\leqslant\omega}$ be the map such that if $s \in 2^*$ then

$$\overline{v}(s) = \sup\{v(s,n):\ n \in \mathbb{N}\},$$

where the sup is relative to the prefix ordering on $2^{\leqslant\omega}$.

Remark 65. The intuition behind maps v is the input/output behaviour of Turing machines such that

$$v(u,t) = \text{current output at time } t \text{ on input } u$$

(independently of whether or not u has been completely read). These functions give a natural characterization of (semi)computable maps $2^* \to 2^{\leqslant\omega}$ (cf. Theorem 81).

Fact 66. *Every map* F: $2^* \to 2^{\leqslant\omega}$ *is* $F = \overline{v}$ *for some* v: $2^* \times \mathbb{N} \to 2^*$ *monotone increasing on* $\mathbb{N}$.

Proof. $\Rightarrow$. Let $v(u,n) = F(u) \upharpoonright n$. $\Leftarrow$. Obvious. □

The next definition formalizes some notions about maps v: $2^* \times \mathbb{N} \to 2^*$ which are related to continuity or computability of $\overline{v}$: $2^* \to 2^{\leqslant\omega}$.

Definition 67. Let v: $2^* \times \mathbb{N} \to 2^*$ be monotone increasing in the second argument with respect to the prefix ordering:

- $Open(v) = \{(s,n)$: $s \in 2^*$ and $\forall m > n\ v(s,n) = v(s,m)\}$,
- $Tree(v) = (2^* \times \mathbb{N}) \backslash Open(v)$.

7.3. Tools for the top-down approach

We introduce some material that we will constantly use in Section 8.2. Maps ∂F and $\partial^\theta F$ are the main tools for the representation Theorem 81.

Definition 68. We denote $gcp(X)$ the greatest common prefix of a set $X \subseteq 2^{\leqslant\omega}$.

Let F: $\mathscr{I} \to 2^{\leqslant\omega}$, $\mathscr{I} = 2^*, 2^\omega, 2^{\leqslant\omega}$.

1. ∂F: $2^* \to 2^{\leqslant\omega}$ denotes the map $s \mapsto gcp(F(s\mathscr{I}))$, i.e. $\partial F(s)$ is the longest $\xi \in 2^{\leqslant\omega}$ such that $\forall \eta \in s\mathscr{I}\ \ \xi \preccurlyeq F(\eta)$.
2. θ: $2^* \to \mathbb{N}$ is totally unbounded if $\lim_{n\to\infty}(\min_{|s|=n} |\theta(s)|) = \infty$.
3. If θ: $2^* \to \mathbb{N}$ be totally unbounded, we let $\partial^\theta F : 2^* \to 2^*$ be the map such that $\partial^\theta F(s)$ is the longest $u \in 2^*$ such that $|u| \leqslant \theta(s)$ and $u \preccurlyeq F(\eta)$ for all $\eta \in s\mathscr{I}$.

Remark 69. The total unboundedness of $\theta(s)$ in Point 3 is to insure that $\partial^\theta F$ has range in 2^*. The simplest example of such a θ is the length function.

The following proposition is easy.

Proposition 70. *Let $\theta : 2^* \to \mathbb{N}$ be totally unbounded.*

1. *Case F: $2^* \to 2^{\leqslant\omega}$. Then ∂F (resp. $\partial^\theta F$) is the largest monotone increasing map $f : 2^* \to 2^{\leqslant\omega}$ such that $f \preccurlyeq F$ (resp. $f \preccurlyeq F$ and $|f| \leqslant \theta$).*
   In particular, if F is monotone increasing then $\partial F = F$ and $\partial^\theta F(s) = F(s) \upharpoonright \theta(s)$.
2. *Case F: $2^\omega \to 2^{\leqslant\omega}$. Then ∂F (resp. $\partial^\theta F$) is the largest monotone increasing map f: $2^* \to 2^{\leqslant\omega}$ such that $\overline{f} \preccurlyeq F$ (resp. $\overline{f} \preccurlyeq F$ and $|f| \leqslant \theta$) where $\overline{f}$ is as in Definition 59. Thus, $\overline{\partial^\theta F} \preccurlyeq \overline{\partial F} \preccurlyeq F$.*
   Also, $Open(\partial^\theta F)\ 2^\omega \subseteq Open(\partial F)\ 2^\omega$.
3. *Case F: $2^{\leqslant\omega} \to 2^{\leqslant\omega}$:*

$$\partial F(s) = gcp(\{(\partial(F \upharpoonright 2^*))(s), (\partial(F \upharpoonright 2^\omega))(s)\},$$
$$\partial^\theta F(s) = gcp(\{(\partial^\theta(F \upharpoonright 2^*))(s), (\partial^\theta(F \upharpoonright 2^\omega))(s)\}.$$

Thus, ∂F (resp. $\partial^\theta F$) is the largest monotone increasing map f: $2^* \to 2^{\leqslant\omega}$ such that

$$f \preccurlyeq F \upharpoonright 2^* \text{ and } \overline{f} \preccurlyeq F \upharpoonright 2^\omega \text{ (resp. and } |f| \leqslant \theta).$$

7.4. Some lemmas about the bottom-up representation

The following lemmas will be used in the proof of the representation Theorem 81.

Lemma 71. *Let f: $2^* \to 2^{\leqslant\omega}$ be monotone increasing such that $\overline{f}$: $2^\omega \to 2^{\leqslant\omega}$ is continuous. Then f is necessarily totally unbounded on Tree(f).*

Proof. If f were not totally unbounded on *Tree*(f) then there would exist an infinite branch α of *Tree*(f) on which f is stationary:
– $\exists i \in \mathbb{N}\ \exists u \in 2^*\ \forall j \geqslant i\ f(\alpha \restriction i) = u$.
– $\forall j\ \alpha \restriction j \in Tree(f)$, whence $\forall j \geqslant i\ \exists v_j f((\alpha \restriction j)v_j) \succ f(\alpha \restriction j) = u$.
Continuity of $\overline{f}$ at α insures that there exists j such that $f((\alpha \restriction j)\beta) = u$ for all $\beta \in 2^\omega$. Taking $\beta \succ v_j$ we get $\overline{f}((\alpha \restriction j)\beta) \succcurlyeq f((\alpha \restriction j)v_j) \succ u$, contradiction. □

Remark 72. The previous result is false if f is not monotonous. For instance, let f be 0 on 0^*1 and 1 elsewhere. Then $\overline{f}$ is constant with finite value 1 hence continuous. Also, $Tree(f) = 0^* \cup 0^*1$ has an infinite branch, namely 0^ω, on which $\overline{f}$ has finite value.

Lemma 73. *Let* $f, g : 2^* \to 2^{\leqslant\omega}$.
1. *If* $\overline{f} = \overline{g}$ *then* f *and* g *coincide on* $Open_{<\omega}(f) \cap Open_{<\omega}(g)$.
2. *If* $\overline{f} = \overline{g}$ *and* f, g *are totally unbounded on their trees then* $\widehat{Open_{<\omega}(f)} = \widehat{Open_{<\omega}(g)}$, *i.e.* $Open_{<\omega}(f)2^\omega = Open_{<\omega}(g)2^\omega$.
3. *If* f *is monotone increasing and totally unbounded on Tree*(f) *and* $f \preccurlyeq g$ *then* $Open_{<\omega}(g) \subseteq \widehat{Open_{<\omega}(f)}$, *i.e.* $Open_{<\omega}(g)2^\omega \subseteq Open_{<\omega}(f)2^\omega$.

Proof. (1) If $s \in Open_{<\omega}(f) \cap Open_{<\omega}(g)$ then $\overline{f}$ and $\overline{g}$ are constant on $s2^\omega$ with respective values $f(s)$ and $g(s)$. Since $\overline{f} = \overline{g}$ we get $f(s) = g(s)$.

(2) Using symmetry and the monotonicity and idempotence of the hat operation, we reduce to prove $Open_{<\omega}(g) \subseteq \widehat{Open_{<\omega}(f)}$.

Let $s \in Open_{<\omega}(g)$ and suppose s is not in $\widehat{Open_{<\omega}(f)}$. Since $Open_{<\omega}$ is closed by extension, Proposition 5, point 2 yields $\alpha \in 2^\omega$ such that $s \prec \alpha$ and no prefix of α is in $Open_{<\omega}(f)$. Using the total unboundedness of f on *Tree*(f), we get $\overline{f}(\alpha) \in 2^\omega$. But (by definition of $Open_{<\omega}$) g is constant on $s2^*$ with finite value $g(s) \in 2^*$, so that $\overline{f}(\alpha) = \overline{g}(\alpha) = g(s)$ cannot be in 2^ω, contradiction.

(3) Let $s \in Open_{<\omega}(g)$ and suppose s is not in $\widehat{Open_{<\omega}(f)}$. As above we get $\alpha \in 2^\omega$ such that $s \prec \alpha$ and no prefix of α is in $Open_{<\omega}(f)$. Using the total unboundedness of f on *Tree*(f), we get $\overline{f}(\alpha) \in 2^\omega$. But $f \preccurlyeq g$ and g is constant with value $g(s)$ on $s2^*$, so that $\overline{f}(\alpha) \preccurlyeq s$ cannot be in 2^ω, contradiction. □

7.5. Effectiveness and the bottom-up representation

Lemma 74. (1) *Suppose*
– f, g: $2^* \to 2^{\leqslant\omega}$ *are semicomputable*,
– f *is monotone increasing*,
– $f \preccurlyeq g$.
Then $\overline{f} \cup g$: $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ *is semicomputable.*

(2) *If* f: $2^* \to 2^{\leqslant\omega}$ *is monotone increasing and semicomputable then* $\overline{f}$: $2^\omega \to 2^{\leqslant\omega}$ *is semicomputable.*

Proof. (1) Let M_1 and M_2 be monotone Turing machines which semicompute f and g. We define a monotone Turing machine M which, on input $\xi \in 2^{\leqslant\omega}$, behaves as follows:

(i) *Phase* 1: M behaves according to the following program (assume the reading head of M is initially in the first dummy cell).

```
$s := \lambda$;
$a := \lambda$
do while $a \neq$ blank
     $s := sa$
     Read in $a$ the next symbol from the input tape
     Print the needed symbols to leave on the output tape the current approximation
of $f(s)$ computed by $M_1$ at time $|s|$.
end do
```

(ii) *Phase* 2: $a =$ `blank`. I.e. $\xi \in 2^*$. M starts emulating M_2.
Since $f \preccurlyeq g$, this emulation of M_2 can be faithful: M can output $u \in 2^*$ such that $Output_{M_2}(|\xi|, \xi) = Output_{M_i}(|\xi|, \xi)u$ (where $Output_{M_i}(t, \xi)$ is the current output of M_i at time t).

If $\xi \in 2^\omega$ then M emulates the sole machine M_1, so that it outputs $\overline{f}(\alpha)$.

If $\xi \in 2^*$ then M eventually emulates M_2, hence outputs $g(\xi)$.

Thus, the input/output behaviour of M is $\overline{f} \cup g$.

(2) Restricted to inputs in 2^ω, the above machine M never enters Phase 2 and semicomputes $\overline{f}$. □

Lemma 75. (1) *Suppose* – $f, g : 2^* \to 2^{\leqslant\omega}$ *are computable,*
– f *is monotone increasing and totally unbounded on* $Tree(f)$,
– $f \preccurlyeq g$ *and* $\overline{f} = \overline{g}$,
– $Open_{<\omega}(g)$ *is recursively enumerable.*
Then $\overline{f} \cup g$: $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ *is computable.*

(2) *If* f: $2^* \to 2^{\leqslant\omega}$ *is computable and* $Open_{<\omega}(f)$ *is recursively enumerable then* $\overline{f}$: $2^\omega \to 2^{\leqslant\omega}$ *is computable.*

Proof. (1) Suppose M_1 and M_2 halt when their output is finite. Let Ω: $\mathbb{N} \to 2^*$ be a recursive map with range $Open_{<\omega}(g)$. We modify the machine M introduced in the proof of Lemma 74 as follows:

> If at some step of Phase 1 some prefix s of the part already read of the input appears in the enumeration of $Open_{<\omega}(g)$ by Ω then M starts emulating M_2 on input s.
> (Since $f \preccurlyeq g$ and g is constant on $s2^*$, the current output is a prefix of $g(s)$ so that there is no problem to perform this emulation).

We show that M *computes* F.

Case $\xi \in 2^*$ *and no prefix of* ξ *in* $Open_{<\omega}(g)$ *appears during Phase 1.*
M eventually enters Phase 2 and emulates M_2, hence outputs $g(\xi) = (\overline{f} \cup g)(\xi)$. In case $g(\xi) \in 2^*$, the emulation of M_2 will halt, hence so will M.

Case $\xi \in 2^*$ *and some prefix of* ξ *in* $Open_{<\omega}(g)$ *appears during Phase 1.*

Let s be this prefix. Then M emulates M_2 on input s and output $g(s)$. Since $s\in Open_{<\omega}(g)$ and $s\preccurlyeq\xi$ we have $g(s)=g(\xi)=(\overline{f}\cup g)(\xi)$.
In case $g(s)\in 2^*$, the emulation of M_2 will halt, hence so will M.
Case $\xi\in 2^\omega$ *and some prefix of* ξ *is in* $Open_{<\omega}(g)$.
Such a prefix s necessarily appears during Phase 1, so that M outputs $g(s)$. Since $\overline{f}=\overline{g}$ and $s\in Open_{<\omega}(g)$, Lemma 73 insures that
– some prefix $s'\succcurlyeq s$ of ξ is in $Open_{<\omega}(f)$
– $f(s')=g(s')$.
Since f is constant on $s'2^*$ and g is constant on $s2^*$, we see that $\overline{f}(\xi)=f(s')=g(s)$. Thus, the output $g(s)$ of M is equal to $\overline{f}(\xi)=(\overline{f}\cup g)(\xi)$.
In case $g(s)\in 2^*$, the emulation of M_2 will halt, hence so will M.

Case $\xi\in 2^\omega$ *and no prefix of* ξ *is in* $Open_{<\omega}(g)$.

Then M emulates the sole machine M_1, hence outputs $\overline{f}(\xi)=(\overline{f}\cup g)(\xi)$.

Using Lemma 73, point 2, we see that no prefix of ξ is in $Open_{<\omega}(f)$. So all are in $Tree(f)$. Since f is totally unbounded on $Tree(f)$, we have $\overline{f}(\xi)\in 2^\omega$. Thus, there is no demand to halt.

(2) Applying point 1 with $g=f$, we see that $\overline{f}\cup f:2^{\leqslant\omega}\to 2^{\leqslant\omega}$ is computable. A fortiori, its restriction to 2^ω, which is $\overline{f}$, is computable. □

The following result explains why we make no hypothesis on $Open_{<\omega}(f)$ in point 1 of the above lemma.

Proposition 76. (1) *If f is monotone increasing and $f\preccurlyeq g$ and $\overline{f}=\overline{g}$ then*

(i) $Open_{<\omega}(g)\subseteq\widehat{Open_{<\omega}(f)}=\widehat{Open_{<\omega}(g)}$,

(ii) $s\in Open_{<\omega}(f)$ *if and only if there exists a finite maximal prefix-free set Z such that $sZ\subseteq Open_{<\omega}(g)$ and f,g are constant on sZ with value $f(s)$.*

(2) *Suppose f is monotone increasing and $f\preccurlyeq g$ and $\overline{f}=\overline{g}$ and f,g are computable. If $Open_{<\omega}(g)$ is recursively enumerable then so is $Open_{<\omega}(f)$.*

Proof. (1)(i) Suppose $s\in Open_{<\omega}(g)$. Then g is constant on $s2^*$ so that $\overline{g}$ is constant on $s2^\omega$ with value $g(s)$. Thus, $\overline{f}$ is also constant on $s2^\omega$ with value $g(s)$. Let Z be the set of shortest u's such that $f(su)=g(s)$. Then Z is prefix-free and $s2^\omega=sZ2^\omega$. Applying Proposition 3, we see that Z is finite maximal prefix-free. Since f is monotone increasing, f is constant on $sZ2^*$. Thus, $s\in\widehat{Open_{<\omega}(f)}$ (cf. Definition 4).

To prove equality $\widehat{Open_{<\omega}(f)}=\widehat{Open_{<\omega}(g)}$ it suffices to prove inclusion $Open_{<\omega}(f)\subseteq\widehat{Open_{<\omega}(g)}$.

Suppose $s\in Open_{<\omega}(f)$. Then, $\overline{f}$ is constant on $s2^\omega$ with value $f(s)$. Hence so is g. Let Z be the set of u's such that g is constant on $su2^*$ with value $f(s)$. Then $Z2^\omega=2^\omega$. Applying Proposition 3, we see that Z contains a finite maximal prefix-free Z'. This proves that $sZ'\subseteq Open_{<\omega}(g)$ hence that $s\in\widehat{Open_{<\omega}(g)}$.

(ii) We have just proved that if $s\in Open_{<\omega}(f)$ then there exists a finite maximal prefix-free Z' such that $sZ'\subseteq Open_{<\omega}(g)$ and g is constant with value $f(s)$ on sZ'. Since f is constant on $s2^*$, it is a fortiori constant on sZ'.

Conversely, suppose $sZ \subseteq Open_{<\omega}(g)$ and Z is finite maximal prefix-free and f, g are constant on sZ with value $f(s)$. Monotonicity of f insures that

(*) f is constant on $sPrefix(Z)$ with value $f(s)$.

Since $sZ \subseteq Open_{<\omega}(g)$, for all $z \in Z$ g is constant on $sz2^*$. But g is also constant on sZ with value $f(s)$. Therefore,

(**) f is constant on $sZ2^*$ with value $f(s)$.

Since Z is maximal prefix-free, grouping (*) and (**), we see that f is constant on $s2^*$, i.e. $s \in Open_{<\omega}(f)$.

(2) Condition (ii) of Point 1 gives a definition of $Open_{<\omega}(f)$ as an r.e. set if $Open_{<\omega}(g)$ is r.e. □

Lemma 77. *If* F: $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ *is computable then* $Open_{<\omega}(F \restriction 2^*)$ *is recursively enumerable.*

Proof. Let M be a monotone Turing machine which computes F. Set

$$Open^M = \{s \in 2^* : M \text{ halts on input } s \text{ and the input head has never moved right to } s\}.$$

$Open^M$ is clearly recursively enumerable. Let us prove

$$(*)\ Open^M \subseteq Open_{<\omega}(F \restriction 2^*) \subseteq \widehat{Open}^M.$$

Left inclusion is straightforward. We prove the right one.

Let $s \in Open_{<\omega}(F \restriction 2^*)$. If s were not in $\widehat{Open}^M$, then (using Proposition 5) there would be some $\alpha \in 2^\omega$ such that $s \prec \alpha$ and $\alpha \restriction n \notin X$ for all n. Since $s \prec \alpha$ we have $F(\alpha \restriction n) = F(s)$ for all $n \geqslant |s|$. Since F is continuous (Theorem 51), we have $F(\alpha) = F(s) \in 2^*$. Thus, M halts on input α at some time t and $\alpha \restriction t$ is necessarily in $Open^M$, a contradiction.

Propositions 5, 3 and (*) lead to the following definition of $Open_{<\omega}(F \restriction 2^*)$ as a recursively enumerable set:

$$s \in Open_{<\omega}(F \restriction 2^*) \Leftrightarrow \exists Z\ (Z \text{ is a finite maximal prefix-free set} \wedge sZ \subset Open^M \wedge \forall z \in Z\ \forall u \preccurlyeq z\ F(su) = F(s)). \qquad \square$$

7.6. *Effectiveness of the top-down approach*

The next theorem insures that the operator ∂ in the top-down approach is effective for maps $2^\omega \to 2^{\leqslant\omega}$ or $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ but not for maps $2^* \to 2^*$ or $2^* \to 2^{\leqslant\omega}$.

Theorem 78. *Let* θ: $2^* \to \mathbb{N}$ *be recursive.*

1. *Let* F: $\mathscr{I} \to 2^{\leqslant\omega}$ *be a total map where* $\mathscr{I}$ is 2^ω *or* $2^{\leqslant\omega}$. *If F is computable* (*resp. semicomputable*) *then* ∂F *and* $\partial^\theta F$ *are computable* (*resp. semicomputable*).

2. *The above result is false when* $\mathcal{I} = 2^*$. *There exists some total recursive* $F\colon 2^* \to \{\lambda, 0\}$ *such that* $\partial F\colon 2^* \to \{\lambda, 0\}$ *is not semicomputable via possibly infinite computation* (*recall* λ *is the empty word*).

Proof. We only consider ∂F for $F\colon 2^{\leqslant\omega} \to 2^{\leqslant\omega}$. The proof for $\partial^0 F$ and/or $F\colon 2^{\omega} \to 2^{\leqslant\omega}$ is quite similar (even simpler when F has domain 2^{ω}).

(1) *Case* $F\colon 2^{\leqslant\omega} \to 2^{\leqslant\omega}$ *is semicomputable.*

Let M be a monotone machine that semicomputes $F\colon 2^{\leqslant\omega} \to 2^{\leqslant\omega}$.

Let $s \in 2^*$. For $t \in \mathbb{N}$, we define $X_t, Y_t \subseteq \{0,1\}^{\leqslant t}$ and $f_t\colon X_t \to 2^*$, $g_t\colon Y_t \to 2^*$ as follows:

(i) We emulate t steps of the computations of M on all possible inputs in $s2^{\leqslant\omega}$. Clearly, at most t letters can be read on the input tape, so that these emulations can be done with all possible finite inputs of length $\leqslant \max(t, |s|)$ which are extensions of s.

(ii) Put u in X_t if the input head of M has moved right to u (hence has scanned the blank symbol which is an end marker for finite inputs) during its first t computation steps on input u.

Clearly, X_t consists of words of length $<t$ which are extensions of s.

(iii) Put u in Y_t if there exists $v \in 2^{\leqslant\omega}$ such that u is the prefix of v read at time t during the computation of M on input v (and there has been no attempt to move right to u).

Clearly, all computations of M on any input in $u2^{\leqslant\omega}$ have exactly the same t first computation steps and Y_t consists of words of length $\leqslant t$ which are prefixes or extensions of s.

(iv) Let $f_t(u), g_t(u)$ be the current output of M at step t in the emulations considered in (ii)–(iii).

Clearly, $t \mapsto X_t$, $t \mapsto Y_t$, $t \mapsto f_t$ and $t \mapsto g_t$ are recursive functions.

We now describe a monotone machine M' to semicompute $\partial F\colon 2^* \to 2^{\leqslant\omega}$.

(1) The computation of M' on input $s \in 2^*$ consists of finitely many or infinitely many successive phases indexed by $t = 0, 1, 2, \ldots$.

(2) During phase t, M' computes X_t, Y_t, f_t, g_t.

(3) At the end of phase t, the current output of M' (on input s) is

$$Output_t^{M'}(s) = gcp(\{f_t(u) :\ u \in X_t\} \cup \{g_t(u) :\ u \in Y_t\}),$$

where *gcp* is the *greatest common prefix* function.

The definition of X_t, Y_t, f_t, g_t show that M' semicomputes ∂F.

(2) *Case* $F\colon 2^{\leqslant\omega} \to 2^{\leqslant\omega}$ *is computable.*

Add the following halting condition to M':

(4) M' halts at the end of phase t if and only if one of the following conditions hold:

(a) Let $w = Output_t^{M'}(s)$. There exist $u_0, u_1 \in X_t \cup Y_t$ such that $w0$ and $w1$ are prefixes of the current outputs of M at time t on inputs u_0, u_1.

(b) At time t, the computation of M on some input $u \in X_t \cup Y_t$ halts with current output equal to $Output_t^{M'}(s)$.

Conditions (4a) and (4b) imply that $\mathit{Output}_t^{M'}(s) = \mathit{Output}_{t'}^{M'}(s)$ for all $t' > t$, so that it is reasonable for M' to halt with its current output.

To prove that M' computes ∂F, it remains to show that if $\partial F(s)$ is finite then one of conditions (4a) and (4b) does hold at some time t.

But $\partial F(s)$ is finite in only two cases:

(i) There are $\xi, \eta \in s2^{\leqslant\omega}$ such that $F(\xi)$ and $F(\eta)$ are incompatible with respect to the prefix ordering on $2^{\leqslant\omega}$.

(ii) Some $F(\xi)$'s, $\xi \in s2^{\leqslant\omega}$, is finite.

If (i) holds then there is some t such that the above condition (4a) holds.

Suppose (ii) holds but (i) does not hold, i.e. all $F(\eta)$'s, $\eta \in s2^{\leqslant\omega}$, are comparable but one of them is finite. Choose $\xi \in s2^{\leqslant\omega}$ such that $F(\xi)$ is finite and has minimum length. Thus, $F(\eta) \succcurlyeq F(\xi)$ for all $\eta \in s2^{\leqslant\omega}$.

Claim. Denote $\mathit{Output}_t^M(\eta)$ the current output of M on input $\eta \in 2^{\leqslant\omega}$ at time t. There exists t such that

$$\forall \eta \in s2^{\leqslant\omega} \quad \mathit{Output}_t^M(\eta) \geqslant F(\xi).$$

Proof of claim. Suppose not and let $(\eta_t)_{t\in\mathbb{N}}$ be such that $\mathit{Output}_t^M(\eta_t) \prec F(\xi)$ for all $t \in \mathbb{N}$. Restricting to some strictly increasing subsequence $(t_i)_{i\in\mathbb{N}}$, one can suppose that $\mathit{Output}_{t_i}^M(\eta_{t_i}) = w$ for all $i \in \mathbb{N}$, where w is some fixed *strict* prefix of $F(\xi)$.

Using compactness of $2^{\leqslant\omega}$ and again restricting to some subsequence, one can suppose that $(\eta_{t_i})_{i\in\mathbb{N}}$ converges towards some $\zeta \in 2^{\leqslant\omega}$.

Case $\zeta \in 2^*$. Then the sequence $(\eta_{t_i})_{i\in\mathbb{N}}$ is stationary: $\exists i_0\ \forall i \geqslant i_0\ \eta_{t_i} = \zeta$. In which case, $\mathit{Output}_{t_i}^M(\zeta) = w$ for all $i \geqslant i_0$, whence $F(\zeta) = w \prec F(\xi)$, contradicting the minimality of $F(\xi)$.

Case $\zeta \in 2^{\omega}$. For all t there exists $t_i > \geqslant t$ such that $\eta_{t_i} \succcurlyeq \zeta \upharpoonright t$, whence $\geqslant t$ such that $\eta_{t_i} \succcurlyeq \zeta \upharpoonright t$, whence

$$\mathit{Output}_t^M(\zeta) = \mathit{Output}_t^M(\zeta \upharpoonright t) = \mathit{Output}_t^M(\eta_{t_i}) \preccurlyeq \mathit{Output}_{t_i}^M(\eta_{t_i}) = w.$$

Therefore $F(\zeta) = w \prec F(\xi)$, again contradicting the minimality of $F(\xi)$.

Let t be as in the claim. Then $\mathit{Output}_{t'}^{M'}(s) \geqslant F(\xi)$ for all $t' \geqslant t$. Since M halts when it has finite output, there exists t_1, such that M on input ξ halts at time $\leqslant t_1$. Therefore, for $t' \geqslant t, t_1$ we have $\mathit{Output}_{t'}^{M'}(s) = F(\xi)$ and condition (4b) holds.

(3) *Case* F: $2^* \to 2^{\leqslant\omega}$.

Compactness of the domain space was crucial in the above proof and, in fact, the result breaks down for maps with domain 2^*.

Let φ: $\mathbb{N}^2 \to \mathbb{N}$ be some universal partial recursive function and define F: $2^* \to 2^*$ as follows (compare with the example of Remark 42):

$$F(0^n) = 0,$$
$$F(0^n 1 s) = \texttt{if}\ \varphi_n(n)\ \text{converges at time} \leqslant |s|\ \texttt{then}\ \lambda\ \texttt{else}\ 0.$$

Then F is total recursive but

$$\partial F(0^n 1) = \texttt{if}\ \varphi_n(n)\ \text{is defined}\ \ \texttt{then}\ \lambda\ \texttt{else}\ 0$$

and ∂F cannot be semicomputable via possibly infinite computations: else, the set of n's such that $\varphi_n(n)$ is *not defined* would be recursively enumerable (as the set of n's such that at some time the computation on input 0^n1 has current output 0), a contradiction. □

8. Representation of (lower semi)continuous and (semi)computable maps

8.1. Extending maps $2^\omega \to 2^{\leqslant\omega}$ to $2^{\leqslant\omega} \to 2^{\leqslant\omega}$

The partial operator introduced in the previous subsection gives an explicit form of the instance of Tietze theorem concerning extensions to $2^{\leqslant\omega}$ of (semi)continuous maps $2^\omega \to 2^{\leqslant\omega}$ together with the (semi)computable version.

Theorem 79. *Let F: $2^\omega \to 2^{\leqslant\omega}$ be a total map. We denote $F \cup \partial F$ the map $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ which extends both F and ∂F: $2^* \to 2^{\leqslant\omega}$.*

If F is continuous (*resp. lower semicontinuous*, *resp. computable*, *resp. semicomputable*) *then so is $F \cup \partial F$.*

Proof. 1. *Lower semicontinuity.* Suppose F: $2^\omega \to 2^{\leqslant\omega}$ is lower semicontinuous. Since every function is continuous at every point in 2^*, it suffices to show that $F \cup \partial F$ is also lower semicontinuous at points $\alpha \in 2^\omega$. The lower semicontinuity of F at α insures

$$\forall n\ \exists p\ \forall \beta \succ \alpha \upharpoonright p\ F(\beta) \succcurlyeq F(\alpha) \upharpoonright n.$$

If $s \in 2^*$ and $s \succcurlyeq \alpha \upharpoonright p$ then

$$\partial F(s) = gcp(F(s2^\omega)) \succcurlyeq gcp(F((\alpha \upharpoonright p)2^\omega)) \succcurlyeq F(\alpha) \upharpoonright n.$$

Thus, $\forall \xi \succ \alpha \upharpoonright p\ (F \cup \partial F)(\xi) \succcurlyeq F(\alpha) \upharpoonright n$, which yields the lower semicontinuity of $F \cup \partial F$ at α.

2. *Continuity.* Suppose F: $2^\omega \to 2^{\leqslant\omega}$ is continuous. We already know that $F \cup \partial F$ is lower semicontinuous. Since every function is continuous at every point in 2^* and every lower semicontinuous function $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ is continuous at every point with image in 2^ω, it suffices to show that $F \cup \partial F$ is also continuous at points $\alpha \in 2^\omega$ such that $F(\alpha) \in 2^*$. In that case, continuity of F at α insures

$$\exists p\ \forall \beta \succ \alpha \upharpoonright p\ F(\beta) = F(\alpha).$$

If $s \in 2^*$ and $s \succcurlyeq \alpha \upharpoonright p$ then

$$\partial F(s) = gcp(F(s2^\omega)) \succcurlyeq gcp(F((\alpha \upharpoonright p)2^\omega)) = F(\alpha).$$

Thus, $\forall \xi \succ \alpha \upharpoonright p\ (F \cup \partial F)(\xi) = F(\alpha)$, which yields the continuity of $F \cup \partial F$ at α.

3. *Semicomputability.* Suppose F is semicomputable, Theorem 78 insures that so is ∂F. Let M and ∂M be monotone Turing machines which semicompute F and ∂F. Define a monotone machine M' such that, on input $\xi \in 2^{\leqslant\omega}$,

(1) M' emulates M and outputs as does M.

(2) If $\xi \in 2^\omega$ or if $\xi \in 2^*$ but the input head never moves right to ξ then this emulation goes on with no restriction (and halts if and only if M halts).

(3) If $\xi \in 2^*$ and at some time t the input head moves right to ξ, then M' stops the emulation of M and starts emulating ∂M and outputs as does ∂M. This emulation of ∂M can be faithful because as long as M does not move right to ξ then M and ∂M have exactly the same behaviour.

Since $\partial F(s) = gcp(s2^\omega)$, we see that, if ∂M is as defined in the proof of Theorem 78, then the current outputs of M and ∂M are identical when (and if) point 3 applies. This shows that M' semicomputes $F \cup \partial F$.

4. *Computability.* If F is computable then so is ∂F and we can suppose that machines M and ∂M do halt when they have finite output. The same is then true for M'. Therefore $F \cup \partial F$ is computable. □

8.2. The representation theorem

An elementary result relates continuous maps F: $2^\omega \to 2^\omega$ to totally unbounded monotone increasing maps f: $2^* \to 2^*$. Moreover, this result has an effective version with F computable and f recursive. Cf. Kechris' book [20] Proposition 2.6, p. 8, or Staiger [39] Proposition 1.6, 2.5 and [41] Theorem 1.1.

Theorem 80. *Let F: $2^\omega \to 2^\omega$. Then F is continuous (resp. computable) if and only if $F = \overline{f}$ for some (resp. recursive) monotone increasing and totally unbounded map f: $2^* \to 2^*$.*

As we shall prove, this characterization extends in different ways:
– to maps $2^\omega \to 2^{\leqslant\omega}$ and $2^{\leqslant\omega} \to 2^{\leqslant\omega}$,
– to continuity and lower semicontinuity,
– to other ways of approximating maps.

And these extensions have effective versions with computable and semicomputable maps.

Theorem 81 (The representation theorem). *Let F: $\mathscr{I} \to \mathbb{O}$ be a total map where $\mathscr{I}, \mathbb{O}$ are among $2^*, 2^\omega, 2^{\leqslant\omega}$.*

The columns of Table 1 give equivalent conditions for F to be continuous or lower semicontinuous or semicomputable or computable in different contexts $\mathscr{I}, \mathbb{O}$ (determined by rows).

Conventions:
– f denotes a monotone increasing map f: $2^* \to 2^*$ or f: $2^* \to 2^{\leqslant\omega}$,
– v denotes a map v: $2^* \times \mathbb{N} \to 2^*$ monotone increasing wrt $\mathbb{N}$,
– $\overline{f}, \overline{v}$ are as in Definitions 59, 64.
– θ: $2^* \to \mathbb{N}$ is some recursive totally unbounded map.

Proof. (1) *Assertions about F: $2^* \to 2^{\leqslant\omega}$.* Straightforward.

(2) *If F: $2^\omega \to 2^\omega$ is lower semicontinuous (resp. semicomputable) then F is continuous (resp. computable).*

Table 1
The representation theorem

	Conventions: f: $2^* \to 2^*$ *is monotone increasing*		v: $2^* \times \mathbb{N} \to 2^*$ *is monot. incr. in its 2d argument*	
F	Lower semicontinuous	Continuous	Semicomputable	Computable
$2^* \to 2^{\leqslant\omega}$	Always trivially true with discrete topologies		$\exists v$ recursive $F = \overline{v}$	$\exists v$ recursive $F = \overline{v}$ *Open*(v) recursive
$2^\omega \to 2^\omega$	Lower semicont. $\Leftrightarrow$	$\exists f\ \ F = \overline{f}$ f totally unbounded	Semicomputable $\Leftrightarrow$	Idem continuous with
	Continuous when *Range*$(F) \subseteq 2^\omega$	$F = \overline{\partial F} = \overline{\partial^\theta F}$	Computable when *Range*$(F) \subseteq 2^\omega$	$f,\ \partial F,\ \partial^\theta F$ Computable
$2^\omega \to 2^{\leqslant\omega}$	$\exists f \quad F = \overline{f}$	$\exists f \quad F = \overline{f}$ f totally unbounded on its tree	Idem semicont. with f semicomputable	Idem *continuous* with f computable and $Open_{<\omega}(f)$ r.e.
	$F = \overline{\partial F}$ (resp. $\overline{\partial^\theta F}$)	$F = \overline{\partial F}$(resp. $\overline{\partial^\theta F}$) $\partial F, \partial^\theta F$ tot. unbdd on their trees	Idem semicont. with ∂F semicomputable (resp. $\partial^\theta F$ semicomp)	Idem continuous with ∂F computable and $Open_{<\omega}(\partial F)$ r.e. (resp. Idem with $\partial^\theta F$)
$2^{\leqslant\omega} \to 2^{\leqslant\omega}$	$\exists f \preccurlyeq F\upharpoonright 2^* \quad F\upharpoonright 2^\omega = \overline{f}$	$\exists f \preccurlyeq F\upharpoonright 2^* \quad F\upharpoonright 2^\omega = \overline{f}$ $F\upharpoonright 2^\omega = \overline{F\upharpoonright 2^*}$ f totally unbounded on its tree	Idem semicont. with $f,\ F\upharpoonright 2^*$ semicomp.	Idem continuous with $f,\ F\upharpoonright 2^*$ comput. $Open_{<\omega}(F\upharpoonright 2^*)$ Recursively enumerable
	$F\upharpoonright 2^\omega = \overline{\partial F}$ (resp. $F\upharpoonright 2^\omega = \overline{\partial^\theta F}$)	$F\upharpoonright 2^\omega = \overline{F\upharpoonright 2^*} = \overline{\partial F}$ ∂F tot. unbdd on its tree (resp. Idem with $\partial^\theta F$)	Idem semicont. with $F\upharpoonright 2^*,\ \partial F$ (resp. $\partial^\theta F$) semicomputable	Idem cont. with $F\upharpoonright 2^*$ ∂F (resp. $\partial^\theta F$) comput. $Open_{<\omega}(F\upharpoonright 2^*)$ r.e.

Cf. Proposition 50, point 2, and Proposition 43.

(3) *If* F: $2^{\leqslant\omega}\to 2^{\leqslant\omega}$ *is lower semicontinuous then* $F\upharpoonright 2^\omega=\overline{\partial^\theta F}=\overline{\partial F}$.

Lower semicontinuity of F at $\alpha\in 2^\omega$ implies that for all n there exists p such that

$$(*)\quad \forall\xi\in 2^{\leqslant\omega}\ (\xi\succcurlyeq\alpha\upharpoonright p\Rightarrow F(\xi)\succcurlyeq F(\alpha)\upharpoonright n).$$

Since θ is totally unbounded, up to some increase of p, one can suppose that $\theta(\alpha\upharpoonright p)\geqslant n$. Thus, (*) yields $\partial^\theta F(\alpha\upharpoonright p)\succcurlyeq F(\alpha)\upharpoonright n$. Since this is true for all n, we get $\overline{\partial^\theta F}(\alpha)\succcurlyeq F(\alpha)$.

Also, $\overline{\partial F}(\alpha)=\lim_{p\to\infty}\partial F(\alpha\upharpoonright p)=\lim_{p\to\infty} gcp(F((\alpha\upharpoonright p)2^{\leqslant\omega}))\preccurlyeq F(\alpha)$.

Since $\partial^\theta F\preccurlyeq\partial F$, we get the equalities $\overline{\partial^\theta F}(\alpha)=\overline{\partial F}(\alpha)=F(\alpha)$.

(3bis) *If* F: $2^\omega\to 2^{\leqslant\omega}$ *is lower semicontinuous then* $F=\overline{\partial^\theta F}=\overline{\partial F}$.

Apply Theorem 79 and point 3 for G: $2^{\leqslant\omega}\to 2^{\leqslant\omega}$ where $G=F\cup\partial F$ and observe that $\partial G=\partial F$.

(3ter) *If* F: $2^\omega\to 2^{\leqslant\omega}$ (*resp.* F: $2^{\leqslant\omega}\to 2^{\leqslant\omega}$) *is semicomputable then so are* $\partial^\theta F$ *and* ∂F (*resp. and also* $F\upharpoonright 2^*$).

Apply Theorem 78. In case F: $2^{\leqslant\omega}\to 2^{\leqslant\omega}$, the semicomputability of F trivially implies that of $F\upharpoonright 2^*$.

(4) *If* F: $2^{\leqslant\omega}\to 2^{\leqslant\omega}$ *is such that* $F\upharpoonright 2^\omega=\overline{f}$ *for some monotone increasing* f: $2^*\to 2^*$ *such that* $f\preccurlyeq F\upharpoonright 2^*$ *then F is lower semicontinuous.*

Moreover, if f and $F\upharpoonright 2^*$ *are semicomputable then F is semicomputable.*

Lower semicontinuity of F at points in 2^* is trivial (Proposition 50). As concerns points $\alpha\in 2^\omega$, equality $F\upharpoonright 2^\omega=\overline{f}$ insures that for all n there exists p such that $F(\alpha)\upharpoonright n\preccurlyeq f(\alpha\upharpoonright p)$ and

(a) $F(\beta)\succcurlyeq F(\alpha)\upharpoonright n$ for all $\beta\in 2^\omega$ such that $\beta\succ\alpha\upharpoonright p$.

Now, since f is monotonous, inequality $f\preccurlyeq F\upharpoonright 2^*$ yields that for all $s\succ\alpha\upharpoonright p$ we have $f(\alpha\upharpoonright p)\preccurlyeq f(s)\preccurlyeq F(s)$, i.e.

(b) $F(s)\succcurlyeq F(\alpha)\upharpoonright n$ for all $s\in 2^*$ such that $s\succ\alpha\upharpoonright p$.

But (a + b) is exactly lower semicontinuity at α.

The assertion about semicomputability follows from Lemma 74.

(4bis) *If* F: $2^\omega\to 2^{\leqslant\omega}$ *is such that* $F=\overline{f}$ *for some monotone increasing* $f:2^*\to 2^*$ *then F is lower semicontinuous.*

Moreover, if f is semicomputable then so is F.

Let $G=F\cup\partial F$: $2^{\leqslant\omega}\to 2^{\leqslant\omega}$. Proposition 70 insures that $f\preccurlyeq\partial F$. Since $\partial G=\partial F$ we get $f\preccurlyeq\partial G$. Thus, we can apply point 4 to f and G. Lower semicontinuity of G on $2^{\leqslant\omega}$ implies that of F on 2^ω.

In case f is semicomputable, apply Theorem 78 and Lemma 74.

(5) *If* F: $2^\omega\to 2^{\leqslant\omega}$ (*resp.* F: $2^{\leqslant\omega}\to 2^{\leqslant\omega}$) *is such that* $F=\overline{\partial^\theta F}$ *or* $F=\overline{\partial F}$ (*resp.* $F\upharpoonright 2^\omega=\overline{\partial^\theta F}$ *or* $F\upharpoonright 2^\omega=\overline{\partial F}$) *then F is lower semicontinuous.*

Moreover, if ∂F *or* $\partial^\theta F$ (*resp. and* $F\upharpoonright 2^*$) *is semicomputable then so is F.*

Apply point 4 or 4bis to $f=\partial^\theta F$ (resp. $f=\partial F$).

(6) *If* F: $2^\omega\to 2^{\leqslant\omega}$ (*resp.* F: $2^{\leqslant\omega}\to 2^{\leqslant\omega}$) *is continuous then* $F=\overline{\partial^\theta F}=\overline{\partial F}$ (*resp.* $F\upharpoonright 2^\omega=\overline{\partial^\theta F}=\overline{\partial F}=\overline{F\upharpoonright 2^*}$).

We first prove the case F: $2^{\leqslant\omega}\to 2^{\leqslant\omega}$. Point 3 insures equalities $F\upharpoonright 2^\omega=\overline{\partial^\theta F}=\overline{\partial F}$. Continuity directly yields $F\upharpoonright 2^\omega=\overline{F\upharpoonright 2^*}$.

As for the case F: $2^\omega \to 2^{\leqslant\omega}$, apply Theorem 79 and the previous case for G: $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ where $G = F \cup \partial F$ and observe that $\partial G = \partial F$.

(6bis) *If F: $2^\omega \to 2^\omega$ (resp. F: $2^\omega \to 2^{\leqslant\omega}$ or F: $2^{\leqslant\omega} \to 2^{\leqslant\omega}$) is continuous then $\partial^0 F$ and ∂F are totally unbounded (resp. totally unbounded on their trees).*

Lemma 71 insures the unboundedness conditions. In case F: $2^\omega \to 2^\omega$, observe that $Treef(f) = 2^*$.

(6ter) *If F: $2^\omega \to 2^{\leqslant\omega}$ or F: $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ is continuous then $\partial^0 F$ and ∂F are computable.*

If F: $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ is continuous then $F\restriction 2^*$ is computable and $Open_{<\omega}(F\restriction 2^*)$ is recursively enumerable.

Apply Theorem 78 and Lemma 77. In case F: $2^{\leqslant\omega} \to 2^{\leqslant\omega}$, computability of $F\restriction 2^*$ is trivially implied by that of F.

(7) *If F: $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ is such that $F\restriction 2^\omega = \overline{f} = \overline{F\restriction 2^*}$ where f: $2^* \to 2^*$ is monotone increasing and $f \preccurlyeq F\restriction 2^*$ and f is totally unbounded on its tree then F is continuous.*

Moreover, if f and $F\restriction 2^*$ are computable and $Open_{<\omega}(F\restriction 2^*)$ is recursively enumerable then F is computable.

Using Proposition 50 and point 4, we already have continuity at points in 2^* and at points in 2^ω having image in 2^ω. Thus, we reduce to prove continuity at points $\alpha \in 2^\omega$ with image in 2^*.

Since $F(\alpha) = \overline{f}(\alpha) \in 2^*$ and f is totally unbounded on $Tree(f)$, there exists p such that $\alpha\restriction p \in Open_{<\omega}(f)$. This means that

(*) F is constant with value $f(\alpha\restriction p)$ on $(\alpha\restriction p)2^\omega$.

Also, equality $\overline{f} = \overline{F\restriction 2^*}$ and Lemma 73 show that

– there exists $q \geqslant p$ such that $(\alpha\restriction q) \in Open_{<\omega}(F\restriction 2^*)$,

– f and $F\restriction 2^*$ are constant and coincide on $(\alpha\restriction q)2^*$.

Therefore

(**) $F\restriction 2^*$ is constant with value $f(\alpha\restriction p)$ on $(\alpha\restriction q)2^*$.

Now, (*) and (**) show that F is constant with value $f(\alpha\restriction p))$ on $(\alpha\restriction q)2^{\leqslant\omega}$, hence F is continuous at α.

The assertion about computability follows from Lemma 75.

(7bis) *If F: $2^\omega \to 2^{\leqslant\omega}$ is such that $F = \overline{f}$ where $f: 2^* \to 2^*$ is monotone increasing and f is totally unbounded on its tree then F is continuous.*

Moreover, if f is computable and $Open_{<\omega}(f)$ is r.e. then so is F.

Note: In case $F = \overline{f}$: $2^\omega \to 2^\omega$ then $Open_{<\omega}(f) = \emptyset$ is trivially r.e.

Apply point 7 for G: $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ where $G = \overline{f} \cup f = F \cup f$. Continuity of G on $2^{\leqslant\omega}$ implies that of F on 2^ω.

In case F: $2^\omega \to 2^\omega$ observe that $Tree(f) = 2^*$.

The assertion about computability follows from Lemma 75.

(8) *If F: $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ is such that $F\restriction 2^\omega = \overline{F\restriction 2^*} = \overline{\partial F}$ and ∂F is totally unbounded on its tree then F is continuous.*

Moreover, if ∂F and $F\restriction 2^*$ are computable and $Open_{<\omega}(F\restriction 2^*)$ is r.e. then F is computable.

If F: $2^\omega \to 2^\omega$ (resp. $F: 2^\omega \to 2^{\leqslant\omega}$) is such that $F = \overline{\partial F}$ and ∂F is totally unbounded (resp. totally unbounded on its tree) then F is continuous.

Moreover, if ∂F is computable (resp. and Open$_{<\omega}(\partial F)$ is r.e) then F is computable. Idem with $\partial^\theta F$ in place of ∂F.

Apply point 7 or 7bis to $f = \partial F$ (resp. $f = \partial^\theta F$). □

9. Traces on 2^* of continuous maps $2^{\leqslant\omega} \to 2^{\leqslant\omega}$

9.1. Checkable maps $2^* \to 2^{\leqslant\omega}$

The topological notions of clopen subsets of $2^{\leqslant\omega}$ (or $\vec{\mathbb{X}} \times 2^*$ where $\vec{\mathbb{X}}$ is any finite product of spaces $\mathbb{N}$ and/or 2^* with the discrete topology) and of continuous maps $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ have no topological counterparts in the *discrete space* 2^*. Nevertheless, they have discrete natural counterparts: that of checkable subset of 2^* (or $\vec{\mathbb{X}} \times 2^*$) (Section 2.2) and of *checkable map* $2^* \to 2^{\leqslant\omega}$, which is obtained via traces and restrictions.

As a corollary of Theorem 81 we obtain the following result which insures that nothing new is got from the traces of lower semicontinuous and semicomputable maps $2^{\leqslant\omega} \to 2^{\leqslant\omega}$.

Proposition 82. *Every* (*semicomputable*) *map* ϕ: $2^* \to 2^{\leqslant\omega}$ *is the restriction to* 2^* *of some lower semicontinuous* (*resp. semicomputable*) *map* F: $2^{\leqslant\omega} \to 2^{\leqslant\omega}$.

Proof. $\Rightarrow$. Let $F\restriction 2^* = \phi$ and $F\restriction 2^\omega = \overline{\partial\phi}$. Then $\partial\phi$ is monotone increasing (Proposition 70) and Theorem 81 insures that F is lower semicontinuous. Also, if F is semicomputable so is $\partial\phi$ (cf. Theorem 78).

$\Leftarrow$. Let $\phi = \partial^\theta F$ where θ: $2^* \to \mathbb{N}$ is recursive and totally unbounded (for instance the length function). □

Things are completely different with continuous and computable maps $2^{\leqslant\omega} \to 2^{\leqslant\omega}$. Their restrictions to 2^* constitute new classes of maps $2^* \to 2^{\leqslant\omega}$, which we name *checkable maps* and *recursively checkable* maps.

The following definition is motivated by the well known property of continuous maps mentioned in Proposition 54, point 1, and its effectivized version Proposition 44.

Definition 83 (Checkable maps). (1) $\phi : 2^* \to 2^{\leqslant\omega}$ is *checkable* (resp. *recursively checkable*) if the relation $\{(s,u) \in 2^* \times 2^*: s \preccurlyeq \phi(u)\}$ is checkable (resp. recursively checkable) relative to its last component, i.e. if $\phi^{-1}(s2^{\leqslant\omega})$ is checkable for all $s \in 2^*$.

In other words, ϕ is checkable (resp. recursively checkable) if there exist sets (resp. recursive sets) $X, Y \subset 2^* \times 2^*$ such that for every $s \in 2^*$, the slices X_s, Y_s are finite and $\phi^{-1}(s2^{\leqslant\omega}) = X_s \cup Y_s 2^*$.

(2) ϕ: $2^* \to 2^{\leqslant\omega}$ is *simply checkable* (resp. *simply recursively checkable*) if $\{(s,u) \in 2^* \times 2^*: s \preccurlyeq \phi(u)\}$ is simply checkable (resp. simply recursively checkable), i.e. if the X set in Point 1 is empty.

9.2. Simple checkability and monotonicity

Proposition 84. *A map ϕ: $2^* \to 2^{\leqslant\omega}$ is simply checkable if and only if it is monotone increasing and checkable.*

Proof. $\Rightarrow$. Let ϕ be simply checkable, and let $\phi(u) \in s2^{\leqslant\omega}$ for some $u, s \in 2^*$. Then $\phi^{-1}(s2^{\leqslant\omega}) = Y2^*$, for some $Y \subseteq 2^*$. If $\phi(u) \not\preccurlyeq \phi(uv)$ then $u \in Y2^*$ but $uv \notin Y2^*$, which is impossible.

$\Leftarrow$. Since ϕ is checkable, we have $\phi^{-1}(s2^*) = X \cup Y2^*$. Since ϕ is monotone increasing, we have $\phi^{-1}(s2^*) = \phi^{-1}(s2^*)2^*$. Therefore $\phi^{-1}(s2^*) = (X \cup Y)2^*$ and ϕ is simply checkable. □

Though the previous result points a relation between monotonicity and checkability, these notions are in no way equivalent.

Proposition 85. *There are monotone increasing maps $\phi: 2^* \to 2^*$ that are not checkable and checkable maps ϕ: $2^* \to 2^*$ that are not monotone increasing.*

Moreover, there are checkable maps ϕ: $2^* \to 2^*$ which have prefix-free range (hence are "nowhere monotone increasing").

Proof. (1) Let $\phi(0^i1s) = 1$ and $\phi(0^i) = 0$. Then ϕ is monotone increasing but $\phi^{-1}(1) = 0^{<\omega}12^*$ is not a checkable set.

(2) Let $\phi(0^i1s) = 0^i$ and $\phi(0^i) = 0^i1$. Then ϕ is not monotone increasing but $\phi^{-1}(0^i2^{\leqslant\omega}) = \{0^i\} \cup 0^i12^*$ and $\phi^{-1}(0^i12^{\leqslant\omega}) = \{0^i\}$ are checkable.

(3) Consider the homomorphisms
– μ: $2^* \to 2^*$ such that $\mu(0) = 00$ and $\mu(1) = 11$,
– ν: $2^* \to 2^*$ such that $\nu(0) = 01$ and $\nu(1) = 10$
and let ϕ: $2^* \to 2^*$ be such that

$$\phi(\lambda) = 01, \quad \phi(s) = \mu(s)\nu(s) \quad \text{for all } s \neq \lambda \quad (\lambda \text{ is the empty word}).$$

The range of ϕ is clearly prefix-free. Let us see that ϕ is checkable.

In fact, observe that every $s \in 2^*$ can be written in one (and only one) of the 16 following forms:

$$\begin{array}{rrrrr}
\lambda & 0 & 1 & & \\
\mu(t) & \mu(t)0 & \mu(t)1 & & \\
\nu(u) & \nu(u)0 & \nu(u)1 & \nu(u)00w & \nu(u)11w \\
\mu(t)\nu(u) & \mu(t)\nu(u)0 & \mu(t)\nu(u)1 & \mu(t)\nu(u)00w & \mu(t)\nu(u)11w
\end{array}$$

where $t, u \neq \lambda$.

Also, for $a = 0, 1$ and $t, u \neq \lambda$, we have
$\phi^{-1}(2^*) = 2^*$, $\phi^{-1}(02^*) = 02^*$, $\phi^{-1}(12^*) = 12^*$,
$\phi^{-1}(\mu(t)2^*) = t2^*$,
$\phi^{-1}(\mu(t)a2^*) =$ IF $a \preccurlyeq t$ THEN $\{t\} \cup ta2^*$ ELSE $ta2^*$,
$\phi^{-1}(\nu(u)v2^*) = \emptyset$ for any $v \in 2^*$,

$\phi^{-1}(\mu(t)\nu(u)2^*)=$ IF $u\preccurlyeq t$ THEN $\{t\}$ ELSE $\emptyset$
$\phi^{-1}(\mu(t)\nu(u)a2^*)=$ IF $ua\preccurlyeq t$ THEN $\{t\}$ ELSE $\emptyset$
$\phi^{-1}(\mu(t)\nu(u)00w2^*)=\phi^{-1}(\mu(t)\nu(u)11w2^*)=\emptyset$.
Therefore, ϕ is recursively checkable, though not simply checkable. □

9.3. Checkability and traces

The following theorem characterizes traces of continuous maps. In particular, point 1 is a functional analog of Proposition 36.

Theorem 86. *Let* ϕ: $2^*\to 2^{\leqslant\omega}$. *The following are equivalent*:
(i) ϕ *is checkable.*
(ii) $\phi\cup\overline{\partial\phi}$: $2^{\leqslant\omega}\to 2^{\leqslant\omega}$ *is continuous.*
(iii) $\phi=F\restriction 2^*$ *for some continuous* F: $2^{\leqslant\omega}\to 2^{\leqslant\omega}$.
(iv) $\partial\phi$ *is totally unbounded on its tree and* $\overline{\phi}=\overline{\partial\phi}$.

Proof. (i) ⇒ (ii). Set $F=\phi\cup\overline{\partial\phi}$.

For every s let X_s, Y_s be finite sets such that $\phi^{-1}(s2^{\leqslant\omega})=X_s\cup Y_s2^*$.

Observe that $F^{-1}(s2^{\leqslant\omega})\cap 2^\omega=Y_s2^\omega$. In fact, for every $\alpha\in 2^\omega$,

$$\begin{aligned}\alpha\in F^{-1}(s2^{\leqslant\omega}) &\Leftrightarrow \overline{\partial\phi}(\alpha)\succcurlyeq s\\ &\Leftrightarrow \exists p\ (\alpha\restriction p)2^*\subseteq\phi^{-1}(s2^{\leqslant\omega})=X_s\cup Y_s2^*\\ &\Leftrightarrow \exists p\ (\alpha\restriction p)\in Y_s2^*\\ &\Leftrightarrow \alpha\in Y_s2^\omega.\end{aligned}$$

Thus,

$$F^{-1}(s2^{\leqslant\omega})=\phi^{-1}(s2^{\leqslant\omega})\cup(F^{-1}(s2^{\leqslant\omega})\cap 2^\omega)=X_s\cup Y_s2^*\cup Y_s2^\omega=X_s\cup Y_s2^\omega$$

is a clopen subset of $2^{\leqslant\omega}$.

Also, $F^{-1}(s)=F^{-1}(s2^{\leqslant\omega})\setminus(F^{-1}(s02^{\leqslant\omega})\cup F^{-1}(s12^{\leqslant\omega}))$ is a boolean combination of clopen sets hence is also clopen.

This proves that F is continuous.

(ii) ⇒ (iii) is trivial.

(iii) ⇒ (i). Assume F is continuous and $\phi=F\restriction 2^*$. Then $F^{-1}(s2^{\leqslant\omega})$ is clopen in $2^{\leqslant\omega}$ hence $\phi^{-1}(s2^{\leqslant\omega})=F^{-1}(s2^{\leqslant\omega})\cap 2^*$ is checkable (cf. Proposition 36).

(i) ⇒ (iii). Assume $\phi^{-1}(s2^{\leqslant\omega})=X\cup Y2^*$ for some finite $X, Y\subset 2^*$ and suppose $(u_n)_{n\in\mathbb{N}}$ is such that $\lim_{n\to\infty}u_n\in 2^\omega$ and $\phi(u_n)\succcurlyeq s$. Then $\forall n\ u_n\in X\cup Y2^*$, and for every n large enough $u_n\in Y2^*$. Since Y is finite, then $\exists y\in Y\ \exists m$ such that $\forall p\geqslant m\ u_p\succ y$.

(ii) ⇔ (iv). Observe that $\partial\phi=\partial(\phi\cup\overline{\partial\phi})$ and apply Theorem 81. □

Remark 87. Equality $\overline{\phi}=\overline{\partial\phi}$ in condition (iv) is necessary. For instance, let ϕ be 0 on 0^* and 1 elsewhere. Then $\partial\phi$ is constant with value λ so that its tree is empty. Also, $\overline{\phi}(0^\omega)=0$ whereas $\overline{\partial\phi}(0^\omega)=\lambda$, so that $\phi\cup\overline{\partial\phi}$ is discontinuous at 0^ω.

Corollary 88. *A monotone increasing map* ϕ: $2^*\to 2^{\leqslant\omega}$ *is totally unbounded on* Tree(ϕ) *if and only if* ϕ *is simply checkable.*

Proof. Apply Proposition 84 and Theorem 86. □

9.4. Recursive checkability and traces

Theorem 89. *Let ϕ: $2^* \to 2^{\leqslant\omega}$. The following are equivalent*:

(i) *ϕ is recursively checkable.*

(ii) *$\phi \cup \overline{\partial\phi}$: $2^{\leqslant\omega} \to 2^{\leqslant\omega}$ is computable.*

(iii) *$\phi = F \restriction 2^*$ for some computable F: $2^{\leqslant\omega} \to 2^{\leqslant\omega}$.*

Proof. (ii) ⇒ (iii) ⇒ (i) is straightforward.

(i) ⇒ (ii). First, notice that $\phi(u)$ is computable as follows:

```
Phase $n$. Test if $u \in X_s \cup Y_s 2^*$ for all words $s$ with length $n$.
IF the above check is true for $s$ THEN let the current output be $s$ and go to Phase
$n+1$ ELSE halt.
```

Also, $\partial\phi(u)$ is computable in a similar way: replace the test $u \in X_s \cup Y_s 2^*$ by the test $u2^* \subseteq X_s \cup Y_s 2^*$.

Finally, observe that $Open_{<\omega}(\phi)$ is recursively enumerable since

$$\begin{aligned} u \in Open_{<\omega} &\Leftrightarrow \exists s\ \forall v\ \ \phi(uv) = s \\ &\Leftrightarrow \exists s\ [(u2^* \subseteq X_s \cup Y_s 2^*) \wedge (u2^* \cap (X_{s0} \cup Y_{s0}2^*) = \emptyset) \\ &\qquad \wedge (u2^* \cap (X_{s1} \cup Y_{s1}2^*) = \emptyset)]. \end{aligned}$$

To conclude, apply Theorem 86. □

We conclude the section with the following representation of recursively checkable maps ϕ: $2^* \to 2^{\leqslant\omega}$ via maps v: $2^* \times \mathbb{N} \to 2^*$ (cf. Section 7.2).

Proposition 90. *If ϕ: $2^* \to 2^{\leqslant\omega}$ is recursively checkable, then $\phi = \overline{v}$ for some v: $2^* \times \mathbb{N} \to 2^*$ which is monotone increasing in $\mathbb{N}$ and such that for each $t \in \mathbb{N}$, the map $s \mapsto v(s,t)$ is recursively checkable.*

Proof. Observe that $\phi^{-1}(s) = \phi^{-1}(s2^{\leqslant\omega}) \setminus (\phi^{-1}(s02^{\leqslant\omega}) \cup \phi^{-1}(s12^{\leqslant\omega}))$. Since $\phi^{-1}(s2^{\leqslant\omega}) = X_s \cup Y_s 2^*$ is clopen, so is $\phi^{-1}(s)$. And there exists finite sets Z_s, T_s such that $\phi^{-1}(s) = Z_s \cup T_s 2^*$ and which are computable from X_s, Y_s, hence computable from s.

Let $v(u,t) = \phi(u) \restriction t$. We write $v_t(s)$ for $v(s,t)$.

For each $t \in \mathbb{N}$, if $t \geqslant |u|$ then $v_t^{-1}(u) = Z_u \cup T_u 2^*$, else $v_t^{-1}(u) = \emptyset$. Thus, $v_t^{-1}(s2^*) = \bigcup_{u \in s2^{\geqslant t-|s|}} v_t^{-1}(u)$. Since the set $\{u \in s2^{\geqslant t-|s|}\}$ is finite and $\{(s,u)\colon \phi(u) \succcurlyeq s\}$ is recursive, v_t is recursively checkable. □

10. Prospective work

The classical notion of Wadge reduction (cf. classical books such as Moschovakis [26] or Kechris [20]) works for any polish spaces, in particular for the compact spaces

2^ω and $2^{\leqslant\omega}$. In a forthcoming paper [1] we study the effectivization of Wadge theory for these particular spaces, where the effectivization of continuous maps are the computable maps. Associated to lower semicontinuous maps into $2^{\leqslant\omega}$ we introduce the notion of *semiWadge reduction*, and its effectivization. All expected results of Wadge theory do hold except for some perturbation at level 2 of the effective Borel hierarchy. This a priori surprising phenomenon is special to the level $n=2$ and related to the fact that 2^* as a subset of $2^{\leqslant\omega}$ is not recursive but merely Σ^0_1.

In the present paper we have studied continuity, lower semicontinuity, and their effectivization, for total maps $\mathscr{I}\to\mathscr{O}$ for $\mathscr{I}$ and $\mathscr{O}$ varying on $2^*, 2^\omega, 2^{\leqslant\omega}$. Since we have considered possibly infinite computations on monotone Turing machines, there is always a limit output so that there is no reason to discard any computation. Thus, such machines compute total maps. However, there are important cases in which partial maps arise. One case is when some extra condition on computations on monotone machines is imposed. For example,

– one can ask that the computation does enter an accepting state at some step, or
– ask that the computation never enters a rejecting state (at which the computation stops but the output currently obtained should be ignored and the input is discarded from the domain), or
– ask that a computation either halts in an accepting state or it goes on forever and enters infinitely often some good state (a Büchi condition).

Some other cases of divergence originate in architectural decisions on Turing machines. If the output head is allowed to move and overwrite with no constraints then the output may suffer of infinite fluctuation. Another source of divergence appears when no blanks (or special symbol outside the input alphabet) are used to delimit finite inputs, so the machine has to realize by itself when to finish reading the input tape. If the machine tries to read beyond the last symbol of the input then the computation diverges. The maps corresponding to these computations have prefix-free domains. This restriction has been independently introduced for maps $2^*\to 2^*$ by Levin [24] and Chaitin [8] for a notion of program-size complexity suitable for a definition of randomness. Also Chaitin and Solovay [9,37] have considered possibly infinite computations, hence maps $2^*\to 2^{\leqslant\omega}$ with prefix-free domains. In a subsequent paper [2] we study continuity and computability of partial maps in the spaces $2^*, 2^\omega, 2^{\leqslant\omega}$.

The above mentioned forthcoming papers together with the present work give the background theory for the results on randomness we prove in [3], where we give sufficient conditions on a given set $\mathscr{O}\subset 2^{\leqslant\omega}$ such that the probability that a universal monotone Turing machine gives an output in $\mathscr{O}$ is random relative to the first jump of the halting problem.

Acknowledgements

We thank Max Dickmann for vivid discussions on different topologies for the $2^{\leqslant\omega}$ space and an anonymous referee for pointing references related to computable analysis and the material in Section 5.

References

[1] V. Becher, S. Grigorieff, Recursion and topology on $2^{\leqslant\omega}$ (II): effective wadge reductions, in preparation.

[2] V. Becher, S. Grigorieff, Recursion and topology on $2^{\leqslant\omega}$ (III): partial maps, in preparation.

[3] V. Becher, S. Grigorieff, $\emptyset'$-Random reals and outputs of possibly infinite computations, submitted for publication.

[4] L. Boasson, M. Nivat, Adherences of languages, J. Comput. System Sci. 20 (1980) 285–309.

[5] N. Bourbaki, Topologie générale, Livre III, Chap. 1, 4ème éd., Hermann, Paris, 1965. English translation available.

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

[7] V. Brattka, Bibliography on Constructivity, Computability and Complexity in Analysis, http://cca-net.de/publications/bibliography.html.

[8] G.J. Chaitin, A theory of program size formally identical to information theory, J. ACM 22 (1975) 329–340, http://www.cs.auckland.ac.nz/CDMTCS/chaitin/.

[9] G.J. Chaitin, Algorithmic entropy of sets, Comput. Math. Appl. 2 (1976) 233–245, http://www.cs.auckland.ac.nz/CDMTCS/chaitin/#PL.

[10] J. Duparc, Wadge hierarchy and Veblen hierarchy, Part I: Borel sets of finite rank, J. Symbolic Logic 66 (1) (2001) 56–86.

[11] J. Engelfriet, H.J. Hoogeboom, X-automata on ω-words, Theoret. Comput. Sci. 110 (1993) 1–51.

[12] Y. Ershov, Computable functionals of finite type, Algebra and Logic 11 (4) (1972) 203–242.

[13] M. Ferbus-Zanda, S. Grigorieff, Refinement of the "up to a constant" ordering using constructive co-immunity. Application to the oracular *Min*/*Max* hierarchy of Kolmogorov complexity, in preparation.

[14] R. Freund, L. Staiger, Numbers defined by Turing machines, Collegium Logicum, Annals of the Kurt Gödel Society, Vol. 2, 1996, pp. 118–137. http://www.informatik.uni-halle.de/∼staiger/.

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

[16] A. Grzegorczyk, On the definition of computable functionals, Fund. Math. 42 (1955) 232–239.

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

[18] T. Head, The adherences of languages as topological spaces, in: M. Nivat, D. Perrin (Eds.), Automata and Infinite Words, Lecture Notes in Computer Science, Vol. 192, Springer, Berlin, 1985, pp. 147–163.

[19] T. Head, The topological structure of adherence of regular languages, RAIRO, Theoret. Inform. Appl. 20 (1986) 31–41.

[20] A.S. Kechris, Classical Descriptive Set Theory, Springer, Berlin, 1995.

[21] K. Kuratowski, Topology, Vol. 1, Academic Press, New York, 1966.

[22] D. Lacombe, Extension de la notion de fonction récursive aux fonctions d'une ou plusieurs variables réelles I, II, III, C. R. Acad. Sci. Paris. 240 (1955) 2478–2480;
D. Lacombe, Extension de la notion de fonction récursive aux fonctions d'une ou plusieurs variables réelles I, II, III, C. R. Acad. Sci. Paris. 241 (1955) 13–14 and 151–153.

[23] D. Lacombe, Quelques procédés de définition en topologie récursive, in: A. Heyting (Ed.), Constructivity in Mathematics, North-Holland, Amsterdam, 1958, pp. 129–158.

[24] L. Levin, On the notion of random sequence, Soviet Math. Dokl. 14 (5) (1973) 1413–1416.

[25] M. Li, P. Vitanyi, An Introduction to Kolmogorov Complexity and its Applications, 2nd Edition, Springer, Amsterdam, 1997.

[26] Y.N. Moschovakis, Descriptive Set Theory, North-Holland, Amsterdam, 1980.

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

[28] P.G. Odifreddi, Classical Recursion Theory, Vol. 1, North-Holland, Amsterdam, 1989.

[29] D. Perrin, J.E. Pin, Infinite Words, Pure and Applied Mathematics, Vol. 141, Academic Press, New York, 2004.

[30] R.S. Pierce, Compact zero-dimensional metric spaces of finite type, Mem. Amer. Math. Soc. 130 (1972) 1–64.

[31] R. Redziejowski, Infinite word languages and continuous mappings, Theoret. Comput. Sci. 43 (1986) 59–79.

[32] H. Rogers Jr., Theory of Recursive Functions and Effective Computability, 2nd Edition, 1987, McGraw-Hill, New York, 1967.

[33] C.P. Schnorr, Process complexity and effective random tests, J. Comput. System Sci. 7 (1973) 376–388.

[34] C.P. Schnorr, A survey of the theory of random sequences, in: R.E. Butts, J. Hintikka (Eds.), Basic Problems in Methodology and Linguistics, D. Reidel, Dordrecht, 1977, pp. 193–210.

[35] J.R. Shoenfield, On degrees of unsolvability, Ann. Math. 69 (1959) 644–653.

[36] J.R. Shoenfield, Recursion Theory, Lecture Notes in Logic, Vol. 1, 1993 (reprinted 2001) A. K. Peters, Natick, Massachussets.

[37] R.M. Solovay, On random r.e. sets, in: A.I. Arruda, N.C.A. da Costa, R. Chuaqui (Eds.), Non-Classical Logics, Model Theory and Computability, North-Holland, Amsterdam, 1977, pp. 283–307.

[38] L. Staiger, Hierarchies of recursive ω-languages, J. Inform. Process. Cybernet. EIK 22 (5/6) (1986) 219–241.

[39] L. Staiger, Sequential mappings of ω-languages, J. Inform. Process. Cybernet. EIK 23 (8/9) (1987) 415–439.

[40] L. Staiger, ω-languages, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Vol. 3, Springer, Berlin, 1997, pp. 339–387, http://www.informatik.uni-halle.de/∼staiger/.

[41] L. Staiger, On the power of reading the whole input tape, in: C.S. Calude, Gh. Pau (Eds.), Finite Versus Infinite: Contributions to an Eternal Dilemma, Discrete Mathematics and Theoretical Computer Science, Springer, Berlin, 2000, pp. 335–348. Preprint 99-15, 1999, http://www.informatik.uni-halle.de/∼staiger/.

[42] L. Staiger, K. Wagner, Rekursive Folgenmengen I, Z. Math. Logik Grundlagen Math. 24 (1978) 523–538.

[43] A. Turing, On computable numbers, with an application to the Entscheidungsproblem, Proceedings of the London Mathematical Society, 2nd Series, Vol. 42, 1936, pp. 230–265, Correction, Ibid, 43: 544–546, 1937.

[44] K. Wagner, Arithmetische Operatoren, Z. Math. Logik Grundlagen Math. 22 (1976) 553–570.

[45] K. Wagner, L. Staiger, Recursive ω-languages, FCT'77, Lecture Notes in Computer Science, Vol. 56, 1977, pp. 532–537.

[46] K. Weihrauch, Type 2 recursion theory, Theoret. Comput. Sci. 38 (1985) 17–33.

[47] K. Weihrauch, Computability, Springer, Berlin, 1987.

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

[49] K. Weihrauch, Computable analysis, An introduction, Springer, Berlin, 2000.

[50] K. Weihrauch, C. Kreitz, Theory of representations, Theoret. Comput. Sci. 38 (1985) 35–53.

[51] K. Weihrauch, C. Kreitz, Type 2 computational complexity of functions on Cantor space, Theoret. Comput. Sci. 82 (1991) 1–18.

[52] N. Zhong, K. Weihrauch, Computability theory of generalized functions, J. ACM 50 (4) (2003) 469–505.